Twinwright Studio 1.2.1

同匠 Studio 使用说明

2026-10-16

目录

用户手册

1. 开始使用
2. 窗口和工作区
3. 机器和上电
4. 装夹
5. 用测头找工件零点
6. 出程序（CAM）
7. 程序和编辑器
8. 运行程序和断点续跑
9. 分析和诊断
10. 在孪生里跑程序
11. 检验、检验报告和装夹单
12. 连真机
13. AI 助手
14. 语言和单位
15. 键盘和无障碍
16. 常见问题

运维手册

1. 系统要求
2. 安装和卸载
3. 授权、席位和激活
4. 你的数据和备份
5. 更新 Studio
6. 隐私和 Studio 发出的东西
7. 实验室、教室和共用的电脑

技术手册

1. 孪生的模型和它们的局限
2. 同匠自研内核
3. G 代码和 M 代码
4. CAM 策略和参数
5. 配合、跳动和检验方法
6. 机床档案
7. MC-1 硬件

接口文档

1. MCP tool reference
2. 设置
3. 报警码
4. 错误码
5. Command line
6. Machine profile schema
7. 协议
8. 文件格式
9. 用发送软件连同匠自研内核

用户手册

开始使用

同匠 Studio 运行你的数控机床的数字孪生。你把零件装夹好，做出或打开程序，先在孪生里运行，再让真机切削。这一章带你从安装走到第一个零件，大约十分钟。

安装

Studio 需要 64 位的 Windows 10 或 11（见系统要求）。

1. 运行 `TwinwrightStudio-<version>-setup.exe`。不需要管理员权限：Studio 只为你自己安装。
2. 阅读许可协议，单击 **我接受。**
3. 保持默认的文件夹，或另选一个。
4. 在最后一页，保持勾选 **运行 同匠 Studio**，单击 **完成。**

开始菜单里的名字是 **同匠 Studio。**给很多台电脑安装，以及什么文件放在哪儿：见安装和卸载。

第一次启动

Studio 在 **运行** 工作区打开，机器 **MC-1 · 我的工作台** 已经开着。什么都不会问你。

机床条显示 **ALARM:11** 和 **下一步：回零。**这是正常的：和真机一样，孪生要回零之后才知道各轴在哪儿。

窗口显示什么

新装的 Studio 显示工程师用到的一切：六个工作区、曲线和着色。如果你只运行程序，或者在教学，就选一个显示得少一些的级别：**工具 > 界面层级**（见用户级别）。

语言和单位

Studio 启动时用你系统的语言，前提是它有这种语言（英语、中文、法语、德语、日语或西班牙语）；没有就用英语。单位和数字的写法跟着你的地区。

要改它们：**工具 > Language / 语言** 和 **工具 > 单位。**见语言和单位。

档位和试用

你现有的档位显示在顶栏里，例如 **学习版。**要查看，单击它，或选择 **帮助 > 授权…。**

档位	多了什么
学习版	型腔、轮廓、钻孔、刻字和三维平行精加工；整个孪生、编辑器和 AI
创客版	自适应开粗、全部三维精加工方式、残料加工、两面加工和多次装夹
专业版	排料、程序优化器、报告（检验报告、装夹单）、用测头检验（与设计对比）

每一档都有表里排在它上面各档的全部功能。更高一档的功能带着那一档的名字显示。单击它，可以读到它做什么，也可以开始试用或打开授权。

试用或授权结束后，Studio 回到学习版。你的项目、程序和已经生成的 G 代码照常能用。

开始试用

创客版和专业版各可以试用 30 天，每台电脑一次。

1. 单击这一档的一个功能，例如 **程序** 工作区里的 **优化。**
2. 单击 **开始 30 天试用…**
3. 单击 **开始试用。**

Studio 向同匠的授权服务器申请一次试用，只发送所试用的版本档、这台电脑指纹的散列和 Studio 的版本。试用结束时，**消息** 页里有一条通知。

激活授权

选择 **帮助 > 授权…。**

在线激活：

1. 输入激活码（`TW-XXXXX-XXXXX-XXXXX-XXXXX`）。大小写、空格和横线都不要紧。
2. 单击 **激活。**结果：**已激活。**

在不能上网的电脑上激活：

1. 在 **离线激活** 下面，复制这台电脑的请求码。
2. 在一台能上网的电脑上，打开授权网页，粘贴请求码，下载授权文件（`.twlic`）。
3. 把文件带回来，单击 **导入授权文件…**

到期前自动续期 在授权到期之前在线续期。它只发出授权编号、这台电脑指纹的散列和 Studio 的版本。

席位、换到另一台电脑和教室：见授权、席位和激活。

十分钟做出第一个零件

最快的第一个零件是一个示例零件：Studio 替你设好毛坯、刀具和工件零点。

打开机器

机器已经开着：**MC-1 · 我的工作台**，在 **运行** 工作区。别的机器在 **文件 > 机器库…** 里。

上电和回零

按 **回零**（F3），或者按机床条建议的那个键。

结果：三个轴一个接一个回零，大约 20 s。机床条显示 **无报警**。

如果机床条显示 **急停**，先把它松开（F1），再按 **上电**（F2）和 **复位**（F12）。

打开一个示例程序

1. 选择 **文件 > 示例零件…**
2. 选择 **步进电机安装板（NEMA 17）**，示例编号 `e01_motor_mount`：配合孔、沉孔和长槽，最后留桥切下。结果：Studio 放好毛坯，装入刀具和程序，并把工件零点设在毛坯上。示例准备好时，控制台里会有一行说明。

在孪生里运行

1. 按 **循环启动**（F5）。
2. 想让它跑得快一些，选择 **仿真 > 倍速 > 实时 ×10**。

三维视图里能看到刀具在切毛坯。**进给保持**（F6）让程序暂停，**循环启动** 接着走。

查看结果

- 在三维视图里拖动来转动视图，滚轮缩放。
- **视图 > 着色 > 工件：与零件的偏差** 给余量（还没切掉的料）和任何过切着色。
- **分析**（Alt+5）显示时间花在了哪些行，以及孪生对切削力和颤振的估计。

设置毛坯

做你自己的零件时，不要载入示例，改在 **准备**（Alt+1）里设置机床：

1. 转到 **装夹** 步骤。
2. 在 **工件与装夹** 下面，选择 **形状**、**材料** 和 **夹具**，再输入尺寸。
3. 单击 **装上新料**。

见装夹。

测量刀具

1. 转到 **对刀** 步骤，选好刀具，单击 **换刀**。
2. 单击 **测参考刀**。

机床必须已经回零，主轴必须已经停转。

设置工件零点

1. 转到 **零点** 步骤，把刀具点动到你要当作零点的位置。
2. 单击 **全部清零**，或者逐个轴清零。

有测头的话，测头卡会替你找零点（见用测头找工件零点）。

然后打开你的程序（**文件 > 打开程序…**），像上面那样运行它。

接下来看什么

- 窗口和工作区：什么在哪儿。
- 出程序（CAM）：Studio 里的 CAM。
- 在孪生里跑程序：孪生检查什么、估计什么。
- 运行程序和断点续跑：运行、报警和重新开始。

面板上的 **?** 会打开本手册里这个面板所在的那一章。**帮助 > 文档** 从头打开手册。

窗口和工作区

Studio 的窗口顶上是菜单和顶栏，中间是六个工作区，底下是机床条。每个工作区只做一件主要的事。不管你在哪个工作区，机床条都显示机床的状态和下一步。

窗口的各部分

从上到下：

- **标题栏**：左边是菜单，中间是窗口的名字，右边是窗口按钮；
- **顶栏**：机器、工作区标签、档位、AI 的状态、级别和 **?**；
- **工作区**：三维视图，和你所选工作区的面板；
- 三维视图下面的 **停靠区**（不是每个工作区都有）；
- **机床条**，每个工作区都一样。

Studio 没有状态栏，状态栏的作用由机床条承担。

标题栏和菜单

菜单有 **文件**、**编辑**、**视图**、**机器**、**程序**、**仿真**、**AI**、**工具** 和 **帮助**。

- 按 Alt 加菜单名后面括号里的字母来打开它：Alt+F 打开 **文件**。
- 单按 Alt，菜单标题会亮起，再用方向键、Enter 和 Esc 操作。

菜单打开时，没有任何键会传到机床。菜单只显示你的级别该有的项（见用户级别）。

顶栏

从左到右：

- 品牌标志：它打开 **文件 > 机器库...**；
- 机器：它的型号、名字和状态。单击它，或按 Ctrl+Shift+M，切换机器：输入文字筛选，用 ↑ ↓ 和 Enter 选择；
- 工作区标签（见六个工作区）；
- 档位，例如 **学习版**：单击它打开 **帮助 > 授权...**；
- AI 的状态：**AI：关**、**AI 已开**、**AI 已接入** 或 **AI 等你同意**。单击它打开 AI 的设置（见AI 助手）；
- 级别：单击它选另一个级别；
- **?**：快捷键列表。

机床条

机床条横贯窗口的最底部。从左到右：

1. **状态**：**空闲**、**运行**、**点动**、**回零中**、**暂停**、**安全门**、**换刀**、**检查模式**、**休眠**、**报警**、**急停** 或 **断电**。
2. **程序**：它的名字、正在执行的行、所在的段落、进度条，以及已运行的时间和剩余的时间。没有程序时，显示 **没有加载程序**、机床是否已回零、刀具和工件坐标。
3. **下一步**：绿色的 **无报警**，或者 **下一步：** 加上做这一步的键。报警时，它显示报警、下一步和 **全部步骤**，后者列出回到能干活的各步（见报警和怎么办）。
4. **时钟**：仿真的时间、它的倍速，和实际达到的速度。沙漏键让仿真暂停。
5. **按键**：**单段**、**循环启动**（F5）、**进给保持**（F6）和 **停止**（F7）。Studio 建议的键是实心的；现在不能用的键变灰。
6. **急停**（F1），最右端的红色圆键。

窗口变窄时，机床条先省掉时钟的时间，再省掉程序的段落和时间（它们转到提示里），再省掉时钟的速度，最后省掉下一步的文字。

三维视图

三维视图显示机床、工件和刀具现在的样子。

- 按住左键拖动来转动视图，按住右键或中键拖动来平移视图。滚轮朝着指针所在处缩放。
- 左上角的键选择视图：轴测、正视、右视或俯视。**跟随刀具** 让刀具保持在中间。**显示** 打开和关闭刀尖轨迹和程序路径。
- **刀具特写** 在右上角的小图里显示切削处的刀具。程序运行时，它出现在 **运行** 工作区里。
- 底部的小标签写明工件和刀具。
- 单击工件或轨迹：视图下面的条会告诉你这一点是哪一行程序切出来的，程序清单跳到那一行。

同样的选择也在 **视图 > 三维视图** 里。**视图 > 着色** 按进给速度、切削力、切屑厚度、颤振裕度或程序行给轨迹和工件着色（工程师级别及以上）。

停靠区

停靠区在三维视图下面。它有哪些页，取决于工作区：

工作区	页
运行	**程序**、**曲线**、**控制台**、**消息**、**AI 活动**
准备	**量件结果**（量过件以后）、**刀具表**
CAM	**检查结果**、**AI 活动**、**版本**、**程序预览**
程序	**控制台**

分析、**诊断** 和车间模式没有停靠区。

- **程序**：程序清单。**跟随当前行** 让正在运行的那一行始终在视野里；用时条显示每一行在这次运行或上一次运行里用的时间。
- **曲线**：这次运行的读数随时间的变化。
- **控制台**：Studio 和控制器说过的话。在底部的输入行里输入一行 G 代码或一条 `$` 命令，按 Enter 发送；↑ 和 ↓ 调出之前输入的行。
- **消息**：警告和发现的问题，按来源分组。页标签上的红色数字是警告的条数。
- **刀具表**：机床的刀具。
- **量件结果**：上一次用测头量的件，一项一行——设计、公差、量到、偏差和孪生的估计（见检验）。
- **AI 活动**：AI 客户端在这台电脑的机器上做过什么。

按 Ctrl+J，或单击页标签右端的箭头，把停靠区折叠成只剩页标签那一行。

六个工作区

用顶栏里的标签、Alt+1 到 Alt+6、Ctrl+Tab 和 Ctrl+Shift+Tab，或者 **视图** 菜单来切换工作区。在 **CAM** 或 **程序** 里按 **循环启动**，会带你到 **运行**。程序运行时，**运行** 标签上有一个绿点。

准备

准备（Alt+1）一步一步让机床和零件就绪：**上电**、**回零**、**装夹**、**对刀**、**零点**、**程序** 和 **加工**。右边的面板跟着下一步走，直到你自己选一步。见装夹；机器、上电和回零见机器和上电。

CAM

CAM（Alt+2）出程序：左边是作业的装夹、形状和工序，右边是所选项的各个字段，底部是 **生成程序**。见出程序（CAM）。工程师级别及以上。

程序

程序（Alt+3）里有程序和示例零件、编辑器，以及一个带程序检查的预览。见程序和编辑器。

运行

运行（Alt+4）运行装入的程序。右栏显示位置、主轴与进给、倍率、切削、面板按键、手动控制，以及刀具与零点。见运行程序和断点续跑。

分析

分析（Alt+5）回头看运行：它们的曲线、每一行的用时和切削力，以及稳定性叶瓣。工程师级别及以上。见分析和诊断。

诊断

诊断（Alt+6）显示机床的电源链、控制器的输入和驱动器，还能让你往孪生里注入故障。技师级别及以上。见分析和诊断。

消息和下一步

Studio 在这几个地方告诉你出了什么事，以及该怎么办：

- 机床条：下一步，或者报警和回到能干活的各步；
- 停靠区的 **消息** 页：警告和发现的问题；
- **控制台**：控制器和 Studio 发来的每一行；
- **工具 > 审计日志…**（Ctrl+L）：所有事情，包括谁在什么时候做的。

机器 > 做下一步 做机床条上显示的那一步，只做这一步。报警让程序停下时，这些步骤的末尾是重新开始的那一行（见从某一行重新开始）。

用户级别

级别显示一个人在那个阶段需要的东西，把其余的藏起来。共有五个级别：

级别	多了什么
演示	**运行** 工作区和示例零件；**程序** 和 **消息** 页
操作员	**准备** 和 **程序**；倍率、面板、手动控制、刀具与零点；**机器** 菜单的各个子菜单；控制台、刀具表、**AI 活动**；设置、审计日志、**AI** 菜单；切换机器
技师	**诊断**；摆放工件；**恢复出厂…**
工程师	**CAM** 和 **分析**；切削读数、曲线、**着色**、导出曲线、用来对比的零件
研究员	**诊断** 里的 **模型与数据** 卡

每个级别都包含表中排在它上面各级别的全部内容。新装的 Studio 从 **工程师** 级别开始。

要改级别，用 **工具 > 界面层级**、顶栏里的级别，或 **工具 > 设置… > 常规**。

> **说明**
>
> 级别是为教和学安排窗口的，它不锁任何东西：机床按键在每个级别都能用。

浅色、深色和车间模式

用 **视图 > 外观** 选择 **跟随系统**、**浅色** 或 **深色**，也能打开或关闭 **车间模式（大号）**。这是两个互相独立的选择：车间模式在哪种外观下都能用。

Studio 自己的面板立刻换成新外观。由系统绘制的列表和文本框，下次启动才换。

车间模式用在机床旁边，屏幕离你一臂远：

- 机床条、它的按键和急停变大，文字和程序清单也变大；
- **运行** 有自己的布局：左边是位置、程序和一个大的程序清单，三维视图下面是表盘，右边是倍率、面板按键和消息，没有停靠区；
- 三维视图只有两个视图键，没有刀具特写。

面板和它们的大小

拖动两个面板之间的边来改变它们的大小；双击这条边回到设计大小。三维视图在 100 % 显示缩放时至少保持 360 × 240 像素。面板的边只能用鼠标移动。**视图 > 恢复默认布局**一次把所有工作区的面板都恢复到设计大小。

Studio 为每台机器记住面板的大小、停靠区的页和它是否折叠，以及你所在的工作区。它还为下次启动记住窗口的位置和大小。

机器和上电

同匠 Studio 里的机器、怎么在机器之间切换，以及一台机器开工前怎么上电、回零。

机器

每台机器有自己的世界：位置、工件、刀具和偏置、温度，以及运行小时数。窗口里的那台机器在运行；其他的机器按存档时的样子等着。1.0 里同一时间只运行一台机器。

切换机器

1. 点顶栏里的机器，或按 Ctrl+Shift+M（**文件 > 切换机器…**）。
2. 输入文字查找机器，用 ↑ ↓ 选中，再按 Enter。结果：Studio 把你离开的机器存起来，再用另一台重新启动。那台机器是冷启动：给它上电，再回零。

程序运行时不能切换。列表里还有 AI 客户端在后台使用的 **练习机**；**查看（只读）** 在 **AI 活动** 里显示它们。

机器库

选 **文件 > 机器库…**，或者点品牌标志，就能看到你的机器，以及它们的状态、工件和刀具，还有机型。再点一次品牌标志即可返回。

对选中的机器：

- **打开**：切换到它。
- **重命名…**：给它起一个你自己的名字。
- **复制为新机器…**：新建一台机器，世界是这台机器上次保存时的样子。
- **在文件夹中显示**：它的世界在磁盘上的位置。
- **恢复出厂…（5 秒内点两次）**：机器回到出厂时的样子（见让一台机器从头开始）。

新建机器… 添加一台这个机型的机器，处于出厂状态。新建机器、复制和恢复出厂需要技师级别及以上。**实机 MC-1** 这一行是给以后留的位置：1.0 里真机从命令行连接（见连真机）。

上电和回零

上电

孪生里有 MC-1 的电源链：总电源 QF1 给控制器供电（24 V）；**上电** 使接触器 KM1 吸合，驱动器得到 48 V，主轴驱动（变频器）也得电。急停使 KM1 断开。

1. 转到 **上电** 步骤。
2. 总电源断开时，点 **合上总电源**。
3. 如果急停键显示 **已按下**，说明急停被按下了：点它，或按 F1，把它松开。
4. 点 **上电**（F2）。结果：面板显示 48 V 有电，每个驱动器都 **就绪**。下一步是 **回零**。

回零

点 **回零**（F3）。先是 Z 轴向上，然后 X 和 Y 一起，各自走到行程一端的开关；实时运行时大约要 20 s。

结果：**已回零：X Y Z**，软限位生效。上电之后、回零之前，控制器处于 ALARM:11；对刀仪、测头和 **回到零点** 要等回零之后才能用。

软限位和行程

MC-1 的行程是 X 200 mm、Y 150 mm、Z 80 mm。回零之后，每个轴的机床坐标在它的回零端为 0：X 从 −200 到 0，Y 从 −150 到 0，Z 从 −80 到 0，Z 的 0 在最高处。

- 会越过限位的点动不执行（error:15）。按住的点动在限位前一点停下。
- 会越过限位的程序行让程序停下，报 ALARM:2。位置保留：先 **复位**、**解锁**，再检查程序和工件零点。
- 轴碰到限位开关时报 ALARM:1：**复位**、**解锁**，把轴点动离开开关，再 **回零**。

预览：这一页的安全提示还没有经过母语审校。

装夹

准备 工作区一步一步把机床和零件准备好：上电、回零、毛坯和它怎么夹、刀具、工件零点。每一步先在孪生里显示出来，机床动之前你就能看到装夹的样子。机器、上电和回零单独成章：机器和上电。

准备工作区

准备（Alt+1）左边是各个步骤，中间是三维视图，右边是所选步骤的面板。视图下面的停靠区里有 **刀具表**。

步骤

步骤	面板	完成条件
上电	总电源、**上电**、**复位**、**解锁**	控制器在运行，驱动器有电
回零	**回零**、**解锁**、**复位**	三个轴都已回零
装夹	毛坯、怎么夹、夹在哪里	毛坯已装好，或者工作台留空
对刀	主轴上的刀、对刀仪	参考刀已量好、刀具表已载入，或者工件 Z 零点已设好
零点	点动、工件零点、测头卡	工件坐标已设好
程序	**打开程序…**、**示例零件…**、**检查程序**	程序已载入
加工	**循环启动**、**单段**	程序已运行到结束

做完的步骤有绿色的对勾；下一步的末端有一个小箭头。面板显示下一步，直到你点另一个步骤；点一下有箭头的那个步骤，面板就又跟着走。步骤下面那一行说明现在该做什么。面板上的 **?** 打开本手册里讲这个面板的那一部分。

装夹时的三维视图

你改毛坯的各个字段时，正在设置的毛坯半透明地画在工作台上：能这样装时是绿色，不能时是红色。**装上新料** 把它放到那里；**还原** 或按 Esc 放弃这次改动。

毛坯

形状、材料和尺寸

在 **装夹** 步骤的 **工件与装夹** 卡里：

1. 选 **形状**：**方块**、**圆棒（竖放）**、**圆管（竖放）**、**六角棒（竖放）**、**圆棒（横放）** 或 **STL 文件（导入外形）…**。
2. 选 **材料**，或者输入它在别的标准里的名称（C45、EN AW-6061、SUS304）：列表里显示 Studio 取其切削数据的牌号，装夹里保留你写的名称。
3. 输入这个形状需要的尺寸：**长 L**、**宽 W**、**高 H**、**直径 D** 等。
4. 选好怎么夹、夹在哪里（见下面各节），再点 **装上新料。**结果：面板写出工作台上是什么，以及它的质量。

尺寸按显示单位填；用英寸时，字段也接受分数（见英寸分数）。

> **说明**
>
> 别的标准里的名称只对应一个相近的牌号：切削数据取自列表里最接近的牌号。材料到底是什么，以供应商的证书为准。

用 STL 文件做毛坯

在形状列表里选 **STL 文件（导入外形）…**，再选文件（单位 mm）。Studio 以外形的最低点为底面，并把它居中；包围盒给出尺寸。悬伸下面的料算作实心。文件读不了时，控制台会说明原因。

夹住毛坯

选 **夹具**：**压板 · 两端**、**压板 · 两侧**、**压板 · 四边**、**平口钳**、**V 型块（横放圆棒）**、**双面胶**、**不固定（只放着）** 或 **理想固定（脚本用）**。

毛坯和它的夹具必须留在工作台上（300 × 220 mm），并避开工作台左后角的对刀仪。程序运行时，孪生把切削力和夹具夹得住的力相比：余量小时给出警告；毛坯一滑动，它在孪生里也跟着动，所以之后切出来的就偏了。

平口钳和平行垫铁

露出钳口 是毛坯顶面高出钳口多少：不改就是 5 mm。Studio 会选能垫出至少这个高度的平行垫铁。平口钳的开口范围是 3.25–77 mm，钳口至少要夹住毛坯 2 mm。

压板

压板装在工作台螺孔（M6，25 mm 网格）里的螺柱上：**两端** 沿毛坯的长度方向，**两侧** 横跨毛坯，**四边** 两个方向都有。每块压板需要在毛坯边缘外 7–35 mm 处有一个螺孔；没有螺孔时，面板会指出找不到螺孔的那块压板。横放的圆棒用 V 型块或平口钳。

双面胶

双面胶按毛坯贴在工作台上的面积夹持毛坯。用于轻切削和扁平的零件，不要用于横放的圆棒。双面胶一旦松脱，就只剩摩擦力夹着毛坯。**不固定（只放着）** 只靠毛坯的自重和摩擦；**理想固定（脚本用）** 永不松动，用于脚本和测试。

料面上方的空间

CAM 写出的程序这样使用料面上方的空间：

- 程序在 Z 行程的最高处开始、结束和换刀。
- 两道工序之间，刀在安全高度上横移：料面以上 5 mm；压板比毛坯高时，压板以上 2 mm。
- 装夹留的空间不够时（刀长、平口钳高、毛坯高），Studio 把安全高度降到现有的空间，离行程顶至少留 0.3 mm，并在程序的结果里说明。降到离料面不到 0.5 mm 时，就不写程序，并说明原因。

装夹单的 **Z 净空**表按每次装夹核对测头和每一把刀。见Z 净空。

摆放毛坯

位置 X 和 **Y** 用孪生的工作台坐标给出毛坯中心的位置（工作台的中心是 X 100、Y 75）。**转角 °** 按从上往下看的逆时针方向转动毛坯。

- **装上新料**：在那里放一块新毛坯。原来的毛坯和它上面已切出的部分都没有了。
- **移到这里**：还是同一块毛坯，移动、转动，或者换一种夹法。已经切出的部分保留。
- **卸下**：把毛坯从工作台上拿走。
- **还原**：让各字段回到工作台上现有的样子。

主轴在转、机床在动或程序在运行时，不能装夹，要等它们停下来。

在三维视图里摆放

点 **摆放工件**（技师级别及以上），或选择 **视图 > 三维视图 > 摆放工件**。然后：

- 拖动毛坯来移动它，拖动橙色手柄来转动它，拖动蓝色手柄来改尺寸（按住 Shift 微调，按住 Ctrl 粗调）；
- 或者用键盘：方向键每次移动 1 mm（Shift 是 0.1 mm，Ctrl 是 10 mm），Q 和 E 每次转 15°（Shift 是 1°），Enter 装夹，Esc 放回原处。

离开 **准备** 工作区，摆放就结束。

毛坯翻面

翻面 让毛坯在夹具里翻过来，绕它的长度方向转 180°：上下对调，前后也对调。已经切出的部分跟着一起翻。工件坐标保持原样，所以要重新设工件零点（见用测头找工件零点）。

零件要从两面加工时，CAM 在各次装夹之间规划翻面（见多次装夹的作业）。

刀具和刀具表

停靠区的 **刀具表** 列出机床的刀具：T 号、刀具，以及程序读取的规格。主轴上的刀标着 **在主轴上。**刀具表来自示例零件、CAM 作业或 AI 客户端；在窗口里不能改它。

有刀具表时，程序自己换刀（`M6`）：刀装上去，机床在对刀仪上测量这把刀，然后继续。没有刀具表时，`M6` 会等你把刀装上（在 **对刀** 步骤里点 **换刀**），再按 **循环启动。**

换主轴上的刀

1. 转到 **对刀** 步骤。
2. 在 **刀具** 列表里选刀，例如 **平底铣刀 Ø6 两刃（伸出 15.5）**。
3. 点 **换刀。**

主轴必须停着，机床必须空闲。新刀比原来的刀长或短时，要测量它（**对刀补偿**），或者重新设工件 Z 零点。列表里是标准刀具，公制的在前（Studio 显示英寸时，英制的在前）。刀具怎么描述，例如 `flat 6 n2 L15.5`：见刀具和刀具表。

用对刀仪量刀

对刀仪在工作台的左后角。机床必须已回零并空闲，主轴必须停着。

1. 把你设工件 Z 零点用的那把刀装在主轴上，点 **测参考刀。**
2. 每次换刀之后，点 **对刀补偿。**结果：Studio 测量新刀，并按它和参考刀的差值设它的刀长补偿（`G43.1`）。

同样的键也在 **机器 > 对刀仪与测头** 里。参考刀的测量结果保留到 Studio 关闭：重启之后要重新测一次。

用来对比的零件

有了零件的设计，孪生就能显示切出来的零件哪里和它不同（工程师级别及以上）。

1. 先设工件零点：零件载入时放在程序的零点上。

2. 在 **装夹** 步骤里，**零件对比** 下面，点 **零件 STL（对比）…**，选零件的 STL 文件（单位 mm，原点在程序零点）。结果：工件按与零件的偏差着色（**视图 > 着色 > 工件：与零件的偏差**）：±20 µm 以内算合格，其他颜色显示余量和过切。

取消对比 去掉这个零件。示例零件会载入自己的零件。

保存和打开装夹

- **保存装夹…** 把卡上的装夹（形状、尺寸、材料、夹具和位置）写成一个 JSON 文件。
- **打开装夹…** 把一个装夹读回卡上；点 **装上新料** 来装夹。它也能把 STL 文件打开成毛坯的形状。
- **导出工件…** 把工件按已切削的状态存成 STL（程序坐标、毛坯坐标或机床坐标），或存成 16 位高度图（PNG）。文件菜单里也有：**文件 > 导出工件…**，在哪个工作区都能用。

每台机器把它的装夹和切过的工件存在它的世界里。机床每次有变化后静止满两秒，Studio 就存一次，关闭窗口时再存一次；所以程序崩溃或断电以后，机床回到它最后一次静止时的样子。

多次装夹的作业

要从两面加工、或者要换一种夹法的零件，是一个分几次装夹的作业。在 **CAM** 工作区里，**装夹…** 给作业加一次装夹。作业树里就多了一层：每次装夹一个节点，下面是它的工序和它的程序；两次装夹之间是**翻面**这一步。只有一次装夹的作业没有这一层。

- 每次装夹有自己的工序和自己的程序。程序按作业名加装夹的序号命名，比如 `plate-1.nc`、`plate-2.nc`。
- 每次装夹都从前面几次装夹留下的零件开始。Studio 在孪生里跑完装夹 1，按**翻面**这一步说的把零件翻过来，再按这块料写下一次装夹的程序。三维视图里，之前切出来、现在翻到下面的部分画成虚线。
- 作业树按零件翻过几次，把一次装夹叫作正面或背面。
- 工序面板下面的**这个装夹**列出这次装夹的夹具、零点、零件怎么翻的，以及它的程序。
- **检查结果**里每次装夹一行，翻面一行。**合计**把机床上的时间加起来，给出最差的颤振裕度和最大的过切，还把成品（最后一次装夹跑完以后）和设计件对比。

一个作业最多 8 次装夹。排料的作业暂时不能分几次装夹。多次装夹在创客版及以上（见档位）。

装夹单上每次装夹一节，并写明两次装夹之间要做什么。见装夹单。

两次装夹之间把零件翻过来

在作业树里选中**翻面**。它的面板说零件怎么到下一次装夹；这几项属于这一步后面的那次装夹。

设置	选项
方式	**绕 U 转 180°**：左右两头不动，前后对调，上下也对调。**绕 V 转 180°**：前后两边不动，左右对调，上下也对调。**不翻（只换夹具）**：零件照原样放着，只换夹具。
零点	**前左上角**或**料面中心**：在哪里重新找工件零点（见沿用工件零点）
夹具	从这次装夹起怎么夹：还用原来的夹具，或者换一种

U 沿机床的 X 轴，V 沿 Y 轴，都穿过零件在上一次装夹里放着的位置。

操作者要做的列出在机床前的步骤：等上一次装夹的程序跑完（主轴停、冷却关），松开，把零件翻过来，夹紧，重新设工件零点，装入下一个程序。**在孪生里**说明孪生怎么处理这块料。

> **注意**
>
> 真零件要照这一步说的翻。下一个程序在孪生里只按这样翻过来的零件检查过；绕另一根轴翻、或者根本没翻，切到的地方都没有检查过。

沿用工件零点

零件翻过来或者换了夹具，它的角就不在原来的地方了：工件零点（G54）要重新找。作业树在这次装夹旁边写明这一点，这次装夹的程序从你选的零点开始：

- **前左上角**：零件现在放着时，顶面的前左角。翻面或换夹具以后，不另选就是它。
- **料面中心**：零件顶面的正中，比如圆形的零件。

用测头找零点（见用测头找工件零点）。装夹单给出每次装夹的测头步骤。

零件既不翻、也不换夹法时，沿用上一次装夹的零点，不再重新找。只换几把刀的第二次装夹就是这样；装夹单上它那一节写明工件零点沿用上一次装夹的，不再重测。

预览：这一页的安全提示还没有经过母语审校。

用测头找工件零点

测头装在主轴上，Studio 能找一条边、一个角、孔或凸台的中心，或者顶面，再按找到的位置设工件零点。循环开始之前，测头卡会对照装夹检查它；孪生按机床的做法把它跑一遍。

准备 工作区的 **零点** 步骤里有测头卡，在点动键的下面。1.0 里测头卡只在孪生上用；在真机上，用机床自己的控制器探测。

开始之前

需要：

- 毛坯已经装在工作台上（**装夹** 步骤）；
- 测头装在主轴上（见刀具表里的测头）；
- 主轴停着。

机床不必先回零：循环发现机床在报警或没回零，会先复位、回零，测头卡会写明这一点。

> **注意**
>
> 在 MC-1 上，用平口钳装夹时，30 mm 的测头在毛坯上方没有空间（见毛坯上方的空间）。探测时，用双面胶或压板固定毛坯。

刀具表里的测头

1. 在 **准备** 里进入 **对刀** 步骤。
2. 在 **对刀** 里选 **3D 测头 Ø2 球（伸出 30，寻边用）**。
3. 点 **换刀。**

结果：测头卡的第一行 **主轴上** 写出这个测头。主轴上装的是别的刀时，测头卡会提示先装上测头。

在刀具描述里，测头写成 `probe 2 L30`：2 mm 的球，从刀柄伸出 30 mm。

毛坯上方的空间：Z 预算

Z 在行程的最高处时，测头的尖要高过装夹的最高点：毛坯顶面、平口钳的钳口或压板。装夹或刀具一有变化，Studio 就为主轴上的刀和刀具表里的每把刀重新算一遍。有一把刀越不过装夹时，**装夹** 步骤会写出它差多少，并建议换更短的刀，或把装夹放低（用压板代替平口钳）。

测头卡对循环也查同样的事：它让测头在毛坯和夹具上方的安全高度横移，并在检查里显示这个高度。

测头卡

从上到下：

- **主轴上**：装的是哪个测头，标定过没有（**已标定** 加时间，或 **没标定**）；
- **测头循环**：六个循环，以及选中的那个循环的各项；
- 各项检查；
- **在三维里看路径** 和 **开始探测…。**

循环做完以后，测头卡改为显示它的结果（见按找到的设工件零点）。测头卡下面是 **量件** 卡，再下面是 **报告** 卡（见检验）。

数字框按显示单位填，用英制时也可以填分数。读不出来的框变红，并写出原因（见输入数字）。

检查

你设置循环的时候，测头卡就在检查它，这时什么都还没动。每项检查是绿色（✓）、琥珀色（!）或红色（×）。只要有一项是红色，**开始探测…** 就是灰的。

检查	绿色	琥珀色或红色
测头	**主轴上是测头，测头没在触发**	主轴上没有测头；测头的输入已经是触发状态
机床	**主轴停着，已回零，机床空闲**	琥珀色：在报警或没回零，循环先回零。红色：主轴在转，或机床不是空闲
软限位	**每一步都在软限位内**	有一步出了软限位
安全高度	**安全高度 Z …**，以及高过毛坯和夹具多少	毛坯上方没有横移的空间
深度	**深度 …（顶面以下）**，毛坯夹在平口钳里时，还有它高出钳口多少	球会深到钳口以下，或离毛坯底面太近

孪生跟着真机时，会多一行红字，说要到真机上用机床自己的控制器探测。

先在孪生里试一遍

这些检查来自一次空运行：孪生把循环推算一遍，什么都不动。**在三维里看路径** 把循环的各个移动画在三维视图里，代替程序的路径，直到你离开这一步。

每一次触碰都是同样的过程：

1. 保护移动到起点（`G38.3`，F600）；

2. 快探（`G38.2`，F300）；
3. 退开；
4. 慢探（F20）：结果就是这一次碰到的位置。

测头循环

选一个循环，填好它的各项，点 **开始探测…**。每个循环从离预计的面 3 mm 的地方开始，再往前搜索 5 mm。**深度** 是球心在毛坯顶面以下多深处接触（不改就是 3 mm）。**起点**：**按毛坯** 以装夹里的毛坯为准；**指定 X Y** 从你给出的一个点开始，坐标是工件坐标。

寻边

找毛坯的一个面。**哪个面**：**左面（X−）**、**右面（X+）**、**前面（Y−）** 或 **后面（Y+）**。**点数**：**1 点**，或 **2 点（还给转角）**，后者还能给出毛坯转了多少。

结果：**面 X** 或 **面 Y**，两点时还有 **转角**。

找角

用两个面找一个角。**哪个角**：**前左**、**前右**、**后左** 或 **后右**；**外角** 是毛坯的角，**内角** 是型腔的角（用 **指定 X Y** 给出它的位置）。**离角** 是碰每个面的位置离角有多远（5 mm）。**3 点（还给转角）** 在 X 面碰两次，还给出转角。

结果：**角点 X**、**角点 Y**，三点时还有 **转角**。

孔和凸台

孔中心 从孔的中间向外碰孔壁；**凸台中心** 从外面碰凸台，要从它的顶面上方越过去。**点数**：**4 点** 或 **8 点**。**直径** 是名义尺寸；留空时取圆形毛坯的直径。方块毛坯要填直径和 **指定 X Y**。

结果：**中心 X**、**中心 Y** 和 **直径**。

顶面

Z 表面 在毛坯的中间，或在 **指定 X Y** 处，碰顶面一次。

结果：**面 Z**。

按找到的设工件零点

结果卡用机床坐标和工件坐标显示每个值、每一次触碰，以及测头的标定是怎样用上的。毛坯转了的话，卡上会写转了多少；你可以在 **装夹** 步骤里把装夹的角度设成它。

要把结果设成工件零点：

1. 点 **设零点…**
2. 勾选要设的轴。循环没测的轴勾不了：找角不测 Z。结果：对话框显示将要发送的那一行，例如 `G10 L2 P1 X-129.982 Y95.011`。
3. 点 **设为 G54 零点**（或者当时在用的工件坐标）。

零点就设在找到的位置上：面、角、中心或表面。**复制程序** 复制循环发出的那些行。**再测一次** 回到各项设置。

> **说明**
>
> 孪生里的数只说明算法和流程，不是这台机床的测量精度。

用环规标定测头

测头在碰到之后要再走一点才触发（预行程），各个方向还不一样。在环规上标定就是把它量出来，以后的循环按它修正。

1. 把环规放在工作台上，选 **环规标定**。
2. 按环规的证书填 **环规直径**。
3. **位置**：环规按毛坯装夹时选 **按毛坯（管子的内孔）**，或者用 **指定** 给出它的中心。
4. 点 **开始标定…** 结果：**有效球半径**、**XY 预行程**、**三瓣** 和 **环规中心**。
5. 点 **存给这台机床**，或 **不存**。

标定属于这台机床和这个球径，跟着机床的世界存档走。恢复出厂会把它忘掉。每个结果都写明用了哪一次标定。没标定以前，测头卡显示 **没标定**：孔会量大两个预行程，凸台会量小两个。

一台机床只存一次标定。换上别的球径的测头（比如量小孔用的 Ø0.5 测针），测头卡显示 **没标定**，并写明机床那次标定是用多大的球、在多大的环规里做的：把新球在环规里标一次。新的标定替换掉原来的——换回原来的球时要再标一次。

循环停下来时

循环做不完时，结果卡用红字写出原因，控制台也记一行，**再测一次** 带你回到各项设置。常见的原因：

- 一次触碰在搜索范围内什么都没碰到：面不在循环预计的地方。控制器报 ALARM:5；
- 测头在去碰之前的路上碰到了东西：毛坯不该在的地方有东西；
- 各次触碰连不成圆（孔、凸台、环规）。

报警之后，照机床条上的步骤做：**复位**，然后 **解锁**（见报警和怎么办）。循环停下时，测头可能在任何位置：点动之前先看三维视图。

> **注意**
>
> 测头在进给保持时碰到了东西，按 **复位**，再重新探测。按 **循环启动** 会越过触点继续走。

预览：这一页的安全提示还没有经过母语审校。

出程序（CAM）

Studio 自己算刀路。你用特征描述零件：型腔、轮廓、孔、刻字。Studio 用孪生的物理模型定切削参数，写出 G 代码，切之前先在孪生里跑一遍。

CAM 做什么

- 形状是数据。特征的形状可以是矩形、圆、多边形、SVG 或 DXF 文件里的路径、文字，三维加工用网格（STL、3MF、OBJ）。
- 刀路是算出来的。Studio 偏置形状、清掉区域、把各段移动连起来。
- 切削参数由孪生定。转速、进给和切深先按材料取，再按孪生的模型调：主轴功率、切削力、颤振、刀具让刀。这些都是估计。
- 程序由孪生检查。切之前，Studio 可以先在孪生里跑一遍程序，跑出来超了的地方就改。

全部在你的电脑上算，出程序不需要联网。

最快的办法是跟 AI 助手描述零件，比如：

```
在 6061-T6 上用 6 mm 平底铣刀出一个 40 × 30 mm、深 3 mm 的型腔程序。
```

助手会用 `program_generate` 工具。它会说出了什么、要多长时间、孪生查到了什么。

作业：特征、刀具和材料

作业是 Studio 为这个零件存下来的东西：

- 名字，也就是程序的名字，比如 `bracket.nc`；
- 材料，比如 6061-T6（别的标准的牌号，比如 EN AW-6061，也认得）；
- 主轴上的刀和刀具表；
- 各个特征，按切的顺序排。

Studio 写的每个程序都存成一个版本，连同出它的作业一起放在程序目录里。重新出程序不会丢东西，旧版本都留着。

AI 助手也能打开一个作业来改，您在 CAM 工作区里做的也行，例如：

```
给我看 bracket.nc 的作业，然后把它的型腔改成 3 mm 深，去掉第二道钻孔。
```

它用 `job_get` 工具读出作业，每道工序一行说明；用 `job_edit` 改值、删掉、加上或挪动工序。改好的作业重新出成这个程序的下一个版本，照样先在孪生里检查。改不了的时候什么都不动，助手会说明原因。孪生自己试过几版（消除颤振的闭环）以后，读到的是最后一版的作业，带着那几版试出来的改动。

工序

工序	切什么	刀具
铣平面	毛坯顶面铣平	立铣刀
型腔	一块区域切到一个深度：矩形、圆、多边形或形状	立铣刀
轮廓	沿外形线：外侧、内侧或线上	立铣刀
开槽	沿一条线开一定宽度的槽	立铣刀
钻孔	在各点钻孔，深了就啄钻	钻头
镗孔	比刀大的孔，走螺旋	立铣刀
铰孔	把钻好的孔铰到尺寸	铰刀
螺纹	内螺纹	螺纹铣刀
环槽	孔壁上的一道环槽：卡簧槽、O 形圈槽、消光槽	T 形槽铣刀
倒角	外形线的边	V 刀或倒角刀
刻字	文字或线条	V 刀或小立铣刀
V 形雕刻	文字或形状，深度随宽度变	V 刀
珍珠纹	面上一个压一个的圆斑，钟表上的装饰	和圆斑一样宽的立铣刀
三维粗加工、三维精加工	网格	立铣刀或球头铣刀

CAM 工作区的面板有铣平面、型腔、轮廓和倒角（外形用导入的形状，或者在面板里给的矩形、圆、多边形）、开槽、钻孔、镗孔、螺纹铣和刻字的表单。作业里有别的工序（AI 客户端做的、示例的）照样能打开：这些工序照作业的设定留着，和其余工序一起生成、检查；选中时面板按 program_generate 的名称列出它们的设置（长度单位 mm），要改它们，请 AI 客户端改，或者改作业文件。

铣平面、型腔和轮廓

型腔分层清。策略决定每层怎么走：

- 环切（默认）：从中间一圈圈往外走到墙；
- 往复：来回走；
- 自适应：刀的吃刀量保持均匀，适合硬一些的材料里的深型腔。

型腔里的岛屿会留下来。残料加工只清前面更大的刀留下的料。

轮廓沿外形线一层一层地切，顺铣。轮廓切下来的零件可以用桥连着。

开槽

槽先按刀宽分层切，再扩到要的宽度。比刀宽的槽可以用摆线切：沿着槽线走小圈，刀的负载小一些。

孔：钻孔、镗孔、铰孔和螺纹

- 钻孔用控制器的钻孔循环。孔深超过钻头直径的三倍就啄钻。
- 镗孔用立铣刀走螺旋，切出比刀大的孔，最大到刀径的两倍。
- 铰孔用和孔的配合同级的铰刀，把钻好的孔铰到尺寸。要先钻孔。
- 螺纹铣削用螺纹铣刀从下往上铣内螺纹。公制（M8）、统一制（1/4-20 UNC）、管螺纹（G1/4）和锥管螺纹（1/4 NPT、Rc1/4）都按名字认。

倒角、刻字和 V 形雕刻

倒角用 V 刀沿外形线的边走，切到你给的宽度。刻字沿线条或文字的笔画走，深度不变；文字从 X 起、以 X 居中或到 X 止（面板里的"对齐"）。V 形雕刻用 V 刀切形状：深度随形状的宽度变，所以角是尖的。

三维粗加工和精加工

网格（STL、3MF 或 OBJ）先分层粗加工，再精加工。精加工的策略有平行、等高、陡坡与缓坡、清根。

刀具和刀具表

刀具写成：种类、直径，再加需要的参数，比如：

写法	刀具
`flat 6 n3 L20`	平底铣刀，6 mm，三刃，伸出 20 mm
`ball 3 L12`	球头铣刀，3 mm
`drill 3.4 118 L20`	钻头，3.4 mm，顶角 118°
`v 6 90 L18`	V 刀，6 mm，90°
`reamer 6 H7 n6 L32`	机用铰刀，6 mm，铰 H7 的孔
`thread 5 60`	螺纹铣刀，5 mm，60° 牙型
`flat 2 n2 s4 l6 L10`	平底铣刀，2 mm，柄 4 mm，柄从刀尖往上 6 mm 起
`thread 2 60 d1.4 n1 s4 l6 L9`	螺纹铣刀，刀头 2 mm，颈 1.4 mm 到 6 mm 高，上面是 4 mm 的柄

每把刀都写上伸出（`L`）。这样 Studio 能查刀柄会不会碰到毛坯，装夹单上的 Z 净空 也能判这把刀。

阶梯柄的小立铣刀，和带下切的刀（螺纹铣刀、T 形槽刀、燕尾刀、棒棒糖刀），要写柄径（`s`）和柄从刀尖往上多高开始（`l`）。柄会蹭到切出来的壁的地方，Studio 不切深过 `l`。孪生会查刀柄和测头本体碰不碰工件和夹具。

刀具表按刀号列出各把刀。特征写明用哪把刀，Studio 会加上换刀（`T2 M6`）。

孪生定的切削参数

Studio 先按材料取这把刀的切削参数，再用孪生的模型调：

- 主轴功率：切削最多用掉大约 80 %；
- 颤振：切深和转速留在孪生的稳定性模型认为稳定的范围里；
- 夹具：切削力远低于夹具夹得住的力；
- 刀具让刀：精加工的刀量放轻。

开了校验时，Studio 会在孪生里跑这个程序。跑出颤振或负载太大，Studio 就改程序，比如把层切浅一些，再试一遍。试过的每个版本都留着。回答里会说改了什么、为什么改。

这些都是物理模型的估计。在真机上切第一件时，要看着、听着。

配合和公差

尺寸可以连配合一起写，比如孔写 `"d": "Ø12 H7"`，轴写 `"fit": "g6"`。Studio 把尺寸切到配合公差带的中间，墙另走一刀精加工。

常用配合可以直接写名字：`sliding`（H7/g6）、`locating`（H7/h6）、`transition`（H7/k6）、`press`（H7/p6）。

又小又紧的孔，铰比铣好。Studio 会提醒；你可以先钻，再用同级的铰刀铰。

孪生会估计刀具的跳动，有配合的尺寸 Studio 会把它算进去。

多次装夹

两面都要加工的零件需要两次装夹。每次装夹有自己的特征，并写明零件怎么过来：绕 X 或 Y 翻面、换一种夹法、重新找工件零点。Studio 每次装夹出一个程序，并按前几次装夹留下的料，在孪生里逐个检查。怎么加装夹、怎么翻面，见多次装夹的作业。

在板料上排料

几个零件可以从一块板料上切出来。Studio 把它们排紧、留出边距，并加上桥，让零件一直连在板上直到最后。

英制尺寸和分数

用英制单位时，尺寸可以写成分数：`1/4`、`1 1/2`、`3/16"`。刀具库里有英制刀，比如 1/4" 平底铣刀。见语言和单位。

在孪生里检查程序

切之前，先在孪生里跑程序。孪生切它自己的工件，报出碰撞、过切、颤振、主轴负载和时间。见在孪生里跑。

> **注意**
>
> 孪生按你给它的装夹查程序。把孪生装得和真机一样：同样的毛坯、夹具、刀具和工件零点。

装夹单

装夹单告诉操作的人这个作业怎么装：毛坯、夹具、刀具、工件零点和 Z 净空。见装夹单。

程序和编辑器

程序 工作区（Alt+3）里有你的程序和示例零件、一个边写边检查程序的编辑器，以及一个显示程序对工件做了什么的预览。每次你保存程序，或把它载入机床，Studio 都会留下一个版本，所以你能看改了什么，也能退回去。

打开程序

工作区分三部分：

- 左边是 **程序**（每个程序和它的各个版本）和 **示例零件**；
- 中间是编辑器；
- 右边是三维预览、**载入并检查** 和程序检查发现的问题。

打开程序：

- **文件 > 打开程序...**（Ctrl+O）把一个 G 代码文件载入机床；编辑器里没有未保存的改动时，也载入编辑器。Studio 只读你的文件，从不写它。
- 把一个 G 代码文件（.nc、.gcode、.ngc、.tap、.txt、.cnc）拖到窗口上任何地方，效果一样。程序运行时什么都不打开，在控制台里说明。
- **文件 > 最近的程序** 列出最近从文件打开的八个程序，最新的在上。选到一个已经挪走或删掉的，它就从列表里拿掉；**清空列表** 把它清空。
- **文件 > 打印程序...**（Ctrl+P）打印机床里的程序：行带行号，白纸黑字，每页上面写程序名、页码和日期。
- 在 **程序** 下双击一个程序或一个版本：它的文本进入编辑器。按 **载入并检查** 之前，它不会进入机床。

编辑器和机床里可以是不同的程序。机床里有一个你没在编辑器里打开的程序时，状态行会说明：Ctrl+K 把编辑器里的程序载入机床，Ctrl+Shift+R 把机床里的程序取到编辑器里（你的改动先存成一个版本）。

示例程序

Studio 1.0 自带四个示例零件：步进电机安装板（NEMA 17）、仪器前面板、微流控液滴芯片和等格栅轻量化面板，更多示例随更新发布。在 **示例零件** 下双击一个，或者从 **文件 > 示例零件...** 里选。

载入示例会为它把整台机床设好：毛坯和夹具、刀具表和主轴上的刀、回零、工件零点、程序，示例带有对比用零件的话，还有这个零件。

> **说明**
>
> 示例会换掉你原来的毛坯、刀具表和 G54 工件零点。同一个列表里的 **示例程序** 只载入程序，毛坯保持原样。

编辑器

编辑器是 G 代码编辑器：

- 程序里的字按种类着色：G 代码和 M 代码、轴、圆弧字、进给、转速、刀具、注释、表达式；
- 边栏显示行号和标记：检查发现的问题（红色错误、琥珀色警告、蓝色提示）、断点，以及机床正在运行的那一行；
- 运行之后，第二栏显示每一行的用时，单位是秒：琥珀色表示没达到进给，红色表示那里出过事（快移切进材料、主轴没转就切、碰撞、报警）。点一下，看这一行在上次运行里的各项数字。

编辑器下面的状态行显示行和列、文本有没有在机床里、版本、检查的计数，以及程序在你所在这一行执行后的状态：模态代码、进给、转速、刀具和位置。

放大 把整个工作区给编辑器；**还原** 把程序列表和预览请回来。

颜色和行号

颜色跟着窗口的外观走（浅色或深色）。控制器已经提前收下、排在正在运行的那一行后面的行是灰色的。

查找和替换

- Ctrl+F 打开查找框。Enter 找下一个，Shift+Enter 找上一个。查找不分大小写，到末尾会绕回开头。
- Ctrl+G 跳到某一行。
- Ctrl+H 在查找框下面打开替换框。Enter 替换选中的那一处并跳到下一处；Ctrl+Enter 全部替换。Ctrl+Z 撤销。

Esc 关闭这些框。

断点和暂停

断点在运行到那一行时让孪生的时间停住；机床的运动不变。按 F5 接着运行。

- Ctrl+B，或在边栏里点一下，在这一行设一个断点。
- **条件暂停**（Ctrl+Shift+B）在出现某种情况时停住孪生：刀开始切入材料、快移切到材料、险些碰撞或碰撞、颤振、报警、控制器的状态变化、切削力、主轴负载或跟随误差超过你选的限值，或者模态变化。孪生实时运行时，颤振在仿真旁边另算，所以因颤振而停、以及视图里的颤振数字，可能比那一刀晚一点到：通常晚几百毫秒，许多新的切削条件挤在一起的地方（比如绕着岛的型腔）可能晚一两秒。一把刀的颤振检查还没给出答案时，裕度写"**计算中...**"；曲线和刀尖轨迹等每一段自己的答案，所以显示的就是那一段的裕度。运行结束时的报告两种情况都一样。
- 这一行的菜单（在边栏里右键，或按 Shift+F10）里还有 **运行到第 ... 行** 和 **清除所有断点和条件**。

程序载入机床之后，断点才生效。

程序检查

编辑器在你输入时（停手片刻之后）、程序载入时和运行结束之后检查程序。检查分两部分：

- 规则检查：每一行都对照控制器和机床能接受的内容来检查。它不需要别的条件；
- 预览检查：按工件现在的样子把程序跑一遍。它要求机床已上电、已回零，没有程序在运行。边输入边检查时，它只查不超过 20 000 行的程序；更长的程序装进机床后整段检查，查出的问题在三维视图里标成红框，也列在 **消息** 页里。

程序 > 检查程序 是另一种检查：它把已载入的程序逐行送过控制器自己的检查模式，机床不动。

检查查什么

每一条发现是错误、警告或提示，显示在它那一行下面，带着规则的编号。例如：

规则	类型	发现什么
S01	警告	第一条移动之前没有起始段：程序依赖控制器当时的状态
S02	警告	结尾没有 `M2` 或 `M30`
S04	错误	第一个切削移动之前没有 `F`
S05	警告	主轴启动时没有 `S`，或 `S` 超出主轴的范围
S07	错误	控制器不支持的代码：表达式、`O` 字、子程序、`G41/G42`、`G68`
S10	警告	快移到工件顶面以下，比程序已经切过的深度还深
S11	错误	控制器会拒绝的圆弧
S12	警告	水冷主轴切削时没开冷却（`M8`）
S14	警告	刀的长度和设工件零点时用的刀不同，而且没有量刀

预览检查再加上程序会对工件做什么：快移切进材料、主轴没转就切、和夹具碰撞、超出行程的移动，以及主轴干转。

改掉一条问题

有些发现带着建议，例如 **建议在前面加：M8。**

1. 把光标放在这一行，或按 Alt+↓ 和 Alt+↑ 在各条发现之间移动。
2. 按 Ctrl+.（或在这一行的菜单里选 **采用建议**）。

结果：这个修改作为一步完成；Ctrl+Z 撤销。采用之前先读建议：它只消除这条发现，不管你的意图。

优化

优化（Ctrl+Shift+O）找一个切同一个零件、但更快的程序：它修改进给和转速、合并短行、降低抬刀。它把原程序和优化后的程序放在机床的两个隐藏副本上跑，装夹相同。它要求已经装好毛坯。**优化** 在专业版及以上。

结果更好的话，它换掉编辑器里的文本（Ctrl+Z 还你原来的），并打开对比。控制台给出前后的用时、最大切削力和颤振裕度，以及改了什么。按 **停止优化** 或 Ctrl+Shift+O 可以停下它。

版本

你做下面这些事时，Studio 会存一个程序版本：

- 按 **存版本**（Ctrl+S）；
- 把它载入机床（**载入并检查**，或 **文件 > 打开程序…**）；
- 优化它，或者用生成的程序、机床里的程序换掉你的改动（你的改动先存起来）。

和上一个版本相同的版本不再重复保存。版本放在 `Documents\Twinwright Studio\programs\<program>\` 里，一个版本一个文件，不会被清理。**程序** 在每个程序下面列出它们，写着每个版本的时间和为什么保存。

比较两个版本

对比（Ctrl+D）在编辑器下面显示与编辑器文本不同的最新一个已存版本：改过的行是琥珀色，加的行是绿色，删的行是红色，只改了 `F` 或 `S` 的行是蓝色。再按 Ctrl+D 看更早的版本，Alt+↓ 和 Alt+↑ 在各处不同之间移动，Esc 关闭。

标题行给出不同之处的数量；两个版本在这次会话里都完整运行过的话，还比较它们的运行时间。

要比较两个已存的版本，先把较新的那个在编辑器里打开，再按 Ctrl+D。

退回到一个版本

1. 在 **程序** 下双击这个版本。结果：它的文本进入编辑器。你没保存的改动先存成一个版本；Ctrl+Z 撤销这次替换。
2. 按 **载入并检查**（Ctrl+K）把它载入机床。

保存和导出

- **存版本**（Ctrl+S）在 Studio 的程序文件夹里存一个版本。
- Ctrl+Shift+S 把编辑器里的程序存成你选的位置上的一个文件。
- 一个版本就是一个普通的 `.nc` 文件：你可以从程序文件夹里把它复制出来。

文件 > 导出曲线（CSV）... 保存这次运行的曲线（见分析和诊断）。装夹单和检验报告在 **准备** 的 **报告** 卡里（见检验）。

预览：这一页的安全提示还没有经过母语审校。

运行程序和断点续跑

运行 工作区运行已载入的程序，并显示机床在切削时做什么。运行停下来时，Studio 告诉你下一步，也能从某一行重新开始运行程序。

装入和启动

把程序载入机床，用下面任何一种办法：

- **文件 > 打开程序...**（Ctrl+O）：选一个 G 代码文件。Studio 只读它，从不写你的文件。
- **文件 > 示例零件...**：选一个示例。Studio 还会为它设好毛坯、刀具和工件零点。
- 在 **程序** 工作区里，**载入并检查**（Ctrl+K）载入编辑器里的程序。

这时机床条显示程序的名字、行数和大约要用多长时间。

要开始运行，按机床条上的 **循环启动**（F5）。程序已载入、机床空闲且已回零时，**循环启动** 是填色的。程序检查发现了问题的话，程序开始时控制台会说明。

在真机上运行程序之前，先检查它：**程序 > 检查程序** 把它送过控制器的检查模式。控制器读每一行，机床不动。

看着运行

程序运行时：

- 三维视图显示刀具、刀尖的轨迹，以及切到现在的工件。**刀具特写** 显示切削处的刀具；
- 停靠区的 **程序** 页显示程序清单：正在运行的行有标记，已经走完的行颜色浅一些，每行有一根条表示它的用时。**跟随当前行** 让当前行始终在视野里；
- 清单旁边的 **程序结构** 按程序的注释把程序分成各节和各层，并给出它们的用时；
- 右栏显示各项读数。

程序结束时，**消息** 页说明运行了多少行，以及这次运行在仿真里用了多长时间。

进度和剩余时间

机床条显示正在运行的是全部行里的第几行、已完成行数的百分比，以及 **已用** 和 **剩**：

- **已用** 是到现在为止的机时，按仿真的时间算；
- **剩** 是对剩下部分的估计，按程序的进给和快移算。

报警之后，机床条显示程序停在哪一行。

读数

运行 的右栏，从上到下：

- **位置**：每根轴的工件坐标和机床坐标，轴已回零时有标记。下面是模态行：有效的 G 代码、刀具、进给、主轴转速和 M 代码。报警时，主轴和冷却显示机床实际的状态，程序要求的值在提示里；
- **主轴与进给**：主轴转速（弧线是转速，标记是要求的转速）、主轴负载（到 80 % 是绿色，超过 100 % 是红色）和进给；
- **切削**（工程师级别及以上）：孪生对切削力、功率、颤振裕度和每齿进给的估计；
- **倍率**、**面板**、**手动** 和 **刀具与零点**（操作员级别及以上）。

点一节的标题可以把它折起来。车间模式下，**位置** 还显示 **待走**：当前这一行里每根轴还剩多少距离。

进给保持、停止和倍率

- **进给保持**（F6 或 Space）让各轴减速停下；主轴继续转。**循环启动** 接着走。
- **停止**（F7）先保持，机床停稳以后复位控制器。再按一次，立即复位。
- **复位**（F12）立即停下程序。

倍率在程序运行时改变程序的进给和主轴转速，范围从 10 % 到 200 %：

倍率	按键	屏幕上
进给	Ctrl+↑、Ctrl+↓：±10 %	**倍率**：拖旋钮；按住 Ctrl 拖，每步 1 %
主轴	Ctrl+→、Ctrl+←：±10 %	**倍率**：拖旋钮
快移	**机器 > 倍率**：25 %、50 %、100 %	**倍率**：100、50、25

Ctrl+0，或双击旋钮，把进给和主轴恢复到 100 %。

单段

机床条上的 **单段** 让程序在每一行之后停下。按 **循环启动**，或 **程序 > 下一段**，运行下一行。你可以在开始之前或程序运行时打开它；再点一次，就不停顿地接着运行。

点动和手轮

在 **运行** 的 **手动** 下点动（它一开始是折起来的），或用按键（见操作机床的键）。

- **步长**：每按一次 0.01、0.1、1、10 或 50 mm，或选 **按住连续**：按着键，轴就一直动。
- **速度**：100、300、1500 或 3000 mm/min。
- 按键区：X 和 Y 摆成十字，Z 在旁边，**停止点动**（Esc）在中间。
- **手轮**：选 **X**、**Y** 或 **Z** 和增量（**×1**、**×10**、**×100**：每格 0.001、0.01 或 0.1 mm），然后拖表盘，或在表盘上转鼠标滚轮。

只有机床已上电、空闲、不在报警，而且没有程序在运行时才能点动。按着点动时，松开按键或窗口失去焦点，点动就停。

工具 > 设置... > 面板与外设 > 手轮 设定手轮的行为：**速度** 在手轮一停时停轴；**路径** 每一格都走到。

报警和怎么办

报警时，机床条上的状态变红，并显示：

- 报警和它的意思，例如 **ALARM:17 电机驱动器报告故障**；
- **下一步：** 现在该做什么，以及做这件事的按键；
- **全部步骤**，不止一步时才有。

点这个键，或用 **机器 > 做下一步**，就做下一步。每做完一步，Studio 都按机床的状态把各步重新算一遍。

报警的其他标志：位置不再可信时，**位置** 下面有一行说明；程序清单里程序停下的那一行是红色的；三维视图里驱动器处有一个标签。

各个报警、它们的原因和怎样才能清除：报警。

恢复步骤

全部步骤 打开恢复步骤：先是原因，然后按顺序列出各步，每一步有它的按键和为什么需要它。现在该做的那一步有标记。点一下只做下一步。

例如，驱动器报警（ALARM:17）之后：

1. **按急停，给驱动器断电**（F1）：驱动器的报警要它断电才会清除。
2. **松开急停**（F1）。
3. **上电**（F2）：驱动器重新有电。
4. **复位**（F12）：驱动器恢复以后，控制器还记着 ALARM:17。
5. **回零**（F3）：断过电，位置不可信。

故障是在 **诊断** 里注入的话，原因里会这样写，还带着时间。

报警让程序停下的话，各步的最后一项是重新开始的那一行，例如它停下时所在那一层的开头。

从某一行重新开始

Studio 能从任意一行开始运行程序。它写一段起始段，把机床设成程序到这一行时应有的状态，然后运行其余的部分。

1. 停下程序。机床要已上电，且没有程序在运行。
2. 在停靠区的 **程序** 页里，右键点这一行，选 **从第 ... 行开始**，再点 **确定**。在编辑器里，右键点这一行的边栏，或按 Shift+F10。结果：Studio 载入“*名称*（从第 *n* 行起）”。它不会开始运行。
3. 按 **循环启动**。

起始段：

1. 把 Z 抬到最高处；
2. 设好程序在这一行时的单位、平面、距离模式和工件坐标；
3. 程序在这一行之前换过刀的话，换刀；
4. 打开冷却，让主轴按程序的转速转起来，并等主轴到速；
5. 在安全高度移到起点上方，下到起点上方 2 mm，再按进给下刀；
6. 从这一行起接着运行程序。

然后，预览检查按工件现在的样子检查其余的部分。

> **注意**
>
> 起点之前的有些行，Studio 无法跟踪：参数、表达式、子程序、`G10`、`G28`、`G30` 和 `G92`。载入时的说明会列出它们。按 **循环启动** 之前，先检查机床的状态。

先在副本上试起始段

报警让程序停下之后，各步的最后是 **预览起始段...**。它打开起始段和关于它的说明。

1. 点 **在副本上试**。结果：起始段在机床的一个隐藏副本上，从工件现在的样子开始运行；Studio 报告它的用时、有没有碰撞、最大的颤振裕度和去除了多少材料。
2. 看起来对的话，照恢复步骤做，然后从这一行开始。

按仿真的速度运行

孪生按实际时间运行。**仿真 > 倍速** 让它按实际时间的 ×0.5 到 ×50 运行。机床条上的时钟显示仿真的时间、你选的速度和电脑实际达到的速度。

仿真 > 暂停仿真，或机床条上的沙漏，让孪生的时间停住：停着的时候什么都不变。**循环启动** 接着走。编辑器里的断点和条件暂停也同样让孪生的时间停住（见程序和编辑器）。

暂停仿真不是进给保持：机床的状态是冻结的，不是停止的。

分析和诊断

分析 回头看一次运行：时间花在哪里、孪生对切削力和颤振的估计，以及切出来的零件和设计比怎么样。**诊断** 显示机床的电源链和信号，也让你在孪生里注入故障，练习找出它们。

看一次运行

程序运行过之后，**分析**（Alt+5，工程师级别及以上）才有内容。

- 左边的 **运行记录** 列出这次会话的各次运行：程序、怎么结束的、机时和什么时候开始的。以报警或出错结束的运行是红色的。
- 中间是曲线，曲线下面有两页：**逐行剖析** 和 **稳定图**。
- 右边的 **上次运行** 总结最后一次运行：走到了哪里、最耗时的行、最大切削力和最大颤振裕度以及它们所在的行、去除的材料、没达到进给的行，以及事件。

曲线、剖析和总结都是最后一次运行的。

逐行剖析 列出花了时间的行，最长的排在前面：用时和占比、编程的进给和实际达到的进给、没达到的原因、切削力、颤振裕度、去除的材料和事件。双击一行，或在这一行上按 Enter，就到 **程序** 编辑器里的那一行。

一行没达到进给的原因：

- 各轴的最高速度；
- 转角挨得太近，两个转角之间加速不起来；
- 圆弧的弦限速；
- 控制器的规划，以及发送软件或控制器的解析跟不上；
- 进给倍率低于 100 %，或进给保持。

时间：估计的和实际的

运行之前，Studio 按程序的进给和快移估计机时：机床条显示这个时间，前面带着 **约**，停靠区的 **程序结构** 按每一节显示它。运行中和运行之后，时间是孪生的机时，里面含着加减速和控制器的规划：机床条显示已用和剩下的时间，**程序结构** 显示整次运行的时间，**逐行剖析** 显示每一行的用时。

所有的时间都是仿真里的机时，不是你的钟表上的时间。

切削力、颤振和切屑厚度

曲线显示这次运行最近 10、30 或 120 s 的读数：进给、实际速度、主轴的转速和负载、切削力、转矩和功率、切屑厚度、颤振裕度和跟随误差。点一条曲线的名字，显示或隐藏它。

颤振裕度是切深除以当前主轴转速下稳定的切深：

裕度	意思
小于 0.8	稳定
0.8–1	接近边界
大于 1	会颤振

稳定图 画出稳定切深随主轴转速的变化，针对最后一刀（还没切过时，针对沿 X 方向、1 mm 深的全宽槽）。点 **计算稳定图**。红点标出当时的那一刀，绿色虚线标出最稳定的转速。

视图 > 着色 按进给速度、切削力、切屑厚度、颤振裕度或程序行给刀尖轨迹着色，按材料与切深、与零件的偏差或程序行给工件着色。

这些是孪生按物理模型给出的估计（见孪生的模型）。

和设计零件比

要看切出来的零件哪里和设计不同：

1. 在 **准备** 的 **装夹** 步骤里点 **零件 STL（对比）…**，选零件的 STL 文件（单位 mm，原点在程序零点）。
2. 运行程序。
3. 选 **视图 > 着色 > 工件：与零件的偏差。**

结果：工件按每一点离零件有多远着色：±20 µm 以内算合格，其他颜色表示还有余量和过切。**取消对比** 去掉这个零件。示例零件会载入它们自己的零件。

比较两次运行

分析 显示最后一次运行。要比较一个程序的两次运行：

- 在 **程序** 编辑器里，**对比**（Ctrl+D）给出两个版本的运行时间，以及改过的那些行的用时，前提是两个版本在这次会话里都完整运行过（见比较两个版本）；
- AI 助手可以把两次运行并排画出来（见AI 助手）。

导出

导出曲线（CSV）…（在 **上次运行** 里，或 **文件 > 导出曲线（CSV）…**）把曲线最近约 160 s 的部分存成一个 CSV 文件：第一列是以秒为单位的时间，然后每条曲线一列，标题里写着它的单位，数值用显示单位。

准备 里的 **导出工件…**（或 **文件 > 导出工件…**）把切过的工件存成 STL 文件（用程序坐标、毛坯坐标或机床坐标），或存成高度图。

诊断工作区

诊断（Alt+6，技师级别及以上）用一张图显示机床的电源链：电从左流向右。实线传电，有电时是绿色；虚线是信号。每个部件，正常时是绿色，断电时是灰色，有警告时是琥珀色，报警或故障时是红色。

点一个部件，在右边一栏里看它的详情。图的下面是控制器的输入、故障注入和驱动器报警。

电源链

MC-1 的电源链，从市电到主轴：

- **总电源 QF1** 和给控制器的 **24 V 电源**；
- **控制器**：它的状态或报警；
- **急停 SB0** 和 **KM1**，也就是急停断开的那个接触器；
- 给驱动器的 **48 V 电源**；
- **X**、**Y**、**Z** 驱动器和它们的轴；
- **变频器**、**主轴** 和 **冷却水泵**。

控制器的输入

每个输入一盏灯：限位开关（**X− 限位**、**X+ 限位**、**Y− 限位**、**Y+ 限位**、**Z 限位**）、**对刀仪**、**急停** 和 **驱动报警**。输入没有触发时灯是灰色，触发时是琥珀色或红色。

对刀仪 这盏灯就是控制器的测头输入：测头碰到东西时它也会亮。

驱动器

每个驱动器显示 **就绪** 和它的跟随误差，或者 **未使能**、**报警**、**启动中**、**断电**。详情给出驱动器的电源，以及每根轴的跟随误差和它的最大值。

主轴驱动（变频器）

变频器 显示它的输出频率和电流，或它的故障：**OC**（过流）、**OL**（过载）或 **LU**（欠压）。详情给出主轴转速和要求的转速、负载和水泵。主轴没开冷却就转过的话，会列在警告里，并写出干转了多久。

在孪生里试故障

故障注入让你在孪生上练习找出并排除故障；被改变的是你窗口里的机床，不是真机。

- **接线**：把机床的任何一根线（限位开关、对刀仪、驱动器的 ALM 链、急停线、KM1 回路、RS485 总线、变频器故障继电器）设成 **断线**、**短路**、**间歇断** 或 **干扰脉冲**（一次 50 μs 的掉线）。**正常** 把这根线恢复。
- **驱动器报警**：选 **X**、**Y** 或 **Z**，让那个驱动器报故障。驱动器的报警要它断电才会清除：照机床条上的恢复步骤做。

注入的每个故障都列在警告里，写进控制台，也记进审计日志。驱动器报警之后，机床条上的原因里写着你注入的故障和时间。

报警：每一个和它的恢复步骤

每个报警、它的意思和原因：报警。机床条会给出你遇到的那个报警的恢复步骤（见恢复步骤）：

报警	恢复步骤
驱动器的故障（ALARM:17，有驱动器在报警）	按急停，松开急停，**上电**，**复位**，**回零**
急停已按下（ALARM:10）	松开急停，**上电**，**复位**，**回零**
ALARM:17，驱动器已恢复	**复位**，**回零**
需要回零（ALARM:11）	**回零**
硬限位（ALARM:1）	**复位**，**解锁**，点动离开限位开关，**回零**
ALARM:3、8、9	**复位**，**回零**
其他报警	**复位**，**解锁**

预览：这一页的安全提示还没有经过母语审校。

在孪生里跑程序

草稿（第 14 轮）：正文在写，各节还会补。

孪生是什么

窗口里的机床是 MC-1 的数字孪生：里面跑的是同匠自研内核，和机床上是同一份代码；电机、驱动器、主轴、开关和切削由物理模型代替。程序在孪生里怎么跑，在机床上就怎么跑，孪生还会切它自己的工件。它给出的关于物理世界的每个数都是估计。

跑之前

孪生按你给的装夹检查程序，所以要照真机的样子设好：

- 机床（**文件 > 切换机器...**）；
- 毛坯和它怎么装夹（**准备** 工作区）；
- 刀具（停靠栏的 **刀具表**）；
- 工件零点（**机器 > 工件零点**）。

文件 > 示例零件... 一次把这些都设好，并载入一个程序。

> **注意**
>
> 检查只对得上你设的装夹：真机上换了毛坯、夹具、刀或工件零点，就不是孪生检查过的那一套。

跑一个程序

1. 打开程序：**文件 > 打开程序...**
2. 先检查：**程序 > 检查程序** 用控制器的检查模式过一遍，控制器读每一行，机床不动。
3. 运行：**循环启动**（F5）。**进给保持**（F6）暂停，**循环启动** 接着走，**停止**（F7）停下。**单段** 一行一行地跑，**下一段** 跑下一行。

不另外选的话，孪生按实际时间跑：**仿真 > 倍速** 可以从一半到 200 倍，或者 **最快**——这台电脑跟得上多快就多快；**仿真 > 暂停仿真** 让孪生的时间停住（停着的时候机床的状态不变）。点机床条右边的时钟，弹出的是同一张倍速表。

×100 起、**最快** 和算到结束时孪生是 **精简** 的（时钟旁边写着）：颤振检查和受力可能晚一点补到曲线上，画面用粗一点的格子、轨迹抽稀。程序做的事——切掉的材料、碰撞、软限位、报警——照 ×1 判，结果和报告一样。**设置 > 常规 > 高倍速也全精度（会慢）** 可以关掉精简。

电脑跟不上时，时钟用琥珀色写实际有多快；把鼠标放上去，会说孪生现在最忙的是哪一块（比如细格子上的材料去除）、怎么更快。

仿真 > 倍速 > 算到结束（不播放） 把载入的程序一口气算到结束，不一帧帧放：机床条上显示算到哪儿、还要多久，3D 视图停在开始时、算完一起换上，分析工作区的"上次运行"照常填好。程序旁边的 **停** 让它停在当时那里。连着真机（影子模式）时只能 ×1：孪生跟着真机的时间走。

孪生检查什么

由程序和装夹的几何决定的：

- 控制器自己的报警和错误，和机床上的控制器给出的一样：软限位、限位开关、探测走完了还没碰到东西；
- 刀柄、夹头螺母和主轴碰到毛坯、夹具和台面，以及差一点碰到；
- 快移（`G0`）时切到料，主轴不转时切到料；
- 切穿毛坯底面、切进夹具或台面；
- 会卡刀或断刀的刀：每齿进给超过这把刀受得了的，刃根的弯曲应力超过刀受得了的；
- Z 向净空：Z 在最高处时，每把刀和测头能不能越过装夹（见 MC-1 硬件）；
- 装夹里有零件的设计时：零件上留下的料和切深了的地方（**视图 > 着色 > 工件：与零件的偏差**）。

发现的问题随时出现在停靠栏的 **消息** 页。

孪生估计什么

物理模型算出来的：

- 切削力、转矩和功率，以及主轴的负载：主轴受载时会和真的一样掉速，负载太大会让变频器跳闸；
- 刀和机床在切削中让开多少，体现在切出的形状里；
- 颤振：沿路径的裕度（**视图 > 着色 > 轨迹：颤振裕度**），以及切削稳定的主轴转速（**分析** 工作区：**计算稳定图**）；
- 刀具跳动：槽和孔因此切宽；
- 加工时间：按控制器自己的规划和驱动器的运动；
- 球头刀两刀之间留下的残脊（它的形状，不是表面粗糙度）。

轨迹还可以按进给速度、切削力或切屑厚度着色（**视图 > 着色**），停靠栏的 **曲线** 页画出这次运行随时间的变化。

> **说明**
>
> 在你的机床实测之前，这些估计用的是典型参数——公开的切削数据、锋利的硬质合金刀、机床的设计值。拿它们比较程序、在切之前找出问题，不要拿它们当零件的测量结果。

孪生没有建模的

- 表面的纹理和粗糙度；
- 积屑瘤、切屑和排屑；
- 薄壁在切削中弯曲；
- 零件自己的发热；
- 刀具磨损：切削一直按锋利的刀算。孪生给每把刀记一本磨损的估计账——刃口在材料里的时间、切过的长度、按 Taylor 公式估的寿命用掉了多少（多数材料的数是估计）——并报出来（MCP machine_status 的 tool_wear），但不改变切削；同规格换一把新刀就从头记（setup_tool 加 new）。切铝的刀多半先因为铝粘在刃口上、零件出毛刺或表面变差而换；
- 断刀之后会怎样；
- 比程序检查的格子还细的东西：开跑之前的程序检查按约 0.1 mm 的格子看工件，程序里有直径小于 2 mm 的刀时更细：刀径的二十分之一，最细 0.05 mm（毛坯大于 100 × 100 mm 时更粗，最多一百万格，有这样的刀时四百万格），比这更薄的墙上的快移、擦边不到这么多的刀，检查可能查不出。要看程序在孪生里实跑的结果：孪生自己切削用 0.02 mm 的格子。

塑料和微小刀

用 0.1–0.5 mm 的刀切 PMMA（亚克力），例如微流控的流道，孪生还没有算：

- 切削的温度：PMMA 接近玻璃化温度（约 105–115 °C）时变软，从大约 70 °C 起切屑会熔粘在刀上和流道里。每齿进给和转速按刀具厂家给的范围取；
- 吹气或冷却液，以及它对热和切屑的作用；
- 流道和型腔边上的毛刺；
- 薄壁：PMMA 里厚度不到约 0.15 mm 的壁会在切削中弯曲或断裂；孪生切出的是刀路和刀的让刀留下的形状，不算壁自己的变形；
- 切削本身的阻尼：这么小的刀弯曲频率在几十千赫，切削速度又低，刀的后刀面蹭着已切表面，会把振动阻尼掉。孪生没有算这一项，所以它对这类刀给的颤振裕度偏保守。

各个模型和它们的假设：孪生的模型。

结果和报告

- 三维视图里的工件就是孪生切出来的；着色显示切深，有零件的设计时显示与它的偏差。
- 停靠栏的 **消息** 页列出检查发现的问题；**曲线** 页是随时间变化的力、负载和转速。**文件 > 导出曲线（CSV）…** 把它们存下来。
- AI 客户端经 Studio 的 MCP 工具读到同样的结果：见参考里的 MCP 工具。

预览：这一页的安全提示还没有经过母语审校。

检验、检验报告和装夹单

零件切完以后，用测头按设计量。Studio 把量到的数和孪生的估计并排写进检验报告。切之前，装夹单告诉操作的人这个作业怎么装。两样都在专业版里。

用测头量零件

零件切完、从机床上取下来之前，就地量。**量件** 卡在 **准备** 工作区的 **零点** 步骤里，测头卡的下面。它拿零件和 CAM 作业的设计比：每个特征的尺寸、配合和公差。1.0 里量件卡和测头卡一样，只在孪生上用；孪生跟着真机时，在真机上用机床自己的控制器量。

需要：

- 切好的零件，还按切的时候那样夹着。不要动工件零点：位置都从它量。
- 测头装在主轴上，已经用环规标定过。见测头。
- 这个程序的 CAM 作业：在 **CAM** 里做的程序，或 AI 助手做的。

> **说明**
>
> 量件在专业版里。别的版本里卡片照样显示，标题旁有 **专业版** 标签，按 **开始量件…** 会说明怎么试用。以前的记录和报告在任何版本里都能打开。



量的步骤：

1. 在 **准备** 里点 **零点** 步骤。结果：卡片上写着拿什么比：程序和切它时用的工件坐标，比如 G54。
2. 选要量的特征。一开始全选。结果：每个特征写着测头在它上面量什么。量不了的特征是灰的，写明原因。
3. 在 **一般公差** 里选级别，给没有自己公差的尺寸用。要查形位就加 H、K 或 L（见公差）。结果：卡片说要碰几次、大约多久。
4. 看各项检查：主轴上是测头、标定过；工件零点就是切这个件时用的那个；主轴停着；机床已回零、空闲。
5. 先清掉孔里和型腔里的屑、冷却液。测头碰到屑，会把屑当成零件量。
6. 想看测头碰哪里，点 **在三维里看触碰点**。结果：三维视图里显示侧面的触碰和向下的触碰。
7. 点 **开始量件…**。结果：测头逐个碰每个特征，卡片上写着正在量什么。

> ⚠ **警告**
>
> 测头在零件附近按进给速度移动。量的时候，手和工具都离开机床。

量什么：

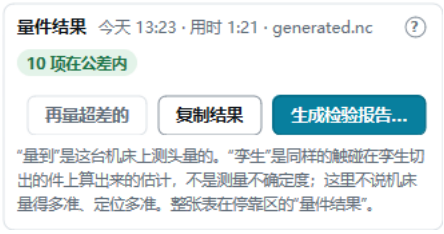
特征	检查
型腔、轮廓	墙的尺寸和位置，底的深度
孔、凸台	直径和位置
槽	宽和深
平面	深度
倒角	宽、深和角度
V 形槽（用 V 形刀的刻字、V 形雕刻）	宽、深、角度，和离它那条线的偏移
用平底刀的刻字、螺纹、三维曲面	不量

形位检查在适用的地方再加平面度、垂直度、圆度和直线度（见形位）。

测头量完，**量件结果** 卡换下量件卡。它显示：

- 什么时候量的、用了多久、哪个程序；
- 几项在公差内、几项超差、几项未判定：**孪生里判不了**（那里的格子比公差带的四分之一粗）、**没有公差**，或者**没量**；
- 每一项超差的：量到的值、它的上限或下限、超出多少；
- 第一个超差的特征：最可能的原因和下一步怎么做。

1.0 里测头量的是孪生切出来的件，所以原因只从孪生里推。孪生自己的件在同样的碰点上也超差，就是程序这么切的：到 **CAM** 里改。否则量到的值和孪生的估计只差在孪生对测头的模型（球的有效半径；典型测头还有预行程，在环规上标定就能去掉）和碰点那里孪生格子的大小。在环规上标定测头，再量一次。原因只是建议，不是诊断。



卡上的按钮：

- **再量超差的**：把超差的几项再量一次，比如重新标定测头以后。
- **复制结果**：把表复制成文字，列之间用制表符，给电子表格用。
- **生成检验报告...**：出报告（见检验报告）。

整张表在停靠区的 **量件结果** 页。每一行是一项检查：**设计** 值、**公差**、**量到** 的值、**偏差**、**孪生** 的估计和结论：✓，或 X 加超出多少。超差的行是红的，三维视图里也把结果标在零件上。

项	设计	公差	量到	偏差	孪生	
孔 1 直径	Ø8	H7 +0.015/0	8.001	+0.001	8.004	✓
孔 1 位置 X	8.000	±0.2 (m)	8.000	+0.000	8.001	✓
Y	32.000	±0.3 (m)	32.000	+0.000	32.000	
孔 1 圆度	○	0.015 (K)	0.002		0.003	✓
孔 1 轴线的垂直度	⊥	0.4 (K)	0.000		0.001	✓
孔 2 直径	Ø8	H7 +0.015/0	8.001	+0.001	8.004	✓
孔 2 位置 X	52.000	±0.3 (m)	52.001	+0.001	52.001	✓
Y	32.000	±0.3 (m)	32.001	+0.001	32.000	
孔 2 圆度	○	0.015 (K)	0.002		0.003	✓
孔 2 轴线的垂直度	⊥	0.4 (K)	0.000		0.001	✓

每次量件都存成一条记录。记录写一次就不改，它的 SHA-256 进审计日志（见拿一份副本对审计日志）。

AI 助手用 `inspect_part` 工具，量法一样。跟它说，比如：“用测头量 bracket.nc 的零件”。

1. 助手先说要量什么，以及形位检查要多碰几下、多花几秒。
2. 测头动之前，你确认。结果：回答里列出每一项检查：量到的值、孪生的估计和结论。

公差

每一项检查都按一个公差判。尺寸自己的公差或配合来自 CAM 作业，就是在尺寸框里写的（比如 8 H7）。别的尺寸按一般公差。

一般公差（ISO 2768-1）

ISO 2768-1 有四级：f（精密）、m（中等）、c（粗糙）、v（最粗）。没有自己公差的尺寸，按所在尺寸段取这一级的公差。在卡片的 **一般公差** 里选级别；或者跟助手说要哪一级，比如：“按 ISO 2768-m 量”。

形位：平面度、垂直度、圆度、直线度（ISO 2768-2）

在级别后面加 H、K 或 L，比如“mK”。测头就会再查形位：

检查	怎么量
平面度	底面按 3 × 3 的格子碰
垂直度	墙或孔的轴线在第二个深度再碰一次
圆度	孔或凸台碰 8 个点
直线度	长墙碰 3 到 8 个点

测头动之前，卡片或助手会说形位检查要多碰几下。

> **说明**
>
> 没有按方向标定的测头有它自己的三瓣误差。报告在每个圆度旁边写出孪生的测头模型估计它有多大。

配合（ISO 286）

写了配合的尺寸，比如 Ø8 H7，按配合的公差带判。见配合和公差。

倒角和 V 形槽

倒角的宽、深、角度，和 V 形槽的宽、深、角度，没给自己的公差时按 ISO 2768-1 的 f 级判。

孪生估计什么

孪生在它模拟的工件上做同样的触碰。它的估计就是程序在孪生里切出来的样子。量到的值和估计不一样时，看机床和装夹：刀长、工件零点、夹具、刀具磨损。

孪生的估计是它的物理模型算出来的，不是机床的测量不确定度。

检验报告

每次检验都存成一条记录。Studio 从记录出报告：

- CSV 文件（UTF-8），给电子表格用；
- PDF 文件，打印、归档用。

报告列出作业、装夹、刀具、测头和它的标定，以及每个特征的：名义尺寸、公差和它的依据、孪生的估计、量到的值、偏差、结论。没量的特征写明原因，还附孪生工件的俯视图。

要出报告，跟助手说，比如："出最近一次检验的报告"。同一条记录出的文件逐字节相同。

报告放在哪里

检验报告和装夹单都放在报告文件夹里：

- Windows：文档\Twinwright Studio\Reports
- Linux：~/Documents/Twinwright Studio/Reports

在设置里改过程序目录的，报告文件夹在那个目录里面。

拿一份副本对审计日志

Studio 把每条记录、每份报告和装夹单的 SHA-256 都记进审计日志，每一条和前一条链在一起。别人发给你的副本，可以这样核对：

- 跟助手说："拿审计日志核对这个文件"，并给出文件的路径。
- 或者在写出它的那台电脑上运行：

```
mc1sim --inspect-verify <文件>
```

回答会说日志里有没有这个文件的 SHA-256、它是什么时候写的、日志的链是否完整。改了一位的副本是找不到的。

装夹单

装夹单是给机床前操作的人的一页纸，写明要装什么、按什么顺序装。Studio 每个作业出一份，用你当前的语言。纸张是 A4；在美国和加拿大是 Letter。

要出装夹单，跟 AI 助手说，比如："出 bracket.nc 的装夹单"。助手会用 `setup_sheet` 工具。

> **说明**
>
> 装夹单在专业版里。

装夹单上有什么

- 程序：每个程序的名字、版本、行数，和它的 SHA-256 的前 8 位十六进制数。拿去和机床上装的程序对一下。
- 毛坯和夹具：毛坯的尺寸和材料，以及孪生怎么装它。平口钳要写垫铁、夹持多少、毛坯高出钳口多少。
- 刀具：每把刀的号、种类、直径、刃数、伸出、规格原文，和用到它的工序。
- 每次装夹：工件怎么过来（翻面、换一种夹法），工件零点在哪里，用哪个测头循环找。循环的名字和测头面板上的一样。
- 行程：程序在工件坐标里走到的 X、Y、Z 的最小值和最大值。
- Z 净空：见 Z 净空。
- 工序：每个工序的策略和下刀方式、刀具、转速、进给、深度，以及它要的时间。
- 俯视图：毛坯、平口钳的钳口、工件零点，和每个特征的轮廓。
- 注意事项：CAM 的提示，是英文的。

时间是按程序估算的：程序里的进给，和机床的快移速度、加速度。换刀不计。在孪生里试运行一遍，可以得到计入加减速的时间。

Z 净空

装夹单逐次装夹检查：测头和每把刀能不能越过装夹；每把刀能不能够到程序的最高点和最低点。高度用孪生 Z 预算的那一套，从工作台面算起。

项目	要够到
测头	Z 接近行程顶时，高出装夹最高处 1 mm
每把刀，往上	高出装夹最高处 0.5 mm，并够到程序的最高点
每把刀，往下	它自己那几个工序的最低点

每一行写 **够**，或者用红字写不够、差多少毫米。

> **注意**
>
> 有红字的行，说明这个程序在这种装夹上跑不了。换伸出更短的刀或更短的测头，或者把工件放低，比如用压板。

压板可能比程序的最高点还高，这时装夹单会写出来：程序不能从压板上方经过。在孪生里试运行会查这一点。

打印和签字

1. 用浏览器打开装夹单。
2. 打印。结果：第一次以后的每次装夹都从新的一页开始。
3. 在机床前填末尾的框：操作人、日期、首件检验（合格还是不合格，以及它的检验报告编号）。

同一个作业再出一份装夹单，会替换原来的文件。它的 SHA-256 记在审计日志里。

预览：这一页的安全提示还没有经过母语审校。

连真机

草稿（第 14 轮）：正文在写，各节还会补。

用真机的两种做法

- 用你自己的发送软件（bCNC、gSender 或 Universal G-code Sender）经 USB 口开 MC-1。见发送软件。
- 让孪生在旁边跑影子模式（预览）：你的发送软件连 Studio 而不连机床，Studio 把每一行转给机床，在孪生里跑同样的行，并比较两边。

用 Studio 自己把程序发给真机，1.0 里没有。

影子模式（预览）

> **说明**
>
> 1.0 里的影子模式是预览，从命令行启动。它只用第二个孪生顶替机床测过，还没接过真机。窗口里还没有地方选串口、连接、看比对结果，以后的版本加。

启动影子模式

要先有：MC-1 开着、用 USB 接到这台电脑；装好你的发送软件。

1. 找到机床的端口。Windows 上：**设备管理器 > 端口（COM 和 LPT）**，例如 COM5。
2. 在命令提示符里带着这个端口启动 Studio：

   ```
   TwinwrightStudio.exe --shadow COM5
   ```

   结果：Studio 打开这个端口，孪生跟着机床走。

3. 在你的发送软件里，连 Studio 的发送端口 `localhost:2323`，不连机床。结果：你的发送软件发的行经 Studio 到机床，孪生跑同样的行。

Studio 的发送端口只对本机开放。要从别的电脑连，再加 `--bind any`。

孪生比什么

- 应答，逐行：每一行的 `ok` 或错误，以及报警的先后。
- 路径：机床报的位置对孪生的路径，孪生领先或落后多少。
- 设置：机床连上之后、你的发送软件连上之前比一次。所以等 Studio 连上机床，再连你的发送软件。
- 控制器：和设置一起比一次。两边是不是同一种固件；两边都是同匠自研内核时，再比它的版本和机床板子上有什么：主轴、编码器、手轮、轴、测头、连接方式。孪生没有建模板子的辅助口和 SD 卡：比对会列出它们，但不算不同。
- 程序各段的用时。

守护

启动 Studio 时再加 `--shadow-guard`，孪生会在你的发送软件发的每一行到机床之前先检查它。孪生预计会撞时，就把这一行拦下，让机床进给保持。窗口里在 **诊断** 下面显示守护的状态。

什么到不了机床

影子模式下，到得了机床的只有你的发送软件发的行，和孪生自己的、不改变任何东西的问话（设置和状态）。

- 窗口里的操作只作用在孪生上；会让机床动或启动的，在孪生跟着机床时一律拒绝。测头面板是锁着的。
- AI 客户端只能读：它看得到机床的状态，什么都不改。
- 接在 Studio 上的外接手轮只作用在孪生上。

> ⚠ **警告**
>
> 影子模式下，窗口里的停止和急停停不了机床。要停机床，用它自己的急停，或你的发送软件的停止。

发送软件

和同匠自研内核一起测过的发送软件，以及怎么设：发送软件。

不一样的时候

出现不同，说明孪生的档案或设置不是这台机床的，或者机床做了模型没有的事。先查设置：比对会列出不同的那几项。

AI 助手

AI 可以用和你一样的工具在孪生上干活：装夹、写程序、检查程序，并在仿真里运行它们。你可以和 Studio 自带的助手对话，也可以连接 Claude 桌面版这样的 AI 客户端。凡是会改动窗口里那台机床的，都要等你同意。AI 不属于任何安全功能。

打开 AI

新装的 Studio 里，AI 是关着的。要打开它，用下面任何一种办法：

- 点顶栏里 AI 的状态（**AI：关**），选 **窗口的 AI 开关**；
- **工具 > 设置… > AI 与 MCP**：**AI 开关**，然后点 **确定**；
- 在助手里发一条消息：这会把开关打开。

Studio 记着你离开时开关的样子，下次启动照此。**AI** 菜单（**AI 助手…**、**AI 活动…**、**连接 AI 客户端…**）从操作员级别起显示。

开关关着时，AI 客户端仍然可以在后台使用练习机，但不能用你窗口里的机床。

AI 服务和它的密钥

助手和你选的一家 AI 服务对话，用你自己的账号：

- **Anthropic（Claude）**、**OpenAI**、**DeepSeek**、**通义千问（阿里云百炼）**、**Kimi（月之暗面）**、**智谱 GLM**；
- **LM Studio（本机）** 或 **Ollama（本机）**：你自己电脑上的模型，不用密钥；
- **其他兼容 OpenAI 的接口**：任何带 OpenAI 兼容接口的服务，地址由你填。

设置办法：

1. 打开 **AI > AI 助手…**
2. 选 **提供方**。Studio 会填上它的 **地址**。
3. 把你的密钥粘贴到 **密钥** 里，点 **保存密钥**。结果：密钥加密存在这台电脑上。在 Windows 上，只有你的 Windows 账户能读它。在 Linux 上它存进桌面的密钥环（Secret Service：GNOME 密钥环、KWallet）；没有密钥环时，用一个只有你的用户账户能读的文件里的密钥加密，密钥下面那行会写明。
4. 点 **列出模型**，选一个 **模型**。

Studio 不把密钥写进它的设置、注册表或审计日志。要删除已保存的密钥，把框清空，再点 **保存密钥**。

> **说明**
>
> 发给 AI 服务的内容，会用你的账号、按那家服务的条款发到那里：你的消息；**附上窗口现在的程序、装夹和检查结果** 打开时，还有程序、装夹和检查结果；以及 AI 所用工具的结果。要让一切都留在你的电脑上，用本机模型。

第一次往某个云服务发消息之前，助手会说一次会发出什么。点 **发送** 接着发，或者点 **改用本机模型…** 换成 LM Studio，消息留在输入框里。**工具 > 设置… > AI 与 MCP** 列出你已同意发送的云服务；点 **重新询问**，助手会再问一次。

助手面板

AI > AI 助手… 在单独的一个窗口里打开助手。

- 在下面的框里输入。Enter 发送；Shift+Enter 换行。**停止** 停下回答。
- **附上窗口现在的程序、装夹和检查结果（有变化时）** 在每条消息里把你窗口里的内容交给 AI。默认是打开的。
- 对话里显示 AI 用的每个工具，成功了标 ✓，没成功标 ✗。
- AI 做的图（工件、刀路、与零件的偏差、对比）出现在对话里。双击一张图可以放大或另存。
- **新对话** 重新开始。关闭 Studio 时，对话不会保存。

关闭这个窗口只是把它隐藏；对话保留到你退出为止。

AI 能做和不能做的

开关打开时，AI 能：

- 读各台机器的状态，并回零、解锁、点动、发送几行 G 代码；
- 设好毛坯、刀具、工件零点和对比用的零件，探测和检验；
- 检查、仿真、优化、生成、编辑和载入程序，开始或重新开始一次运行；
- 读各次运行和它们的结果，比较它们，并为它们作图；
- 读审计日志和授权的状态；
- 讲解时在窗口里指给你看：切工作区、换 3D 视图的方向、显示轨迹和程序路径、换着色、翻到停靠栏的某一页、翻到准备面板的某一步——只改窗口显示的东西；
- 查这本手册，照手册回答；
- 改语言、地区、单位、币种和机时费——要你点了**允许**（见下文）才改，和**工具 > 设置…**里改的效果一样。

属于付费档位的工具（检验、报告、排料、优化）只在那一档里能用。

AI 不能改 Studio 的其他设置：界面级别、AI 开关、Studio 问你的方式、隐私、日志和外观；也不能改授权和审计日志。

改动机床之前先问

AI 改动窗口的机床之前（在顶栏里 AI 的菜单中，以及 **工具 > 设置... > AI 与 MCP** 里）有两个选项：

- **每次都问我**（默认）；
- **不问（AI 直接操作窗口）**。

AI 要改动窗口里的机床时（移动机床、上电、松开急停、载入或运行程序、改装夹或刀具），或者要改它能改的那几项设置时，机床条上方会出现一条栏。栏里写着是哪个 AI 在问、它想做什么，并在倒计时。

1. 读这个问题。
2. 点 **同意** 或 **拒绝**。Enter 不算回答：用鼠标，或按 Tab 移到按键上，再按 Space。

60 s 内没有回答，就算 **拒绝**。一次同意，两分钟内也涵盖这个动作自己的后续请求。

AI 停止机床、让机床进给保持、复位机床、按急停，或者只是读取时，Studio 从不先问。练习机和 AI 用来仿真的隐藏副本也不问。

连真机时只读

窗口里的孪生跟着一台真机时（影子模式，1.0 里是预览；见连真机），AI 只能读取和分析。它不能移动机床、改机床的装夹或工件坐标、在那里运行程序，而且不会来问：Studio 直接拒绝这些请求。练习机照常工作。

AI 活动

AI > AI 活动... 显示 AI 客户端在这台电脑上做的事：

- **后台机器**：AI 用的练习机，你在窗口里看不到它们（“window”是你窗口里的那台机器）；
- **运行**：最近 16 次仿真和优化；
- **连着的 AI 客户端**：谁连着、怎么连的，以及它们最后一次调用。

在这里可以停止一次运行、看一次运行的图、关掉一台后台机器，以及停止或启动孪生机主机。孪生机主机是管着练习机的后台程序。AI 客户端一连上，它就启动；没人用满 30 分钟，它就退出。

停靠区的 **AI 活动** 页列出 AI 对窗口里那台机床的调用：时间、调用、机器、谁和结果。

连接别的 AI 客户端（MCP）

支持 MCP（Model Context Protocol）的 AI 客户端可以使用 Studio 的工具。

1. 打开 **AI > 连接 AI 客户端...**
2. 点你的客户端：**Claude 桌面版**、**ChatGPT 桌面版（Codex、Work 模式）**、**Cursor**、**VS Code（GitHub Copilot）** 或 **LM Studio（本机的开源模型）**。结果：对 Claude 桌面版和 ChatGPT 桌面版，Studio 先显示它要对客户端的设置做的改动，你点 **确定** 之后才去做；客户端原来的文件留在它旁边。其他客户端会打开各自的安装页。
3. 完全退出这个客户端，再重新打开。
4. 在 Studio 里打开 **窗口的 AI 开关**。

别的客户端，用 **其他客户端：复制配置** 把配置复制下来，粘贴到客户端里。客户端在后台启动 `TwinwrightStudio.exe --mcp`；**命令和配置** 显示确切的命令。

只接受网址的客户端

只接受网址的客户端，要在 AI 菜单里打开 **本机网址（HTTP，给只接受网址的客户端）**，再用 **复制本机网址（HTTP）**。这个网址只在这台电脑上有效，还要带上随它一起的令牌。**换一个新令牌** 会让旧令牌失效。

各个工具、它们的参数和结果：MCP 工具。

记下了什么

工具 > 审计日志...（Ctrl+L）记录 AI 做的事，AI 的名字写在 **谁** 这一列里。把 **谁做的** 筛选成 **AI**，就只看 AI 的记录；双击一次调用，把它的所有记录放在一起看。

在默认的 **标准** 级别下，日志记下：

- AI 开关的打开和关闭，以及 Studio 询问方式的改变；
- 每个问题、你的回答，以及 AI 对窗口里那台机床做的每个动作；
- 每次工具调用，附带参数的简短形式；
- 客户端连上和断开；
- 助手的消息：摘要里记前 180 字节，记录里最多 2000 字节。

密钥、令牌和密码一律写成 `***`。

语言和单位

第一次启动时，Studio 跟着系统的语言。语言、单位和数字、日期、金额的写法，随时都能改。程序和文件不受影响。

换语言

选择 **工具 > Language / 语言**，再选一种语言：English、简体中文（Chinese, Simplified）、Français、Deutsch、日本語 或 Español。这个菜单的名字在每种语言里都一样，所以你读不懂界面语言时也能找到它。

窗口立刻变成新的语言，不用重启 Studio。控制台里已有的行保持原来写的语言。

工具 > 设置… > 常规 > 语言 还多一个选项 **跟随 Windows**：这时 Studio 每次启动都跟着系统的语言。Studio 没有语言包的语言，退回英语。

单位：毫米和英寸

用 **工具 > 单位** 选择 **跟随地区**、**公制（mm）** 或 **英制（in）**。改了立刻生效。

选英制时，Studio 显示：

量	公制	英制
长度	mm	in
进给	mm/min	in/min
切削速度	m/min	ft/min
力	N	lbf
转矩	N·m	lbf·in
功率	W	hp
温度	°C	°F

主轴转速仍用 rpm。点动步长变成 0.0005、0.005、0.05、0.5 和 2 in，点动速度变成 4、12、60 和 120 in/min。

单位改变的是 Studio 显示和读取的东西，不改你的文件。程序按它自己写的运行，用 `G20` 或 `G21`。Studio 生成的程序用毫米写（`G21`）。AI 的工具和日志用毫米。

地区：数字、日期和金额

用 **工具 > 设置… > 常规 > 地区** 设定数字、金额和日期的写法：**跟随 Windows（设置 > 时间和语言）**、**跟随语言**，或者列表里的一个地区。同一页上还有：

- **货币**：ISO 4217 代码，例如 USD、EUR 或 CNY；空表示跟随地区；
- **货币符号**：空表示用货币自己的符号；
- **机时费**：你一小时机时的成本，用于成本估算；0 表示不估算。

地区决定 Studio 的数字里的小数分隔符和千位分隔符，运行、版本和报告里的日期，以及报告的纸张大小（美国和加拿大用 Letter，别处用 A4）。属于机床的数字，比如位置和程序里的数，一律用小数点，不用千位分隔符。

输入数字

在输错了可能报废零件的字段里，Studio 严格地读你输入的数：CAM 的字段、**准备** 里的尺寸和位置、测头的长度，以及主轴转速。

- 只用一个小数分隔符：小数点或逗号（`13.3` 或 `13,3`）。
- 不要写千位分隔符。
- 可以在数后面写这个字段的单位：`13 mm`、`0.5 in`、`30°`。
- 用输入法输入的数字（全角数字）也能读。

你的地区用小数点写小数时，`1,000` 是个疑问：你指的是 1，还是 1000？Studio 不猜，它要你写明你指的是哪个。地区用逗号写小数时，`1,000` 就是 1。

“读作”是什么意思

你输入的写法和 Studio 写数字的方式不一样时，字段会显示 Studio 读到的值：输入 `13,3`，它显示 **读作 13.3**。用英寸时显示四位小数：**读作 0.3750**。继续之前，请先核对一下。

Studio 读不懂你输入的内容时，字段变红，下面一行说明原因，例如：“1,2,3”不是长度。Studio 不采用这个值：在你改正之前，字段保持它上一次采用的值。

英寸分数

用英寸时，长度字段也接受分数：

- `3/8`、`5/4`
- `1 1/2` 或 `1-1/2`
- `2 3/8"`、`1/2 in`（英寸符号可以是 `"`、`”` 或 `in`）

字段显示它读到的小数和分数：**读作 0.3750（3/8"）**。Studio 不接受带小数点的分数（`1.5/2`）、整数后面跟假分数（`1 5/4`），也不接受斜线两边有空格。

Studio 也把英寸尺寸显示成分数，最细到 1/64 in：刀具的直径和长度、毛坯的尺寸、孔的直径和深度，写成 `1/4"` 或 `1 1/2"`。不是 1/64 整数倍的尺寸，显示四位小数，如 `0.2010 in`。坐标、进给、转速和测量值一律用小数。

用毫米时不读分数："毫米的字段不收分数：写 4.7625 或换成英制"。

刀具描述在两种单位制下都读英寸尺寸，例如 `flat 1/4in n2`。

主轴转速

除了每次按 1000 rpm 一档地调，你也可以直接输入主轴转速：

1. 在 **运行** 里，打开右栏的 **手动**。
2. 双击 **主轴转速** 读数，或者在它上面按 Enter。
3. 输入转速，按 Enter。按 Esc 或单击别处，保留原来的转速。

主轴在手动转动（没有程序在运行）时，新转速立刻送给主轴。主轴停着时，Studio 记住这个转速，留给下一次 **主轴正转** 或 **主轴反转**。程序运行时，转速由程序定，字段打不开。

超出主轴范围的转速会让字段变红，不被采用。MC-1 的范围是 1000–24000 rpm。

哪些保持英文

- G 代码，以及 Studio 写和读的文件（作业、装夹、录制、授权文件）；
- AI 的工具，以及发往 AI 客户端和从 AI 客户端发来的内容；
- 从程序本身生成的接口参考章节：MCP 工具、命令行和机床档案 schema；
- 每条审计记录里供程序读取的消息（日志里还会用你的语言存一份）；
- 1.0 里的德语、法语、日语和西班牙语：控制器的错误和报警说明、控制器设置的帮助，以及部分授权消息。

检验报告的 PDF，在它的字体有你的语言所用的字时用你的语言，没有就用英语；它的 CSV 文件总是用你的语言。

预览：这一页的安全提示还没有经过母语审校。

键盘和无障碍

机床、菜单、面板上的键和编辑器，都能用键盘操作。转动和缩放三维视图、在里面选一个点、调整面板大小，要用鼠标。在 Windows 上，读屏软件能读面板上的键和读数。

帮助 > 快捷键（或 **?**）列出 Studio 里的每一个键。

操作机床的键

键	作用
F1	**急停**：按下；再按一次松开
F2	**上电**
F3	**回零**
F4	**解锁**
F5	**循环启动**
F6 或 Space	**进给保持**
F7	**停止**：先进给保持，机床停稳后再复位；再按一次，立即复位
F8	**冷却水** 开或关
F9	**主轴正转**
F10	**主轴反转**
F11	**主轴停**
F12 或 Ctrl+X	**复位**

点动和手轮：

键	作用
← →	点动 X
↓ ↑	点动 Y
Page Down, Page Up	点动 Z
Shift 加以上任一个键	按住键时一直点动
1–5	步长 0.01、0.1、1、10 或 50 mm
6	**按住连续**：每个点动键在你按住时一直点动
Ctrl+1 到 Ctrl+4	点动速度 100、300、1500 或 3000 mm/min
Esc	**停止点动**
X, Y, Z	选择手轮的轴
, 和 .	把手轮向下或向上转一格（加 Shift：十格）

倍率：

键	作用
Ctrl+↑, Ctrl+↓	进给 +10 %、−10 %
Ctrl+→, Ctrl+←	主轴 +10 %、−10 %
Ctrl+0	进给和主轴回到 100 %

每按一下点动键，移动一个步长。按住的点动，在你松开键或窗口失去焦点时停止。用英寸时，步长是 0.0005–2 in，速度是 4–120 in/min。

F 键在哪里都能用。你在文本框、列表或编辑器里输入时，其余的机床键不起作用：单击三维视图，把它们还给机床。

数字键以及 , 和 . 键按它们在键盘上的位置工作，所以任何键盘布局都能用。

急停

按 F1，或单击机床条最右端的红色圆键。在孪生里，这会像机床上的按钮一样断开急停回路：驱动器和主轴驱动失去电源，控制器报 ALARM:10。再按一次 F1 松开它，然后照回到能干活的几步做：**上电**、**复位**、**回零**。

> ⚠ **警告**
>
> F1 和红色键只让孪生停下。它们不是真机的急停。在真机旁，请用机床自己的急停按钮。

操作窗口的键

键	作用
Alt+1 到 Alt+6	转到 **准备**、**CAM**、**程序**、**运行**、**分析** 或 **诊断**
Ctrl+Tab, Ctrl+Shift+Tab	下一个或上一个工作区
Ctrl+J	折叠或展开停靠区
Ctrl+Shift+M	切换机器
Ctrl+O	**打开程序...**
Ctrl+P	**打印程序...**
Ctrl+,	**设置...**
Ctrl+L	**审计日志...**
Ctrl+Q	**退出**
? 或 /	**快捷键**
Alt 加菜单名后括号里的字母	打开那个菜单；单按 Alt，菜单标题亮起
Tab, Shift+Tab	在面板的键之间移动；Space 按下带焦点框的键
Esc	关闭弹出层或子窗口

> **说明**
>
> F1 是急停，不是帮助。要打开手册，用 **帮助 > 文档**，或面板上的 **?** 键。

程序 工作区里的编辑器有自己的键：

键	作用
Ctrl+E	进入编辑器，再回到机床
Ctrl+K	**载入并检查**
Ctrl+S	**存版本**
Ctrl+Shift+S	把程序存成文件
Ctrl+D	与一个存过的版本 **对比**
Ctrl+F, Ctrl+G, Ctrl+H	查找、跳到某一行、替换
Ctrl+B	这一行的 **断点**
Ctrl+Shift+B	**条件暂停**
Ctrl+.	采用检查对这一行的建议
Alt+↓, Alt+↑	下一个或上一个发现的问题或不同之处
Shift+F10 或菜单键	这一行的菜单
Ctrl+Shift+O	**优化**
Ctrl+Shift+G	**生成...**
Ctrl+Shift+R	换成机床里的程序
Esc	关闭对比

手册的窗口：Ctrl+F 搜索，Alt+← 和 Alt+→ 后退和前进，Esc 关闭它。

在 **准备** 里摆放工件时：方向键移动它（加 Shift 每步更小，加 Ctrl 每步更大），Q 和 E 转动它，Enter 放下它，Esc 放回原处。

完整的列表

帮助 > 快捷键 把上面所有的键放在一张列表里。下面这些命令没有键，请用它们的菜单：

- **单段** 和 **下一段**（**程序** 菜单和机床条）；
- **检查程序**；
- 给各轴清零和回到零点（**机器 > 工件零点**）；
- 对刀仪和测头的动作（**机器 > 对刀仪与测头**）；
- 按 1000 rpm 一档调整主轴转速，以及快移倍率（**机器**）；
- 暂停仿真和调整它的倍速（**仿真**）。

用读屏软件

在 Windows 上，Studio 通过系统的辅助功能接口和读屏软件配合。Studio 的每个面板里，读屏软件能得到：

- 每个键的名字、它是哪一类键、它是开着、展开着还是不能用、它的快捷键和它的提示；
- 文字形式的读数：机床条（状态、程序、进度、下一步、时钟）、顶栏（机器、状态、AI、级别、档位）和右栏（位置、主轴、表盘、模态行）；
- 用 Tab 移动焦点时的每一次移动；
- 菜单：每一项的文字、快捷键、是不是不能用或勾着，以及你在哪一项上。

有两样事一发生就会播报，不管焦点在哪儿：报警（连同下一步），以及 AI 请求改动机床。

手册的窗口把页面的文字连同标题和链接交给读屏软件。

在 1.0 里：

- 机床状态的其他变化不会主动播报：把焦点移到机床条上才能听到；
- 程序清单、控制台、消息、编辑器和刀具表按读屏软件自己的规则读；曲线不读；
- 在 Linux 上不支持读屏软件。

三维视图里读什么

三维视图本身不给读屏软件任何内容。它显示的数字也在右栏和机床条里，读屏软件读的是那里。你单击工件上的一点时，视图下面的那句话（程序行和位置）会被读出来。

大字和车间模式

Studio 跟随你在 Windows 里设的显示缩放，每块屏幕各按各的。它自己没有字号设置。

视图 > 外观 > 车间模式（大号） 让机床条、它的按键、急停、文字、程序清单和位置变大，并给 **运行** 工作区一个适合隔一臂远阅读的布局（见浅色、深色和车间模式）。

系统的动画关闭时，三维视图直接跳到新视图，不再滑过去。

常见问题

事情没有照你想的那样走时怎么办。每一节说你看到什么、通常是什么意思、完整的答案在哪一章。都对不上的话，收集诊断信息写信给我们（最后一节）。

Studio 打不开

- Studio 要 64 位的 Windows 10 或 11；别的版本上安装包会停下。见系统要求。
- 窗口打开了，三维视图却一直是空的，或者说它用不了：**工具 > 设置… > 文件**，勾选 **软件渲染（显卡出问题时用；下次启动生效）**，再启动一次 Studio。更新显卡驱动往往能解决根本原因。见三维视图起不来时。
- Studio 在启动时关掉了的话，下次启动会提出那份报告（见Studio 意外关闭）。

机床报警了

报警时，机器栏上的状态变红。栏上写着是哪个报警、什么意思，**下一步：** 后面是现在该做的事和做这件事的按键；不止一步时，**全部步骤** 列出回到工作的路。点这个键，或用 **机器 > 做下一步**：每做完一步，Studio 都重新算一遍。

- 探测移动什么也没碰到，就以 ALARM:5 停下：照步骤做（**复位**、**解锁**），再看循环原以为那个面在哪里。
- 移动超出机床行程时会停在软限位：看程序列表里标红的那一行，和运行时用的工件零点。
- 报警之后位置可能不再可信（**位置** 下面会有一行说明）：切削之前重新回零。

每种情况的步骤：报警和怎么办。所有报警、原因和怎么解除：报警。

程序的某一行回了错误

`error:` 加一个号，意思是控制器拒绝了那一行：它没有执行，机床也没有为它动。消息里有号码和它的意思。

- **程序 > 检查程序** 让整个程序过一遍控制器的检查模式，机床不动，并列出它拒绝的头几行：“第 5 行：error:22（…）”。改好这些行，再查一遍。
- 最常见的原因：控制器不认的代码；缺了字（圆弧没有圆心或半径，切削移动之前没有 `F`）；同一行里有不能放在一起的两个代码。

所有错误和原因：错误。

程序检查发现问题

编辑器在你打字时、载入时和运行之后检查程序。每条发现写在它那一行下面，带规则号和类别：错误、警告或提示。

- **错误** 是控制器会拒绝的行，或者装夹承受不了的移动；运行之前先改。
- **警告** 是能运行、但未必是你想要的行：在工件顶面以下快移；启动主轴没给 `S`；换刀之后还没到起点上方就往下走。
- 检查里的预览部分要机床已上电、已回零、没有程序在运行；在那之前它会等着。

规则和怎么改：程序检查。在孪生里运行还会查什么：孪生查什么。

探测循环开不了

测头卡上有一项检查是红的，或者缺了什么，**开始探测…** 就是灰的；按键下面那一行会说缺什么（给出 X 和 Y、给出环规直径、等正在进行的调用结束）。

- **主轴上没有测头**：在 **准备** 的 **对刀** 一步选测头，点 **换刀**。
- **主轴在转**：先停（**主轴停**，F11）。
- **毛坯上方没有横移的空间**：那一行给出 Z 在顶时测头尖的高度、装夹的顶面和差多少。换短一点的测头，或者把毛坯夹低一点：用压板或双面胶代替平口钳。
- **孪生在跟随一台真机**：到真机上用它自己的控制来探测。
- 循环什么也没碰到时以 ALARM:5 停下（见机床报警了）。

见循环停下时。

授权和激活

帮助 > 授权… 显示现在用的版本、授权的状态和到期时间。

- **在线激活**：照收到的样子输入激活码；读得通时 **激活** 才能点。服务器的回答显示在它下面。回答是授权已在太多电脑上、已到期、已暂停、未付款或已退款时，**打开授权网页** 会打开官网上的授权页：在那里释放电脑、续期或管理订阅。
- **没有网络**：同一个对话框里有离线的步骤。Studio 给出请求码；在一台能上网的电脑上，到授权网页为它取授权文件，再回来导入。
- 订阅没有续上时，还能照常用 14 天（授权显示 **宽限中**）。之后，或者授权到期时，Studio 回到学习版；项目和程序不受影响。

见授权和授权到期时。

AI 助手不回答

- 输入框下面那一行说助手正在做什么（等服务回答、孪生在执行一个工具）。
- 查 **提供方**、**地址** 和 **模型**；云服务还要查密钥保存了没有（**保存密钥**）。
- 本地模型要它的服务器在运行（LM Studio 的服务器、Ollama），并且选好了模型（**列出模型**）。
- 回答可能要一会儿；**停止** 可以结束它。

- 第一次给云服务发消息时会先等你：Studio 列出这次要发给它的东西，你同意以后才发（每个云服务只问一次）。**改用本机模型…** 能让一切都留在这台电脑上。见隐私。
- AI 想改动窗口里的机器时，窗口底部会有一条栏问你。60 秒内没有回答，就当拒绝。

见AI 助手。

Studio 意外关闭

下次启动时，Studio 问你怎么处理那份记录了经过的报告：打开它所在的文件夹，或另存为文件。什么也不发出。报告里不含程序、文件和文件夹的名字、密钥、激活码和人名；超过 30 天的报告在启动时删除。见隐私。

求助

帮助 > 报告问题… 把出的问题、不知道怎么用的地方或者建议发给同匠的支持。写清楚你做了什么、期望什么、实际看到什么；想让回复也发到邮箱就填上邮箱。自动带上、对话框里写明的（**看全文…** 能看完整的）：版本和构建号、版本档位、系统和它的语言、屏幕，还有审计日志最后 50 条的摘要——做了什么、结果如何，不带程序内容，你的用户文件夹写成 `~`。**附上诊断包** 和 **附上窗口截图** 打开时都不勾，不勾就不发。按 **发送** 之后对话框里写工单号（例如 `TW-1042`）。没连上网时报告存成一个文件 `problem-<时刻>.zip`，在"文档"文件夹的 `Twinwright Studio\Reports` 里：能上网时在对话框里再发一次，或者在官网支持页上传。**存成文件…** 只存不发。

AI 客户端可以替你起草（report_draft）：打开的是同一个对话框，带着它写的话，还有一行写明是哪个客户端起草的。你看一遍，哪里都能改；只有你能发，诊断包和截图也只有你能勾。

帮助 > 我的问题 列出从这台电脑发出的报告：工单号、状态（已收到、处理中、已回复、已解决）、最后一次变动的时间；有没读的回复时前面一个点。选中一个看来回的对话；**补充一句…** 往同一个工单里再写（**附上诊断包…** 把诊断包一起附上）。回复在客户中心的"我的工单"里也看得到。有没解决的报告时，Studio 每 30 分钟问一次有没有新回复，帮助菜单上写有几条新的；**设置 > 常规 > 每 30 分钟看一次我的问题有没有新回复** 可以关掉。发什么、存多久：见 隐私。

帮助 > 收集诊断信息… 把支持需要的东西放进一个 ZIP 文件，存在"文档"文件夹的 `Twinwright Studio\Reports` 里：版本、设置（密钥写成 `***`）、机床档案、控制器的设置、最近 7 天的审计日志、崩溃报告和授权的状态。**现在的程序和装夹** 不勾就不放进去。路径、文件名和电脑的用户名都去掉了。

发出去之前先打开看一看；这个文件不会自己发出去（报告只在你勾了 **附上诊断包** 时才带上它）。写信的地址在 **帮助 > 关于**。

运维手册

系统要求

运行同匠 Studio 的电脑要什么，三维视图对显卡要什么。

Windows

- 64 位的 Windows 10 或 Windows 11（x64）。在其他版本上，安装程序会停止。
- 不用再装别的东西：Studio 是一个程序文件，只用 Windows 自带的库。

Linux

- x86-64 上的 Linux，glibc 2.38 或更新：Ubuntu 24.04、Debian 13、Fedora 39 及以后的版本，以及基于它们的发行版。
- 有 X11 或 Wayland 的桌面。
- AppImage 需要 FUSE 2（Ubuntu 上是 `libfuse2t64` 包）。没有的话，用 `.tar.gz`。
- 联网激活和试用：要有 libcurl（Ubuntu 上是 `libcurl4t64` 包，桌面系统一般都有）。没有的话，Studio 用离线激活。
- AI 服务的密钥：存在桌面的密钥环里（Secret Service：GNOME 密钥环或 KWallet）。没有密钥环时，Studio 把密钥存在一个只有你能读的文件里。

显卡

三维视图用 Direct3D 11，任何有它的显卡或集成显卡都行（功能级别 10.0 或更高）。没有的话，Studio 用 Windows 的软件渲染器（WARP）来画：能用，但视图较慢（每秒最多 30 帧），而且没有刀具特写。

帮助 > 关于 在 **显示** 下面显示 Studio 用的是哪一种。

三维视图起不来时

没有三维视图，窗口的其余部分照样能用。如果视图一直是空的，或提示它不可用：

1. 打开 **工具 > 设置… > 文件**。
2. 勾选 **软件渲染（显卡出问题时用；下次启动生效）**。
3. 关闭 Studio，再重新启动。

更新显卡驱动常常能消除原因。

Linux

在 Linux 上，三维视图用 OpenGL 4.1 或更高版本：近十年的显卡用 Mesa 驱动都有。没有显卡驱动时，Mesa 的软件渲染器也能画，只是慢一些；想试软件渲染，就从终端启动 Studio，在程序名前面加 `LIBGL_ALWAYS_SOFTWARE=1`。

内存和磁盘

- 内存：8 GB 或更多。
- 磁盘：程序和它的手册约 62 MB，再加上放你的数据的空间。你的数据会越用越多：审计日志默认保留 180 天、最多 2 GB（**工具 > 设置… > 日志**），每个程序版本和报告都会保留（见你的数据和备份）。

屏幕和缩放

- 至少 1280 × 720；Studio 的版面按 1600 × 1000 及更大的屏幕设计。
- 任何显示缩放比例都行。Studio 跟随每块屏幕的缩放比例，把窗口移到另一块屏幕上，它会重新缩放。

连真机

在 1.0 里，连真机是预览功能，从命令行连（见连真机）：用串口（USB），波特率 115 200，或者连网络上的控制器。

Studio 这样使用网络：

- G 代码发送软件连到这台电脑上的孪生，端口 2323；
- 浏览器视图用这台电脑上的端口 8765；
- 用地址连接的 AI 客户端使用这台电脑上的端口 8766，在你把它打开时；
- 在线激活和续期连到授权服务器。

Studio 的端口只在这台电脑上监听：别的电脑连不上。

在 Linux 上，串口属于 `dialout` 组：把你的账户加进去一次（`sudo usermod -aG dialout $USER`），再重新登录。

安装和卸载

怎么安装同匠 Studio，怎么静默地在很多台电脑上同时安装，什么文件放在哪儿，以及怎么卸载。

Linux 见 在 Linux 上。

安装

1. 运行 `TwinwrightStudio-<version>-setup.exe`。
2. 阅读许可协议，单击 **我接受。**
3. 保持默认的文件夹（`%LOCALAPPDATA%\Programs\Twinwright Studio`），或另选一个。
4. 在最后一页，选择是否启动 Studio，以及是否在桌面上放一个快捷方式。

安装程序有英文和中文两种界面，跟随系统的语言。

只给你装，还是给这台电脑上的每个人装

安装程序为运行它的那个人安装 Studio。它不需要管理员权限，开始菜单里的项和桌面快捷方式属于那个人。电脑上每个要用 Studio 的人，各自安装一次。

授权属于电脑，不属于个人：电脑上的每个用户共用它（见授权文件夹）。

Studio 不改文件关联，不改 `PATH`，不改防火墙设置，也不安装服务。

静默安装

```
TwinwrightStudio-<version>-setup.exe /S [/DESKTOP] [/D=<folder>]
```

- `/S` 不显示安装页面，也不启动 Studio。
- `/DESKTOP` 加上桌面快捷方式。
- `/D=` 指定文件夹。它必须是最后一个参数，不加引号，即使文件夹名里有空格。

要静默卸载 Studio：在程序的文件夹里运行 `uninstall.exe /S`。

什么放在哪儿

什么	在哪儿
程序	`%LOCALAPPDATA%\Programs\Twinwright Studio`
你的工作：程序、版本、报告、审计日志	`Documents\Twinwright Studio`
程序自己的状态：设置、机器、AI 密钥	`%LOCALAPPDATA%\Twinwright\Studio`
授权	`%ProgramData%\Twinwright\licence`

程序的文件夹

里面有 `TwinwrightStudio.exe`、手册（`manual\`）、`README.txt`、`THIRD-PARTY-NOTICES.txt` 和 `uninstall.exe`。更新时这个文件夹不变，所以你连过的 AI 客户端照常能用。

你的文档

`Documents\Twinwright Studio`（Windows 给你的“文档”文件夹，改到别处也一样）：

- `programs\`：每个程序的各个版本，和 CAM 作业的各个版本；
- `Reports\`：检验报告、装夹单和诊断信息文件；
- `logs\`：审计日志。

工具 > **设置...** > **文件** 和 **日志** 可以把程序和日志挪到别处（下次启动生效）。

程序自己的状态

`%LOCALAPPDATA%\Twinwright\Studio`：设置、窗口的位置、每台机器的“世界”（它的工件、刀具、偏置和控制器设置）、AI 服务的密钥（按你的 Windows 账户加密），以及意外退出的报告（保留 30 天）。

授权文件夹

`%ProgramData%\Twinwright\licence` 放着授权文件（含试用），电脑上的每个用户都用它。Studio 在第一次激活或试用时建这个文件夹。Studio 写不进去的话，管理员可以把文件夹建好，并让用户能往里写。

特殊的配置里，这些环境变量可以挪动文件夹：

变量	挪动什么
`TWINWRIGHT_STATE_DIR`	程序自己的状态

变量	挪动什么
`TWINWRIGHT_WORKDIR`	存放程序版本的文件夹
`TWINWRIGHT_LOGDIR`	审计日志
`TWINWRIGHT_LICENCE_DIR`	授权文件夹

更新到新版本

关闭 Studio，再运行新版本的安装程序。它装进同一个文件夹；你的工作、设置、机器和授权都保持原样。Studio 自己不检查更新（见更新 Studio）。

如果还有一份 Studio 在后台运行（可能是 AI 客户端启动的），安装程序会把旧的程序文件挪到一边，在下一次更新或卸载时删掉它。

卸载

在 Windows 里用 **设置 > 应用 > 同匠 Studio > 卸载**，或者在程序的文件夹里运行 `uninstall.exe`。

卸载后留下什么

卸载会删掉程序、它的快捷方式和它在 **应用** 里的条目。下面这些会留下：

- 你的工作，在 `Documents\Twinwright Studio` 里；
- 程序自己的状态，在 `%LOCALAPPDATA%\Twinwright\Studio` 里；
- 授权文件夹；
- 你连过的 AI 客户端的设置里的 `twinwright` 条目。

要全部清除，请手动删掉它们。删掉授权文件夹不会把试用还给你：授权服务器给每台电脑每档只发一次试用。

在 Linux 上

安装

Linux 版 Studio 是一个文件，一个 AppImage。系统里什么也不装。

1. 核对文件：`sha256sum TwinwrightStudio-<version>-x86_64.AppImage` 的结果要和下载页上的 SHA-256 一致。
2. 把它放到能一直放着的地方，名字里不带版本号，这样更新以后你连接的 AI 客户端照样能用；再加上可执行权限：

```
mkdir -p ~/Applications
mv TwinwrightStudio-<version>-x86_64.AppImage ~/Applications/TwinwrightStudio.AppImage
chmod +x ~/Applications/TwinwrightStudio.AppImage
```

1. 启动：`~/Applications/TwinwrightStudio.AppImage`，或者在文件管理器里双击它。

AppImage 需要 FUSE 2。如果因为缺它启动不了，就装上（Ubuntu 24.04 及以后：`sudo apt install libfuse2t64`），或者改用 `.tar.gz`：解压到任何地方，启动文件夹里的 `TwinwrightStudio`。

Linux 上什么文件放在哪儿

什么	在哪儿
程序，连同手册	那个 AppImage，你放在哪儿就在哪儿
你的工作：程序、版本、报告、审计日志	`~/Documents/Twinwright Studio`
程序自己的状态：设置、机床	`~/.local/state/twinwright/studio`（设了 `$XDG_STATE_HOME` 时在它下面）
AI 服务的密钥	桌面的密钥环；没有密钥环时，是状态文件夹里一个只有你能读的文件
授权	`/etc/twinwright/licence`，或 `~/.local/share/Twinwright/licence`

你的账户能建或者能写授权文件夹时，它属于这台电脑，每个用户共用。否则 Studio 把授权放在 `~/.local/share/Twinwright/licence`（设了 `$XDG_DATA_HOME` 时在它下面）：只给你用，而且仍然只在这台电脑上有效。要让这台电脑的所有用户共用一份授权，用 `sudo install -d -m 1777 /etc/twinwright/licence` 建一次这个文件夹。

Linux 上的试用和激活

试用、激活和续期和 Windows 一样联网进行，用的是系统自己的 libcurl。没有 libcurl 或者没有网时，离线激活（见授权）。

在 Linux 上更新和卸载

更新：退出 Studio，用新的 AppImage 替换旧的，名字不变（见上面第 2 步）。卸载：删掉那个 AppImage。两种做法都会留下你的文档、状态文件夹、授权文件夹、密钥环里的密钥，以及你连接过的 AI 客户端设置里的 `twinwright` 一项；要全部清掉，手动删除这些。

授权、席位和激活

没有授权时，Studio 按学习版运行。付费授权加上它那一档的功能。这一章写给买授权、激活和管授权的人。价格在官网上。

如果你读到这里时付费授权还没开放购买，官网的价格页上会写明。在此之前 Studio 按学习版运行，创客版、专业版都可以免费试用 30 天（见试用付费档）；讲激活的几节在开放购买后适用。

档位

档位	多了什么	谁可以用
学习版	孪生的全部仿真、编辑器、连真机、示例、AI 连接；CAM 的型腔、轮廓、钻孔、刻字和三维平行精加工	任何人，免费，用于学习、教学和不挣钱的个人项目
创客版	自适应开粗、全部三维精加工方式、残料加工、两面加工和多次装夹	一个人，包括出售这个人自己做的东西
专业版	创客版的全部，加上排料、程序优化器、用测头检验（与设计对比）和检验报告	你的单位，任何商业用途，按席位

MC-1 机主授权是一个席位的专业版，配合授权上写明序列号的那台 MC-1 使用；机器转让时随机器转给新机主。

付费授权有这几种：

- **订阅**：按月或按年，取消之前一直续。
- **买断**：永久归你，覆盖购买（或付费续更新）后 12 个月内发布的所有版本。
- **MC-1 机主**：和买断一样，跟着机器走。

席位和电脑

一个付费席位同一时间由一个人使用，最多用在三台电脑上；几个人共用的一台电脑只算一台。

每激活一次，授权就用这台电脑指纹的散列绑定这台电脑；指纹本身不离开电脑。要看授权绑了哪些电脑，打开授权网页 twinwright.com/zh/licence，输入激活码。

试用付费档

每个付费档可以免费试用 30 天，每台电脑每档一次。

1. 用一个付费功能，比如 CAM 面板里的自适应开粗。结果：Studio 说明这个功能要哪一档。
2. 选 **开始 30 天试用…**。
3. 看清试用多了什么、到哪天，再选 **开始试用**。结果：Studio 向同匠的授权服务器申请试用，到显示的日期之前，这一档的功能都能用。

申请试用时，Studio 只发送所试用的版本档、这台电脑指纹的散列和 Studio 的版本。服务器给每台电脑每档一次试用：试用期内重装了 Studio，再开始试用会拿回同一个到期日。到期后 Studio 回到学习版，项目和程序不受影响。

不能上网的电脑上，Studio 会改为显示一个请求码。在能上网的电脑上打开授权网页 twinwright.com/zh/licence，在"在不能上网的电脑上试用"里选好版本、粘贴请求码，下载试用文件；再照激活不能上网的电脑把文件导入原来那台电脑。

在线激活

需要：随授权发来的邮件里的激活码，形如 `TW-XXXXX-XXXXX-XXXXX-XXXXX`；能上网。

1. 选 **帮助 > 授权…**。
2. 选 **在线激活…**。
3. 输入激活码。大小写、空格和横线都不要紧。
4. 选 **激活**。结果：授权文件放进了授权文件夹，对话框里显示档位。

激活只发出激活码、这台电脑指纹的散列和 Studio 的版本。

要是授权已经用满了能用的电脑，先在授权网页上释放一台（见把授权换到另一台电脑）。

激活不能上网的电脑

实验室、车间里从不联网的电脑用这个办法。需要：激活码，另一台能上网的电脑，一个能带小文件的东西，比如 U 盘。

1. 在不能上网的电脑上选 **帮助 > 授权…**。
2. 选 **离线激活…**。
3. 选 **复制**。结果：请求码复制到了剪贴板。它只含这台电脑指纹的散列和 Studio 的版本，不含任何个人信息。
4. 把请求码粘贴进一个文本文件，带到能上网的电脑上。
5. 在那台电脑上打开 twinwright.com/zh/licence，在"不能上网的电脑"下面粘贴激活码和请求码。
6. 选"下载授权文件"。结果：下载一个形如 `twinwright-TWL-….twlic` 的文件。
7. 把文件带回去。在 Studio 里选 **帮助 > 授权…** 和 **导入授权文件…**，打开这个文件。结果：授权文件放进了授权文件夹。

不能上网的电脑不会自己续期。下面几种情况要再做一遍：

- 订阅本期结束以前；
- 买断的授权，过了头 30 天（退款期）以后再做一次，换成永久文件；
- 买了更多年的更新以后。

授权文件夹

授权文件夹是这台电脑的，不属于哪一个 Windows 账户；里面的授权对用这台电脑的每个人都有效。

系统	文件夹
Windows	`%ProgramData%\Twinwright\licence`
Linux	`/etc/twinwright/licence`；写不进时：`~/.local/share/Twinwright/licence`

环境变量 `TWINWRIGHT_LICENCE_DIR` 可以另指一个文件夹。要打开它，选 **帮助 > 授权...** 和 **打开授权文件夹**。

文件夹里有授权文件（`*.twlic`，含试用），和一个记着 Studio 上次看时间的文件。可以同时放几个授权文件，比如一份 MC-1 机主授权加一份订阅；现在的档位取所有有效授权文件里最高的一档。

> **说明**
>
> 不要改这些文件。改过的授权文件就不算数了：到授权网页上重新下载，再导入。

Studio 建不了或写不进这个文件夹时，由管理员建好，并让用户能写。在 Linux 上用 root（`sudo`）导入，或者设 `TWINWRIGHT_LICENCE_DIR`。

续期和 14 天宽限

打开 **到期前自动续期（只发出授权编号、这台电脑指纹的散列和 Studio 的版本）** 时，Studio 联网就自己续期：

- 订阅：本期结束前 5 天开始续；
- 买断：退款期满时续一次；买了更多年的更新，在更新到期前 14 天开始续。

Studio 在启动时试一次，每天最多一次；没续上也不打扰你。发出的只有这个选项里写的那几样。

订阅没续上时，所有功能再照常用 14 天，授权显示 **宽限中** 和剩下的天数。过了宽限期，Studio 回到学习版。

授权到期以后

到期、取消或核验不过，都只会回到学习版：

- 项目照常打开；
- 已经生成的 G 代码照常能导出、能发给机床；
- 付费的操作等你续期以后再做。

Studio 从不删你的文件，也不锁你的数据。

买断授权的更新

买断的授权覆盖购买（或付费续更新）后 12 个月内发布的版本。更晚发布的版本按学习版运行，授权显示 **更新已到期**。可以接着用它覆盖的最后一个版本，或者在授权网页上再续一年更新。见更新 Studio。

把授权换到另一台电脑

1. 在授权网页上输入激活码，选"查看我的电脑"。
2. 在不再用的那台电脑旁边选"释放这台电脑"。
3. 在新电脑上激活。

30 天里最多释放三台电脑；还要释放，写信到 hello@twinwright.com。

换了硬件或重装了系统，指纹会变，Studio 会说 **这份授权绑定的是别的电脑。** 先在网页上释放旧的那一项，再重新激活；在这之前 Studio 按学习版运行。

激活码丢了

用授权网页上的 **激活码丢了?** 。那里要你填邮箱时，填购买时用的邮箱：用它买的每份授权都会收到一封带新激活码的邮件。那里请你写信时，用收到激活码的那个邮箱写信到 hello@twinwright.com，我们核对后给你一个新码。两种办法下，旧码都不能再激活，已经激活的电脑照常使用。

电脑的时钟

时钟被往回调了两天以上时，Studio 会说 **这台电脑的时钟比上次早。** 按较晚的时间算，授权和试用都不会因此多出天数。请把日期和时间调对。

Studio 把它见过的最晚时间记在两处，按较晚的算，所以删掉其中一处不会多出天数。

如果时钟曾被设到将来、授权或试用因此显示提前结束了，请把日期和时间调对，并写信到 hello@twinwright.com。

给管理员：命令行

程序带上这些参数时不开窗口。文件夹可以不写，不写就用这台电脑的授权文件夹。

```
TwinwrightStudio.exe --licence-status [FOLDER] [--json]
TwinwrightStudio.exe --licence-request [FOLDER]
TwinwrightStudio.exe --licence-import FILE [FOLDER]
```

- `--licence-status` 列出每个授权文件和试用：成不成立、为什么；加 `--json` 给出全部字段。
- `--licence-request` 打出这台电脑的请求码。
- `--licence-import` 核对一个授权文件，再拷进授权文件夹。

在 Linux 上程序叫 `twinwright-studio`。AI 客户端可以用只读的 MCP 工具 `licence_status` 读到同样的状态。

Studio 发出什么、我们的服务器留什么：见隐私和 Studio 发出的东西。

你的数据和备份

Studio 把你的工作放在你的文档里，机器的状态放在它自己的文件夹里。这一章列出什么放在哪里、该备份什么，以及怎么搬到另一台电脑。

什么在哪儿

有两个文件夹放着所有属于你的东西：

- `Documents\Twinwright Studio`：你的工作；
- `%LOCALAPPDATA%\Twinwright\Studio`：程序自己的状态。

作业、程序和它们的版本

`Documents\Twinwright Studio\programs\<program>\` 里，程序的每个版本一个文件（`v0001.nc`），另有一个说明文件（`v0001.json`），记着它是什么时候、由谁、为什么保存的。做出这个程序的 CAM 作业也按版本存在它旁边（`job-v0001.json`）。Studio 从不清理它们。

工具 > 设置... > 文件 可以把程序文件夹放到别处（下次启动生效）。

报告

`Documents\Twinwright Studio\Reports\`：检验报告（PDF 和 CSV）、装夹单（HTML）和诊断信息（ZIP）。

机器和它们的世界

每台机器在状态文件夹里有一个世界：工作台、带着切痕的工件、刀具和刀具表、工件坐标、控制器的设置、温度、运行小时数和测头的标定。你工作时，机床每次有变化后静止满两秒，Studio 就写入它一次；关闭窗口时再写一次。第一台机器的世界在 `world\` 里，其他机器的在 `machines\<id>\world\` 里。

设置

状态文件夹里的 `settings.json` 放着 **工具 > 设置...** 里的全部内容；`window.json` 放窗口的位置和大小。

审计日志

`Documents\Twinwright Studio\logs\` 里每天一个文件（`audit-YYYY-MM-DD.jsonl`）。记录是链在一起的，所以校验日志时（**工具 > 设置... > 日志 > 校验完整性**），改动或缺口都会显出来。默认日志保留 180 天、最多 2 GB；**工具 > 设置... > 日志** 可以改。要让链保持完整，就复制整个文件夹。

备份什么

先关闭 Studio，再复制这两个文件夹：`Documents\Twinwright Studio` 和 `%LOCALAPPDATA%\Twinwright\Studio`。

有两样东西不会随复制一起搬走：

- AI 服务的密钥：它们是按这台电脑上你的 Windows 账户加密的。到新电脑上重新输入；
- 授权：它属于电脑。到新电脑上重新激活（见授权）。

搬到另一台电脑

1. 在旧电脑上关闭 Studio，复制这两个文件夹。
2. 在新电脑上安装 Studio，启动一次，再关闭。
3. 把这两个文件夹复制到新电脑上同样的位置。
4. 你在设置里挪过程序文件夹或日志文件夹的话，检查这些文件夹是否存在。
5. 启动 Studio，激活授权，重新输入 AI 密钥。
6. 重新连接你的 AI 客户端（**AI > 连接 AI 客户端...**）。

让一台机器从头开始

恢复出厂把窗口里的机床恢复成刚来时的样子：它的世界没了（工件、刀具、工件坐标、控制器的设置、测头的标定），Studio 重新启动。

1. 选 **文件 > 恢复出厂...**（技师级别及以上）。
2. 5 秒内再选一次来确认。

只重置这一台机器。你的程序、报告、审计日志、设置和其他机器都保留。重置会记进审计日志。

更新 Studio

Studio 不自己查新版本，也从不自己更新。什么时候更新由你定。

看有没有新版本

1. 在 Studio 里选 **帮助 > 关于**。结果：程序名下面那一行是你的版本，比如"版本 1.0.0 · 2026-10-10 构建 · 提交 4f1c2ab"。
2. 打开下载页 twinwright.com/zh/download。结果：页面上写着现在的版本和安装包的 SHA-256 校验值。

两边的版本不一样，下载页上的就是新的。

安装更新

需要：新的安装包。不需要管理员权限：Studio 装在你的 Windows 账户下。

1. 从下载页下载安装包。
2. 可选：核对它的 SHA-256 校验值和页面上的是否一样。在 PowerShell 里对这个文件运行 `Get-FileHash`。
3. 运行安装包。要是 Windows 说发布者未知，不要运行，写信到 hello@twinwright.com。
4. Studio 开着的话，安装包请你关时就关掉。
5. 照安装包的提示做完。结果：新版本在同一个文件夹里替换了程序。

你的设置、AI 密钥、程序版本、审计日志和授权都不在程序文件夹里，照旧不动。接好的 AI 客户端照常能用，因为程序的路径不变。

电脑多时，安装包可以静默安装：`setup.exe /S`。见安装 Studio。

你的授权覆盖哪些版本

授权	能用付费功能的版本
学习版	所有版本
订阅	订阅期间的所有版本
买断、MC-1 机主	购买（或付费续更新）后 12 个月内发布的版本

授权的更新期过了以后才发布的版本，按学习版运行，授权显示 **更新已到期**。更新之前，先在 **帮助 > 授权…** 里看一下日期；把授权覆盖的最后一个版本的安装包留好。

要接着更新，在授权网页上再续一年更新。见授权、席位和激活。

不能上网的电脑

1. 在能上网的电脑上下载安装包。
2. 用 U 盘之类把它带到不能上网的电脑上。
3. 照上面的步骤安装。

新版本不会改授权文件，续期照旧：见激活不能上网的电脑。

退回旧版本

1. 更新之前先备份你的文件。
2. 卸载 Studio：在 Windows 设置的"应用"里。
3. 运行旧版本的安装包。

> **说明**
>
> 旧版本不一定能打开新版本存的项目或设置。确定没问题之前，留着备份。

预览：这一页的安全提示还没有经过母语审校。

隐私和 Studio 发出的东西

Studio 在你的电脑上工作。Studio 自己发出去的，只有开始试用、激活和续期付费授权要用的东西：没有遥测，不自己查新版本，不自动上传崩溃信息。你把 AI 助手接到云服务时，你的消息和附上的内容会发给那家服务，见AI 助手和云服务。这一章列出发出去的、留在电脑上的，以及怎么删掉。以官网的隐私政策 twinwright.com/zh/privacy 为准。

Studio 发出什么

什么时候	发给谁	发什么
你开始试用一个付费档	同匠的授权服务器，走 HTTPS	所试用的版本档、这台电脑指纹的散列、Studio 的版本
你激活付费授权	同匠的授权服务器，走 HTTPS	激活码、这台电脑指纹的散列、Studio 的版本
Studio 续期付费授权（见续期）	同匠的授权服务器，走 HTTPS	授权编号、同一个散列、Studio 的版本

Studio 自己发给同匠的就这些。指纹本身不离开电脑，离开的只是它的散列。离线激活的电脑什么都不发。

下面三样只在你自己设了以后才会发出去：

- **接到云服务的 AI 助手**：把你的消息和附上的内容发给那家服务，见下面。
- **你经 MCP 接的 AI 客户端**，比如 Claude、ChatGPT：客户端把它从孪生读到的东西发给它自己的服务，按它的条款处理。
- **Studio 打开的网页**，比如授权网页：由你的浏览器打开，Studio 不往里加任何东西。

AI 助手和云服务

AI 助手用你在里面选的提供方：云服务（比如 Anthropic、OpenAI、DeepSeek），用你自己的账户和 API 密钥；或者本机模型。

你给云服务发消息时，Studio 从你的电脑、用你的密钥，把这些发给那家服务：

- 你的消息，连同到这时为止的对话；
- 勾着 **附上窗口现在的程序、装夹和检查结果（有变化时）** 时：窗口里的机床、装夹、带行号的程序和检查发现的问题。这一项一开始就是勾着的。Studio 随第一条消息附上它们，以后只在它们变了时再附；
- 助手在 Studio 里调用工具得到的结果，比如程序的检查或仿真、机床的状态、日志里的记录；对接受图片的服务，还有结果的图。

第一条消息发往云服务之前，Studio 会提示一次要发什么。

这些都按那家服务自己的条款处理；同匠收不到其中任何东西。想少发：发送前去掉勾，或者按 **新对话** 重新开始。

用本机模型（LM Studio、Ollama）时，这些都留在电脑上。

在这台电脑上，审计日志记下每条消息的前 180 字节（见审计日志），API 密钥加密保存（见Studio 在电脑上留什么）。

Studio 在电脑上留什么

什么	在哪儿（Windows）	谁能看到
设置、孪生的状态、AI 密钥	`%LOCALAPPDATA%\Twinwright\Studio`	你的 Windows 账户
程序版本	`文档\Twinwright Studio`	你的 Windows 账户
审计日志	`文档\Twinwright Studio\logs`，设置里另指了文件夹的除外	你的 Windows 账户
检验报告、装夹单、诊断信息文件	`文档\Twinwright Studio\Reports`	你的 Windows 账户
意外退出的报告	`%LOCALAPPDATA%\Twinwright\Studio\crash`，30 天后删掉	你的 Windows 账户
授权文件（含试用）	`%ProgramData%\Twinwright\licence`	这台电脑上的每个人

AI 密钥按你的 Windows 账户加密保存（Windows DPAPI），同一台电脑上的别的账户读不到。

审计日志

审计日志每天一个文件 `audit-YYYY-MM-DD.jsonl`，在日志文件夹里。它记下：

- AI 和你让机器做过的事；
- AI 面板里每条消息的前 180 字节；
- 授权核验：授权编号、档位和结果，不记持有人的名字。

密钥、令牌、密码一律记成 `***`。记多少、留多久在设置里定。日志只在你的电脑上，Studio 从不上传。

Studio 意外关闭时

出事的那一刻，Studio 写一份很小的报告：在哪儿出的错、出错前的调用顺序、程序的模块表。不写内存里的内容。

下次启动时，Studio 加上版本信息和日志的最后 200 行，先清洗一遍：

- 路径和文件名换成占位；

- 名字和电子邮箱去掉；
- 不带 G 代码、AI 对话、授权码和电脑指纹。

然后问你一次：可以打开报告所在的文件夹、把报告存成文件，或者关掉提示。什么都不发出。想帮我们查原因，就自己把文件发到 hello@twinwright.com。

诊断信息

找我们帮忙时，我们可能请你给一份诊断信息。在 **帮助** 菜单里选"收集诊断信息"。Studio 把你勾选的几项存成一个 ZIP 文件，放进报告文件夹：

项目	默认勾选
版本、系统、显卡、语言和单位	是
设置，不含 AI 密钥	是
机床档案和控制器的 `$` 设置	是
最近 7 天的日志和意外退出的报告，按上面的办法清洗过	是
授权状态：档位和到期日，不含授权码和电脑指纹	是
现在的程序和装夹：你的零件数据	否

发出去之前，先打开文件看一遍。什么都不会自动发出。

网络上谁在听

Studio 给 AI 客户端（MCP）和发送软件开的服务只在这台电脑上听（`127.0.0.1`），别的电脑连不上。

命令行参数 `--bind any` 会把孪生的网络端口（比如给发送软件的那个）开给你的网络；MCP 服务仍然只在本机。

> ⚠ **警告**
>
> 影子模式下，发送软件的端口会把命令转给真机。用了 `--bind any`，你网络上任何能连到这个端口的人都可能让机器动起来。只在孪生上、只在你自己管的网络里用 `--bind any`。

我们的服务器留什么

对每份付费授权，授权服务器留下：

- 每个席位最多三台电脑的散列，以及每台第一次和最后一次出现的日期，好让你在授权网页上释放一台；
- 激活码只存散列，连我们自己也读不出来；
- 试用、激活、续期和授权变动的日志，留 90 天：电脑的散列、Studio 的版本、截到网段的 IP 地址。

每次试用，服务器留下这台电脑的散列、版本档和试用的起止日期，这样每台电脑每档只能试用一次。

和你的工作有关的东西，一样都到不了服务器。

删掉 Studio 留下的东西

1. 在授权网页上释放这台电脑，让它不再占名额（见把授权换到另一台电脑）。
2. 卸载 Studio。
3. 删掉上面表里你不再需要的文件夹。结果：电脑上不再留有 Studio 的任何东西。

卸载只删程序，不删你的文件：项目、程序版本、日志和报告都留着，等你自己删。

实验室、教室和共用的电脑

这一章写给在几个人共用的电脑上装 Studio 的人：实验室、车间或教室。

一份授权管整台电脑

授权文件夹是这台电脑的，不属于哪一个 Windows 账户（见授权文件夹）。电脑上激活过的授权，对登录这台电脑的每个人都有效。

一个付费席位同一时间由一个人使用，最多用在三台电脑上；几个人共用的一台电脑只算一台。所以：

- 共用的电脑，比如机床旁边那台，单给它一个席位；
- 有人在自己的电脑上也要用付费功能，那些电脑另要席位。

共用的电脑只要激活一次，用管这台电脑的人的账户。

每个人自己的东西

Studio 装在各自的 Windows 账户下。每个人有自己的：

- 设置和 AI 密钥：密钥按这个账户加密，别的账户读不到；
- 程序版本、报告和审计日志，在各自的 `文档\Twinwright Studio` 里。

一个档位的试用对这台电脑只算一次，电脑上的人共用：授权服务器给每台电脑每档只发一次试用。

在很多账户、很多电脑上安装，见安装 Studio。

教室

学习版在每台电脑上都免费，用于学习和教学，什么都不用激活：装上 Studio，孪生的全部仿真、编辑器和日常零件的 CAM 就都能用。

能的话，给每个学生一个自己的 Windows 账户，各人的设置、程序和日志就分开了。

要用付费功能教学，学校和培训机构可以买专业版席位，写信到 hello@twinwright.com。

每次开机都还原的电脑

有些实验室的电脑每次开机都回到保存的镜像，授权文件夹也随之清空。这样的电脑：

1. 先离线激活一次（见激活不能上网的电脑），把它的授权文件留好。
2. 把文件放进镜像里的授权文件夹，或者每次开机时导入：

```
TwinwrightStudio.exe --licence-import D:\licences\lab-pc-3.twlic
```

一个授权文件只对一台电脑有效。机房里每台电脑都要有自己的请求码和自己的文件。

不能上网的电脑

离线激活，再手动续期：订阅本期结束以前，以及买断授权过了头 30 天以后各做一次。见激活不能上网的电脑。

授权文件夹写不进去时

在 Windows 上，任何账户都能建 `%ProgramData%\Twinwright\licence`。建它的账户能改里面的文件，别的账户能读。这就够了：一个人导入授权，大家都能用。

单位的策略不允许的话：

- 由管理员建好这个文件夹，并让用户能写；或者
- 用系统环境变量 `TWINWRIGHT_LICENCE_DIR` 指一个大家都能读的文件夹。

在 Linux 上，你的账户能建或者能写 `/etc/twinwright/licence` 时，文件夹就是它；否则 Studio 用你自己的 `~/.local/share/Twinwright/licence`（设了 `$XDG_DATA_HOME` 时在它下面），只给你用。要让这台电脑的所有用户共用一份授权，用 `sudo install -d -m 1777 /etc/twinwright/licence` 建一次这个文件夹。

共用电脑上的 AI

每个人接自己的 AI 服务或 AI 客户端，用自己的密钥。每个账户的审计日志记下这个人和 AI 让机器做过的事。见隐私和 Studio 发出的东西。

技术手册

孪生的模型和它们的局限

孪生就是写成程序的 MC-1：同匠自研内核在里面跑，控制器周围的机床部分由物理模型代替。孪生给出的关于实物的数，都是物理模型的估计。它的参数来自机床的设计、零件的手册和公开资料；每台 MC-1 计划在调试时实测，并按实测设定它的孪生。下面每节最后写这一节的参数依据什么。

草稿（第 14 轮）：正文在写，各节还会补。

控制器

孪生里的控制器就是同匠自研内核，和机床上的固件从同一份源码编出来。模拟的是它周围的板子：定时器和步进脉冲，带去抖的输入，和电脑之间的链路（USB 串口和它的握手），控制器前台用的时间。

依据：内核自己的代码；前台用时在板子上测过之前是估计。

运动和驱动

每根轴由闭环步进驱动器和滚珠丝杠带动。孪生分四级细节模拟它们，默认是三质量模型：

- 驱动器的位置环：电机在指令位置附近的磁弹簧，力矩随转速下降，负载太大时失步，跟随误差太大时驱动器报警；
- 导轨、螺母和轴承的摩擦，Z 的重力；
- 电机、丝杠和工作台之间传动的柔度，螺母的反向间隙，X 轴下面的机架；
- 驱动器的供电：急停回路、接触器、驱动器上电要多久。

依据：设计（质量、惯量，刚度来自手算和有限元分析），驱动器和电机的手册；摩擦和驱动器内部参数在测过之前是假设值。

几何和精度模型

打开时，孪生按机床的几何误差移动刀尖：每根轴的导程误差和它的周期部分，两个直线度和三个转角误差（按阿贝臂作用），轴之间的垂直度，螺母的反向间隙，丝杠和主轴受热的伸长。限位开关看的是滑座，对刀仪和测头看的是刀尖，和机床上一样。这个模型默认关：名义孪生是理想的。

依据：按零件等级取的一台典型装好的机床的表。它不是机床的规格；调试时用这台机床的 ISO 230-2 测量换掉（计划）。

热

主轴和滚珠丝杠用着会升温、停着会冷下来；孪生在两次会话之间记着它们的温度，并按温度移动刀尖、改丝杠的导程。

依据：一阶热模型，参数在测过机床的预热之前是假设值。

主轴和变频器

主轴的变频器照机床上的模拟：控制器经 RS-485 用变频器自己的协议和它通讯，按设定的波特率一个字节一个字节地传，变频器照它的手册应答，包括决定它听不听的那些参数。后面是：按转差算的电机转矩，电流和变频器的过流、过载、欠压跳闸，短时掉电时撑住变频器的直流母线，水泵。

依据：变频器和电机的资料；母线电容、电流的励磁分量和摩擦在测过之前是假设值。

切削

毛坯是一张高度场，你要的地方可以加细。刀具每走一步，就去掉它的外形扫过的部分，所以零件的形状来自程序本身。

切削力来自材料的切削系数（随切屑厚度和切削速度变化），按刀具真实的外形（平底、球头、牛鼻、V 刀）逐齿算。孪生由切削力估计主轴的负载和功率、刀具的变形和断刀的危险：刃根的弯曲应力对比硬质合金的强度。

依据：各材料公开的切削数据（每种材料的出处写在机床档案里），锋利的刀，典型的刀具几何。

颤振

每一段切削，孪生按刀具和机床的振动模态、这一刀的几何，算出在当前转速下会开始颤振的切深，并挑一个切削稳定的转速。它还按切削的强迫振动估计墙面最后在哪里。

刀尖每个方向有三个模态：机床的结构、装在夹头和主轴里的刀柄、刀自身的弯曲。弯曲模态的频率按刀的形状算——刃部、颈、柄和伸出长度：6 mm 立铣刀伸出 20 mm 约 14 kHz，0.2 mm 的刀超过 100 kHz。1 mm 以下的刀如果颤振，多半是在它自己的弯曲上，切深几乎不随转速变；这个切深孪生每一刀都在时间上算。

依据：机架的模态来自有限元分析，刀柄的模态是假设值，刀的弯曲按刀的形状用梁的理论估算；三个模态的阻尼都是假设的（3 %）。这些都还没有实测：要在造好的机床上做敲击试验测，测之前稳定切深都是估计值。切削本身的阻尼没有建模：对小刀，裕度偏保守。

跳动

刀的轴线偏离主轴时，最外面的刀齿在切：孔切大、凸台切小（发现 187）。孪生按这个包络切；CAM 在配合里补偿它。刀具规格里可以写这把刀自己的跳动——按精密夹头的等级，或者按实测（`tir3`：它的 TIR，单位微米）；不写就用机床的。

依据：测过之前是假设的跳动。

刀具磨损

1.0 里每把刀都是新的：不管切了多久，它切出的形状和受的力都按锋利的刀算。为每把刀记程序用掉的寿命，计划在以后的版本里做。切铝的刀多半先因为铝粘在刃口上、零件出毛刺或表面变差而换，而不是因为后刀面磨损。

测头

测头是一种刀具：它的球碰到零件时，测头输入翻转。孪生按毛坯的形状和切过的表面（包括斜面）精确算碰到的那一刻。有两种模型：理想测头，球一碰就翻转；典型测头，预行程随方向不同，有三瓣、散布，推得太远测针会断。

依据：触发式测头的典型值；用它测量之前，先在环规上标定。

Z 净空

对于台面上的装夹，孪生算出要越过的最高点，以及主轴上的刀和刀具表里每把刀在 Z 最高处时刀尖的高度。越不过装夹的刀，会在装夹或刀具改变时报出来（发现 188）。

依据：机床的设计，和给定的刀长。

碰撞和装夹

孪生检查刀具、刀柄和主轴与零件、夹具和机床之间的碰撞，快要碰撞之前就提醒。装夹（压板、虎钳、V 形块、胶带、固定）用一个力夹住零件；切削推得更大时，零件会打滑。

依据：机床和夹具的几何来自设计；夹紧力是估计。

没有模拟的

- 墙面位置以外的表面纹理和粗糙度；
- 积屑瘤、切屑和排屑；
- 薄壁在切削下的变形；
- 零件自己的发热；
- 刀断了以后的事。

要造好机床才能做的验收（驱动器和声音的测量、切削力、表面、热、机架模态、影子模式接真机）写在孪生的设计文档里（docs/06 第 7 节）。

同匠自研内核

同匠自研内核是同匠自己的数控控制器固件。在 Studio 里，它控制孪生里的机床；同一份代码也编成 MC-1 控制板的固件。

它说 Grbl 1.1 的串口协议，G 代码发送软件照常能连。设置、报警、错误的编号都是发送软件认得的那一套。

它做什么

- **运动：** 前瞻规划 100 个块，拐角速度按 `$11`，圆弧走真圆。运动按 1 ms 一片执行。每根轴都守住自己的最高速度（`$110`–`$112`）和加速度（`$120`–`$122`）。
- **机床的功能：**
  - 回零、硬限位和软限位、探测和对刀仪；
  - 经 Modbus 的主轴驱动（变频器），以及冷却；
  - 按 `$341` 的方式换刀；
  - 带停车的安全门；
  - 倍率、点动和手轮。
- **控制器上的程序：** 控制器文件系统里的程序可以用 O 字子程序和宏（见 O 字）。
- **G 代码：** 照 RS274/NGC 和 LinuxCNC，包括刀具表、刀具半径补偿、路径融合、表达式和参数（见 G 代码和 M 代码）。

串口协议、实时命令和 `$` 命令见协议。全部设置见设置，报警见报警码，错误见错误码。

和其他 Grbl 控制器相比

为 Grbl 写的发送软件都能连这个控制器。用惯了别的 Grbl 控制器的人会注意到下面这些不同。

控制器会停的地方

下面这些情形，有的 Grbl 控制器照常往下走；同匠自研内核会停下或报警。

- **主轴驱动自己停了**（跳闸、过载，或有人按了它面板上的停止）：0.2 s 内停进给，报 `ALARM:14`。
- **主轴驱动不应答：** 轴先减速停下，位置不丢，再报 `ALARM:19`。
- **换刀：**
  - 换刀时主轴一定停。
  - 换刀没有确认就接着走时，主轴和冷却恢复到换刀前的样子。
- **进给保持或安全门之后恢复：** 主轴先回到转速，刀再动。
- **停车回去的路上又开门：** 机床停下，重新抬到停车的高度，等下一次循环启动。
- **发送软件断开了**（拔了 USB、关了端口）：作业停下。由 `$1030` 管，默认开。
- **测头和回零开关：** 线上的毛刺不当成碰到。
- **程序里的 UTF-8 字**（中文、“Ø”、“–”）当文字读，不会被当成实时命令。
- **R 格式的圆弧在平面里没有弦时**（比如 `G91 G2 X0 Z1 R5`）答 `error:33`。

更多的 G 代码

下面这些，有的 Grbl 控制器报错或不做，同匠自研内核照标准做：

- 刀具表（`G43 Hn`、`G43.2`、`G10 L1/L10/L11`）；
- 刀具半径补偿（`G41`、`G42`）；
- 路径融合（`G64 P`）和精确停止（`G61.1`）；
- O 字子程序和流程控制，`#` 参数和表达式；
- `G52`、极坐标（`@`、`^`）和 `M99`；
- `G85`、`G86`、`G89` 真的钻、真的镗，有的 Grbl 控制器收下却不动；
- CAM 后处理常写的：带名字的 `%` 行、`G49 H0`、Fanuc 的程序号行（`O1000`）、Centroid 的 `M25`。

更贴程序

- **圆弧** 走真圆。有的 Grbl 控制器把它切成弦，离圆最多 4 × `$12`。
- **反时间（`G93`）下的圆弧** 按编程的时间走。有的 Grbl 控制器快好几倍。
- **停留（`G4`）** 编多长就停多长。有的 Grbl 控制器按 50 ms 一档凑整。
- **啄钻：** 每个孔只钻该钻的刀数。孔深正好是啄钻深度的整数倍时，有的 Grbl 控制器会从退刀高度再多钻一整刀。
- **软限位附近的大圆弧** 不误报警。
- `M72` 会恢复 `M70` 或 `M73` 存下的刀长补偿。存下的是 0、偏置方式（`G43.1`）又和现在一样时，有的 Grbl 控制器不恢复，留着后来设的值，所以同一个程序在 `M72` 之后，在那样的控制器上可能是另一个刀长补偿。
- **加速度：** 每根轴都守住 `$120`–`$122`，小圆弧上也一样。所以小圆弧特别多的程序，比在有的 Grbl 控制器上最多慢约 2 %；一般的程序快约 0.1 %。

协议上的细节

- **状态报告：**
  - 进给保持或开门时，`FS:` 里的进给报 0：轴停着。
  - 剩余距离一直报到轴停下。
  - 严重报警里 `Bf:` 报规划器全空。
  - `$10` 打开解析器状态时，`[GC:]` 每变一次只发一行。
- **输入检查：**
  - 写坏的设置号（比如 `$45x`）答 `error:3`。

    - 整数设置写成 `$1=25.0` 存 25；有的 Grbl 控制器存 250。
    - `M6` 和运动写在同一行答 `error:40`。
    - `G65` 或 `G66` 和别的命令写在同一行答 `error:21`。
- **报警和开机：**
    - 报警后不停 100 ms。
    - 第一次开机、设置存储是空的时，不报 `error:7`。
- **说明文字和设置：**
    - 设置、报警、错误的说明是同匠自己写的。
    - 控制器有几个自己的设置：`$1030`（发送软件断开就停），`$1000` 起是输入整形（默认关）。
- **`$39=0`**（关掉老式实时字符）：`?`、`!`、`~` 在 `$` 行和注释里是文字，在别处照常是命令。控制器按收到的字节判断，所以开门或进给保持时也照答 `?`；有的 Grbl 控制器那时有时不答。

不支持的

- `G65 P7`（Modbus 消息）。
- Fanuc 的系统变量。`#4120` 这类当普通参数读，读出 0。
- 没有主轴编码器时的螺纹（`G33`、`G76`、`G33.1`）。MC-1 没有编码器，这几个答 `error:20`。
- 车削的粗车循环（别的 Grbl 控制器也没有）。
- 网络、Trinamic 驱动、别的牌子的主轴驱动（有的 Grbl 控制器以插件提供）。

发送软件

验过的发送软件和它们的设置见 G 代码发送软件。要拿真机和它的孪生对比，见真机和影子模式。

G 代码和 M 代码

同匠自研内核执行下面这些 G 代码。代码照 RS274/NGC 和 LinuxCNC。控制器不支持的代码答 `error:20`，这一行什么也不做。

实时命令（`?`、`!`、`~` 这样的单个字节）见协议。和其他 Grbl 控制器有哪些不同，见同匠自研内核。

G 代码

代码	意思	说明
`G0`、`G1`	快移、直线进给	
`G2`、`G3`	顺时针、逆时针圆弧（I J K 或 R），螺旋线	刀走真圆。R 格式的圆弧在平面里没有弦时答 `error:33`。
`G4 P`	停留 P 秒	编多长就停多长。
`G5`、`G5.1`	三次样条、二次样条	不带 I、J 的 `G5` 顺着上一段结束时的方向接下去。
`G7`、`G8`	车床的直径方式、半径方式	只在车床方式（`$32=2`）下。
`G10 L2`、`G10 L20`	设工件坐标	
`G10 L1`、`G10 L10`、`G10 L11`	设刀具表	32 把刀。控制器关电后刀具表还在。
`G17`、`G18`、`G19`	选平面	
`G20`、`G21`	英制、公制	
`G28`、`G30`	经中间点去存着的位置	
`G28.1`、`G30.1`	存下现在的位置	
`G33`、`G33.1`、`G76`	主轴同步运动、刚性攻丝、螺纹循环	要主轴编码器。MC-1 没有，这几个答 `error:20`。
`G38.2–G38.5`	探测	结果是测头触发的那一点，报成 `[PRB:]`。零点按这一点设，不按轴停下的地方设。
`G40`	关刀具半径补偿	
`G41`、`G42`、`G41.1`、`G42.1`	刀具半径补偿：D 取刀具表里的半径，`.1` 直接给直径	只在 G17 平面。进入、退出的动作照 LinuxCNC。会过切的动作被拒绝。
`G43 Hn`、`G43.2 Hn`	按刀具表的刀长补偿、加上的补偿	
`G43.1`	动态刀长补偿	
`G49`	关刀长补偿	
`G50`、`G51`	关缩放、开缩放	
`G52`	局部坐标系	和 `G92` 共用一份偏置，照 LinuxCNC。
`G53`	这一行用机床坐标	
`G54–G59`、`G59.1–G59.3`	工件坐标系	`$486` 能锁住 `G59.1–G59.3`。
`G61`	精确路径	
`G61.1`	精确停止	每个编程动作的终点轴都停下。
`G64 P Q`	路径融合	能省时间的地方，两段直线之间的拐角变成一段离拐角不超过 P 的圆弧。Q 收下但不用。
`G65 P1–P6`	自带的宏	读写设置、刀具偏置、参数、机床状态、选测头、主轴不等转速。`P7`（Modbus 消息）不支持。
`G65 P100` 起、`G66`、`G66.1`、`G67`	文件宏、模态宏调用	P 要整数。这一行不能有别的命令（`error:21`）。
`G73`、`G81`、`G82`、`G83`	钻孔循环、啄钻	每个孔只钻该钻的刀数。
`G84`、`G85`、`G86`、`G89`	攻丝、镗孔循环	
`G80`	取消循环	
`G90`、`G91`	绝对、增量	
`G91.1`	圆心用增量	
`G92`、`G92.1`、`G92.2`、`G92.3`	坐标偏置：设、清除、暂时停用、恢复	
`G93`、`G94`、`G95`	反时间、每分钟、每转进给	反时间下的圆弧按编程的时间走。
`G96`、`G97`	恒线速、恒转速	只在车床方式下。
`G98`、`G99`	循环退回到起始高度、退回到 R 平面	

下面这些 G 代码不支持（`error:20`）：

- `G5.2`、`G5.3`（NURBS）；
- `G17.1–G19.1`（UVW 平面）；
- `G70–G72`（车削循环）；
- `G90.1`。

M 代码

代码	意思	说明
`M0`、`M1`	程序暂停、选择停止	实时命令 `0x88` 打开或关掉 `M1` 的停止。
`M2`、`M30`	程序结束	
`M3`、`M4`、`M5`	主轴正转、反转、停	
`M6`	换刀	`$341` 定 M6 做什么。M6 和运动写在同一行答 `error:40`。
`M7`、`M8`、`M9`	冷却：喷雾、浇注、关	在 MC-1 上，M8 开的是主轴冷却水。
`M25`	Z 回到回零后的高度	Centroid 的代码，有的 CAM 后处理会写。
`M48`、`M49`	开、关倍率	
`M50–M53`	进给倍率、主轴倍率、自适应进给、进给保持：开或关	
`M56`	开、关停车	
`M60`	换托盘暂停	
`M61`	不换刀，直接设当前刀号	
`M62–M65`	数字输出开或关，随下一段动作或马上	板子有辅助输出时才有。
`M66`	等输入	前面的动作走完了才开始等。
`M70–M73`	存下、恢复模态	
`M98 P L`、`M99`	调子程序、返回	在从控制器的文件系统里跑的程序中用。一行一行送来的 `M98` 答 `error:39`。主程序里的 `M99` 当程序结束。

下面这些 M 代码不支持（`error:20`）：

- `M19`（主轴定向）；
- `M67`、`M68`（模拟输出，MC-1 没有）；
- `M100–M199`（用户 M 代码）。

字、语法和参数

程序行

- 可以写注释 `( … )` 和 `;`。
- `(MSG,…)` 把文字作为消息发给发送软件。
- `(PRINT,…)` 发一条消息，里面写的参数换成它们的值。
- `(DEBUG,…)` 也这样，只在 `#5599` 打开时发。
- 可以用跳段 `/`、行号 `N`，以及 `%` 行（带不带名字都行）。
- 大小写不论，字中间的空格不算。

参数和表达式

- 编号参数 `#1–#5000` 归程序用，程序没设过时读出来是 0。
- `#5001` 起的参数和带名字的系统参数只能读。
- 带名字的参数 `#<name>`：读一个从没设过的名字答 `error:39`。`EXISTS[#<name>]` 告诉你它设过没有。
- `[ ]` 里的表达式，运算符和函数照 LinuxCNC。角度用度。
- 一行里设的值，这一行自己读不到。

O 字

- 从控制器的文件系统里跑的程序，可以用子程序、`if`、`elseif`、`else`、`while`、`do`、`repeat`。
- 一行一行送来的子程序和循环先收齐，收齐了再跑。
- 子程序也可以单独成一个文件：`<名字>.ngc` 或 `<名字>.macro`。

极坐标

`@` 是半径，`^` 是角度，在 XY 平面里绕工件原点。

一行里的错误

- 同一模态组的两个代码写在一行，答 `error:21`。
- 一行里有两个都要用轴字的命令，答 `error:24`。
- 全部错误码见错误码。

CAM 策略和参数

每种策略做什么、参数是什么意思、孪生怎么定切削参数。孪生给出的数都是它的物理模型的估计。

下面的参数是作业里特征的写法（`program_generate` 工具的 features）。长度单位是毫米，角度是度。

清型腔：环切、往复和自适应

型腔的区域是它的形状减去岛屿，再向内偏一个刀具半径。一层一层地清，每层不超过切深（`ap`）。策略（`strategy`）决定一层里怎么走：

策略	走法	适合
`rings`（默认）	区域的偏置环，从最深处往外走到墙，每圈相隔 `ae`	大多数型腔
`zigzag`	按 `angle` 的方向来回直走，相隔 `ae`	开敞、浅的区域
`adaptive`	处处让刀的吃刀量保持在 `ae` 附近	深型腔、硬一些的材料

有配合的型腔，墙另走一刀切到尺寸。见配合。

自适应开粗用较小的径向吃刀量、较深的 `ap`。各段之间的连接移动避开材料；区域太窄走不开时，退回到环切。

精加工一圈在壁的内角上，刀被包住的角度比在直壁上大得多。包角超过直壁的 1.2 倍时，那一段壁先分步切（`corner_steps`，默认开；特征自己给了 `finish_ap` 时不分，除非同时给 `corner_steps: true`；`corner_steps: false` 关掉）。

残料加工

写 `rest: true` 时，型腔只清前面那些刀留下的料：更大的刀够不到的角和窄处。剩下多少，Studio 按同一个程序里前面的型腔算；孪生里有工件时按孪生的工件算。

轮廓和桥

轮廓（`side`：`outside`、`inside` 或 `on`）沿外形线一层一层地切，顺铣。`outside` 和 `inside` 把外形线偏一个刀具半径；`on` 就沿外形线本身走。

桥（`tabs`：个数、宽、高）在底部留下短短的连接，让切下来的零件留在原处。没给高度时桥高 1 mm；切深不超过 1 mm 的地方，桥高取切深的一半。

开槽：分层和摆线

槽（`x0`、`y0`、`x1`、`y1`、`width`）按策略切：

- `layers`（默认）：先在槽中间按整个刀宽分层切，再往两边扩到槽宽；
- `trochoidal`：沿槽走小圈，每圈前进 `step`。刀只在一小段弧上吃刀，所以一刀可以切得更深。槽要比刀宽。

下刀：螺旋、斜线、垂直

立铣刀不直接扎进实料。每一层这样开始：

- 区域里放得下螺旋时，走螺旋（直径和角度写在回答的 `entry` 里）；
- 放不下时，沿刀路斜线下刀；
- 只有在孔里或毛坯外面才垂直下刀。

回答里会写出每个工序第一次是怎么下刀的。

孔

钻孔循环

在 `points` 各点钻 `depth` 深的孔，用控制器的循环。孔深超过钻头直径的三倍就啄钻，每次 1.5 倍直径，除非 `peck` 另给。循环的退刀平面在顶面上方 1 mm。

螺旋插补镗孔

比立铣刀大的直径 `d` 的孔，走螺旋切出，到底再切到尺寸。孔最大到刀径的两倍；更大的孔用型腔切（`shape: circle`）。

铰孔

用和孔的配合同级的铰刀（`reamer 6 H7`），按它的进给进去，再按同样的进给退出，不快速退出孔。同一个作业里前面要先钻这个孔，孔径小 0.05–0.6 mm；或者标 `predrilled`。铰刀的线速度取这种材料立铣刀的四分之一，每转进给取钻头的 2.5 倍。

螺纹铣削，含锥管螺纹

螺纹铣刀先下到底孔里，再从下往上走螺旋，每圈一个螺距：右旋逆时针（顺铣），左旋顺时针。螺纹按名字给（`thread`：`M8`、`M8x1`、`1/4-20 UNC`、`G1/4`），或者给 `d` 和 `pitch`。锥管螺纹（`1/4 NPT`、`Rc1/4`）沿它的锥面切，每圈分成 48 段圆弧。

孔里的环槽

`groove` 用 T 形槽铣刀（细颈上的一片圆盘）在孔壁上铣一道环槽（卡簧槽、O 形圈槽，或消光槽）。刀沿孔的中心下到槽的位置，出到离孔壁 0.1 mm，再每次往壁里进 `step` 绕一整圈（顺铣），最后回到中心、抬起。作业给槽径 `d`、孔径 `bore`、槽的下沿 `depth` 和槽高 `width`（不给就是圆盘厚；更高的槽从下往上分几趟铣）。下面几种情况会拒绝：主轴上不是 T 形槽铣刀；圆盘比孔小不了 0.2 mm、下不去；槽比圆盘伸出颈部的部分还深；槽径不大于孔径。

倒角、刻字和 V 形雕刻

- 倒角：V 刀沿外形线的边走，深度按要的 `width` 算。宽度受刀锥的限制。
- 刻字：刀沿线条，或单线字体、轮廓字体的文字走，深度是同一个 `depth`。
- V 形雕刻：V 刀沿形状的中轴线走。每一点的深度给出那里的宽度，所以形状的角是尖的。`max_depth` 让宽的地方在这个深度上铣平。

要在曲面上刻字，给 `surface`：`geometry_import` 导入的网格名。每一笔每 0.2 mm 取一点，刀放到网格上，在它碰到的高度下切 `depth`；在斜面上 V 刀先碰到上坡一侧，所以任何地方都不会比 `depth` 切得更深。笔画走出网格就拒绝。

珍珠纹（钟表上一个压一个的圆斑）用 `perlage`：和圆斑一样宽的立铣刀在一个轮廓里的网格点上逐个下到 `depth`（最多 0.2 mm），停 `dwell` 秒（默认 0.2），抬到面上 0.3 mm 再去下一个。每行错开半个 `pitch`（默认刀径的 0.65），每个斑压住上一个的一部分。`margin` 让斑心离轮廓至少这么远。

三维粗加工和精加工

网格先分层粗加工（`rough3d`），每层用自适应清。再精加工（`finish3d`），策略有：

策略	走法
`parallel`	朝一个方向横过零件来回走
`waterline`	等高的环，适合陡墙
`steep_shallow`（默认）	陡的地方走等高，缓的地方走平行
`pencil`	最后沿内角走

`rest_from` 只精加工前面那把刀留下的地方。

残料粗加工：给 `rest_from`（之前粗加工的那把更大的平底或牛鼻刀，写规格或刀表里的号），`rough3d` 只切那把刀留下的料，比如它进不去的筋间空隙；`rest_allowance` 是它留的余量（不给就用这一刀自己的）。

孪生定的切削参数

Studio 先按刀的种类取这种材料的数据：线速度范围和每齿进给。你在 `cutting` 里给的值（`rpm`、`feed`、`ap`、`ae`、`finish_ap`）照你给的用。其余的由孪生的模型定。

稳定性和颤振

孪生的颤振模型算出刀具装在刀柄里时的稳定性叶瓣图：每个主轴转速下保持稳定的切深。叶瓣的位置只和刀具、机床的模型一样准，所以 Studio 不瞄准叶瓣的峰：每个转速取它 ±10 % 范围内最浅的稳定切深，在材料给的转速的 0.7–1.8 倍之内选转速，切深最多取这个深度的 80 %。

功率、切削力和刀具让刀

- 功率：切削最多用掉主轴在这个转速下功率的大约 80 %。
- 切削力：低于夹具侧向夹得住的力的一半。
- 让刀：精加工的刀量放轻，让孪生估计的刀具让刀保持很小。
- 转角和圆弧：圆弧上的进给受机床加速度的限制，免得控制器降到比计划的进给还慢。精加工走到型腔内角时减速，因为那里刀的吃刀量变大。

闭环：在孪生里试跑程序

开了校验时，Studio 在一台新的孪生上跑这个程序。跑出颤振、负载太大或刀具让刀太多，Studio 就改程序再试：

- 把层切浅一些；
- 换一个转速（原转速的 0.7–1.8 倍之内），叶瓣的位置稍有偏差也照样稳定；
- 给配合的墙加光刀。

试过的每个版本都存下来。长的程序在调用之后接着试，报告稍后才有（`run_status`）。回答里会说改了什么、为什么改。

多次装夹和排料

- 多次装夹：Studio 写出每次装夹的程序，再在孪生里按顺序检查，每次装夹都在前面几次留下的工件上跑。那个工件和毛坯不一样时，就按它重写这次装夹的程序。
- 排料：零件按外形排在板料上，彼此留出边距。每个零件带着自己的特征，跟着它一起转、一起移。

局限

- 三轴。没有第四轴，刀不倾斜。
- 不读 STEP 文件；请在 CAD 软件里导出 STL 或 3MF。

- 切削参数和孪生的检查，都是机床、刀具和材料的物理模型的估计。刀具或材料和它的模型不一样，切出来也会不一样。
- 档位：自适应开粗、`parallel` 以外的三维精加工策略、残料加工和多次装夹，创客版起有；排料和程序优化在专业版里。

一个作业能有多大

超过下面这些上限，Studio 不写程序，并说明原因和该改什么。

什么	上限
一个作业的装夹次数	8
排料	16 种零件，每种 200 件
网格（STL、3MF、OBJ、glTF）	500 万个三角形；文件最大 1 GB
SVG、DXF 或文字文件	64 MB；DXF 图最多 200 万段
自适应开粗	走刀网格约 150 万格。和 MC-1 行程一样大的型腔，用 3 mm 立铣刀能开；用 1 mm 的，请改用环切或往复，或者加大切宽。

一个作业最多 32000 个特征；同类的孔写成一个特征，列出各点的位置。工序列表、理由和闭环的补救只覆盖前 64 个特征；后面的特征颤振时只报告，不补救。装夹单只列前 64 道工序，并写明这一点。

程序检查和装夹单最多读 200 万个移动段。圆弧按它的短弦段计：半径 3 mm 的整圆约 50 段，50 mm 的约 220 段。超过时，它们写明读到哪里为止。装夹单仍给出整个程序的时间，但那一行以后各工序的时间和各把刀的高度不给。孪生里的试运行会查整个程序。

Studio 写一个程序要多久（台式电脑的两个核上量的；孪生里的试运行另算）：

作业	时间	内存
180 × 130 mm 板上 20 个带岛的型腔	不到 1 s	18 MB
和行程一样大的型腔，自适应，6 mm 立铣刀	12 s	47 MB
同上，3 mm 立铣刀	37 s	70 MB
刻 1000 个字	12 s	25 MB
在板料上排 200 个垫圈	5 s	30 MB
三维开粗，和行程一样大的浮雕（50 万个三角形），6 mm 立铣刀	8 分钟	240 MB
同一块浮雕，网格 300 万个三角形	20 分钟	1 GB
它的三维平行精修，2 mm 球刀，残高 0.01 mm（50 万个三角形）	4 分钟	270 MB
同上，300 万个三角形	18 分钟	1 GB

做三维时：

- 网格只要细到零件够用就行。三角形越多，精修越慢，时间大致成正比。
- 起伏平缓的曲面，精修用 `parallel` 或 `steep_shallow`，不要用 `waterline`。等高在这种面上会走出很多小圈：上面那块浮雕用它时，空走比切削还多，程序有 280 万行。

配合、跳动和检验方法

Studio 怎么把尺寸切到配合，怎么考虑刀具的跳动，怎么用测头量零件。这里孪生给出的数都是它的物理模型的估计；Studio 不对机床的定位精度作任何说法。

配合（ISO 286）

尺寸可以带配合：孔的直径（`"d": "Ø12 H7"`）、轴的直径（外轮廓上写 `"fit": "g6"`）、槽宽，或者矩形的 X、Y 尺寸。Studio 按尺寸从 ISO 286-2 的表里取配合的上下偏差，也可以按你写的上下偏差（`"tol": "+0.015/0"`，单位毫米）。

切到公差带中间

墙先留余量粗加工，再单独走一刀精加工，切到公差带的中间。IT6 及更细的配合加一刀光刀：精加工那一圈在尺寸上再走一遍，切掉刀具受力弯曲留下的那一点。孪生估计的让刀需要时，闭环也会加光刀。

回答里会写出每个配合的公差带、切到的直径和原因。

常用配合

名字	配对	用途
`sliding`	H7/g6	能转动、能滑动，间隙小
`locating`	H7/h6	间隙配合里最紧的：手能装拆
`transition`	H7/k6	可能有一点间隙或一点过盈：定位、轻压
`press`	H7/p6	过盈：压装

配对写两边：孔取前一个，轴取后一个。

刀具的跳动

有跳动的刀切出的孔比刀本身大。孪生把刀装在刀柄里的跳动建成一个包络。有配合的尺寸，Studio 把精加工刀路按这个包络挪：孔往里、轴往外，让孪生估计的尺寸仍在公差带中间。铰孔不挪：孔就是铰刀的尺寸。

这个包络是孪生按它的刀具和刀柄模型给的估计。第一件要量一下。

用测头量

一次触碰和它的补偿

每次触碰是一串探测移动（G38）：

1. 保护移动到表面附近，途中测头一触发就停；
2. 快速探到表面；
3. 退回到测头松开，再多退一点；
4. 慢速探到表面：只用这一次的触发点；
5. 退回出发的地方。

触发点是球心。Studio 沿触碰方向加上球的有效半径。有效半径和测头的三瓣误差，来自它在已知直径的环规上的标定。

孔、凸台、墙和底

- 孔和凸台：在一个深度上沿墙碰四点或八点。Studio 用最小二乘把球心拟合成圆，再加上有效半径。
- 墙：垂直于墙碰，沿墙取几个点。
- 底面和顶面：竖直向下碰。
- 位置用程序坐标，从工件零点算。

一般公差（ISO 2768-1）

没有自己公差的尺寸，按要求的级别（f、m、c、v）从 ISO 2768-1 的表里，按它的尺寸段取一般公差：线性尺寸，和倒角的高度、角度。

孪生的网格对某个公差来说太粗时，这一项的孪生估计会注明太粗、不判；量到的值照样判。

形位检查（ISO 2768-2，第一步）

在 ISO 2768-1 的级别后面加 H、K 或 L（比如“mK”），就加上形位检查，按 ISO 2768-2 的表判：

检查	怎么碰	表
平面度	底面按 3 × 3 的格子；各点离它们拟合平面的散布	1
直线度	长墙沿它碰 3 到 8 点；各点离它们拟合直线的散布	1
垂直度	墙在第二个深度再碰；按墙高折算的倾斜，相对机床的 Z	2

检查	怎么碰	表
圆度	孔或凸台碰 8 点；各半径的散布	4（直径的公差）

墙上的两次触碰要隔得够远才分得清。墙太短（比如上沿有倒角），就列出原因，不给值。

只标定了有效半径的测头，有它自己的三瓣误差，会出现在圆度里：报告写出孪生的测头模型估计它有多大。

倒角和 V 形槽

- 倒角：沿斜面从上往下碰两次，分别在离外形线 0.3 和 0.7 个宽度的地方。两个球心连成的线和斜面平行，由它得出倒角的宽、深和角度。
- V 形槽（刻字或 V 形雕刻）：每个槽面各碰两次。两条线给出槽在尖底的深度、顶部的宽度、夹角，以及离它所沿的那条线的偏移。
- 螺纹不量。它的牙顶就是直的底孔，孪生的测头只碰得到格子的顶面；回答里会写明。

孪生的估计是什么意思

孪生用同一个程序切它模拟的工件，再用同样的测头模型做同样的触碰。它的估计就是这样算出来的结果，反映程序本身对每个尺寸的影响，到孪生建模的程度为止：刀具的让刀、跳动、切削力。

这个估计不是机床的测量不确定度。报告对机床测量或定位的精细程度不作任何说法。

局限

- 测头只量它碰得到的。比球还窄的特征、球不能垂直碰到的表面，不量；回答里写明原因。
- 孪生的工件是网格。它的估计只能细到这个网格；网格对公差来说太粗的地方，孪生的估计不判。
- 形位检查是 ISO 2768-2 的第一步：平面度、直线度、垂直度和圆度。对称度和平行度还不查。

机床档案

草稿（第 14 轮）：正文在写，各节还会补。

机床档案是什么

机床档案是孪生据以建立机床的文件：一种机型一个，JSON 格式。里面有机床的几何和各轴、驱动器和滚珠丝杠、主轴和变频器、传感器和接线、装夹、每种材料的切削数据、对刀仪和测头、声音，以及控制器编进了什么。控制器的设置另有一个设置文件，随档案一起。

档案记着它的值从哪里来（出处）：设计、零件手册、公开资料、假设，或者实测。

从哪里来

MC-1 的档案从机床的参数化设计生成：出图纸、出物料清单的同一组参数也出这份档案，所以孪生和机床的设计不会走样。档案里留着生成它的那些设计文件的校验和。一份 schema 检查它的格式；参考里的机床档案格式一章列出每一项。

你这台机床的值

每台 MC-1 计划在调试时实测，并按实测设定它的孪生（流程和工具已写好，只在孪生上测过）：

- 限位开关：在哪里动作、回差、重复性；
- 每根轴按 ISO 230-2：反向间隙、导程误差表、每毫米步数；
- 铣试槽得到的切削系数；
- 主轴自由停车得到的惯量和摩擦；
- 测头在环规上标定。

实测的值连同日期、用哪个工具、怎么测的一起记下；档案用它们重新生成，每一项标成"实测"，设计值留在它的历史里；它们改变的控制器设置发到机床上。

几台机床

Studio 可以放同一机型的几台机床。每台在两次会话之间各留各的世界（它停在哪里、它的工件、它的刀具、它控制器的存储）和自己的窗口布局；在实测某一台之前，同一机型的几台用同一份档案。

参考表

档案的每一项，带类型、单位和 MC-1 的值：机床档案格式。

预览：这一页的安全提示还没有经过母语审校。

MC-1 硬件

草稿（第 14 轮）：正文在写，各节还会补。

这一章讲设计出来的 MC-1：里面的数都是设计值。每台造好的机床计划在调试时实测，并按实测设定它的孪生（见机床档案）。

机床

MC-1 是加工铝、黄铜和工程塑料的桌面三轴铣床。龙门固定；工作台沿 Y 移动，龙门上的滑座沿 X 移动，主轴沿 Z 移动。

行程 X / Y / Z	200 / 150 / 80 mm
工作台	300 × 220 × 35 mm，M6 螺孔网格 25 mm
外形 宽 × 深 × 高	约 508 × 490 × 564 mm，含电机、拖链、调平脚，不含电控箱
质量	约 57 kg，不含电控箱
机身	6061-T6 铝板，螺栓连接
直线导轨	HIWIN HGR15 导轨 6 根，HGW15CC 法兰滑块 12 个
滚珠丝杠	SFU1204（12 mm，导程 4 mm），C7，BK10 / BF10 支撑
电机	57 闭环步进，1.2–1.3 N·m，3 台，梅花联轴器直连
主轴	800 W 水冷，Ø65 × 195 mm，ER11 夹头
电控箱	独立的箱子，放在机床右边

轴和驱动

每根轴是一台闭环步进电机经梅花联轴器直接带动丝杠。驱动器的故障走一条线：三台驱动器的报警输出串联接到控制器的电机故障输入，任何一台驱动器出故障或掉电，机床都会报警停下。

设计的最快移动速度：X、Y 3000 mm/min，Z 1500 mm/min。

限位开关和回零

五只 M8 电感式接近开关标出行程两端：X 两端、Y 两端各一只，各接一路输入；Z 滑板上一只，管 Z 的上下两端。开关是常闭的：断线、插头松了都读作"已触发"，所以控制器会报警、拒绝回零，而不是一直走到硬限位。

回零先 Z、后 X 和 Y，各自向正端（X+、Y+、Z+）。开机后必须先回零。

Z 轴和重力

Z 滑板带着主轴，约 6.35 kg。驱动器有电时它们把 Z 保持住；没电时（急停、驱动器故障、断电之后），Z 连着主轴可能慢慢往下滑。

> ⚠ **警告**
>
> 急停或驱动器故障之后，先重新回零再接着干。驱动器断电时不要把手伸到主轴下面。

调试时要做一次驱动器断电的试验：Z 会下滑，就换带抱闸的 Z 电机，或者加配重。

主轴

主轴是 800 W 水冷电主轴，Ø65 × 195 mm，ER11 夹头，2 极，最高 24 000 rpm。由欢洋 HY01D523B 变频器（单相 220 V 输入）驱动；控制器经 RS-485（接隔离模块）设它的转速、读它的状态。一台水泵给主轴冷却。

- 变频器的故障继电器也串在电机故障那条线上：变频器跳闸时，控制器报警、停下程序，不会让停转的刀接着进给切进工件。
- 急停时变频器跟驱动器一起断电，主轴靠惯性自己停下（没有制动）。

控制器和接线

控制器是插在 T41U5XBB 底板上的 Teensy 4.1，输入是光耦隔离的。它跑同匠自研内核（Grbl 1.1 串口协议兼容），经 USB 连电脑。

输入	接的是什么
限位	X−、X+、Y−、Y+ 和 Z，五路
探针	对刀仪和测头
急停	急停按钮的第二组触点
电机故障	三台驱动器的报警输出和变频器的故障继电器，串联
循环启动、进给保持	电控箱上的按钮 SB2、SB3

供电

市电先过断路器。24 V 电源一直有电：给控制器、传感器和控制回路供电，所以急停时控制器和电脑的连接不断，还能报出急停。接触器 KM1 切换驱动器的 48 V 电源、变频器和水泵。

电控箱上的按钮

电控箱门上四个 Ø22 按钮，从门外看：

	左	右
上排	SB1 上电（绿）	SB0 急停（红）
下排	SB2 循环启动（绿）	SB3 进给保持（黄）

- **SB1 上电**：让 KM1 吸合，之后它自己保持。急停或断电之后，机床不会自己重新上电：松开急停，再按一次 SB1。
- **SB3 进给保持**：和发送软件的进给保持（`!`）一样：各轴受控停下，主轴照转。它在急停正下方，伸手去拍急停没拍准，碰到的是进给保持，不是循环启动。
- **SB2 循环启动**：接着走，和发送软件的循环启动（`~`）一样。

> **注意**
>
> 进给保持不是安全功能。为了安全要停机，按急停。

对刀仪和测头

对刀仪装在工作台左后角，两颗 M6 螺钉锁在网格孔里，占去工作区约 X 0–57、Y 130–150 mm 的一角。对刀仪和测头共用控制器的探针输入。

Z 在最高处时，伸出夹头 15.5 mm 的刀，刀尖离台面 80 mm，主轴端面 130 mm；刀伸出得更长，刀尖就低多少。装在 ER 夹头里的测头，从螺母到针尖一般 60–90 mm：Z 在最高处时，这样的测头只越得过很低的装夹，70 mm 的测头约为台面以上 20 mm。

Studio 会替你查：装夹时、换刀时、改刀具表时，它拿装夹的最高点和每把刀在 Z 最高处时的刀尖比，告诉你哪把刀越不过去、差多少。

装夹

工作台上 25 mm 网格的 M6 孔用来装压板、底板或虎钳。孪生里有：75 mm 自定心虎钳加 10 mm 铝底板、垫铁、压板、双面胶、V 形块。

- 虎钳装在底板上时，钳口顶面离台面约 66 mm，所以夹在虎钳里的零件高约 71 mm 或更高。
- 要用测头测的零件，用低装夹：压板压在底板上（毛坯顶面离台面 15–30 mm），或者矮虎钳。夹在 75 mm 虎钳里的零件上方放不下测头。
- 拿掉 10 mm 底板，刀能多出 10 mm；对测头还不够。

安全

- **急停（SB0）**：断开接触器的线圈回路，切掉驱动器、变频器和水泵的电，并经第二组触点告诉控制器。各轴和主轴断电；主轴靠惯性自己停下；Z 可能下滑（见 Z 轴和重力）。松开急停什么都不会重新启动：按 SB1，等驱动器起来，复位控制器，再回零。
- **故障**：驱动器故障、变频器跳闸、限位开关断线，都让机床报警停下。
- **进给保持**是为加工的受控停止，不是为安全。
- MC-1 没有防护罩。主轴转着、各轴可能动的时候离主轴远一点，戴护目镜；X、Z 轴的丝杠上加风琴罩或刷条挡屑。
- 机床的安全功能都不用 AI；Studio 里的 AI 助手不是安全功能。

调试时实测什么

上面的设计值计划在每台造好的机床上核对，并按实测设定它的孪生：

- 每只限位开关的触发点（电感开关被挡片从侧面扫过时，可能比图上晚约 2 mm 触发），然后定回零后退距离和软限位；
- 驱动器的报警输出：拔掉一根，或者断开 48 V，都必须报警；
- 驱动器断电时 Z 的下滑试验；
- 变频器的参数，先在它的面板上空载试转，再改成经 RS-485 控制；
- 电控箱：急停和 SB1 照上面那样；逐只拔下限位开关和对刀仪的插头（每次都必须报警）；面板按钮。

孪生怎么接收实测的值：机床档案。

接口文档

这一页是从程序生成的接口参考，只有英文版。

MCP tool reference

Twinwright Studio offers its twin to AI clients (Claude, ChatGPT, Cursor and others) over the Model Context Protocol. A client starts the program in the background as `TwinwrightStudio.exe --mcp` (the command is shown under AI > Connect AI clients..., The command and configuration) and calls these 47 tools. Each acts on one machine: the one the user sees in the window (`"machine": "window"`, while the window's AI switch is on) or an invisible practice machine (`"sandbox"`). Parameters marked * are required.

Contents

- Machines: `machine_status`, `machine_control`, `machine_library`, `machine_mdi`
- Set-ups: `setup_workpiece`, `setup_tool`, `setup_work_zero`, `inspect_part`, `inspect_report`, `setup_sheet`, `probe_cycle`, `setup_part`, `setup_copy`, `setup_example`
- Programs: `program_check`, `program_simulate`, `program_run`, `program_load`, `program_debug`, `program_restart`, `program_optimize`, `geometry_import`, `geometry_nest`, `program_generate`, `program_edit`, `program_versions`, `job_get`, `job_edit`
- Runs and results: `run_status`, `run_profile`, `result_compare`, `result_image`, `result_export`, `part_save`, `part_export`, `part_load`, `log_query`
- The licence: `licence_status`
- The part library: `part_list`, `part_get`, `part_update`, `part_delete`
- Reports to support: `report_draft`, `report_list`
- The user's window: `window_show`, `settings`
- The manual: `manual_search`

Machines

`machine_status` - Machine status

read-only · the same call twice does no more than once

The state of a machine: controller state and alarm (with its meaning), homed or not, machine and work positions, G54, the tool in the spindle, the spindle, the workpiece on the table and the program's progress - and the other machines, a line each. detail "full" adds the controller's modes, offsets and planner queue; detail "program" gives the loaded program's text in pages. Changes nothing. Tell the user which machine you looked at: only the window's is the one they see.

Parameters

Name	Type	Default	Description
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"
`detail`	string	`"brief"`	(one of `brief`, `full`, `program`)
`offset`	integer	`0`	detail program: the first line (0-based)
`limit`	integer	`400`	

Returns (structured content): `machine`, `notice`, `status`, `machines`, `program`.

`machine_control` - Operate a machine

changes the machine (destructive)

An operator's action on a machine, then its state. home: homes (resets and unlocks first if it is in an alarm; does nothing when already homed - force: true homes again). unlock: $X, when the position is still known. reset: Ctrl-X. hold / resume: feed hold / cycle start. stop: stops the running program. move: to x, y and/or z (mm) in the work (G54) or machine frame, rapid or at feed; moving X/Y it first goes up to the top of Z, where homing left it (safe_z). power_on / power_off: the main switch (and START). estop / estop_release. restart: a practice machine starts afresh (never the window's). The window's clock, as its Simulation > Speed menu sets it: speed (factor 0.5 .. 1000 times real time - from x100 lite: what only the views see is thinned, the results the same - or 0 for the fastest the computer goes); to_end (on: the program computed to its end, not shown frame by frame, started if only loaded; on false stops it there, as stop); pause_sim / resume_sim (the simulation's own pause: virtual time stands). A practice machine runs as fast as it can already. override: the feed (10 .. 200 %), spindle (10 .. 200 %) and rapid (25, 50, 100 %) overrides, as the window's knobs set them (the status says them: overrides). On the user's machine the user allows each change in the window; when the window's twin follows a real machine (shadow mode) it takes none of these, stops included - the real machine is stopped at its own controls or in its sender.

Parameters

Name	Type	Default	Description
`action` *	string		(one of `home`, `unlock`, `reset`, `hold`, `resume`, `stop`, `move`, `power_on`, `power_off`, `estop`, `estop_release`, `restart`, `speed`, `to_end`, `pause_sim`, `resume_sim`, `override`)
`feed`	number		mm/min (G1); omitted: rapid (G0)

Name	Type	Default	Description
`spindle`	number		override: the spindle speed, % (10 .. 200)
`rapid`	number		override: the rapids, % (one of `25`, `50`, `100`)
`factor`	number		speed: times real time (0.5 .. 1000; 0 the fastest)
`on`	boolean	`true`	to_end: start (true) or stop (false)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"
`x`	number		
`y`	number		
`z`	number		
`frame`	string	`"work"`	(one of `work`, `machine`)
`safe_z`	boolean	`true`	
`force`	boolean	`false`	
`drive`	string		restart: the drive model (L0 ideal motion, fastest; L1 closed-loop drives, the default; L2 flexible transmission with backlash; L3 electrical) (one of `L0`, `L1`, `L2`, `L3`)

Returns (structured content): `machine`, `notice`, `status`, `action`, `did`, `next`.

`machine_library` - The window's machine library

changes the machine

The user's machines in the Studio window's library (several machines of a model, each with its own world - where it stands, its workpiece, tools, controller - and its own interface): list them (which one the window runs); new (name: from the factory state), copy (id, name: its world copied - it starts where that one stands), rename (id, name) - written at once, the window's library page follows; switch (id): the window saves its machine and starts again with that one, with the user's yes in the window (without a window open: the machine it starts with next).

Parameters

Name	Type	Default	Description
`action`	string	`"list"`	(one of `list`, `new`, `copy`, `rename`, `switch`)
`id`	string		copy, rename, switch: the machine (list gives the ids)
`name`	string		new, copy, rename: what the user will call it

Returns (structured content): `machine`, `notice`, `action`, `entry`, `machines`.

`machine_mdi` - Send MDI lines

changes the machine (destructive)

Send G-code or $ lines to a machine as an operator types them: one at a time, each with the controller's answer - ok, or the error and what it means. After a refused line the rest are not sent (the controller would refuse them too) and the result says so. Waits until the machine is idle again and gives its state. For homing, unlocking, stopping or moving, machine_control is simpler.

Parameters

Name	Type	Default	Description
`lines` *	array of string		
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"
`timeout_s`	number	`120`	simulated seconds a line may take

Returns (structured content): `machine`, `notice`, `status`, `lines`.

Set-ups

`setup_workpiece` - Place a blank

changes the machine (destructive)

Put a blank on a machine's table (replacing the one there), take it off (op "remove") or turn it over where it stands (op "flip": the other side up, what was machined stays - for a part's second side; refused while a program runs). Table coordinates in mm: X 0..200 left to right, Y 0..150 front to back, centre [100, 75]. Homes first if needed. The twin refuses setups that do not fit and says why. A part cut through: fixture tape (a thin sheet) or clamps on a sacrificial plate (sacrificial) - the blank that much higher, the cuts through it go into the plate, not the table. Clamps hold the blank's two ends unless sides says otherwise. The window's setup file: op "save" writes the setup on the machine to path (.json, not over a file unless overwrite), op "open" sets it up from one.

Parameters

Name	Type	Default	Description
`op`	string	`"place"`	(one of `place`, `remove`, `flip`, `save`, `open`)
`path`	string		op save / open: the setup file (.json, one line - the window's Setup > Save setup... / Open setup...)
`overwrite`	boolean	`false`	
`turn`	string	`"u"`	op flip: u about the blank's u axis (its length as placed, X at angle 0): front and back change places; v about its v axis: left and right change places (the blank also turns 180 degrees about Z) (one of `u`, `v`)
`shape`	string	`"block"`	(one of `block`, `cylinder`, `tube`, `hex`, `rod`, `stl`)
`size`	array of number		block: [x, y, z] mm; cylinder/rod: [diameter, length]; tube: [outer, inner, length]; hex: [across flats, length]
`material`	string	`"6061-T6"`	(one of `6061-T6`, `7075-T6`, `ADC12`, `HPb59-1`, `H59`, `T2`, `Q235`, `45`, `304`, `HT200`, `POM`, `PMMA`, `PA6`, `HDPE`, `MDF`, `hardwood`, `wax`)
`fixture`	string	`"vise"`	(one of `vise`, `clamps`, `tape`, `fixed`, `vblock`, `none`)
`sides`	string	`"u"`	fixture clamps: the edges the strap clamps hold - u at the blank's two ends along its u axis (its length as placed, X at angle 0), v on its two sides along it, all on all four (one of `u`, `v`, `all`)
`vise`	string	`"precision"`	fixture vise: which - precision (75 mm jaws on a 10 mm subplate, their tops 66 mm over the table) or low (the low-profile vise bolted to the table's grid, jaw tops 20 mm over it: more of the Z travel left for a long tool or a probe, a little less hold) (one of `precision`, `low`)
`protrude`	number		fixture vise: how far the blank's top stands over the jaws, mm (5 if left out; the twin puts it on the jaws' step or on parallels for that, 2 mm gripped at least) - less for a thin plate, e.g. 3 for 6 mm in the low vise
`at`	array of number	`[100, 75]`	blank centre on the table [x, y] mm
`angle`	number	`0`	rotation about Z, degrees
`res`	number		simulation cell size in mm (0.01..0.2; smaller is finer and slower)
`file`	string		shape stl: the blank's STL on this computer
`keep_zero`	string		set G54 on the new blank at once, at this corner (as setup_work_zero does); left out, G54 stays where it was (one of `front-left-top`, `center-top`)
`sacrificial`	object		fixture tape or clamps: the blank on a sacrificial plate, so that what is cut through goes into the plate, not the table. The blank sits higher by the plate (and the tape); the tool may cut into the plate (the summary's setup.sacrificial_cut_mm says how deep), the collet and the spindle may not; the tape holds as on the table, the clamps hold the blank and the plate down together (their tapped holes clear of the plate)
`sacrificial.material`	string	`"MDF"`	a material of the list above (for the drawing and the set-up sheet)
`sacrificial.thickness`	number	`10`	mm, 3 to 25
`sacrificial.size`	array of number		[length, width] mm along the blank's own axes; left out: the blank's footprint and 10 mm each side
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `placed`, `flipped`, `refused`, `reason`, `setup`.

`setup_tool` - Tool and tool table

changes the machine · the same call twice does no more than once

The tool in a machine's spindle (spec) and/or its tool table for programs that change tools with M6 (tools: {"1": spec, ...}; the change and the length measurement then happen by themselves). new: true - a new tool of that spec: its wear book (an estimate, machine_status's tool_wear) starts again. Tool spec: kind and diameter in mm, then options. Kinds: 'flat D', 'ball D', 'bull D RC' (corner radius), 'v D ANGLE' (V-bit), 'drill D ANGLE', 'tslot D THICKNESS dNECK', 'dovetail D ANGLE dNECK', 'thread D PITCH', 'probe D' (a touch probe with a ball of D; ' typical' a kinematic one: it triggers its pretravel past the touch, scattered with a seed; ' ideal' at the touch, the default). Options: 'nN' flutes (default 2), 'LMM' stick-out from the collet (the length reference tool sticks out 15.5 mm), ' sD lR' a shank of diameter D and the reach R below it, ' tirU' the cutter's runout in the collet, TIR in um (a precision collet's class, or as measured; default: the machine's). Examples: 'flat 6 n3 L20', 'ball 3 n2 L16', 'v 6 60 L18', 'flat 1.5 n2 s4 l5 L14', 'flat 0.2 n2 s4 l1 L15 tir3', 'probe 2 L30 typical'.

Parameters

Name	Type	Default	Description
`spec`	string		e.g. 'flat 6 n3 L20'
`new`	boolean	`false`	a new tool: its wear book from 0
`tools`	object of string		the tool table for M6: {"1": "flat 6 n3 L20", ...}
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `tool`, `tool_table`.

`setup_work_zero` - Set G54 on the blank

changes the machine · the same call twice does no more than once

Set a machine's G54 work zero on its blank - the front-left top corner (default: the corner at the machine's front left, X and Y least, when the blank lies square - turned over "v" too) or the top centre, with the tool now in the spindle as the length reference; an optional offset [x, y, z] mm. Homes first if needed.

Parameters

Name	Type	Default	Description
`corner`	string	`"front-left-top"`	(one of `front-left-top`, `center-top`)
`offset`	array of number	`[0, 0, 0]`	
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `g54_mpos`.

`inspect_part` - Inspect the part with the probe

changes the machine

Check the finished part on the machine with the touch probe in the spindle against its design: from the features of a program program_generate made (program; or features as it takes them) the checks - a pocket's sizes, centre and depth, a hole's diameter and position, a slot's width and depth, a faced step, a rectangle's or a circle's contour - each measured by a probe cycle (as probe_cycle runs them: results from the trigger points) and judged against the design (the tolerance given, else ISO 2768-1 - class f unless tol.class says) and against the twin's prediction (the same touches on the workpiece the twin cut: the twin's estimate, not a measuring uncertainty). Depths from a touch on the top where no feature reaches. A chamfer: two touches down onto its face - its angle, its width from the outline and its depth there (its V-bit: the feature's tool, else the job's; the material's side from the job's pocket or contour with that outline). V-carving, engraving, threads and 3D surfaces are said not measured. ISO 2768-2's class after part 1's as a drawing writes it (tol.class "mK") adds form checks and touches: a floor's flatness (a grid of 9 instead of 1), the walls' perpendicularity to Z (4 more at a second depth), a long wall's straightness (5 along it, 20 mm and longer), a circle's circularity (8 round instead of 4) and its axis' perpendicularity (4 more) - the answer's form_cost and the window's question say how many and how long before; only: [..] does part of them; one letter adds none. Least-squares estimates from a few touches, not ISO 1101's minimum zone nor a CMM's form measurement (summary.form_method). The work offset must be the one the part was cut with. Each inspection is kept as a record (written once, its SHA-256 in the audit log): inspect_report makes its CSV and PDF. plan: true only says what it would do. On the window's machine the touches take their machine time at the simulation speed the user set; longer than one call may wait, the call says how to measure it in calls that fit (only: [..] each) and the speed that fits one; continue: "last" makes each batch after the first go on from the one before - one record, one report. Professional edition.

Parameters

Name	Type	Default	Description
`program`	string		a program program_generate made: the features of the job kept with it
`version`	integer		that job's version (0: the last)
`features`	array of object		instead of program: the features as program_generate takes them
`tool`	string		with features: the tool they were cut with (a drilling's holes are its size)
`only`	array of integer		only these features (1, 2 ... in the job)
`continue`	string		a batch: the inspection it goes on from ("last", or its record) - the same program, set-up and work offset; the record and its report then hold all the batches (a feature measured again: this batch's)
`tol`	object		+/- mm (an angle: degrees) instead of ISO 2768-1
`tol.size`	number		

Name	Type	Default	Description
`tol.depth`	number		
`tol.position`	number		
`tol.angle`	number		a chamfer's, degrees
`tol.class`	string	`"f"`	ISO 2768-1's class where no tolerance is given (a drawing's ISO 2768-mK: m) - with ISO 2768-2's after it (mK: H, K or L) its form checks too, more touches (one of `f`, `m`, `c`, `v`, `fH`, `fK`, `fL`, `mH`, `mK`, `mL`, `cH`, `cK`, `cL`, `vH`, `vK`, `vL`)
`probe`	object		the probe's compensation instead of this machine's calibration: r_eff, lobing, phase_deg, pretravel_z
`plan`	boolean	`false`	only plan it, nothing moves and nothing is recorded: each feature, what it measures there or why not, the touches and about how long, the form checks' share, the classes, the program zero it would measure in (zero_mpos), each touch's point and way (points) - with the spindle's probe, else the tool table's first, else probe 2 L30 (planned_with)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `plan`, `features`, `summary`, `reference`, `cycles`, `twin_cell_mm`, `inspection`, `record`, `sha256`, `record_error`, `error`, `form_cost`.

`inspect_report` - Inspection report

changes the machine · the same call twice does no more than once

The report of an inspection inspect_part recorded, written by the program itself: a CSV (UTF-8, for a spreadsheet) and a PDF - the job, setup, tools, the probe and its ring-gauge calibration, each feature's nominal, tolerance and its basis (ISO 2768-1's class and table, or given), the twin's estimate, the measured value, the deviation and the verdict, why a feature was not measured, and the twin's workpiece from above. Into the reports folder (Documents/Twinwright Studio/Reports); their SHA-256 into the audit log. The same record gives the same bytes again. The twin's estimate is the twin's own calculation for the same touches on its simulated workpiece, not the machine's measuring uncertainty; the report says nothing about how precisely the machine measures or positions. The PDF is in English where the language in use needs letters its fonts lack (the CSV is in it). verify: a file checked instead - its SHA-256 searched for in the audit log and the log's chain checked. It moves nothing. Professional edition.

Parameters

Name	Type	Default	Description
`inspection`	string	`"last"`	the record's name as inspect_part gave it (last: the one recorded last)
`format`	string	`"both"`	(one of `csv`, `pdf`, `both`)
`unit`	string		lengths (default: the units in use) (one of `mm`, `in`)
`paper`	string		the PDF's (default: the region's) (one of `a4`, `letter`)
`verify`	string		a file's path: check it against the audit log instead of writing a report

Returns (structured content): `machine`, `notice`, `inspection`, `record`, `record_sha256`, `csv`, `pdf`, `found`, `chain_whole`, `sha256`.

`setup_sheet` - Setup sheet

changes the machine · the same call twice does no more than once

The setup sheet of a job program_generate made, for the operator at the machine: one HTML page to print (A4 or Letter: the region's) in the language in use - each program's name, lines and SHA-256 (to match it with the one loaded), the blank and its fixture as the twin sets it up (the vise's parallels, the grip), the tools, each setup's way there (turned over, held another way), its work zero and the probe panel's cycles that find it, the program's travel, its Z clearance (the probe and each tool over the set-up and to the program's highest and lowest points - the twin's Z budget: enough, or how much short), the operations (strategy, tool, speed, feed, depth, the twin's estimate of each one's time), a top view, the notes, a box to sign. The blank as the machine has it set up (the call's machine), else the job's. Into the reports folder (Documents/Twinwright Studio/Reports) as <job>-setup-sheet.html; its SHA-256 into the audit log. The times are the twin's estimates from the program, not measured. It moves nothing. Professional edition.

Parameters

Name	Type	Default	Description
`program` *	string		the name program_generate was given (its job is kept with it; a job in setups: each setup's program)
`version`	integer		the job's version (left out: the last)
`paper`	string		left out: the region's (one of `a4`, `letter`)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `file`, `sha256`, `bytes`, `paper`, `programs`, `z_clearance`.

`probe_cycle` - Touch-probe cycle

changes the machine

Measure with the touch probe in the spindle (setup_tool "probe 2 L30") and, if asked, set the work zero from what it finds: cycle edge (a face of the blank: side x-, x+, y-, y+), corner (front-left, front-right, back-left, back-right; inside: a pocket's corner), bore (its centre and diameter, 4 or 8 points), boss (the same from outside), surface (its Z) or calibrate (a ring gauge of known d: the probe's effective ball radius, its lobes and pretravel - this machine's later cycles here use them). Each touch: a protected move to its start (G38.3), a fast touch, the release (G38.4), a slow touch (F25 at most); the results come from the slow touches' trigger points ([PRB:]), not from where the axes stopped. Left out, at comes from the blank the twin knows. Refused: no probe in the spindle, its input already triggered, the spindle turning, a move outside the soft limits. set {x, y, z}: the active work offset moved so the feature gets these work coordinates (G10 L2); the result's set_parts give each axis found as its G10 L2 word (the feature at work 0, or at set's value). Uncalibrated, the results are off by the probe's pretravel. dry: the plan only - checks (probe, machine, soft limits, safe height, depth: ok, warn or bad), each touch's path (machine coordinates), the program and its time; nothing moves. calibrate keep false: its numbers and a cal_id, kept later with save (this session's measurement only).

Parameters

Name	Type	Default	Description
`cycle` *	string		(one of `edge`, `corner`, `bore`, `boss`, `surface`, `calibrate`)
`side`	string		edge: the face (x- the blank's least X, touched from outside moving +X; y- its front) (one of `x-`, `x+`, `y-`, `y+`)
`corner`	string	`"front-left"`	(one of `front-left`, `front-right`, `back-left`, `back-right`)
`inside`	boolean	`false`	corner: a pocket's corner, its walls touched from inside (at needed)
`at`	array of number		[x, y] in the active work coordinates: a point on the face (edge), the corner, the centre (bore, boss, calibrate), where to touch (surface) - left out: from the blank
`d`	number		bore, boss: the nominal diameter; calibrate: the ring gauge's known diameter, mm (left out: a tube's or cylinder blank's)
`depth`	number	`3`	how far below the top the ball's centre touches, mm
`top`	number		the work Z the depth is from (left out: the blank's top)
`points`	integer		edge 1, or 2 (span apart: the face's turn too), 3, 4 or 8 over span (its straightness too: about their least-squares line); corner 2, or 3 (two on its X face: the blank's turn and where its faces meet); bore, boss 4 (default) or 8 (its roundness too) (one of `1`, `2`, `3`, `4`, `8`)
`span`	number		mm between the first and the last touch on one face (left out: 0.6 of the face)
`grid`	array of integer		surface: [nx, ny] touches over size (25 at most): the plane through them - its tilt and flatness
`size`	array of number		surface grid: [w, h] mm about at (left out: 0.6 of the blank)
`inset`	number	`5`	corner: mm from the corner along each face
`clear`	number	`3`	mm between the probe and where the face should be when a touch starts
`search`	number	`5`	mm past where it should be a touch looks
`fast`	number	`300`	
`slow`	number	`20`	(at most 25)
`set`	object		move the active work offset so the feature found gets these work coordinates (the axes the cycle finds)
`set.x`	number		
`set.y`	number		
`set.z`	number		
`probe`	object		the probe's compensation instead of this machine's last calibration here: r_eff, lobing, phase_deg, pretravel_z (mm, degrees)
`dry`	boolean	`false`	the plan only: the checks, the path, the program and its time - nothing moves, nobody is asked
`keep`	boolean	`true`	calibrate: keep the calibration on the machine (false: its numbers and a cal_id only)
`save`	string		calibrate: keep the calibration this session measured with keep false (its cal_id) - nothing measured
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `cycle`, `touches`, `result`, `work`, `set`, `comp`, `archive`, `notes`, `program`, `sent`, `error`, `dry`, `checks`, `plan`, `seconds`, `set_parts`, `cal_id`.

`setup_part` - Set the design part

changes the machine · the same call twice does no more than once

The design part (an STL of the finished part, mm) to compare a machine's workpiece with. origin: machine mm of the part's origin; omitted: the program zero (as CAM writes it). An empty stl takes it away.

Parameters

Name	Type	Default	Description
`stl` *	string		
`origin`	array of number		
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `part`.

`setup_copy` - Copy a setup

changes the machine (destructive) · the same call twice does no more than once

Copy one machine's setup - blank and fixture, the tool in the spindle, the tool table, G54, the design part - onto an invisible practice machine, to try programs there at full speed (default: from the user's window to "sandbox"). The workpiece comes as it is there, cuts and all (stock: "fresh" for a new blank).

Parameters

Name	Type	Default	Description
`stock`	string	`"current"`	current: the machine's workpiece as it is, cuts and all (carry on with a part already cut); fresh: a new blank (a whole program from the start) (one of `current`, `fresh`)
`from`	string		the machine to copy (default: the user's window)
`to`	string	`"sandbox"`	a practice machine: sandbox or sandbox-<name>

Returns (structured content): `machine`, `notice`, `copied_from`, `workpiece`, `tool`.

`setup_example` - Example parts

changes the machine (destructive)

Without id: the example parts that come with the twin (id, title, material, blank, summary). With id: a machine set up as the example's operator would - blank, tool table, homing, G54 on the blank, the design part - with the example's program loaded, not started. To verify an example without setting anything up: program_simulate {example: id}.

Parameters

Name	Type	Default	Description
`id`	string		
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `examples`, `example`, `ready`.

Programs

`program_check` - Check a program

read-only · the same call twice does no more than once

Check a program without running it: the controller's own check mode ($C) - every line it refuses, with the error explained - the preview of its path against a machine's setup (estimated time, collisions, rapids into the stock, travel) and the static rules S01-S22 (start block, end, F and S missing or out of range, dense short lines, plunges, rapids below Z0, the water-cooled spindle without M8, arcs in inverse time, G33 without a spindle encoder, characters the controller takes for real-time commands, drilling cycles' holes a single-precision controller plunges once more, Z down after a tool change before X and Y, a work coordinate system turned about Z - the twin does not turn it yet, the Z surface map ($HM) on - the twin does not follow its offsets yet, ...) with a suggested line for each. Runs on a hidden copy of the machine's setup, its workpiece as it is (stock): the machine does not change.

Parameters

Name	Type	Default	Description
`gcode`	string		the program's text - or leave it out and give name (and version) of a program in the working folder (program_versions)
`name`	string	`"program.nc"`	
`version`	integer		with name, no gcode: that saved version's text (0: the last)

Name	Type	Default	Description
`file`	string		instead of gcode: the program's path on this computer (a .nc .ngc .gcode .tap ... file, up to 32 MB) - read and kept in the working folder as a version under its name
`stock`	string	`"current"`	current: the machine's workpiece as it is, cuts and all (carry on with a part already cut); fresh: a new blank (a whole program from the start) (one of `current`, `fresh`)
`example`	string		an example's setup (and program, when gcode is omitted) instead of the machine's: see setup_example
`setup`	object		an explicit setup instead of the machine's: {spec: the workpiece's setup as machine_status shows it (status.workpiece), tool, tools: {T: spec}, g54: [x,y,z] MPos, part: STL path}
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `ok`, `refused`, `preview`, `static_rules`.

`program_simulate` - Simulate a program

read-only · the same call twice does no more than once

Run a program to its end on a hidden copy of a machine's setup - the machine itself does not change; the copy starts from the machine's workpiece as it is, cuts and all (stock: "fresh" for a new blank) - as fast as the computer goes: moves hundreds of times faster than real time, cutting a few times faster (a heavy cut can be slower than real time). The report: completed or not, machine time, alarms, collisions, near misses, rapids into material, cuts with the spindle stopped, workpiece slip, chatter margin, force and power peaks, removed volume, deviation from the design part, the slowest lines - and a run id for run_profile, result_compare and result_image. Programs with M6 need tools. Longer than budget_s: the answer is the run id (run_status waits).

Parameters

Name	Type	Default	Description
`gcode`	string		the program's text - or leave it out and give name (and version) of a program in the working folder (program_versions)
`name`	string	`"program.nc"`	
`version`	integer		with name, no gcode: that saved version's text (0: the last)
`file`	string		instead of gcode: the program's path on this computer (a .nc .ngc .gcode .tap ... file, up to 32 MB) - read and kept in the working folder as a version under its name
`tools`	object of string		the tool table for M6: {"1": "flat 6 n3 L20", ...}
`stock`	string	`"current"`	current: the machine's workpiece as it is, cuts and all (carry on with a part already cut); fresh: a new blank (a whole program from the start) (one of `current`, `fresh`)
`example`	string		an example's setup (and program, when gcode is omitted) instead of the machine's: see setup_example
`setup`	object		an explicit setup instead of the machine's: {spec: the workpiece's setup as machine_status shows it (status.workpiece), tool, tools: {T: spec}, g54: [x,y,z] MPos, part: STL path}
`check_first`	boolean	`true`	
`timeout_s`	number	`3600`	simulated seconds allowed
`budget_s`	number	`40`	wall-clock seconds this call waits before it answers with a run id (at most 45: clients give up after 60 s; run_status waits for the rest)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `run`, `completed`, `simulated_time_s`, `issues`, `status`.

`program_run` - Run a program on the machine

changes the machine (destructive)

Run a program on the machine itself - the one loaded, or gcode (it replaces the loaded one): on the user's machine in real time, after the user allows it in the window; on a practice machine (sandbox) at full speed, cutting its workpiece. Checked first, the loaded program too (check_first): not run when the controller refuses a line or the check ends in an alarm - and on the user's machine not when the check's preview finds a collision, a rapid move into material, a cut with the spindle stopped or a move out of travel. Same report as program_simulate. To find out first what a program would do, use program_simulate.

Parameters

Name	Type	Default	Description
`gcode`	string		the program's text - or leave it out and give name (and version) of a program in the working folder (program_versions)
`name`	string	`"program.nc"`	

Name	Type	Default	Description
`version`	integer		with name, no gcode: that saved version's text (0: the last)
`file`	string		instead of gcode: the program's path on this computer (a .nc .ngc .gcode .tap ... file, up to 32 MB) - read and kept in the working folder as a version under its name
`tools`	object of string		the tool table for M6: {"1": "flat 6 n3 L20", ...}
`check_first`	boolean	`true`	
`fits_only`	boolean	`false`	nothing run, nothing moved: only the twin's grid made finer where the touch probe will measure the fits of the saved program name (its CAM job), as a run does before it starts - for a program started some other way (the window's own start); the answer's fit_regions says how many regions, where (program mm) and their cells, or why none (no job, no fits, several setups). Before the cut, never after: what is cut already stays as coarse as it was
`timeout_s`	number	`3600`	simulated seconds allowed
`budget_s`	number	`40`	wall-clock seconds this call waits before it answers with a run id (at most 45: clients give up after 60 s; run_status waits for the rest)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `run`, `completed`, `simulated_time_s`, `issues`, `status`.

`program_load` - Load a program

changes the machine · the same call twice does no more than once

Load a program into a machine without starting it. On the user's machine the window shows how it differs from the program loaded there and the user allows it; they may then start it themselves.

Parameters

Name	Type	Default	Description
`gcode`	string		the program's text - or leave it out and give name (and version) of a program in the working folder (program_versions)
`name`	string	`"program.nc"`	
`version`	integer		with name, no gcode: that saved version's text (0: the last)
`file`	string		instead of gcode: the program's path on this computer (a .nc .ngc .gcode .tap ... file, up to 32 MB) - read and kept in the working folder as a version under its name
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `lines`.

`program_debug` - Debug a program

read-only

Debug a program on a hidden copy of a machine's setup. run_to: runs gcode from its start until a file line (line) or a condition (when: force_n, load_pct, ferr_mm, margin, near_mm as limits; cut, rapid_cut, collision, alarm, state true; modal ["units","wcs","tool","spindle",...]) and stops virtual time there - mid-motion, exactly as the program has it, at the same instant every time. Then inspect (the line, modes, positions, speed, the cut, the planner's queue), step (until: time with ms, block, or pause), continue (to the next pause or the end), end.

Parameters

Name	Type	Default	Description
`action` *	string		(one of `run_to`, `inspect`, `step`, `continue`, `end`)
`gcode`	string		the program's text
`name`	string	`"program.nc"`	
`file`	string		instead of gcode: the program's path on this computer (a .nc .ngc .gcode .tap ... file, up to 32 MB) - read and kept in the working folder as a version under its name
`line`	integer		file line (1-based)
`when`	object		conditions, e.g. {"force_n": 60, "cut": true}
`tools`	object of string		the tool table for M6: {"1": "flat 6 n3 L20", ...}
`until`	string	`"time"`	(one of `time`, `block`, `pause`)

Name	Type	Default	Description
`ms`	number	`10`	
`stock`	string	`"current"`	current: the machine's workpiece as it is, cuts and all (carry on with a part already cut); fresh: a new blank (a whole program from the start) (one of `current`, `fresh`)
`example`	string		an example's setup (and program, when gcode is omitted) instead of the machine's: see setup_example
`setup`	object		an explicit setup instead of the machine's: {spec: the workpiece's setup as machine_status shows it (status.workpiece), tool, tools: {T: spec}, g54: [x,y,z] MPos, part: STL path}
`timeout_s`	number	`3600`	simulated seconds allowed
`budget_s`	number	`40`	wall-clock seconds this call waits before it answers with a run id (at most 45: clients give up after 60 s; run_status waits for the rest)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `reached`, `pause`, `where`.

`program_restart` - Restart from a line

changes the machine (destructive)

The program loaded on a machine from a file line on, as an operator restarts after a broken tool: a start block from the program's own state before that line (units, plane, work offset, tool, spindle speed and direction with its spin-up, coolant, over the start point at the safe height, down to it, feed, distance mode), then the rest. start: true loads and runs it (on the user's machine after the user allows it).

Parameters

Name	Type	Default	Description
`line` *	integer		
`start`	boolean	`false`	
`safe_z`	number		machine Z to go up to first (default 2 mm below home)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `line`, `started`, `program`.

`program_optimize` - Optimize a program

changes the machine · the same call twice does no more than once

Make a program faster without asking more of any tool than the program already does: it runs on a hidden copy of the setup; lighter cuts get feeds up to the heaviest cut the same tool already makes (or the limits in goals), moves in air go at the axes' top speed, finishing passes and plunges keep theirs, a chattering tool gets the steadiest speed near its own (0.7 to 1.8 times it, the depth within 10 % of it the deepest) where that takes the cut half as deep again, collinear lines are merged, rapid retracts come down to just over the blank and the fixture, arcs are fitted where the sender could not keep up. The result runs again on a second copy (lines over a limit are lowered and it runs once more). Answers with the program, its changes and the comparison; saved as a version (author optimizer) when it is better in every respect. Takes two or three runs of the program: a run id comes back after budget_s.

Parameters

Name	Type	Default	Description
`gcode`	string		the program's text
`name`	string	`"program.nc"`	
`file`	string		instead of gcode: the program's path on this computer (a .nc .ngc .gcode .tap ... file, up to 32 MB) - read and kept in the working folder as a version under its name
`stock`	string	`"current"`	current: the machine's workpiece as it is, cuts and all (carry on with a part already cut); fresh: a new blank (a whole program from the start) (one of `current`, `fresh`)
`example`	string		an example's setup (and program, when gcode is omitted) instead of the machine's: see setup_example
`setup`	object		an explicit setup instead of the machine's: {spec: the workpiece's setup as machine_status shows it (status.workpiece), tool, tools: {T: spec}, g54: [x,y,z] MPos, part: STL path}
`goals`	object		
`goals.feeds`	boolean	`true`	
`goals.speeds`	boolean	`true`	
`goals.merge`	boolean	`true`	

Name	Type	Default	Description
`goals.retracts`	boolean	`true`	
`goals.arcs`	boolean		default: when the sender could not keep up
`goals.chip_mm`	number		
`goals.force_n`	number		
`goals.load_pct`	number		
`goals.load_cap`	number	`80`	
`goals.feed_max`	number		
`goals.arc_acc`	number		mm/s2 towards an arc's centre: its feed no more than 60 sqrt(arc_acc r) - the controller's (the MC-1's kernel: sqrt(3)/2 of the axes' 300 = 260); default 0, no bound
`goals.air_feed`	number		
`goals.finish_ae`	number	`0.15`	
`timeout_s`	number	`3600`	simulated seconds allowed
`budget_s`	number	`40`	wall-clock seconds this call waits before it answers with a run id (at most 45: clients give up after 60 s; run_status waits for the rest)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `optimized`, `verdict`, `gcode`, `status`, `run`.

`geometry_import` - Import a shape

changes the machine · the same call twice does no more than once

Make a shape the programs can cut, from what you write - SVG path data (svg_path, its numbers in mm: e.g. a heart "M 10,30 A 20,20 0,0,1 50,30 A 20,20 0,0,1 90,30 Q 90,60 50,90 Q 10,60 10,30 z"), an SVG or DXF drawing (svg, dxf: the text) or file (an .svg or .dxf path; a DXF's layers: [names] reads only those), formulas in t (curve: {x: "16*sin(t)^3", y: "13*cos(t)-5*cos(2*t)-2*cos(3*t)-cos(4*t)", t: [0, "2*pi"]}), basic shapes added or cut in order (shapes: rect, circle, ellipse, slot, polygon, star, path), single-stroke text (text, font, text_height, align) or points - instead of working out its points yourself. Then where it goes: width_mm or height_mm (scaled to fit), scale, rotate, mirror, and center [x, y] (its box's middle) or at [x, y] (its lower left corner) in the program's mm (Z0 on the blank's top). SVG's y points down: it is turned the right way up. Saved under name as a version (the same shape again: not saved again); the answer says its size and box, outlines and islands, area, the sharp inner corners a pocket's tool cannot get into, the largest tool that fits and the narrowest place, its symmetry - with a picture: look at it before cutting. Then program_generate {features: [{op: "pocket", geometry: name, depth}]} (or contour, engrave). A mesh (file: .stl, .obj, .3mf, .glb, .gltf; units, up, width_mm or scale, depth_mm, rotate, mirror, center or at, top) is saved the same way for 3D: rough3d and finish3d take it. With only name: a saved shape again (version: an older one); with nothing: the shapes there are. Units are mm; a shape is data - it does not touch a machine.

Parameters

Name	Type	Default	Description
`name`	string		the shape's name: saved under it, used by it (geometry: name)
`version`	integer		with only name: that saved version (default the last)
`svg_path`	string		SVG path data: M L H V C S Q T A Z, lower case relative
`svg`	string		an SVG document's text
`dxf`	string		a DXF drawing's text (ASCII): lines, arcs, circles, polylines, splines, ellipses, blocks
`file`	string		an .svg or .dxf file's path - or a mesh for 3D: .stl, .obj, .3mf, .glb, .gltf
`layers`	array or string		DXF: only these layers (default: all that are on)
`join`	number	`0.01`	DXF: mm within which ends are joined into closed outlines
`curve`	object		formulas in t (radians; sin cos tan sqrt abs exp ln pow pi ...)
`curve.x` *	string		
`curve.y` *	string		
`curve.t`	array of ['number', 'string']		
`curve.closed`	boolean	`true`	
`shapes`	array of object		rect {x,y,w,h[,r]}, circle {cx,cy,d}, ellipse {cx,cy,rx,ry}, slot {x0,y0,x1,y1,w}, polygon {points} or {cx,cy,n,d}, star {cx,cy,n,d,d_inner}, path {svg}; op: add, cut, intersect
`text`	string		text: single strokes to engrave (font simplex, duplex, script: Latin letters, digits, signs) or the letters' outlines to pocket or V-carve (an outline font)

Name	Type	Default	Description
`font`	string		simplex (default), duplex, script: single strokes; sans (Noto Sans, comes with Studio) or a font of the computer's by its name - "微软雅黑", "宋体", "Arial" ... (Chinese: a computer's font): outlines
`text_height`	number		the capitals' height, mm (10)
`align`	string		(one of `left`, `center`, `right`)
`points`	array of array		a polygon: [[x, y], ...]
`units`	string	`"mm"`	svg_path's numbers; a DXF's that does not say its own; an STL's or OBJ's (mm, cm, m, in) (one of `mm`, `cm`, `m`, `in`, `pt`, `px`)
`up`	string	`"z"`	a mesh: which of its file's axes points up (most OBJ: y) (one of `z`, `-z`, `y`, `-y`, `x`, `-x`)
`depth_mm`	number		a mesh: its Z size (Z alone: a relief made shallower)
`top`	number	`0`	a mesh: the Z its highest point goes to
`bottom`	number		a mesh: the Z its lowest point goes to (instead of top)
`px_per_in`	number	`96`	
`flip_y`	boolean		
`scale`	number		
`width_mm`	number		
`height_mm`	number		
`mirror`	string		(one of `x`, `y`)
`rotate`	number		degrees, counter-clockwise
`center`	array of number		
`at`	array of number		
`tolerance`	number	`0.005`	mm: curves that are not circles become lines this close
`fill_rule`	string	`"evenodd"`	(one of `evenodd`, `nonzero`)

Returns (structured content): `machine`, `notice`, `name`, `version`, `size_mm`, `bbox`, `shapes`.

`geometry_nest` - Lay parts out on the blank

read-only · the same call twice does no more than once

Lay several parts out on the machine's blank (or a plate given) and cut them out - nesting. Each part a shape: geometry (a name from geometry_import) or one given as geometry_import takes it (svg_path, dxf with layers, shapes ...); count copies of it, each turned by quarter turns where that fits them better (turns: 0, 90, 180, 270; default all), its own features (holes, pockets ...) written in its shape's coordinates. They go in the largest first, each where it fits lowest, then leftmost - the shapes, not their boxes: a small part in a larger one's notch or its hole (in_holes) - gap apart (default the tool's diameter + 1 mm) and margin (1) inside the plate (default the blank's top). Nothing is machined: the answer says where each copy went, how much of the plate the parts use, the copies that did not fit (placed less than asked), with a picture; and features - each copy's own where it went, then every copy cut out, those in holes first (a contour outside its shape, depth deep - default through the blank and 0.2 - with tabs, 3 unless given) - for program_generate as they are, with nest (kept with the job).

Parameters

Name	Type	Default	Description
`parts` *	array of object		the parts, 1 to 16, e.g. [{geometry: "bracket", count: 4}, {shapes: [{type: "circle", cx: 0, cy: 0, d: 20}], count: 3}]
`plate`	array of number		[x0, y0, x1, y1] where the parts go, the program's mm (default the blank's top)
`depth`	number		how deep each part is cut out (mm; default the blank's thickness + 0.2)
`gap`	number		mm between two parts (default the tool's diameter + 1)
`margin`	number	`1`	mm from the plate's edge to the parts' outlines
`tabs`	integer or object		the cut-outs' tabs: a count or {count, width, height, holes: false - none round the parts' holes} (default 3)
`tool`	string		the tool the gap is for (default the one in the spindle)
`setup`	object		the setup to measure the blank and the tool on instead of the machine's, as program_generate takes it (spec, tool, tools, g54): a hidden copy set up so
`in_holes`	boolean	`true`	a small part may go into a larger one's hole (cut out before it); false: the holes taken as filled
`outlines`	boolean	`false`	each placed copy's outline where it went and each part's own: {outer: [[x, y], ...], holes: [...]}; each copy's tabs where the program will leave them: [[x, y, nx, ny], ...] on its outline, the normal away from its material - for drawing the layout

Name	Type	Default	Description
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `placed`, `parts`, `used_pct`, `features`, `nest`.

`program_generate` - Generate a program from features

changes the machine · the same call twice does no more than once

Write a program from machining features instead of line by line: face, pocket, slot, contour (outside, inside or on an outline; tabs hold a part cut through), drill, bore (helix), ream (a reamer widening holes drilled before to its size and class: drill, then ream), engrave (single-stroke text, or a shape's lines), chamfer (a V-bit along edges), thread (a thread mill in a bored hole), vcarve (a V-bit's tip along an outline's middle lines, each point as deep as the V reaches: lettering in an outline font, any closed shape). A pocket or a contour takes any outline, islands and all: a shape made with geometry_import (geometry: its name), SVG path data right in the feature (svg_path with geometry_import's width_mm, center ...), a rectangle with corner radius, a circle, polygon points, or the closed outline another program cuts (from_program). E.g. {op: "pocket", geometry: "heart", depth: 4}; {op: "pocket", svg_path: "M 10,30 A 20,20 0,0,1 50,30 A 20,20 0,0,1 90,30 Q 90,60 50,90 Q 10,60 10,30 z", width_mm: 40, center: [30, 30], depth: 4}; {op: "pocket", from_program: "heart.nc", offset: 3, depth: 13} (a hollow where heart.nc cut the heart's outline; offset: grown by the tool's radius). A pocket is cleared in rings from its deepest spot out - or, strategy "adaptive", by a walk that cuts a set width (ae, 10 % of the tool) all the way: deeper layers, a faster feed for the thinner chips, no slotting but the entry; or "zigzag", straight passes to and fro - its walls, round the outside and every island, finished to size (a slot: strategy "trochoidal", loops along it); the answer says where the tool cannot reach (corners narrower than its radius stay round). Speeds, feeds and depths come from the blank's material and the tool, within the spindle's range: with verify, the twin's stability model picks the speed and keeps each depth (roughing steps, the walls' finishing passes) to 80 % of the stable one before the program is written. Ramps and helices into material, climb milling. The clearance height stays within what the setup leaves above the blank. With verify (default) it runs on a hidden copy of the setup, its workpiece as it is (stock), and where the tool would still chatter it is written again shallower there - a program too long to try within the call likewise behind it (verification.check_run: run_status gives the last version's report, closed_loop what it changed). Saved as a version (author generator; each one the loop tried too). operations says each feature's lines, layers and values - the first 64; past them (a nest's) feature_spans gives every feature's [first line, last line, layers, tool]. A hole's or a shaft's diameter (a circle's pocket or contour, a bore, a reaming) may carry its fit (ISO 286): d "Ø8 H7", or d 8 with fit "H7", a pair "H7/g6" (the hole takes H7, the shaft g6) or a common fit (sliding, locating, transition, press), or tol "+0.015/0" - cut to the middle of its zone, a spring pass where the twin's estimate of the tool's give asks for one (operations[].fit says what and why: the twin's estimate). So may a slot's width (width "6 N9", or width 6 with fit "N9": a keyway's N9, P9, JS9 - cut with a cutter narrower than it, its walls to size) and a rectangle's pocket or contour, both its sizes (fit "H7"; outside: a shaft's class, a key's h9). The program's zero: the machine's G54 (setup_work_zero). A part machined from two sides or more: setups instead of features - a program each, setup 1 on the machine as it is, each later one turned over (turn) or held another way (fixture) from the one before, G54 set again on the blank (zero); the job and every program saved together.

Parameters

Name	Type	Default	Description
`stock`	string	`"current"`	current: the machine's workpiece as it is, cuts and all (carry on with a part already cut); fresh: a new blank (a whole program from the start) (one of `current`, `fresh`)
`features`	array of object		each {op, ...}: face {x0,y0,x1,y1,depth}; pocket {geometry name | svg_path (+ width_mm, center) | shape rect x0,y0,x1,y1[,r] | shape circle cx,cy,d | shape polygon points [[x,y],..] | from_program name[,version]; [offset]; [strategy adaptive[, ae] | zigzag[, angle]]; [rest true]; depth}; slot {x0,y0,x1,y1[,width],depth[,strategy trochoidal[,step]]}; contour {the same outlines, side outside|inside|on, depth}; drill {points [[x,y]..],depth[,peck]}; bore {points,d,depth}; engrave {text,x,y,height,depth[,font] | geometry name, depth[, surface: a mesh's name - on a curved top, depth under it]}; chamfer {an outline, width} (a V-bit); thread {points,d,pitch,depth[,hand]} (a thread mill); groove {points,d,bore,depth[,width,step]} (a T-slot cutter: a ring cut into a round hole's wall); perlage {an outline, depth[, pitch, stagger, margin, dwell]} (a flat tool: overlapping spots over a face); vcarve {an outline, or text with font sans or a computer's font, text_height, center[, max_depth]} (a V-bit); rough3d {geometry: a mesh's name[, allowance, stepdown, ae, area, rest_from, rest_allowance, rest_stepdown]} (a flat end mill); finish3d {geometry[, strategy, scallop or stepover, angle, steep_angle, area, rest_from]} (a ball end mill; strategy pencil: along the creases); contour tabs: n or {count,width,height,holes}
`setups`	array of object		instead of features: a part machined from two sides or more, a program each, in order. Setup 1 is the machine's setup as it is; each later one turned over (turn) or held another way (fixture) from the one before, G54 set again on the blank (zero). Within the call each later program is written for the blank as it was (what the setups before cut is not known yet). name, tool, material, cutting and safe_z here are every setup's defaults. E.g. setups: [{features: [{op: "pocket", shape: "rect", x0: 8, y0: 8, x1: 24, y1: 22, depth: 3}]}, {turn: "u", features: [{op: "face", x0: 0, y0: 0, x1: 40, y1: 30, depth: 1}]}]
`nest`	object		geometry_nest's nest (its parts laid out on a plate): kept with the job beside the features it made; given without features, the parts are laid out here (plate, depth and the parts needed) - not with setups
`tool`	string		default: the machine's tool in the spindle
`material`	string		default: the machine's blank's; a code of the machine's table (6061-T6, 45, 304 ...) or another standard's name for one (EN AW-6061, C45, SUS304 ...: read as that code, the nearest grade - for its cutting data)
`cutting`	object		rpm, feed, ap (depth of cut), ae (stepover), finish_ap to use instead
`setup`	object		the setup to generate for and verify on instead of the machine's (as program_optimize takes it)
`safe_z`	number	`5`	the clearance height over Z0; kept within what the setup leaves above the blank
`name`	string	`"generated.nc"`	
`title`	string		the job's name to show (one line, 80 characters at most; kept with the job, not in the program)
`description`	string		a line about the job (200 characters at most; kept with the job)

Name	Type	Default	Description
`verify`	boolean	`true`	
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `gcode`, `verification`, `operations`, `setups`, `total`.

`program_edit` - Change a program by intent

changes the machine · the same call twice does no more than once

Change a program without sending it again line by line: edits applied in order - set_s value (every S), set_f value or scale (every F), shift_z by and scale_z factor (the cuts: Z at or below the program's zero; all: every Z), split into (from..to: one pass cut in that many layers, a ramp kept a ramp - back up and over to its start in between), safe_z z (the clearance height: every Z above zero); from/to: file lines (default all). G53, G28, G30, G10 and G92 lines are left alone; programs in absolute mm (G90 G21). The program: gcode, or name (and version: default the last) from the working folder. The result is saved as a new version (save false: not) with the difference; check it with program_check {name, version}.

Parameters

Name	Type	Default	Description
`gcode`	string		the program's text (or name and version)
`name`	string		
`version`	integer		with name, no gcode: that saved version (0: the last)
`edits` *	array of object		each {op, ...}: set_s {value}; set_f {value | scale}; shift_z {by[, all]}; scale_z {factor[, all]}; split {from, to, into}; safe_z {z}; any of them with from, to (file lines)
`save`	boolean	`true`	

Returns (structured content): `machine`, `notice`, `gcode`, `version`, `changed_lines`, `diff`.

`program_versions` - Program versions

changes the machine · the same call twice does no more than once

Program versions in the working folder (Documents/Twinwright Studio/programs/<name>/vNNNN.nc with who, when and why). action list: the versions (without name: every program, the last changed first); get: a version's text (default the last); save: gcode as a new version (reason: what changed and why; the same text is not saved twice); diff: how b differs from a - texts, or name with a_version and b_version - with their simulation results side by side when both ran on this host. The user's own files are never written.

Parameters

Name	Type	Default	Description
`action`	string	`"list"`	(one of `list`, `get`, `save`, `diff`)
`name`	string		
`version`	integer		
`gcode`	string		
`reason`	string		
`a`	string		
`b`	string		
`a_version`	integer		
`b_version`	integer		

Returns (structured content): `machine`, `notice`, `name`, `version`, `versions`, `gcode`, `programs`.

`job_get` - A program's CAM job

read-only · the same call twice does no more than once

The CAM job a program was generated from - by program_generate or in the window's CAM page, the user's or an AI's - as it was kept with a version (default the last): its operations (features, or setups), tool, tool table, material, cutting values, safe_z, title; a line on each operation (job_edit numbers them so), who made it and when, its versions. An example loaded as shipped: the example's own job. Changes nothing.

Parameters

Name	Type	Default	Description
`name` *	string		the program's name (program_versions lists them)
`version`	integer		the job's version (0: the last)

Returns (structured content): `machine`, `notice`, `name`, `version`, `program_version`, `author`, `versions`, `job`, `operations`.

`job_edit` - Change a program's CAM job

changes the machine · the same call twice does no more than once

Change the CAM job a program was generated from (job_get shows it) and generate it again - as program_generate does, with its answer: the program's next version, the job's beside it. Each edit in turn: {feature: k, set: {depth: 3, tool: 2}} changes operation k's values (null removes one); {feature: k, remove: true}; {feature: k, move: j} moves it to place j, the others in their order (with set: changed too); {insert: k, op: {op: \"face\", ...}} a new operation before k (k one past the last: at the end); {set: {safe_z: 8, material: \"7075-T6\"}} the job's own values (tool, tools, material, cutting, safe_z, title, description - not its operations or name). A job in setups: setup: s in each edit. Nothing is changed or generated when an edit cannot be made.

Parameters

Name	Type	Default	Description
`name` *	string		the program's name
`version`	integer		the job's version to start from (0: the last)
`edits` *	array of object		
`verify`	boolean	`true`	
`stock`	string	`"current"`	current: the machine's workpiece as it is, cuts and all (carry on with a part already cut); fresh: a new blank (a whole program from the start) (one of `current`, `fresh`)
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `gcode`, `verification`, `operations`, `setups`, `total`.

Runs and results

`run_status` - Runs

changes the machine

Without run: the runs kept on this host (the last 16) - program, machine, who started it, running or done, a summary. With run: waits up to wait_s for it (at most 45 s: clients give up after 60 s) and gives its full report - how it ended first (done, stopped by whom, an alarm, out of time) and at which line; stop: true ends it early. A call that is cancelled or times out never stops a run.

Parameters

Name	Type	Default	Description
`run`	string		a run id (program_simulate's, program_run's), or "latest": your newest run
`wait_s`	number	`40`	at most 45
`stop`	boolean	`false`	

Returns (structured content): `machine`, `notice`, `run`, `completed`, `simulated_time_s`, `issues`, `status`, `runs`.

`run_profile` - Where a run's time went

read-only · the same call twice does no more than once

A run line by line (a run id from program_simulate or program_run; default the latest): the time each line took (they add up to the run), the feed asked for and reached and why it fell short (the axes' top speed, corners, arcs, the planner's look-ahead, a sender that could not keep up, the override, a hold), the cutting force, chip, chatter margin, spindle load, material removed, cutting air, events. sort: time (slowest first), risk (events first) or line; from/to: a range of file lines; paged (offset, limit).

Parameters

Name	Type	Default	Description
`run`	string		a run id (program_simulate's, program_run's), or "latest": your newest run
`sort`	string	`"time"`	(one of `time`, `risk`, `line`)
`limit`	integer	`20`	

Name	Type	Default	Description
`offset`	integer	`0`	
`from`	integer		
`to`	integer		
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `run`, `total_s`, `lines`.

`result_compare` - Compare with the design part

read-only · the same call twice does no more than once

How a machined workpiece deviates from its design part: cells within tolerance, material left above the part, cut into it. run: the workpiece a run left; else the machine's workpiece now.

Parameters

Name	Type	Default	Description
`run`	string		a run id (program_simulate's, program_run's), or "latest": your newest run
`tol`	number	`0.02`	
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `comparison`, `in_tolerance_pct`.

`result_image` - Picture of a result

read-only · the same call twice does no more than once

A picture (PNG) and its key numbers in words - kind: workpiece = the workpiece in 3D (view iso, top, front, right; color material, height, line = the program line that cut there last, deviation; path: a run's toolpath over it); toolpath = the path coloured by feed, force, chip, margin, line, z or limit (why the programmed feed was not reached), rapids dashed, the highest value marked with its line (view top or iso); deviation = the design part's deviation map from above, the worst places marked with their program lines (needs a part); profile = the slowest lines, coloured by what limited them; timeline = cutting force, spindle load, feed, chatter margin and the program line over time; comparison = two runs side by side (runs). run: a run of program_simulate - its end state; without it, workpiece and deviation show the machine now and the others its latest simulation (a simulation runs on a copy: to see its result, name its run). A picture larger than about 100 KB is drawn smaller (clients limit a result's size). The key numbers come in the text too.

Parameters

Name	Type	Default	Description
`kind`	string	`"workpiece"`	(one of `workpiece`, `toolpath`, `deviation`, `profile`, `timeline`, `comparison`)
`run`	string		a run id (program_simulate's, program_run's), or "latest": your newest run
`runs`	array of string		comparison: two run ids, A then B
`view`	string		(one of `iso`, `top`, `front`, `right`)
`color`	string		workpiece: material, height, line, deviation; toolpath: feed, force, chip, margin, line, z, limit
`path`	boolean	`false`	
`top`	integer	`15`	profile: lines shown, at most 40
`tol`	number	`0.02`	deviation: the tolerance band, mm
`width`	integer	`960`	
`height`	integer	`640`	
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `image`, `kind`, `of`.

`result_export` - Export the machined part

changes the machine

The workpiece as the machine has cut it, into a file the user can open in other programs (CAD, slicers, viewers) or send on: format stl = a closed binary STL mesh of the material left, in frame program (the program's zero, as CAM draws the part), blank (the blank's front left corner at its lowest point) or machine (machine coordinates); format png = a 16-bit height map seen from above. res: the cells' size, mm (left out: the finest of 0.02 .. 0.5 mm within about a million cells, never finer than the workpiece's own cells). Written to path - a .stl / .png file, or a folder (made if missing) for a name of its own - else into the exports folder (Documents/Twinwright Studio/Exports); an existing file is kept unless overwrite. The same cells as inspect_part and result_compare read. Answers the file's path, size and triangles; after program_run or program_simulate on that machine it is the machined part.

Parameters

Name	Type	Default	Description
`format`	string	`"stl"`	(one of `stl`, `png`)
`frame`	string	`"program"`	(one of `program`, `blank`, `machine`)
`res`	number		cell size, mm (left out: chosen as said)
`path`	string		a .stl / .png file, or a folder; left out: the exports folder
`overwrite`	boolean	`false`	
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `path`, `bytes`, `format`, `frame`, `res_mm`, `triangles`, `removed_mm3`.

`part_save` - Part library: save the workpiece

changes the machine

The machine's workpiece as cut into the user's part library (the same library as the Studio window's File > Part library..., the same record as its "Save to part library..."): a folder in Documents/Twinwright Studio/Parts with workpiece.stl (the machined part, the program's frame), top.png (it from above) and part.json - the program's name, version and SHA-256 (the program itself stays in the programs' folder), the blank, fixture and tool table, the machine, the last run's time, alarms and collisions, and what you add: name (left out: the program's), tags, notes, and the inspection (inspect_part's summary) or comparison (result_compare's) to keep with it. Saved as by you ("made_by": "ai:<your name>"). part_list lists the records.

Parameters

Name	Type	Default	Description
`name`	string		
`tags`	array of string		
`notes`	string		
`inspection`	object		
`comparison`	object		
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `id`, `folder`, `files`, `record`.

`part_export` - Part library: export a record's part

changes the machine

A part library record's machined part (format stl: workpiece.stl, a closed mesh in the program's frame) or its picture from above (png: top.png) copied to path - a .stl / .png file, or a folder (made if missing) - else into the exports folder (Documents/Twinwright Studio/Exports), named after the record; an existing file is kept unless overwrite. For the workpiece on a machine now: result_export.

Parameters

Name	Type	Default	Description
`id` *	string		
`format`	string	`"stl"`	(one of `stl`, `png`)
`path`	string		
`overwrite`	boolean	`false`	

Returns (structured content): `id`, `path`, `bytes`, `format`.

`part_load` - Part library: make another the same

changes the machine

A part library record set up again on a machine as it was cut, as the Studio window's "Make another the same…" does and as an example loads: its blank and fixture, T1 in the spindle, the tool table, homing, the work zero as recorded, and the program - the version it was cut with, its SHA-256 checked - loaded, not started (program_run starts it). The window's machine only with the user's yes there. Refused when that version is not in the programs' folder any more or its text changed; when it would not be as it was cut - another machine, a tool the program calls missing from the recorded table - refused saying what differs, nothing changed: tell the user, and only if they still want it call again with force: true.

Parameters

Name	Type	Default	Description
`id` *	string		
`force`	boolean	`false`	
`machine`	string		which machine. Leave it out for the machine the user sees and means - the one in their Twinwright Studio window ("window"; needs the window open with its AI switch on). "sandbox" (or "sandbox-<name>") is an invisible practice machine of yours, faster than real time (far faster moving, a few times while cutting), shared by the user's AI clients: the user does not see it - use it only to try something out, and say so when you report. No other names. Left out while the window's AI switch is off: "sandbox"

Returns (structured content): `machine`, `notice`, `id`, `name`, `loaded`, `program`, `version`, `zero`, `started`, `differs`.

`log_query` - Audit log

read-only · the same call twice does no more than once

The audit log: who did what, when, to which machine - the user in the window, the AI clients, the machines themselves (alarms, homing, programs started and ended, collisions), the twin host. Newest first; filters: since / until (a time's start, e.g. "2026-09-29T17:5"), category (machine, operator, program, setup, ai, system), action (a prefix: "machine.alarm", "ai.tool"), who (user, ai, system, or a name), machine, text, severity (warn, error); paged. Useful to see what the user did in the window, or what became of a machine.

Parameters

Name	Type	Default	Description
`since`	string		
`until`	string		
`category`	string		(one of `machine`, `operator`, `program`, `setup`, `ai`, `system`)
`action`	string		
`who`	string		
`machine`	string		
`text`	string		
`severity`	string		(one of `warn`, `error`)
`limit`	integer	`30`	
`offset`	integer	`0`	

Returns (structured content): `machine`, `notice`, `records`, `matching`.

The licence

`licence_status` - Licence and edition

read-only · the same call twice does no more than once

This copy's edition and licence: the edition in force (in_force: learning, maker, pro, shop; classroom, trial), the licence files in the computer's licence folder - each one's kind, holder, seats, dates (valid_until, renew_after, updates_until), days left (a subscription past its period: its days of grace) and whether it holds, and why not - the local trials used, whether a bought licence's updates cover this release, every paid feature with the edition it needs, why it is closed and whether this release has it at all (shipped false: not built yet - never offer it), and the site's pages for the next step (pricing: buy or upgrade; licence: a year more of updates, managing a subscription, offline activation). Changes nothing; gives no activation code and no computer fingerprint. Use it to explain a refusal ("edition: … comes with the pro edition") and what the user can do.

No parameters.

Returns (structured content): `edition`, `in_force`, `files`, `trials_used`, `features`, `links`.

The part library

`part_list` - Part library: the records

read-only · the same call twice does no more than once

The user's part library: machined parts kept with what made them (Documents/Twinwright Studio/Parts/<id>/), the newest first - each record's id, name, created, made_by (user or ai:<client>), the machine, the program (name, version, SHA-256), the setup, material, tools, results (removed volume, machine time, chatter, collisions, alarms), the inspection and the comparison with the design, files (the machined workpiece as an STL in the program's coordinates, a picture), tags and notes. Filters: query (in the name, notes or program), material, program, tag, since (an ISO time's start, e.g. "2026-10-12"); limit (default 50). deleted: the trash instead (kept 30 days). Changes nothing.

Parameters

Name	Type	Default	Description
`query`	string		
`material`	string		
`program`	string		
`tag`	string		
`since`	string		
`limit`	integer	`50`	
`deleted`	boolean	`false`	

Returns (structured content): `parts`, `matched`, `total`, `folder`.

`part_get` - Part library: one record

read-only · the same call twice does no more than once

One record of the part library (part_list gives the ids): all it keeps, and the paths of its files on this computer (the workpiece's STL, its picture, the program as it was cut). Changes nothing.

Parameters

Name	Type	Default	Description
`id` *	string		

Returns (structured content): `id`, `record`, `paths`.

`part_update` - Part library: name, tags, notes

changes the machine · the same call twice does no more than once

A record's name, tags (a list of words: replaces them) or notes ("" clears them). Nothing else of a record changes: it says what was made, and how.

Parameters

Name	Type	Default	Description
`id` *	string		
`name`	string		
`tags`	array of string		
`notes`	string		

Returns (structured content): `id`.

`part_delete` - Part library: delete or restore

changes the machine (destructive)

A record into the part library's trash - only with the user's yes in the Studio window, asked whatever the settings say (the window not open: refused - the user deletes it there, File > Part library...). It stays in the trash 30 days; restore: true brings it back (no question). part_list {deleted: true} lists the trash.

Parameters

Name	Type	Default	Description
`id` *	string		
`restore`	boolean	`false`	

Returns (structured content): `id`, `deleted`, `restored`.

Reports to support

`report_draft` - Report a problem: draft it for the user

changes the machine

Drafts a report to Twinwright Studio's support (a problem, how to use something, an idea) and opens it in the user's Studio window (Help > Report a problem..., marked as drafted by you). Only the user sends it - you cannot - and only the user attaches the diagnostics package or a picture of the window. Write what happened as the user would: what was done, expected, seen (the version, the system and the audit log's last records go with it anyway). No Studio window open: nothing is shown - tell the user to report it there.

Parameters

Name	Type	Default	Description
`kind`	string	`"problem"`	(one of `problem`, `howto`, `idea`)
`text` *	string		what happened (at most 4000 bytes)

Returns (structured content): `shown`.

`report_list` - My reports: their state and replies

read-only · the same call twice does no more than once

The reports sent from Studio on this computer (Help > My reports) as the window last saw them: each ticket's id ("TW-1042"), title, kind, status (new, in_progress, answered, solved), when it last changed, the replies not read yet, and its conversation as the window last fetched it (messages: from you or support, at, text, attachments). id: that one only. Read-only: nothing is asked of the support service.

Parameters

Name	Type	Default	Description
`id`	string		

Returns (structured content): `reports`, `count`.

The user's window

`window_show` - Show the user something in the Studio window

changes the machine · the same call twice does no more than once

As you guide the user, show them in their own Studio window what you mean: a workspace (setup, cam, program, run, analysis, diagnostics), the 3D view's side (iso, front, right, top), the tool tip's trail, the program's path and the tool's close-up on or off, the trail's colouring (plain, feed, force, chip, margin, line) and the workpiece's (material, deviation, line), the dock's page (program, curves, console, messages, tools, ai; in the CAM workspace checks, versions, preview; in Setup inspect), a step of the Setup panel (power, homing, workpiece, tools, zero, program, cut). The window does it as the user's own clicks would: only what it shows changes - nothing on the machine, no yes asked. Give one or more; they are done in that order (the workspace first).

Parameters

Name	Type	Default	Description
`workspace`	string		(one of `setup`, `cam`, `program`, `run`, `analysis`, `diagnostics`)
`view`	string		(one of `iso`, `front`, `right`, `top`)
`trail`	boolean		
`path`	boolean		
`close_up`	boolean		
`trail_colour`	string		(one of `plain`, `feed`, `force`, `chip`, `margin`, `line`)
`stock_colour`	string		(one of `material`, `deviation`, `line`)
`dock`	string		(one of `program`, `curves`, `console`, `messages`, `tools`, `ai`, `checks`, `versions`, `preview`, `inspect`)
`setup_step`	string		(one of `power`, `homing`, `workpiece`, `tools`, `zero`, `program`, `cut`)

Returns (structured content): `shown`, `what`.

`settings` - The window's settings: read, or change with the user's yes

changes the machine · the same call twice does no more than once

The settings of the user's Studio an AI may read and change: language ("auto" or a language pack's tag such as "zh-CN"), region (how numbers, money and dates look: "auto", "lang" or such as "de-DE"), units ("auto", "metric", "imperial" ...), currency (ISO 4217, "": the region's), currency_symbol and machine_rate (the machine-hour rate for cost estimates, in that currency; 0: none). Without arguments: their values and the choices. With some: the user says yes in the window, then it applies them as its own Settings' Apply. The others (your own access, privacy, the log, the look) only the user, in Tools > Settings.

Parameters

Name	Type	Default	Description
`language`	string		
`region`	string		
`units`	string		
`currency`	string		
`currency_symbol`	string		
`machine_rate`	number		(at least 0)

Returns (structured content): `settings`, `choices`, `set`, `what`.

The manual

`manual_search` - Search Studio's manual

read-only · the same call twice does no more than once

The manual Studio has on this computer (the user manual, the operations manual, the technical manual, the interface reference - what Help > Documentation shows), searched as the window searches it: every section with all of q's words (any case). Each hit: its chapter, where it is (the breadcrumb), its page (what Help > Documentation opens) and a snippet with the word found between **. lang: the manual's language (left out: the interface's; one it has not: English). Read-only; answer the user from it rather than from memory.

Parameters

Name	Type	Default	Description
`q` *	string		the words to find
`lang`	string		such as zh-CN, de, ja
`max`	integer	`8`	(at least 1; at most 30)

Returns (structured content): `lang`, `places`, `pages`, `hits`.

设置

控制器的全部设置。`$<n>` 看一个，`$<n>=<value>` 改它，`$$` 列出全部。 表里的值是 MC-1 的。 标着 *重启* 的设置要控制器重启以后才生效。

目录

- 一般
- 控制信号
- 限位
- 冷却
- 主轴
- 换刀
- 回零
- 探测
- 停车和安全门
- 点动
- Modbus
- 步进
- 变频器
- X 轴
- Y 轴
- Z 轴

一般

设置	名字	单位	类型	范围	MC-1	说明
`$10`	状态报告的内容		位域		15871	实时状态报告里有什么。解析器状态单独发一行，只在变了时发。 各位： 0 机床坐标的位置, 1 缓冲区状态, 2 行号, 3 进给和转速, 4 引脚状态, 5 工件坐标的偏置, 6 倍率, 7 探测坐标, 8 工件坐标偏置变了时等缓冲区走空, 9 解析器状态, 10 报警的子状态, 11 运行的子状态, 12 回零时也发, 13 剩余距离
`$11`	拐角偏差	mm	小数		0.002	两段运动之间的拐角处，路径可以偏开多远；它决定拐角的速度。越小越慢、越贴路径。
`$12`	圆弧公差	mm	小数		0.002	圆弧走得多贴。
`$13`	按英制报告		开关		0	报告里的位置和速度用英制。
`$28`	G73 的退刀距离	mm	小数		0.100	G73 每次啄钻之间的短退刀。
`$32`	工作方式		选项		0	铣床、激光或车床。 选项： 0 普通, 1 激光方式, 2 车床方式
`$39`	老式实时字符		开关		1	'?'、'!'、'~' 在哪里都当实时命令，$ 命令和注释里也一样。
`$60`	恢复倍率		开关		0	程序结束时倍率回到 100 %。
`$62`	允许休眠		开关		0	空闲很久以后进入休眠。
`$63`	进给保持时做的事		位域		3	进给保持除了停下运动还做什么。 各位： 0 保持时关激光, 1 恢复时恢复主轴和冷却, 2 转速降到最低（激光的 M4 除外）
`$64`	开机时报警		开关		0	开机后处于报警状态。
`$384`	不保存 G92		开关		0	断电再上电后不保留 G92 偏置。
`$398`	规划器的块数 *(重启)*		整数	30–1000	100	规划器向前看多少段运动。
`$481`	自动报告的间隔 *(重启)*	ms	整数	100–1000	0	按这个间隔主动发状态报告（0：不发）。
`$484`	急停后要解锁		开关		1	急停以后机床一直锁着，直到解锁（$X）或回零。
`$486`	锁住坐标系		位域		0	G10 不能改的坐标系。 各位： 0 G59.1, 1 G59.2, 2 G59.3
`$676`	复位时做的事		位域		15	复位时清掉什么。 各位： 0 位置丢了时清掉回零状态, 1 清掉偏置（G92 除外）, 2 清掉快移倍率, 3 清掉进给倍率

设置	名字	单位	类型	范围	MC-1	说明
$1030	发送软件断开就停作业		开关		1	运行作业的发送软件的链路断开 0.1 s（拔了 USB、关了端口）时，作业像收到停止（0x19）那样停下：运动减速停下，主轴和冷却关掉，它发来还没执行的行被扔掉，位置不丢。发送软件连回来后会收到通知。从文件系统跑的程序不靠发送软件，照常往下跑。

控制信号

设置	名字	单位	类型	范围	MC-1	说明
$14	控制输入反相		位域		0	反相的控制输入。 各位： 1 进给保持, 2 循环启动, 6 急停, 8 电机故障
$17	控制输入不用上拉		位域		0	不用内部上拉电阻的控制输入。 各位： 1 进给保持, 2 循环启动, 6 急停, 8 电机故障

限位

设置	名字	单位	类型	范围	MC-1	说明
$5	限位输入反相		轴掩码		7	反相的限位输入，比如接常闭开关时。
$18	限位输入不用上拉		轴掩码		0	不用内部上拉电阻的限位输入。
$20	打开软限位		开关		1	拒绝超出行程（$130 ……）的运动。要先能回零，位置才知道。
$21	打开硬限位		位域		1	限位开关一动作就立刻停下。严格方式：开关动作着时完全不能动。 各位： 0 打开, 1 严格方式 （第 0 位打开其余各位）

冷却

设置	名字	单位	类型	范围	MC-1	说明
$15	冷却输出反相		位域		0	低电平有效的冷却输出。 各位： 0 浇注, 1 喷雾
$673	冷却打开的延时	s	小数	0.5–20	0.0	M7/M8 之后给冷却启动的时间。

主轴

设置	名字	单位	类型	范围	MC-1	说明
$30	主轴最高转速	转/分	小数		24000.000	主轴的最高转速。
$31	主轴最低转速	转/分	小数		6000.000	主轴的最低转速。
$340	主轴到速的容差	%	小数		5.0	M3/M4 之后的运动要等主轴转速离编程转速在这个百分比以内（0：不等）。
$394	主轴启动的延时	s	小数	0.5–60	15.0	M3/M4 之后给主轴升到转速的时间（设了 $340 时：报警前最多等多久）。
$539	主轴停转的延时	s	小数	0.5–60	0.0	M5 之后给主轴停下的时间。

换刀

设置	名字	单位	类型	范围	MC-1	说明
$341	换刀方式		选项		3	M6 做什么。 选项： 0 普通, 1 手动对刀, 2 在 G59.3 手动对刀, 3 在 G59.3 自动对刀, 4 不理 M6
$342	换刀时的探测距离	mm	小数		70.0	找对刀仪时搜索多远。
$343	换刀时的定位速度	mm/min	小数		25.0	精确找到对刀仪用的慢速。
$344	换刀时的搜索速度	mm/min	小数		200.0	搜索对刀仪用的快速。
$345	换刀时退离测头的速度	mm/min	小数		200.0	离开对刀仪的速度。
$346	换刀选项		位域		0	换刀的选项。 各位： 0 M6 之后回到原来的位置, 1 在 G30 换刀, 2 快速退离测头
$485	重启后保留刀号		开关		0	断电再上电后保留主轴上的刀号。

回零

设置	名字	单位	类型	范围	MC-1	说明
$22	回零		位域		541	回零和它的选项。 各位： 0 打开, 1 允许单轴回零命令, 2 开机后必须回零, 3 回零后机床原点设为 0, 4 两个开关共用一个输入, 5 允许手动, 6 倍率锁, 9 每根轴单独的速度, 10 回零完成后才执行开机行 （第 0 位打开其余各位）
$23	回零方向反向		轴掩码		0	朝负方向回零的轴。

设置	名字	单位	类型	范围	MC-1	说明
$26	回零开关的去抖延时	毫秒	整数		250	开关动作以后，等多久再做下一段回零动作。
$27	回零开关的退离距离	mm	小数		2.000	回零时从开关退回多远。
$43	回零的趟数		整数	1–128	1	找开关动作点的次数。
$44	第一步回零的轴		轴掩码		4	最先回零的轴。
$45	第二步回零的轴		轴掩码		3	第二步回零的轴。
$46	第三步回零的轴		轴掩码		0	第三步回零的轴。

探测

设置	名字	单位	类型	范围	MC-1	说明
$6	测头输入反相		位域		1	测头输入反相，比如接常闭的对刀仪时。 各位： 0 测头
$19	测头输入不用上拉		位域		0	测头输入不用内部上拉电阻。 各位： 0 测头
$65	探测选项		位域		0	探测运动怎么处理。 各位： 0 允许进给倍率, 1 受软限位约束

停车和安全门

设置	名字	单位	类型	范围	MC-1	说明
$41	停车		位域		0	安全门打开时把主轴退开。 各位： 0 打开, 1 开机时关闭, 2 允许程序控制停车 （第 0 位打开其余各位）
$42	停车的轴		选项		2	停车用的轴。 选项： 0 X, 1 Y, 2 Z
$56	停车时慢退的距离	mm	小数		5.0	刀先慢慢退出多远。
$57	停车时慢退的速度	mm/min	小数		100.0	这段慢退的速度。
$58	停车的目标	mm	小数	≥ -100000	-5.0	停车的轴去哪里，用机床坐标。
$59	停车的快速	mm/min	小数		500.0	停车其余那段的速度。
$61	安全门选项		位域		0	安全门开着时哪些保持开着。 各位： 1 开门时冷却保持不变
$392	主轴启动的延时	s	小数	0.5–60	4.0	门关上以后给主轴升到转速的时间。
$393	冷却打开的延时	s	小数	0.5–20	1.0	门关上以后给冷却启动的时间。

点动

设置	名字	单位	类型	范围	MC-1	说明
$40	点动受软限位约束		开关		0	点动到软限位处停下，而不是被拒绝。

Modbus

设置	名字	单位	类型	范围	MC-1	说明
$374	Modbus 波特率		选项		3	Modbus 线的波特率（变频器的要一致）。 选项： 0 2400, 1 4800, 2 9600, 3 19200, 4 38400, 5 115200
$375	Modbus 接收超时	毫秒	整数	50–250	50	等 Modbus 应答等多久。
$681	Modbus 串口格式		选项		0	Modbus 线的校验方式（变频器的要一致）。 选项： 0 8 位，无校验, 1 8 位，偶校验, 2 8 位，奇校验

步进

设置	名字	单位	类型	范围	MC-1	说明
$0	步进脉冲宽度	微秒	小数	≥ 2.0	5.0	步进脉冲的宽度，单位微秒。驱动器有最小值（见它的说明书；MC-1 的闭环驱动器要 2.5）。
$1	步进空闲延时	毫秒	整数	≤ 65535	255	运动停下以后驱动器还使能多久，让轴先稳住再关电流。255 表示一直使能。
$2	步进脉冲反相		轴掩码		0	低电平有效的步进输出。
$3	步进方向反相		轴掩码		5	方向输出反相的轴（不然轴会走反）。
$4	驱动器使能输出反相		轴掩码		7	低电平有效的驱动器使能输出（大多数驱动器如此）。只有一个共用使能信号时只看 X。
$29	脉冲延时	微秒	小数	≤ 20	5.0	设好方向输出到发步进脉冲之间的时间。

设置	名字	单位	类型	范围	MC-1	说明
$37	一直使能的驱动器		轴掩码		7	机床空闲或报警时仍然使能的驱动器。
$680	驱动器使能的延时	ms	整数	≤ 500	0	使能驱动器到第一步之间的时间。

变频器

设置	名字	单位	类型	范围	MC-1	说明
$460	变频器的 Modbus 地址		整数	≤ 255	1	变频器的 Modbus 地址。

X 轴

设置	名字	单位	类型	范围	MC-1	说明
$100	X 轴每毫米步数	步/mm	小数		800.00000	这根轴每毫米的步数：电机每转的步数乘细分，再除以丝杠导程。
$110	X 轴最高速度	mm/min	小数		3000.000	这根轴的最高速度，快移也一样。
$120	X 轴加速度	mm/s²	小数		300.000	这根轴的加速度。
$130	X 轴最大行程	mm	小数		202.000	这根轴从回零位置算起的行程，给软限位用。
$180	X 轴回零的定位速度	mm/min	小数		50.0	精确找到这根轴开关动作点用的慢速。
$190	X 轴回零的搜索速度	mm/min	小数		1200.0	搜索这根轴的开关用的快速。
$1000	X 轴输入整形		选项		0	给这根轴的运动整形，免得激起机床在下面那个频率上的共振：ZV 正好消掉这个频率，其余几种消掉更宽的一段、但没那么准。运动最多晚 1.5 个周期停稳。 选项： 0 无, 1 ZV, 2 ZVD, 3 MZV, 4 EI, 5 2HUMP_EI
$1010	X 轴输入整形的频率	Hz	小数	15–500	100.0	这根轴的输入整形要消掉的共振频率。
$1020	X 轴输入整形的阻尼比		小数	≤ 0.5	0.100	这个共振的阻尼比。

Y 轴

设置	名字	单位	类型	范围	MC-1	说明
$101	Y 轴每毫米步数	步/mm	小数		800.00000	这根轴每毫米的步数：电机每转的步数乘细分，再除以丝杠导程。
$111	Y 轴最高速度	mm/min	小数		3000.000	这根轴的最高速度，快移也一样。
$121	Y 轴加速度	mm/s²	小数		300.000	这根轴的加速度。
$131	Y 轴最大行程	mm	小数		152.000	这根轴从回零位置算起的行程，给软限位用。
$181	Y 轴回零的定位速度	mm/min	小数		50.0	精确找到这根轴开关动作点用的慢速。
$191	Y 轴回零的搜索速度	mm/min	小数		1200.0	搜索这根轴的开关用的快速。
$1001	Y 轴输入整形		选项		0	给这根轴的运动整形，免得激起机床在下面那个频率上的共振：ZV 正好消掉这个频率，其余几种消掉更宽的一段、但没那么准。运动最多晚 1.5 个周期停稳。 选项： 0 无, 1 ZV, 2 ZVD, 3 MZV, 4 EI, 5 2HUMP_EI
$1011	Y 轴输入整形的频率	Hz	小数	15–500	100.0	这根轴的输入整形要消掉的共振频率。
$1021	Y 轴输入整形的阻尼比		小数	≤ 0.5	0.100	这个共振的阻尼比。

Z 轴

设置	名字	单位	类型	范围	MC-1	说明
$102	Z 轴每毫米步数	步/mm	小数		800.00000	这根轴每毫米的步数：电机每转的步数乘细分，再除以丝杠导程。

设置	名字	单位	类型	范围	MC-1	说明
$112	Z 轴最高速度	mm/min	小数		1500.000	这根轴的最高速度，快移也一样。
$122	Z 轴加速度	mm/s²	小数		200.000	这根轴的加速度。
$132	Z 轴最大行程	mm	小数		82.000	这根轴从回零位置算起的行程，给软限位用。
$182	Z 轴回零的定位速度	mm/min	小数		50.0	精确找到这根轴开关动作点用的慢速。
$192	Z 轴回零的搜索速度	mm/min	小数		600.0	搜索这根轴的开关用的快速。
$1002	Z 轴输入整形		选项		0	给这根轴的运动整形，免得激起机床在下面那个频率上的共振：ZV 正好消掉这个频率，其余几种消掉更宽的一段、但没那么准。运动最多晚 1.5 个周期停稳。 选项： 0 无, 1 ZV, 2 ZVD, 3 MZV, 4 EI, 5 2HUMP_EI
$1012	Z 轴输入整形的频率	Hz	小数	15–500	100.0	这根轴的输入整形要消掉的共振频率。
$1022	Z 轴输入整形的阻尼比		小数	≤ 0.5	0.100	这个共振的阻尼比。

报警码

控制器的报警。报警让机床停下并锁住，直到你解除它：按报警说的用 `$X` 或回零。`$EA` 在控制器上列出它们。

代码	意思
`ALARM:1`	硬限位：运动中碰到了限位开关。位置很可能已丢，请重新回零
`ALARM:2`	软限位：运动的目标在机床行程以外。位置没有丢，可以解锁
`ALARM:3`	运动中复位或急停。位置很可能已丢，请重新回零
`ALARM:4`	测头：探测开始时测头的状态不对
`ALARM:5`	测头：在编程的行程内没有碰到
`ALARM:6`	回零：回零过程被复位
`ALARM:7`	回零：回零时安全门被打开
`ALARM:8`	回零：退离以后限位开关仍然动作。请检查退离距离（$27）和接线
`ALARM:9`	回零：在搜索距离内没有找到限位开关。请检查行程设置和接线
`ALARM:10`	急停。松开急停，再复位
`ALARM:11`	需要回零：请执行回零（$H）
`ALARM:12`	有限位开关在动作。先让它复位再继续
`ALARM:13`	测头保护：不在探测中时测头被触发
`ALARM:14`	主轴没有按时到达转速
`ALARM:15`	回零：没有找到自动找正轴的第二个开关
`ALARM:16`	开机自检没有通过
`ALARM:17`	电机驱动器报告故障
`ALARM:18`	回零：回零的设置无效
`ALARM:19`	Modbus：设备（比如变频器）没有应答或应答出错
`ALARM:20`	I/O 扩展器没有应答
`ALARM:21`	非易失存储（存设置的存储器）出错
`ALARM:22`	缓冲区溢出

错误码

控制器拒绝一行时答的 `error:N`。这一行什么也没做。 `$EE` 在控制器上列出它们。

代码	意思
`error:1`	G 代码的字缺了字母：每个字都是一个字母加一个数
`error:2`	缺了数，或者数的写法不对
`error:3`	不认识或不支持的 '$' 命令
`error:4`	这里只能用正数，给的却是负数
`error:5`	设置里没有打开回零（$22）
`error:6`	步进脉冲至少要 2 微秒
`error:7`	存着的设置读不出来；受影响的设置已改回默认值
`error:8`	这条 '$' 命令只在机床空闲时接受
`error:9`	报警时或点动中不接受 G 代码
`error:10`	软限位要先能回零：请先打开回零（$22）
`error:11`	这一行超过了控制器能接受的长度，没有执行
`error:12`	这个值会让步进频率超过控制器能发出的上限
`error:13`	安全门开着
`error:14`	文字太长，存不下（开机行或构建信息）
`error:15`	点动的目标在机床行程以外，没有点动
`error:16`	点动命令要有 '='，而且只能有点动用的字
`error:17`	激光方式要有 PWM 主轴输出
`error:18`	控制器已复位
`error:19`	值必须大于零
`error:20`	这一段里有不支持或无效的 G 代码命令
`error:21`	一段里有同一模态组的两个命令
`error:22`	还没有设进给速度
`error:23`	这条命令要整数
`error:24`	这一段里有两个命令都要用轴字
`error:25`	这一段里同一个字出现了两次
`error:26`	这条命令要轴字，这一段里却没有
`error:27`	行号不对
`error:28`	缺了这条命令要的值字
`error:29`	不支持 G59.1、G59.2、G59.3
`error:30`	G53 只能和 G0、G1 一起用
`error:31`	这一段里有没有命令用的轴字
`error:32`	圆弧（G2/G3）在它的平面里至少要有一个轴字
`error:33`	运动的目标无效
`error:34`	圆弧的半径无效
`error:35`	圆弧（G2/G3）在它的平面里至少要有一个圆心偏移字（I、J 或 K）
`error:36`	这一段里有没有命令用的值字
`error:37`	G43.1 要用设定的刀长轴
`error:38`	刀号超出支持的范围，或者没有这把刀
`error:39`	值超出范围
`error:40`	等着换刀时不允许
`error:41`	这种运动要主轴转着（恒线速或主轴同步）
`error:42`	车螺纹要在 ZX 平面（G18）
`error:43`	进给速度超过了最大值
`error:44`	主轴转速超出范围
`error:45`	有限位开关在动作：只能回零

代码	意思
`error:46`	请先回零（$H）
`error:47`	不知道当前的刀：请用 M61 设
`error:48`	数值字相互冲突
`error:49`	开机自检没有通过：请把控制器关掉再开
`error:50`	急停按着
`error:51`	电机驱动器报告故障
`error:52`	设置值超出范围
`error:53`	这个控制器上没有这个设置
`error:54`	退刀位置低于钻孔深度
`error:55`	两根自动找正的轴不能一起回零
`error:56`	这个坐标系被锁住了（$486）
`error:57`	程序开头或结尾标记（%）出现得不对
`error:58`	这个端口不可用
`error:60`	SD 卡挂载不上
`error:62`	列不出这个目录
`error:63`	没有这个目录
`error:64`	SD 卡没有挂载
`error:65`	没有挂载文件系统（没插卡？）
`error:66`	文件系统是只读的
`error:71`	表达式：不认识的运算符或函数
`error:72`	表达式：除以零
`error:73`	表达式：参数超出范围
`error:74`	表达式：参数无效
`error:75`	表达式：语法错误
`error:76`	表达式：结果不是一个数
`error:77`	需要验证身份
`error:78`	没有权限
`error:79`	严重事件没有解除时不允许
`error:80`	这条 O 字语句要控制器能再读一遍的程序（从它的文件系统里跑）
`error:81`	不认识的 O 字语句，或位置不对
`error:82`	同时开着的 O 字块或调用太多
`error:83`	放不下这条 O 字语句了
`error:84`	打不开这个文件
`error:85`	文件系统格式化不了
`error:86`	这个端口用不了
`error:87`	主轴上有刀
`error:88`	主轴上没有刀
`error:89`	删不掉这个文件
`error:253`	用户定义的错误

这一页是从程序生成的接口参考，只有英文版。

Command line: mc1sim

The options of the twin's program `mc1sim`, as its usage text gives them (86 options).

```
usage: mc1sim [options] scenario.nc ...     (batch, deterministic)
       mc1sim [--tcp PORT] [--ws PORT] [options]  (real time: TCP sender, browser UI)
       mc1sim --replay FILE [options]         (replay a recorded real-time session)
       mc1sim --play FILE.mcap [--ws PORT | --gui] (play a --mcap recording in the UI, with a timeline)
       mc1sim                                 (single-file build: the machine in its own window,
                                               a sender port on 2323 and the web view on 8765)
```

option	what it does
`--machine FILE`	machine archive (default: built in, or twin/machine/mc1.machine.json)
`--max-time S`	stop after S seconds of virtual time (default 3600 in batch, none in real time)
`--fg-quantum US`	longest foreground wait between resumes (default 50)
`--link KIND`	the sender's line: usb_cdc (the Teensy's native USB), uart (through a USB-serial adapter) or telnet (Ethernet); default the board's (machine archive controller.board)
`--baud N`	a UART at N baud (implies --link uart)
`--link-param K=V`	change a number of the link, e.g. adapter_timer_ms=1 or host_in_latency_us=0; host_dtr=0: the twin itself does not hold DTR (the native USB answers only once a host raises it), connect_dtr=0: a TCP sender does not raise it either
`--fg-cost off|L,P,S`	the foreground's CPU time in us: a line parsed, a block planned, a step segment prepared (default the board's; off: none)
`--pos X,Y,Z`	power-on position, machine mm (default from the archive)
`--sw-delay US`	limit sensors' response time (default from the archive)
`--tool-offset MM`	tool tip below the reference tool (default 0)
`--drive L0|L1|L2|L3 feed drives: L0 ideal steps, L1 closed-loop steppers moving rigid axes,`	L2 rotor, screw and table apart (coupling, torsion, Hertzian nut, play, ringing), L3 the motors' two phases and the drives' current loops on rigid axes
`--l3-dyno [V]`	the L3 motor on a dynamometer: torque against speed, as JSON (V: another supply)
`--lobes SPEC`	a side-milling cut's stability lobes by three methods (zero-order, linear time domain, semi-discretisation), JSON: "tool=flat 6 n3 L20;mat=6061-T6;ae=1;ap=1;dir=x; rpm=6000:24000:500;m=100;methods=zoa,td,sdm"
`--chatter-sim SPEC`	the cut in time, nonlinear (each tooth out of the cut where its chip is gone): the vibration and whether it chatters, JSON: "...;ap=2;rpm=15000;fz=0.02;t=0.5" (default: the machine archive's drives.model)
`--geom on|off`	the G- and T-layer: geometric errors, lead error, backlash, thermal offsets (machine archive "accuracy"; default off)
`--fidelity batch|live`	what feeds back into the simulation (docs/06 section 8.3): batch - the full resolution, as tests and the AI's practice machines have it; live - what a virtual machine at 1x keeps up with (the cut's path merged to an eighth of a cell). Default live with the window, --ws or --tcp and no --lockstep, else batch; a replay or a branch takes its recording's
`--threads N`	threads for the cut's sweep (the results are the same with any number): 1 this one alone; default the processors less one, at most 8 (or TWINWRIGHT_THREADS)
`--chatter-async on|off`	the chatter check on a thread of its own, below the simulation's: its answers taken in the order the cuts came - the same as at once, later (the summary waits for them; the views show the tool's last one meanwhile). Default on where the machine keeps the wall clock's pace (the window, --ws, --tcp; not --lockstep)
`--chatter-cache FILE|off`	the chatter check's time-domain depths kept in FILE - the same results, the second run faster. Default: the state folder's chatter-depths.bin for the window and the MCP's practice machines; off elsewhere (TWINWRIGHT_CHATTER_CACHE too)
`--set PATH=VALUE`	change a machine archive entry as it is read, e.g. drives.motor.detent_nm=0.01 or drives.alm.wired=false (repeatable)
`--mpg N`	the handwheel UART as if the firmware were built with MPG_ENABLE N (2: the handwheel's own 0x8B takes control; 0: none) - default as the archive's firmware options say (a replay: as its log says)
`--mpg-port PORT`	an external handwheel on the handwheel UART's cable, over TCP (with --mpg 2): its bytes reach the controller at the handwheel's baud rate (logged), the controller's come back
`--panel SPEC`	how the operator's buttons are wired: cycle_start=pin|sender|mpg, feed_hold=... (pin: the board's input, Pn: and its debounce; sender: '~' '!' from the sender; mpg: on the handwheel's stream) - default as the archive's controller.panel, else the sender's (a replay: as its log says)
`--fault WIRE=MODE`	wiring fault from power-on: x_min x_max ... z_switch setter alm estop estop_km1 rs485 (the VFD's bus) vfd_relay (its fault relay in the ALM chain) = open|short|intermittent (wire names follow the machine archive)

option	what it does
`--seed N`	sensors with their scatter (repeatability, response jitter) drawn from seed N; without: nominal sensors. Intermittent wires use it too
`--nvs FILE`	load EEPROM image; --save-nvs FILE writes it at exit. Batch runs start blank; real-time and replay runs from the MC-1 settings image (built in, or build/mc1.nvs made by make) unless --nvs FILE or --nvs blank
`--world DIR`	the persistent world (docs/06 8.5): the machine as the last session left it - where the axes stood, the workpiece clamped and cut, the tool, the EEPROM (written through to DIR/eeprom.bin), the temperatures and the board's clock moved on by the time it was off, the odometers; saved there at the end. The window keeps one in its state folder; batch runs and replays only with this
`--world-reset`	start the world from the factory state (its files removed first)
`--world-off S`	as if the machine had been off S seconds (else the wall clock's time since)
`--predict FILE`	with --shadow: the program the operator streams; each run of it the real machine starts is predicted by a background twin set up like the machine (homed, the workpiece, tool and offsets), flat out - its own run time against the machine's
`--nvs-through FILE`	the controller's EEPROM kept in FILE, as the board's flash keeps it: loaded from FILE if it exists (else from the image above, and FILE made), every write landing in FILE at once - settings, offsets and tools carry over to the next session. Not in replays (a recording has its start image beside it: LOG.nvs)
`--params-trace FILE the machine archive's values the run used (JSON): where each sits, its`	provenance (source, uncertainty), what it feeds, whether --set changed it
`--steps FILE`	write every step pulse (tick step_bits dir_pins)
`--spindle-trace FILE`	each millisecond the spindle turns: seconds, its angle (revolutions) and where the steps put the axes (machine mm) - a thread's pitch and phase (src/spenc.h)
`--hil-log FILE`	write what crosses between the controller and the machine as the HIL emulator's frames (src/hil.h): step groups, enables, coolant, RS-485, the machine's answers
`--hil-replay FILE`	the machine driven by such a log, the controller off (HIL's follower)
`--wav FILE`	the run's sound as the machine makes it (src/sound.h): 48 kHz stereo, 32-bit float; the same run gives the same file
`--sound-player FILE,FACTOR`	the window's player of the sound at speed FACTOR, without a device: the live frames played in this thread into a WAV (tests: the pitch at any factor)
`--sound-analyse FILE[,DBFS_AT_94[,T0,T1]]`	a WAV measured as a sound level meter reads it: LZeq, LAeq, 1/3-octave bands, the strongest lines (JSON; default -20: 94 dB SPL is -20 dBFS rms)
`--sound-listener X,Y,Z`	where --wav and --sound-player listen: model mm at the GLB pose (the machine archive's "sound"), facing the tool; default the archive's operator in front
`--sound-room off|small|workshop`	the room --wav and --sound-player hear it in (RT60 0.35 s, 0.7 s; default off: the dry sound)
`--events FILE`	write inputs, output lines and markers with ticks
`--echo`	print controller output with virtual time
`--factor F`	real-time factor (default 1)
`--www DIR`	serve the UI from DIR (default: built in, or micro_cnc/)
`--bind local|any`	listen on localhost only (the default) or on every interface (in shadow mode the sender port passes to the real machine: any lets the network move it; a twin in WSL reached from Windows needs any)
`--open`	open the web UI in the default browser
`--gui`	the native window (Windows; the default when started without arguments)
`--lang TAG`	the language (en, zh-CN, a pack's tag; default: the settings, else the system's; qps-ploc: the pseudo-language that shows untranslated texts and tight layouts)
`--lang-dir DIR`	one more folder with language packs (.po); see twin/lang/README.md
`--units U`	metric | imperial | a units.json system; --region auto|lang|de-DE; --currency EUR
`--lang-list`	the languages found; --lang-check FILE.po checks a pack; --lang-template FILE.pot writes the texts to translate
`--warp`	draw the 3D view in software (WARP) even with a graphics card
`--selftest DIR`	window self-test: off-screen, homes and runs the demo program at --factor 5, pictures and selftest.json into DIR, exit status 1 on failure
`--selftest-setup DIR`	the same for setting up workpieces (panel, dragging, keys, clamps, vise)
`--record FILE`	log real-time input for --replay
`--mcap FILE`	record signals, events, serial lines and UI frames (PlotJuggler, Foxglove)
`--vcd FILE`	record the controller's pins and the machine's lines (GTKWave, PulseView)
`--rec-period MS`	sampling period of the recordings (default 1)

option	what it does
`--stock x0,x1,y0,y1,ztop,zbot[,res]`	a block held as if bolted down (machine mm; cells default 0.02 mm)
`--setup JSON|@FILE`	the workpiece set up as on the machine: shape, size, material, fixture (clamps, vise, vblock, tape, none, fixed), position; e.g. '{"shape":"block","size":[60,40,20], "material":"6061-T6","fixture":"vise","at":[100,75]}' (see twin/core/README)
`--tool 'flat 6 n2'`	the cutter: flat|ball|bull|v|drill D [Rc|angle] [nN] (default from the archive)
`--part FILE.stl[@X,Y,Z]`	the part the program should make, its origin at X,Y,Z machine mm (default 0,0,0)
`--stock-stl FILE[,RES[,FRAME]]`	write the workpiece as a binary STL at the end (cells RES, default 0.1; FRAME machine (default), program: the program's zero, blank: its own corner)
`--stock-png FILE[,RES]`	write its height map, or the deviation from --part, as a PNG
`--branch S`	with --replay and a real-time option: replay the log to S seconds, then carry on live
`--mtconnect PORT`	MTConnect adapter (SHDR) for an agent on PORT (7878 is usual)
`--lockstep PORT`	a script drives the twin over localhost PORT (twin/py/mc1twin): virtual time moves only when it says, as fast as the host allows (api.h has the requests); 0: any free port, printed on standard output as lockstep-port N
`--api-token T`	the lockstep client's first request must be hello with token T
`--exit-with-client`	the session ends when its lockstep client goes (or none came within 30 s)
`--control PORT`	the control port (AI access, as the window's AI switch opens it): the lockstep API's requests from several local clients while time runs as usual; 0 any port
`--control-file F`	write its port and token to F as JSON (the MCP server reads it)
`--workdir DIR`	programs' versions here (default Documents/Twinwright Studio/programs; TWINWRIGHT_WORKDIR)
`--control-token T`	its token (default: a random one)
`--control-confirm yes|no|ask`	how its confirmations are answered (default yes: no window to ask)
`--own-call JSON`	{"tool":..., "args":{...}}: a tool called from the window's own interface (its own channel, the AI switch off or on), printed as "own-call N {...}" (tests; up to 12)
`--mcp`	the MCP server for AI clients on standard input and output (they start it; see twin/docs/05 section 9): background twins, and the window when its AI switch is on
`--mcp-http PORT`	the same over Streamable HTTP at http://127.0.0.1:PORT/mcp
`--ai`	(Windows) the window with its AI switch on (it remembers the switch anyway)
`--mtconnect-devices FILE`	write the matching MTConnect device model (Devices.xml; - for stdout)
`--fmi-model-description FILE`	write the FMI 2.0 modelDescription.xml of the lockstep API's FMI-shaped interface (fmi_* requests; twin/py/mc1twin/fmi.py; - for stdout)
`--shadow HOST[:PORT]|COMn|/dev/ttyACM0`	shadow mode: the sender port relays to the real machine's controller (Telnet, port 23, or its USB serial port); the twin runs the same lines alongside and compares answers, path, settings and run times (use --factor 1, the default)
`--shadow-guard`	with --shadow: each line of the sender's is looked at before it reaches the machine; a predicted crash holds it back and feed-holds the machine (the views release it)
`--version`	the program's version and the kind of build (public, internal; a public build with the test key says so)
`--licence-status [DIR] [--json] | --licence-request [DIR] | --licence-import FILE [DIR] | --licence-canon FILE | --licence-keys`	the licence files (docs/14): their state; this computer's request code for offline activation; a licence file into the folder (DIR: the machine's licence folder unless given); a file's signed bytes; the keys this build takes
`--licence-activate CODE [DIR] [--json] | --licence-trial maker|pro [DIR] [--json] | --licence-renew [DIR] [--json] | --licence-renew-due [DIR] | --licence-code CODE | --licence-server`	online (product/23, product/30): activate this computer with an activation code; ask the licence server for this computer's 30-day trial of an edition; renew the licence that is due, now; whether the window's start would renew (a day since the last try); a typed code in its canonical form; the server this build talks to (the last two send nothing)

这一页是从程序生成的接口参考，只有英文版。

Machine profile schema: MC-1

Every key of the machine archive the twin is built from (24 sections), with the value the MC-1's archive gives - for the twin's internal error model (`accuracy`, the coupling's play, and where their values come from) the keys only. The values are the model's parameters - typical values until a machine is measured - not specifications of the machine and not its accuracy; every number the twin gives about the physical world is an estimate.

`schema`

the archive format and its version

key	type	unit	value	what it is
`schema`	string		"mc1-machine/0.1"	

`generated`

what made this archive, from which inputs (hashes of params.py and wiring.py)

key	type	unit	value	what it is
`generated.by`	string		"twin/gen/gen_machine.py"	
`generated.params_sha256`	string		"ac86b6d281b81f5ca58fa7acf69c2237bee4a1a143490b3b555ad911…	
`generated.wiring_sha256`	string		"b4a817153a66f8458fe8ef3bd563b7ac0f6174733b8757570188bcf9…	
`generated.report_json`	object			the design checks' report the archive was made from: its path and when it was written
`generated.report_json.path`	string		"out/report.json"	
`generated.report_json.modified`	string		"2026-10-13T05:04:23"	

`units`

the base units

key	type	unit	value	what it is
`units.length`	string		"mm"	
`units.mass`	string		"kg"	
`units.time`	string		"s"	

`name`

key	type	unit	value	what it is
`name`	string		"MC-1"	

`coordinates`

the machine coordinate frame in words (MPos = machine - home offset)

key	type	unit	value	what it is
`coordinates`	string		"machine X 0..200 left to right, Y 0..150 tool relative t…	

`power_on_pos`

where the axes stand at power-on, machine mm (X, Y, Z)

`axes`

the three linear axes, in the order X, Y, Z

key	type	unit	value	what it is
`axes[].id`	string		"X"	
`axes[].moves`	string		"X carriage with the Z axis and spindle"	

key	type	unit	value	what it is
axes[].travel	array (2)		[0.0, 200.0]	the axis' travel [min, max], machine mm
axes[].home_dir	string		"+"	the direction homing seeks its switch
axes[].steps_per_mm	number	per mm	800.0	steps per mm (the controller's $100-$102)
axes[].dir_sign	number		-1	+1: the motor's positive direction is the axis' positive direction
axes[].direction	object			which way the motor turns the screw and the axis moves: for the drive model and the wiring's polarity
axes[].direction.motor_end	string		"+"	
axes[].direction.screw_hand	string		"right"	
axes[].direction.dir_low_turns	string		"cw (seen from the shaft end; assumed)"	
axes[].direction.machine_per_model	number		1	
axes[].direction.dir_low_moves	number		-1	
axes[].switch	object			the axis' limit / home switches: position, contact, sensor physics
axes[].switch.min	number		-2.0	
axes[].switch.max	number		202.0	
axes[].switch.contact	string		"NC"	
axes[].switch.inputs	string		"separate"	
axes[].switch.type	string		"M8 inductive NPN NC, Sn 2.0 mm"	
axes[].switch.sn_mm	number	mm	2.0	
axes[].switch.gap_design_mm	number	mm	1.0	
axes[].switch.gap_mm	array (2)	mm	[1.0, 1.0]	
axes[].switch.spot_r_cad_mm	number	mm	3.0	
axes[].switch.anchor	string		"cad"	
axes[].switch.target	array (2)		["steel", "steel"]	
axes[].switch.unit_sr	array (2)		[1.0, 1.0]	
axes[].switch.lambda_mm	number	mm	0.7	---------- limit sensors and wiring The inductive sensors' switching model follows research note F (1.3, 1.4, 7): the flag sweeps across the sensing face at gap g; the damping of the oscillator is D(g, e) = exp(-g / lambda) * F(e / w) with a logistic coverage F, and the sensor switches when it reaches the damping of a target at distance Sr straight in front: F(e_on / w) = exp(-(Sr - g) / lambda). e_on is the flag edge's position relative to the sensor axis. Sr = Sn * material factor * unit spread * (1 + tc * (T - 20 C)). The CAD places the sensors for an idealised sensing spot of radius SENSE_R (e_on = -SENSE_R): "anchor": "cad" keeps the designed switch points for the nominal sensor and moves them by what the model predicts for other gaps, materials, units and temperatures; "anchor": "model" uses the model's switch point outright (about 1.9 mm later: the design review's sensitivity case). All values unverified until measured (F 7).
axes[].switch.w_mm	number	mm	1.0	
axes[].switch.hyst_mm	number	mm	0.15	
axes[].switch.response_ms	number	ms	1.0	
axes[].switch.jitter_ms	number	ms	0.25	
axes[].switch.repeat_sigma_mm	number	mm	0.003	
axes[].switch.temp_c	number	°C	20.0	
axes[].switch.temp_coeff_per_k	number	1/K	0.0	
axes[].switch.ready_ms	number	ms	30.0	IEC 60947-5-2 "time delay before availability": after its supply comes up the output stays off (F 5: 30 ms); the controller boots later (controller.boot_ms), so it never sees it (unverified)
axes[].hard_stop	array (2)		[-8.02, 209.02]	where the moving part hits the frame [min, max], machine mm

key	type	unit	value	what it is
`axes[].moving_mass_kg`	number	kg	13.9	

`sensor_materials`

inductive sensors: switching distance reduction factor by target material

key	type	unit	value	what it is
`sensor_materials.steel`	number		1.0	
`sensor_materials.stainless`	number		0.7	
`sensor_materials.brass`	number		0.45	
`sensor_materials.aluminium`	number		0.4	
`sensor_materials.copper`	number		0.35	

`wires`

the field wiring by signal: the controller input, cable, route and length

key	type	unit	value	what it is
`wires.x_min`	object			one signal's wire: the controller input, the cable, its route and length, and the cable chains it bends in (flex)
`wires.x_min.input`	string		"X_LIMIT"	
`wires.x_min.cable`	string		"W09"	
`wires.x_min.route`	string		"固定，横梁后侧"	
`wires.x_min.length_mm`	number	mm	728.0	
`wires.x_min.flex`	string		""	
`wires.x_max`	object			
`wires.x_max.input`	string		"X_LIMIT_MAX"	
`wires.x_max.cable`	string		"W10"	
`wires.x_max.route`	string		"固定，横梁后侧"	
`wires.x_max.length_mm`	number	mm	488.0	
`wires.x_max.flex`	string		""	
`wires.y_min`	object			
`wires.y_min.input`	string		"Y_LIMIT"	
`wires.y_min.cable`	string		"W12"	
`wires.y_min.route`	string		"固定，底板下方"	
`wires.y_min.length_mm`	number	mm	167.0	
`wires.y_min.flex`	string		""	
`wires.y_max`	object			
`wires.y_max.input`	string		"Y_LIMIT_MAX"	
`wires.y_max.cable`	string		"W11"	
`wires.y_max.route`	string		"固定，底板下方"	
`wires.y_max.length_mm`	number	mm	212.0	
`wires.y_max.flex`	string		""	
`wires.z_switch`	object			
`wires.z_switch.input`	string		"Z_LIMIT"	
`wires.z_switch.cable`	string		"W08"	
`wires.z_switch.route`	string		"Z 随动线束 + X 拖链，与主轴线分开绑扎"	
`wires.z_switch.length_mm`	number	mm	1493.0	
`wires.z_switch.flex`	string		"XZ"	
`wires.setter`	object			
`wires.setter.input`	string		"PROBE"	

key	type	unit	value	what it is
wires.setter.cable	string		"W13"	
wires.setter.route	string		"Y 拖链"	
wires.setter.length_mm	number	mm	1267.0	
wires.setter.flex	string		"Y"	
wires.alm	object			
wires.alm.input	string		"MOTOR_FAULT"	
wires.alm.cable	string		""	
wires.alm.route	string		"电控箱内：三台驱动器 ALM 与变频器故障继电器串联"	
wires.alm.length_mm	number	mm	0.0	
wires.alm.flex	string		""	
wires.estop	object			
wires.estop.input	string		"E-STOP"	
wires.estop.cable	string		""	
wires.estop.route	string		"电控箱：急停 SB0 第二组常闭（SB0-2）"	
wires.estop.length_mm	number	mm	0.0	
wires.estop.flex	string		""	
wires.estop_km1	object			
wires.estop_km1.input	string		""	
wires.estop_km1.cable	string		""	
wires.estop_km1.route	string		"电控箱：急停 SB0 第一组常闭（SB0-1），在 KM1 线圈的自锁回路里"	
wires.estop_km1.length_mm	number	mm	0.0	
wires.estop_km1.flex	string		""	
wires.rs485	object			
wires.rs485.input	string		"RS485"	
wires.rs485.cable	string		""	
wires.rs485.route	string		"电控箱内：控制板 RS485 A/B → 变频器 RS+/RS-，屏蔽双绞线"	
wires.rs485.length_mm	number	mm	0.0	
wires.rs485.flex	string		""	
wires.vfd_relay	object			
wires.vfd_relay.input	string		"MOTOR_FAULT"	
wires.vfd_relay.cable	string		""	
wires.vfd_relay.route	string		"电控箱内：变频器故障继电器的常闭触点 TB–TC，串在 ALM 链里（发现 107）"	
wires.vfd_relay.length_mm	number	mm	0.0	
wires.vfd_relay.flex	string		""	

faults

wiring fault models (glitches, intermittent contacts)

key	type	unit	value	what it is
faults.glitch_us	number	µs	50.0	injected faults: a glitch opens the contact this long; an intermittent wire drops out while its cable flexes (per metre of travel of the axes it follows), for a random time
faults.intermittent	object			an intermittent wire: it drops out while its cable flexes, so often per metre of travel of the axes it follows, each time for a random time in the range (ms)
faults.intermittent.per_m	number	per m	5.0	
faults.intermittent.ms	array (2)		[0.2, 3.0]	

`drives`

the feed drives: ball screws, closed-loop steppers, their power (KM1), the L1/L2 models

key	type	unit	value	what it is
`drives.model`	string		"L1"	
`drives.note`	string		"rigid axis driven by a closed-loop stepper: magnetic spr…	
`drives.lead_mm`	number	mm	4.0	
`drives.motor`	object			the stepper motor: holding torque, rotor inertia, steps, detent torque, phase resistance and inductance, torque constant, rated current, iron loss
`drives.motor.type`	string		"NEMA23 closed-loop hybrid stepper, 23HS22-2804D-E1000 cl…	
`drives.motor.holding_nm`	number	N·m	1.26	
`drives.motor.rotor_inertia_kgm2`	number	kg·m²	3e-05	
`drives.motor.full_steps_per_rev`	number	per revolution	200	
`drives.motor.detent_nm`	number	N·m	0.05	
`drives.motor.power_up_jump`	boolean		true	the drive's first current after a power-up (its phase reset): the rotor snaps to the nearest stable position of that state, up to two full steps either way (architecture 4.4; unverified on the drive)
`drives.motor.phase_resistance_ohm`	number	Ω	0.9	L3 (research note D, L3): the phases - R and L of the 2.8 A class (typical catalogue values, unverified), the torque constant fitted to the 48 V pull-out plateau (0.8 N m at 2.8 A; the 1.26 N m holding torque is both phases at rated current), the detent's amplitude as above
`drives.motor.phase_inductance_mh`	number	mH	2.9	
`drives.motor.torque_constant_nm_a`	number	N·m/A	0.286	
`drives.motor.rated_current_a`	number	A	2.8	
`drives.motor.iron_loss_nm_per_krpm`	number	N·m per 1000 rpm	0.0377	the laminations' hysteresis and eddy losses as a drag growing with speed: fitted so that L3 on its dynamometer gives the 48 V pull-out curve above (9 W at 1500 rpm; to be measured, architecture 7.3 step 8)
`drives.coupling_inertia_kgm2`	number	kg·m²	2.6e-06	
`drives.pullout_48v_rpm_nm`	array (9)	rpm/(N·m)		
`drives.drive`	object			the closed-loop drive: encoder counts, holding current, damping, the following error it alarms at and how that alarm is reset, current loop, DC bus, the L3 sub-steps
`drives.drive.type`	string		"CL57T class closed-loop drive, 48 V"	
`drives.drive.loop`	string		"closed"	
`drives.drive.encoder_counts_per_rev`	number	per revolution	4000	
`drives.drive.holding_current`	number		0.5	
`drives.drive.damping_ratio`	number		0.5	
`drives.drive.smoothing_ms`	number	ms	0.0	
`drives.drive.following_error_limit_counts`	number		400	
`drives.drive.alarm_reset`	string		"power"	
`drives.drive.open_loop_below_rpm`	number	rpm	0	
`drives.drive.current_loop_hz`	number	Hz	2000.0	then only the holding current at standstill (0: always closed) L3: its current loop's bandwidth, its bus capacitance and overvoltage alarm (assumed), 5 us steps
`drives.drive.bus_capacitance_uf`	number	µF	470.0	
`drives.drive.overvoltage_v`	number	V	60.0	
`drives.drive.l3_sub_steps`	number		20	

key	type	unit	value	what it is
drives.drive.max_phase_advance_deg	number	°	0.0	
drives.alm	object			the alarm chain: the drives' ALM outputs and the spindle drive's fault relay in series to the controller's motor-fault input
drives.alm.wired	boolean		true	
drives.alm.vfd_relay	boolean		true	
drives.alm.chain	string		"the three ALM outputs and the VFD's fault relay (its mul…	
drives.power	object			the drives' supply: the 48 V supply behind the contactor KM1, its timing, the supply's hold-up and rise, the drives' boot
drives.power.source	string		"PSU1 48 V behind KM1 (drives, VFD); the 24 V supply befo…	
drives.power.km1	string		"24 V DC coil, latched by the START button SB1 through it…	
drives.power.km1_latched	boolean		true	
drives.power.km1_drop_ms	number	ms	15	
drives.power.km1_pickup_ms	number	ms	25	
drives.power.psu_holdup_ms	number	ms	40	
drives.power.psu_rise_ms	number	ms	150	
drives.power.drive_boot_ms	number	ms	800	
drives.step_s	number	s	0.0001	
drives.l2	object			L2 (--drive L2): the axial chain between the motor side and the table - stiffness by the nut's distance from its BK bearing (the design's hand calculation at 100 N), the nut's play (the accuracy section's backlash), modal damping assumed; integrated in sub-steps of step_s. The drive's current loop (lag) and velocity loop (damping of the motor side on its magnetic spring) are assumed: a closed-loop drive is tuned not to overshoot - fit both to encoder logs (architecture 7.3 step 1) The nut's play closes with an impact: the grease film makes it strongly damped at a few mm/s (Hunt-Crossley damping, growing with the penetration; restitution assumed)
drives.l2.zeta_axial	number		0.03	
drives.l2.sub_steps	number		5	
drives.l2.current_lag_ms	number	ms	0.2	
drives.l2.servo_damping_ratio	number		0.7	
drives.l2.smoothing_ms	number	ms	1.0	
drives.l2.contact_restitution	number		0.0	
drives.l2.impact_speed_mm_s	number	mm/s	5.0	
drives.l2.masses	number		3	three masses (note D, L2): the coupling's mode sits near 1.75 kHz - 20 sub-steps of 5 us; its damping assumed; the friction model of guides and nut: "karnopp" (Stribeck + viscous with a stick band) or "lugre" (pre-sliding, friction lag at reversals)
drives.l2.sub_steps_3	number		20	
drives.l2.zeta_coupling	number		0.05	
drives.l2.friction	string		"karnopp"	
drives.l2.stribeck_ratio	number		1.25	
drives.l2.stribeck_v_mm_s	number	mm/s	2.0	
drives.l2.viscous_n_s_mm	number	N·s/mm	0.02	
drives.l2.lugre_sigma0_n_um	number	N/µm	10.0	
drives.axes	array (3)			per axis (X, Y, Z): screw, inertias, masses, friction, the L2 elastic model
drives.axes[].id	string		"X"	
drives.axes[].screw_length_mm	number	mm	319.0	

key	type	unit	value	what it is
drives.axes[].screw_inertia_kgm2	number	kg·m²	3.2918581767123845e-06	
drives.axes[].rotating_inertia_kgm2	number	kg·m²	3.589185817671239e-05	
drives.axes[].moving_mass_kg	number	kg	13.9	
drives.axes[].equivalent_mass_kg	number	kg	102.45961035603891	
drives.axes[].gravity_n	number	N	0.0	
drives.axes[].friction_n	number	N	91.62684499333461	
drives.axes[].l2	object			the axis in the L2 drive model (--drive L2): the masses, the axial and torsional stiffness by the nut's distance from the fixed bearing, friction, the coupling, the frame under the bearing
drives.axes[].l2.rotating_mass_kg	number	kg	88.5596103560389	
drives.axes[].l2.table_mass_kg	number	kg	13.9	
drives.axes[].l2.k_axial_n_um	array (2)	N/µm		
drives.axes[].l2.friction_motor_n	number	N	81.68140899333461	
drives.axes[].l2.friction_table_n	number	N	9.945436	
drives.axes[].l2.motor_mass_kg	number	kg	77.22965443852422	three masses: the rotor with half the coupling, the screw with the other half, the table
drives.axes[].l2.screw_mass_kg	number	kg	11.329955917514674	
drives.axes[].l2.coupling_n_um	number	N/µm	3817.0695021213087	
drives.axes[].l2.torsion_gj_nmm2	number	N·mm²	64770810.92360646	the screw's torsion from the coupling to the nut: its root section's G J over the BK journal, the drive journal (as lengths of root section) and the distance BK -> nut
drives.axes[].l2.torsion_extra_mm	number	mm	53.184712792968746	
drives.axes[].l2.k_rest_n_um	array (2)	N/µm		the axial chain without the nut (shaft, BK, mount), by the nut's distance from the BK
drives.axes[].l2.nut_hertz_n_um15	number	N/µm^1.5	79.28988773038022	
drives.axes[].l2.frame	object			the frame under the axis' fixed bearing as one mode along the axis: stiffness, mass, damping
drives.axes[].l2.frame.k_n_um	number	N/µm	33.2	
drives.axes[].l2.frame.m_kg	number	kg	17.65	
drives.axes[].l2.frame.zeta	number		0.02	

drives.axes[].l2.coupling_play_um

The coupling's play, part of the twin's internal error model like the backlash: its value is not published.

key	type	unit	what it is
drives.axes[].l2.coupling_play_um	number	µm	

accuracy

geometric and thermal errors of the tool relative to the work (Phase 4)

The twin's internal error-model parameters, off by default; their values are not published.

key	type	unit	what it is
accuracy.enabled	boolean		
accuracy.note	string		
accuracy.units	string		
accuracy.sign	string		
accuracy.axes	object		
accuracy.axes.X	object		
accuracy.axes.X.lead_um	array (45)	µm	
accuracy.axes.X.periodic_um	number	µm	

key	type	unit	what it is
accuracy.axes.X.periodic_mm	number	mm	
accuracy.axes.X.periodic_phase	number		
accuracy.axes.X.straight_um	object	µm	
accuracy.axes.X.straight_um.Y	array (45)		
accuracy.axes.X.straight_um.Z	array (45)		
accuracy.axes.X.angle_urad	object	µrad	
accuracy.axes.X.angle_urad.A	array (45)		
accuracy.axes.X.angle_urad.B	array (45)		
accuracy.axes.X.angle_urad.C	array (45)		
accuracy.axes.X.backlash_um	number	µm	
accuracy.axes.X.gravity_loaded	boolean		
accuracy.axes.Y	object		
accuracy.axes.Y.lead_um	array (35)	µm	
accuracy.axes.Y.periodic_um	number	µm	
accuracy.axes.Y.periodic_mm	number	mm	
accuracy.axes.Y.periodic_phase	number		
accuracy.axes.Y.straight_um	object	µm	
accuracy.axes.Y.straight_um.X	array (35)		
accuracy.axes.Y.straight_um.Z	array (35)		
accuracy.axes.Y.angle_urad	object	µrad	
accuracy.axes.Y.angle_urad.A	array (35)		
accuracy.axes.Y.angle_urad.B	array (35)		
accuracy.axes.Y.angle_urad.C	array (35)		
accuracy.axes.Y.backlash_um	number	µm	
accuracy.axes.Y.gravity_loaded	boolean		
accuracy.axes.Z	object		
accuracy.axes.Z.lead_um	array (21)	µm	
accuracy.axes.Z.periodic_um	number	µm	
accuracy.axes.Z.periodic_mm	number	mm	
accuracy.axes.Z.periodic_phase	number		
accuracy.axes.Z.straight_um	object	µm	
accuracy.axes.Z.straight_um.X	array (21)		
accuracy.axes.Z.straight_um.Y	array (21)		
accuracy.axes.Z.angle_urad	object	µrad	
accuracy.axes.Z.angle_urad.A	array (21)		
accuracy.axes.Z.angle_urad.B	array (21)		
accuracy.axes.Z.angle_urad.C	array (21)		

key	type	unit	what it is
accuracy.axes.Z.backlash_um	number	µm	
accuracy.axes.Z.gravity_loaded	boolean		
accuracy.square_urad	object	µrad	
accuracy.square_urad.xy	number		
accuracy.square_urad.xz	number		
accuracy.square_urad.yz	number		
accuracy.abbe	object		Abbe arms: where each axis' angular errors are referred to (params.py geometry, model frame: y towards the beam, z up from the base top; the reference tool tip is at z = tip_z0 + Z)
accuracy.abbe.tip_z0	number		
accuracy.abbe.X	object		
accuracy.abbe.X.y	number		
accuracy.abbe.X.z	number		
accuracy.abbe.Z	object		
accuracy.abbe.Z.y	number		
accuracy.abbe.Z.dz	number		
accuracy.abbe.Y	object		
accuracy.abbe.Y.x	number		
accuracy.abbe.Y.y	number		
accuracy.abbe.Y.z	number		
accuracy.thermal	object		
accuracy.thermal.alpha_um_m_k	number	µm/(m·K)	
accuracy.thermal.screw_dt_k	array (3)	K	
accuracy.thermal.bk_at	array (3)		
accuracy.thermal.spindle_growth_um	number	µm	
accuracy.thermal.dynamic	object		first-order dynamics (research note D 8 T-layer, "second version"): each screw warmed by its nut and BK drag times its speed through a thermal resistance, the spindle grown by its bearings (by speed) and its motor's copper (by current) - time constants and gains assumed, to be fitted to a 2 h warm-up log (architecture 7.3 step 14); off unless asked for (the static offsets above, and %thermal, stay as set)
accuracy.thermal.dynamic.enabled	boolean		
accuracy.thermal.dynamic.screw_k_per_w	number	K/W	
accuracy.thermal.dynamic.screw_tau_s	number	s	
accuracy.thermal.dynamic.spindle_um_per_w	number	µm/W	
accuracy.thermal.dynamic.spindle_tau_s	number	s	
accuracy.thermal.dynamic.spindle_bearing_w_at_24000	number	W	
accuracy.thermal.dynamic.spindle_copper_w_rated	number	W	
accuracy.thermal.frame	object		
accuracy.thermal.frame.alpha_um_m_k	number	µm/(m·K)	
accuracy.thermal.frame.dt_k	number	K	
accuracy.thermal.frame.tool_to_table_um_k	number	µm/K	
accuracy.thermal.frame.capacity_j_k	number	J/K	
accuracy.thermal.frame.g_w_k	number	W/K	
accuracy.thermal.frame.motors_w	number	W	
accuracy.thermal.frame.spindle_share	number		
accuracy.thermal.ambient_dt_k	number	K	the room above the reference (20 C): the frame follows it (at once without the dynamics)
accuracy.thermal.part	object		the workpiece (research note L 8): 10-25 % of the cutting power goes into it (0.15 taken), out through the fixture - 0.2 K/W with good contact (vise, clamps), 1 K/W on tape or lying loose
accuracy.thermal.part.heat_share	number		

key	type	unit	what it is
`accuracy.thermal.part.g_good_w_k`	number	W/K	
`accuracy.thermal.part.g_poor_w_k`	number	W/K	
`accuracy.source`	string		

`spindle`

the spindle drive: the Huanyang VFD (HY protocol, PD parameters), the motor, the pump

key	type	unit	value	what it is
`spindle.vfd`	object			the spindle drive (Huanyang HY series): rating, supply, the RS-485 port and the HY protocol, timing, the DC bus and its protection levels, its parameters
`spindle.vfd.model`	string		"Huanyang HY01D523B (HY series)"	
`spindle.vfd.rated_power_w`	number	W	1500	
`spindle.vfd.rated_current_a`	number	A	7.0	
`spindle.vfd.supply`	string		"1 x 220 V through KM1 (build.py ELECTRICAL)"	
`spindle.vfd.port`	string		"RS-485 to RS+/RS-/DCM (wiring.py PIN_MAP 主轴)"	
`spindle.vfd.protocol`	string		"HY: address, function, length, data, CRC-16 (Huanyang v1)"	
`spindle.vfd.reply_latency_ms`	number	ms	5.0	
`spindle.vfd.min_gap_ms`	number	ms	5.0	
`spindle.vfd.boot_s`	number	s	1.5	
`spindle.vfd.bus_capacitance_uf`	number	µF	680.0	
`spindle.vfd.lu_v`	number	V	200.0	DC bus: without mains it feeds the control and the motor until undervoltage (LU: output off), then the control's own supply stops below control_off_v
`spindle.vfd.control_off_v`	number	V	100.0	
`spindle.vfd.control_w`	number	W	15.0	
`spindle.vfd.oc_factor`	number		2.0	
`spindle.vfd.overload_factor`	number		1.5	
`spindle.vfd.overload_s`	number	s	60.0	
`spindle.vfd.dc_bus_v`	number	V	311.0	
`spindle.vfd.factory`	object			the drive's PD parameters as they leave the factory (its manual)
`spindle.vfd.factory.PD000`	number		0	
`spindle.vfd.factory.PD001`	number		0	
`spindle.vfd.factory.PD002`	number		0	
`spindle.vfd.factory.PD004`	number		50.0	
`spindle.vfd.factory.PD005`	number		50.0	
`spindle.vfd.factory.PD006`	number		2.5	
`spindle.vfd.factory.PD007`	number		0.5	
`spindle.vfd.factory.PD008`	number		220.0	
`spindle.vfd.factory.PD009`	number		15.0	
`spindle.vfd.factory.PD010`	number		8.0	
`spindle.vfd.factory.PD011`	number		0.0	
`spindle.vfd.factory.PD014`	number		10.0	
`spindle.vfd.factory.PD015`	number		10.0	
`spindle.vfd.factory.PD026`	number		0	

key	type	unit	value	what it is
spindle.vfd.factory.PD119	number		150	
spindle.vfd.factory.PD141	number		220.0	
spindle.vfd.factory.PD142	number		7.0	
spindle.vfd.factory.PD143	number		4	
spindle.vfd.factory.PD144	number		1440	
spindle.vfd.factory.PD163	number		0	
spindle.vfd.factory.PD164	number		1	
spindle.vfd.factory.PD165	number		0	
spindle.vfd.factory.PD118	number		1	the protection and compensation group as the twin reads it (numbers and meanings from the manual's list, unverified on the unit): PD118 overvoltage stall prevention (1 on: the decel ramp waits while the DC bus is high), PD120 stall prevention level at constant speed (% of PD142), PD121 its deceleration time (s), PD123 over-torque detection (0 off, 1 warn at constant speed, 2 trip), PD124 its level (% of PD142), PD125 its time (s), PD147 slip compensation (% of the rated slip made up at rated current; 0 off)
spindle.vfd.factory.PD120	number		150	
spindle.vfd.factory.PD121	number		5.0	
spindle.vfd.factory.PD123	number		0	
spindle.vfd.factory.PD124	number		150	
spindle.vfd.factory.PD125	number		0.1	
spindle.vfd.factory.PD147	number		0	
spindle.vfd.factory.PD041	number		8	PD041 the PWM carrier: 0-15 for 0.7, 1, 1.5, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 15, 17, 20 kHz; the manual's factory value for the A01D523B (the HY01D523B) is 8 = 8 kHz (research note N 2.3). Only the sound uses it
spindle.vfd.params	object			the PD parameters commissioning sets (the design's table wiring.VFD_PARAMS)
spindle.vfd.params.PD001	number		2	
spindle.vfd.params.PD002	number		2	
spindle.vfd.params.PD004	number		400.0	
spindle.vfd.params.PD005	number		400.0	
spindle.vfd.params.PD006	number		20.0	
spindle.vfd.params.PD007	number		0.5	
spindle.vfd.params.PD008	number		220.0	
spindle.vfd.params.PD009	number		16.0	
spindle.vfd.params.PD010	number		4.0	
spindle.vfd.params.PD011	number		100.0	
spindle.vfd.params.PD014	number		6.0	
spindle.vfd.params.PD015	number		6.0	
spindle.vfd.params.PD141	number		220.0	
spindle.vfd.params.PD142	number		5.0	
spindle.vfd.params.PD143	number		2	
spindle.vfd.params.PD144	number		3000	
spindle.vfd.params.PD163	number		1	
spindle.vfd.params.PD164	number		2	
spindle.vfd.params.PD165	number		3	
spindle.motor	object			the spindle motor, an induction motor: rating, poles, slip, breakdown torque, no-load current, inertia, friction
spindle.motor.model	string		"GDZ65-800"	
spindle.motor.rated_power_w	number	W	800.0	
spindle.motor.rated_voltage_v	number	V	220.0	
spindle.motor.rated_current_a	number	A	5.0	
spindle.motor.rated_freq_hz	number	Hz	400.0	

key	type	unit	value	what it is
spindle.motor.poles	number		2	
spindle.motor.rated_rpm	number	rpm	24000.0	
spindle.motor.rated_slip	number		0.04	
spindle.motor.breakdown_torque_ratio	number		2.0	
spindle.motor.no_load_current_a	number	A	2.0	
spindle.motor.inertia_kgm2	number	kg·m²	8e-05	
spindle.motor.friction_nm	number	N·m	0.005	
spindle.motor.viscous_nms	number	N·m·s	1e-05	
spindle.coolant_pump	string		"on relay K2 (COOLANT flood), supplied through KM1 (wirin…	
spindle.encoder	object			an incremental encoder on the spindle (G33, G76): lines a revolution and an index pulse once a revolution
spindle.encoder.ppr	number		0	lines a revolution (0: none); the controller's $38 is this (the other firmware: one channel's pulses) or 4 times it (the kernel: quadrature counts)

supply

the 24 V supply of controller, sensors and relays

key	type	unit	value	what it is
supply.psu24	string		"PSU2 24 V 60 W, before KM1: controller, sensors, the coi…	QF1 (the utility) feeds PSU2 directly and, through KM1, PSU1 and the VFD (wiring.py)
supply.psu24_holdup_ms	number	ms	80.0	
supply.psu24_rise_ms	number	ms	100.0	

tool_setter

the tool length sensor: where it stands (machine X, Y), its top (machine Z)

key	type	unit	value	what it is
tool_setter.x	number		25.0	
tool_setter.y	number		150.0	
tool_setter.top_z	number		-53.0	the contact (research note F 7, all unverified): it opens after the plunger has gone down pretravel_mm, the point scatters by repeat_sigma_mm (with --seed), it closes again reset_hyst_mm higher, and it bounces (closes for a moment) bounces times within bounce_ms as it opens (mech-r22: it sits SETTER_RECESS deep in a pocket in the table top - its top SETTER_TOP over the table)
tool_setter.overtravel	number		3.0	
tool_setter.contact	string		"NC"	
tool_setter.diameter	number		30.0	
tool_setter.pretravel_mm	number	mm	0.2	
tool_setter.repeat_sigma_mm	number	mm	0.001	
tool_setter.reset_hyst_mm	number	mm	0.01	
tool_setter.bounces	number		2	
tool_setter.bounce_ms	number	ms	0.5	

touch_probe

a touch probe tool (ball stylus on the PROBE input): ideal - it switches as the ball touches - or typical: a kinematic seat that opens once the stylus has given way by its pretravel (by the approach, scattered with a seed), closes again less deflected; its stylus breaks past its travel (docs/06 12.31)

key	type	unit	value	what it is
touch_probe.model	string		"ideal"	

key	type	unit	value	what it is
touch_probe.stylus_travel_mm	number	mm	2.0	A touch probe is a tool ("probe 2 L30": a ball stylus on the PROBE input with the setter). "ideal": it switches as the ball touches; "typical": a kinematic touch-trigger probe's seat opens once the stylus has given way by its pretravel - along Z, in XY with three lobes round the stylus (three seats), each touch's scattered by repeatability_sigma_mm with --seed - and closes again release_hysteresis_mm less deflected (docs/06 12.31). The MC-1's probe is an option not chosen yet: the typical values are orders of magnitude for a hobby-class probe, an assumption (research O: such probes claim 1-5 um repeatability, unverified), no specification of any probe.
touch_probe.geometry	object			the probe's shape in the collision zone (the jaws, the clamps, the workpiece): its stylus over the ball, its body from body_from_mm above the ball's lowest point up to the collet; 0: not given
touch_probe.geometry.stylus_d_mm	number	mm	1.5	the stylus' diameter over the ball (thinner than the ball: the ball stands for it)
touch_probe.geometry.body_d_mm	number	mm	20.0	the body's diameter (0: no body)
touch_probe.geometry.body_from_mm	number	mm	15.0	where the body starts, above the ball's lowest point (the stylus' length below it)
touch_probe.styli	array (3)			the styli by ball size (1.1, e09's Ø2 H7 features: a Ø2 ball does not fit) - each its own stem and where the body starts above the ball's lowest point; a probe tool takes the nearest ball's (within 25 %), else geometry's. Typical M2 ruby styli, an assumption until the user's are entered
touch_probe.styli[].ball_d_mm	number	mm	2.0	
touch_probe.styli[].stylus_d_mm	number	mm	1.5	
touch_probe.styli[].body_from_mm	number	mm	15.0	
touch_probe.typical	object			the typical probe's model: pretravel along Z and sideways, the three-lobed pattern, repeatability, release hysteresis - typical of touch-trigger probes, until a probe is calibrated
touch_probe.typical.pretravel_z_mm	number	mm	0.005	
touch_probe.typical.pretravel_xy_mm	number	mm	0.015	
touch_probe.typical.lobing_xy_mm	number	mm	0.005	
touch_probe.typical.lobing_count	number		3	
touch_probe.typical.lobing_phase_deg	number	°	0.0	
touch_probe.typical.repeatability_sigma_mm	number	mm	0.001	
touch_probe.typical.release_hysteresis_mm	number	mm	0.003	

sound

where the sounds come from (model mm at the GLB pose, the motion group each rides on) and the default listener, for the sound engine's panning

key	type	unit	value	what it is
sound.frame	string		"model mm, Z up, at the GLB pose (viewer.glb_pose); moves…	
sound.sources	object			the machine's sources of sound, by name
sound.sources.x_motor	object			one source of sound: where it is (model mm at the GLB pose) and the motion group it rides on
sound.sources.x_motor.at	array (3)		[254.0, 160.0, 250.0]	
sound.sources.x_motor.moves	string		"frame"	
sound.sources.y_motor	object			
sound.sources.y_motor.at	array (3)		[0.0, 245.0, 22.0]	
sound.sources.y_motor.moves	string		"frame"	
sound.sources.z_motor	object			
sound.sources.z_motor.at	array (3)		[0.0, 91.0, 467.0]	
sound.sources.z_motor.moves	string		"X"	
sound.sources.spindle	object			
sound.sources.spindle.at	array (3)		[0.0, 0.0, 206.5]	
sound.sources.spindle.moves	string		"Z"	
sound.sources.tool_tip	object			

key	type	unit	value	what it is
`sound.sources.tool_tip.at`	array (3)		[0.0, 0.0, 59.0]	
`sound.sources.tool_tip.moves`	string		"Z"	
`sound.sources.box`	object			
`sound.sources.box.at`	array (3)		[430.0, 50.0, 140.0]	
`sound.sources.box.moves`	string		"frame"	
`sound.sources.setter`	object			
`sound.sources.setter.at`	array (3)		[-75.0, 75.0, 86.0]	
`sound.sources.setter.moves`	string		"Y"	
`sound.sources.pump`	object			
`sound.sources.pump.at`	array (3)		[-335.0, 0.0, 40.0]	
`sound.sources.pump.moves`	string		"frame"	
`sound.listener`	object			where the listener stands and what it faces (model mm)
`sound.listener.at`	array (3)		[0.0, -695.0, 559.0]	
`sound.listener.looks_at`	array (3)		[0.0, 0.0, 99.0]	
`sound.radiation`	object			the cut's sound through the machine's modes (cutting.dynamics): Pa/N well above them, each mode's share of it (the rest radiates flat)
`sound.radiation.pa_per_n`	number	Pa/N	0.005	the cut's sound: the force's harmonics through the machine's modes (cutting.dynamics: the structure's X and Y modes, the tool's) as their accelerations radiate (research note N 2.4) - pa_per_n well above the modes, each mode's share of it (a peak of share / (2 zeta) at its frequency), the shares' rest flat
`sound.radiation.structure_x`	number		0.35	
`sound.radiation.structure_y`	number		0.35	
`sound.radiation.tool`	number		0.3	

`collision`

collision proxies: the table, the spindle stack above the tool tip (coaxial cylinders), what stands on the table (table-local mm, origin at its centre)

key	type	unit	value	what it is
`collision.table`	object			the table's proxy: its size, its origin in machine XY, the reference tool tip's machine Z on it
`collision.table.size`	array (2)		[300.0, 220.0]	
`collision.table.origin_machine_xy`	array (2)		[100.0, 75.0]	
`collision.table.ref_tip_on_table_z`	number		-80.0	
`collision.spindle`	array (6)			
`collision.spindle[].id`	string		"tool"	
`collision.spindle[].r`	number		3.0	
`collision.spindle[].h`	array (2)		[0.0, 15.5]	
`collision.table_items`	array (7)			
`collision.table_items[].id`	string		"setter_flange"	
`collision.table_items[].shape`	string		"box"	
`collision.table_items[].min`	array (3)		[-107.0, 66.0, -11.0]	
`collision.table_items[].max`	array (3)		[-43.0, 84.0, -6.0]	
`collision.groups`	array (3)			
`collision.groups[].id`	string		"z_slide"	
`collision.groups[].moves`	string		"XZ"	
`collision.groups[].min`	array (3)		[-73.5, 45.0, 95.0]	
`collision.groups[].max`	array (3)		[73.5, 65.0, 255.0]	

key	type	unit	value	what it is
`collision.near_miss_mm`	number	mm	1.0	a gap this small without contact is a near miss (0: none)

`cutting`

cutting-force coefficients (Altintas' linear edge-force model), materials, tool compliance and structural dynamics for the stock model

key	type	unit	value	what it is
`cutting.material`	string		"6061-T6 (assumed, research note C 1.5)"	
`cutting.ktc_n_mm2`	number	N/mm²	750.0	
`cutting.krc_n_mm2`	number	N/mm²	250.0	
`cutting.kac_n_mm2`	number	N/mm²	150.0	
`cutting.kte_n_mm`	number	N/mm	5.0	
`cutting.kre_n_mm`	number	N/mm	5.0	
`cutting.kae_n_mm`	number	N/mm	2.0	
`cutting.kc_plunge_j_mm3`	number	J/mm³	2.4	
`cutting.tool_default`	string		"flat 6 n2"	
`cutting.stock_res_mm`	number	mm	0.02	
`cutting.chunk_mm`	number	mm	0.5	the cut's time resolution (A10): chunks of 0.5 mm / 20 ms, or "tooth": a tooth period (at most 0.05 mm) at a time; polyline points within merge_tol_um of a segment's chord merged into it (the drives' tiny deviations at L1/L2 stay in the swept surface); the height pyramid skips the blocks the tool passes over; the sweep looks only at the cells each segment can reach below the blank's top (its profile's inverse: a V-bit engraving shallow reaches a tenth of a millimetre - round 16; false: as before, the same cut)
`cutting.chunk_ms`	number	ms	20.0	
`cutting.force_step`	string		"chunk"	
`cutting.merge_tol_um`	number	µm	0.1	
`cutting.pyramid`	boolean		true	
`cutting.sweep_reach`	boolean		true	
`cutting.variation`	boolean		true	the shear coefficients follow chip thickness and cutting speed where a material has orthogonal cutting data ("oblique"); false: constant coefficients everywhere (--set cutting.variation=0)
`cutting.wear`	object			the tools' wear book: kept and reported only (the cut stays a sharp tool's); the wear band's curve, the edge's clearance angle, a small tool's criterion
`cutting.wear.track`	boolean		true	the tools' wear book (1.1, twin/core/src/wear.h; handover/virtual-cnc-tools/wear_design.md): track - kept and reported only, the cut a sharp tool's; the wear band's curve (run in to run_in[1] VB_c by u = run_in[0], late_slope times as steep past its life), the edge's clearance angle (VB tan alpha: how far it goes back); a small tool's criterion at most vb_c_d x D, not under vb_c_min_mm (micro-milling studies: VB 12 um ends a 0.5 mm tool's life, 37.5 um a 1 mm one's - about 3 % of D; wear_life_data.md section 2)
`cutting.wear.alpha_deg`	number	°	10.0	
`cutting.wear.run_in`	array (2)		[0.05, 0.3]	
`cutting.wear.late_slope`	number		3.0	
`cutting.wear.vb_c_d`	number		0.03	
`cutting.wear.vb_c_min_mm`	number	mm	0.01	
`cutting.tool_geometry`	object			the end mills' geometry for that transformation: a two-flute aluminium cutter (the values Ozsoy used); the helix also spreads the force's harmonics (stock.c force_harmonics, docs/06 M3)

key	type	unit	value	what it is
cutting.tool_geometry.helix_deg	number	°	30.0	
cutting.tool_geometry.rake_deg	number	°	10.5	
cutting.tool_strength	object			the end mill's strength (research note L 4.4; the twin's M4): carbide rod's transverse rupture strength, the share of it a tool takes at once, the flute core's diameter for strength (x D)
cutting.tool_strength.trs_mpa	number	MPa	4000.0	
cutting.tool_strength.allowable	number		0.35	
cutting.tool_strength.core_d	number		0.7	
cutting.force_model	string		"tooth"	the cutting forces (V0 P1, stock.c and tooth.c): "tooth" - the tooth-level kernel, disc by disc and tooth by tooth (engagement from the cells cut, the tool's profile, the helix, runout, the minimum chip, Kienzle's law where a material has it); "chunk" - one engagement arc per chunk and the flat end mill's closed forms, as before. The runout at the tool's tip (TIR; research note L 3.3: ER11 collet and spindle 5-15 um), the offset's direction drawn per cutter from the seed unless phase_deg gives it; envelope: the outermost tooth cuts the part, the cutter reaching R + rho max cos(lambda - j 2 pi / N) - holes larger, bosses smaller (finding 187; false: the nominal radius, as before); the edge of a new carbide tool (radius; the minimum chip as a share of it, aluminium 0.2-0.4: note L 3.2)
cutting.runout	object			the cutter's runout in the collet (TIR), its direction (null: drawn for each tool from the session's seed), and whether the cut takes its envelope (finding 187)
cutting.runout.tir_um	number	µm	10.0	
cutting.runout.phase_deg	null	°	null	
cutting.runout.envelope	boolean		true	
cutting.edge	object			the cutting edge: its radius, and the minimum chip as a share of it - a thinner chip only ploughs
cutting.edge.radius_um	number	µm	2.0	
cutting.edge.h_min_ratio	number		0.3	
cutting.materials	array (17)			the workpiece materials the setup offers, with their cutting data
cutting.materials[].key	string		"6061-T6"	
cutting.materials[].name	string		"Aluminum alloy 6061-T6"	
cutting.materials[].group	string		"VDI 22 / ISO N"	
cutting.materials[].density_g_cm3	number	g/cm³	2.7	
cutting.materials[].ktc_n_mm2	number	N/mm²	750.0	
cutting.materials[].krc_n_mm2	number	N/mm²	250.0	
cutting.materials[].kac_n_mm2	number	N/mm²	150.0	
cutting.materials[].kte_n_mm	number	N/mm	5.0	
cutting.materials[].kre_n_mm	number	N/mm	5.0	
cutting.materials[].kae_n_mm	number	N/mm	2.0	
cutting.materials[].kc_plunge_j_mm3	number	J/mm³	2.4	
cutting.materials[].vc_m_min	array (2)	m/min	[152, 335]	
cutting.materials[].alpha_um_m_k	number	µm/(m·K)	23.6	
cutting.materials[].c_j_kg_k	number	J/(kg·K)	896	
cutting.materials[].color	string		"#b9bdc3"	
cutting.materials[].source	string		"aluminium sharp-tool set (note C 1.5)"	
cutting.materials[].note	string		"measured sharp-carbide set (note C 1.5); Kienzle 800/0.2…	

key	type	unit	value	what it is
`cutting.materials[].oblique`	object			orthogonal cutting data: shear stress, shear and friction angles as functions of the chip, the speed and the rake; the reference chip and speed; the source
`cutting.materials[].oblique.tau_s_mpa`	array (4)	MPa	[205.928, 204.038, 0.016, -0.056]	
`cutting.materials[].oblique.phi_n_deg`	array (4)	°	[13.866, 62.929, 0.005, 0.259]	
`cutting.materials[].oblique.beta_a_deg`	array (4)	°	[20.835, -4.901, -0.007, 0.291]	
`cutting.materials[].oblique.h_ref_mm`	number	mm	0.02	
`cutting.materials[].oblique.vc_ref_m_min`	number	m/min	452.4	
`cutting.materials[].oblique.source`	string		"M. Ozsoy, MATECA 6(3) 263-271 (2025), Table 1 (orthogona…	
`cutting.materials[].life`	object			a carbide end mill's life in it for the wear book (wear.h): Taylor's pair and exponent, the flank wear criterion, judged by length or by VB; estimates where the source says est
`cutting.materials[].life.taylor_n`	number	N	0.4	
`cutting.materials[].life.vc_ref_m_min`	number	m/min	300.0	
`cutting.materials[].life.t_ref_min`	number		60.0	
`cutting.materials[].life.vb_c_mm`	number	mm	0.3	
`cutting.materials[].life.basis`	string		"length"	
`cutting.materials[].life.source`	string		"est: carbide start speeds 244-366 m/min, life set by BUE…	
`cutting.compliance`	object			the static give at the tool tip: the tool (modulus, core, flute length), the holder's joint, the frame, the drives by the nut's position
`cutting.compliance.note`	string		"static compliance at the tool tip, force on the tool: to…	
`cutting.compliance.tool_e_n_mm2`	number	N/mm²	600000.0	
`cutting.compliance.flute_core`	number		0.8	
`cutting.compliance.flute_length_d`	number		2.5	
`cutting.compliance.tool_density_kg_m3`	number		14500.0	the tool's mass for its own bending mode (cutting.dynamics): cemented carbide, the fluted section a share of the full circle's area (two to four flutes: about 0.6-0.75; assumed)
`cutting.compliance.flute_area`	number		0.65	
`cutting.compliance.joint_um_per_100n`	number	µm per 100 N	10.0	
`cutting.compliance.frame_um_per_100n`	array (3)	µm per 100 N		
`cutting.compliance.frame_portal_um_per_100n`	array (3)	µm per 100 N	[2.3495, 0.0, 0.0]	
`cutting.compliance.drive_um_per_100n`	object	µm per 100 N		each axis' drive, axially, by the nut's distance from the fixed bearing
`cutting.compliance.drive_um_per_100n.X`	array (2)			
`cutting.compliance.drive_um_per_100n.Y`	array (2)			
`cutting.compliance.drive_um_per_100n.Z`	array (2)			
`cutting.compliance.source`	string		"research note C 3.1-3.3; frame: out/fea/fea_result.json …	
`cutting.dynamics`	object			the tool point's frequency response for chatter: the structure's and the tool's natural frequencies and damping, the spindle's speed range, input shaping

key	type	unit	value	what it is
cutting.dynamics.note	string		"tool-point FRF, per direction x (feed axes X, Y): struct…	
cutting.dynamics.structure_fn_hz	array (2)	Hz	[161.0, 196.0]	
cutting.dynamics.tool_fn_hz	number	Hz	1500.0	
cutting.dynamics.zeta	array (3)		[0.03, 0.03, 0.03]	
cutting.dynamics.rpm_range	array (2)	rpm	[6000.0, 24000.0]	
cutting.dynamics.source	string		"research note C 4.1, 4.4 (ZOA; the holder mode's frequen…	
cutting.dynamics.input_shaping	object			the kernel's input shaping (K1.10, kernel/docs/08-input-shaping.md): what the structure modes ask of it
cutting.dynamics.input_shaping.note	string		"not needed at the commissioned accelerations ($120-$122 …	
cutting.dynamics.input_shaping.commissioned	object			the input shaping settings as commissioned
cutting.dynamics.input_shaping.commissioned.$1000	number		0	
cutting.dynamics.input_shaping.commissioned.$1001	number		0	
cutting.dynamics.input_shaping.commissioned.$1002	number		0	
cutting.dynamics.input_shaping.raised_acceleration	object			the input shaping settings for raised accelerations: the modes they suppress
cutting.dynamics.input_shaping.raised_acceleration.$1000	number		2	
cutting.dynamics.input_shaping.raised_acceleration.$1010	number		161.0	
cutting.dynamics.input_shaping.raised_acceleration.$1020	number		0.05	
cutting.dynamics.input_shaping.raised_acceleration.$1001	number		2	
cutting.dynamics.input_shaping.raised_acceleration.$1011	number		196.0	
cutting.dynamics.input_shaping.raised_acceleration.$1021	number		0.05	
cutting.dynamics.input_shaping.source	string		"kernel line K1.10 (0bd7ffe): tests/shaper_test.py residu…	

workholding

the fixtures: the tapped grid, vise, parallels, strap clamps, tape, V-blocks, friction

key	type	unit	value	what it is
workholding.grid	object			the table's tapped grid (model.py: M6 x 12 in columns GRID_XS either side, 9 rows from y -100)
workholding.grid.thread	string		"M6"	
workholding.grid.x	array (8)		[-125.0, -100.0, -75.0, -50.0, 50.0, 75.0, 100.0, 125.0]	
workholding.grid.y	array (9)		[-100.0, -75.0, -50.0, -25.0, 0.0, 25.0, 50.0, 75.0, 100.0]	
workholding.vise	object			a 75 mm self-centring vise (5th Axis V75100X class, note H 2.2 [R30]) on a 10 mm aluminium subplate bolted into the grid: the only low vise that leaves room under MC-1's 80 mm of Z (jaws 66 mm up)
workholding.vise.name	string		"75 mm self-centering precision vise (V75100X class) + 10…	
workholding.vise.self_centring	boolean		true	
workholding.vise.jaw_width	number		75.0	
workholding.vise.jaw_height	number		21.15	
workholding.vise.jaw_thickness	number		12.0	
workholding.vise.step_height	number		2.5	

key	type	unit	value	what it is
workholding.vise.min_opening	number		3.25	
workholding.vise.max_opening	number		77.0	
workholding.vise.base_height	number		35.0	
workholding.vise.body_width	number		75.0	
workholding.vise.body_length	number		100.0	
workholding.vise.plate_thickness	number		10.0	
workholding.vise.plate_length	number		130.0	
workholding.vise.plate_width	number		110.0	
workholding.vise.clamp_force_n	number	N	7400.0	
workholding.vise_low	object			the low-profile vise on the grid (a set-up's "vise": "low"): jaw tops low enough for a short probe to get over the blank with Z at its top
workholding.vise_low.name	string		"Low-profile vise, 75 mm jaws, 128 mm opening (Inception ...	
workholding.vise_low.self_centring	boolean		false	
workholding.vise_low.jaw_width	number		75.0	
workholding.vise_low.jaw_height	number		15.0	
workholding.vise_low.jaw_thickness	number		15.0	
workholding.vise_low.step_height	number		3.0	
workholding.vise_low.min_opening	number		0.0	
workholding.vise_low.max_opening	number		128.0	
workholding.vise_low.base_height	number		5.0	
workholding.vise_low.body_width	number		75.0	
workholding.vise_low.body_length	number		210.0	
workholding.vise_low.plate_thickness	number		0.0	
workholding.vise_low.plate_length	number		0.0	
workholding.vise_low.plate_width	number		0.0	
workholding.vise_low.clamp_force_n	number	N	5000.0	
workholding.parallels	object			a standard parallel set (note H 2.6 [R45]): 1/8" thick, 1/2"..1-5/8" high in 1/8" steps, 6" long
workholding.parallels.thickness	number		3.18	
workholding.parallels.min_height	number		12.7	
workholding.parallels.step	number		3.175	
workholding.parallels.max_height	number		41.3	
workholding.parallels.length	number		152.4	
workholding.clamp	object			M6 serrated step clamp (Te-Co 30501M, 64 x 25 x 13) with a step block behind the stud; the stud screws into the grid; 5.4 N m gives 4.45 kN in the stud (Carr Lane, note H 2.3 [R37][R39])
workholding.clamp.strap_length	number		64.0	
workholding.clamp.strap_width	number		25.0	
workholding.clamp.strap_thickness	number		13.0	
workholding.clamp.nose_on_part	number		8.0	
workholding.clamp.heel_length	number		14.0	
workholding.clamp.reach_min	number		7.0	
workholding.clamp.reach_max	number		35.0	
workholding.clamp.stud_diameter	number		6.0	
workholding.clamp.washer_diameter	number		12.0	
workholding.clamp.nut_height	number		7.0	
workholding.clamp.bolt_preload_n	number	N	4450.0	

key	type	unit	value	what it is
workholding.tape	object			3M VHB 4910, 1.0 mm acrylic foam (note H 2.4 [R41]): 480 kPa overlap shear after 72 h; half of it counted (peel from the moment of a lateral force above the bond line, imperfect contact) [assumed]
workholding.tape.thickness	number		1.0	
workholding.tape.shear_mpa	number	MPa	0.48	
workholding.tape.effective_share	number		0.5	
workholding.vblock	object			a large V-block pair (Fowler X-Blox, 50.8 x 38.1 x 38.1, 38 mm capacity, 90 deg [R43]) with its strap clamp (force assumed)
workholding.vblock.length	number		38.1	
workholding.vblock.width	number		50.8	
workholding.vblock.height	number		38.1	
workholding.vblock.v_width	number		28.0	
workholding.vblock.clamp_force_n	number	N	2000.0	
workholding.friction	object			friction coefficients - the part on the table, in the jaws, under a strap - and the safety factor of the slip check
workholding.friction.part_on_table	number		0.3	physics, not design values (note H 2.7): the part slips when the cut beats these
workholding.friction.jaw_on_part	number		0.35	
workholding.friction.strap_on_part	number		0.35	
workholding.friction.safety_factor	number		3.0	

total_mass_kg

key	type	unit	value	what it is
total_mass_kg	number	kg	66.2	

controller

the controller: firmware, input filtering, boot time, the $-settings, the pin map

key	type	unit	value	what it is
controller.firmware	string		"Twinwright controller firmware (the kernel, 1.0) with it…	
controller.board_profile	string		"Teensy 4.1 + T41U5XBB: 24 MHz step timer, USB-CDC"	
controller.board	object			the controller board as the MCU side is modelled: step timer, step pulses and the $29 delay, input debounce, the sender's link (architecture 4.2)
controller.board.id	string		"teensy41_t41u5xbb"	
controller.board.text	string		"Teensy 4.1 (i.MX RT1062, Cortex-M7 600 MHz) on a Brookwo…	
controller.board.my_machine_board	string		"BOARD_T41U5XBB"	
controller.board.f_step_timer_hz	number	Hz	24000000	the step timer's clock (hal.f_step_timer); the twin's virtual clock runs at it
controller.board.timer_period_extra	number		1	
controller.board.timer_max_cycles	number		1048575	
controller.board.wake_up_ms	number	ms	2.0	
controller.board.step_pulse_latency_us	number	µs	0.2	
controller.board.step_toff_min_us	number	µs	2.0	
controller.board.pulse_timer_hz	number	Hz	150000000	
controller.board.signals	array (4)		["estop", "feed_hold", "cycle_start", "motor_fault"]	
controller.board.debounce_ms	number	ms	40	control inputs are acted on this long after their edge, if still on (0: at once)

key	type	unit	value	what it is
controller.board.debounced	array (4)		["reset", "feed_hold", "cycle_start", "safety_door"]	
controller.board.link_default	string		"usb_cdc"	the sender's line as the firmware is configured (my_machine.h USB_SERIAL_CDC): native USB, a UART, or Telnet
controller.board.links	object			the lines the board offers the sender: usb_cdc and telnet are packets the foreground polls, uart a byte per slot with a receive interrupt and the host's USB-serial adapter
controller.board.links.usb_cdc	object			the USB CDC link: frames, packets, buffers, polling, the host's latencies and jitter
controller.board.links.usb_cdc.frame_us	number	µs	125.0	
controller.board.links.usb_cdc.packet	number		512	
controller.board.links.usb_cdc.rx_capacity	number		4096	
controller.board.links.usb_cdc.poll_us	number	µs	50.0	
controller.board.links.usb_cdc.poll_bytes	number		20	
controller.board.links.usb_cdc.tx_buffer	number		6124	
controller.board.links.usb_cdc.tx_flush_us	number	µs	75.0	
controller.board.links.usb_cdc.host_out_latency_us	number	µs	250.0	
controller.board.links.usb_cdc.host_in_latency_us	number	µs	500.0	
controller.board.links.usb_cdc.jitter_us	number	µs	250.0	
controller.board.links.uart	object			the UART link through a USB-serial adapter: baud rate, the adapter's packets and timer, the host's latencies and jitter
controller.board.links.uart.baud	number		115200.0	
controller.board.links.uart.adapter	string		"FT232R-class USB-serial adapter"	
controller.board.links.uart.adapter_packet	number		62	
controller.board.links.uart.adapter_timer_ms	number	ms	16.0	the adapter's latency timer: a part-filled packet waits this long
controller.board.links.uart.host_out_latency_us	number	µs	1000.0	
controller.board.links.uart.host_in_latency_us	number	µs	1000.0	
controller.board.links.uart.jitter_us	number	µs	500.0	
controller.board.links.telnet	object			the network (Telnet) link: packets, buffers, polling, latencies and jitter
controller.board.links.telnet.frame_us	number	µs	0.0	
controller.board.links.telnet.packet	number		1460	
controller.board.links.telnet.rx_capacity	number		2920	
controller.board.links.telnet.poll_us	number	µs	1000.0	
controller.board.links.telnet.poll_bytes	number		1460	
controller.board.links.telnet.tx_buffer	number		1024	
controller.board.links.telnet.tx_flush_us	number	µs	1000.0	
controller.board.links.telnet.host_out_latency_us	number	µs	200.0	
controller.board.links.telnet.host_in_latency_us	number	µs	200.0	
controller.board.links.telnet.jitter_us	number	µs	300.0	
controller.board.fg_cost_us	object	µs		the foreground's CPU time per unit of work (a line parsed, a block planned, a step segment prepared), us

key	type	unit	value	what it is
controller.board.fg_cost_us.line	number		15.0	The foreground's CPU time (architecture 4.2 "前台耗时模型"): charged per unit of work - a line parsed and executed, a block planned (with the planner's recalculation), a step segment prepared - so a slow foreground starves the step segment buffer as on a real MCU. Estimates for the 600 MHz Cortex-M7 with its FPU (assumed): measure on the board with a GPIO toggled around each (03-commissioning).
controller.board.fg_cost_us.plan_block	number		5.0	
controller.board.fg_cost_us.segment	number		3.0	
controller.board.identity	object			What the controller says about itself in $I (research note I 1.2, docs/06 G3): the driver's own lines as the iMXRT1062 driver sets them (driver.c: hal.info, f_mcu, driver_version; USB_SERIAL_CDC 2: the options) and the board map's name - read the board's $I at commissioning and put its lines here (FREE MEMORY then too). twin_marker: one more line, [PLUGIN:<marker>], tells the twin apart; "" gives the board's $I exactly (a sender cannot tell them apart - docs/06 8.6: by choice only)
controller.board.identity.driver	string		"iMXRT1062"	
controller.board.identity.mcu_mhz	number	MHz	600	
controller.board.identity.driver_version	string		"260616"	
controller.board.identity.driver_options	string		"USB.2"	
controller.board.identity.board	string		"T41U5XBB"	
controller.board.identity.twin_marker	string		"MC-1 twin"	
controller.firmware_options	object			what the controller's firmware is built with: its build options, signals, probe, plugins, storage, axes, the driver's options
controller.firmware_options.my_machine_h	object			the build options as the firmware's my_machine.h names them
controller.firmware_options.my_machine_h.BOARD_T41U5XBB	number		1	
controller.firmware_options.my_machine_h.USB_SERIAL_CDC	number		2	
controller.firmware_options.my_machine_h.SPINDLE0_ENABLE	string		"SPINDLE_HUANYANG1"	
controller.firmware_options.my_machine_h.MODBUS_ENABLE	number		1	
controller.firmware_options.my_machine_h.ESTOP_ENABLE	number		1	
controller.firmware_options.my_machine_h.MOTOR_FAULT_ENABLE	number		1	
controller.firmware_options.my_machine_h.MPG_ENABLE	number		0	
controller.firmware_options.signals	array (4)		["estop", "feed_hold", "cycle_start", "motor_fault"]	
controller.firmware_options.probe	boolean		true	
controller.firmware_options.toolsetter	boolean		false	
controller.firmware_options.plugins	array (2)		["MODBUS", "HUANYANG VFD"]	
controller.firmware_options.nvs	string		"flash"	
controller.firmware_options.axes	number		3	
controller.firmware_options.driver	object			the board driver's options: debounce, reboot, clock, planner blocks, receive buffer
controller.firmware_options.driver.software_debounce	boolean		true	the iMXRT1062 driver's own (assumed from its source; the twin's driver the same): the control inputs are debounced in software (OPT S), $REBOOT restarts the MCU (NEWOPT REBOOT), the RT1062's RTC (NEWOPT RTC - though that driver never sets hal.driver_cap.rtc, so a real board answers $RTC with error:3: finding 94)
controller.firmware_options.driver.reboot	boolean		true	
controller.firmware_options.driver.rtc	boolean		true	

key	type	unit	value	what it is
controller.firmware_options.driver.planner_blocks	number		100	
controller.firmware_options.driver.rx_buffer	number		1024	
controller.panel	object			the control box's buttons and the controller inputs they are on
controller.panel.cycle_start	object			the cycle start button (SB2): how it reaches the controller, its input and pin, its contact
controller.panel.cycle_start.via	string		"pin"	
controller.panel.cycle_start.input	string		"CYCLE_START"	
controller.panel.cycle_start.teensy_pin	number		17	
controller.panel.cycle_start.button	string		"循环启动按钮 SB2（绿色，1 常开），电控箱门"	
controller.panel.cycle_start.contact	string		"NO"	
controller.panel.feed_hold	object			the feed hold button (SB3): how it reaches the controller, its input and pin, its contact
controller.panel.feed_hold.via	string		"pin"	
controller.panel.feed_hold.input	string		"FEED_HOLD"	
controller.panel.feed_hold.teensy_pin	number		16	
controller.panel.feed_hold.button	string		"进给保持按钮 SB3（黄色，1 常开），电控箱门"	
controller.panel.feed_hold.contact	string		"NO"	
controller.input_filter_us	number	µs	20.0	the board's isolated inputs: shorter pulses do not reach the pin (optocoupler; an RC filter would add to it). Limit inputs are not debounced in software (research F 2, 6)
controller.boot_ms	number	ms	50.0	
controller.settings	object			the $-settings (setting number -> value), also in mc1_settings.nc
controller.settings.$3	number		5	
controller.settings.$100	number		800	
controller.settings.$101	number		800	
controller.settings.$102	number		800	
controller.settings.$110	number		3000.0	
controller.settings.$111	number		3000.0	
controller.settings.$112	number		1500.0	
controller.settings.$120	number		300.0	
controller.settings.$121	number		300.0	
controller.settings.$122	number		200.0	
controller.settings.$130	number		202.0	
controller.settings.$131	number		152.0	
controller.settings.$132	number		82.0	
controller.settings.$10	number		15871	
controller.settings.$11	number		0.002	
controller.settings.$12	number		0.002	
controller.settings.$0	number		5.0	
controller.settings.$29	number		5.0	
controller.settings.$1	number		255	
controller.settings.$37	number		7	
controller.settings.$5	number		7	
controller.settings.$6	number		1	
controller.settings.$22	number		541	
controller.settings.$20	number		1	

key	type	unit	value	what it is
controller.settings.$21	number		1	
controller.settings.$23	number		0	
controller.settings.$44	number		4	
controller.settings.$45	number		3	
controller.settings.$46	number		0	
controller.settings.$180	number		50	
controller.settings.$181	number		50	
controller.settings.$182	number		50	
controller.settings.$190	number		1200	
controller.settings.$191	number		1200	
controller.settings.$192	number		600	
controller.settings.$26	number		250	
controller.settings.$27	number		2.0	
controller.settings.$43	number		1	
controller.settings.$65	number		32	
controller.settings.$341	number		3	
controller.settings.$342	number		70	
controller.settings.$343	number		25	
controller.settings.$344	number		200	
controller.settings.$346	number		0	
controller.settings.$30	number		24000	
controller.settings.$31	number		6000	
controller.settings.$340	number		5	
controller.settings.$374	number		3	
controller.settings.$375	number		50	
controller.settings.$394	number		15	
controller.settings.$460	number		1	
controller.settings_file	string		"twin/machine/mc1_settings.nc"	
controller.pin_map	array (16)			
controller.design_settings_table	array (28)			

viewer

the 3D views: the GLB model and its node groups, pose conventions, cable chains

key	type	unit	value	what it is
viewer.glb	string		"out/web/mc1.glb"	Everything twin/web needs to draw the machine: the GLB (moving groups) plus the drag chains, cable routes and Z service loop, which the viewer rebuilds per pose.
viewer.groups	object			the 3D model's motion groups: which part moves with which axis
viewer.groups.frame	string		"fixed_frame"	
viewer.groups.Y	string		"Y_table"	
viewer.groups.X	string		"X_carriage"	
viewer.groups.Z	string		"Z_slide"	
viewer.groups.dyn	string		"cable_chains"	
viewer.glb_pose	string		"model mm, Z up; exported at x_c = y_t = z_up = 0 where x…	
viewer.table_top_z	number		59.0	

key	type	unit	value	what it is
viewer.tool_tip	string		"spindle axis at model y = 0; tool tip = (x_c, 0, table_t…	
viewer.chains	object			the cable chains' geometry for the 3D view
viewer.chains.x	object			the X cable chain: link size, pitch, bend radius, length, where it is fixed, how it moves
viewer.chains.x.h	number		22.0	
viewer.chains.x.w	number		42.0	
viewer.chains.x.pitch	number		20.0	
viewer.chains.x.R	number		38.0	
viewer.chains.x.L	number		400.0	
viewer.chains.x.fixed	number		150.0	
viewer.chains.x.y	number		204.5	
viewer.chains.x.z	number		340.0	
viewer.chains.x.move_dx	number		40.0	
viewer.chains.x.lift	number		1.0	
viewer.chains.x.fill	number		0.85	
viewer.chains.x.loop_sign	number		-1	
viewer.chains.y	object			the Y cable chain: link size, pitch, bend radius, length, where it is fixed, how it moves
viewer.chains.y.h	number		15.0	
viewer.chains.y.w	number		22.0	
viewer.chains.y.pitch	number		15.0	
viewer.chains.y.R	number		20.0	
viewer.chains.y.L	number		360.0	
viewer.chains.y.fixed	number		150.0	
viewer.chains.y.x	number		-169.0	
viewer.chains.y.z	number		0.0	
viewer.chains.y.move_dy	number		0.0	
viewer.chains.y.lift	number		1.0	
viewer.chains.y.fill	number		0.85	
viewer.chains.y.loop_sign	number		-1	
viewer.routes	object			the cables' routes for the 3D view, as points (model mm)
viewer.routes.x_motor	array (5)			
viewer.routes.y_motor	array (6)			
viewer.routes.y_sensor_1	array (4)			
viewer.routes.y_sensor_2	array (4)			
viewer.routes.x_sensor_L	array (7)			
viewer.routes.x_sensor_R	array (7)			
viewer.routes.x_chain_in_zmotor	array (4)			
viewer.routes.x_chain_out	array (7)			
viewer.routes.y_chain_in_setter	array (5)			
viewer.routes.y_chain_out	array (6)			
viewer.route_groups	object			the routes that ride on a cable chain, and which

key	type	unit	value	what it is
viewer.route_groups.x_chain_in_zmotor	string		"x"	
viewer.route_groups.y_chain_in_setter	string		"y"	
viewer.cable_dia	object			each route's cable diameter (mm)
viewer.cable_dia.x_motor	number		9.0	
viewer.cable_dia.y_motor	number		9.0	
viewer.cable_dia.y_sensor_1	number		4.0	
viewer.cable_dia.y_sensor_2	number		4.0	
viewer.cable_dia.x_sensor_L	number		4.0	
viewer.cable_dia.x_sensor_R	number		4.0	
viewer.cable_dia.x_chain_in_zmotor	number		9.0	
viewer.cable_dia.x_chain_out	number		18.0	
viewer.cable_dia.y_chain_in_setter	number		4.0	
viewer.cable_dia.y_chain_out	number		4.0	
viewer.service_loop	object			the Z cables' service loop at three heights of Z, as points (model mm)
viewer.service_loop.0.0	array (7)			
viewer.service_loop.40.0	array (7)			
viewer.service_loop.80.0	array (7)			

provenance

where the values come from: a path pattern ("a[]" every item of a, "x/y_*" these keys, * a prefix; it covers what is under it; the most specific pattern wins) -> its record (B1)

key	t
provenance.axes[].travel	
provenance.axes[].travel.source	s
provenance.axes[].travel.note	s
provenance.axes[].travel.uncertainty	s

key	t
provenance.axes[].travel.refs	a (
provenance.axes[].travel.values	s
provenance.axes[].travel.history	a (
provenance.axes[].travel.history[].date	s
provenance.axes[].travel.history[].change	s
provenance.axes[].steps_per_mm	c
provenance.axes[].steps_per_mm.source	s
provenance.axes[].steps_per_mm.note	s
provenance.axes[].steps_per_mm.uncertainty	s
provenance.axes[].steps_per_mm.refs	a (
provenance.axes[].steps_per_mm.values	s
provenance.axes[].steps_per_mm.history	a (
provenance.axes[].steps_per_mm.history[].date	s
provenance.axes[].steps_per_mm.history[].change	s
provenance.axes[].switch.min/max	c
provenance.axes[].switch.min/max.source	s
provenance.axes[].switch.min/max.note	s
provenance.axes[].switch.min/max.uncertainty	s
provenance.axes[].switch.min/max.refs	a (
provenance.axes[].switch.min/max.values	s
provenance.axes[].switch.min/max.history	a (
provenance.axes[].switch.min/max.history[].date	s
provenance.axes[].switch.min/max.history[].change	s
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm	c
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm.source	s
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm.note	s
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm.uncertainty	s
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm.refs	a (
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm.values	s
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm.history	a (
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm.history[].date	s
provenance.axes[].switch.sn_mm/gap_mm/gap_design_mm/spot_r_cad_mm.history[].change	s

key	t
provenance.axes[].switch.lambda_mm/w_mm	c
provenance.axes[].switch.lambda_mm/w_mm.source	s
provenance.axes[].switch.lambda_mm/w_mm.note	s
provenance.axes[].switch.lambda_mm/w_mm.uncertainty	s
provenance.axes[].switch.lambda_mm/w_mm.refs	a (
provenance.axes[].switch.lambda_mm/w_mm.values	s
provenance.axes[].switch.lambda_mm/w_mm.history	a (
provenance.axes[].switch.lambda_mm/w_mm.history[].date	s
provenance.axes[].switch.lambda_mm/w_mm.history[].change	s
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm	c
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm.source	s
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm.note	s
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm.uncertainty	s
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm.refs	a (
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm.values	s
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm.history	a (
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm.history[].date	s
provenance.axes[].switch.hyst_mm/response_ms/jitter_ms/repeat_sigma_mm.history[].change	s
provenance.axes[].switch.temp_c/temp_coeff_per_k	c
provenance.axes[].switch.temp_c/temp_coeff_per_k.source	s
provenance.axes[].switch.temp_c/temp_coeff_per_k.note	s
provenance.axes[].switch.temp_c/temp_coeff_per_k.uncertainty	s
provenance.axes[].switch.temp_c/temp_coeff_per_k.refs	a (
provenance.axes[].switch.temp_c/temp_coeff_per_k.values	s
provenance.axes[].switch.temp_c/temp_coeff_per_k.history	a (
provenance.axes[].switch.temp_c/temp_coeff_per_k.history[].date	s
provenance.axes[].switch.temp_c/temp_coeff_per_k.history[].change	s
provenance.axes[].switch.ready_ms	c
provenance.axes[].switch.ready_ms.source	s
provenance.axes[].switch.ready_ms.note	s
provenance.axes[].switch.ready_ms.uncertainty	s
provenance.axes[].switch.ready_ms.refs	a (
provenance.axes[].switch.ready_ms.values	s
provenance.axes[].switch.ready_ms.history	a (
provenance.axes[].switch.ready_ms.history[].date	s
provenance.axes[].switch.ready_ms.history[].change	s
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr	c
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr.source	s
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr.note	s
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr.uncertainty	s

key	t
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr.refs	a (
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr.values	s
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr.history	a (
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr.history[].date	s
provenance.axes[].switch.contact/inputs/type/anchor/target/unit_sr.history[].change	s
provenance.axes[].hard_stop	o
provenance.axes[].hard_stop.source	s
provenance.axes[].hard_stop.note	s
provenance.axes[].hard_stop.uncertainty	s
provenance.axes[].hard_stop.refs	a (
provenance.axes[].hard_stop.values	s
provenance.axes[].hard_stop.history	a (
provenance.axes[].hard_stop.history[].date	s
provenance.axes[].hard_stop.history[].change	s
provenance.axes[].moving_mass_kg	o
provenance.axes[].moving_mass_kg.source	s
provenance.axes[].moving_mass_kg.note	s
provenance.axes[].moving_mass_kg.uncertainty	s
provenance.axes[].moving_mass_kg.refs	a (
provenance.axes[].moving_mass_kg.values	s
provenance.axes[].moving_mass_kg.history	a (
provenance.axes[].moving_mass_kg.history[].date	s
provenance.axes[].moving_mass_kg.history[].change	s
provenance.axes[].id/moves/home_dir	o
provenance.axes[].id/moves/home_dir.source	s
provenance.axes[].id/moves/home_dir.note	s
provenance.axes[].id/moves/home_dir.uncertainty	s
provenance.axes[].id/moves/home_dir.refs	a (
provenance.axes[].id/moves/home_dir.values	s
provenance.axes[].id/moves/home_dir.history	a (
provenance.axes[].id/moves/home_dir.history[].date	s
provenance.axes[].id/moves/home_dir.history[].change	s
provenance.axes[].dir_sign/direction	o
provenance.axes[].dir_sign/direction.source	s
provenance.axes[].dir_sign/direction.note	s
provenance.axes[].dir_sign/direction.uncertainty	s
provenance.axes[].dir_sign/direction.refs	a (
provenance.axes[].dir_sign/direction.values	s
provenance.axes[].dir_sign/direction.history	a (
provenance.axes[].dir_sign/direction.history[].date	s

key	t
provenance.axes[].dir_sign/direction.history[].change	s
provenance.sensor_materials	o
provenance.sensor_materials.source	s
provenance.sensor_materials.note	s
provenance.sensor_materials.uncertainty	s
provenance.sensor_materials.refs	a (
provenance.sensor_materials.values	s
provenance.sensor_materials.history	a (
provenance.sensor_materials.history[].date	s
provenance.sensor_materials.history[].change	s
provenance.wires	o
provenance.wires.source	s
provenance.wires.note	s
provenance.wires.uncertainty	s
provenance.wires.refs	a (
provenance.wires.values	s
provenance.wires.history	a (
provenance.wires.history[].date	s
provenance.wires.history[].change	s
provenance.faults	o
provenance.faults.source	s
provenance.faults.note	s
provenance.faults.uncertainty	s
provenance.faults.refs	a (
provenance.faults.values	s
provenance.faults.history	a (
provenance.faults.history[].date	s
provenance.faults.history[].change	s
provenance.controller.input_filter_us	o
provenance.controller.input_filter_us.source	s
provenance.controller.input_filter_us.note	s

key	t
provenance.controller.input_filter_us.uncertainty	s
provenance.controller.input_filter_us.refs	a (
provenance.controller.input_filter_us.values	s
provenance.controller.input_filter_us.history	a (
provenance.controller.input_filter_us.history[].date	s
provenance.controller.input_filter_us.history[].change	s
provenance.controller.boot_ms	c
provenance.controller.boot_ms.source	s
provenance.controller.boot_ms.note	s
provenance.controller.boot_ms.uncertainty	s
provenance.controller.boot_ms.refs	a (
provenance.controller.boot_ms.values	s
provenance.controller.boot_ms.history	a (
provenance.controller.boot_ms.history[].date	s
provenance.controller.boot_ms.history[].change	s
provenance.controller.board	c
provenance.controller.board.source	s
provenance.controller.board.note	s
provenance.controller.board.uncertainty	s
provenance.controller.board.refs	a (
provenance.controller.board.values	s
provenance.controller.board.history	a (
provenance.controller.board.history[].date	s
provenance.controller.board.history[].change	s
provenance.controller.board.links	c
provenance.controller.board.links.source	s
provenance.controller.board.links.note	s
provenance.controller.board.links.uncertainty	s
provenance.controller.board.links.refs	a (
provenance.controller.board.links.values	s
provenance.controller.board.links.history	a (
provenance.controller.board.links.history[].date	s
provenance.controller.board.links.history[].change	s
provenance.controller.board.identity	c
provenance.controller.board.identity.source	s
provenance.controller.board.identity.note	s
provenance.controller.board.identity.uncertainty	s
provenance.controller.board.identity.refs	a (
provenance.controller.board.identity.values	s
provenance.controller.board.identity.history	a (

key	t
provenance.controller.board.identity.history[].date	s
provenance.controller.board.identity.history[].change	s
provenance.controller.board.fg_cost_us	o
provenance.controller.board.fg_cost_us.source	s
provenance.controller.board.fg_cost_us.note	s
provenance.controller.board.fg_cost_us.uncertainty	s
provenance.controller.board.fg_cost_us.refs	a (
provenance.controller.board.fg_cost_us.values	s
provenance.controller.board.fg_cost_us.history	a (
provenance.controller.board.fg_cost_us.history[].date	s
provenance.controller.board.fg_cost_us.history[].change	s
provenance.controller.settings	o
provenance.controller.settings.source	s
provenance.controller.settings.note	s
provenance.controller.settings.uncertainty	s
provenance.controller.settings.refs	a (
provenance.controller.settings.values	s
provenance.controller.settings.history	a (
provenance.controller.settings.history[].date	s
provenance.controller.settings.history[].change	s
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file	o
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file.source	s
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file.note	s
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file.uncertainty	s
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file.refs	a (
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file.values	s
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file.history	a (
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file.history[].date	s
provenance.controller.firmware/board_profile/pin_map/design_settings_table/settings_file.history[].change	s
provenance.controller.firmware_options	o
provenance.controller.firmware_options.source	s
provenance.controller.firmware_options.note	s
provenance.controller.firmware_options.uncertainty	s
provenance.controller.firmware_options.refs	a (
provenance.controller.firmware_options.values	s
provenance.controller.firmware_options.history	a (
provenance.controller.firmware_options.history[].date	s
provenance.controller.firmware_options.history[].change	s
provenance.controller.firmware_options.driver	o
provenance.controller.firmware_options.driver.source	s
provenance.controller.firmware_options.driver.note	s

key	t
provenance.controller.firmware_options.driver.uncertainty	s
provenance.controller.firmware_options.driver.refs	a (
provenance.controller.firmware_options.driver.values	s
provenance.controller.firmware_options.driver.history	a (
provenance.controller.firmware_options.driver.history[].date	s
provenance.controller.firmware_options.driver.history[].change	s
provenance.controller.board.signals	o
provenance.controller.board.signals.source	s
provenance.controller.board.signals.note	s
provenance.controller.board.signals.uncertainty	s
provenance.controller.board.signals.refs	a (
provenance.controller.board.signals.values	s
provenance.controller.board.signals.history	a (
provenance.controller.board.signals.history[].date	s
provenance.controller.board.signals.history[].change	s
provenance.controller.panel	o
provenance.controller.panel.source	s
provenance.controller.panel.note	s
provenance.controller.panel.uncertainty	s
provenance.controller.panel.refs	a (
provenance.controller.panel.values	s
provenance.controller.panel.history	a (
provenance.controller.panel.history[].date	s
provenance.controller.panel.history[].change	s
provenance.supply	o
provenance.supply.source	s
provenance.supply.note	s
provenance.supply.uncertainty	s
provenance.supply.refs	a (
provenance.supply.values	s
provenance.supply.history	a (
provenance.supply.history[].date	s
provenance.supply.history[].change	s
provenance.power_on_pos	o
provenance.power_on_pos.source	s
provenance.power_on_pos.note	s
provenance.power_on_pos.uncertainty	s
provenance.power_on_pos.refs	a (
provenance.power_on_pos.values	s

key	t
provenance.power_on_pos.history	a (
provenance.power_on_pos.history[].date	s
provenance.power_on_pos.history[].change	s
provenance.drives.motor	o
provenance.drives.motor.source	s
provenance.drives.motor.note	s
provenance.drives.motor.uncertainty	s
provenance.drives.motor.refs	a (
provenance.drives.motor.values	s
provenance.drives.motor.history	a (
provenance.drives.motor.history[].date	s
provenance.drives.motor.history[].change	s
provenance.drives.motor.detent_nm	o
provenance.drives.motor.detent_nm.source	s
provenance.drives.motor.detent_nm.note	s
provenance.drives.motor.detent_nm.uncertainty	s
provenance.drives.motor.detent_nm.refs	a (
provenance.drives.motor.detent_nm.values	s
provenance.drives.motor.detent_nm.history	a (
provenance.drives.motor.detent_nm.history[].date	s
provenance.drives.motor.detent_nm.history[].change	s
provenance.drives.motor.torque_constant_nm_a	o
provenance.drives.motor.torque_constant_nm_a.source	s
provenance.drives.motor.torque_constant_nm_a.note	s
provenance.drives.motor.torque_constant_nm_a.uncertainty	s
provenance.drives.motor.torque_constant_nm_a.refs	a (
provenance.drives.motor.torque_constant_nm_a.values	s
provenance.drives.motor.torque_constant_nm_a.history	a (
provenance.drives.motor.torque_constant_nm_a.history[].date	s
provenance.drives.motor.torque_constant_nm_a.history[].change	s
provenance.drives.motor.iron_loss_nm_per_krpm	o
provenance.drives.motor.iron_loss_nm_per_krpm.source	s
provenance.drives.motor.iron_loss_nm_per_krpm.note	s
provenance.drives.motor.iron_loss_nm_per_krpm.uncertainty	s
provenance.drives.motor.iron_loss_nm_per_krpm.refs	a (
provenance.drives.motor.iron_loss_nm_per_krpm.values	s
provenance.drives.motor.iron_loss_nm_per_krpm.history	a (
provenance.drives.motor.iron_loss_nm_per_krpm.history[].date	s
provenance.drives.motor.iron_loss_nm_per_krpm.history[].change	s
provenance.drives.motor.power_up_jump	o

key	t
provenance.drives.motor.power_up_jump.source	s
provenance.drives.motor.power_up_jump.note	s
provenance.drives.motor.power_up_jump.uncertainty	s
provenance.drives.motor.power_up_jump.refs	a (
provenance.drives.motor.power_up_jump.values	s
provenance.drives.motor.power_up_jump.history	a (
provenance.drives.motor.power_up_jump.history[].date	s
provenance.drives.motor.power_up_jump.history[].change	s
provenance.drives.pullout_48v_rpm_nm	c
provenance.drives.pullout_48v_rpm_nm.source	s
provenance.drives.pullout_48v_rpm_nm.note	s
provenance.drives.pullout_48v_rpm_nm.uncertainty	s
provenance.drives.pullout_48v_rpm_nm.refs	a (
provenance.drives.pullout_48v_rpm_nm.values	s
provenance.drives.pullout_48v_rpm_nm.history	a (
provenance.drives.pullout_48v_rpm_nm.history[].date	s
provenance.drives.pullout_48v_rpm_nm.history[].change	s
provenance.drives.lead_mm/coupling_inertia_kgm2	c
provenance.drives.lead_mm/coupling_inertia_kgm2.source	s
provenance.drives.lead_mm/coupling_inertia_kgm2.note	s
provenance.drives.lead_mm/coupling_inertia_kgm2.uncertainty	s
provenance.drives.lead_mm/coupling_inertia_kgm2.refs	a (
provenance.drives.lead_mm/coupling_inertia_kgm2.values	s
provenance.drives.lead_mm/coupling_inertia_kgm2.history	a (
provenance.drives.lead_mm/coupling_inertia_kgm2.history[].date	s
provenance.drives.lead_mm/coupling_inertia_kgm2.history[].change	s
provenance.drives.model/note/step_s	c
provenance.drives.model/note/step_s.source	s
provenance.drives.model/note/step_s.note	s
provenance.drives.model/note/step_s.uncertainty	s
provenance.drives.model/note/step_s.refs	a (
provenance.drives.model/note/step_s.values	s
provenance.drives.model/note/step_s.history	a (
provenance.drives.model/note/step_s.history[].date	s
provenance.drives.model/note/step_s.history[].change	s
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n	c
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n.source	s
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n.note	s
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n.uncertainty	s
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n.refs	a (

key	t
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n.values	s
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n.history	a (
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n.history[].date	s
provenance.drives.axes[].id/screw_length_mm/screw_inertia_kgm2/rotating_inertia_kgm2/moving_mass_kg/gravity_n.history[].change	s
provenance.drives.axes[].equivalent_mass_kg	c
provenance.drives.axes[].equivalent_mass_kg.source	s
provenance.drives.axes[].equivalent_mass_kg.note	s
provenance.drives.axes[].equivalent_mass_kg.uncertainty	s
provenance.drives.axes[].equivalent_mass_kg.refs	a (
provenance.drives.axes[].equivalent_mass_kg.values	s
provenance.drives.axes[].equivalent_mass_kg.history	a (
provenance.drives.axes[].equivalent_mass_kg.history[].date	s
provenance.drives.axes[].equivalent_mass_kg.history[].change	s
provenance.drives.axes[].friction_n	c
provenance.drives.axes[].friction_n.source	s
provenance.drives.axes[].friction_n.note	s
provenance.drives.axes[].friction_n.uncertainty	s
provenance.drives.axes[].friction_n.refs	a (
provenance.drives.axes[].friction_n.values	s
provenance.drives.axes[].friction_n.history	a (
provenance.drives.axes[].friction_n.history[].date	s
provenance.drives.axes[].friction_n.history[].change	s
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg	c
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg.source	s
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg.note	s
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg.uncertainty	s
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg.refs	a (
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg.values	s
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg.history	a (
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg.history[].date	s
provenance.drives.axes[].l2.rotating_mass_kg/table_mass_kg/motor_mass_kg/screw_mass_kg.history[].change	s
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um	c
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um.source	s
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um.note	s
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um.uncertainty	s
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um.refs	a (
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um.values	s
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um.history	a (
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um.history[].date	s
provenance.drives.axes[].l2.k_axial_n_um/torsion_gj_nmm2/torsion_extra_mm/k_rest_n_um.history[].change	s

key	t
provenance.drives.axes[].l2.coupling_n_um	o
provenance.drives.axes[].l2.coupling_n_um.source	s
provenance.drives.axes[].l2.coupling_n_um.note	s
provenance.drives.axes[].l2.coupling_n_um.uncertainty	s
provenance.drives.axes[].l2.coupling_n_um.refs	a (
provenance.drives.axes[].l2.coupling_n_um.values	s
provenance.drives.axes[].l2.coupling_n_um.history	a (
provenance.drives.axes[].l2.coupling_n_um.history[].date	s
provenance.drives.axes[].l2.coupling_n_um.history[].change	s
provenance.drives.axes[].l2.nut_hertz_n_um15	o
provenance.drives.axes[].l2.nut_hertz_n_um15.source	s
provenance.drives.axes[].l2.nut_hertz_n_um15.note	s
provenance.drives.axes[].l2.nut_hertz_n_um15.uncertainty	s
provenance.drives.axes[].l2.nut_hertz_n_um15.refs	a (
provenance.drives.axes[].l2.nut_hertz_n_um15.values	s
provenance.drives.axes[].l2.nut_hertz_n_um15.history	a (
provenance.drives.axes[].l2.nut_hertz_n_um15.history[].date	s
provenance.drives.axes[].l2.nut_hertz_n_um15.history[].change	s
provenance.drives.axes[].l2.frame	o
provenance.drives.axes[].l2.frame.source	s
provenance.drives.axes[].l2.frame.note	s
provenance.drives.axes[].l2.frame.uncertainty	s
provenance.drives.axes[].l2.frame.refs	a (
provenance.drives.axes[].l2.frame.values	s
provenance.drives.axes[].l2.frame.history	a (
provenance.drives.axes[].l2.frame.history[].date	s
provenance.drives.axes[].l2.frame.history[].change	s
provenance.drives.axes[].l2.friction_motor_n/friction_table_n	o
provenance.drives.axes[].l2.friction_motor_n/friction_table_n.source	s
provenance.drives.axes[].l2.friction_motor_n/friction_table_n.note	s
provenance.drives.axes[].l2.friction_motor_n/friction_table_n.uncertainty	s
provenance.drives.axes[].l2.friction_motor_n/friction_table_n.refs	a (
provenance.drives.axes[].l2.friction_motor_n/friction_table_n.values	s
provenance.drives.axes[].l2.friction_motor_n/friction_table_n.history	a (
provenance.drives.axes[].l2.friction_motor_n/friction_table_n.history[].date	s
provenance.drives.axes[].l2.friction_motor_n/friction_table_n.history[].change	s
provenance.drives.drive	o
provenance.drives.drive.source	s
provenance.drives.drive.note	s
provenance.drives.drive.uncertainty	s

key	t
provenance.drives.drive.refs	a (
provenance.drives.drive.values	s
provenance.drives.drive.history	a (
provenance.drives.drive.history[].date	s
provenance.drives.drive.history[].change	s
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg	c
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg.source	s
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg.note	s
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg.uncertainty	s
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg.refs	a (
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg.values	s
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg.history	a (
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg.history[].date	s
provenance.drives.drive.open_loop_below_rpm/current_loop_hz/bus_capacitance_uf/overvoltage_v/max_phase_advance_deg.history[].change	s
provenance.drives.drive.l3_sub_steps	c
provenance.drives.drive.l3_sub_steps.source	s
provenance.drives.drive.l3_sub_steps.note	s
provenance.drives.drive.l3_sub_steps.uncertainty	s
provenance.drives.drive.l3_sub_steps.refs	a (
provenance.drives.drive.l3_sub_steps.values	s
provenance.drives.drive.l3_sub_steps.history	a (
provenance.drives.drive.l3_sub_steps.history[].date	s
provenance.drives.drive.l3_sub_steps.history[].change	s
provenance.drives.alm	c
provenance.drives.alm.source	s
provenance.drives.alm.note	s
provenance.drives.alm.uncertainty	s
provenance.drives.alm.refs	a (
provenance.drives.alm.values	s
provenance.drives.alm.history	a (
provenance.drives.alm.history[].date	s
provenance.drives.alm.history[].change	s
provenance.drives.power	c
provenance.drives.power.source	s
provenance.drives.power.note	s
provenance.drives.power.uncertainty	s
provenance.drives.power.refs	a (
provenance.drives.power.values	s
provenance.drives.power.history	a (
provenance.drives.power.history[].date	s

key	t
provenance.drives.power.history[].change	s
provenance.drives.l2	o
provenance.drives.l2.source	s
provenance.drives.l2.note	s
provenance.drives.l2.uncertainty	s
provenance.drives.l2.refs	a (
provenance.drives.l2.values	s
provenance.drives.l2.history	a (
provenance.drives.l2.history[].date	s
provenance.drives.l2.history[].change	s
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s	o
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s.source	s
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s.note	s
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s.uncertainty	s
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s.refs	a (
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s.values	s
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s.history	a (
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s.history[].date	s
provenance.drives.l2.zeta_axial/zeta_coupling/servo_damping_ratio/current_lag_ms/smoothing_ms/contact_restitution/impact_speed_mm_s.history[].change	s
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um	o
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um.source	s
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um.note	s
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um.uncertainty	s
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um.refs	a (
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um.values	s
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um.history	a (
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um.history[].date	s
provenance.drives.l2.stribeck_ratio/stribeck_v_mm_s/viscous_n_s_mm/lugre_sigma0_n_um.history[].change	s
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k	o
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k.source	s
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k.note	s
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k.uncertainty	s
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k.refs	a (
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k.values	s
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k.history	a (
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k.history[].date	s
provenance.cutting.materials.alpha_um_m_k/c_j_kg_k.history[].change	s
provenance.spindle.vfd.params	o
provenance.spindle.vfd.params.source	s
provenance.spindle.vfd.params.note	s

key	t
provenance.spindle.vfd.params.uncertainty	s
provenance.spindle.vfd.params.refs	a (
provenance.spindle.vfd.params.values	s
provenance.spindle.vfd.params.history	a (
provenance.spindle.vfd.params.history[].date	s
provenance.spindle.vfd.params.history[].change	s
provenance.spindle.vfd.factory	o
provenance.spindle.vfd.factory.source	s
provenance.spindle.vfd.factory.note	s
provenance.spindle.vfd.factory.uncertainty	s
provenance.spindle.vfd.factory.refs	a (
provenance.spindle.vfd.factory.values	s
provenance.spindle.vfd.factory.history	a (
provenance.spindle.vfd.factory.history[].date	s
provenance.spindle.vfd.factory.history[].change	s
provenance.spindle.vfd.protocol	o
provenance.spindle.vfd.protocol.source	s
provenance.spindle.vfd.protocol.note	s
provenance.spindle.vfd.protocol.uncertainty	s
provenance.spindle.vfd.protocol.refs	a (
provenance.spindle.vfd.protocol.values	s
provenance.spindle.vfd.protocol.history	a (
provenance.spindle.vfd.protocol.history[].date	s
provenance.spindle.vfd.protocol.history[].change	s
provenance.spindle.vfd.reply_latency_ms/min_gap_ms	o
provenance.spindle.vfd.reply_latency_ms/min_gap_ms.source	s
provenance.spindle.vfd.reply_latency_ms/min_gap_ms.note	s
provenance.spindle.vfd.reply_latency_ms/min_gap_ms.uncertainty	s
provenance.spindle.vfd.reply_latency_ms/min_gap_ms.refs	a (
provenance.spindle.vfd.reply_latency_ms/min_gap_ms.values	s
provenance.spindle.vfd.reply_latency_ms/min_gap_ms.history	a (
provenance.spindle.vfd.reply_latency_ms/min_gap_ms.history[].date	s
provenance.spindle.vfd.reply_latency_ms/min_gap_ms.history[].change	s
provenance.spindle.vfd.boot_s/oc_factor/overload_*	o
provenance.spindle.vfd.boot_s/oc_factor/overload_*.source	s
provenance.spindle.vfd.boot_s/oc_factor/overload_*.note	s
provenance.spindle.vfd.boot_s/oc_factor/overload_*.uncertainty	s
provenance.spindle.vfd.boot_s/oc_factor/overload_*.refs	a (
provenance.spindle.vfd.boot_s/oc_factor/overload_*.values	s
provenance.spindle.vfd.boot_s/oc_factor/overload_*.history	a (

key	t
provenance.spindle.vfd.boot_s/oc_factor/overload_*.history[].date	s
provenance.spindle.vfd.boot_s/oc_factor/overload_*.history[].change	s
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v	c
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v.source	s
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v.note	s
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v.uncertainty	s
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v.refs	a (
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v.values	s
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v.history	a (
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v.history[].date	s
provenance.spindle.vfd.bus_capacitance_uf/lu_v/control_off_v/control_w/dc_bus_v.history[].change	s
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port	c
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port.source	s
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port.note	s
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port.uncertainty	s
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port.refs	a (
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port.values	s
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port.history	a (
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port.history[].date	s
provenance.spindle.vfd.model/rated_power_w/rated_current_a/supply/port.history[].change	s
provenance.spindle.motor.model/rated_*/poles	c
provenance.spindle.motor.model/rated_*/poles.source	s
provenance.spindle.motor.model/rated_*/poles.note	s
provenance.spindle.motor.model/rated_*/poles.uncertainty	s
provenance.spindle.motor.model/rated_*/poles.refs	a (
provenance.spindle.motor.model/rated_*/poles.values	s
provenance.spindle.motor.model/rated_*/poles.history	a (
provenance.spindle.motor.model/rated_*/poles.history[].date	s
provenance.spindle.motor.model/rated_*/poles.history[].change	s
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a	c
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a.source	s
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a.note	s
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a.uncertainty	s
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a.refs	a (
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a.values	s
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a.history	a (
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a.history[].date	s
provenance.spindle.motor.rated_slip/breakdown_torque_ratio/no_load_current_a.history[].change	s
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms	c
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms.source	s

key	t
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms.note	s
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms.uncertainty	s
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms.refs	a (
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms.values	s
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms.history	a (
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms.history[].date	s
provenance.spindle.motor.inertia_kgm2/friction_nm/viscous_nms.history[].change	s
provenance.spindle.coolant_pump	o
provenance.spindle.coolant_pump.source	s
provenance.spindle.coolant_pump.note	s
provenance.spindle.coolant_pump.uncertainty	s
provenance.spindle.coolant_pump.refs	a (
provenance.spindle.coolant_pump.values	s
provenance.spindle.coolant_pump.history	a (
provenance.spindle.coolant_pump.history[].date	s
provenance.spindle.coolant_pump.history[].change	s
provenance.spindle.encoder	o
provenance.spindle.encoder.source	s
provenance.spindle.encoder.note	s
provenance.spindle.encoder.uncertainty	s
provenance.spindle.encoder.refs	a (
provenance.spindle.encoder.values	s
provenance.spindle.encoder.history	a (
provenance.spindle.encoder.history[].date	s
provenance.spindle.encoder.history[].change	s
provenance.tool_setter	o
provenance.tool_setter.source	s
provenance.tool_setter.note	s
provenance.tool_setter.uncertainty	s
provenance.tool_setter.refs	a (
provenance.tool_setter.values	s
provenance.tool_setter.history	a (
provenance.tool_setter.history[].date	s
provenance.tool_setter.history[].change	s
provenance.tool_setter.overtravel/contact	o
provenance.tool_setter.overtravel/contact.source	s
provenance.tool_setter.overtravel/contact.note	s
provenance.tool_setter.overtravel/contact.uncertainty	s
provenance.tool_setter.overtravel/contact.refs	a (
provenance.tool_setter.overtravel/contact.values	s

key	t
provenance.tool_setter.overtravel/contact.history	a (
provenance.tool_setter.overtravel/contact.history[].date	s
provenance.tool_setter.overtravel/contact.history[].change	s
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms	c
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms.source	s
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms.note	s
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms.uncertainty	s
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms.refs	a (
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms.values	s
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms.history	a (
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms.history[].date	s
provenance.tool_setter.pretravel_mm/repeat_sigma_mm/reset_hyst_mm/bounces/bounce_ms.history[].change	s
provenance.touch_probe.model	c
provenance.touch_probe.model.source	s
provenance.touch_probe.model.note	s
provenance.touch_probe.model.uncertainty	s
provenance.touch_probe.model.refs	a (
provenance.touch_probe.model.values	s
provenance.touch_probe.model.history	a (
provenance.touch_probe.model.history[].date	s
provenance.touch_probe.model.history[].change	s
provenance.touch_probe.stylus_travel_mm	c
provenance.touch_probe.stylus_travel_mm.source	s
provenance.touch_probe.stylus_travel_mm.note	s
provenance.touch_probe.stylus_travel_mm.uncertainty	s
provenance.touch_probe.stylus_travel_mm.refs	a (
provenance.touch_probe.stylus_travel_mm.values	s
provenance.touch_probe.stylus_travel_mm.history	a (
provenance.touch_probe.stylus_travel_mm.history[].date	s
provenance.touch_probe.stylus_travel_mm.history[].change	s
provenance.touch_probe.geometry	c
provenance.touch_probe.geometry.source	s
provenance.touch_probe.geometry.note	s
provenance.touch_probe.geometry.uncertainty	s
provenance.touch_probe.geometry.refs	a (
provenance.touch_probe.geometry.values	s
provenance.touch_probe.geometry.history	a (
provenance.touch_probe.geometry.history[].date	s
provenance.touch_probe.geometry.history[].change	s
provenance.touch_probe.styli	c

key	t
provenance.touch_probe.styli.source	s
provenance.touch_probe.styli.note	s
provenance.touch_probe.styli.uncertainty	s
provenance.touch_probe.styli.refs	a (
provenance.touch_probe.styli.values	s
provenance.touch_probe.styli.history	a (
provenance.touch_probe.styli.history[].date	s
provenance.touch_probe.styli.history[].change	s
provenance.touch_probe.typical	o
provenance.touch_probe.typical.source	s
provenance.touch_probe.typical.note	s
provenance.touch_probe.typical.uncertainty	s
provenance.touch_probe.typical.refs	a (
provenance.touch_probe.typical.values	s
provenance.touch_probe.typical.history	a (
provenance.touch_probe.typical.history[].date	s
provenance.touch_probe.typical.history[].change	s
provenance.sound.sources	o
provenance.sound.sources.source	s
provenance.sound.sources.note	s
provenance.sound.sources.uncertainty	s
provenance.sound.sources.refs	a (
provenance.sound.sources.values	s
provenance.sound.sources.history	a (
provenance.sound.sources.history[].date	s
provenance.sound.sources.history[].change	s
provenance.sound.sources.pump	o
provenance.sound.sources.pump.source	s
provenance.sound.sources.pump.note	s
provenance.sound.sources.pump.uncertainty	s
provenance.sound.sources.pump.refs	a (
provenance.sound.sources.pump.values	s
provenance.sound.sources.pump.history	a (
provenance.sound.sources.pump.history[].date	s
provenance.sound.sources.pump.history[].change	s
provenance.sound.listener	o
provenance.sound.listener.source	s
provenance.sound.listener.note	s
provenance.sound.listener.uncertainty	s

key	t
provenance.sound.listener.refs	a (
provenance.sound.listener.values	s
provenance.sound.listener.history	a (
provenance.sound.listener.history[].date	s
provenance.sound.listener.history[].change	s
provenance.sound.radiation	o
provenance.sound.radiation.source	s
provenance.sound.radiation.note	s
provenance.sound.radiation.uncertainty	s
provenance.sound.radiation.refs	a (
provenance.sound.radiation.values	s
provenance.sound.radiation.history	a (
provenance.sound.radiation.history[].date	s
provenance.sound.radiation.history[].change	s
provenance.collision	o
provenance.collision.source	s
provenance.collision.note	s
provenance.collision.uncertainty	s
provenance.collision.refs	a (
provenance.collision.values	s
provenance.collision.history	a (
provenance.collision.history[].date	s
provenance.collision.history[].change	s
provenance.collision.groups	o
provenance.collision.groups.source	s
provenance.collision.groups.note	s
provenance.collision.groups.uncertainty	s
provenance.collision.groups.refs	a (
provenance.collision.groups.values	s
provenance.collision.groups.history	a (
provenance.collision.groups.history[].date	s
provenance.collision.groups.history[].change	s
provenance.collision.near_miss_mm	o
provenance.collision.near_miss_mm.source	s
provenance.collision.near_miss_mm.note	s
provenance.collision.near_miss_mm.uncertainty	s
provenance.collision.near_miss_mm.refs	a (
provenance.collision.near_miss_mm.values	s
provenance.collision.near_miss_mm.history	a (
provenance.collision.near_miss_mm.history[].date	s

key	t
provenance.collision.near_miss_mm.history[].change	s
provenance.cutting.materials	o
provenance.cutting.materials.source	s
provenance.cutting.materials.note	s
provenance.cutting.materials.uncertainty	s
provenance.cutting.materials.refs	a (
provenance.cutting.materials.values	s
provenance.cutting.materials.history	a (
provenance.cutting.materials.history[].date	s
provenance.cutting.materials.history[].change	s
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3	o
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3.source	s
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3.note	s
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3.uncertainty	s
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3.refs	a (
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3.values	s
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3.history	a (
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3.history[].date	s
provenance.cutting.material/ktc_n_mm2/krc_n_mm2/kac_n_mm2/kte_n_mm/kre_n_mm/kae_n_mm/kc_plunge_j_mm3.history[].change	s
provenance.cutting.tool_default/tool_geometry	o
provenance.cutting.tool_default/tool_geometry.source	s
provenance.cutting.tool_default/tool_geometry.note	s
provenance.cutting.tool_default/tool_geometry.uncertainty	s
provenance.cutting.tool_default/tool_geometry.refs	a (
provenance.cutting.tool_default/tool_geometry.values	s
provenance.cutting.tool_default/tool_geometry.history	a (
provenance.cutting.tool_default/tool_geometry.history[].date	s
provenance.cutting.tool_default/tool_geometry.history[].change	s
provenance.cutting.tool_strength	o
provenance.cutting.tool_strength.source	s
provenance.cutting.tool_strength.note	s
provenance.cutting.tool_strength.uncertainty	s
provenance.cutting.tool_strength.refs	a (
provenance.cutting.tool_strength.values	s
provenance.cutting.tool_strength.history	a (
provenance.cutting.tool_strength.history[].date	s
provenance.cutting.tool_strength.history[].change	s
provenance.cutting.force_model	o
provenance.cutting.force_model.source	s
provenance.cutting.force_model.note	s

key	t
provenance.cutting.force_model.uncertainty	s
provenance.cutting.force_model.refs	a (
provenance.cutting.force_model.values	s
provenance.cutting.force_model.history	a (
provenance.cutting.force_model.history[].date	s
provenance.cutting.force_model.history[].change	s
provenance.cutting.runout	o
provenance.cutting.runout.source	s
provenance.cutting.runout.note	s
provenance.cutting.runout.uncertainty	s
provenance.cutting.runout.refs	a (
provenance.cutting.runout.values	s
provenance.cutting.runout.history	a (
provenance.cutting.runout.history[].date	s
provenance.cutting.runout.history[].change	s
provenance.cutting.edge	o
provenance.cutting.edge.source	s
provenance.cutting.edge.note	s
provenance.cutting.edge.uncertainty	s
provenance.cutting.edge.refs	a (
provenance.cutting.edge.values	s
provenance.cutting.edge.history	a (
provenance.cutting.edge.history[].date	s
provenance.cutting.edge.history[].change	s
provenance.cutting.wear	o
provenance.cutting.wear.source	s
provenance.cutting.wear.note	s
provenance.cutting.wear.uncertainty	s
provenance.cutting.wear.refs	a (
provenance.cutting.wear.values	s
provenance.cutting.wear.history	a (
provenance.cutting.wear.history[].date	s
provenance.cutting.wear.history[].change	s
provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach	o
provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach.source	s
provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach.note	s
provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach.uncertainty	s
provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach.refs	a (
provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach.values	s

key	t
`provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach.history`	a (
`provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach.history[].date`	s
`provenance.cutting.stock_res_mm/chunk_mm/chunk_ms/force_step/merge_tol_um/pyramid/sweep_reach.history[].change`	s
`provenance.cutting.variation`	o
`provenance.cutting.variation.source`	s
`provenance.cutting.variation.note`	s
`provenance.cutting.variation.uncertainty`	s
`provenance.cutting.variation.refs`	a (
`provenance.cutting.variation.values`	s
`provenance.cutting.variation.history`	a (
`provenance.cutting.variation.history[].date`	s
`provenance.cutting.variation.history[].change`	s
`provenance.cutting.compliance`	o
`provenance.cutting.compliance.source`	s
`provenance.cutting.compliance.note`	s
`provenance.cutting.compliance.uncertainty`	s
`provenance.cutting.compliance.refs`	a (
`provenance.cutting.compliance.values`	s
`provenance.cutting.compliance.history`	a (
`provenance.cutting.compliance.history[].date`	s
`provenance.cutting.compliance.history[].change`	s
`provenance.cutting.dynamics`	o
`provenance.cutting.dynamics.source`	s
`provenance.cutting.dynamics.note`	s
`provenance.cutting.dynamics.uncertainty`	s
`provenance.cutting.dynamics.refs`	a (
`provenance.cutting.dynamics.values`	s
`provenance.cutting.dynamics.history`	a (
`provenance.cutting.dynamics.history[].date`	s
`provenance.cutting.dynamics.history[].change`	s
`provenance.workholding`	o
`provenance.workholding.source`	s
`provenance.workholding.note`	s
`provenance.workholding.uncertainty`	s
`provenance.workholding.refs`	a (
`provenance.workholding.values`	s
`provenance.workholding.history`	a (
`provenance.workholding.history[].date`	s
`provenance.workholding.history[].change`	s
`provenance.workholding.vise_low`	o

key	t
provenance.workholding.vise_low.source	s
provenance.workholding.vise_low.note	s
provenance.workholding.vise_low.uncertainty	s
provenance.workholding.vise_low.refs	a (
provenance.workholding.vise_low.values	s
provenance.workholding.vise_low.history	a (
provenance.workholding.vise_low.history[].date	s
provenance.workholding.vise_low.history[].change	s
provenance.total_mass_kg	c
provenance.total_mass_kg.source	s
provenance.total_mass_kg.note	s
provenance.total_mass_kg.uncertainty	s
provenance.total_mass_kg.refs	a (
provenance.total_mass_kg.values	s
provenance.total_mass_kg.history	a (
provenance.total_mass_kg.history[].date	s
provenance.total_mass_kg.history[].change	s
provenance.viewer	c
provenance.viewer.source	s
provenance.viewer.note	s
provenance.viewer.uncertainty	s
provenance.viewer.refs	a (
provenance.viewer.values	s
provenance.viewer.history	a (
provenance.viewer.history[].date	s
provenance.viewer.history[].change	s
provenance.schema/generated/units/name/coordinates	c
provenance.schema/generated/units/name/coordinates.source	s
provenance.schema/generated/units/name/coordinates.note	s
provenance.schema/generated/units/name/coordinates.uncertainty	s
provenance.schema/generated/units/name/coordinates.refs	a (
provenance.schema/generated/units/name/coordinates.values	s

key	t
provenance.schema/generated/units/name/coordinates.history	a (
provenance.schema/generated/units/name/coordinates.history[].date	s
provenance.schema/generated/units/name/coordinates.history[].change	s

provenance.accuracy

Where the twin's internal error-model parameters come from (the error model is off by default); the records' values are not published.

key	type	unit	what it is
provenance.accuracy	object		
provenance.accuracy.source	string		the category: design, cad (off the CAD model), datasheet, literature (research notes, catalogues), calc (from other values), assumption, measured (on this machine), model (a simulation setting)
provenance.accuracy.note	string		where exactly, and what to measure
provenance.accuracy.uncertainty	string		how far off the values may be, in words ("+-20 %", "exact")
provenance.accuracy.refs	array (3)		references: research notes, datasheets, source files
provenance.accuracy.values	string		a digest of the values the pattern covers: when they change, the history gets an item
provenance.accuracy.history	array (3)		the entry's changes, dated (source, note, uncertainty, values)
provenance.accuracy.history[].date	string		
provenance.accuracy.history[].change	string		
provenance.accuracy.thermal.frame	object		
provenance.accuracy.thermal.frame.source	string		
provenance.accuracy.thermal.frame.note	string		
provenance.accuracy.thermal.frame.uncertainty	string		
provenance.accuracy.thermal.frame.refs	array (2)		
provenance.accuracy.thermal.frame.values	string		
provenance.accuracy.thermal.frame.history	array (1)		
provenance.accuracy.thermal.frame.history[].date	string		
provenance.accuracy.thermal.frame.history[].change	string		
provenance.accuracy.thermal.ambient_dt_k	object		
provenance.accuracy.thermal.ambient_dt_k.source	string		
provenance.accuracy.thermal.ambient_dt_k.note	string		
provenance.accuracy.thermal.ambient_dt_k.uncertainty	string		
provenance.accuracy.thermal.ambient_dt_k.refs	array (0)		
provenance.accuracy.thermal.ambient_dt_k.values	string		
provenance.accuracy.thermal.ambient_dt_k.history	array (1)		
provenance.accuracy.thermal.ambient_dt_k.history[].date	string		
provenance.accuracy.thermal.ambient_dt_k.history[].change	string		
provenance.accuracy.thermal.part	object		
provenance.accuracy.thermal.part.source	string		
provenance.accuracy.thermal.part.note	string		
provenance.accuracy.thermal.part.uncertainty	string		
provenance.accuracy.thermal.part.refs	array (1)		
provenance.accuracy.thermal.part.values	string		
provenance.accuracy.thermal.part.history	array (1)		

key	type	unit	what it is
`provenance.accuracy.thermal.part.history[].date`	string		
`provenance.accuracy.thermal.part.history[].change`	string		
`provenance.accuracy.thermal.dynamic`	object		
`provenance.accuracy.thermal.dynamic.source`	string		
`provenance.accuracy.thermal.dynamic.note`	string		
`provenance.accuracy.thermal.dynamic.uncertainty`	string		
`provenance.accuracy.thermal.dynamic.refs`	array (1)		
`provenance.accuracy.thermal.dynamic.values`	string		
`provenance.accuracy.thermal.dynamic.history`	array (1)		
`provenance.accuracy.thermal.dynamic.history[].date`	string		
`provenance.accuracy.thermal.dynamic.history[].change`	string		

`provenance.drives.axes[].l2.coupling_play_um`

Where the coupling's play comes from; the record's values are not published.

key	type	unit	what it is
`provenance.drives.axes[].l2.coupling_play_um`	object		
`provenance.drives.axes[].l2.coupling_play_um.source`	string		the category: design, cad (off the CAD model), datasheet, literature (research notes, catalogues), calc (from other values), assumption, measured (on this machine), model (a simulation setting)
`provenance.drives.axes[].l2.coupling_play_um.note`	string		where exactly, and what to measure
`provenance.drives.axes[].l2.coupling_play_um.uncertainty`	string		how far off the values may be, in words ("+-20 %", "exact")
`provenance.drives.axes[].l2.coupling_play_um.refs`	array (1)		references: research notes, datasheets, source files
`provenance.drives.axes[].l2.coupling_play_um.values`	string		a digest of the values the pattern covers: when they change, the history gets an item
`provenance.drives.axes[].l2.coupling_play_um.history`	array (2)		the entry's changes, dated (source, note, uncertainty, values)
`provenance.drives.axes[].l2.coupling_play_um.history[].date`	string		
`provenance.drives.axes[].l2.coupling_play_um.history[].change`	string		

1820 keys.

协议

同匠自研内核说 Grbl 1.1 的串口协议，带现在的 Grbl 发送软件用的那些扩展。为 Grbl 写的发送软件都能连。验过的发送软件和它们的设置见 G 代码发送软件。

连接

连哪里	怎么连
Studio 里的孪生	本机的 TCP 端口 2323，一次一个发送软件。
MC-1	它的原生 USB 口，是一个串口（Windows 上是 COM*n*）。波特率不起作用；发送软件要填时填 115200。发送软件拉高 DTR 以后控制器才应答。

行和应答

- **行尾：** LF 或 CR 结束一行。CR LF 或 LF CR 连着只算一个行尾；两个 LF 是两行。
- **应答：** 每一行控制器都答 `ok` 或 `error:N`，按行来的顺序。每个 N 见错误码。
- **行长：** 最多 256 个字符。更长的行答 `error:11`。
- **字符计数：** 接收缓冲区 1024 字节。发送软件最多可以有这么多字节已发出、还没得到应答。状态报告里的 `Bf:` 给出剩下的空间。
- **规划器：** 100 个块。`Bf:` 也给出空着的块数。
- **等着的行：** 进给保持或开门时，行在缓冲区里等着，循环再启动后才得到应答。复位、`0x19`（停止）、`0x03` 或 `0x85`（取消点动）会扔掉还在等的行。
- **出错之后：** 下一行 G 代码不执行，照答同一个错误，直到来一条 `$` 命令、一个空行或复位。

实时命令

实时命令是单个字节。控制器一收到就执行，哪怕正在一行的中间。它不属于任何一行，也没有 `ok`。

字节	命令	做什么
`0x18`（Ctrl-X）	复位	立刻全部停下，清掉排着的东西。轴在走时复位会报 `ALARM:3`：要重新回零。急停按着时不接受。
`0x03`（Ctrl-C）	复位（急停按着时也行）	同 Ctrl-X，急停按着时也接受。
`0x19`（Ctrl-Y）	停止	轴减速停下，扔掉还排着的行。位置不丢，不报警。
`ESC` 再 `0x14`	重启	重启控制器，只在能自己重启的板子上。
`?`	状态报告	一份状态报告。
`0x80`（或 `0x05`）	状态报告	一份状态报告；自动报告开着时不答。
`0x87`	完整的状态报告	一份带全部字段的状态报告。
`~`、`0x81`	循环启动	从进给保持、安全门、`M0`/`M1` 或换刀之后接着走。
`!`、`0x82`	进给保持	轴减速停下并保持。点动中则取消点动。
`0x83`	解析器状态	报出模态，同 `$G`。
`0x84`	安全门	当作安全门打开：保持，设了停车就停车。
`0x85`	取消点动	结束点动，扔掉还排着的点动行。
`0x88`	选择停止	打开或关掉 `M1` 的停止。
`0x89`	单段	打开或关掉单段。
`0x8B`	手轮方式	接了手轮时切换它的方式；没接时不理。
`0x8C`	自动报告	`$481` 设了间隔时，打开或关掉自动状态报告；否则不理。
`0x90`	进给倍率 100 %	
`0x91`、`0x92`	进给倍率 +10 %、−10 %	范围 10–200 %。
`0x93`、`0x94`	进给倍率 +1 %、−1 %	范围 10–200 %。
`0x95`、`0x96`、`0x97`、`0x98`	快移倍率	100 %、50 %、25 %、5 %。
`0x99`	主轴倍率 100 %	
`0x9A`、`0x9B`	主轴倍率 +10 %、−10 %	范围 10–200 %。
`0x9C`、`0x9D`	主轴倍率 +1 %、−1 %	范围 10–200 %。
`0x9E`	主轴停	只在进给保持时：停主轴，或再开起来。
`0x9F`	软急停	立刻停下，报急停的报警。
`0xA0`	冷却：浇注	打开或关掉浇注。
`0xA1`	冷却：喷雾	打开或关掉喷雾。
`0xA3`	换刀完成	确认手动换刀做完了。

别的控制字节，以及 `0x80–0xBF` 里没有命令的字节，控制器都扔掉。

- 注释和消息里的 **UTF-8 文字** 当文字读。属于这种字的 `0x80–0xBF` 字节，从不当成实时命令。
- **`$39=0`**（关掉老式实时字符）：`?`、`!`、`~` 在 `$` 行和注释里是文字，在别处照常是命令。`$39=1`（默认）时它们总是命令。

欢迎行和身份

开机以后、每次复位以后，控制器都说：

```
Grbl 1.1f ['$' or '$HELP' for help]
```

发送软件把以 “Grbl” 开头的行当作控制器重启了。

`$I` 答下面这些行，最后是 `ok`：

行	说什么
`[VER:Twinwright 1.1f.<日期>:<文字>]`	协议版本和它的构建日期。`<文字>` 是 `$I=<文字>` 存的。
`[OPT:…]`	选项字母、规划器的块数（100）和接收缓冲区（1024）。
`[NEWOPT:…]`	选项名：`ENUMS`、`RT+` 或 `RT-`（`$39`）、`HOME`、`ES`、`MPG`、`LATHE`、`REBOOT`、`TC`、`SS`（主轴同步）、`SED`、`RTC`、`SD`、`FS`、`YM`。
`[FIRMWARE:Twinwright]`	固件的名字。
`[SIGNALS:…]`	板子有的控制输入，用字母表示。
`[NVS STORAGE:…]`	设置存在哪里。
`[DRIVER:…]`、`[DRIVER VERSION:…]`、`[DRIVER OPTIONS:…]`、`[BOARD:…]`	板子。
`[KERNEL:<版本>+<构建>]`	控制器的版本和构建：发布版是提交号，开发版是 `dev`。
`[KNCAP:…]`	板子有什么（见下）。
`[PLUGIN:…]`	用发送软件认得的名字说的功能：主轴驱动是 `MODBUS` 和 `HUANYANG VFD`。

板子有什么：KNCAP

`[KNCAP:]` 是一行 `键=值`，用逗号分开。键只往后加，意思永远不变。按键名读，不要按位置读。缺了某个键表示“不知道”。

键	值	意思
`port`	`teensy41`、`stm32h7`、`stm32f4`、`twin`、`host`、`sim`、`unknown`	这是控制器的哪一种构建。
`pv`	一个数	控制器端口接口的版本。
`view`	一个数	控制器状态视图的版本。
`spindle`	`vfd`、`pwm`、`none`	主轴驱动：Modbus、一路 PWM 输出，或者没有。
`encoder`	`0`、`1`	装了主轴编码器。
`sd`	`0`、`1`	板子有 SD 卡槽。
`mpg`	`0`、`2`	手轮口，按它的方式号。
`aux`	`di/do/ai/ao`	辅助输入输出：数字入、数字出、模拟入、模拟出。
`axes`	字母	轴，比如 `XYZ`。
`probe`	`0`、`1`	测头输入。MC-1 的对刀仪也用它。
`link`	`usb`、`uart`、`tcp`、`none`	发送软件那条链路。`usb` 是 USB 串口，DTR 拉高后才应答。

影子模式拿真机和它的孪生的 `[KERNEL:]`、`[KNCAP:]` 对比（见真机和影子模式）。

状态报告

收到 `?`、`0x80` 或 `0x87` 时发一份状态报告。自动报告开着时（`$481`、`0x8C`）也自己发。它是一行：

```
<Idle|MPos:0.000,0.000,0.000|Bf:100,1023|FS:0,0|Pn:XYZ>
```

最前面是状态：

- `Idle`（空闲）、`Run`（运行）、`Jog`（点动）、`Home`（回零）、`Check`（检查模式）、`Sleep`（休眠）、`Tool`（换刀）；
- `Hold:0`（已停住）和 `Hold:1`（正在减速）；
- `Door:0`（门关了，可以接着走）、`Door:1`（停住了，门还开着）、`Door:2`（门开了：保持或停车中）、`Door:3`（门关了，正在恢复）；
- `Alarm`（报警）。

后面的字段看 `$10` 和当时的情况：

字段	意思
`MPos:` 或 `WPos:`	机床坐标或工件坐标（`$10` 第 0 位）。
`Bf:`	空着的规划块，接收缓冲区剩下的字节。
`Ln:`	正在执行的那一段的行号。
`FS:`	进给速度、要的主轴转速、驱动报回来的转速。只能开关的主轴报 `F:`。进给保持时进给是 0。
`Pn:`	有信号的输入：限位按轴的字母，`P` 测头、`D` 安全门、`H` 进给保持、`S` 循环启动、`R` 复位、`E` 急停。
`WCO:`	工件坐标的偏置：隔几份报告一次，变了以后一次。
`Ov:`	进给、快移、主轴的倍率（%）：隔几份报告一次，变了以后一次。
`A:`	附件：`S` 或 `C` 主轴正转或反转，`F` 浇注，`M` 喷雾。
`Sl:`	主轴驱动的负载。
`DTG:`	剩余距离，一直报到轴停下（`$10` 第 13 位）。
`H:`	回过零没有。
`T:`	换刀时的刀号。
`TLR:`	设了参考刀长（`$TLR`）。
`MPG:`	手轮在控制（1）还是没有（0）。
`In:`	上一个 `M66` 的结果。
`WCS:`、`Sc:`、`D:`	工件坐标系、缩放、车床直径方式（完整报告里有）。
`AR`、`AR:`	自动报告关着，或开着和它的间隔（完整报告里有）。
`FW:`	固件的名字（完整报告里有）。
`SD:`	从控制器文件系统跑的程序走到哪了。

控制器发的别的行

行	什么时候
`ALARM:N`	报警。每个 N 见报警码。
`[MSG:…]`	消息：控制器自己的，或者程序里的 `(MSG,…)`、`(PRINT,…)`、`(DEBUG,…)`。
`[GC:…]`	模态，在 `$G` 或 `0x83` 之后。`$10` 第 9 位打开时，每变一次发一行。
`[PRB:x,y,z:1]`	测头触发的那一点，以及碰到了（1）还是没碰到（0）。
`[TLO:…]`	刀长补偿。
`[PARAM:n=值]`	在 `$#=n` 之后。
`[G54:…]` … `[G59.3:…]`、`[G28:…]`、`[G30:…]`、`[G92:…]`、`[T:…]`	在 `$#` 之后：各偏置、存的位置、刀具表。
`>行:ok`、`>行:error:N`	开机时跑了一条开机行（`$N0`、`$N1`）。

`$` 命令

`$HELP Commands` 在控制器上列出它们。

命令	做什么
`$`、`$HELP`、`$HELP <组>`	帮助；命令；一组设置和它们的详细说明。
`$$`、`$<n>`、`$<n>=<值>`、`$$=<n>`、`$+`	全部设置；一个设置；改它；一个设置的详细说明；再列一遍全部设置。
`$SED=<n>`	说明一个设置。
`$#`、`$#=<n>`、`$#=<名字>`	各偏置、探测结果、存的位置；一个参数。
`$G`	模态。
`$I`、`$I+`、`$I=<文字>`	控制器是什么（见上）；全部；和它一起存一段文字。
`$N`、`$N0=<行>`、`$N1=<行>`	看或存开机行。
`$C`	检查模式：只检查 G 代码，什么也不动。复位退出。
`$X`	报警后解锁。
`$H`、`$HX`、`$HY`、`$HZ`	全部轴回零，或一根轴回零。
`$J=<G 代码>`	点动。
`$B`、`$S`、`$O`	跳段、单段、选择停止：开或关。
`$SLP`	休眠。

命令	做什么
`$TLR`、`$TPW`	把上一次探测当作参考刀长；在对刀板上量新刀。
`$RST=$`、`$RST=#`、`$RST=*`	设置回 MC-1 的出厂值（回零和限位都开、它的行程）；清掉偏置；两样都做，并清掉开机行。
`$EA`、`$EAG`、`$EE`、`$EEG`、`$EG`、`$ES`、`$ESG`、`$ESH`、`$E*`	列出报警、错误码、设置分组和设置，按行或按 CSV。
`$SPINDLES`、`$SPINDLESH`	主轴。
`$PINS`、`$PINSTATE`、`$PORTS`	板子的引脚；辅助输入输出；串口。
`$LIM`、`$HSS`、`$LEV`	限位输入、回零开关、最近的输入事件。
`$RTC`、`$RTC=<日期和时间>`	控制器的时钟（ISO 8601）。
`$REBOOT`	重启控制器。
`$F`、`$F+`、`$F=<文件>`、`$F<=<文件>`、`$FR`	列出控制器文件系统里的程序，跑一个，送回一个，从头再来。
`$FD=`、`$FMD=`、`$FRD=`、`$FF=yes`、`$FI`、`$FM`、`$FU`、`$CWD=`、`$PWD`	删除，建或删文件夹，格式化，信息，挂载，卸载，工作文件夹。

全部设置见设置。

文件格式

草稿（第 14 轮）：正文在写，各节还会补。

程序

程序是 G 代码文本文件，用 Grbl 1.1 的写法，同匠的控制器认这种写法。这样的文件 Studio 都能读。它的 CAM 写出的程序是这样的：

```
(Sensor bracket)
(tool T1 flat 6 n3 L20, 6061-T6: S13263, F836, plunge F279, depth of cut 1.20 mm, stepover 2.40 mm)
(program zero: on the blank's top; the machine's top first; T1 is the length reference)
G21 G90 G17 G94 G54
G53 G0 Z0
T1 M6
G53 G0 Z0
M3 S13263
G4 P3 (up to speed)
(face X-2..62 Y-2..42, 0.3 mm deep: 1 layer of 12 passes)
...
M5
M9
G53 G0 Z0
M30
```

部分	是什么
程序头	注释：作业的标题；第一把刀、材料和主要切削参数；程序零点在哪里
模式	`G21 G90 G17 G94 G54`：毫米、绝对坐标、XY 平面、每分钟进给、工件坐标 G54
`G53 G0 Z…`	按机床坐标移到 Z 的最高处：开头、每次换刀和结尾。这个高度是回零后 Z 停的地方。
`T… M6`	换刀。控制器在对刀仪上量新刀。
`M3 S…`、`G4 P…`	开主轴，再暂停一会儿等它转到转速。水冷主轴先开它的水泵（`M8`）。
特征的注释	每个特征开头有一行注释，说明切什么
移动	`G0`、`G1`，圆弧 `G2/G3` 带 `I` 和 `J`。钻孔用 `G81`，啄钻用 `G83`，`R` 平面在顶面上方 1 mm。
结尾	`M5`、`M9`、移到最高处、`M30`

注释写在括号里。行不编号。

Studio 把程序的每个版本存在程序目录（文档/Twinwright Studio/programs）里，以程序名为名的文件夹中：

文件	存什么
`v0001.nc`	程序的文本，按存的时候原样
`v0001.json`	它的版本号、时间（UTC）、谁出的、为什么、从哪个版本来、哈希和行数
`v0001.ops.json`	每个特征变成了什么（工序、策略、刀具、行号、转速、进给）：装夹单读它

作业和装夹

作业是 CAM 出程序所依据的零件描述，是一个 JSON 对象：

键	意思
`name`	程序的名字，比如 `bracket.nc`
`title`、`description`	写进程序头和装夹单的文字
`material`	材料，比如 `6061-T6`；别的标准的牌号也认得
`tool`	主轴上的刀，按规格写（`flat 6 n3 L20`）
`tools`	刀具表：`{"1": "flat 6 n3 L20", "2": "drill 3.4 118 L20"}`。不写时用机床的。
`cutting`	覆盖自动选定值的参数：`rpm`、`feed`、`ap`、`ae`、`finish_ap`；`corner_steps`（true 或 false：壁的内角先分步切）
`safe_z`	横移的高度，从工件零点往上算，单位毫米
`zero`	工件零点在哪里：`front-left-top`（默认）或 `center-top`
`features`	各个特征，按切的顺序。见出程序。
`setups`	代替 `features`：每次装夹一个对象
`nest`	代替 `features`：在板料上排料的零件

每次装夹有自己的 `features`，并写明零件从上一次装夹怎么过来：

键	意思
`turn`	`u`：绕 X 翻面（前后对调）；`v`：绕 Y 翻面（左右对调）
`fixture`	从这次装夹起换一种夹法（`vise`、`clamps`、`tape` …）
`zero`	重新找的工件零点：`front-left-top` 或 `center-top`。翻面或换了夹具又没写时，取 `front-left-top`；否则沿用上一次装夹的零点。
`name`、`tool`、`tools`、`cutting`、`safe_z`	这次装夹自己的，覆盖作业的

```
{"name": "bracket.nc", "title": "Sensor bracket", "material": "6061-T6",
 "tools": {"1": "flat 6 n3 L20", "2": "drill 3.4 118 n2 L20"},
 "features": [
  {"op": "face", "x0": -2, "y0": -2, "x1": 62, "y1": 42, "depth": 0.3, "tool": 1},
  {"op": "pocket", "shape": "rect", "x0": 6, "y0": 6, "x1": 28, "y1": 34, "r": 3, "depth": 4, "tool": 1},
  {"op": "pocket", "shape": "circle", "cx": 44, "cy": 20, "d": "Ø12 H7", "depth": 6, "tool": 1},
  {"op": "drill", "points": [[44, 6], [44, 34]], "depth": 8, "tool": 2}]}
```

作业的每个版本都存在它出的程序旁边，叫 `job-v0001.json`：版本号、时间（UTC）、谁出的（`operator` 或 `ai:<客户端>`）、备注、它出的程序版本，以及作业本身。分几次装夹的作业还列出每次装夹的程序、出的版本和它的哈希。

录制（MCAP）

`--mcap 文件` 把一次运行录成 MCAP 文件（mcap.dev，格式版本 0）：每类数据一个通道，消息是 JSON，每个通道带它的 JSON Schema，不分块、不压缩，文件末尾有统计和 CRC 的摘要。Studio 的测试按 MCAP 规范把文件读回来核对过；PlotJuggler、Foxglove 这类读 MCAP 的工具应当打得开，但还没实际试过。

- `--rec-period 毫秒` 设采样通道多久写一次（不给就是 1 ms）。
- 录制不改变运行：录着的一次和没录的同一次逐位相同。
- `--play 文件.mcap` 在 Studio 里放一段录制，带时间轴：播放、暂停、拖动定位、0.25–10 倍速。放的时候什么都不仿真。

时间

每条消息的记录时间是孪生的虚拟时间，从上电起算的纳秒。多数通道还带 `t`，是同一个时间，单位秒。采样通道只在内容变了的时候才写一条。

文件头

一条名为 `mc1sim` 的元数据记录，说这是一次什么样的运行：

键	意思
`machine`	机床的名字
`drive_model`	这次运行的驱动模型：`L0`、`L1`、`L2`、`L2/3`（三质量）或 `L3`
`f_cpu`	控制器的时钟，Hz
`nvs`	运行开始时控制器所存设置的散列，16 位十六进制
`sample_period_ms`	采样周期
`time`	`virtual, ns from power-on`

通道

通道	什么时候写	字段	类型	单位	意思
`/machine`	采样	`pos`	number[3]	mm	各轴在哪里：机械本身，不是控制器以为的
		`mpos`	number[3] 或 null	mm	控制器的机床位置（MPos）；控制器没开时为 null
		`lost`	number[3]		各轴丢了多少步
		`feed`	number	mm/min	控制器此刻的进给速度
		`tip`	number[3]	mm	刀尖相对工件：孪生的误差模型开着时按它的误差移过，关着时和 `pos` 一样
`/drives`	采样，驱动模型从 L1 起	`ferr_um`	number[3]	µm	各驱动器的跟随误差
		`state`	string[3]		各驱动器的状态
		`km1`	boolean		接触器 KM1 吸合
		`bus48`	boolean		驱动器的 48 V 电源有电
`/spindle`	采样，有变频器时	`f_out`	number	Hz	变频器的输出频率
		`rpm`	number	rpm	主轴的转速
		`amps`	number	A	变频器的输出电流

通道	什么时候写	字段	类型	单位	意思
		`load_pct`	number	%	变频器报的负载
		`torque_nm`	number	N·m	电机的转矩
		`bus_v`	number	V	变频器的直流母线电压
		`cmd_rpm`	number	rpm	控制器要的转速
		`run`、`pump`	boolean		变频器在运行；水泵开着
		`power`	string		变频器自己的供电：`ready`、`boot` 或 `off`
		`fault`	string		变频器的故障，照它面板上显示的
`/controller`	变化时	`state`	string		控制器的状态（没开时为 `Off`）
		`alarm`	integer		当前的报警（0：没有）
		`homed`	integer		已回零的轴，每轴一位：X 1、Y 2、Z 4
		`limits`	integer		有效的限位输入，每路一位
		`probe`、`estop`、`motor_fault`、`door`	boolean		探针、急停、电机故障、门的输入（MC-1 没有门）
		`on`	boolean		控制器有电
`/power`	变化时	`mains`、`v24`、`mcu`、`km1`、`bus48`	boolean		市电；24 V 电源；控制器在运行；KM1；48 V 电源
		`vfd_mains`、`vfd`、`pump`	boolean、string、boolean		变频器的市电；它的状态（`none`、`ready`、`boot`、`off`）；水泵
`/serial/in`	来的时候	`text`	string		控制器收到的字节；控制字符写成 `\u` 转义
`/serial/out`	来的时候	`line`	string		控制器写出的每一行
`/events`	发生的时候	`text`	string		事件日志的每一条：标记、供电、驱动器、变频器、碰撞、断线
`/cut`	采样，有毛坯时	`cutting`	boolean		正在切下材料；不切时下面的数除 `rpm` 和 `removed_mm3` 外都是 0
		`mrr_mm3_s`	number	mm³/s	材料去除率
		`ap_mm`、`ae_mm`	number	mm	轴向切深和径向切宽
		`fz_mm`	number	mm	每齿进给
		`rpm`	number	rpm	主轴的转速
		`fx_n`、`fy_n`、`fz_n`	number	N	沿机床各轴的切削力
		`torque_nm`、`power_w`	number	N·m、W	切削的转矩和功率
		`kc_j_mm3`	number	J/mm³	比切削能
		`removed_mm3`	number	mm³	到此刻一共去掉的材料
`/input`	来的时候	`line`	string		会话的输入，一行一行，照 `--record` 写的（见会话日志）
`/shadow`	来的时候，影子模式下	`host_s`	number	s	电脑的时间
		`dir`	string		`t` 发往真机，`r` 从真机来
		`data`	string		那些字节，每个字节一个字符 U+0000–U+00FF，一条消息最多 1 KB
`/ui/state`、`/sound`	给播放用				界面的状态帧（变化时每秒 30 次）和声音的帧（变化时每 10 ms 虚拟时间一次）。是 Studio 自己用的：字段随版本可能会变。

引脚波形（VCD）

`--vcd 文件` 把控制器的引脚写成 VCD 文件（IEEE 1364），GTKWave、PulseView 读得了：作用域 `mcu` 是控制器的引脚，每个变化记在它发生的那一拍（步进脚画成 `$0` 微秒宽的脉冲）；作用域 `machine` 是控制器外面的线：各路电源、KM1、变频器、水泵、急停按钮、对刀仪、五只开关。时间刻度 1 ns。

会话日志

`--record 文件` 把一次会话的输入连同各自的虚拟时间写下来：发送软件发的、实时命令、界面的命令。`--replay 文件` 把这次会话再跑一遍，逐位相同，前提是同一版本的 Studio、同样的机床文件。`--replay 文件 --branch 秒` 重放到那一秒，再从那里实时接着走；加 `--record` 时分叉有自己的日志，从父日志开始（`# branch <父日志> <拍数>`）。日志是文本；它的行是 Studio 自己用的，随版本可能会变。

世界存档

孪生在两次会话之间留着的东西，就像真机关了再开：各轴停在哪里、毛坯和切过的地方、主轴上的刀、控制器的存储和板上时钟，还有热——关机这段时间里慢慢凉下来。控制器是冷启动：机床要重新回零。

窗口把每台机床的世界放在状态文件夹里（Windows 上是 `%LOCALAPPDATA%\Twinwright\Studio`）：第一台的在 `world`，其他的在 `machines\<编号>\world`。`--world 目录` 用别的文件夹；`--world-reset` 从出厂状态开始。批量运行、测试、重放、放录制、AI 的后台机器都不留世界，除非给了 `--world`。同一个世界同时只有一个程序能打开；第二个从出厂状态开始，什么也不存。

文件	存的是什么
`world.json`	会话结束时的样子（见下），先写成 `world.json.tmp` 再改名
`stock.snap`	毛坯的格子，切过的和没切的（二进制，`TWSTOCK1`）
`eeprom.bin`	控制器的存储：它的设置、偏置和刀具数据，一变就写进去

`world.json`：

键	意思
`world`	格式的版本：1
`saved_at`、`saved_epoch`	什么时候存的：UTC，以及 1970 年以来的秒数
`version`	存它的 Studio 的版本
`pos`	各轴停在哪里，mm（X、Y、Z）
`tool`	主轴上的刀，写成刀具规格（`flat 6 n3 L20`）
`tool_table`	作业的刀具表：`{"1": "flat 6 n3 L20", ...}`
`tool_auto`	换刀时孪生替操作员做：按刀具表装上刀、按循环启动
`setup`	装夹和毛坯，和窗口的"保存装夹"写的一样（null：没有）
`cells`	`stock.snap` 里存着毛坯的格子
`thermal`	热：`screws`（各丝杠高于机架）、`frame`（高于 20 °C）、`ambient`（室温高于 20 °C）、`part_heat` 和 `part_offset`（工件高于机架：切削带进去的，以及给定的偏置），单位都是 K；`spindle_um` 是主轴的伸长，单位 µm
`rtc`	板上时钟的读数，s
`probe_cal`	测头最近一次标定的结果：球的直径、有效半径、三瓣和它的相位、环规的直径、最大偏差、什么时候、谁标的（标定过才有）
`odometer`	各轴的 `travel_mm` 和 `reversals`，`spindle_h`（主轴转过的小时数）和 `on_h`（程序开着的小时数），跨会话累计

下次开机时，各轴还在原处；同一块毛坯带着它的格子回来，刀和刀具表照旧；热按关机的时间凉下来，各部分按各自的时间常数回到静止的状态；板上时钟走过了这段时间。`--world-off 秒` 指定关了多久（测试用；不给就是存档以来的时间）。启动时给了 `--pos`、`--tool`、`--stock`、`--setup`、`--part`，就用它们，不用存档里的。

授权文件

授权文件（`.twlic`）是一个 UTF-8 的 JSON 对象，用 Ed25519 签名。Studio 在本机核验，不用联网。文件放在哪儿、怎么用：见授权、席位和激活。

下面这个例子取自共用的测试向量，用测试密钥签（`kid` 是 `test`），正式发布的版本不认它：

```
{"v": 1, "kid": "test", "id": "TWL-TEST000001", "edition": "pro", "kind": "subscription", "seats": 1,
 "holder": "Weber Lab, TU Example", "issued": "2026-11-02T10:00:00Z", "valid_until": "2027-11-02T10:00:00Z",
 "renew_after": "2027-10-28T10:00:00Z", "updates_until": null, "fp": ["0123456789abcdef0123456789abcdef"],
 "mc1_serial": null,
 "sig": "h3FTqZnDYtq8Q7_0ZuELLeDrTfyZnkTPgvDWSywTyl7BCJJxGIyhrBQUjYHJiE1_WOWaFarqbveFiMLrEfUyCA"}
```

字段

字段	类型	意思
`v`	整数	格式版本，是 1。别的版本的文件，Studio 按无效处理。
`kid`	字符串	用同匠哪一把密钥签的
`id`	字符串	授权编号：`TWL-` 加 10 位字母和数字。续期时发的就是它。
`edition`	字符串	`maker` 或 `pro`
`kind`	字符串	`subscription`（订阅）、`perpetual`（买断）或 `owner`（MC-1 机主）
`seats`	整数	席位数
`holder`	字符串	授权给谁：个人或单位的名字，从不放电子邮箱
`issued`	日期时间	签发时间，UTC，`YYYY-MM-DDTHH:MM:SSZ`
`valid_until`	日期时间或 null	订阅：已付费那一期的结束。买断：第一个文件里是退款期的结束，永久文件里是 null。
`renew_after`	日期时间或 null	Studio 从什么时候开始试着续期
`updates_until`	日期或 null	买断、MC-1 机主：这一天以前发布的版本能用付费功能

字段	类型	意思
`fp`	字符串数组	绑定的电脑：它们指纹的散列，每个席位最多三个
`mc1_serial`	字符串或 null	MC-1 机主：机器的序列号。Studio 运行时不查机器。
`sig`	字符串	其余字段的 Ed25519 签名，base64url，不补 `=`

Studio 不认识的字段也在签名里；Studio 不管它的意思。

签名的字节

签名覆盖去掉 `sig` 以后的文件，按规范的写法写出来：

- 键按 Unicode 码点排序；
- 没有空白；
- 字符串用 JSON 最短的转义；
- 整数不带小数点；不许有别的数，也不许同一个键出现两次。

Studio 核验签名之前，先把文件按这个写法写一遍。所以文件里的空格、换行和键的顺序都不要紧；改了任何一个值，核验就不过。

指纹的散列

散列是对这些字节算 SHA-256 后取前 32 位十六进制：`twinwright-fp-v1`、一个零字节、电脑的 ID。

系统	电脑的 ID
Windows	`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography` 下的 `MachineGuid`
Linux	`/etc/machine-id`

两样都没有时，Studio 生成一个随机 ID，存在授权文件夹里（`machine.id`）；删了它就等于换了一台电脑。ID 本身从不离开电脑。

例如 ID `8b1c3a0e-5f2d-4c7a-9e61-2d0f4b7a1c55` 的散列是 `0a2d31bb82176516415efa7b3a340562`。

请求码

离线激活用的请求码，是规范写法的 UTF-8 JSON `{"fp": <散列>, "version": <Studio 的版本>}` 做 base64url，不补 `=`。例如散列 `0123456789abcdef0123456789abcdef`、版本 1.0.0 得到：

```
eyJmcCI6IjAxMjM0NTY3ODlhYmNkZWYwMTIzNDU2Nzg5YWJjZGVmIiwidmVyc2lvbiI6IjEuMC4wIn0
```

`TwinwrightStudio.exe --licence-request` 打出这台电脑的请求码。

试用文件

试用就是一份授权文件，和别的授权文件一样由同匠的授权服务器签名：`"kind": "trial"`，`fp` 里是这台电脑的散列，`valid_until` 是开始试用后 30 天。试用文件从不续期。手改过的试用文件不再有效。

机床档案

机床档案的每一项，带 MC-1 的值：机床档案格式。

用发送软件连同匠自研内核

这一页写给想用自己的发送软件（bCNC、gSender 或 Universal G-code Sender，下称 UGS）、而不是用 Studio 来开 MC-1 的人。MC-1 的控制器是同匠自研内核（兼容 Grbl 1.1 串口协议，带现在的发送软件用的那些扩展），所以这几个发送软件都能用，只是要注意下面几处设置和限制。

这里写的都在 MC-1 的数字孪生上实测过；孪生跑的是和机床同一份控制器代码。真机上的核对还没做。测过的版本：bCNC 0.9.16、gSender 1.6.4、UGS Platform 2.1.27。

连接

连哪里	怎么连
Studio 窗口里的孪生	Studio 在 `localhost:2323` 等发送软件，一次只接一个，只对本机开放（启动 Studio 时加 `--bind any`，局域网里才连得上）。bCNC：端口填 `socket://localhost:2323`。gSender：选以太网，地址 `127.0.0.1`，端口 `2323`。UGS：`tcp://localhost:2323`。
MC-1 本身	原生 USB：控制器在电脑上是一个串口（Windows 上是 COM*n*），选它就行。原生 USB 不看波特率，有的发送软件非要填一个，填 115200 即可。

控制器怎么报自己：

- 欢迎行是 `Grbl 1.1f ['$' or '$HELP' for help]`，和经典的 Grbl 控制器一样。发送软件靠以 "Grbl" 开头的欢迎行认出控制器重启了（见第 3 节）。
- 名字在 `$I` 的回答里：`[VER:Twinwright 1.1f.…]`、`[FIRMWARE:Twinwright]`。会显示固件名的发送软件显示的是 "Twinwright"。

发送软件里选哪种固件

发送软件	设置	原因
bCNC 0.9.16	控制器选 **GRBL1**	bCNC 不读固件的名字。测的是 GRBL1。
gSender 1.6.4	默认固件选 **Grbl** 以外的那一项	gSender 靠 `$I` 里的名字认固件。我们的控制器报的是自己的名字，gSender 就按你设的默认固件工作。测的是那一项。
UGS 2.1.27	固件选 **GRBL**	测过：设置、跑程序、报警和复位后回零都正常，复位后不用多按 Soft Reset（第 3 节）。UGS 的另一种固件选项也能用，但复位后要多按一次 Soft Reset；点动只在那一项下测过。

报警和复位以后

- **急停。** MC-1 的急停会断开动力接触器。松开急停，按机床上的启动按钮，再在发送软件里复位、回零。不按启动，报警（ALARM:10）清不掉。
- **ALARM:11（要回零）。** 回零（`$H`）。用 `$X` 解锁会被拒绝（error:46）。
- **ALARM:2（软限位）。** 程序要求的移动超出了行程。回零以后 Z 停在行程的最高处（机床坐标 Z 0），比它再高的移动，比如 `G53 G0 Z5`，都会报 ALARM:2。复位，再回零或解锁。
- **bCNC** 任何复位以后都能恢复：它认出以 "Grbl" 开头的欢迎行，丢掉复位前还在路上的那些行，所以 `$H` 发得出去。
- **bCNC 按"停止"之后。** bCNC 的停止是先进给保持、再复位。复位以后 bCNC 把停止前的模态（坐标系、单位、刀长补偿）用一两行发回控制器，有几种情况控制器会拒收，和别的 Grbl 控制器一样：刀长补偿 `G43.1` 生效时，那一行报 error:24（bCNC 头一次停止以后自己会补发一行 `G43.1Z…`，所以从第二次停止起每次都会遇到）；在 `G80` 下，`G43.1Z…` 报 error:31；缩放 `G51` 生效时，`G51:X` 报 error:1。出了错，后面的 G 代码行都回同一个错，直到一个空行、一条 `$` 命令或者复位。所以按了停止以后，先用 `$G` 和 `$#` 看坐标系、单位和刀长补偿对不对，再点动或者续跑。对不上就重新设零、重新量刀长（2026-10-06 在孪生上测过）。
- **UGS 选另一种固件时。** 复位如果不是 UGS 自己发的——比如从 Studio 的窗口复位，或者控制器自己重启——回零之前要在 UGS 里**再按一次 Soft Reset**，不然按 Home 没有反应。原因是 UGS 选那种固件时，只把另一种欢迎行当作重启。用 UGS 自己的 Soft Reset 复位就没有这个问题。MC-1 本身没有复位输入、只有一个 USB 主机，机床上很少遇到这种复位。UGS 的固件选 "GRBL" 就没有这个问题：这种复位以后 UGS 显示 CONNECTING，接着直接按 Home 就能回零（2026-10-06 在孪生上测过）。

换刀（M6）

三个发送软件在默认设置下，都不会让 MC-1 换刀并量新刀：

发送软件	默认的做法	怎么办
bCNC	跳过每一个 M6 行：下一把刀的活用主轴上原来那把刀切。	换刀方式设成**手动换刀（刀长补偿）**，在 MC-1 的对刀仪上量刀（第 5 节）。
gSender	换刀方式是 "Ignore"：M6 当注释发出去，程序不停。	换刀方式设成**固定对刀仪（Fixed Tool Sensor）**，位置是 MC-1 的对刀仪（第 5 节）。
UGS	自己处理 M6：抬刀、停主轴、让你换刀，但从来不量刀。换上的刀长一些，就会切得太深，真机上可能撞进工件。	一个程序只用一把刀，每个程序前设好刀长或工件零点；或者带换刀的程序用 Studio 来跑。

UGS 和 M6 还有几点：

- 注释里的 "M6" 它也当成换刀，比如 FreeCAD 写的 `( M6 T1 )`。
- 刚换完一次刀，紧跟着的下一个 M6 可能被它漏掉，直接发给控制器。控制器于是停在换刀状态（Tool），等一个 UGS 永远不会发的确认，UGS 窗口里的按钮从此都按不了。MC-1 上只能按急停收场：按下、松开、按启动，再在 UGS 里 Soft Reset、回零。
- UGS 跑程序途中、还没到 M6 时控制器被复位，UGS 可能把作业记成"已完成"，可它的换刀对话框照样会弹出来；把它按掉，整个程序可能从头再跑一遍。作业中途复位以后，先在 UGS 里关掉文件再往下做。

在 MC-1 的对刀仪上量刀长

对刀仪在工作台左后角，`G59.3` 存的就是它正上方的位置。

- **bCNC，手动换刀（刀长补偿）。** 换刀位置在对刀仪上方，探测位置 `G59.3`；探测距离 50 mm，快探 200 mm/min，慢探 25 mm/min（MC-1 的 `$342`–`$344`），探测命令 G38.2。跑程序之前先按"校准"量第一把刀，之后每换一把刀，bCNC 都会设 `G43.1`。
- **gSender，固定对刀仪。** 对刀仪位置 `G59.3`，在向导里先量初始刀。之后每换一把刀，gSender 不用刀长补偿，而是用 `G10 L20` 挪工件的 Z。它按探到以后轴停下的位置定刀长，不按控制器报的触发点，所以每换一次刀都可能差一点。在孪生里换刀时用 1 倍速：gSender 隔一段固定的短时间才去读这个位置，孪生跑得比实时快，它读到的可能还是停下之前的。
- **换刀开始时，Z 至少要比最高处低 0.1 mm**（机床坐标 Z −0.1 或更低）。这两个发送软件量完刀都会回到这个高度。如果就从最高处开始，量刀那一点点散布可能让回去的目标高出软限位，作业还没开切就停在 ALARM:2。

用测头定工件零点

- **UGS**（它的探测模块）按控制器报的触发点（`[PRB:]`）算零点，这是对的做法。
- **gSender**（3D 测头）和 **bCNC**（探测页，再按 Z=0、X、Y 按钮）把零点设在探到以后轴停下的地方。机床总要越过触发点走一小段才停住，最后一下探得越快，越过得越多。最后一下要慢；或者用 Studio 的测头循环定零点，它按触发点算。
- **UGS 的孔中心**（探测模块的 Hole Center）：从你放下测头的地方起，向 −X、+X、−Y、+Y 各探"填的孔径的一半"，取两次慢探的触发点的中点，孔中心就设在那里。它能探多远取决于填的孔径：起点离孔中心最多 =（填的孔径 − 实际孔径）/2 + 测球半径。照实填孔径时，起点要放到离中心不到一个测球半径（Ø2 的球就是 1 mm）；把孔径填大几毫米（孔 20 填 24 到 26），就给手放留出了余量。
- **UGS 的孔中心探不到孔壁时**（发现 192，UGS 自己的行为）：那一下报 ALARM:5，UGS 停下。但它在开探之前已经把 G54 的 X0 Y0 设在了起点，循环没走完，G54 就留在起点。解锁以后一定先重新定零点，再加工。它的探测循环也卡在半路，断开再连上才能再用。另外，控制器报的没碰到的探测点（`[PRB:…:0]`），它也照样当触点收下。
- 先标定测头（Studio 的测头循环：环规）。没标定的测头量孔会偏大、量凸台会偏小，偏两个预行程。

另外要知道的

- **bCNC 走网络。** Windows 上 bCNC 读 TCP 连接大约每 2 秒才一次，它显示的可能是 2 秒前的状态。bCNC 显示"空闲"不等于机床已经停了。走 USB 没有这个延迟。
- **gSender 的 M0。** 程序前 10 行里的 M0 不会暂停（gSender 把它改成了注释）。M1 总会暂停。
- **gSender 的连续点动。** 按住点动按钮在 MC-1 上会被拒绝（error:15），因为 gSender 要的是一段超出行程的长点动。用一步一步的点动。`$40=1`（限制点动命令）在有的 Grbl 控制器上会让长点动停在软限位上、不拒绝；MC-1 的控制器还不认这个设置，照样拒绝（2026-10-06 在孪生上和 gSender 一起试过）。
- **UGS 的长行。** 超过 127 个字符的行 UGS 永远不发，作业就无声地停住。行要短一些（UGS 会先去掉注释再算长度）。
- **UGS 和 M30。** UGS 会去掉 M30。靠 M30 来停主轴和冷却的程序，跑完后主轴、冷却还开着：程序结尾在 M30 前写上 M5 和 M9。
- **UGS 和注释。** 一行里有两段注释时，UGS 连它们中间的代码也一起删掉。
- **作业中断开连接。** 控制器会把已经收到的行照样跑完。gSender 和 UGS 断开时什么停止都不发，机床接着走，主轴也接着转。断开之前先停下作业。bCNC 断开前会先进给保持、再复位。
- **注释里的非 ASCII 字符**（中文、"Ø"、"–" 等）：三个发送软件发送前都会去掉注释，这些字到不了控制器。控制器自己把注释里和 `(MSG,...)` 里的 UTF-8 字当文字读，从不当实时命令。