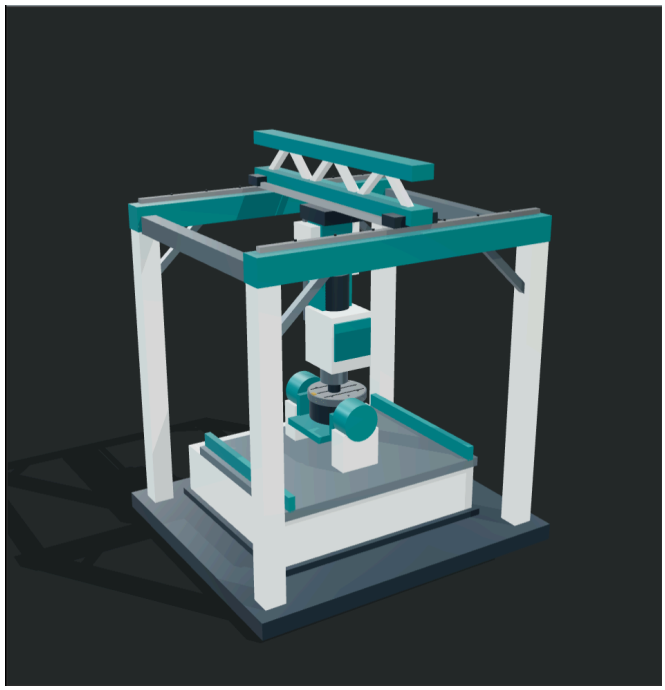
五轴 RTCP 与工件坐标

从坐标零点到 G 代码，再到每根轴的运动

面向初学者的完整学习路径：先认识“相对谁量位置”，再理解“角度改变后，控制器如何保持刀尖相对工件的目标关系”。



本项目原创 XYZBC 桁架模型：刀具侧 XYZ，工件侧 B→C。

你将学会回答

工件原点放在哪里？X/Y/Z方向由谁确定？G54有没有让坐标跟着转台转？为什么一个X10有时会让多根轴运动？为什么只改C角度也会引起XYZ联动？

包含：基础概念、图解、逐字解析、逐行数值演算、实验步骤、排错方法、练习和答案。
建议分 4–6 次授课，也可按目录自学。

读完后，应能把“程序写的数值”和“电机实际要到的位置”分开解释。

导读

怎样使用这份讲义

不要求你先会矩阵或五轴编程。先把坐标看成贴在物体上的尺子，再把RTCP看成依据机床结构换算位置的计算过程。公式附录可以最后再读。

三种标签，避免把不同层次混在一起

层次	在讲义中怎样使用
通用概念	坐标、相对位置、程序段、绝对/增量、物理关节等基础知识。
本课模型	XYZ移动刀具；A/C或B/C转动工件；按本项目正方向和父子层级计算。
已实测算例	通过本项目WASM解释器、运动学和规划器验证的数值；不是手工猜测。

推荐阅读顺序

1. 第1–3章：先弄清坐标系、原点、方向及工件偏置。
2. 第4–6章：理解RTCP，再学习G代码各地址字与模式。
3. 第7章：跟着一份完整程序逐行计算，自己复算。
4. 第8–10章：操作实验、排错、练习；最后查数学附录。

本项目仿真只覆盖LinuxCNC的一部分能力。讲到G52/G92、坐标旋转、完整刀补等功能时，会区分官方控制器语义和项目已验证的范围。

> 不要把某个控制器或示例的约定，当作所有机床都必须遵循的规则。

学习地图

内容索引

本书从基础坐标开始，逐步建立程序与关节之间的计算链。PDF目录书签可直接跳转章节。节号不是PDF页码；前面另有封面和导读。

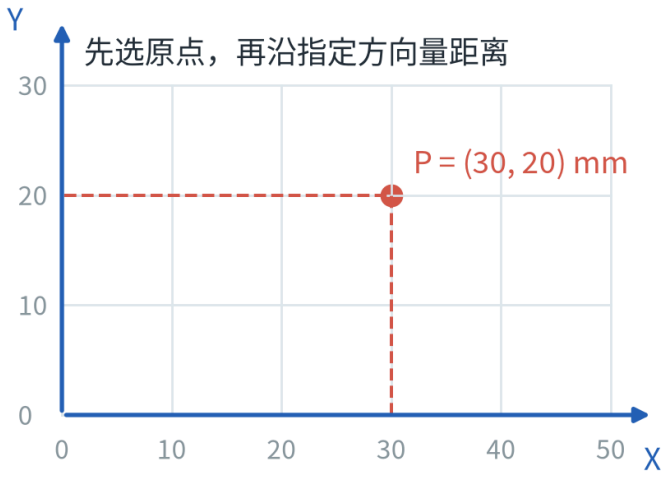
章节	PDF起始页
第1章　坐标与机床基础	4
第2章　机床、工件与中间参考	10
第3章　原点和方向如何确定	15
第4章　G54、G10与其他偏置	21
第5章　RTCP的点、方向与标定	27
第6章　G代码逐字与模态解析	34
第7章　完整程序逐行实算	44
第8章　连续轨迹、刀轴与显示	54
第9章　课堂实验、限位与排错	58
第10章　问答、练习与答案	63
附录A　术语表	68
附录B　可选数学推导	69
附录C　软件验证范围	71
附录D　资料来源与复现	72

建议每次学习1–2章，完成相应小实验后再继续。数值实算章建议边读边写出W、K、角度、J四列。

> 先学概念，后读代码；遇到陌生词可跳到附录术语表。

01　坐标是“按照约定量出来的位置”

想象一张方格纸。先指定一个交点为原点O，再规定向右是+X、向上是+Y，每格代表10mm。这样你就能用两个数告诉别人某个孔的位置，而不必每次都画整张图。



P=(30,20)mm：从原点沿+X量30mm，再沿+Y量20mm。

这里的30和20不是电机编号，也不是速度。它们是点P相对于这套尺子的两个坐标分量。换一个原点，或者把尺子的方向转动，同一个点的坐标数字就会改变。

> **最重要的习惯**
>
> 说“X等于30”之前，先补一句：相对于哪一个坐标系？单位是什么？如果这两点没说明，数字还不能唯一描述空间位置。

> 坐标数字必须和“原点、方向、单位”一起理解。

02　一套坐标系至少要规定什么？

要素	初学者理解	不说明会怎样
原点	从哪里开始量。	相同数值落在不同位置。
方向	X、Y、Z分别朝哪边。	相同增量可能走向不同方向。
单位	例如长度mm、旋转角度°。	X1可能是1mm，也可能是25.4mm。
正负号与手性	三轴之间的正向约定。	模型可能镜像、旋转方向可能相反。

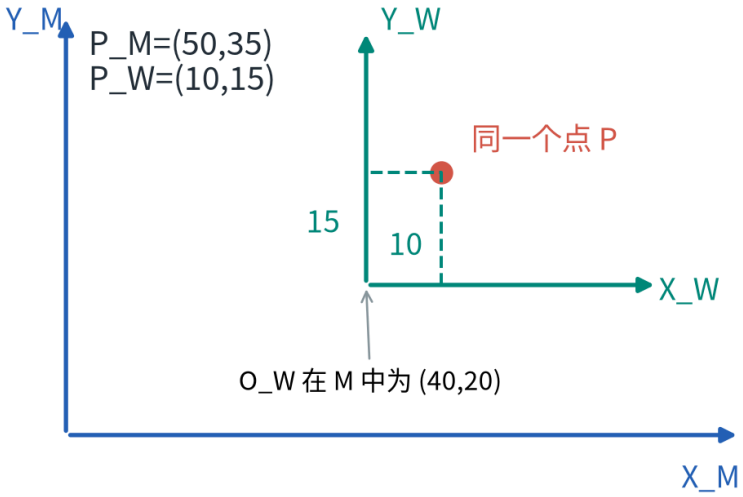
三维直角坐标系有三条互相垂直的轴。一般按右手规则建立：+X和+Y确定后，+Z的方向也随之确定。不能任意把其中一根轴反向，同时还声称保持同一套右手坐标。

坐标系在图上常画成三根箭头；在控制器中，它由参数和变换关系表示。屏幕上的箭头只是它的可视化，拖动摄像机不会改变机床的真实坐标约定。

> **单位提醒**
>
> 机床运行示例使用mm和度；GLB/URDF采用米与弧度。1m=1000mm，180°=πrad。软件应显式转换，不能只把数值原样传过去。

三个方向不是三条新导轨，而是描述位置的三个基准方向。

03　同一个点，可以有两套不同的坐标



工件原点在机床中为(40,20)。同一P点：机床坐标(50,35)，工件坐标(10,15)。

假设两套坐标的方向相同，只是原点平移。绿色工件原点比蓝色机床原点多了X40、Y20。于是工件上的P=(10,15)，在机床坐标中就是(50,35)。

> 机床中的位置 = 工件原点的位置 + 工件内的坐标

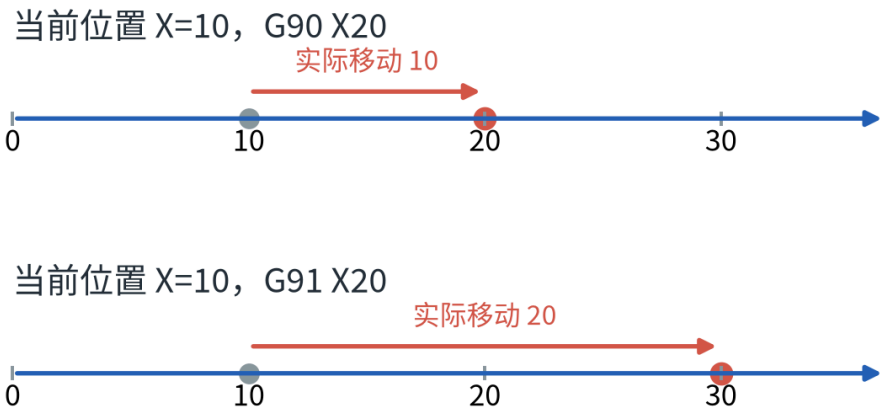
反过来，若已知机床中的点位置，需要减去工件原点，才能得到工件内的读数。这就是最简单的“工件偏置”直觉。

> **注意适用条件**
>
> 这一页只有平移，方向没有转动，也没有刀长补偿。后面工件发生旋转时，还必须先把工件方向换算到机床方向，不能始终只做三个数相加。

> 点没有动，坐标数字也可以变；改变的是量它的尺子。

04　点坐标与移动增量不是一回事

“去X20”描述一个目的地；“向+X再走20”描述一段位移。当前位置不同，这两句话会产生不同结果。



当前位置X=10：绝对目标X20移动10；增量X20移动20，终点到X30。

点坐标需要一个原点。位移向量描述从一个点到另一个点的差，原点平移不会改变这段差值；但是坐标方向转动，会改变它在另一套坐标中的分量。

写法	意思
p=(10,5,2)	一个目标点的位置。
Δp=(10,0,0)	沿当前编程+X方向增加10。
X轴方向向量	只说明方向；不能直接当作某个目标点。

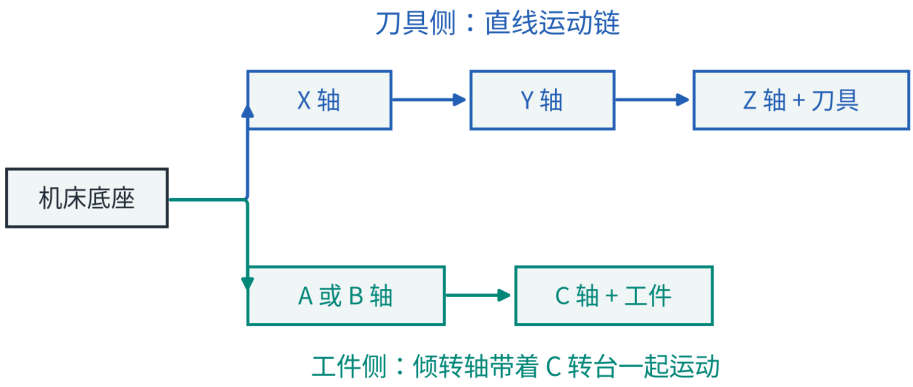
后面会反复用到：p表示点，Δp表示增量。不要因为都含X/Y/Z三个数就把它们混用。

> “目标在哪里”和“这一步走多远”是两个问题。

05　五轴机床里，到底是哪五根轴?

本课选择双转台桁架结构：三根直线轴X、Y、Z让刀具移动；两根旋转轴让工件转动。AC型使用A和C，BC型使用B和C。它们不是XYZABC六根独立轴都同时存在。

坐标字母	典型含义	本课执行部件
X/Y/Z	直线位置	横梁、滑座、滑枕/刀具
A	绕X方向的轴线转动	AC外层倾转台
B	绕Y方向的轴线转动	BC外层倾转台
C	绕本地Z方向轴线转动	安装在倾转台上的内层转台



两条独立分支：刀具侧XYZ；工件侧A/B→C。

C的轴线装在A/B上，所以A/B倾转会连C的轴线一起带走。这不是软件的小细节，而是决定运动学计算次序的机械事实。

> 先查结构图，确认谁安装在谁上面，再讨论坐标如何变换。

[S1][S3]；本项目桁架模型。

06　关节位置，不等于工件坐标

把“关节”理解为一个执行机构及它的位置。在本项目里，J0/J1/J2对应XYZ直线轴，J3是A或B，J4是C。每个关节位置属于机构自身的运动参数。

对象	回答什么问题
程序点	刀尖相对于选定编程参考系要到哪里?
关节指令	各执行轴应该到哪里?
实际反馈	编码器报告机械轴实际上在哪里?
三维模型刀尖	将关节位置代入结构模型后，画面上的刀尖在哪里?

简单三轴、没有偏置时，这几类数字可能很相近，容易造成“它们本来就是同一回事”的错觉。五轴RTCP会把这种差别明显表现出来：程序XYZ不变，关节XYZ却需要变化。

> **理想仿真与真实机床**
>
> 本项目通常把关节指令和理想反馈看成相同。真实机床还存在跟随误差、伺服滞后、机械误差和测量误差，不能仅凭指令就断言真实刀尖已精确到位。

> 程序描述加工目标；关节是执行这个目标的机构参数。

07　先认识 M、W、J，再引入 K 和 T

符号	在本讲义中的含义	在哪里看
M	固连机床底座的物理笛卡尔参考系。	观察真实空间位置与方向。
W	设计给零件编程用的工件坐标系。	图纸、CAD/CAM、G54编程。
J / q	各机械关节的位置集合。	J0…J4、编码器/关节显示。
K	已经解释偏置、尚未经过RTCP逆解的坐标输入。	本项目解释器输出到运动学的接口。
T	刀具参考系或刀具受控点的定义。	主轴参考点、刀长标定、TCP。

初学者先记住M、W、J三个概念。K是在复杂机床中避免混淆的“中间层”：控制器界面叫作Machine的XYZ，有时是运动学坐标，不能不看定义就认定它一定等于物理直线关节。

本讲义用K做教学记号，不是要求控制器界面必须新增一个叫K的按钮。它也不是G代码中的圆弧中心地址K；同一个字母处在不同上下文中，意义不同。

> **读图颜色**
>
> 蓝色：机床/基础参考；绿色：工件参考；红色：目标点或刀尖。颜色帮助区分对象，不代表电气接线。

> 本讲义的符号是为讲清层次而定义的，不是所有控制器统一的界面名称。

08 回零解决什么？工件对刀又解决什么？

回零建立机器执行机构的参考，使控制器知道关节位置相对于设定基准如何计量。它解决“机床自己在哪里”的问题。工件可能每次装在不同位置，回零并不会自动知道零件角点或孔中心在哪里。

工件对刀、测头找正和设置工件偏置，解决“这次装夹的零件参考在哪里”的问题。这两类基准相互有关，但不能互相替代。

操作	主要作用	不会自动完成
回零/HOME	建立关节或机床参考。	不会自动测得G54零点。
选G54	选择已保存的编程参考。	不会自动重新测量工件。
设置工件零点	更新零件与基础参考的配准。	不会重新标定转台中心。
测刀/刀长标定	建立主轴参考与刀尖的关系。	不会确定所有工件方向。

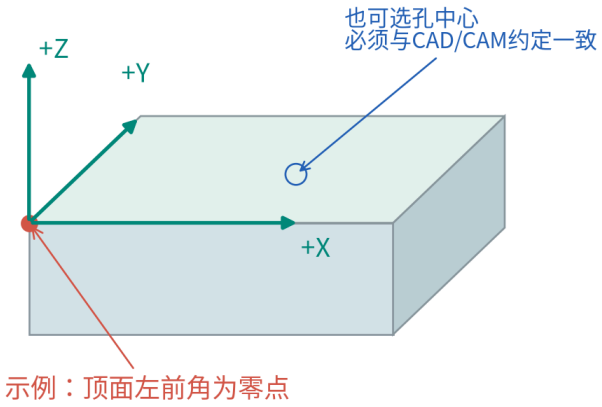
> **G28不等于点击HOME**
>
> 在LinuxCNC中，G28涉及预定义返回位置；它不是本项目回零按钮的同义词。初学阶段不要仅凭“回原点”的中文说法混用两种动作。

> 回零、找工件零点、测刀，是不同的准备步骤。

[S2][S4]；实际回零方法取决于机床配置。

09　工件坐标 W 代表什么?

工件坐标是一套为零件几何与加工意图服务的参考。设计人员可能把顶面角点定义为(0,0,0)，把孔中心写成(30,20,−5)。CAM和操作者必须采用兼容的约定，程序中的点才会落在预期部位。



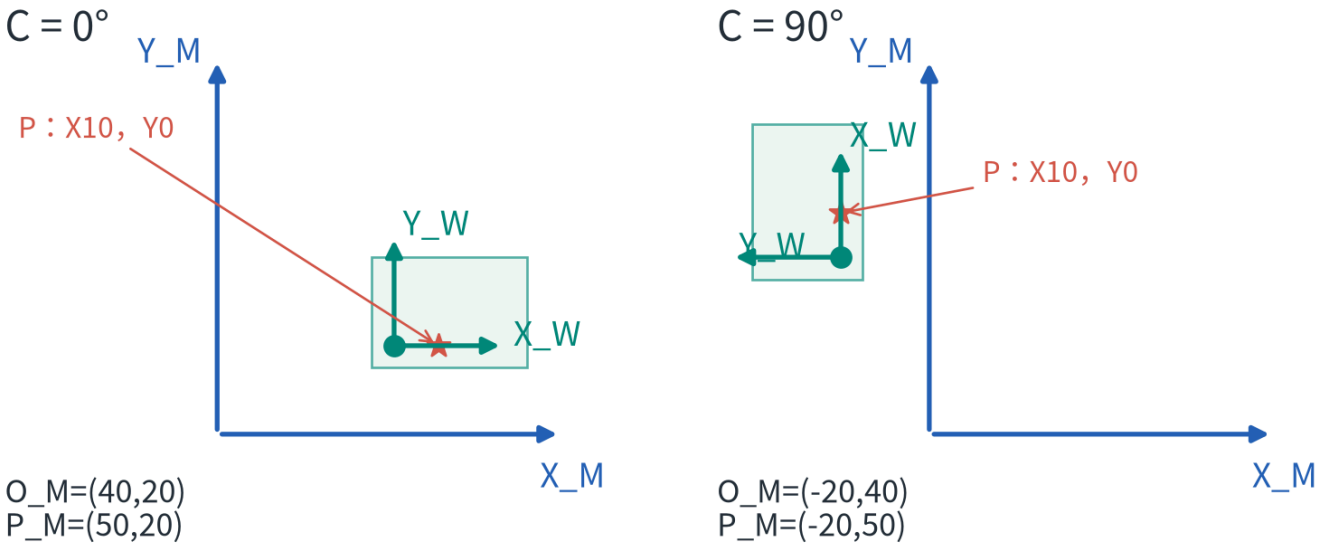
原点可以选角点、中心、基准面交点等；不一定是转台中心。

零件本身并没有天然写着“这里必须是X0”。原点是根据图纸、工艺、装夹和测量便利性选定的。选好以后，需测量或标定出它与机床结构的关系。

> **原点自己的坐标永远为零**
>
> 工件原点在工件坐标中一直是(0,0,0)。但它在机床空间的位置可能随转台转动。两个说法并不矛盾，因为参考系不同。

> 原点是工艺约定，控制器必须得到该约定与实际装夹之间的对应关系。

10　工件转了，原点与方向一起转



零位工件原点o=(40,20,10)mm；只看XY俯视，Z另计。

设工件原点在零位参考中是(40,20,10)。当C=0°时，它在机床空间仍在(40,20,10)。当C=90°、旋转中心为零时，它移到(−20,40,10)。

此时工件+X方向也转成机床+Y方向。工件上的P=(10,0,5)始终是“从工件原点沿工件X走10，再沿工件Z走5”，但它在机床空间从(50,20,15)变到(−20,50,15)。

> **两个都要算**
>
> 仅改变坐标方向、却让原点一直留在原地，一般不正确。只有原点恰好位于所转轴线上等特殊情形，原点才可能不移动。

随工件固连的坐标架是一个整体：原点和三根方向箭头一起变换。

11 “物理工件坐标架”和“G54表”不是同一对象

物理上，你可以想象把一套坐标箭头画在工件上。转台转动时，这套箭头会随工件一起转。软件里，G54则是用于选取一套工件坐标设置的指令，其数据保存在控制器的偏置表中。

在本课RTCP模型中，G54主要给出零位配准偏置；姿态改变时的空间变换由另一层运动学完成。因此，偏置表中的数字可以保持不变，但它所对应的物理工件原点会在机床空间移动。

层次	本课中的作用
零件上的W坐标架	描述固定在零件上的位置和方向。
G54等偏置数据	告诉解释器如何把编程坐标换算到基础输入K。
RTCP运动学	利用当前A/B、C及几何参数把K换算成物理关节目标。

如果只打开G54而没有正确的五轴运动学，控制器并不会因此自动追踪物理工件架的旋转。非RTCP时，软件的直接坐标解释与转动后的零件局部方向可能不一致。

> **不要过度推广**
>
> 不同控制器可能把工件旋转、倾斜工作平面、TCP功能组合成不同指令。本课展示一条明确的数据链，不能把普通三轴G54直接称为完整五轴随动变换。

> G54负责选参考数据，RTCP负责根据结构和姿态换算运动。

[S1][S2][S4]；后半为本课模型与项目实测数据链。

12　先约定机床零姿态，再谈工件装在哪里

要描述工件装夹位置，必须先有一套明确的基础参考：旋转轴的零角度是什么姿态？倾转轴中心在哪里？C轴中心在哪里？直线轴与刀具基准怎样关联？这些由机床配置与几何标定决定。

把转台放到定义的A/B=0、C=0姿态，可以建立零位配准参考K。零件的工件原点相对于这套参考的偏置，记作o。若工件方向也存在初始夹角，还需要初始方向变换R_0。

编程点 p_W → 零位配准 → p_K → 当前姿态变换 → 关节 q

要确定的量	谁来提供
轴的零角度、正方向	回零/编码器基准与机床配置。
旋转中心、轴间偏置	机床结构数据、测量与标定。
零件原点及初始方向	工艺定义、装夹、测头或找正。
刀具长度与参考点	刀具测量、刀表与补偿设置。

这些数据不是点击“开启RTCP”时凭空生成的。RTCP只会使用已有数据计算，不能自动猜出实际工件的全部装夹关系。

> RTCP能正确跟随的前提，是机床几何与工件配准都已经正确。

13　工件原点放在哪里比较合适?

常见选择有：毛坯顶面角点、工件中心、定位孔中心、某个基准面与两条基准边的交点。没有一种位置对所有零件都最优。应与图纸和CAM编程基准一致，并且能够稳定测量。

1. 先看图纸和CAM：尺寸从哪里标起，程序预期哪个位置为零?
2. 再看装夹：该基准在机床上是否可接近、可找正、可重复?
3. 确认方向：+X沿哪条边，+Y朝哪边，+Z朝工件外侧还是另一约定方向?
4. 记录采用的是毛坯面、成品基准面还是夹具基准，避免同名不同面。

> **例子：顶面角点**
>
> 若顶面定义为Z0，向材料内部加工的深度通常写成负Z；但这取决于你建立的+Z方向。不能看到“钻深5”就不加判断地写Z+5。

换一次装夹，零点位置可能变化。程序里零件局部几何可以保持不变，只更新该次装夹的配准数据。这正是工件坐标的价值。

转台中心可以作为机床标定基准，却不必与零件编程原点重合。把两者强行设成同一点，会让工艺选择受到不必要限制。

> 方便编程、方便测量、装夹可重复，比“零点看起来居中”更重要。

14　怎样把选好的原点告诉控制器?

首先必须让刀尖或测头与已知基准建立真实关系。可以使用测头、寻边器、对刀棒或其他已标定的测量方法。测量工具本身的半径、长度及接触位置要考虑进去；刀具中心碰到边，不等于刀尖已经位于所选坐标原点。

一种便于理解的零位流程

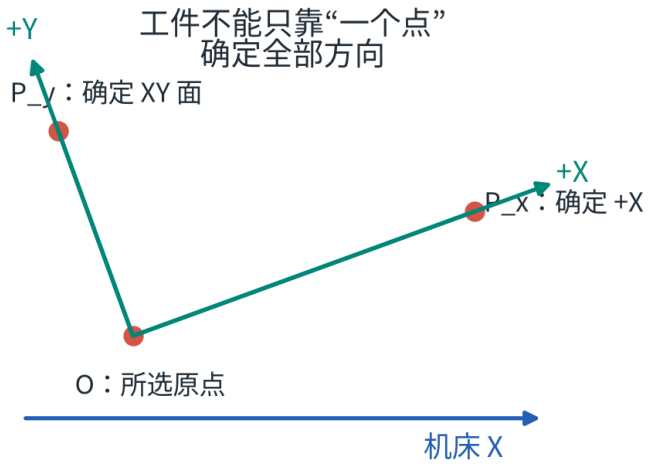
1. 完成机床回零和刀具/测头标定。
2. 把转台置于已知参考角度，按工艺要求找正工件。
3. 测得原点相对于基础参考的位置，写入目标工件坐标表。
4. 选择该坐标系，用已知的另一个点检查读数和方向。
5. 用小幅、空载的受控运动验证，再进入实际加工。

> **在倾斜姿态下对刀更要谨慎**
>
> 当前关节XYZ、RTCP输入K和工件局部坐标W可能不同。不能直接把屏幕上的J0/J1/J2当作G54偏置写入。应使用控制器提供的、与当前运动学兼容的坐标设置/测量流程。

本讲义的数字测量例仅解释偏置计算，不构成真实机床的探测轨迹。设备现场必须遵循该机床和控制系统的操作流程。

> 找零点不是把三个显示框都随意清零，而是建立可验证的几何关系。

15　一个原点不能确定三维方向



平面内示意：O给出位置，O→P_x确定+X，再用不共线点确定平面及+Y。

你把一个孔中心设为零，只确定了原点位置。围绕这个点，坐标架仍可以转动。要确定方向，还需要已知的边、面或更多测量点。

常见做法是让工件基准边与预定方向对齐，并用基准面确定法向。若需要软件方向补偿，应由控制器支持的工件坐标旋转或倾斜平面功能来表达，不能只修改X/Y/Z三个平移偏置。

> **完整三维配准**
>
> 三个不共线的点可帮助确定一个平面及平面内方向，但仍须明确点的对应关系、顺序、正负号和右手规则。共线点无法唯一确定平面。

> 原点回答“从哪里量”，方向回答“朝哪里量”；它们需要不同信息。

16 装夹方向和转台姿态，要分开处理

假设工件在C=0时，长边已经相对基础X方向偏转了20°。这20°是装夹的初始方向关系；后来C又旋转30°，属于机床当前姿态。两者来源不同，但都影响最终空间方向。

量	何时确定	通常何时变化
初始工件方向R_0	装夹/找正/CAM配准时。	重新装夹或修改坐标设置时。
转台角度A/B、C	回零基准与每条运动指令。	程序或Jog执行过程中。
几何中心P、偏置D	机床结构测量和标定时。	结构调整或重新标定时。

LinuxCNC可用特定工件坐标设置表达XY平面内旋转，例如G10 L2的R参数；它不是一个任意三维倾斜平面描述。摆头/转台结构、倾斜平面和TCP功能的组合仍要按具体控制器核对。

> **本项目的数值实验范围**
>
> 下面逐行实算只使用平移G54，即R_0为单位变换。讲义会解释一般方向关系，但不宣称当前Web示例已完整实现任意三维坐标旋转或所有厂商TCP指令。

> 没有声明初始装夹方向时，不能擅自认为工件边一定平行于机床导轨。

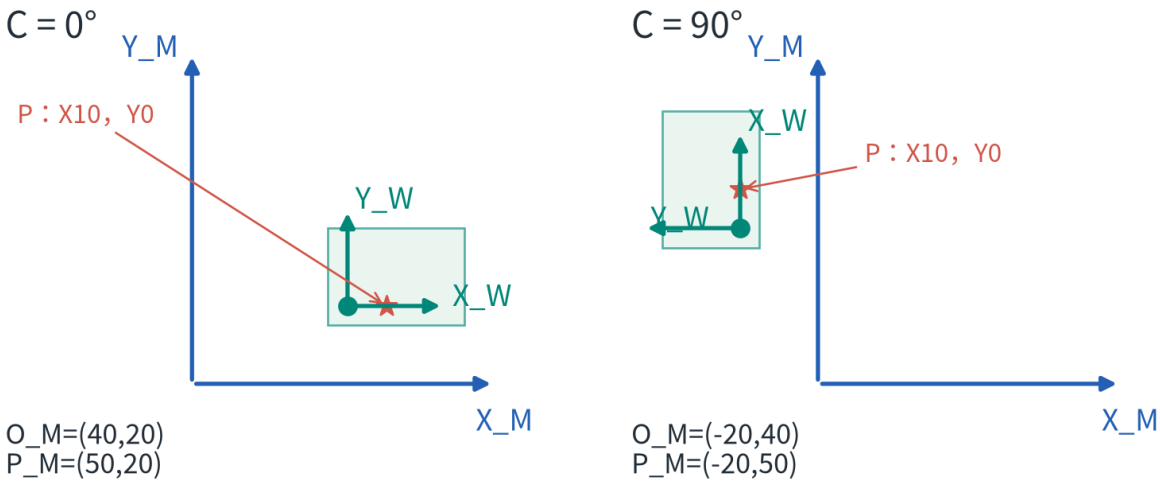
[S4] G10 L2；项目实验边界见后文。

17　一个点怎样从工件坐标换算到机床空间?

先不看公式。把工件原点在机床空间的位置标出来；再把工件+X、+Y、+Z三个方向换算到机床空间。最后，沿这些方向分别走程序给定的距离。

> 空间目标 = 当前工件原点 + X · 工件X方向 + Y · 工件Y方向 + Z · 工件Z方向

用数学记号写为 r_M = O_M + E_MW · p_W。O_M是工件原点在机床中的位置，E_MW的三列是工件三个单位方向在机床中的分量。矩阵只是把“三个方向加权相加”缩写起来。



转台改变姿态时，O_M和E_MW一般都变化；工件点p_W可以保持。

方向与位置的不同

增量只用方向关系：Δr_M = E_MW · Δp_W，前提是这一增量过程中姿态保持不变。若角度也在变化，原点和方向同时变，必须计算完整运动过程。

固定姿态时可以只变换增量；姿态在变时必须处理完整的点变换。

18　G54究竟做了什么?

G54选择第一套工件坐标设置。LinuxCNC中还有G55、G56、G57、G58、G59、G59.1、G59.2、G59.3，合计九套。你可以给不同工件、不同装夹或不同夹具保存不同偏置。

选择指令	G10中的P编号	本页教学含义
G54	P1	第一套坐标设置
G55	P2	第二套坐标设置
G56…G59	P3…P6	第三至第六套
G59.1…G59.3	P7…P9	第七至第九套

G54具有模态性：选中后，后续适用的运动指令继续使用它，直到选了同组的另一个坐标系。单独执行G54没有轴目标时，不会自动让刀具移动到工件零点。

> **注意同一行可能同时有运动**
>
> “G54”只选择坐标系；但“G54 G1 X10”还包含运动目标，后者会按该坐标系及当前模式解释X10。不能只看到行里有G54就认定不会运动。

> 选择参考，不等于走到参考原点。

[S2][S4] 坐标选择；实测G54/G55见gcode-evidence.json。

19　G54的40、20、10代表什么?

本课统一例子：在零角度、零几何偏置且方向对齐的教学模型里，设工件原点相对于K参考为o=(40,20,10)mm。意思是先沿K的X方向40、Y方向20、Z方向10，才到工件原点。

```
G21 G90
G10 L2 P1 X40 Y20 Z10
G54
```

当程序要求工件点X10 Y0 Z5时，解释器得到K输入(50,20,15)。到这一步还没有完成五轴运动学计算；随后还要依据B/C或A/C及几何参数得到关节目标。



同一例子继续取C=90°，RTCP最终关节XYZ为(−20,50,15)。

因此，G54偏置中的40不能理解成“无论转台怎么转，工件原点都固定在机床X=40”。它是零位配准数据，动态空间位置由RTCP继续换算。

> 工件偏置先进入编程坐标解释，转台姿态随后进入运动学。

20　G10 L2：把已知偏置写入坐标表

```
G21
G10 L2 P1 X40 Y20 Z10
```

地址字	本行含义
G10	设置某类数据，具体对象由L等参数决定。
L2	本例为直接设置工件坐标偏置。
P1	第一套工件坐标设置，即G54。
X40 Y20 Z10	在本课约定下，直接写入三项长度偏置。
G21	当前长度单位是mm；不要把单位上下文省略。

G10 L2适合“已经知道偏置是多少”的情况。它不是“把当前位置显示成某个值”的自动计算命令。若误把L20的用法套在L2上，偏置会被设置到完全不同的值。

G10里的X不一定是运动目标

同一个地址X，在运动段里可以表示目标；在G10设置段里表示配置参数。一个数到底有什么作用，必须结合该行的G代码和L/P上下文。

这条指令设置参考数据，不是命令XYZ运动到(40,20,10)。

[S4]；同程序实际执行未产生运动段。

21　G10 L20：让当前位置具有指定工件坐标

假设刀尖已处在本课K位置(70,30,20)，你确认它就是工件原点。想让此时工件读数变为(0,0,0)，控制器需要把G54偏置设置为(70,30,20)。

```
G10 L20 P1 X0 Y0 Z0
```

偏置 o = 当前位置 p_K − 希望此刻显示的工件坐标 p_W

若当前位置仍为(70,30,20)，但希望此点显示为工件X10 Y0 Z0，则偏置变成(60,30,20)。改变的是坐标设置，不是机床执行轴的位置。

动作	K中的X	G54工件X
刀尖保持当前位置	70	取决于原偏置
L20 P1 X0	70	0
之后G54 G90 G1 X10	80	10

倾斜时不能随意代入关节

公式中的当前位置必须是该控制器用于工件坐标设置的参考位置。本课RTCP下它是经正解/同步的K，通常不是J0/J1/J2。实际使用前应按控制器的TCP兼容测量流程确认。

L2写“偏置是多少”；L20给“当前位置应读成多少”，由控制器算偏置。

[S4]；零旋转实测L20偏置与后续X10运动。

22 换G54/G55为什么可能只改数字、不动机床?

给G54设置o_1=(40,20,10)，给G55设置o_2=(80,−10,10)。刀尖已在K=(50,20,15)。选择不同坐标系时，同一位置会产生不同局部读数。

状态	关节/刀尖是否立即移动	读数或下一步目标
当前G54	否	W_1=(10,0,5)
单独选择G55	否	W_2=(−30,30,5)
再执行G90 G1 X10	是	K的X变为90；未写的Y/Z保持原位置

同一个X10在G54和G55中指向不同目的地。若程序意图是加工两个不同装夹位置的相同零件，这正是需要的行为；若是无意中选错坐标系，就会整段偏移。

> **工件坐标切换与RTCP切换不是一回事**
>
> G54→G55更换配准数据；M429→M428更换运动学解释。两类切换本身都不应作为“无目标的移动指令”，但它们改变后续运动字值的含义。

下方数值使用方向对齐的参考K来展示偏置。转台已倾斜时，物理空间关系还要经过RTCP变换。

换尺子可以只改变读数；按新尺子再给目标，才可能让机床运动。

[S2][S4]；G54/G55选择及省略轴实测。

23 G52、G92、G53：先学会不混用

指令/功能	通用理解	初学者要点
G52	在已有工件参考下使用局部偏置。	不是自动测得新的装夹原点。
G92	通过坐标重新定义产生附加偏置。	可能与持久化设置有关，不能假设程序结束就完全清掉。
G53	本段使用控制器定义的机床坐标方式。	非模态；不等同于HOME，也不保证直接控制电机关节。

多层偏置叠加后，程序点到基础参考的换算不再只有一个G54。排错时必须查看当前有效的所有层，不能只盯着G54表里的三个数字。

尤其在非平凡运动学机床上，控制器的“机床坐标”仍可能是运动学输入空间。G53不意味着跳过所有运动学，G53 Z0也不能被当作通用安全退刀动作。

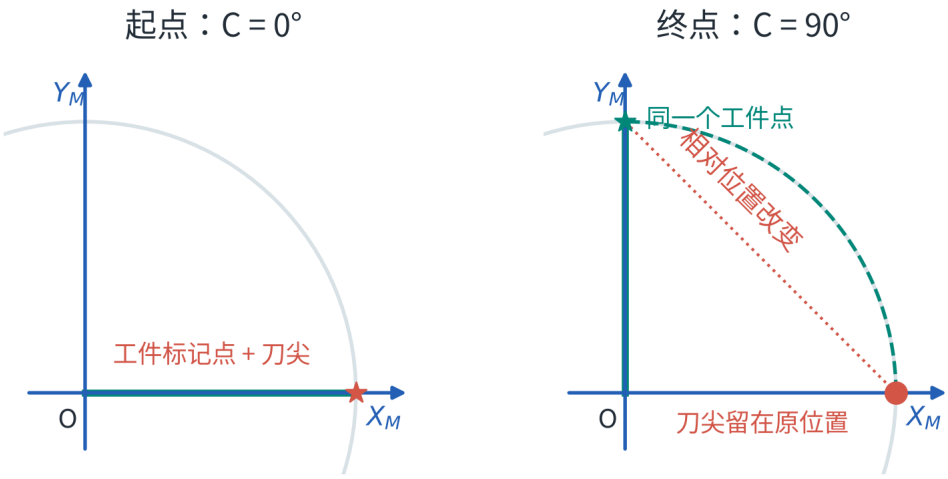
> **本讲义的实验边界**
>
> 主程序刻意不用G52/G92，也不依靠G53绕过RTCP。这一页讲官方语义与排错意识；当前Web切片没有在本课中完成这些功能的全行为验收。

只在理解所有有效偏置后再组合使用；初学实验先保持参考层次简单。

[S2][S4]；对G53关节语义需按具体运动学核对。

24 非RTCP：“独立”的准确含义

对于本课直接映射模式，经过单位/偏置解释后，X/Y/Z对应直线关节，A/C或B/C对应旋转关节。只命令C转动，不会自动追加保持刀尖相对工件点的XYZ补偿。



只转C：工件标记点被带走，刀尖留在机床原位置。

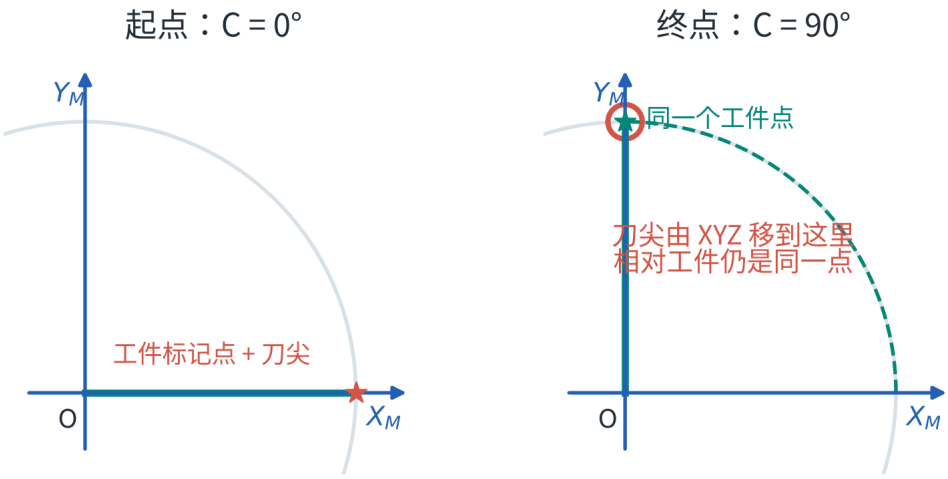
这不表示各轴不能同时运动。例如一行同时给X、Y、B、C目标，控制器仍可协调插补。也不表示零件与刀具之间“完全没有关联”：物理工件转了，二者的相对位置当然会变。

> **没有RTCP也可以五轴加工**
>
> CAM/后置可以事先针对指定机床结构计算各轴轨迹，交给控制器执行。但换机床几何、刀长或装夹后可能需要重新计算；它与控制器实时进行RTCP不是同一机制。

独立是“没有自动的姿态几何补偿”，不是“没有同步插补或相对运动”。

25 RTCP控制的是相对目标关系

RTCP可理解为旋转姿态变化时的刀尖位置补偿。给定目标工件点以及A/B、C角度，控制器结合机床几何计算所需关节位置，让刀尖实现规定的相对工件轨迹。



保持同一工件点，角度改变时XYZ按需跟随。图为零偏置简化模型。

“刀尖保持”有两种不同说法：保持在机床空间同一点；保持相对工件的同一点。本课RTCP讲的是后者。工件转动时，该点在机床空间可能沿弧线移动，所以刀尖也需要移动。

正确的观察方式

在工件上选一个离旋转轴有距离的标记点，先让刀尖对准它，再保持编程XYZ、改变角度。应看刀尖与标记点是否继续对齐，而不只是看屏幕XYZ有没有变。

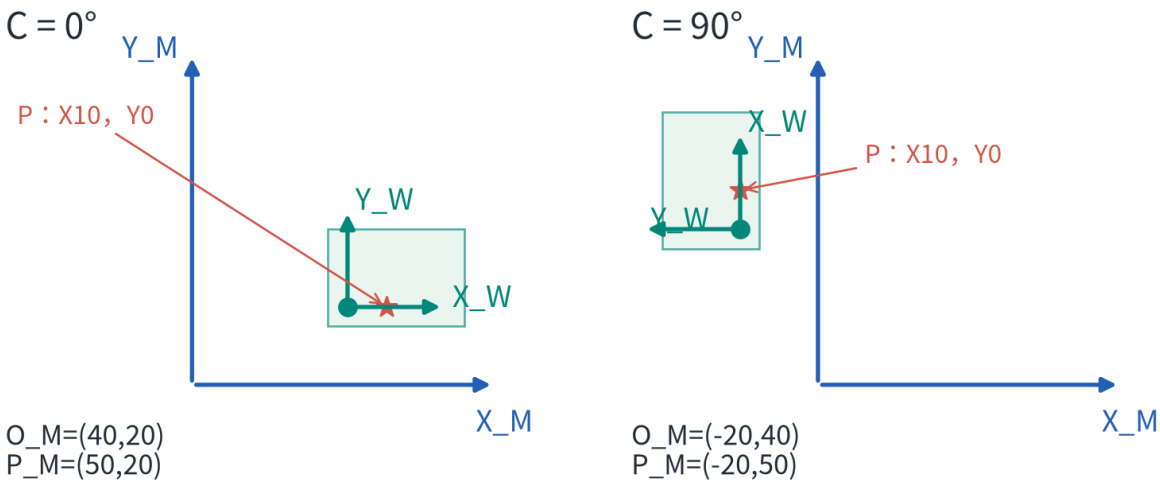
RTCP不要求刀尖在世界里绝对不动，而要求它满足指定的相对工件关系。

[S1] 五轴运动学；本项目原生AC/BC正逆解。

26　用G54例子看“原点随动”

现在使用贯穿后续章节的例子：BC结构，B=0；几何中心P=0、偏置D=0、刀长差L=0；G54零位偏置o=(40,20,10)。程序目标始终为W=(10,0,5)。

C角度	工件原点的机床位置O_M	刀尖目标/关节XYZ	程序W
0°	(40,20,10)	(50,20,15)	(10,0,5)
90°	(−20,40,10)	(−20,50,15)	(10,0,5)



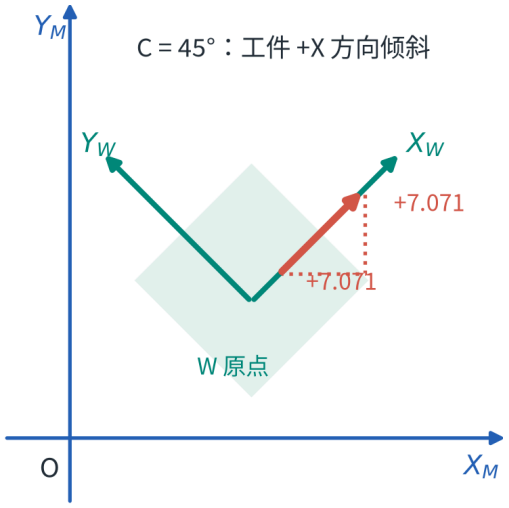
两种姿态下，刀尖相对工件原点的局部分量始终相同。

注意：G54表仍是40、20、10，没有每次改写成−20、40、10。后一个值是由当前姿态算出来的空间位置，不是要求操作者实时改G54。

> 固定的是工件内的定义与零位配准数据，变化的是它们在机床空间的表达。

27　沿工件X前进，哪些电机会动?

在C=45°、倾转角为0°时，工件+X方向斜着指向机床+X和+Y。沿工件X增加10mm，必须由两根固定导轨合成：机床X约增加7.071mm，机床Y约增加7.071mm。



绿色方向随工件，蓝色机床导轨方向不转。

$$\text{姿态固定：}\Delta q_X = 10\cos 45°;\ \Delta q_Y = 10\sin 45°$$

XYZ导轨并没有在机械上旋转。变的是工件方向在机床中的表达。编程沿一根工件轴走，实际可以由两根或三根直线关节一起完成。

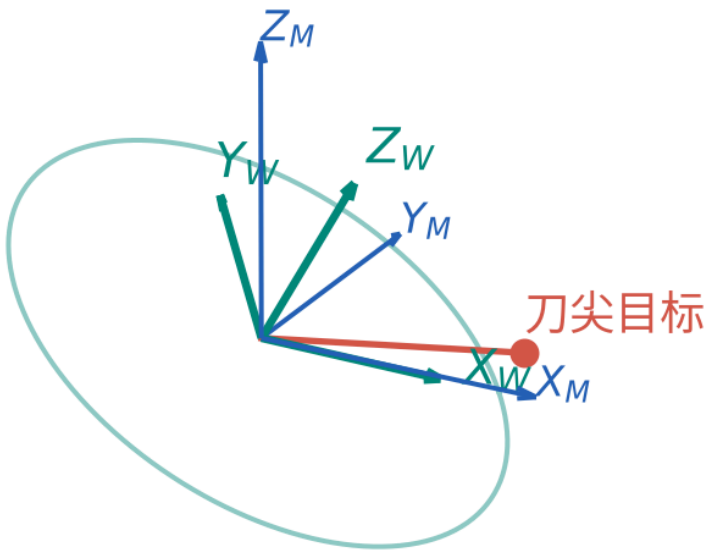
原点为何没有出现在增量式里?

因为这一页姿态固定，比较的是两个目标点的差，平移项相减后消失。若B/C也在同一过程中变化，就不能只用这条固定方向增量式。

一个X字，不一定只驱动X电机；先问它属于哪套参考系。

28　加入B或A后，方向可以进入三个维度

BC结构中，B倾转带着C轴一起倾斜。采用本项目约定，零偏置变换为R=R_Y(B)R_Z(C)。当B=30°、C=45°时，工件+X的机床方向分量约为(0.6124,0.7071,−0.3536)。



保持B=30°、C=45°时，沿工件X前进10mm，会同时改变三个直线轴。

$$\Delta q \approx (6.124,\ 7.071,\ -3.536)\ \text{mm}$$

AC则使用R_X(A)R_Z(C)。同样写30°、45°，AC与BC的刀尖空间位置一般不同。矩阵从右向左作用，表达父子结构的坐标转换顺序，并不是要求先完成C动作再单独执行A/B。

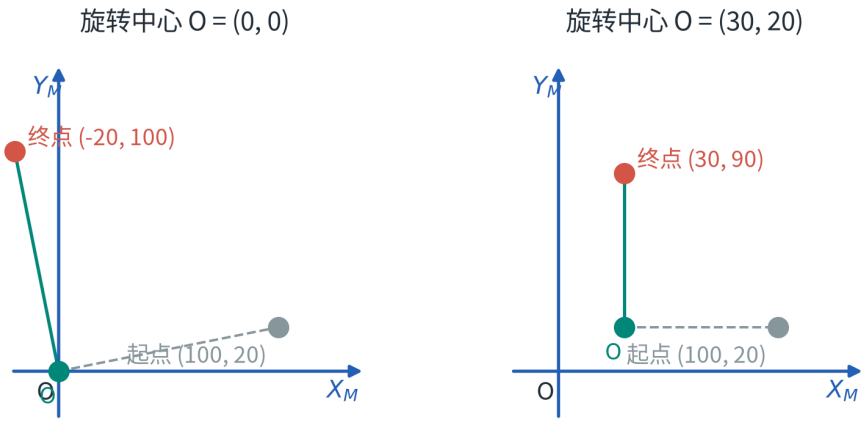
这里的正方向采用本项目原生计算约定；实际机械正转、工件视角和摄像机视角可能使视觉直觉相反，必须用正确标定核对。

> RTCP用的是完整机构链，不能只把A字母改成B就期待相同空间姿态。

29 G54、旋转中心、轴间偏置各管什么?

参数	属于什么关系	不能拿来替代
G54偏置o	零件编程原点与基础参考K的配准。	不是转台的物理轴心标定。
rot-point P	本项目原生TRT的旋转参考位置。	不是每次装夹都应随便改的零件零点。
axis offset D	倾转轴与C轴之间的结构偏置。	不是普通工件坐标平移。
tool-offset L	刀具参考与受控点的标定量。	不是看到网格长度后直接填的任意长度。

这些量都可能以mm显示，但含义和进入公式的位置不同。如果刀尖偏差源于转台中心错误，单靠修改G54可能只在某个角度碰巧对齐，换一个姿态就再次偏离。



起点相同、角度相同，绕不同中心会产生不同目标位置。

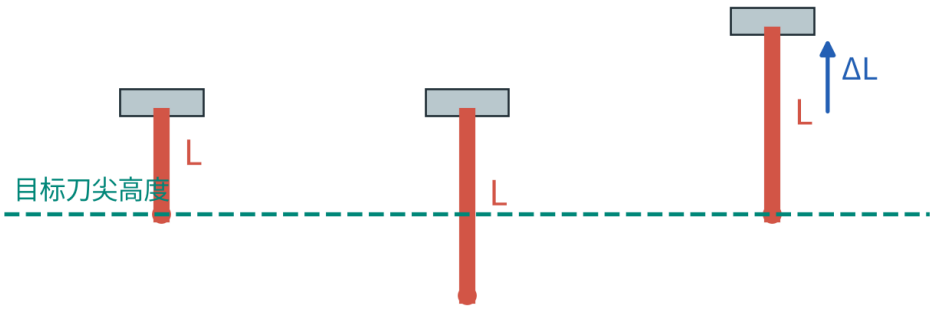
标定与装夹分开

机床结构参数应通过结构测量和标定确定；工件偏置用于表达该次装夹。教学可以故意改参数做对照，但真实设备不能把结构标定当成随手找零工具。

看到“偏移多少毫米”时，还必须问：是哪个参考系之间的偏移?

30 为什么必须知道刀尖究竟在哪里?

电机驱动的通常是滑台和主轴箱，程序关注的却是刀尖。主轴参考点与刀尖之间相隔一段刀具长度。如果更换刀具后长度不同，保持同一主轴位置并不能保持同一刀尖位置。



短刀　　长刀，主轴未补偿　　长刀，主轴上移

相同刀尖目标：刀更长时，主轴参考位置需要按正确约定调整。

在LinuxCNC的完整刀补语义中，G43 H3表示使用指定刀具长度偏置条目；H3不是“刀长等于3mm”。T表示工具选择，M6表示换刀流程，S表示主轴速度，也不能与刀长混用。

> **本项目边界特别说明**
>
> 主数值实验统一L=0。当前Web解释器切片的H/G43处理不能当作完整LinuxCNC刀表语义的验收依据，因此讲义不把“程序成功解析”冒充“完整刀补行为正确”。真实应用须使用经过验证的刀表和测刀链路。

RTCP、刀长补偿、倾斜工作平面是相互配合的功能，不是一个开关的三个名字。

> 先标定刀具参考点，再讨论刀尖目标；网格看起来像刀具，并不代表它已正确标定。

[S4] G43；本项目sim_interpreter.c的能力范围与本课测试边界。

31 一行G代码由“地址字”组成

```
N50 G1 X10 Y0 Z5 B0 C0 F600
```

一个地址字通常由字母和数值组成，例如X10。字母说明这个数属于什么类别；当前G/M代码和模态状态进一步决定它的具体作用。空格用于阅读，不是每个空格后就执行一次运动。

字	本行解释
N50	程序段编号，便于识别，不是运动距离。
G1	按进给方式生成直线类型的运动段。
X10 Y0 Z5	在当前绝对/增量和坐标设置下给出XYZ。
B0 C0	给出旋转轴角度目标；本课为BC结构。
F600	设置进给值，含义受进给模式和单位影响。

控制器应先读完整个程序段，检查语法与模态组，再求出整体目标。不能把它理解成先走X10、走完再走Y0、再走Z5、再转B和C。

本课旋转轴的工件偏置为零，所以B/C字直接成为相应关节角度目标；若启用了角度偏置，也要先换算，不能只看字面角度。

先问三个问题

当前在用哪套坐标？G90还是G91？RTCP是否已开启？这三项不清楚，仅看X10这个字不能确定电机终点。

地址字是数据；程序段和当前模式共同决定数据的含义。

32　没有写出来的内容，可能沿用上一行

模态可以理解为控制器记住的当前设置。G1、G90/G91、G20/G21、G54等在不同模态组中生效。后续行省略这些字时，常会沿用之前的设置。

```
G21 G90 G54 G1 X10 Y0 Z5 F600
C90
G91 X10
G90 X20
```

第二行只有C90，但它仍处于G1、G90、G21、G54环境，F仍为600。第三行把距离模式切换成G91，X10变成增量。第四行恢复G90，X20又变成绝对目标。

省略什么	通常怎样理解
省略G1	沿用有效运动模式；不是自动停止解析。
省略F	在允许沿用的进给模式下保留原值；G93另有要求。
省略XYZ某一轴	通常保持该坐标分量的当前目标，不自动归零。
省略角度字	保持当前角度目标，不自动回零角度。

同组冲突

不要在同一段同时写互相矛盾的同组模式，例如G90和G91。合法的执行顺序不等于可以把冲突指令堆到一行。

读一行程序时，要带着“当前状态表”一起读。

[S4][S5] 模态与程序段；本文只涵盖常用运动模式。

33　G20/G21改变长度单位，不把角度变成长度

数值类别	G21示例	G20示例
X/Y/Z长度	X1为1mm	X1为1inch=25.4mm
A/B/C角度	本课以度计	仍按控制器旋转角约定，不把1°换成25.4°
G94直线进给F	F600通常为600mm/min	F2通常为2inch/min=50.8mm/min
G93的F	逆时间，见进给专页	不是长度值，不能机械乘25.4

本课WASM已实测：G20 G91 X1产生25.4mm的直线增量，G94下F2换算为50.8mm/min。然后换回G21时，后续长度又按mm解释。

变量、刀表、测量结果以及URDF还可能使用各自定义的单位。教学数据进入某一接口前应先统一，不要因为屏幕只显示一个数字就忽略单位。

> **一个常见大错误**
>
> 把毫米程序放在英寸模式下运行，会让每个长度字放大25.4倍。反过来也不正确。程序头明确声明单位，通常比依赖上次运行残留状态更可靠。

> 长度、角度、转速、时间、索引号是不同量纲；不能用一个换算规则处理所有数字。

34　G90 X10：去当前编程系的X=10

在G90中，X10通常表示当前有效编程参考系中的绝对目标X值为10。它不是“再走10”，也不是“无论有什么偏置，电机都去10”。

本课条件：G54的o=(40,20,10)，方向对齐，B=0、C=90°，RTCP开启，当前目标W=(20,0,5)。执行G90 G1 X10，新的W目标变为(10,0,5)。

计算层	新目标
工件坐标W	(10,0,5)
加G54得到K	(50,20,15)
C=90°后关节XYZ	(−20,50,15)

原来W=(20,0,5)对应关节(−20,60,15)，所以这一动作让机床Y减少10mm。虽然程序只写X10，实际发生变化的是本例的Y关节。

> **不要把G90看成“机床坐标开关”**
>
> G90只是绝对距离模式。它仍与G54等坐标设置共同起作用，并没有替代工件坐标系选择。

> “绝对”必须说明是相对于哪套当前有效参考，而不是自动等于机床电机绝对位置。

35　G91 X10：在当前编程方向增加10

仍用G54 o=(40,20,10)、B=0、C=90°、RTCP开启。当前位置W=(10,0,5)。执行G91 G1 X10后，W变为(20,0,5)，K从(50,20,15)变为(60,20,15)。

$$\text{新目标W} = \text{旧目标W} + (10,0,0)$$

角度保持C=90°，因此该增量在机床空间沿+Y方向，关节从(−20,50,15)变为(−20,60,15)。偏置不应被重复加两次。

```
G91 G1 X10
G90
```

如果同一段还有B10或C20，它们在G91下通常也表示角度增量。控制器先形成新的整体坐标/角度目标，再插补并求解；不能把变化的姿态当成整段保持不变，只旋转一次XYZ增量就结束。

省略轴的正确理解

G91只写X10时，未写的轴不增加数值；在RTCP中，为实现这个编程增量，多个物理关节仍可能同时变化。记得之后是否需要恢复G90。

旋转轴还可能有展开角度或回绕设置。C=0与360的空间方向可以相同，但关节位置、行程和运动路径未必相同；本课不假设控制器自动选择最短转动路径。

G91改变目标的形成方式，不取消工件方向和RTCP计算。

36 G0/G1分别如何产生运动?

G0用于快速定位类运动，G1用于按进给规则执行直线类型运动。给出的通常是该段目标，而不是每个伺服时刻的位置。控制器的规划器决定加减速、插补进度及与相邻段的衔接。

层次	G1 XYZBC在本课中的处理
编程空间	定义XYZ与角度的起点/终点以及运动类型。
插补	逐步得到中间位置和中间角度。
运动学	每个采样按该时刻姿态和结构求解关节。
执行	输出连续的关节位置/速度，并检查限制。

G1说的“直线”属于控制器定义的编程空间运动，不意味着所有机械部件在世界中都走一条直线。保持同一工件点、转动C时，刀尖可以在机床空间走弧线。

> **不要只检查终点**
>
> 如果在两个端点做了正确逆解，却在关节之间用不相符的方式连线，中途可能偏离工件轨迹。正确RTCP必须在整个运动过程中维持目标关系。

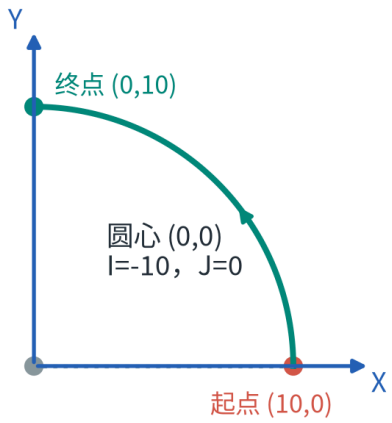
G0的具体协调方式、G64圆滑和G61精确路径等规则因控制系统而定。初学者不应把快速定位当作任何场景下的安全路径。

程序给段目标，规划器给连续过程，运动学给各轴实现方式。

37　G2/G3里的I、J、K不是三根新电机轴

G17通常选XY圆弧平面；G18、G19选择其他主要平面。G2/G3给出圆弧方向，圆心可以由I/J/K等地址表达。圆心方式有独立的绝对/增量规则，不能仅看G90/G91就猜圆心如何解释。

```
(教学条件：G17，圆心增量方式)
(起点 X10 Y0，圆心 X0 Y0)
G3 X0 Y10 I-10 J0 F300
```



圆心增量从圆弧起点量：I=-10、J=0；不是从工件原点量两个绝对坐标。

本例圆心增量等于“圆心−起点”。R有时也可表示圆弧半径，但它在其他指令中又可能表示不同参数。姿态固定时，工件平面中的圆弧映射到机床中的相应倾斜平面。

本课不混合演示

圆心模式、整圆/多圈、R长短弧以及圆弧期间同时转台运动，需要按控制器文档逐项确认。不要把一段已知平面圆弧随意扩展成未验证的复杂五轴程序。

同一个字母的含义由当前指令规定；I/J/K经常是圆心参数，不是位置轴。

38 F到底是速度、时间，还是每转进给?

模式	F的基本含义	例子
G94	每分钟进给；线性、纯旋转或混合段需按控制器规则。	直线段F600mm/min≈10mm/s。
G93	逆时间进给，F的单位为1/min。	F2对应理想段时间1/2分钟=30秒。
G95	每主轴转一圈的进给量。	F0.1mm/rev，配合实际主轴转速。

RTCP下，仅仅保持编程XYZ、改变角度，也可能引发XYZ关节移动。F不等于“每根电机都以这个数值运动”。各关节速度是插补过程和运动学变化共同作用的结果。

LinuxCNC对G94纯旋转段和含直线分量的运动有具体规则；G93可用于表达段时间，但仍受速度、加速度等限制。G93下G1/G2/G3通常每段需要F，不能直接照搬G94省略F的习惯。

> **单位计算只说明请求，不保证实际达成**
>
> F600=10mm/s是单位换算。实际运动可能因加减速、限位和机床能力更慢。控制器不应为了追求F而越过能力约束。

当前Web切片的全部进给模式并非本课程验收对象；主算例采用G94，额外模式依据官方语义讲解。

> 看到F，先查进给模式；判断轴运动，还要查规划器和机器约束。

[S3][S4] G93/G94/G95。

39 不是每个数字都代表位置

字/指令	常见含义	容易误解的地方
S12000	主轴转速请求。	不是刀具线速度或XYZ距离。
T3 / M6	选择工具号 / 执行换刀流程。	不等于自动设定工件原点。
G43 H3	使用指定刀长偏置条目。	H3不是刀长3mm。
D	常见为刀具半径补偿索引。	不是通用“深度”字。
G10 L2 P1	L指定设置类别，P选目标坐标表。	P1不是转台转1°。
G4 P…	停留参数，按控制器定义。	同一P在G10里却是编号。
R / Q	圆弧/循环/设置中的上下文参数。	不能固定解释为半径或角度。
M3/M5/M8等	主轴/冷却等辅助动作。	不是每个M码都有跨品牌同义。

本项目M428/M429是重映射到运动学切换和同步的命令。其他控制器可能采用完全不同的代码组合；把本项目的编号直接复制到真实机床上没有通用性。

> **逐字阅读的方法**
>
> 先识别G/M指令和当前模式，再给每个地址字标注“长度、角度、速度、时间、编号或设置值”，最后再计算目标。

> 字母相同不保证意义相同；上下文比数字本身更重要。

[S4][S5]；M428/M429为本地INI重映射。

40　控制器不是从左到右逐个字移动

1. 读取程序段：识别注释、地址字、数值和参数表达式。
2. 检查语法与模式：单位、距离方式、坐标系、运动类型是否有效且无冲突。
3. 形成整体目标：G90写绝对值，G91加增量；处理省略轴和当前状态。
4. 应用坐标设置：工件偏置、允许的坐标旋转/补偿，得到规范输入。
5. 规划轨迹：速度、加速度、路径衔接和同步屏障。
6. 逐采样计算运动学：根据该时刻位置、角度、中心/偏置/刀具数据求关节。
7. 执行与反馈：伺服驱动/仿真推进；界面再显示实际状态与轨迹。

同一行中的X和B是一个联合目标，不是先走X再转B。控制系统有规定的执行顺序与模态语义；文本排列不能代替这些规则。变量和表达式也先按解释器规则求值，再成为数值输入。

> **“解析成功”不等于“可以安全运动”**
>
> 后面还可能发生缺少回零、越限、刀补不完整、速度约束或几何错误。教学验证至少要看解释结果、规划结果和实际采样。

> 每一层都应有可观察的输出，不能把程序文本直接当成模型关节值。

41　统一实验条件与完整程序

以下每一行都用相同数据核验：BC双转台；旋转中心P=0、轴间偏置D=0、刀长差L=0；工件方向与零位K对齐；G54偏置o=(40,20,10)mm；初始物理关节均为0。

```
N10 G21 G90 G94 G17
N20 G10 L2 P1 X40 Y20 Z10
N30 G54
N40 M428
N50 G1 X10 Y0 Z5 B0 C0 F600
N60 C90
N70 G91 X10
N80 G90 X20 Y0 Z5 B30 C90 F300
N90 X0 Y10
N100 M429
N110 M428
M2
```

N10–N90加M2已通过真实项目解释器、LinuxCNC TP和WASM运动学完整执行；N100/N110另用保持物理关节的切换接口逐段验证。每条运动的关键终点都与解析+逆解结果对照。

下载示例与教学参数不同

本项目下载GLB自带非零中心/偏置。这里为讲清G54与RTCP分层，使用独立零几何测试配置；不是在任意当前机床状态下粘贴代码就应得到同样数字。

本章的数字是固定条件下的可复算例子；改变条件，结果也会改变。

42 N10–N30：设置模式与工件参考

```
N10 G21 G90 G94 G17
N20 G10 L2 P1 X40 Y20 Z10
N30 G54
```

程序段	改变什么	是否生成运动
N10	单位mm、绝对方式、每分钟进给、XY平面。	没有目标轴，不运动。
N20	把G54偏置设为(40,20,10)。	设置数据，不运动。
N30	选择G54为有效坐标系。	单独选择，不运动。

初始物理关节仍是0。G54偏置设定后，原来的空间位置在这套编程坐标中可显示为(−40,−20,−10)。不是刀具突然往负方向跑了，而是这套尺子的原点不在刀具当前位置。

读数变化的第一种原因

修改偏置表就可能改变当前相对读数。验证有没有运动，要检查关节/反馈和刀尖物理位置，不能仅凭屏幕上相对XYZ变化判断。

N10的G17目前只是声明圆弧平面。本程序只有直线类型段，不会因为写了G17就产生一段圆弧。

程序准备阶段可能只改解释状态，还没有开始移动。

43　N40 M428：先切模式，不凭空移动

```
N40 M428
```

本项目中，M428选择RTCP运动学并同步解释器与运动状态。切换时要先读取当前物理关节，再用新模式正解出对应的坐标。它不是“保持旧XYZ数字，再反算出一组新的关节”。

在本章初始姿态下，关节为0、角度为0、几何偏置为0。Identity和RTCP在这个姿态碰巧给出相同的基础位置K=(0,0,0)，因此G54相对读数仍为(−40,−20,−10)。

量	N40之后
关节XYZ/B/C	全部保持0。
基础输入/同步坐标K	(0,0,0)。
G54相对XYZ	(−40,−20,−10)。
运动学模式	从Identity改为本项目TCP。

不能由这个特例推广

角度不为零或中心/偏置不为零时，切换后的坐标数值一般可能改变。但关节与物理刀尖仍应保持。后面N100/N110会看到明显的数字差异。

模式切换准备新的坐标解释；有明确运动段时才产生后续轨迹。

44　N50：把第一个工件点送到刀尖

```
N50 G1 X10 Y0 Z5 B0 C0 F600
```

1. G90生效：新的工件XYZ目标为(10,0,5)，不是在旧位置上增加。
2. B0/C0给出零角度；F600更新G94进给请求。
3. 加G54偏置：(10,0,5)+(40,20,10)=(50,20,15)。
4. 旋转角度为零、几何P/D/L均为零，逆解XYZ与K相同。
5. 关节目标得到(50,20,15,0°,0°)，随后由规划器从当前位置插补到此。

层次	数值
程序工件点W	(10,0,5)mm
运动学输入K	(50,20,15)mm
XYZ关节目标	(50,20,15)mm
B/C关节目标	0° / 0°

Y0和Z5的读法

Y0指向工件Y=0，不是把机床Y归零；本例机床Y目标实际是20。Z5是工件基准面上方5mm，前提是本课+Z方向朝该侧。

先算工件参考里的目标，再算偏置和运动学，不跳过中间层。

45　N60 C90：XYZ字没出现，XYZ关节却动了

```
N60 C90
```

G1、G90、G21、G54、F600仍然有效。只更新C的绝对目标为90°，工件XYZ仍保持(10,0,5)，B仍为0。运动学输入的XYZ继续为(50,20,15)。

$$q_X = 50\cos 90^\circ - 20\sin 90^\circ = -20$$
$$q_Y = 50\sin 90^\circ + 20\cos 90^\circ = 50$$
$$q_Z = 15$$

比较	N50末端	N60末端
工件XYZ	(10,0,5)	(10,0,5)
基础K的XYZ	(50,20,15)	(50,20,15)
关节XYZ	(50,20,15)	(−20,50,15)
B/C	0° / 0°	0° / 90°

工件原点本身也从(40,20,10)移动到(−20,40,10)。在新姿态下，再沿工件X走10、Z走5，就到刀尖目标(−20,50,15)。

这就是RTCP跟随

省略XYZ不表示禁止XYZ电机动作；它表示程序点不变。为了维持这个点相对工件的位置，XYZ关节可以需要运动。

程序量保持与关节量保持，是两个完全不同的约束。

46　N70 G91 X10：沿转后的工件方向走

```
N70 G91 X10
```

这一行把距离模式改为增量。原工件点W=(10,0,5)，现在沿工件+X增加10，得到W=(20,0,5)。B/C没有写，保持0°/90°。

步骤	结果
形成工件终点	(10,0,5)+(10,0,0)=(20,0,5)
转换到K	(20,0,5)+(40,20,10)=(60,20,15)
C=90°的逆解	关节XYZ=(−20,60,15)
相对上一步的关节变化	Δq=(0,+10,0)mm

所以程序写了X10，机床Y关节却增加10mm。这不是轴接错，而是“工件+X”当前指向机床+Y。图形上刀尖沿转动后的工件方向移动。

> **偏置不要重复累加**
>
> G91是在当前编程位置上增加位移。错误地把旧K位置再加一遍G54偏置，会让目标多出(40,20,10)。本课解析实测验证没有这种重复偏置。

N70之后G91仍有效。下一行如果继续写X20而不恢复G90，就又是增加20，而不是去X=20。

> 每次出现G90/G91，都应更新你的“当前模式表”。

47　N80：工件点不变，B倾转带来X/Z补偿

```
N80 G90 X20 Y0 Z5 B30 C90 F300
```

先恢复G90，工件XYZ绝对目标仍为(20,0,5)，与N70末端相同。C保持90°，B从0°变为30°，F改为300。K的XYZ保持(60,20,15)。

先按C=90°换算，得到(−20,60,15)；再按B=30°绕Y方向倾转：

$$q_X = (-20)\cos 30^\circ + 15\sin 30^\circ \approx -9.821$$

$$q_Y = 60$$

$$q_Z = -(-20)\sin 30^\circ + 15\cos 30^\circ \approx 22.990$$

此时工件原点的空间位置约为(−12.321,40,18.660)。工件X方向当前沿机床+Y；工件Z方向则含机床X和Z两个分量。把W=(20,0,5)按这些方向加到原点上，也得到同一个关节XYZ。

两种计算方式，结果相同

可以“先形成K，再按转台链变换”；也可以“求当前工件原点与基方向，再组合W点”。它们是同一几何关系的两种表达，不应产生两套不同答案。

倾转只改变B字，也可能要求直线轴联动；这正是补偿的任务。

48　N90：省略的Z、B、C保持当前目标

```
N90 X0 Y10
```

因为N80恢复了G90，这里的X0/Y10是工件绝对目标，而不是增量。Z没有写，保持5；B保持30°，C保持90°；G1、G54、G21、F300继续有效。

量	新的目标
完整工件XYZ	(0,10,5)
完整角度B/C	30° / 90°
K的XYZ	(40,30,15)
C旋转后的中间点	(−30,40,15)
B倾转后的XYZ关节	(−18.481,40,27.990)

完整程序在这一段后执行M2。真实WASM规划器跑到程序完成时，关节终点与上述逐字解析、逆解计算一致；不是只在纸上进行了三角函数演算。

省略不代表零

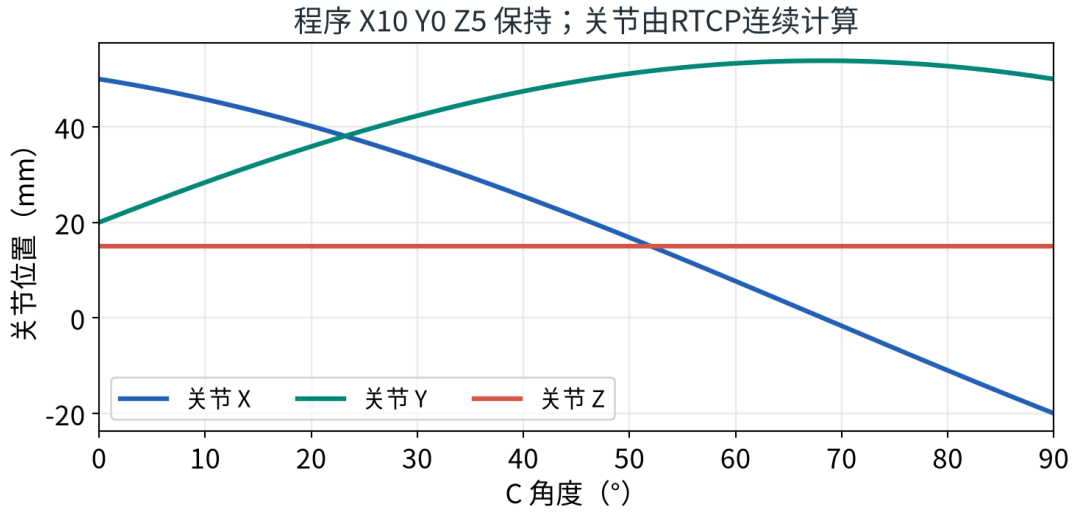
如果你把省略的Z读成0，会得到不同的点。把省略的B/C读成0更会改变整套空间方向。初学时可以把每行补成“完整目标表”，即使程序文本只写了两个字。

程序短，不代表有效状态少；把沿用状态补全后再计算。

49 把五组数字同时放在一张表里

段号	工件XYZ	K输入XYZ	B/C(°)	关节XYZ
N50	(10.000, 0.000, 5.000)	(50.000, 20.000, 15.000)	0 / 0	(50.000, 20.000, 15.000)
N60	(10.000, 0.000, 5.000)	(50.000, 20.000, 15.000)	0 / 90	(-20.000, 50.000, 15.000)
N70	(20.000, 0.000, 5.000)	(60.000, 20.000, 15.000)	0 / 90	(-20.000, 60.000, 15.000)
N80	(20.000, 0.000, 5.000)	(60.000, 20.000, 15.000)	30 / 90	(-9.821, 60.000, 22.990)
N90	(0.000, 10.000, 5.000)	(40.000, 30.000, 15.000)	30 / 90	(-18.481, 40.000, 27.990)

比较N50与N60：工件XYZ和K的XYZ都不变，关节随C改变。比较N60与N70：在C=90°时沿工件X增加10，反映为机床Y增加10。比较N70与N80：工件点相同，只倾转B，X/Z关节补偿。



N60整段C从0到90°：图中是XYZ关节变化，不是程序工件XYZ变化。

图中的横轴是C角度，不是时间。实际时间还受进给、加减速和机床约束影响，不能用图上每一度的等距刻度推断匀速运行。

> 一张正确的记录表，应区分程序坐标、运动学输入、角度和物理关节。

50　N100/N110：同一刀尖位置，坐标解释改变

```
N100 M429
N110 M428
```

N90末端关节固定为(−18.481,40,27.990,30°,90°)。M429把解释切到Identity，不能因此让这些关节无目标地跳动。新模式下基础XYZ与直线关节直接对应。

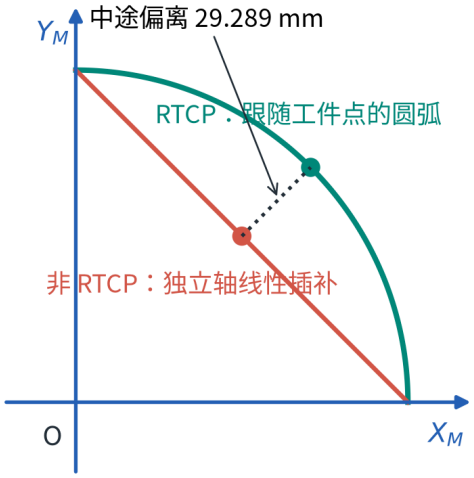
量	M429后	再M428后
物理XYZ关节	(−18.481,40,27.990)	保持相同
基础K的XYZ	(−18.481,40,27.990)	(40,30,15)
当前G54相对XYZ显示	(−58.481,20,17.990)	(0,10,5)
实际刀尖位置	不动	不动

在Identity下，当前软件坐标不再通过RTCP映射到转动后的零件局部坐标架。因此该模式的相对XYZ读数不能继续当作刚才物理工件架W中的同一点坐标来解释。

> **后续指令必须按新模式读**
>
> 切换本身不动，不代表切换后再发“X0”也不动。相同字值在不同解释模式中可能指向完全不同的机械目标。

切换时固定关节、重新解释坐标；不要为了保持旧XYZ数字而挪动机床。

51　RTCP要在运动过程中持续计算



图为半径100mm的零偏置例：两种方法终点相同，中途却相差29.289mm。

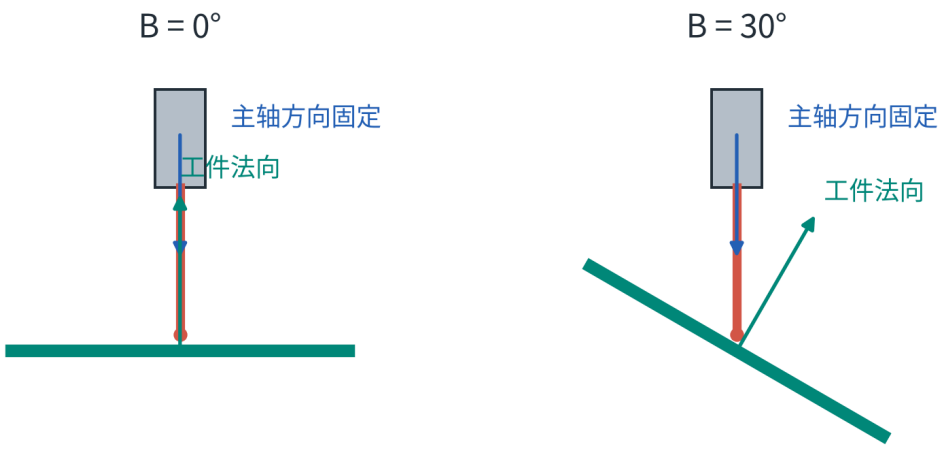
假设工件点绕C轴转90°。正确跟随时，刀尖走该点在机床空间里的圆弧。若只求首末关节位置，然后不顾几何地把X/Y用直线连接，可能走成弦线。

所以控制器不能只在段末才补偿。它需要在连续采样中使用当前插补出的XYZ和角度，得到相应关节值。各关节的变化不一定是线性的，也不必在每个时刻都非零。

> **五轴联动的正确含义**
>
> 系统具备同时协调五个自由度的能力，不意味着每条指令、每个时刻五根轴都必须运动。特殊几何下某些轴不动，可能正是正确答案。

只看“最后有没有对上”不足以检验RTCP。

52 RTCP不自动决定刀轴该朝哪里



双转台中主轴在机床里可保持竖直，但相对于工件的刀轴方向会改变。

本课机床不摆动主轴头。A/B、C带动工件后，主轴方向相对于工件发生变化。程序或CAM负责给出希望的角度/姿态，RTCP负责按这些角度保持所需刀尖位置关系。

RTCP不会仅凭一个曲面网格就自动知道应当以什么刀轴角加工，也不会自动保证刀轴始终垂直曲面。后者还涉及刀位规划、刀具几何、工艺策略和碰撞检查。

给定条件	仍需解决的问题
知道一个刀尖点	刀轴方向可能有多种选择。
知道点和A/B、C	还需用正确机床几何求关节。
关节在限位内	还不能证明刀具/夹具/机床无碰撞。

工件坐标方向、刀具方向、表面法向是三个概念。某些工艺要求它们满足特定关系，但它们不会天然全部重合。

> RTCP执行指定的几何关系，不能替代CAM工艺与碰撞规划。

53 预览画面中的三个对象要分开看

显示对象	代表什么	常见误判
机床网格	结构外形及由关节驱动的部件位姿。	把网格缩放误当成运动学标定。
工件/程序路径	表达程序几何，通常附着在工件分支。	把摄像机转动当作工件坐标改变。
刀尖/实时轨迹	显示当前采样的受控点或路径。	只看路径颜色，不核对实际关节/参考系。

正确结构中，X/Y/Z移动刀具侧，A/B→C移动工件侧。路径必须与其定义的参考系一致。如果C错误地挂到Z下面，XYZ就会把工件也带走；再正确的RTCP数字也无法弥补错误的显示树。

导入GLB后修改中心/偏置，应保留模型并同步有效URDF与WASM。参数改变可以使部件相对位置改变，但不应该悄悄替换成另一台内置机床。

> **显示不是第二套控制器**
>
> 不要在Three.js里编写另一套补偿公式把刀尖“画到看起来正确的位置”。关节应由真实运动学算出，图形只按同一结构进行刚体投影。

外观网格中的铸件、支架形状不一定随参数自动重建，模型校正也不等于通过了真实碰撞/刚度/制造尺寸验收。

> 验证要同时看程序参考、关节数据和场景层级，不能只凭视觉效果。

54　准备数据时，按三层顺序核对

第一层：机床自身

确认结构是AC还是BC、每个节点的父级、各轴正方向、回零基准、倾转中心与轴间偏置。轴向或结构类型错误时，先修复模型/配置，不用G54去“抵消”它。

第二层：刀具与受控点

确认主轴参考点、刀尖节点、刀长定义和补偿状态。把换刀之后的受控点映射建立正确，再判断实际加工位置。

第三层：零件装夹与编程

确认CAD/CAM原点、工件方向、G54等偏置、当前单位、绝对/增量方式和进给模式。用多个已知点、多个姿态核对，避免只在零角度看起来对。

检查方法	能发现的问题
零角度检查已知点	基本零点/长度/单位错误。
小角度分别转A/B和C	轴向、正负号、父子顺序问题。
不同角度反复定位同一点	旋转中心、偏置与刀尖标定问题。
重复换模式但不发运动	坐标同步错误导致的无故跳变。

> **不要用单点验证全部参数**
>
> 错误的结构参数可以在某一点被工件偏置抵消，却在其他点或其他角度暴露。标定必须覆盖足够的空间位置与姿态。

> 先机床，再刀具，最后工件；分层定位错误比同时改所有参数可靠。

55　实验一：观察“转动后的坐标方向”

1. 进入仿真，关机且空闲时导入AC或BC桁架示例，查看当前几何参数。
2. 完成开机和回零，在Identity下把刀尖放到离C轴线有一定距离的位置，避免特殊点导致XYZ补偿不明显。
3. 停止运动，选择M428。记录关节与物理刀尖没有瞬移，同时记录当前XYZ显示可能变化。
4. 旋转C一个小角度并停止。保持当前姿态，再做一个小的编程X增量，观察实际哪几根直线关节变化。
5. 比较工件方向与机床导轨方向：导轨没转，XYZ合成了工件方向。
6. 先停止，再切回M429，重复小幅直接轴Jog作为对照。

项目默认示例有非零中心/偏置，所以不要拿本章零偏置算例的目标关节值逐项硬比。对默认机床，正确比较的是“同一配置下的运动学与模型是否一致”。

> **每组记录四列**
>
> 当前模式；编程/显示XYZ与角度；J0…J4；刀尖相对工件标记点是否保持。只截图一个角度，不足以说明全部行为。

> 实验目标是理解参考系关系，不是让学生背住某一组关节数字。

56　实验二：G54与G90/G91的逐行记录

本实验适合教师准备的专用仿真配置：零中心、零轴间偏置、零刀长差，BC结构，回零后从已知状态开始。不得直接在真实机床套用。

1. 设置G21/G90/G94，写入G54偏置(40,20,10)，选择G54。
2. 执行M428，记录此时关节不变。
3. 执行N50到N90，每段等待运动结束，再记录工件坐标、K输入、关节。
4. 把结果与第49节表格对照，重点检查N60和N70。
5. 尝试只选择G55，不发轴字；观察读数变化但关节保持。
6. 恢复已知模式与偏置后再重复，检查结果是否一致。

测试点	应该看到
只选择/设置参考	没有自动的定位运动。
N60 C90	程序XYZ保持，关节XYZ变化。
N70 G91 X10	W的X加10；本例关节Y加10。
N80恢复G90	X20是绝对目标，不能再加20。

本讲义提供完整验证数据而非截图推测。如果现场结果不同，先核对几何与模态条件，不要立刻断言某根轴接错。

> 逐行记录让隐含模态变成看得见的数据。

57　实验三：原点移动，程序几何不改

目标：理解同一段零件程序如何适应不同装夹位置。先保持角度固定，避免初学者同时处理太多变量；再在RTCP下加入姿态对照。

1. 为G54设置第一套零位偏置，为G55设置第二套偏置，保持初始方向相同。
2. 选G54运行一条很短、经过检查的定位段，记录目标。
3. 只选G55，不发运动字，确认执行轴不会自动移动。
4. 再给同样的工件XYZ目标，观察物理目标随配准变化。
5. 停止后在适当角度下重复；说明原点和方向怎样一起进入RTCP。

> **配准变化不是机床标定变化**
>
> 移动工件装夹时通常更新工件设置；不要为了让两个工件都能加工而随意改动倾转轴物理中心。机床结构没有变，工件放置关系变了。

若工件不仅平移，还在夹具上旋转，需要另外确定初始方向。仅改变G54的XYZ平移不会自动把一条斜边校正成你期望的+X方向。

作业：让学生画出两套工件原点和同一局部点，先预测实际目标变化，再用仿真核对。解释预测为何成立比得到一次正确数字更重要。

> 工件坐标把零件几何与装夹位置分开，但方向配准仍不可省略。

58 位置不对时，从哪一层开始查?

现象	优先检查	不要马上做什么
全程序平移一块	G54/G55、附加偏置、装夹原点。	先别改所有转台参数。
距离约差25.4倍	G20/G21和数据单位。	别靠缩放模型掩盖。
小动作不断累加	G90/G91是否残留。	别把每次误差当随机漂移。
零角度对，倾转就偏	中心/偏置、刀长、正方向、结构链。	别只在一个角度改G54。
只开RTCP却不跟点	实际运动模式、目标参考、父子关系。	别只看按钮颜色。
切模式机床瞬移	是否固定关节并同步坐标。	别冻结画面掩盖。
Connection ERROR	首个Worker/协议/运动错误。	别把后续sequence_gap当成唯一根因。

查看第一个失败原因通常比反复点击更有效。显示错误可能是运动计算错误的后果，也可能是模型投影错误；两类问题应通过关节数据与物理结构分别判断。

> **一条诊断原则**
>
> 保持一层已知正确，只改变一个待验证因素。否则多个错误可能暂时互相抵消，无法解释修复是否真正有效。

先找最早的因果点，不要只处理最后出现的提示。

59 RTCP跟得上，还必须到得了

工件XYZ看起来没有变化，不等于机械关节没有运动。转台角度改变时，为保持同一工件点，某根XYZ直线关节可能需要大幅补偿，最终触及行程或速度限制。

约束	检查对象
编程/坐标限位	控制器定义的坐标允许范围。
关节限位	各机械轴实际能够达到的范围。
速度/加速度	每个执行轴的动态能力。
碰撞	刀具、工件、夹具、转台和机床实体间的空间关系。

本项目曾出现BC坐标范围较大，但B关节上限只有50°的情况。连续Jog越过该物理约束后，运动计算失败，又触发事件时钟错误。现已区分限位阻止与连接故障，并允许停止后反向退出。

> **不要把仿真边界扩大成硬件声明**
>
> 项目对RTCP耦合关节越界采用最后有效伺服样本停止。它不等同于真实设备的全路径预测减速、安全PLC、碰撞检测或硬件急停系统。

课堂先用小位移、小角度、低速度、空载模型理解概念；实际机床的操作与保护必须使用其正式流程。

RTCP正确解决几何映射，不自动取消行程、动力学和碰撞约束。

60　初学者最容易问错的八个问题

问题	简明回答
工件坐标原点一定是转台中心吗?	不是。零件原点由工艺选，转台中心是结构标定。
工件坐标的零点会变化吗?	自身读数恒为0；在机床空间的位置可随角度变化。
只选G54能让刀尖跟着转台吗?	不能。还需要正确结构、姿态与RTCP运动学。
写X10为什么Y轴动?	X可能是工件方向，需由固定导轨合成。
没写XYZ为什么XYZ动?	程序点保持，但转台姿态变化需要补偿。
RTCP时所有轴必须一直动吗?	不必，取决于轨迹和当前几何。
H3就是3mm刀长吗?	不是，完整LinuxCNC里H是刀补条目索引。
界面“Machine XYZ”就是J0–J2吗?	不能默认；非线性运动学中需核对显示定义。

再遇到一个看似矛盾的说法，先问它说的是哪个参考系、哪个受控点和哪个控制模式。许多矛盾只是省略了这些条件。

> 补全条件后再下结论，是理解五轴的基本方法。

61　练习A：原点、模式和地址字

1. 工件原点在零位K中为(40,20,10)，方向对齐。W=(10,0,5)对应的K是多少？
2. 当前位置W的X=10。分别执行G90 X20和G91 X20，各得到什么W终点？
3. 单独执行G54会不会自动走到工件零点？若同一行是G54 G1 X0呢？
4. 当前位置K=(70,30,20)，希望此刻W=(5,0,0)。在无附加旋转/补偿的教学条件下，L20算出的偏置o是多少？
5. T3、H3、S12000、F600分别属于哪种数据？哪些数字不能被直接当作毫米距离？
6. 工件原点测对了，但长边相对机床X倾斜20°。只改G54的X/Y/Z平移，是否足以纠正方向？

答题要求

每题至少写一句解释。不要只交三个数字；需说明参考系、模式和假设条件。可以画原点与方向箭头辅助推理。

这些题不需要矩阵，关键是把词语和参考对象对应起来。

62　练习B：从工件点推到物理关节

1. 零中心/偏置/刀长差，B=0、C=90°，o=(40,20,10)，W=(10,0,5)。求XYZ关节和工件原点的空间位置。
2. 保持上题姿态，执行G91 X10。哪根直线关节增加多少？
3. 从N70末端执行G90 X20 Y0 Z5 B30 C90，工件点是否变化？为什么X/Z关节仍会变？
4. M429后XYZ读数变化，关节和刀尖却不动，是正常还是故障？如果反过来XYZ不变、关节跳动呢？
5. 程序只改C，但XYZ没有动，一定说明RTCP坏了吗？给出一个可以正确不动的几何例子。
6. C轴错误挂在Z滑枕下，会造成什么显示/结构错误？能否通过开启RTCP修复？
7. 为什么只校验首末关节位置，不足以证明整段RTCP正确？

扩展题：用你自己的一个工件原点和目标点，手算C=0°与90°的物理位置；再说明旋转中心改成非零后，哪一步需要修改。

> 把中间层K写出来，可以明显减少漏加偏置和重复加偏置。

63　练习A答案

1. **K=(50,20,15)。**方向对齐、仅平移时，p_K=p_W+o。
2. **G90终点X=20；G91终点X=30。**前者是绝对目标，后者在当前10基础上加20。
3. **单独G54不定位。**G54 G1 X0同时含有运动目标，会结合当前坐标与模式形成轨迹。
4. **o=(65,30,20)。**当前位置K减去希望的W：(70,30,20)−(5,0,0)。
5. **T3为工具选择号，H3为刀补条目索引，S12000为主轴速度请求，F600由进给模式解释。**不能将这些数全当作mm。
6. **不够。**三个平移参数只改变原点，不能补偿初始方向旋转。需机械找正或使用受支持的坐标旋转/方向配准。

> **进一步检查**
>
> 第4题的公式只在声明的简单条件下使用。存在旋转、刀补或非平凡运动学时，必须使用相应参考下的当前位置，而不是直接拿原始关节读数代入。

> 答案的条件是答案的一部分，不能省略。

64　练习B答案

1. **XYZ关节=(−20,50,15)，工件原点O_M=(−20,40,10)。**先求K=(50,20,15)，再绕C90°。
2. **机床Y增加10mm。**当前工件+X就是机床+Y，新的关节为(−20,60,15)。
3. **工件点不变，仍是(20,0,5)。**B姿态改变，新的XYZ关节约为(−9.821,60,22.990)。
4. **坐标重新解释、关节保持是正常。**单纯切换却改变物理关节，一般说明同步/切换逻辑有问题。
5. **不一定。**点在C旋转轴线上且其他条件不变时，只转C可以不改变该点的机床空间位置。
6. **工件会错误地随XYZ链移动。**必须修复结构父子关系，开关RTCP不能替代正确拓扑。
7. **端点相同，中间路径仍可能不同。**例如圆弧与弦线；必须检查连续采样和相对工件轨迹。

> 每个答案都可以画成“原点 + 方向 + 点”的几何关系。

65　把常见名词翻译成容易理解的话

术语	初学者解释
坐标系/参考系	放在某处、有指定方向和单位的一套尺子。
原点/零点	这套尺子的起点，不一定是电机回零位置。
工件偏置/WCS offset	让零件程序与当前装夹参考对齐的数据。
关节/joint	被控制的机械自由度及其位置。
正运动学/FK	由关节与结构计算对应坐标/受控点。
逆运动学/IK	由目标点、姿态和结构计算关节目标。
姿态/orientation	方向关系；本课由A/B、C指定。
TCP/RTCP	与刀尖目标、姿态变化补偿相关的功能；需核对控制器定义。
插补/interpolation	在段起点和终点间连续生成中间目标。
模态/modal	指令生效后持续保留，直到被同组设置替换。
配准/registration	建立设计坐标与实际装夹之间的对应关系。
同步/synchronization	让解释器、规划器和运动状态使用一致的当前参考。

> 看到新术语，先找它属于“几何、程序、机械、执行”哪一层。

66　用最少公式描述完整坐标链

以下公式供进阶阅读，不是初学者读前面章节的前提。向量用列向量，R表示正交旋转矩阵；长度统一为mm。

1. 初始工件配准

$$p_K = R_0\, p_W + o$$

o给出工件原点在零位参考中的位置；R_0给出初始工件方向。本章实算$R_0=I$，因此只做平移相加。

2. 当前双转台姿态与几何，L=0

$$r_M = P + D + R_tilt\,[\, R_C\,(p_K - P) - D\,]$$

AC取R_tilt=R_X(A)，BC取R_tilt=R_Y(B)。因此可写为r_M=O_M+E_MW p_W，其中：

$$E_MW = R_tilt\, R_C\, R_0$$

$$O_M = P + D + R_tilt\,[\, R_C(o - P) - D\,]$$

这里r_M是本课零刀长差模型的目标空间位置，并对应直线关节目标。加入刀具标定后还需从刀尖求主轴/执行参考；任意机床结构不能直接套用这一特定式子。

> 原点项O_M与方向项E_MW同时随姿态变化，分开写有助于理解。

67　反过来：从当前物理位置求工件坐标

在姿态与几何已知、没有奇异歧义且标定一致时，前页r_M=O_M+E_MW p_W可以反算：

$$p_W = E_MW^{T} (r_M - O_M)$$

先从当前空间位置减去当前工件原点，得到相对位移；再用逆旋转，把这个位移投回工件三个方向。正交旋转矩阵的逆等于转置，所以这里写上标T。

这正是“坐标数值可以变，而物理点不变”的数学基础。同一r_M在不同O/E下有不同p_W。模式切换、工件系选择或重新设置参考时，必须明确变了哪一层。

本项目刀长差的特殊约定

$$D_L = D + L \cdot e_z$$

$$q_XYZ = P + D_L + R_tilt\,[\, R_C(p_K - P) - D_L \,]$$

本项目原生TRT的L是master-tool刀长差约定。这个式子返回关节目标，不是任意GLB刀尖网格的世界位置。完整工具标定、工件架与刀尖显示须遵循同一个约定；其他控制器应按其文档重新建立链路。

> 先减原点、再投影方向；关节到刀尖之间还可能存在刀具参考变换。

68 这份讲义验证了什么，没有声称什么？

内容	本次证据
G10 L2、G54/G55平移、G90/G91	实际项目WASM逐段解析，核对规范目标。
G20长度/G94直线进给换算	X1→25.4mm，F2→50.8mm/min实测。
G10 L20位置重定义	偏置计算及后续目标实测。
RTCP BC例子的目标关节	真实WASM逆解计算；关键数值强断言。
N10–N90完整运动	真实Session/解释器/TP执行完成，终点与逐行结果一致。
M429/M428保持物理关节	原生切换接口+坐标同步逐段验证。
G52/G92、任意方向变换、完整刀表与所有F模式	依据官方语义讲解，不冒充Web切片已全面实现。

项目调用实际WASM并不能自动证明它覆盖了完整LinuxCNC。特别是本地解释器适配层属于功能切片，不能把接受某个字母或设置内部字段当作完整语义和工艺正确性证据。

> **对照两层资料**
>
> 学习通用指令时查官方文档；复现本课数字时查同目录gcode-evidence.json和具体测试条件。两者各自解决不同问题。

> 严谨的教学需要同时给出结果和结果成立的范围。

69　官方资料、验证数据与自学路径

[S1] LinuxCNC · 5-Axis Kinematics
用于核对AC/BC机构、前后向变换、旋转中心、偏置与刀具参考。

[S2] LinuxCNC · Coordinate Systems
用于核对机床/工件坐标、G54–G59.3及附加偏置。

[S3] LinuxCNC · CNC Machine Overview
轴命名、受控点、单位、旋转与协调运动。

[S4] LinuxCNC · G-Codes
G10、G20/G21、G43、G53、G90/G91、圆弧/进给等指令语义。

[S5] LinuxCNC · G-code Overview
程序段、地址字、表达式、模态组及执行顺序。

[S6] LinuxCNC · Switchable Kinematics
切换模式及解释器/运动同步。

本地依据：sim_interpreter.c、sim_program.c、trtfuncs.c、sim-session.mjs，以及428remap.ngc/429remap.ngc。图示为原创教学图和项目原创GLB渲染。官方页面核对日期：2026-09-18。

> **随讲义提供**
>
> handbook.html为可编辑源稿，outline.json为逐页目录，gcode-evidence.json保存逐字结果与完整轨迹，verification/保存版面/PDF核验。原31页投影片保留作课堂概要，本讲义作为可独立阅读的完整版。

最后记住：先定义原点和方向，再解释程序数值，最后由运动学求出各轴运动。