Twinwright Studio 1.0.2

Twinwright Studio manual

2026-10-11

目次

User manual

1. はじめに
2. ウィンドウとワークスペース
3. 機械と電源
4. ジョブの段取り
5. プロービングとワーク原点の検出
6. プログラムの作成（CAM）
7. プログラムとエディタ
8. プログラムの実行と再開
9. 分析と診断
10. ツインでのプログラム実行
11. 検査、レポート、段取りシート
12. 実機の接続
13. AIアシスタント
14. 言語と単位
15. キーボードとアクセシビリティ
16. トラブルシューティング

Operations manual

1. システム要件
2. インストールとアンインストール
3. ライセンス、シート、アクティベーション
4. データとバックアップ
5. Studio のアップデート
6. プライバシーと Studio が送信するもの
7. ラボ、教室、共用コンピューター

Technical manual

1. ツインのモデルとその限界
2. Twinwright 独自カーネル
3. G コードと M コード
4. CAM の加工方式とパラメータ
5. はめあい、振れ、検査方法
6. 機械プロファイル
7. MC-1 のハードウェア

Reference

1. MCP tool reference
2. 設定リファレンス
3. アラームコード
4. エラーコード
5. Command line
6. Machine profile schema
7. プロトコル
8. ファイル形式
9. Twinwright 独自カーネルで使う送信ソフト

User manual

はじめに

Twinwright Studio は、お使いの CNC 機械のデジタルツインを動かします。部品を段取りし、プログラムを作成するか開き、実機で切削する前にツインで実行します。この章では、インストールから最初の部品までを約 10分で説明します。

インストール

Studio には 64ビット版の Windows 10 または 11 が必要です（システム要件を参照）。

1. `TwinwrightStudio-<version>-setup.exe` を実行します。管理者権限は不要です。Studio はお使いのユーザー用にインストールされます。
2. ライセンス契約を読み、**I Agree** をクリックします。
3. インストール先のフォルダーはそのままにするか、別のフォルダーを選びます。
4. 最後のページで **Run Twinwright Studio** にチェックを入れたままにして、**Finish** をクリックします。

Studio はスタートメニューに **Twinwright Studio** として登録されます。複数のコンピューターへのインストールと、何がどこに置かれるかについては、インストールとアンインストールを参照してください。

初回起動

Studio は **実行** ワークスペースで開き、機械 **MC-1・マイ作業台** の電源が入っています。何も尋ねられることはありません。

機械バーには **ALARM:11** と **次の手順：原点復帰** が表示されます。これは正常です。実機と同様に、ツインも原点復帰が済むまでは各軸の位置を把握していません。

ウィンドウの表示内容

新規インストールでは、エンジニアが使うものがすべて表示されます。6つのワークスペースすべて、グラフ、色分けです。プログラムの実行だけを行う場合や教育に使う場合は、表示の少ないレベルを選びます。**ツール > UIレベル** を使います（ユーザーレベルを参照）。

言語と単位

Studio は、システムの言語に対応している場合（英語、中国語、フランス語、ドイツ語、日本語、スペイン語）はその言語で起動し、それ以外は英語で起動します。単位と数値の表記は、お使いの地域に従います。

変更するには、**ツール > Language / 语言** と **ツール > 単位** を使います。言語と単位を参照してください。

エディションと試用

お使いのエディションは上部のバーに表示されます（例：**Learning**）。確認するには、それをクリックするか、**ヘルプ > ライセンス...** を使います。

エディション	追加されるもの
Learning	ポケット加工、輪郭加工、穴あけ、彫刻、3D 平行仕上げ。ツイン全体、エディタ、AI
Maker	アダプティブ加工、すべての 3D 仕上げ方式、残り加工、両面加工、複数の段取り
Pro	ネスティング、プログラム最適化、レポート（検査レポート、段取りシート）、プローブを使った設計に対する検査

各エディションには、表でそれより上にあるエディションのすべてが含まれます。上位エディションの機能には、そのエディション名が表示されます。クリックすると、その機能の内容を読めるほか、試用の開始やライセンスの画面を開くことができます。

試用期間またはライセンスが終了すると、Studio は Learning エディションに戻ります。作成したプロジェクト、プログラム、G コード は、そのまま使えます。

試用の開始

Maker エディションと Pro エディションは、それぞれ 30日間、コンピューター 1台につき 1回、試せます。

1. そのエディションの機能をクリックします。例：**プログラム** ワークスペースの **最適化**。
2. **30 日間試用...** をクリックします。
3. **試用を開始** をクリックします。

Studio は Twinwright のライセンスサーバーに試用を 1 回申請し、エディション、このコンピューターのフィンガープリントのハッシュ、Studio のバージョンを送信します。試用が終了すると、**メッセージ** タブでお知らせします。

ライセンスの有効化

ヘルプ > ライセンス... を開きます。

オンラインで有効化するには：

1. アクティベーションコード（`TW-XXXXX-XXXXX-XXXXX-XXXXX`）を入力します。大文字と小文字、スペース、ハイフンは問いません。
2. **有効化** をクリックします。結果：**有効化しました。**

インターネットに接続していないコンピューターを有効化するには：

1. **オフラインで有効化…** の下にある、このコンピューターの要求コードをコピーします。
2. インターネットに接続できるコンピューターで、ライセンスのページを開き、要求コードを貼り付けて、ライセンスファイル（`.twlic`）をダウンロードします。
3. そのファイルを持ち帰り、**ライセンスファイルを読み込む…** をクリックします。

期限前に更新 は、ライセンスの期限が切れる前にオンラインで更新します。送信するのは、ライセンスの番号、このコンピューターのフィンガープリントのハッシュ、Studio のバージョンだけです。

シート、別のコンピューターへの移行、教室での利用については、ライセンスを参照してください。

10分で最初の部品を作る

最初の部品として最も手早いのは、サンプル部品を使う方法です。素材、工具、ワーク原点は Studio が設定します。

機械を開く

機械はすでに開いています。**実行** ワークスペースの **MC-1・マイ作業台** です。ほかの機械は **ファイル > 機械ライブラリ…** にあります。

電源投入と原点復帰

原点復帰（F3）を押すか、機械バーが示すキーを押します。

結果：3つの軸が順に原点復帰し、約 20 s で完了します。機械バーに **アラームなし** と表示されます。

機械バーに **非常停止** と表示されている場合は、まず非常停止を解除し（F1）、続けて **電源投入**（F2）と **リセット**（F12）を押します。

サンプルプログラムを開く

1. **ファイル > サンプル部品…** を選びます。
2. **NEMA 17 モーター取付板**（e01_motor_mount）を選びます。はめあい穴、座ぐり、長穴のあるプレートで、タブで切り離します。結果：Studio が素材を配置し、工具とプログラムを読み込み、素材上にワーク原点を設定します。サンプルの準備ができると、コンソールに表示されます。

ツインで実行する

1. **サイクルスタート**（F5）を押します。
2. より速く確認するには、**シミュレーション > 速度 > 実時間の 10 倍** を選びます。

3Dビューに、工具が素材を切削する様子が表示されます。**フィードホールド**（F6）でプログラムが一時停止し、**サイクルスタート** で再開します。

結果を確認する

- 3Dビューをドラッグすると回転し、ホイールで拡大・縮小します。
- **表示 > 色分け > ワーク：部品との偏差** は、まだ削っていない部分と削り過ぎを色分けします。
- **分析**（Alt+5）には、どの行に時間がかかったかと、ツインによる力とびびりの推定値が表示されます。

素材を設定する

自分の部品の場合は、サンプルを読み込む代わりに **準備**（Alt+1）で機械を設定します。

1. **段取り** のステップに進みます。
2. **ワークと段取り** の下で、**形状**、**材質**、**治具** を選び、寸法を入力します。
3. **素材を配置** をクリックします。

ジョブの段取りを参照してください。

工具を測定する

1. **工具** のステップに進み、工具を選んで **工具交換** をクリックします。
2. **基準工具を測定** をクリックします。

機械が原点復帰済みで、主軸が停止している必要があります。

ワーク原点を設定する

1. **原点** のステップに進み、原点にしたい位置まで工具をジョグします。
2. **全軸をゼロに** をクリックするか、軸を 1つずつゼロにします。

タッチプローブを使う場合は、プローブカードが原点を見つけます（プロービングを参照）。

続いて、プログラムを開き（**ファイル > プログラムを開く…**）、上記と同じように実行します。

次に読む章

- ウィンドウとワークスペース：どこに何があるか。
- プログラムの作成：Studio の CAM。

- ツインでのプログラム実行：ツインが確認すること、推定すること。
- プログラムの実行と再開：実行、アラーム、やり直し。

パネルの **?** を押すと、そのパネルに対応する章でこのマニュアルが開きます。**ヘルプ > ドキュメント** を選ぶと、先頭から開きます。

ウィンドウとワークスペース

Studio のウィンドウは、上部にメニューとトップバー、中央に 6つのワークスペース、下部に機械バーがあります。ワークスペースはそれぞれ、1つの主な作業を受け持ちます。機械バーには、どのワークスペースでも、機械の状態と次の手順が表示されます。

ウィンドウの各部

上から順に：

- **タイトル行**：左にメニュー、中央にウィンドウの名前、右にウィンドウのボタン。
- **トップバー**：機械、ワークスペースのタブ、エディション、AI の状態、ユーザーレベル、**?**。
- **ワークスペース**：選択したワークスペースの 3Dビューとパネル。
- 3Dビューの下の **ドック**（ドックのあるワークスペースのみ）。
- **機械バー**（どのワークスペースでも共通）。

Studio にステータスバーはありません。その役目は機械バーが果たします。

タイトル行とメニュー

メニューは次のとおりです：**ファイル**、**編集**、**表示**、**機械**、**プログラム**、**シミュレーション**、**AI**、**ツール**、**ヘルプ**。

- Alt とメニューの文字を押すと、そのメニューが開きます。Alt+F で **ファイル** が開きます。
- Alt だけを押すとメニュー名が強調表示されます。そのあと、矢印キー、Enter、Esc で操作します。

メニューを開いている間は、どのキーも機械に届きません。メニューには、お使いのユーザーレベルの項目だけが表示されます（ユーザーレベルを参照）。

トップバー

左から右へ：

- ブランドマーク：クリックすると **ファイル > 機械ライブラリ...** が開きます。
- 機械：そのモデル、名前、状態。クリックするか Ctrl+Shift+M を押すと、機械を切り替えられます。文字を入力して絞り込み、↑ ↓ と Enter で選びます。
- ワークスペースのタブ（6つのワークスペースを参照）。
- エディション（例：**Learning**）：クリックすると **ヘルプ > ライセンス...** が開きます。
- AI の状態：**AIオフ**、**AIオン**、**AI接続中**、**AIが承認待ち**。クリックすると AI の設定が開きます（AIアシスタントを参照）。
- ユーザーレベル：クリックして別のレベルを選びます。
- **?**：キーボードショートカットの一覧。

機械バー

機械バーはウィンドウの下端に沿って伸びています。左から右へ：

1. **状態**：**アイドル**、**実行中**、**ジョグ**、**原点復帰中**、**ホールド**、**扉**、**工具交換**、**チェックモード**、**スリープ**、**アラーム**、**非常停止**、**電源オフ** のいずれかです。
2. **プログラム**：名前、実行中の行、いまいるセクション、進捗バー、経過時間と残り時間。プログラムがない場合は、**プログラムが読み込まれていません**、機械が原点復帰済みかどうか、工具、ワーク座標オフセットを表示します。
3. **次の手順**：緑の **アラームなし**、または **次の手順：** とそれを実行するキー。アラーム中は、アラーム、次の手順、**すべての手順** を表示します。**すべての手順** は、作業に戻るまでの手順を並べたものです（アラームと対処を参照）。
4. **時計**：シミュレーションの時刻、その倍速、実際に出ている速度。砂時計のキーでシミュレーションを一時停止します。
5. **キー**：**シングルブロック**、**サイクルスタート**（F5）、**フィードホールド**（F6）、**停止**（F7）。Studio が示すキーは塗りつぶされ、いま使えないキーは薄く表示されます。
6. **非常停止**（F1）：右端の丸い赤いキー。

ウィンドウが狭いと、バーは次の順に表示を省きます。まず時計の時刻、次にプログラムのセクションと時間（ツールチップに移ります）、次に時計の速度、最後に次の手順の文言です。

3Dビュー

3Dビューには、機械、ワーク、工具の現在の様子が表示されます。

- 左ボタンのドラッグでビューを回転し、右ボタンまたは中ボタンのドラッグで移動します。ホイールで、ポインターに向かってズームします。
- 左上のキーで視点を選びます。アイソメトリック、正面、右、上です。**工具に追従** は工具を中央に保ちます。**表示** は、工具先端の軌跡とプログラムの経路の表示を切り替えます。
- **工具の拡大表示** は、切削点にある工具を右上の小さな画像で表示します。**実行** ワークスペースでは、プログラムの実行中に表示されます。
- 下部のチップには、ワークと工具の名前が表示されます。
- ワークまたは軌跡をクリックすると、ビューの下のバーに、その点を作ったプログラムの行が表示され、リストがその行に移動します。

同じ選択肢は **表示 > 3Dビュー** にもあります。**表示 > 色分け** は、軌跡とワークを、送り速度、切削抵抗、切りくず厚さ、びびり指標、プログラムの行のいずれかで色分けします（エンジニアレベル以上）。

ドック

ドックは 3Dビューの下にあります。タブはワークスペースによって異なります：

ワークスペース	タブ
実行	**プログラム**、**グラフ**、**コンソール**、**メッセージ**、**AIアクティビティ**
準備	**測定結果**（部品を測定した後）、**工具テーブル**

ワークスペース	タブ
CAM	**検査結果**、**AIアクティビティ**、**バージョン**、**プログラムプレビュー**
プログラム	**コンソール**

分析、**診断**、工場モードにはドックがありません。

- **プログラム**：プログラムのリスト。**実行中のブロックに追従** は、実行中のブロックが見える状態を保ちます。時間バーには、今回または前回の実行での各行の時間が表示されます。
- **グラフ**：実行中の測定値の時間変化。
- **コンソール**：Studio とコントローラが出力した内容。下部の入力行に G コード の 1行または `$` コマンドを入力し、Enter を押すと送信されます。↑ と ↓ で、前の行を呼び出せます。
- **メッセージ**：警告と検出事項を、発生元ごとにまとめて表示します。タブの赤い数字は警告の数です。
- **工具テーブル**：機械の工具。
- **測定結果**：タッチプローブで最後に測定した部品を、検査項目ごとに表示します。設計値、公差、測定値、偏差、ツインの推定値（検査を参照）。
- **AIアクティビティ**：AIクライアントが、このコンピューター上の機械で行った操作。

Ctrl+J を押すか、タブの右端の矢印をクリックすると、ドックがタブの行だけの状態に折りたたまれます。

6つのワークスペース

ワークスペースは、トップバーのタブ、Alt+1 から Alt+6、Ctrl+Tab と Ctrl+Shift+Tab、または **表示** メニューで切り替えます。**CAM** または **プログラム** で **サイクルスタート** を押すと、**実行** に移ります。プログラムの実行中は、**実行** タブに緑の点が付きます。

準備

準備（Alt+1）は、機械と部品を 1ステップずつ整えます。ステップは **電源**、**原点復帰**、**段取り**、**工具**、**原点**、**プログラム**、**加工** の順です。右のパネルは、ステップを自分で選ぶまで次の手順に追従します。ジョブの段取りを参照してください。機械、電源、原点復帰については、機械と電源を参照してください。

CAM

CAM（Alt+2）はプログラムを作ります。左にジョブの段取り、形状、加工工程、右に選択した項目の入力欄、下に **プログラム生成** があります。プログラムの作成を参照してください。エンジニアレベル以上です。

プログラム

プログラム（Alt+3）には、プログラムとサンプル部品、エディタ、プログラムのチェック結果を含むプレビューがあります。プログラムとエディタを参照してください。

実行

実行（Alt+4）は、読み込んだプログラムを実行します。右の列には、位置、主軸と送り、オーバーライド、切削、操作盤のキー、手動操作、工具とゼロ位置が表示されます。プログラムの実行と再開を参照してください。

分析

分析（Alt+5）は、実行を振り返ります。そのグラフ、各行の時間と力、安定限界線図です。エンジニアレベル以上です。分析と診断を参照してください。

診断

診断（Alt+6）には、機械の電源系統、コントローラの入力、ドライバが表示され、ツインに故障を注入できます。テクニシャンレベル以上です。分析と診断を参照してください。

メッセージと次にすること

Studio は、何が起きたか、次に何をすればよいかを、次の場所で伝えます：

- 機械バー：次の手順、またはアラームと作業に戻るまでの手順。
- ドックの **メッセージ** タブ：警告と検出事項。
- **コンソール**：コントローラと Studio からの各行。
- **ツール > 監査ログ...**（Ctrl+L）：すべての記録。誰がいつ行ったかも分かります。

機械 > 次の手順を実行 は、機械バーが示す手順を実行します。その手順だけです。アラームでプログラムが止まった場合、手順の最後に、やり直しを始めるブロックが示されます（ブロックからの再開を参照）。

ユーザーレベル

ユーザーレベルは、その段階で必要なものを表示し、それ以外を隠します。レベルは 5つあります：

レベル	追加されるもの
デモ	サンプル部品のある **実行** ワークスペース。**プログラム** タブと **メッセージ** タブ
オペレータ	**準備** と **プログラム**。オーバーライド、操作盤、手動操作、工具とゼロ位置。**機械** のサブメニュー。コンソール、工具テーブル、**AIアクティビティ**。設定、監査ログ、**AI** メニュー。機械の切り替え
テクニシャン	**診断**。ワークの配置の調整。**工場出荷時にリセット...**
エンジニア	**CAM** と **分析**。切削の測定値、グラフ、**色分け**、グラフの書き出し、比較する部品

レベル	追加されるもの
研究者	**診断** の **モデルとデータ** カード

各レベルには、表でそれより上にあるレベルのすべてが含まれます。新規インストールは **エンジニア** で始まります。

レベルを変更するには、**ツール** > **UIレベル**、トップバーのレベル、または **ツール** > **設定...** > **全般** を使います。

> **補足**
>
> レベルは、教えるときや学ぶときのために、ウィンドウの表示を整えるものです。何もロックしません。機械のキーは、どのレベルでも使えます。

ライト、ダーク、工場モード

表示 > **外観** で、**システムに合わせる**、**ライト**、**ダーク** のいずれかを選び、**工場モード**（**大**） のオンとオフを切り替えます。2つの選択は独立しています。工場モードは、どちらの外観でも使えます。

Studio 自身のパネルは、すぐに切り替わります。システムが描画するリストとテキストボックスは、次回の起動時に新しい外観になります。

工場モードは、機械のそばで、腕を伸ばした距離から操作するためのものです：

- 機械バー、そのキー、非常停止が大きくなります。テキストとリストも大きくなります。
- **実行** には専用のレイアウトがあります。左に位置、プログラム、大きなリスト、3Dビューの下にゲージ、右にオーバーライド、操作盤のキー、メッセージが並び、ドックはありません。
- 3Dビューの視点キーは 2つで、工具の拡大表示はありません。

パネルとそのサイズ

2つのパネルの境界をドラッグするとサイズを変更できます。境界をダブルクリックすると、設計どおりのサイズに戻ります。3Dビューは、表示倍率 100 % で少なくとも 360 × 240 ピクセルを保ちます。パネルの境界を動かせるのはマウスだけです。

Studio は、機械ごとに、パネルのサイズ、ドックのタブと折りたたみの有無、最後にいたワークスペースを記憶します。次回の起動のために、ウィンドウの位置とサイズも記憶します。

機械と電源

Studio の機械、機械の切り替え方、そして機械が作業を始める前の電源投入と原点復帰について説明します。

機械

各機械は、それぞれ独自の状態一式を保持します。位置、ワーク、工具とオフセット、温度、稼働時間です。ウィンドウに表示されている機械が動作する機械で、ほかの機械は保存された状態のまま待機します。1.0 では、同時に動作する機械は 1台です。

機械の切り替え

1. トップバーの機械をクリックするか、Ctrl+Shift+M を押します（**ファイル > 機械を切り替え...**）。
2. 文字を入力して機械を探し、↑ ↓ で選んで Enter を押します。結果：Studio は、切り替え前の機械を保存し、もう一方の機械で再び起動します。その機械はコールドスタートになります。電源を投入し、原点復帰してください。

プログラムの実行中は切り替えられません。一覧には、AIクライアントがバックグラウンドで使う **練習用機械** も表示されます。**表示（読み取り専用）** を選ぶと、**AIアクティビティ** に表示されます。

機械ライブラリ

ファイル > 機械ライブラリ...、またはブランドマークのクリックで、お使いの機械が、その状態、ワーク、工具とともに表示されます。機械のモデルも表示されます。ブランドマークをもう一度クリックすると戻ります。

選択した機械について：

- **開く**：その機械に切り替えます。
- **名前の変更...**：独自の名前を付けます。
- **新しい機械としてコピー...**：この機械の状態一式を最後に保存した時点のまま持つ、新しい機械。
- **フォルダーを表示**：ディスク上にある、その機械の状態一式。
- **工場出荷時に戻す...（5 s 以内に 2 回クリック）**：出荷時のままの機械（機械の初期状態へのリセットを参照）。

新しい機械... は、そのモデルの機械を出荷時の状態で追加します。新しい機械、コピー、工場出荷時に戻す操作には、テクニシャンレベルが必要です。**MC-1 実機** の行は、将来のために確保された場所です。1.0 では、実機はコマンドラインから接続します（実機を参照）。

電源投入と原点復帰

電源投入

ツインには MC-1 の電源系統があります。主電源スイッチ QF1 がコントローラに 24 V を供給し、**電源投入** がコンタクタ KM1 を閉じて、ドライバに 48 V を供給し、主軸インバータに給電します。非常停止は KM1 を開きます。

1. **電源** のステップに進みます。
2. 主電源スイッチがオフの場合は、**主電源スイッチ オン** をクリックします。
3. キーに **押下中** と表示されている場合は、非常停止が押された状態です。そのキーをクリックするか F1 を押して、解除します。
4. **電源投入**（F2）をクリックします。結果：パネルに 48 V がオンになったことが表示され、各ドライバが **準備完了** になります。次の手順は **原点復帰** です。

原点復帰

原点復帰（F3）をクリックします。最初に Z が上がり、次に X と Y が同時に、それぞれ移動範囲の端にあるスイッチまで動きます。実時間では約 20 s かかります。

結果：**原点復帰済み：X Y Z** と表示され、ソフトリミットが有効になります。電源投入後に原点復帰するまで、コントローラは ALARM:11 の状態で、ツールセッター、プローブ、**ワーク原点へ移動** は待機します。

ソフトリミットとストローク

MC-1 のストロークは、X が 200 mm、Y が 150 mm、Z が 80 mm です。原点復帰後、各軸の機械座標は、その軸の原点側の端で 0 になります。X は −200 から 0、Y は −150 から 0、Z は −80 から 0 までで、0 は Z の最上部です。

- リミットを越えるジョグは実行されません（error:15）。ジョグを押し続けた場合は、リミットの手前で止まります。
- リミットを越えるプログラムの行があると、プログラムは ALARM:2 で停止します。位置は保持されます。**リセット**、**ロック解除** の順に操作し、そのあとプログラムとワーク原点を確認します。
- 軸がリミットスイッチに当たると、ALARM:1 が発生します。**リセット**、**ロック解除** の順に操作し、スイッチから離れるようにジョグして、**原点復帰** します。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

ジョブの段取り

準備 ワークスペースは、機械と部品を 1ステップずつ整えます。電源、原点復帰、素材とその固定方法、工具、そしてワーク原点です。どのステップも先にツインに表示されるので、何かが動く前に段取りを確認できます。機械、電源、原点復帰は別の章で説明しています：機械と電源。

準備ワークスペース

準備（Alt+1）は、左にステップ、中央に 3Dビュー、右に選択したステップのパネルがあります。ビューの下のドックには **工具テーブル** があります。

ステップ

ステップ	そのパネル	完了の条件
電源	主電源スイッチ、**電源投入**、**リセット**、**ロック解除**	コントローラが動作し、ドライバに電力が供給されている
原点復帰	**原点復帰**、**ロック解除**、**リセット**	3つの軸すべてが原点復帰済みである
段取り	素材、その固定方法と位置	素材が設定されている、またはテーブルを空のままにしている
工具	主軸の工具、ツールセッター	基準工具が測定されている、工具テーブルが読み込まれている、またはワークの Z 原点が設定されている
原点	ジョグ、ワーク原点、プローブカード	ワーク座標オフセットが設定されている
プログラム	**プログラムを開く...**、**サンプル部品...**、**プログラムを検査**	プログラムが読み込まれている
加工	**サイクルスタート**、**シングルブロック**	プログラムが最後まで実行された

完了したステップには緑のチェックマークが付き、次のステップの端には小さな矢印が付きます。パネルは、別のステップをクリックするまで次のステップを表示します。次のステップをクリックすると、パネルが再び追従します。ステップの下の行には、いま何をするかが書かれています。パネルの **?** は、このマニュアルのそのパネルに対応する箇所を開きます。

段取り中の 3Dビュー

素材の入力欄を変更している間、設定中の素材は、テーブル上に透けた状態で描かれます。そのまま設定できる場合は緑、できない場合は赤です。**素材を配置** で素材がそこに置かれます。**元に戻す** または Esc で、変更を破棄します。

素材

形状、材質、寸法

段取り のステップの **ワークと段取り** カードで：

1. **形状** を選びます：**直方体**、**丸棒（縦置き）**、**パイプ（縦置き）**、**六角棒（縦置き）**、**丸棒（横置き）**、または **STLファイル（形状を読み込み）...**。
2. **材質** を選ぶか、別の規格での名称を入力します（C45、EN AW-6061、SUS304）。一覧には、Studio が切削条件を採用するグレードが表示され、段取りには入力した名称がそのまま保持されます。
3. 形状に必要な寸法を入力します：**長さ L**、**幅 W**、**高さ H**、**直径 D** など。
4. 固定方法と位置（後述）を選び、**素材を配置** をクリックします。結果：パネルに、テーブル上にあるものとその質量が表示されます。

寸法は表示単位で指定します。インチでは、入力欄に分数も使えます（インチの分数を参照）。

> **補足**
>
> 別の規格の名称は、近いグレードとして扱われます。切削条件は、一覧の中で最も近いグレードのものです。材質は、供給元の証明書に記載されているものです。

STL ファイルからの素材

形状の一覧で **STLファイル（形状を読み込み）...** を選び、ファイルを指定します（単位は mm）。Studio は、形状の最も低い点がテーブルに接するように置き、中央に寄せます。寸法は外接直方体から決まります。オーバーハングの下の材料は、中実として扱われます。ファイルを読み込めない場合は、コンソールにその理由が表示されます。

素材の固定

治具 を選びます：**クランプ・両端**、**クランプ・両側**、**クランプ・四辺**、**バイス**、**Vブロック（横置きの丸棒）**、**両面テープ**、**固定なし（置いただけ）**、**理想固定（スクリプト用）**。

素材と治具は、テーブル（300 × 220 mm）の上に収まり、左奥のツールセッターを避けている必要があります。プログラムの実行中、ツインは切削抵抗と治具の保持力を比較します。余裕が小さいと警告します。素材が滑ると、ツイン内で素材が動くため、そのあとに切削される部分はずれた位置になります。

バイスとパラレルブロック

口金からの突出し は、素材の上面が口金より上に出る量です。変更しない限り 5 mm です。Studio は、それ以上の突出しになるパラレルブロックを選びます。バイスの開き幅は 3.25 から 77 mm までで、口金は素材を少なくとも 2 mm つかむ必要があります。

クランプ

テーブルのねじ穴（M6、25 mm 間隔の格子状）に立てたスタッドを使う、ストラップクランプです。**両端** は素材の長さ方向、**両側** は幅方向、**四辺** はその両方に取り付けます。各クランプには、素材の端から 7〜35 mm 外側に穴が必要です。穴がない場合は、パネルに、穴が見つからないクランプが示されます。横置きの棒材には、Vブロックまたはバイスを使います。

両面テープ

テープは、素材がテーブルに接する面積で素材を保持します。軽い切削や平たい部品に使い、横置きの棒材には使いません。一度はがれると、素材を保持するのは摩擦だけです。**固定なし（置いただけ）** は、重さと摩擦だけで保持します。**理想固定（スクリプト用）** は決して滑らず、スクリプトとテスト用です。

素材上方の空き

CAM が作成するプログラムは、素材の上方の空きを次のように使います：

- プログラムは、Z ストロークの最上部で開始し、終了し、工具を交換します。
- 加工工程の間、工具は安全高さで横に移動します。安全高さは、素材の上面の 5 mm 上、または素材より高いクランプの 2 mm 上です。
- 段取りの結果、空きがそれより少ない場合（長い工具、高いバイス、背の高い素材）、Studio は安全高さを、取れる範囲まで下げます。ストロークの最上部より 0.3 mm 以上下です。そのことは、プログラムとともに通知されます。素材の上面から 0.5 mm 未満になる場合は、プログラムを書き出さず、その理由を表示します。

段取りシートの **Z クリアランス** の表は、各段取りについて、タッチプローブとすべての工具を確認します。Z クリアランスを参照してください。

素材の配置

位置 X と **Y** は、素材の中心を、ツインのテーブル座標で指定します（テーブルの中心は X 100、Y 75）。**角度 °** は、上から見て反時計回りに素材を回します。

- **素材を配置**：そこに新しい素材を置きます。それまでの素材と、その素材に加えた切削は失われます。
- **ここへ移動**：同じ素材を、移動し、回転し、または別の方法で固定し直します。切削した部分は残ります。
- **取り外し**：素材をテーブルから取り除きます。
- **元に戻す**：入力欄を、テーブル上の状態に戻します。

主軸が回転中、機械が移動中、またはプログラムの実行中は、段取りは待機します。

3Dビューでの配置調整

ワーク配置を調整 をクリックするか（テクニシャンレベル以上）、**表示 > 3Dビュー > ワーク配置を調整** を使います。続けて：

- 素材をドラッグして移動し、オレンジのハンドルで回転し、青いハンドルでサイズを変えます（Shift で細かい刻み、Ctrl で粗い刻み）。
- またはキーを使います。矢印キーで 1 mm 移動し（Shift で 0.1 mm、Ctrl で 10 mm）、Q と E で 15° 回転し（Shift で 1°）、Enter で設定し、Esc で元に戻します。

準備 ワークスペースを離れると、配置調整は終了します。

素材の反転

反転 は、素材を治具に取り付けたまま、長さ方向を軸に 180° 裏返します。上面と下面が入れ替わり、前と後ろも入れ替わります。切削した部分も一緒に回ります。ワーク座標オフセットは元のままなので、ワーク原点を設定し直してください（プロービングを参照）。

両面から加工する部品では、CAM が段取りの間の反転を計画します（複数の段取りを持つジョブを参照）。

工具と工具テーブル

ドックの **工具テーブル** には、機械の工具が一覧されます。T 番号、工具、プログラムが読み取る仕様です。主軸にある工具には **主軸に装着** と表示されます。テーブルは、サンプル部品、CAM のジョブ、または AIクライアントによって与えられます。ウィンドウ上では変更できません。

工具テーブルがあると、プログラムは自動で工具を交換します（`M6`）。工具が装着され、機械がツールセッターで測定して、続行します。工具テーブルがない場合、`M6` は、工具を装着するのを待ちます（**工具** のステップの **工具交換**）。そのあと **サイクルスタート** を押します。

主軸の工具を交換する

1. **工具** のステップに進みます。
2. **工具** の一覧で工具を選びます。例：**スクエアエンドミル Ø6、2枚刃（突き出し長さ 15.5）**。
3. **工具交換** をクリックします。

主軸が止まっていて、機械がアイドル状態である必要があります。新しい工具の長さが前の工具と違う場合は、測定する（**工具長補正を設定**）か、ワークの Z 原点を設定し直します。一覧には標準工具があり、メートル系が先に並びます（Studio がインチを表示しているときは、インチ系が先）。工具の記述の仕方（例：`flat 6 n2 L15.5`）は、工具と工具テーブルを参照してください。

ツールセッターで工具を測定する

ツールセッターは、テーブルの左奥にあります。機械が原点復帰済みでアイドル状態であり、主軸が停止している必要があります。

1. ワークの Z 原点を設定したときの工具を主軸に装着した状態で、**基準工具を測定** をクリックします。
2. 工具を交換するたびに、**工具長補正を設定** をクリックします。結果：Studio が新しい工具を測定し、その差から工具長補正（`G43.1`）を設定します。

同じキーは **機械 > ツールセッターとプローブ** にもあります。基準は Studio を閉じるまで保持されます。再起動したら、もう一度測定してください。

比較用の部品

部品の設計データがあれば、ツインは、切削した部品が設計とどこで異なるかを表示します（エンジニアレベル以上）。

1. 先にワーク原点を設定します。部品は、読み込んだときにプログラムの原点に配置されます。

2. **段取り** のステップの **部品との比較** で、**部品STL（比較用）...** をクリックし、部品の STL ファイルを選びます（単位は mm、原点はプログラムの原点）。結果：ワークが、部品からの偏差で色分けされます（**表示 > 色分け > ワーク：部品との偏差**）。±20 µm 以内は良好として扱われ、ほかの色は削り残しと削り過ぎを示します。

比較を終了 は部品を取り除きます。サンプル部品には、それぞれ専用の部品が読み込まれます。

段取りの保存と読み込み

- **段取りを保存...** は、カードの段取り（形状、寸法、材質、治具、位置）を JSON ファイルに書き出します。
- **段取りを開く...** は、保存した段取りをカードに読み戻します。**素材を配置** をクリックして設定します。STL ファイルを素材の形状として開くこともできます。
- **ワークをエクスポート...** は、切削したままのワークを、STL（プログラム座標、素材座標、または機械座標）、または 16ビットの高さマップ（PNG）として保存します。

ウィンドウを閉じるとき、各機械は、段取りと切削したワークを、その機械の状態一式に保存します。

複数の段取りを持つジョブ

両面から加工する部品や、2通りの方法で固定する部品は、複数の段取りを持つ 1つのジョブです。**CAM** ワークスペースで **工程...** をクリックすると、ジョブに段取りが追加されます。すると、ジョブツリーに、段取りごとの階層ができ、その下にその段取りの加工工程とプログラムが並び、2つの段取りの間には **反転** のステップが入ります。段取りが 1つのジョブには、このような階層はありません。

- 各段取りには、独自の加工工程と独自のプログラムがあります。プログラム名は、ジョブ名と段取りの番号から付けられます。例： `plate-1.nc`、`plate-2.nc`。
- 段取りは、その前の段取りが残した部品の状態から始まります。ツインでは、Studio が段取り 1 を実行し、**反転** のステップの指定どおりに部品を裏返し、その部品に対して次の段取りのプログラムを作成します。3Dビューでは、以前に切削して現在は下側になっている部分が破線で描かれます。
- ツリーでは、部品を反転した回数から、段取りを表面または裏面と呼びます。
- 加工工程のパネルの下にある **この工程** には、その段取りの治具、原点、部品の反転方法、プログラムが表示されます。
- **検査結果** には、段取りごとの行と、反転の行があります。**合計** は、機械での時間を合計し、最悪のびびり指標と最大の削り過ぎを示します。また、最後の段取りのあとの完成した部品を、設計上の部品と比較します。

1つのジョブに持てる段取りは最大 8つです。ネスティングしたジョブには、まだ段取りを持たせられません。複数の段取りは Maker エディション以上で使えます（エディションを参照）。

段取りシートには、段取りごとの節があり、その間にすることも記載されます。段取りシートを参照してください。

段取りの間の部品の反転

ジョブツリーで **反転** を選びます。そのパネルで、部品が次の段取りにどのように渡されるかを指定します。設定は、そのステップの後の段取りに属します。

設定	選択肢
反転方法	**U 軸まわりに 180°**：左右の端は元の位置のままで、前と後ろが入れ替わり、上と下も入れ替わります。**V 軸まわりに 180°**：前後の端は元の位置のままで、左と右が入れ替わり、上と下も入れ替わります。**反転なし（持ち替え）**：部品は置かれたままで、治具だけが変わります。
原点	**上面の左手前角** または **上面の中心**：ワーク原点を再び見つける場所（ワーク原点の引き継ぎを参照）
治具	この段取り以降の部品の固定方法：同じ治具、または別の治具

U 軸は機械の X 軸に沿い、V 軸は Y 軸に沿って、前の段取りで置かれている状態の部品を貫きます。

オペレータの作業 には、機械での作業が順に書かれています。前の段取りのプログラムが終わるのを待ち（主軸停止、クーラントオフ）、クランプを外し、部品を裏返し、クランプし、ワーク原点を設定し直して、次のプログラムを読み込みます。**ツイン内** には、ツインが部品に対して行うことが書かれています。

> **注意**
>
> 実際の部品は、ステップの指定どおりに反転してください。次のプログラムがツインで検査されるのは、その向きに反転した部品に対してだけです。別の軸まわりに反転した部品や、まったく反転しなかった部品では、検査されていない場所が切削されます。

ワーク原点の引き継ぎ

部品を反転したり、別の治具で固定したりすると、部品の角は元の位置にありません。ワーク原点（G54）を見つけ直す必要があります。ジョブツリーは、そのことを段取りの横に表示します。段取りのプログラムは、選んだ原点から始まります：

- **上面の左手前角**：部品が現在置かれている状態での、上面の左手前の角。反転または治具の変更のあとは、別の原点を選ばない限りこれが原点になります。
- **上面の中心**：部品の上面の中央。例えば丸い部品に使います。

原点は、タッチプローブで見つけます（プロービングを参照）。段取りシートには、段取りごとのプローブの手順が記載されます。

部品が反転されず、別の方法でも固定されない場合は、前の段取りの原点がそのまま保持され、見つけ直しません。ほかの工具を使うだけの 2つ目の段取りは、この方法で動作します。段取りシートのその節には、ワーク原点を再び測定しないことが書かれます。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

プロービングとワーク原点の検出

主軸にタッチプローブを装着すると、Studio はエッジ、角、穴やボスの中心、または上面を見つけ、そこからワーク原点を設定します。プローブカードは、サイクルを開始する前に、段取りと照らし合わせて確認します。ツインは、機械が行うのと同じようにサイクルを実行します。

プローブカードは、**準備** の **原点** ステップにあり、ジョグキーの下にあります。1.0 では、プローブカードはツインでのみ動作します。実機では、機械自身の制御装置でプロービングしてください。

始める前に

必要なもの：

- テーブルに設定した素材（**段取り** のステップ）
- 主軸のタッチプローブ（工具テーブルのプローブを参照）
- 停止した主軸

機械が原点復帰済みである必要はありません。機械がアラーム状態、または原点復帰していない状態でサイクルを開始すると、先にリセットして原点復帰します。カードにもそのことが表示されます。

> **注意**
>
> MC-1 では、バイスと 30 mm のプローブの組み合わせでは、素材の上方に空きがありません（素材上方の空きを参照）。プロービングするときは、素材を両面テープまたはクランプで保持してください。

工具テーブルのプローブ

1. **準備** で、**工具** のステップに進みます。
2. **工具** で、**3Dプローブ Ø2 ボール（突き出し長さ 30、エッジ検出用）** を選びます。
3. **工具交換** をクリックします。

結果：プローブカードの最初の行 **主軸に** に、プローブの名前が表示されます。主軸にほかの工具が入っている場合は、先にプローブを装着するようカードに表示されます。

工具の記述では、プローブは `probe 2 L30` と書きます。2 mm のボールで、ホルダーからの突き出しが 30 mm です。

素材上方の空き：Z バジェット

Z がストロークの最上部にあるとき、プローブの先端は、段取りの最も高い点（素材の上面、バイスの口金、またはクランプ）を越えている必要があります。Studio は、段取りまたは工具が変わるたびに、主軸の工具と工具テーブルの各工具について、これを計算します。工具が段取りを越えられない場合は、**段取り** のステップに、どれだけ足りないかが表示され、より短い工具か、より低い段取り（バイスの代わりにクランプ）が提案されます。

プローブカードは、サイクルについても同じことを確認します。プローブを、素材と治具の上方の安全高さで横に移動し、その高さをチェックに表示します。

プローブカード

上から順に：

- **主軸に**：プローブと、校正済みかどうか（時刻付きの **校正済み**、または **未校正**）。
- **タッチプローブサイクル**：6つのサイクルと、選択したサイクルの入力欄。
- チェック。
- **3D で経路を表示** と **プロービング開始...**。

サイクルのあとは、カードにその結果が代わりに表示されます（見つけた位置からワーク原点を設定するを参照）。プローブカードの下には **部品を測定** カード、その下に **レポート** カードがあります（検査を参照）。

数値の入力欄は表示単位で入力します。インチでは分数も使えます。読み取れない入力欄は赤くなり、その理由が表示されます（数値を入力するを参照）。

チェック

カードは、サイクルを設定している間、何かが動く前に、サイクルをチェックします。各チェックは、緑（✓）、アンバー（!）、または赤（✕）で表示されます。赤のチェックが 1つでもあると、**プロービング開始...** はグレーのままです。

チェック	緑	アンバーまたは赤
プローブ	**主軸にタッチプローブ、トリガーなし**	主軸にプローブがない、またはプローブの入力がすでにトリガーされている
機械	**主軸停止、原点復帰済み、アイドル**	アンバー：アラーム状態または未原点復帰（サイクルが先に原点復帰します）。赤：主軸が回転中、または機械が動作中
ソフトリミット	**すべての動きがソフトリミット内**	動きがソフトリミットの外に出る
安全高さ	**安全高さ Z ...**、および素材と治具の上方どれだけか	素材の上方に、横に移動する余裕がない
深さ	**深さ：上面から ... 下**、バイスに保持した素材では、口金の上にどれだけ出ているか	ボールが口金より下に入る、または素材の底に近すぎる

ツインが実機に追従しているときは、赤い行で、実機では機械自身の制御装置でプロービングするよう表示されます。

先にツインでサイクルを試す

チェックはドライランから得られます。ツインは、何も動かさずにサイクルを計算します。**3D で経路を表示** を使うと、サイクルの動きが 3Dビューに描かれます。プログラムの経路の代わりに、ステップを離れるまで表示されます。

各接触は、同じ手順で行われます：

1. 開始位置までの保護付き移動（`G38.3`、F600）
2. 高速の接触（`G38.2`、F300）
3. 退避
4. 低速の接触（F20）：結果は、この回の接触で触れた位置です。

プロービングサイクル

サイクルを選び、入力欄を埋めて、**プロービング開始...** をクリックします。各サイクルは、面があると見込む位置の 3 mm 手前から始め、さらに 5 mm 先まで探します。**深さ** は、ボールの中心が素材の上面からどれだけ下で接触するかです（変更しない限り 3 mm）。**開始位置**：**素材から** は、段取りの素材を基準にします。**X Y を指定** は、ワーク座標で指定した点から開始します。

エッジ

素材の 1つの面を見つけます。**どの面**：**左面**（**X−**）、**右面**（**X+**）、**手前**（**Y−**）、**奥**（**Y+**）。**点数**：**1 点**、または **2 点**（**回転も**）。後者は、素材がどれだけ回転しているかも求めます。

結果：**面 X** または **面 Y**。2点の場合は **回転** も。

角

2つの面から角を見つけます。**どの角**：**手前左**、**手前右**、**奥左**、**奥右**。**外側** は素材の角、**内側** はポケットの角です（その位置は **X Y を指定** で与えます）。**角から** は、角からどれだけ離れた位置で各面に接触するかです（5 mm）。**3 点**（**回転も**） は、X の面に 2回接触し、回転も求めます。

結果：**角 X**、**角 Y**。3点の場合は **回転** も。

穴とボス

穴中心 は、穴の中央から外側に向かって穴に接触します。**ボス中心** は、ボスの上面を越えて外側から、ボスに接触します。**点数**：**4 点** または **8 点**。**直径** は呼び寸法です。空欄の場合は、丸い素材の直径が使われます。直方体の素材では、直径と **X Y を指定** が必要です。

結果：**中心 X**、**中心 Y**、**直径**。

上面

上面 Z は、素材の中央、または **X Y を指定** で与えた位置で、上面に 1回接触します。

結果：**面 Z**。

見つけた位置からワーク原点を設定する

結果カードには、各値が機械座標とワーク座標で、また各接触と、プローブの校正がどのように適用されたかが表示されます。素材が回転している場合は、その量も表示されます。**段取り** のステップで、段取りの角度をその値に設定できます。

結果をワーク原点にするには：

1. **原点を設定...** をクリックします。
2. 設定する軸にチェックを入れます。サイクルが測定しなかった軸にはチェックを入れられません。角からは Z が得られません。結果：ダイアログに、送信する行が表示されます。例：`G10 L2 P1 X-129.982 Y95.011`。
3. **G54 の原点に設定** をクリックします（または、使用中だったワーク座標オフセット）。

原点は、見つけた位置に置かれます。面、角、中心、または上面です。**プログラムをコピー** は、サイクルが送信した行をコピーします。**もう一度測定** は、入力欄に戻ります。

> **補足**
>
> ツインの数値は、方法と流れを示すものです。この機械の測定精度ではありません。

リングゲージによるプローブの校正

プローブは、接触した少しあとにトリガーされます（プリトラベル）。その量は方向によって異なります。リングゲージで校正すると、これが測定され、以降のサイクルはその値で補正されます。

1. リングゲージをテーブルに置き、**リングゲージ校正** を選びます。
2. **リング径** に、リングの証明書の値を入力します。
3. **位置**：リングを素材として設定した場合は **素材から**（**管の内径**）、そうでない場合はリングの中心を **指定** で与えます。

4. **校正開始...** をクリックします。結果：**有効球半径**、**XY プリトラベル**、**3 ローブ**、**リング中心**。
5. **この機械に保存** または **保存しない** をクリックします。

校正は、機械と、そのボール径に属し、機械の状態一式のアーカイブに含まれます。工場出荷時の状態に戻すと、校正は消えます。各結果には、使用した校正が示されます。校正するまで、カードには **未校正** と表示されます。このとき、穴はプリトラベル 2つ分大きく、ボスは 2つ分小さく測定されます。

サイクルが止まるとき

サイクルが完了できない場合は、結果カードにその理由が赤で表示され、コンソールにも記録されます。**もう一度測定** で入力欄に戻れます。よくある理由は次のとおりです：

- プロービングが、探索範囲内で何にも接触しなかった：面がサイクルの見込んだ位置にありません。コントローラは ALARM:5 を発生させます。
- 測定点へ向かう途中でプローブが何かに接触した：素材があってはならない位置に何かがあります。
- 接触した点が円にならない（穴、ボス、リング）。

アラームのあとは、機械バーの手順に従います。**リセット**、続いて **ロック解除** です（アラームと対処を参照）。サイクルが止まったとき、プローブはどこにあってもおかしくありません。ジョグする前に 3Dビューを確認してください。

> **注意**
>
> 機械がフィードホールド中にプローブが接触した場合は、**リセット** を押して、もう一度プロービングしてください。**サイクルスタート** を押すと、接触した位置を越えてそのまま進んでしまいます。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

プログラムの作成（CAM）

Studio は、工具経路を自分で計算します。部品を、ポケット、輪郭、穴、彫刻といったフィーチャとして記述します。Studio は、ツインの物理モデルで切削条件を選び、G コード を書き出します。そのうえで、切削する前に、そのプログラムをツインで実行します。

CAM の働き

- 形状はデータです。フィーチャの形状は、長方形、円、多角形、SVG または DXF ファイルのパス、文字、または 3D 加工用のメッシュ（STL、3MF、OBJ）です。
- 工具経路は計算されます。Studio は、形状をオフセットし、領域を削り取り、動きをつなぎます。
- ツインが切削条件を選びます。主軸回転数、送り、深さは、まず材質から決まり、次に、主軸の出力、切削抵抗、びびり、工具のたわみについてのツインのモデルから決まります。これらは推定値です。
- ツインがプログラムを検査します。Studio は、切削する前にプログラムをツインで実行し、実行の結果から過大とわかった部分を変更できます。

すべてお使いのコンピューター上で動作します。プログラムの作成にインターネット接続は不要です。

最も手早い方法は、AIアシスタントに部品を説明することです。例：

```
Make a program for a pocket 40 × 30 mm, 3 mm deep, in 6061-T6, with a 6 mm flat end mill.
```

アシスタントは `program_generate` ツールを使います。作成したもの、所要時間、ツインで判明したことを報告します。

ジョブ：フィーチャ、工具、材質

ジョブは、Studio が部品について保持する情報です：

- 名前。プログラムの名前にもなります（例：`bracket.nc`）。
- 材質（例：6061-T6）。ほかの規格での名称（EN AW-6061 など）も認識されます。
- 主軸の工具と工具テーブル。
- フィーチャ。切削する順に並びます。

Studio が書き出したプログラムは、それを作成したジョブとともに、プログラムフォルダーにバージョンとして保存されます。作り直しても何も失われません。古いバージョンは残ります。

加工工程

加工工程	切削するもの	工具
面削り	素材の上面を平らに	エンドミル
ポケット	ある深さまでの領域：長方形、円、多角形、または任意の形状	エンドミル
輪郭	輪郭に沿って：外側、内側、または輪郭上	エンドミル
溝	線に沿った、幅のある溝	エンドミル
穴あけ	点の位置の穴。深い場合はペッキング	ドリル
ボーリング	工具より大きな穴を、らせん状に回りながら	エンドミル
リーマ加工	あけた穴を寸法に仕上げる	リーマ
ねじ	めねじ	スレッドミル
面取り	輪郭の縁	Vビットまたは面取りカッター
彫刻	文字または線	Vビットまたは小径エンドミル
V彫刻	文字または形状。深さが幅に追従する	Vビット
3D 荒加工、3D 仕上げ	メッシュ	エンドミルまたはボールエンドミル

面削り、ポケット、輪郭

ポケットは、層ごとに削り取られます。加工方式が、各層の経路を決めます：

- リング（既定）：中央から壁に向かって、リング状に削ります。
- ジグザグ：往復するパス。
- アダプティブ：工具の係合量を一定に保つ経路。より硬い材料の深いポケット向けです。

ポケット内のアイランドは、残されます。残り加工では、それより前の大きな工具が残した部分だけを削ります。

輪郭加工は、輪郭に沿って、1層ずつ、ダウンカットで削ります。輪郭で切り抜く部品は、タブで保持できます。

溝

溝は、層ごとに工具の全幅で切削し、そのあと溝の幅まで広げます。工具より幅の広い溝は、トロコイド状に切削できます。線に沿って小さなループを描き、工具への負荷を減らします。

穴：穴あけ、ボーリング、リーマ加工、ねじ

- 穴あけは、コントローラの穴あけサイクルを使います。ドリル径の 3倍より深い穴は、ペッキングで加工します。

- ボーリングは、エンドミルより大きな穴を、らせん状に回りながら切削します。穴径は、工具径の最大 2倍までです。
- リーマ加工は、その穴のはめあいに合ったリーマで、あけた穴を寸法に仕上げます。先に穴をあけておきます。
- スレッドミル加工は、スレッドミルでめねじを、底から上に向かって切削します。メートルねじ（M8）、ユニファイねじ（1/4-20 UNC）、管用平行ねじ（G1/4）、管用テーパねじ（1/4 NPT、Rc1/4）は、名称で認識されます。

面取り、彫刻、V彫刻

面取りは、Vビットで輪郭の縁に沿って、指定した幅まで削ります。彫刻は、線や文字の線画を、一定の深さでたどります。V彫刻は、Vビットで形状を削ります。深さが形状の幅に追従するので、角がシャープに仕上がります。

3D 荒加工と仕上げ

メッシュ（STL、3MF、OBJ）は、レベルごとに荒加工してから、仕上げます。仕上げ方式は、走査線、等高線、急斜面と緩斜面、ペンシルです。

工具と工具テーブル

工具は、種類、直径、およびそのほかに必要なものを使って記述します。例：

仕様	工具
`flat 6 n3 L20`	スクエアエンドミル、6 mm、3枚刃、突き出し長さ 20 mm
`ball 3 L12`	ボールエンドミル、3 mm
`drill 3.4 118 L20`	ドリル、3.4 mm、先端角 118°
`v 6 90 L18`	Vビット、6 mm、90°
`reamer 6 H7 n6 L32`	マシンリーマ、6 mm、H7 の穴用
`thread 5 60`	スレッドミル、5 mm、60° プロファイル
`flat 2 n2 s4 l6 L10`	スクエアエンドミル、2 mm、先端から 6 mm 上で始まる 4 mm のシャンク付き
`thread 2 60 d1.4 n1 s4 l6 L9`	スレッドミル、ヘッド 2 mm、首径 1.4 mm（先端から 6 mm まで）、その上に 4 mm のシャンク

すべての工具に、突き出し長さ（`L`）を指定してください。そうすると、Studio は、ホルダーが素材に当たらないことを確認し、段取りシートの Z クリアランスで、その工具を判定できます。

段付きシャンクの小径エンドミルと、アンダーカット用のカッター（スレッドミル、Tスロットカッター、アリ溝カッター、ロリポップカッター）には、シャンク径（`s`）と、先端からシャンクが始まる高さ（`l`）も指定してください。シャンクが切削した壁にこすれる箇所では、Studio は `l` より深く切りません。ツインは、シャンクとタッチプローブ本体がワークや治具に当たらないかを確認します。

工具テーブルは、工具を番号ごとに一覧にします。フィーチャには、使う工具を指定し、Studio が工具交換（`T2 M6`）を追加します。

ツインからの切削条件

Studio は、その工具に対する材質の切削条件から始めます。そのうえで、ツインのモデルを使います：

- 主軸の出力：切削が使う出力は、最大でもその約 80 % です。
- びびり：深さと主軸回転数は、ツインの安定性モデルが安定した切削を見込む範囲にとどまります。
- 治具：切削抵抗は、治具の保持力を大きく下回る範囲にとどまります。
- 工具のたわみ：仕上げパスは軽く保たれます。

検証をオンにすると、Studio はプログラムをツインで実行します。実行でびびりや過大な負荷が見つかると、Studio は、層を浅くするなどしてプログラムを変更し、もう一度試します。試した各バージョンは保存されます。応答には、何をなぜ変更したかが記載されます。

これらは物理モデルによる推定値です。実機での最初の切削は、よく見て、音を聞いてください。

はめあいと公差

寸法は、はめあいとともに書きます。例えば、穴には `"d": "Ø12 H7"`、軸には `"fit": "g6"` と書きます。Studio は、寸法を公差域の中央に合わせて切削し、壁面を専用の 1パスで仕上げます。

よく使うはめあいは、名前で指定できます：`sliding`（H7/g6）、`locating`（H7/h6）、`transition`（H7/k6）、`press`（H7/p6）。

小さくて公差の厳しい穴は、フライス加工よりリーマ加工のほうが適しています。Studio はそのことを知らせます。その場合は、穴をあけてから、その公差等級のリーマで仕上げることができます。

ツインはカッターの振れを推定し、Studio は、はめあいのある寸法でその振れを考慮します。

複数の段取り

両面から加工する部品には、2つの段取りが必要です。各段取りには、それぞれのフィーチャがあり、部品がその段取りにどう渡されるかも指定します。X または Y まわりの反転、別の固定方法への持ち替え、ワーク原点の見つけ直しです。Studio は、段取りごとに 1つのプログラムを書き出し、それぞれを、前の段取りが残した状態に対してツインで検査します。段取りの追加方法と部品の反転方法については、複数の段取りを持つジョブを参照してください。

板にネスティングした部品

1枚の板から、複数の部品を切り出せます。Studio は、部品を近づけて配置し、余白を確保し、最後まで部品が板に残るようにタブを追加します。

インチ寸法と分数

インチ単位では、寸法を分数で書けます：`1/4`、`1 1/2`、`3/16"`。インチ系の工具は、工具ライブラリにあります（例：1/4" のスクエアエンドミル）。言語と単位を参照してください。

ツインでプログラムを検査する

切削する前に、プログラムをツインで実行してください。ツインは自身のワークを切削し、衝突、削り過ぎ、びびり、主軸の負荷、時間を報告します。シミュレーションを参照してください。

> **注意**
>
> ツインは、プログラムを、与えられた段取りに対して検査します。ツインを、実機と同じ状態に設定してください。同じ素材、治具、工具、ワーク原点です。

段取りシート

段取りシートは、オペレータに、ジョブの段取りの仕方を伝えます。素材、治具、工具、ワーク原点、Z クリアランスです。段取りシートを参照してください。

プログラムとエディタ

プログラム ワークスペース（Alt+3）には、プログラムとサンプル部品、入力しながらプログラムを検査するエディタ、そしてプログラムがワークに対して何をするかを示すプレビューがあります。Studio は、プログラムを保存するたび、また機械に読み込むたびに、そのバージョンを保存します。そのため、何が変わったかを確認でき、元に戻すこともできます。

プログラムを開く

ワークスペースは 3つの部分で構成されています。

- 左：**プログラム一覧**（各プログラムとそのバージョン）と **サンプル部品**
- 中央：エディタ
- 右：3D プレビュー、**読み込んで検査**、プログラム検査の検出事項

プログラムを開くには、次のようにします。

- **ファイル > プログラムを開く...**（Ctrl+O）は、G コード ファイルを機械に読み込みます。エディタに未保存の変更がない場合は、エディタにも読み込みます。Studio はファイルを読み取るだけで、書き込むことはありません。
- **プログラム一覧** のプログラムまたはバージョンをダブルクリックすると、そのテキストがエディタに入ります。**読み込んで検査** を押すまでは、機械には読み込まれません。

エディタと機械は、別々のプログラムを保持できます。機械にあるプログラムをエディタで開いていない場合は、ステータス行にその旨が表示されます。Ctrl+K はエディタのプログラムを機械に読み込み、Ctrl+Shift+R は機械のプログラムをエディタに取り込みます（編集内容は先にバージョンとして保存されます）。

サンプルプログラム

Studio 1.0 には 4種類のサンプル部品が付属しています。NEMA 17 モーター取付板、計測器のフロントパネル、マイクロ流体液滴チップ、アイソグリッド軽量パネルです。今後のアップデートで追加されます。**サンプル部品** のいずれかをダブルクリックするか、**ファイル > サンプル部品...** から選びます。

サンプルを読み込むと、そのサンプル用に機械全体が段取りされます。素材と治具、工具テーブルと主軸の工具、原点復帰、ワーク原点、プログラム、そしてサンプルに比較用の部品がある場合はその部品です。

> **補足**
>
> サンプルを読み込むと、それまでの素材、工具テーブル、G54 のワーク原点は置き換えられます。同じ一覧にある **デモプログラム** はプログラムだけを読み込み、素材はそのままにします。

エディタ

エディタは G コード エディタです。

- ワードは種類ごとに色分けされます：G コードと M コード、軸、円弧のワード、送り、回転数、工具、コメント、式
- マージンには行番号とマークが表示されます：検査の検出事項（赤はエラー、黄は警告、青は注記）、ブレークポイント、機械が実行中の行
- 実行後は、2つ目のマージンに各行の時間が秒単位で表示されます。送りに達しなかった行は黄、何かが起きた行（材料への早送り、主軸停止中の切削、衝突、アラーム）は赤です。クリックすると、前回の実行でのその行の数値が表示されます。

エディタの下のステータス行には、行と列、テキストが機械に入っているかどうか、バージョン、検査の件数、そして今いる行の後のプログラムの状態（モーダルコード、送り、回転数、工具、位置）が表示されます。

拡大 を押すと、エディタがワークスペース全体を使います。**元のサイズ** を押すと、プログラム一覧とプレビューが戻ります。

色と行番号

色はウィンドウの外観（ライトまたはダーク）に従います。コントローラがすでに取り込んでいる、実行中の行より先の行は、灰色で表示されます。

検索と置換

- Ctrl+F で検索ボックスを開きます。Enter で次の一致箇所へ、Shift+Enter で前の一致箇所へ移動します。検索では大文字と小文字を区別せず、末尾まで進むと先頭に戻ります。
- Ctrl+G で、行番号を指定して移動します。
- Ctrl+H で、検索ボックスの下に置換ボックスを開きます。Enter で選択中の一致箇所を置換して次へ進み、Ctrl+Enter ですべて置換します。Ctrl+Z で元に戻せます。

Esc でボックスを閉じます。

ブレークポイントと一時停止

ブレークポイントは、実行がその行に達したときにツインの時間を止めます。機械の動作は変わりません。F5 で続行します。

- Ctrl+B、またはマークのマージンのクリックで、その行にブレークポイントを置きます。
- **条件で一時停止**（Ctrl+Shift+B）は、次のようなことが起きたときにツインを止めます：工具が切削を始めたとき、早送りが材料を削ったとき、ニアミスまたは衝突、びびり、アラーム、コントローラの状態の変化、切削抵抗・主軸負荷・追従誤差が指定した上限を超えたとき、モーダルの変更。ツインがリアルタイムで動いているときは、びびりの検査はシミュレーションと並行して計算されます。そのため、びびりによる一時停止や、ビューに表示されるびびりの値は、切削より少し遅れることがあります。通常は数百ミリ秒ですが、新しい切削条件が多く重なる箇所（アイランドを囲むポケットなど）では 1〜2 秒遅れることがあります。工具についての検査結果が出るまで、びびり指標は **計算中...** と表示されます。グラフと工具先端の軌跡は各切削の結果を待つので、その切削のびびり指標を表示します。実行終了時のレポートはどちらの場合も同じです。
- 行のメニュー（マージンを右クリック、または Shift+F10）には、**... 行目まで実行** と **すべてのブレークポイントと条件を解除** もあります。

ブレークポイントは、プログラムが機械に読み込まれてから有効になります。

プログラム検査

エディタは、入力中（入力を止めてから少し後）、読み込み時、そして実行後にもう一度、プログラムを検査します。検査は 2つの部分からなります。

- ルール検査は、コントローラと機械が受け付ける内容に照らして、すべての行を調べます。ほかに必要なものはありません。
- プレビュー検査は、現在のワークに対してプログラムを実行して調べます。機械の電源が入って原点復帰済みであること、プログラムが実行中でないことが必要です。入力中は 20,000 行以下のプログラムを検査します。それより長いプログラムは機械に読み込んだ時点で全体を検査し、問題は 3Dビューと **メッセージ** タブに赤い枠で表示されます。

プログラム > プログラムを検査 は、別の検査です。読み込み済みのプログラムを、機械を動かさずに、コントローラ自身のチェックモードで 1行ずつ通します。

検査で検出されるもの

検出事項はエラー、警告、注記のいずれかで、該当する行の下にルール番号とともに表示されます。例：

ルール	種別	検出内容
S01	警告	最初の移動の前に開始ブロックがない：プログラムがコントローラの状態に頼っている
S02	警告	末尾に `M2` も `M30` もない
S04	エラー	`F` より前に切削移動がある
S05	警告	`S` なしで主軸が始動する、または `S` が主軸の範囲外
S07	エラー	コントローラが受け付けないコード：式、`O` ワード、サブプログラム、`G41/G42`、`G68`
S10	警告	ワークの上面より下で、プログラムがここまでに切削した深さより深い位置への早送り
S11	エラー	コントローラが拒否する円弧
S12	警告	水冷主軸がクーラント（`M8`）なしで切削している
S14	警告	ワーク原点を設定したときとは別の長さの工具で、その長さを測るものがない

プレビュー検査は、プログラムがワークに対して何をするかも検出します：材料への早送り、主軸停止中の切削、治具との衝突、ストロークを超える移動、クーラントなしでの主軸運転。

検出事項の修正

検出事項には、提案が付くものがあります。例：**提案（前に挿入）：M8**。

1. カーソルをその行に置くか、Alt+↓ と Alt+↑ で検出事項から検出事項へ移動します。
2. Ctrl+. を押します（または、行のメニューで **提案を適用** を選びます）。

結果：変更は 1ステップで行われ、Ctrl+Z で元に戻せます。適用する前に提案を読んでください。提案が直すのは検出事項であり、意図ではありません。

最適化

最適化（Ctrl+Shift+O）は、同じ部品を切削するより速いプログラムを探します。送りと回転数を変更し、短い行をまとめ、退避の高さを下げます。元のプログラムと最適化後のプログラムを、同じ段取りで、機械の 2つの非表示コピー上で実行します。素材が段取りされている必要があります。**最適化** は Pro エディション以上で使えます。

結果が良ければ、エディタのテキストが置き換えられ（Ctrl+Z で元のプログラムに戻せます）、比較が開きます。コンソールには、最適化の前後の時間、最大切削抵抗、びびり指標と、変更内容が表示されます。止めるには、**最適化を停止** または Ctrl+Shift+O を押します。

バージョン

次の操作をすると、Studio はプログラムのバージョンを保存します。

- **バージョン保存**（Ctrl+S）を押す
- 機械に読み込む（**読み込んで検査**、または **ファイル > プログラムを開く...**）
- 最適化する、または、生成されたプログラムや機械のプログラムを編集内容の代わりに取り込む（編集内容は先に保存されます）

直前のバージョンと同じ内容のバージョンは、重ねて保存されません。バージョンは `ドキュメント\Twinwright Studio\programs\<program>\` に 1ファイルずつ保存され、削除されることはありません。**プログラム一覧** には、各プログラムの下にバージョンが、保存した時刻と保存した理由とともに表示されます。

2つのバージョンの比較

比較（Ctrl+D）は、エディタのテキストと異なる最新の保存済みバージョンを、エディタの下に表示します。変更された行は黄、追加された行は緑、削除された行は赤、`F` または `S` だけが変わった行は青で表示されます。Ctrl+D をもう一度押すと、さらに古いバージョンが表示されます。Alt+↓ と Alt+↑ で差分から差分へ移動し、Esc で閉じます。

比較の見出しには差分の数が表示され、両方のバージョンがこのセッションで最後まで実行されている場合は、実行時間も比較されます。

保存済みの 2つのバージョンを比較するには、まず新しいほうをエディタで開き、Ctrl+D を押します。

バージョンに戻す

1. **プログラム一覧** で、そのバージョンをダブルクリックします。結果：そのテキストがエディタに入ります。未保存の編集内容は、先にバージョンとして保存されます。Ctrl+Z でこの変更を元に戻せます。

2. **読み込んで検査**（Ctrl+K）を押して、機械に読み込みます。

保存とエクスポート

- **バージョン保存**（Ctrl+S）は、Studio のプログラムフォルダーにバージョンを保存します。
- Ctrl+Shift+S は、エディタのプログラムを、選んだ場所にファイルとして保存します。
- バージョンは通常の `.nc` ファイルです。プログラムフォルダーからコピーできます。

ファイル > グラフをエクスポート (CSV)... は、実行のグラフを保存します（分析と診断を参照）。段取りシートと検査レポートは、**準備** の **レポート** カードにあります（検査を参照）。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

プログラムの実行と再開

実行 ワークスペースは、読み込まれたプログラムを実行し、加工中の機械の動きを表示します。実行が止まると、Studio は次の手順を案内し、プログラムをあるブロックから再び開始することもできます。

読み込みと開始

プログラムを機械に読み込むには、次のいずれかを行います。

- **ファイル > プログラムを開く...**（Ctrl+O）：G コード ファイルを選びます。Studio はファイルを読み取るだけで、書き込むことはありません。
- **ファイル > サンプル部品...**：サンプルを選びます。Studio は、そのサンプル用の素材、工具、ワーク原点も段取りします。
- **プログラム** ワークスペースで **読み込んで検査**（Ctrl+K）を押すと、エディタのプログラムが読み込まれます。

機械バーには、プログラム名、ブロック数、おおよその所要時間が表示されます。

開始するには、機械バーの **サイクルスタート**（F5）を押します。**サイクルスタート** は、プログラムが読み込まれ、機械がアイドル状態で原点復帰済みのときに、塗りつぶし表示になります。プログラム検査で問題が見つかっていた場合は、プログラムの開始時にコンソールがそのことを知らせます。

実機でプログラムを実行する前に、検査してください。**プログラム > プログラムを検査** は、プログラムをコントローラのチェックモードに通します。コントローラはすべてのブロックを読み取り、機械は動きません。

実行を見守る

プログラムの実行中は、次のように表示されます。

- 3Dビューに、工具、工具先端の軌跡、切削されていくワークが表示されます。**工具の拡大表示** では、切削中の工具を拡大して見られます。
- ドックの **プログラム** タブにリストが表示されます。実行中のブロックにはマークが付き、完了したブロックは薄く表示され、バーで各ブロックの時間がわかります。**実行中のブロックに追従** を使うと、そのブロックが常に表示範囲に入ります。
- リストの横にある **プログラム構造** は、プログラムを（プログラム中のコメントに基づいて）セクションとレイヤーに分け、それぞれの時間を表示します。
- 右の列には、表示値が並びます。

プログラムが終わると、**メッセージ** タブに、実行されたブロック数と、シミュレーション内で実行にかかった時間が表示されます。

進捗と残り時間

機械バーには、全ブロック中の実行中のブロック、完了したブロックの割合、**経過** と **残り** が表示されます。

- **経過** は、これまでの機械時間で、シミュレーションの時間で表されます。
- **残り** は、残りの部分の推定値で、プログラムの送りと早送りから求めます。

アラームの後は、バーにプログラムが停止したブロックが表示されます。

表示値

実行 の右の列には、上から順に次の項目があります。

- **位置**：各軸のワーク座標と機械座標。軸が原点復帰済みであれば、印が付きます。その下にモーダル行があり、有効な G コード、工具、送り、主軸回転数、M コードが表示されます。アラーム中は、主軸とクーラントが機械の実際の状態で表示され、プログラムの値はツールチップに表示されます。
- **主軸と送り**：主軸回転数（円弧が実際の回転数、マークが指令された回転数）、主軸負荷（80 % までは緑、100 % を超えると赤）、送り。
- **切削**（エンジニアレベル以上）：ツインによる推定値。切削抵抗、パワー、びびり指標、1刃当たりの送り。
- **オーバーライド**、**操作盤**、**手動操作**、**工具と原点**（オペレータレベル以上）。

セクションのタイトルをクリックすると、折りたたまれます。工場モードでは、**位置** に **残移動量** も表示されます。これは、各軸について、現在のブロックの残りの距離です。

フィードホールド、停止、オーバーライド

- **フィードホールド**（F6 または Space）は、軸を減速して停止させます。主軸は回り続けます。**サイクルスタート** で続行します。
- **停止**（F7）は、まず送りを保持し、機械が止まってからコントローラをリセットします。もう一度押すと、すぐにリセットします。
- **リセット**（F12）は、プログラムをすぐに停止します。

オーバーライドは、実行中のプログラムの送りと主軸回転数を、10 % から 200 % の範囲で変更します。

オーバーライド	キー	画面上の操作
送り	Ctrl+↑、Ctrl+↓：±10 %	**オーバーライド**：ノブをドラッグ。Ctrl を押しながらで 1 % 刻み
主軸	Ctrl+→、Ctrl+←：±10 %	**オーバーライド**：ノブをドラッグ
早送り	**機械 > オーバーライド**：25 %、50 %、100 %	**オーバーライド**：100、50、25

Ctrl+0、またはノブのダブルクリックで、送りと主軸が 100 % に戻ります。

シングルブロック

機械バーの **シングルブロック** は、1ブロックごとにプログラムを停止します。**サイクルスタート**、または **プログラム > 次のブロック** を押すと、次のブロックを実行します。開始前でも実行中でもオンにできます。もう一度クリックすると、停止せずに続行します。

ジョグとハンドル

ジョグは、**実行** の **手動操作**（最初は折りたたまれています）で行うか、キーで行います（機械のキーを参照）。

- **移動量**：1回押すごとに 0.01、0.1、1、10、または 50 mm。または **押している間ジョグ**：キーを押している間、軸が動きます。
- **速度**：100、300、1500、または 3000 mm/min。
- パッド：X と Y を十字に、Z をその横に配置し、中央に **ジョグ停止**（Esc）があります。
- **ハンドル**：**X**、**Y**、**Z** のいずれかと刻み（**×1**、**×10**、**×100**：1目盛りあたり 0.001、0.01、0.1 mm）を選び、ダイヤルをドラッグするか、その上でマウスホイールを回します。

ジョグできるのは、機械の電源が入っていて、アイドル状態で、アラームが出ておらず、プログラムが実行されていないときだけです。押し続けているジョグは、キーを離したとき、またはウィンドウがフォーカスを失ったときに止まります。

ツール > 設定... > パネルとデバイス > 手動パルス発生器（ハンドル） で、ハンドルの動作を設定します。**速度** は、ホイールが止まると軸も止まります。**経路** は、すべての目盛りを移動します。

アラームと対処

アラームが発生すると、機械バーの状態表示が赤になり、バーには次の内容が表示されます。

- アラームとその意味。例：**ALARM:17 モータードライバが異常を報告しています**
- **次の手順：** 今すべきことと、それを実行するキー
- **全手順**（手順が複数ある場合）

そのキーをクリックするか、**機械 > 次の手順を実行** を使うと、次の手順を実行します。各手順の後、Studio は機械の状態から手順を計算し直します。

アラームのその他の表示：位置が信頼できなくなったときに **位置** の下に表示される行、リストでプログラムが停止した箇所の赤い行、3Dビューのドライバに付くラベル。

アラームの種類、原因、解除方法：アラーム。

作業に戻る手順

全手順 を押すと、作業に戻る手順が開きます。原因、続いて順番に並んだ手順があり、各手順には、そのキーと、その手順が必要な理由が付いています。今実行する手順にはマークが付きます。1回のクリックで実行されるのは、次の手順だけです。

たとえば、ドライバのアラーム（ALARM:17）の後は、次のようになります。

1. **非常停止を押してドライバの電源を切る**（F1）：ドライバのアラームは、ドライバの電源が切れたときにしか解除されません。
2. **非常停止を解除**（F1）。
3. **電源投入**（F2）：ドライバに再び電源が入ります。
4. **リセット**（F12）：ドライバが復帰した後も、コントローラは ALARM:17 を保持しています。
5. **原点復帰**（F3）：電源が切れていたため、位置は信頼できません。

故障が **診断** で注入されたものである場合は、原因にそのことが時刻とともに表示されます。

アラームでプログラムが停止した場合は、手順の最後に、そこから再開するブロックが示されます。たとえば、停止したレイヤーの先頭です。

ブロックからの再開

Studio は、プログラムを任意のブロックから開始できます。そのブロックの時点でプログラムが機械に設定しているはずの状態に機械を整える開始ブロックを書き出し、続いて残りを実行します。

1. プログラムを停止します。機械の電源が入っていて、プログラムが実行中でない必要があります。
2. ドックの **プログラム** タブで、そのブロックを右クリックして **ブロック ... から開始** を選び、**OK** を押します。エディタでは、行のマージンを右クリックするか、Shift+F10 を押します。結果：Studio が「*name*（ブロック *n* から）」を読み込みます。開始はしません。
3. **サイクルスタート** を押します。

開始ブロックは、次の動作を行います。

1. Z を上端まで上げる
2. そのブロックの時点でプログラムが設定していた単位、平面、距離モード、ワーク座標系を設定する
3. そのブロックより前にプログラムが工具を交換していた場合は、工具を交換する
4. クーラントと主軸をプログラムの回転数でオンにし、主軸が回転数に達するまで待つ
5. 安全な高さで開始点の上へ移動し、その 2 mm 上まで下降してから、送りで下降する
6. そのブロックからプログラムを続ける

その後、プレビュー検査が、残りの部分を現在のワークに対して検査します。

> **注意**
>
> 開始位置より前のブロックには、Studio が追跡できないものがあります：パラメータ、式、サブプログラム、`G10`、`G28`、`G30`、`G92`。読み込み時の注記にそれらが一覧表示されます。**サイクルスタート** を押す前に、機械の状態を確認してください。

先にコピーで開始部分を試す

アラームでプログラムが停止した後は、手順の最後に **開始部分をプレビュー...** が表示されます。これを押すと、開始ブロックとその注記が開きます。

1. **コピーで試す** をクリックします。結果：開始部分が、現在のワークを使って機械の非表示コピー上で実行され、Studio が、所要時間、衝突（あれば）、最大のびびり指標、除去量を報告します。
2. 問題なさそうであれば、作業に戻る手順を実行してから、そのブロックから開始します。

シミュレーション速度での実行

ツインは実時間で動作します。**シミュレーション > 速度** を使うと、実時間の 0.5倍から 50倍まで変更できます。機械バーの時計には、シミュレーションの時間、選んだ倍速、コンピューターが実際に達している倍速が表示されます。

シミュレーション > シミュレーションを一時停止、または機械バーの砂時計で、ツインの時間が止まります。止まっている間は、何も変化しません。**サイクルスタート** で続行します。エディタのブレークポイントと一時停止条件も、同じようにツインの時間を止めます（プログラムとエディタを参照）。

シミュレーションの一時停止は、フィードホールドではありません。機械の状態は凍結されるだけで、停止するわけではありません。

分析と診断

分析 は、実行を振り返ります。時間がどこで使われたか、切削抵抗とびびりに関するツインの推定値、そして加工した部品と設計との比較がわかります。**診断** は、機械の電源系統と信号を表示し、ツインに故障を注入して、その原因を見つける練習ができるようにします。

実行を振り返る

分析（Alt+5、エンジニアレベル以上）は、プログラムが実行されると内容が表示されます。

- 左の **実行履歴** には、このセッションの実行が一覧表示されます。プログラム、終了の仕方、機械時間、開始時刻です。アラームまたはエラーで終了した実行は赤で表示されます。
- 中央にグラフがあり、その下に **ブロック別プロファイル** と **安定限界線図** の 2つのタブがあります。
- 右の **前回の実行** には、前回の実行の要約が表示されます。どこまで進んだか、最も時間のかかったブロック、最大切削抵抗と最大のびびり指標、およびそれぞれのブロック、除去量、送りに達しなかったブロック、イベントです。

グラフ、プロファイル、要約は、前回の実行のものです。

ブロック別プロファイル には、時間のかかったブロックが、長い順に並びます。時間とその割合、指令された送りと実際に達した送り、達しなかった理由、切削抵抗、びびり指標、除去量、イベントが表示されます。ブロックをダブルクリックするか、Enter を押すと、**プログラム** エディタのそのブロックへ移動します。

ブロックが送りに達しなかった理由は、次のとおりです。

- 軸の最高速度
- コーナー同士が近すぎて、その間で加速しきれない
- 円弧の弦による制限
- コントローラのプランニング、送信ソフトの遅れ、またはコントローラの読み取りの遅れ
- 100 % 未満の送りオーバーライド、またはフィードホールド

時間：推定値と実際の値

実行前に、Studio はプログラムの送りと早送りから機械時間を推定します。機械バーにはその時間が **約** を付けて表示され、ドックの **プログラム構造** にはセクションごとの時間が表示されます。実行中と実行後の時間は、加速度とコントローラのプランニングを含む、ツインの機械時間です。機械バーには経過時間と残り時間が、**プログラム構造** には実行全体の時間が、**ブロック別プロファイル** には各ブロックの時間が表示されます。

すべての時間は、シミュレーション内の機械時間であり、実際の時計の時間ではありません。

切削抵抗、びびり、切りくず厚さ

グラフには、直近 10、30、または 120 s の実行中の値が表示されます。送り、実速度、主軸の回転数と負荷、切削抵抗、トルク、パワー、切りくず厚さ、びびり指標、追従誤差です。グラフ名をクリックすると、表示と非表示が切り替わります。

びびり指標は、切込み深さを、使用中の主軸回転数で安定な深さで割った値です。

指標	意味
0.8 未満	安定
0.8–1	限界付近
1 を超える	びびり発生

安定限界線図 は、直前の切削について（まだ切削がない場合は、X 方向の深さ 1 mm のフルスロットについて）、主軸回転数に対する安定な切込み深さを描きます。**安定限界線図を計算** をクリックします。赤い点は実際の切削条件を、緑の破線は最も安定な回転数を示します。

表示 > 色分け で、工具先端の軌跡を、送り速度、切削抵抗、切りくず厚さ、びびり指標、プログラムのブロックで色分けできます。ワークは、材料と深さ、部品との偏差、プログラムのブロックで色分けできます。

これらは、物理モデルに基づくツインの推定値です（ツインのモデルを参照）。

設計形状との比較

加工した部品が設計とどこで異なるかを確認するには、次のようにします。

1. **準備** の **段取り** ステップで、**部品STL（比較用）…** をクリックし、部品の STL ファイルを選びます（単位は mm、原点はプログラムの原点）。
2. プログラムを実行します。
3. **表示 > 色分け > ワーク：部品との偏差** を選びます。

結果：ワークは、各点が部品からどれだけ離れているかによって色分けされます。±20 µm 以内は良好とみなされ、それ以外の色は削り残しと削り過ぎを示します。**比較を終了** で、部品が取り除かれます。サンプル部品を読み込むと、それぞれの部品が読み込まれます。

2回の実行の比較

分析 には前回の実行が表示されます。プログラムの 2回の実行を比較するには、次の方法があります。

- **プログラム** エディタで **比較**（Ctrl+D）を使うと、両方のバージョンがこのセッションで最後まで実行されている場合に、両方のバージョンの実行時間と、変更された行の時間がわかります（2つのバージョンの比較を参照）。
- AIアシスタントは、2回の実行を並べて描画できます（AIアシスタントを参照）。

エクスポート

グラフをエクスポート (CSV)...（**前回の実行** 内、または **ファイル > グラフをエクスポート (CSV)...**）は、グラフの直近約 160 s を CSV ファイルとして保存します。時間の列（単位は秒）に続いて、グラフごとに 1列があり、見出しに単位が付き、値は表示単位です。

準備 の **ワークをエクスポート...** は、加工したワークを STL ファイル（プログラム座標、素材座標、または機械座標）または高さマップとして保存します。

診断ワークスペース

診断（Alt+6、テクニシャンレベル以上）は、機械の電源系統を図で表示します。電力は左から右へ流れます。実線は電力を運び、通電中は緑です。破線は信号です。各部品は、正常なときは緑、オフのときは灰色、警告のときは黄、アラームまたは故障のときは赤で表示されます。

部品をクリックすると、右の列に詳細が表示されます。図の下には、コントローラの入力、故障注入、ドライバアラームがあります。

電源系統

MC-1 の電源系統は、商用電源から主軸まで、次のとおりです。

- **主電源スイッチ QF1** と、コントローラ用の **24 V 電源**
- **コントローラ**：その状態またはアラーム
- **非常停止 SB0** と、それが開放する接触器 **KM1**
- ドライバ用の **48 V 電源**
- **X**、**Y**、**Z** のドライバと各軸
- **インバータ**、**主軸**、**クーラントポンプ**

コントローラの入力

各入力にランプがあります。リミットスイッチ（**X− リミット**、**X+ リミット**、**Y− リミット**、**Y+ リミット**、**Z リミット**）、**ツールセッター**、**非常停止**、**ドライバアラーム** です。ランプは、入力がオフのときは灰色、オンのときは黄または赤になります。

ツールセッター のランプは、コントローラのプローブ入力です。タッチプローブが接触したときにも点灯します。

ドライバ

各ドライバには、**準備完了** と追従誤差、または **無効**、**アラーム**、**起動中**、**電源オフ** が表示されます。詳細には、ドライバの電源と、各軸の追従誤差が、最大値とともに表示されます。

主軸インバータ（VFD）

インバータ には、出力周波数と電流、または異常が表示されます。異常は **OC**（過電流）、**OL**（過負荷）、**LU**（不足電圧）です。詳細には、主軸回転数と指令された回転数、負荷、ポンプが表示されます。クーラントなしで回転した主軸は、その時間とともに、警告に一覧表示されます。

ツインへの故障の注入

故障注入を使うと、ツイン上で故障を見つけて解消する練習ができます。変更されるのはウィンドウ内の機械であり、実機ではありません。

- **配線**：機械の任意の配線（リミットスイッチ、ツールセッター、ドライバのアラームチェーン、非常停止線、KM1 の回路、RS485 バス、インバータの故障リレー）を、**断線**、**短絡**、**間欠断線**、**瞬断**（50 µs の途切れが 1回）のいずれかにします。**正常** を選ぶと、配線が元に戻ります。
- **ドライバアラーム**：**X**、**Y**、**Z** のいずれかを選ぶと、そのドライバが異常を報告します。ドライバのアラームは、ドライバの電源が切れたときにしか解除されません。機械バーの、作業に戻る手順に従ってください。

注入した故障は、警告の下に一覧表示され、コンソールに書き込まれ、監査ログに記録されます。ドライバのアラームの後は、機械バーの原因に、注入した故障とその時刻が表示されます。

アラーム：各アラームと復旧手順

すべてのアラーム、その意味と原因：アラーム。機械バーには、発生しているアラームに応じた、作業に戻る手順が表示されます（作業に戻る手順を参照）。

アラーム	作業に戻る手順
ドライバの異常（ドライバがアラーム状態の ALARM:17）	非常停止を押して解除し、**電源投入**、**リセット**、**原点復帰**
非常停止が押された（ALARM:10）	非常停止を解除し、**電源投入**、**リセット**、**原点復帰**
ドライバが復帰した後の ALARM:17	**リセット**、**原点復帰**
原点復帰が必要（ALARM:11）	**原点復帰**
ハードリミット（ALARM:1）	**リセット**、**ロック解除**、ジョグでリミットスイッチから脱出、**原点復帰**
ALARM:3、8、9	**リセット**、**原点復帰**
その他のアラーム	**リセット**、**ロック解除**

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

ツインでのプログラム実行

草案（第14ラウンド）：本文は執筆中で、節が増える可能性があります。

ツインとは

ウィンドウ内の機械は、MC-1 のデジタルツインです。その中では Twinwright 独自カーネルが動作しており、機械上と同じコードです。モーター、ドライバ、主軸、スイッチ、切削は、物理モデルが担います。プログラムはツイン内で機械上と同じように実行され、ツインは自身のワークを切削します。ツインが物理世界について出す数値は、すべて推定値です。

実行の前に

ツインは、与えられた段取りに対してプログラムを検査します。そのため、実機と同じになるように段取りしてください。

- 機械（**ファイル > 機械を切り替え...**）
- 素材とその固定方法（**準備** ワークスペース）
- 工具（ドックの **工具テーブル**）
- ワーク原点（**機械 > ワーク原点**）

ファイル > サンプル部品... は、これらすべてを設定し、プログラムも一度に読み込みます。

> **注意**
>
> 検査の価値は、段取りの正しさで決まります。実機の素材、治具、工具、ワーク原点が違えば、それはツインが検査したものではありません。

プログラムを実行する

1. プログラムを開きます：**ファイル > プログラムを開く...**
2. 最初に検査します：**プログラム > プログラムを検査** は、プログラムをコントローラのチェックモードに通します。コントローラはすべてのブロックを読み取り、機械は動きません。
3. 実行します：**サイクルスタート**（F5）。**フィードホールド**（F6）で一時停止し、**サイクルスタート** で続行し、**停止**（F7）で停止します。**シングルブロック** は 1ブロックずつ実行し、**次のブロック** は次のブロックを実行します。

ツインは、特に指定しなければ実時間で動作します。**シミュレーション > 速度** で実時間の半分から 50倍まで変更でき、**シミュレーション > シミュレーションを一時停止** でツインの時間が止まります（止まっている間、機械の状態は変化しません）。

ツインが検査するもの

プログラムと段取りの形状から導かれるもの：

- コントローラ自身のアラームとエラー（機械のコントローラが出すものと同じ）：ソフトリミット、リミットスイッチ、何にも接触しないプローブ移動
- ツールホルダー、コレットナット、主軸と、素材、治具、テーブルとの衝突、およびニアミス
- 早送り（`G0`）での切削、主軸が止まっている状態での切削
- 素材の底面を突き抜けて、治具やテーブルに達する切削
- 工具が噛み込むか折損する状況：工具に対して高すぎる 1刃当たりの送り、刃溝の根元の曲げ応力が工具の許容を超える場合
- Z クリアランス：各工具とタッチプローブが、Z を上端にした状態で段取りを越えられるか（MC-1 ハードウェアを参照）
- 段取りに部品の設計形状がある場合：部品に残った材料と、深すぎる切削（**表示 > 色分け > ワーク：部品との偏差**）

検出事項は、発生するたびに、ドックの **メッセージ** タブに表示されます。

ツインが推定するもの

物理モデルが計算するもの：

- 切削抵抗、トルク、パワー、主軸負荷：主軸は実物と同様に負荷がかかると回転が落ち、負荷が大きすぎると主軸のインバータがトリップします
- 切削中に工具と機械がたわむ様子が、切削された形状に現れます
- びびり：経路に沿った指標（**表示 > 色分け > 軌跡：びびり指標**）と、切削が安定する主軸回転数（**分析** ワークスペース：**安定限界線図を計算**）
- 工具の振れ：スロットと穴は、その分だけ広く仕上がります
- 機械時間：コントローラ自身のプランニングとドライバの動きから求めます
- ボールエンドミルがパスの間に残すカスプ（その形状であり、表面粗さではありません）

軌跡は、送り速度、切削抵抗、切りくず厚さで色分けすることもでき（**表示 > 色分け**）、ドックの **グラフ** タブには、実行の経過が時間に対してプロットされます。

> **補足**
>
> 推定値は、お使いの機械が測定されるまでは、一般的なパラメータ（公開されている切削データ、鋭利な超硬工具、機械の設計値）に基づいています。プログラムどうしの比較や、切削前の問題の発見に使ってください。部品の測定値として使うものではありません。

ツインがモデル化しないもの

- 表面のテクスチャと粗さ
- 構成刃先、切りくず、およびその排出
- 切削中に薄い壁がたわむこと
- 部品自体の発熱
- 工具摩耗：1.0 では、すべての工具が新品です（アルミニウムを切削する工具は、アルミニウムが刃先に付着し、部品にバリが出たり仕上がりが悪くなったりするため、ほとんどの場合、先に交換されます）
- 工具が折れた後に起こること
- プログラム検査のグリッドより細かい部分：実行の前に、プログラム検査はワークを約 0.1 mm のセルで調べます（100 × 100 mm を超える素材ではもっと粗く、最大 100 万セル）。そのため、それより薄い壁を通る早送りや、壁をそれより小さくかすめる工具は、検査で気付かれないことがあります。確認するには、プログラムをツインで実行してください。ツイン自身の切削は 0.02 mm のセルで計算します。

樹脂とマイクロ工具

0.1〜0.5 mm の工具で PMMA（アクリル）を切削する場合（たとえばマイクロ流路）、ツインはさらに次のものを扱いません。

- 切削温度：PMMA はガラス転移点（約 105〜115 °C）に近づくと軟化し、約 70 °C を超えると切りくずが工具や流路に溶着して付着します。1 刃あたりの送りと回転数は、工具メーカーが示す範囲に収めてください
- エアブローやクーラントと、それらが熱や切りくずに及ぼす影響
- 流路やポケットの縁に沿ったバリ
- 薄い壁：PMMA では、厚さ約 0.15 mm 未満の壁は切削でたわむか折れます。ツインが削るのは工具の経路とそのたわみが残す形で、壁自体のたわみは計算しません
- 切削自体の減衰：これほど小さな工具は数十 kHz でたわみ、ゆっくり切削します。工具の逃げ面が切削面をこすることで振動が減衰しますが、ツインはこれを扱わないため、こうした工具のびびり指標は安全側になります

モデルとその前提：ツインのモデル。

結果とレポート

- 3Dビューのワークは、ツインが切削したものです。色分けは切削の深さを示し、部品の設計形状がある場合は、それとの偏差を示します。
- ドックの **メッセージ** タブには検査の検出事項が、**グラフ** タブには切削抵抗、負荷、速度の時間変化が表示されます。**ファイル > グラフをエクスポート (CSV)...** で保存できます。
- AIクライアントは、Studio の MCP ツールを通じて、同じ結果を読み取れます：MCP ツール のリファレンスを参照してください。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

検査、レポート、段取りシート

部品を削り終えたあと、タッチプローブが設計と照らし合わせて部品を測定します。Studio は、測定した値を、ツインの推定値と並べて検査レポートに書き出します。部品を削る前には、段取りシートが、ジョブの段取りの仕方をオペレータに伝えます。どちらも Pro エディションに含まれます。

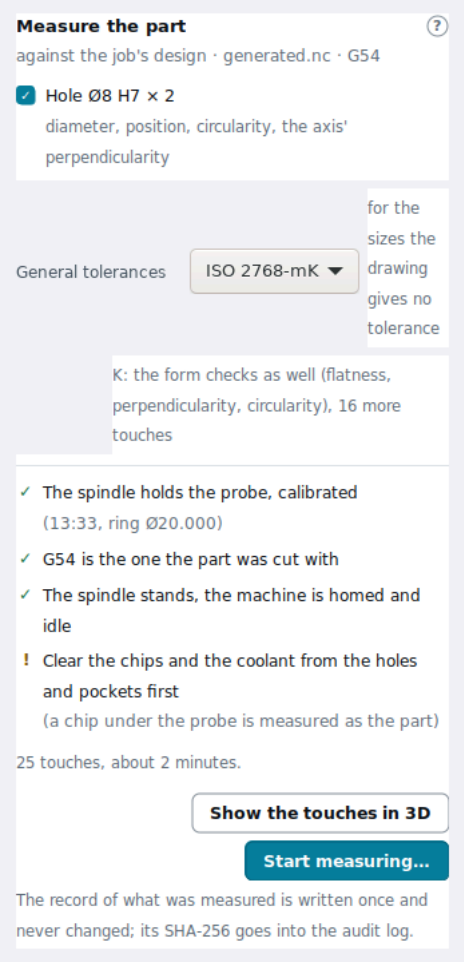
タッチプローブで部品を測定する

部品は、削ったその場で、機械から取り外す前に測定してください。**部品を測定** カードは、**準備** の **原点** ステップにあり、プローブのカードの下にあります。このカードは、部品を、その CAM ジョブの設計と照らし合わせます。対象は、各フィーチャの寸法、はめあい、公差です。1.0 では、このカードはプローブのカードと同じくツイン上で測定します。ツインが実機に追従している場合は、実機側で、その機械自身の制御を使って測定してください。

必要なもの：

- 削り終えたときのまま固定されている部品。ワーク原点を動かさないでください。位置は、ワーク原点を基準に測定されます。
- 主軸に取り付けた、リングゲージで校正したタッチプローブ。プロービングを参照してください。
- **CAM** または AIアシスタントで作成したプログラムの CAM ジョブ。

> **補足**
>
> 測定は Pro エディションに含まれます。ほかのエディションでは、カードに **Pro** のラベルが表示され、**測定開始...** が試用の方法を示します。以前に作成した記録とレポートは、どのエディションでも開けます。



測定するには：

1. **準備** で、**原点** ステップをクリックします。結果：カードに、部品を何と照らし合わせるかが表示されます。それは、部品を削ったときのプログラムとワーク座標オフセット（たとえば G54）です。
2. 測定するフィーチャを選択します。最初は、すべて選択されています。結果：各フィーチャに、プローブが測定する項目が表示されます。プローブで測定できないフィーチャは灰色で表示され、その理由が示されます。
3. **普通公差** で、個別の公差のない寸法に適用する等級を選択します。幾何公差の検査を行うには、H、K、L のいずれかを付けます（公差を参照）。結果：カードに、接触の回数と、おおよその所要時間が表示されます。
4. 確認項目を読みます：主軸に取り付けたタッチプローブが校正済みであること、ワーク原点が部品を削ったときのものであること、主軸が停止していること、機械が原点復帰済みでアイドルであること。
5. 穴とポケットから、切りくずとクーラントを取り除きます。プローブの下にある切りくずは、部品として測定されます。
6. プローブがどこに接触するかを見るには、**接触点を 3D で表示** をクリックします。結果：3Dビューに、側面からの接触と下向きの接触が表示されます。

7. **測定開始...** をクリックします。結果：プローブが各フィーチャに接触し、カードに、いま何を測定しているかが表示されます。

> ⚠ **警告**
>
> プローブは、部品の近くを送り速度で動きます。測定中は、機械から手や工具を離してください。

測定する項目：

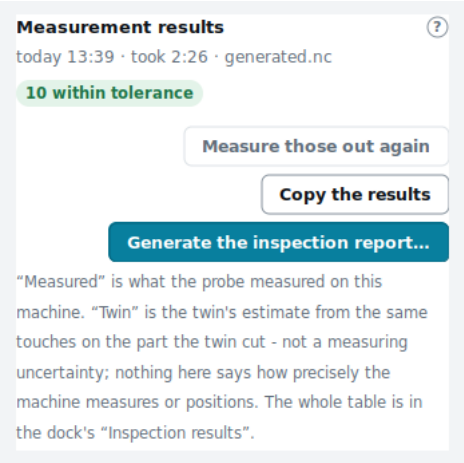
フィーチャ	検査項目
ポケット、輪郭	壁の寸法と位置、底面の深さ
穴、ボス	直径と位置
溝	幅と深さ
正面削り	深さ
面取り	幅、深さ、角度
V溝（Vカッターで行う彫刻または V彫刻）	幅、深さ、角度、その線からのずれ
エンドミルによる彫刻、ねじ、3D 曲面	測定しません

幾何公差の検査では、該当する場合に、平面度、直角度、真円度、真直度が追加されます（形状を参照）。

プローブの測定が終わると、測定のカードに代わって **測定結果** カードが表示されます。このカードには、次の内容が示されます。

- 部品を測定した日時、所要時間、プログラム
- 公差内、公差外、未判定それぞれの検査項目の数。未判定には、**ツインでは未判定**（その箇所のセルが公差の 4分の1 より粗い）、**公差なし**、**未測定** があります
- 公差外の各検査項目の、測定値、限界値、超過量または不足量
- 最初に公差外となったフィーチャの、最も可能性の高い原因と、次にすること

1.0 では、プローブはツインが削った部品を測定するため、原因はツインだけから判断します。ツイン自身の部品も同じ接触点で公差外であれば、プログラムがそのように削っています。**CAM** で修正してください。そうでなければ、測定値とツインの推定値の違いは、ツインがモデル化しているプローブの特性（球の有効半径と、一般的なプローブではプリトラベル。プリトラベルはリングゲージでの校正で取り除けます）と、プローブが接触する箇所でのツインのセルの大きさだけです。リングゲージでプローブを校正し、もう一度測定してください。示される原因は候補であり、診断ではありません。



カードのボタン：

- **公差外を再測定**：公差外の検査項目をもう一度測定します。たとえば、プローブを校正し直したあとに使います。
- **結果をコピー**：表を、列をタブで区切ったテキストとしてコピーします。表計算ソフト用です。
- **検査レポートを作成...**：レポートを書き出します（検査レポートを参照）。

表全体は、ドックの **測定結果** タブにあります。各行が 1つの検査項目で、**設計値**、**公差**、**測定値**、**偏差**、**ツイン** の推定値、判定が並びます。判定は、✓ か、どれだけ外れているかを添えた X です。公差外の行は赤で表示され、3Dビューでは部品の上に結果が示されます。

Check	Desi...	Tolerance	Measu...	Deviation	Twin	
Hole 1 diameter	Ø8	H7 +0.015/0	8.001	+0.001	8.004	✓
Hole 1 position X	8.000	±0.2 (m)	8.000	+0.000	8.001	✓
Y	32.000	±0.3 (m)	32.000	+0.000	32.000	
Hole 1 circularity	○	0.015 (K)	0.002		0.003	✓
Hole 1 axis' perpendicularity	⊥	0.4 (K)	0.000		0.001	✓
Hole 2 diameter	Ø8	H7 +0.015/0	8.001	+0.001	8.004	✓
Hole 2 position X	52.000	±0.3 (m)	52.001	+0.001	52.001	✓
Y	32.000	±0.3 (m)	32.001	+0.001	32.000	
Hole 2 circularity	○	0.015 (K)	0.002		0.003	✓
Hole 2 axis' perpendicularity	⊥	0.4 (K)	0.000		0.001	✓

各検査は、記録として保存されます。記録は 1回だけ書き込まれ、変更されることはありません。その SHA-256 は、監査ログに記録されます（コピーを監査ログと照合するを参照）。

AIアシスタントも、`inspect_part` ツールを使って、同じ方法で測定します。たとえば「bracket.nc の部品をプローブで検査して」と頼みます。

1. アシスタントが、何を測定するか、および幾何公差の検査で追加される接触の回数と秒数を伝えます。
2. プローブが動く前に確認します。結果：応答には、各検査項目が、測定値、ツインの推定値、判定とともに一覧されます。

公差

各検査項目は、公差に照らして判定されます。寸法の個別の公差やはめあいは、CAM ジョブで寸法の欄に入力したとおりに使われます（たとえば 8 H7）。それ以外の寸法には、普通公差が適用されます。

普通公差（ISO 2768-1）

ISO 2768-1 には、f（精級）、m（中級）、c（粗級）、v（極粗級）の 4つの等級があります。個別の公差のない寸法には、その寸法の区分に対する等級の公差が適用されます。等級は、カードの **普通公差** で選択します。アシスタントに指定することもできます。たとえば「ISO 2768-m で検査して」と頼みます。

形状：平面度、直角度、真円度、真直度（ISO 2768-2）

等級の後ろに H、K、L のいずれかを付けます。たとえば「mK」です。このとき、プローブは次のように形状も検査します。

検査項目	測定方法
平面度	底面を 3 × 3 の格子状にプロービング
直角度	壁または穴の軸線を 2つ目の深さでもう一度プロービング
真円度	穴またはボスを 8点でプロービング
真直度	長い壁を 3〜8点でプロービング

プローブが動く前に、幾何公差の検査で追加される接触の回数をカードまたはアシスタントが伝えます。

> **補足**
>
> 各方向で校正されていないタッチプローブには、固有のロビングがあります。レポートには、ツインのプローブモデルがその大きさをどれだけと見なしているかが、各真円度の横に記載されます。

はめあい（ISO 286）

はめあいとともに書かれた寸法（たとえば Ø8 H7）は、そのはめあいの公差域に照らして判定されます。はめあいと公差を参照してください。

面取りと V溝

面取りの幅、深さ、角度と、V溝の幅、深さ、角度には、個別の公差を指定しない限り、ISO 2768-1 の f 級（精級）が適用されます。

ツインが推定するもの

ツインは、シミュレーション上のワークに対して、同じプロービングを行います。ツインの推定値は、ツイン上でプログラムが削った結果です。測定値と推定値が異なる場合は、機械と段取りを確認してください。確認するのは、工具の長さ、ワーク原点、治具、摩耗した工具です。

ツインの推定値は、物理モデルによる計算です。機械の測定の不確かさではありません。

検査レポート

検査は、すべて記録として保存されます。Studio は、その記録から次のレポートを書き出します。

- 表計算ソフト用の CSV ファイル（UTF-8）
- 印刷と保管用の PDF ファイル

レポートには、ジョブ、段取り、工具、プローブとその校正が記載され、フィーチャごとに、呼び寸法、公差とその根拠、ツインの推定値、測定値、偏差、判定が記載されます。測定しなかったフィーチャについてはその理由が記載され、ツインのワークを上から見た図も示されます。

書き出すには、アシスタントに、たとえば「最新の検査のレポートを作成して」と頼みます。同じ記録からは、バイト単位まで同じファイルが得られます。

レポートの保存場所

検査レポートと段取りシートは、レポートフォルダーにあります。

- Windows：ドキュメント\Twinwright Studio\Reports
- Linux：~/Documents/Twinwright Studio/Reports

設定でプログラムフォルダーを移動した場合、レポートフォルダーはその中にあります。

コピーを監査ログと照合する

Studio は、すべての記録、レポート、段取りシートの SHA-256 を、監査ログに記録します。各エントリーは、直前のエントリーに連鎖しています。誰かから送られてきたコピーは、次のいずれかの方法で確認できます。

- アシスタントに、ファイルのパスを添えて「このファイルを監査ログと照合して」と頼みます。
- または、そのファイルが書き出されたコンピューターで、次を実行します。

```
mc1sim --inspect-verify <file>
```

応答には、ログがそのファイルの SHA-256 を保持しているか、いつ書き出されたか、ログの連鎖が途切れていないかが示されます。1ビットでも変更されたコピーは見つかりません。

段取りシート

段取りシートは、機械のそばにいるオペレータ向けの 1ページです。何を、どの順序で段取りするかが書かれています。Studio は、ジョブごとに 1枚を、使用中の言語で作成します。用紙は A4 で、米国とカナダでは Letter です。

作成するには、AIアシスタントに、たとえば「bracket.nc の段取りシートを作成して」と頼みます。アシスタントは `setup_sheet` ツールを使います。

> **補足**
>
> 段取りシートは Pro エディションに含まれます。

段取りシートの内容

- プログラム：各プログラムの名前、バージョン、行数と、その SHA-256 の先頭 16進数 8桁。機械に読み込まれているプログラムと照らし合わせてください。
- 素材と治具：素材の寸法と材質、およびツインでの素材の取り付け方。バイスの場合は、パラレル、くわえ代、素材がジョーの上に出ている高さ。
- 工具：各工具の番号、種類、直径、刃数、突き出し長さ、仕様、およびその工具を使う加工工程。
- 各段取り：部品がその段取りにどう至るか（裏返す、別の方法で固定する）、ワーク原点の位置、およびそれを見つけるプローブサイクル。サイクルの名前は、プローブパネルでの名前と同じです。
- 移動範囲：プログラムが移動する X、Y、Z の最小値と最大値（ワーク座標）。
- Z クリアランス：Z クリアランスを参照してください。
- 加工工程：各加工工程の方式と材料への入り方、工具、回転数、送り、深さ、所要時間。
- 上面図：素材、バイスのジョー、ワーク原点、各フィーチャの輪郭。
- 注記：CAM の注記。英語で表示されます。

所要時間は、プログラムの送り速度と、機械の早送り速度および加速度から推定されます。工具交換は含まれません。ツインで試運転すると、加減速を含めた時間が得られます。

Z クリアランス

段取りごとに、タッチプローブと各工具が段取りの上を越えられることを、シートが確認します。また、各工具がプログラムの最高点と最低点に届くことも確認します。高さは、ツインの Z バジェットによるもので、テーブル上面から測ります。

項目	届く必要がある位置
タッチプローブ	Z が可動範囲の上端近くにあるときに、段取りの最高点の 1 mm 上
各工具、上げたとき	段取りの最高点の 0.5 mm 上、およびプログラムの最高点
各工具、下げたとき	その工具自身の加工工程の最低点

各行には **十分** と表示されます。足りない場合は、何 mm 足りないかが赤で表示されます。

> **注意**
>
> 赤い行は、この段取りではプログラムを実行できないことを意味します。突き出し長さの短い工具か短いプローブを使うか、たとえばクランプを使って、部品をより低く固定してください。

クランプが、プログラムの最高点より高く出ていることがあります。その場合、シートにそのことが記載されます。プログラムは、クランプの位置を横切ってはいけません。ツインでの試運転で、それを確認できます。

印刷と署名

1. 段取りシートを Web ブラウザで開きます。
2. 印刷します。結果：最初の段取りより後の各段取りは、新しいページから始まります。
3. 機械のそばで、末尾の欄に記入します。オペレータ、日付、初品検査（合格か不合格か、およびその検査レポートの番号）です。

同じジョブの新しい段取りシートは、ファイルを置き換えます。その SHA-256 は、監査ログに記録されます。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

実機の接続

草案（第14ラウンド）：本文は執筆中で、節が増える可能性があります。

機械を扱う 2つの方法

- お使いの送信ソフト（bCNC、gSender、Universal G-code Sender）から、USB ポート経由で MC-1 を動かします。送信ソフトを参照してください。
- ツインを実機と並べて、シャドーモード（プレビュー）で動かします。送信ソフトは機械ではなく Studio に接続します。Studio はすべてのブロックを機械に渡すと同時に、同じブロックをツインで実行し、両者を比較します。

Studio 自身から実機にプログラムを送信する機能は、1.0 には含まれません。

シャドーモード（プレビュー）

> **補足**
>
> 1.0 のシャドーモードは、コマンドラインから起動するプレビューです。機械の代わりに 2つ目のツインを使ってテストされており、実機ではまだテストされていません。ウィンドウには、ポートの選択、接続、比較の確認を行う場所がまだありません。これは今後のバージョンで追加されます。

シャドーモードを開始する

必要なもの：電源を入れて USB でこのコンピューターに接続した MC-1、およびインストール済みの送信ソフト。

1. 機械のポートを調べます。Windows では **デバイス マネージャー > ポート (COM と LPT)** で確認できます（例：COM5）。
2. コマンドプロンプトから、そのポートを指定して Studio を起動します。

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

   結果：Studio がポートを開き、ツインが機械に追従します。

3. 送信ソフトで、機械ではなく Studio の送信ポート `localhost:2323` に接続します。結果：送信ソフトのブロックが Studio を経由して機械に送られ、ツインが同じブロックを実行します。

Studio の送信ポートは、このコンピューターだけで待ち受けます。別のコンピューターから接続するには、`--bind any` を追加します。

ツインが比較するもの

- 応答：ブロックごとの `ok` またはエラー、およびアラームの順序。
- 経路：機械が報告する位置とツインの経路の比較、およびツインがどれだけ先行しているか、または遅れているか。
- 設定：機械が接続されたとき、送信ソフトが接続する前に、1回だけ。Studio が機械とリンクした後で、送信ソフトを接続してください。
- コントローラ：設定と同時に、1回だけ。両者が同じファームウェアで動作しているか。また、両方で Twinwright 独自カーネルが動作している場合は、そのバージョンと、機械のボードが備えているもの（主軸、エンコーダ、ハンドル、軸、プローブ、接続）。ツインはボードの補助ポートと SD カードをモデル化していません。比較結果にはそれらが一覧表示されますが、相違としては数えません。
- プログラムの各部分の実行時間。

シャドーモード保護

Studio を `--shadow-guard` も付けて起動すると、ツインは、送信ソフトからの各ブロックを、機械に届く前に検査します。ツインが衝突を予測すると、そのブロックを保留し、機械をフィードホールドにします。ウィンドウでは、**診断** にシャドーモード保護の状態が表示されます。

機械に届かないもの

シャドーモードで機械に届くのは、送信ソフトのブロックと、何も変更しないツイン自身の問い合わせ（設定とステータス）だけです。

- ウィンドウの操作はツインにだけ作用します。ツインが機械に追従している間は、機械を動かしたり開始したりする操作は拒否されます。プローブパネルはロックされます。
- AIクライアントは読み取りしかできません。機械の状態を見ることはできますが、何も変更しません。
- Studio に接続された外付けのハンドルは、ツインにだけ作用します。

> ⚠ **警告**
>
> シャドーモードでは、ウィンドウの停止と非常停止は機械を止めません。機械を止めるには、機械自身の非常停止か、送信ソフトの停止を使ってください。

送信ソフト

Twinwright 独自カーネルでテストされた送信ソフトと、その設定方法：送信ソフト。

相違があるとき

相違があるのは、ツインのプロファイルまたは設定がこの機械のものと一致していないか、機械がモデルにない動作をしているためです。まず設定を確認してください。比較結果には、相違のある設定が一覧表示されます。

AIアシスタント

AI は、Studio で使えるのと同じツールを使ってツインを操作できます。段取りを設定し、プログラムを作成・検査して、シミュレーションで実行します。Studio 自身のアシスタントで AI と会話することも、Claude Desktop などの AIクライアントを接続することもできます。ウィンドウの機械を変更することになる操作は、許可を待ってから実行されます。AI は、どの安全機能にも含まれません。

AI をオンにする

新規インストールでは、AI はオフです。オンにするには、次のいずれかを行います。

- トップバーの AI の状態（**AIオフ**）をクリックし、**ウィンドウの AIスイッチ** を選びます。
- **ツール > 設定... > AI と MCP** で **AIスイッチ** をオンにし、**OK** をクリックします。
- アシスタントでメッセージを送信します。これでスイッチがオンになります。

Studio は、スイッチを最後に設定した状態のまま、次回の起動に引き継ぎます。**AI** メニュー（**AIアシスタント...**、**AIアクティビティ...**、**AIクライアントを接続...**）は、オペレータレベル以上で表示されます。

スイッチがオフでも、AIクライアントは、バックグラウンドで練習用の機械を使えます。ただし、ウィンドウの機械は使えません。

AIサービスとそのキー

アシスタントは、ご自身のアカウントを使って、選択した AIサービスと通信します。選べるサービスは次のとおりです。

- **Anthropic (Claude)**、**OpenAI**、**DeepSeek**、**Qwen (Tongyi)**、**Kimi (Moonshot AI)**、**Zhipu (GLM)**
- **LM Studio**（**ローカル**） または **Ollama**（**ローカル**）：ご自身のコンピューター上のモデル。キーは不要です。
- **カスタム**（**OpenAI 互換**）：入力したアドレスにある、OpenAI 互換のインターフェースを持つ任意のサービス。

設定するには、次のようにします。

1. **AI > AIアシスタント...** を開きます。
2. **プロバイダー** を選びます。Studio が **ベース URL** を入力します。
3. キーを **APIキー** に貼り付けて、**キーを保存** をクリックします。結果：キーは暗号化されて、このコンピューターに保存されます。Windows では、お使いの Windows アカウントだけが読み取れます。Linux では、デスクトップのキーリング（Secret Service：GNOME Keyring、KWallet）に保存します。キーリングがない場合は、ご自分のユーザーアカウントだけが読めるファイル内のキーで暗号化し、キーの下の行にそのことが表示されます。
4. **モデル一覧** をクリックして、**モデル** を選びます。

Studio は、キーを、設定、レジストリ、監査ログのいずれにも書き込みません。保存したキーを削除するには、欄を空にして **キーを保存** をクリックします。

> **補足**
>
> AIサービスに送る内容は、そのサービスに、ご自身のアカウントのもとで、そのサービスの規約に従って送られます。送られるのは、メッセージです。**ウィンドウのプログラム、段取り、検査結果を添付** をオンにしている場合は、プログラム、段取り、検査結果も送られます。AI が使うツールの結果も送られます。すべてをコンピューター上にとどめるには、ローカルモデルを使ってください。

クラウドサービスへの最初のメッセージの前に、アシスタントは、何がそこに送られるかを一度だけ伝えます。続けるには **送信** を、LM Studio に切り替えるには **ローカルモデルを使う...** をクリックします。メッセージは入力欄に残ります。**ツール > 設定... > AI と MCP** には、同意したサービスの一覧があります。**もう一度確認する** を押すと、アシスタントが再び確認します。

アシスタントパネル

AI > AIアシスタント... は、アシスタントを専用のウィンドウで開きます。

- 下部の入力欄に入力します。Enter で送信し、Shift+Enter で改行します。**停止** で応答を止めます。
- **ウィンドウのプログラム、段取り、検査結果を添付**（**変更があった場合**） は、メッセージごとに、ウィンドウの内容を AI に渡します。既定でオンです。
- 会話には、AI が使った各ツールが表示されます。うまくいったときは ✓、いかなかったときは ✗ が付きます。
- AI が作る画像（ワーク、ツールパス、部品からの偏差、比較）が、会話に表示されます。画像をダブルクリックすると、拡大したり保存したりできます。
- **新しい会話** は、会話を新しく始めます。Studio を閉じると、会話は保存されません。

ウィンドウを閉じても、非表示になるだけです。終了するまで、会話は残ります。

AI にできること、できないこと

スイッチがオンのとき、AI には次のことができます。

- 機械の状態の読み取り、原点復帰、ロック解除、ジョグ、G コード の行の送信
- 素材、工具、ワーク原点、部品の設定、プロービング、検査
- プログラムの検査、シミュレーション、最適化、作成、編集、読み込み、および実行の開始または再開始
- 実行とその結果の読み取り、比較、画像の作成
- 監査ログとライセンスの状態の読み取り

有料エディションに属するツール（検査、レポート、ネスティング、最適化）は、そのエディションでのみ動作します。

AI は、Studio の設定、言語、単位、ユーザーレベル、AIスイッチ、Studio の確認方法、ライセンス、監査ログを変更できません。

機械を変更する前の確認

AI がウィンドウの機械を変更する前（トップバーの AI のメニューと、**ツール > 設定... > AI と MCP** にあります）には、2つの選択肢があります。

- **毎回確認する**（既定）
- **確認しない**（**AI がウィンドウを直接操作**）

AI がウィンドウの機械を変更しようとすると（動かす、電源を投入する、非常停止を解除する、プログラムを読み込むか実行する、段取りや工具を変更する）、機械バーの上にバーが開きます。このバーには、どの AI が何をしようとしているかが表示され、カウントダウンが行われます。

1. 質問を読みます。
2. **許可** または **拒否** をクリックします。Enter では答えられません。マウスを使うか、Tab でボタンまで移動して Space を押してください。

60秒以内に答えがない場合は、**拒否** として扱われます。許可は、その操作に付随する後続の要求にも、2分間適用されます。

Studio は、AI が機械を停止する、フィードホールドする、リセットする、非常停止を押す、または読み取るだけの場合、事前に確認しません。練習用の機械と、AI がシミュレーションに使う非表示のコピーについても、確認しません。

実機では読み取り専用

ウィンドウのツインが実機に追従している場合（シャドーモード。1.0 ではプレビューです。実機を参照）、AI は読み取りと分析しか行いません。機械を動かすこと、段取りやワーク座標オフセットを変更すること、そこでプログラムを実行することはできず、確認も求めません。Studio がそれらの要求を拒否します。練習用の機械は、これまでどおり動作します。

AIアクティビティ

AI > AIアクティビティ... は、このコンピューター上で AIクライアントが行っていることを表示します。

- **バックグラウンドの機械**：AI が使う練習用の機械で、ウィンドウには表示されません（ここでの「ウィンドウ」は、ウィンドウに表示されている機械を指します）。
- **実行履歴**：直近 16件のシミュレーションと最適化。
- **接続中の AIクライアント**：誰がどのように接続しているか、およびその最後の呼び出し。

ここから、実行の停止、実行の画像の表示、バックグラウンドの機械を閉じること、ツインホストの停止と起動ができます。ツインホストは、練習用の機械を保持するバックグラウンドのプログラムです。AIクライアントが接続すると起動し、使われないまま 30分たつと終了します。

ドックの **AIアクティビティ** タブには、ウィンドウの機械に対する AI の呼び出しが一覧されます。時刻、呼び出し、機械、呼び出し元、結果です。

他の AIクライアントを接続する（MCP）

MCP（Model Context Protocol）に対応する AIクライアントは、Studio のツールを使えます。

1. **AI > AIクライアントを接続...** を開きます。
2. 使うクライアントをクリックします。**Claude Desktop**、**ChatGPT desktop**（**Codex モードと Work モード**）、**Cursor**、**VS Code (GitHub Copilot)**、**LM Studio**（**ローカルのオープンモデル**）のいずれかです。結果：Claude Desktop と ChatGPT desktop では、Studio がクライアントの設定に加える変更を表示し、**OK** をクリックするとその変更を行います。クライアントの元のファイルは、その隣に残されます。そのほかのクライアントでは、それぞれのインストールページが開きます。
3. クライアントを完全に終了し、もう一度開きます。
4. Studio で **ウィンドウの AIスイッチ** をオンにします。

ほかのクライアントでは、**その他のクライアント：構成をコピー** で、クライアントに貼り付ける設定がコピーされます。クライアントは、バックグラウンドで `TwinwrightStudio.exe --mcp` を起動します。**コマンドと構成** に、実際のコマンドが表示されます。

アドレスだけを受け付けるクライアント

アドレスだけを受け付けるクライアントの場合は、AI のメニューで **ローカル URL**（**HTTP、URL のみ受け付けるクライアント用**）をオンにして、**ローカル URL**（**HTTP**）**をコピー** を使います。このアドレスは、このコンピューター上でのみ機能し、付属のトークンが必要です。**新しいトークン** を使うと、古いトークンは使えなくなります。

ツール、その引数、その結果については、MCP toolsを参照してください。

記録される内容

ツール > 監査ログ...（Ctrl+L）には、AI の操作が、**実行者** 列に AI の名前とともに記録されます。AI の記録だけを見るには、**実行者** を **AI** で絞り込みます。呼び出しをダブルクリックすると、その呼び出しのすべての記録がまとめて表示されます。

既定の **標準** レベルでは、ログには次の内容が残ります。

- AIスイッチのオンとオフ、および Studio の確認方法の変更
- 各質問とその答え、ウィンドウの機械に対する AI の各操作
- 各ツール呼び出しと、その引数の短縮形
- クライアントの接続と切断
- アシスタントのメッセージ：要約には先頭の 180バイト、記録には最大 2000バイト

キー、トークン、パスワードは `***` として記録されます。

言語と単位

Studio は、初回起動時にシステムの言語に従います。言語、単位、数値・日付・金額の表示形式は、いつでも変更できます。プログラムとファイルはそのままです。

言語を変更する

ツール > Language / 语言 を選び、続けて言語を選びます。English、简体中文（中国語、簡体字）、Français、Deutsch、日本語、Español から選べます。このメニューの名前はどの言語でも同じなので、読めない言語で表示されていても見つけられます。

ウィンドウはすぐに切り替わるため、Studio を再起動する必要はありません。コンソールにすでにある行は、書かれたときの言語のままです。

ツール > 設定... > 全般 > 言語 には、もう 1つ選択肢 **Windows に合わせる** があります。これを選ぶと、Studio は起動のたびにシステムの言語に従います。Studio に言語パックがない言語は、英語になります。

単位：ミリメートルとインチ

ツール > 単位 では、**地域の設定に従う**、**メートル法 (mm)**、**インチ** のいずれかを選びます。変更はすぐに反映されます。

インチを選ぶと、Studio は次のように表示します。

量	メートル法	インチ
長さ	mm	in
送り速度	mm/min	in/min
切削速度	m/min	ft/min
力	N	lbf
トルク	N·m	lbf·in
出力	W	hp
温度	°C	°F

主軸回転数は min^{-1} のままです。ジョグのステップは 0.0005、0.005、0.05、0.5、2 in になり、ジョグ速度は 4、12、60、120 in/min になります。

単位が変えるのは、Studio が表示し、読み取る内容であり、ファイルは変わりません。プログラムは、`G20` または `G21` のどちらで書かれていても、書かれたとおりに実行されます。Studio が作成するプログラムは、ミリメートル（`G21`）で書かれます。AI のツールとログは、ミリメートルで動作します。

地域：数値、日付、金額

ツール > 設定... > 全般 > 地域 では、数値、金額、日付の表示形式を設定します。選択肢は、**Windows に合わせる**（**設定 > 時刻と言語**）、**言語に合わせる**、またはリストの中の地域です。同じページには、次の項目があります。

- **通貨**：USD、EUR、CNY などの ISO 4217 コード。空欄の場合は、地域の通貨になります。
- **通貨記号**：空欄の場合は、通貨本来の記号になります。
- **機械の時間単価**：機械時間 1時間あたりの費用。コストの推定に使われます。0 の場合は、何も表示されません。

地域は、Studio の数値の小数点記号と桁区切り記号、実行履歴・バージョン・レポートの日付、レポートの用紙サイズ（米国とカナダは Letter、それ以外は A4）を決めます。位置やプログラム中の数値など、機械に属する数値には、常にピリオドを使い、桁区切りは使いません。

数値を入力する

Studio は、間違えると部品がだめになりかねない入力欄では、入力された数値を厳密に読み取ります。対象は、CAM の入力欄、**準備** の寸法と位置、プローブの長さ、主軸回転数です。

- 小数点記号は 1つだけ使います。ピリオドまたはカンマです（`13.3` または `13,3`）。
- 桁区切りは書きません。
- 数値の後ろに、その入力欄の単位を付けてもかまいません：`13 mm`、`0.5 in`、`30°`。
- IME で入力した数字（全角数字）も読み取られます。

地域が小数点記号をピリオドで書く場合、`1,000` は疑問になります。1 のつもりか、1000 のつもりか。Studio は推測せず、意図するほうを書くよう求めます。地域がカンマで書く場合、`1,000` は 1 です。

「と解釈」の意味

入力した内容が、Studio の数値の書き方と異なる場合、入力欄には Studio が読み取った値が表示されます。`13,3` と入力すると、**13.3 と解釈** と表示されます。インチでは、小数点以下 4桁で表示されます：**0.3750 と解釈**。先に進む前に確認してください。

入力した内容を Studio が読み取れない場合、入力欄が赤くなり、その下の行に理由が表示されます。たとえば「『1,2,3』は長さではありません」です。Studio はその値を受け付けません。修正するまで、入力欄は最後に受け付けた値のままです。

インチの分数

インチでは、長さの入力欄に分数も入力できます。

- `3/8`、`5/4`
- `1 1/2` または `1-1/2`
- `2 3/8"`、`1/2 in`（インチ記号は `"`、`”`、`in` のいずれでもかまいません）

入力欄には、読み取った小数と分数が表示されます：**0.3750（3/8"）と解釈**。Studio は、小数点を含む分数（`1.5/2`）、整数の後ろの仮分数（`1 5/4`）、スラッシュの前後のスペースを受け付けません。

Studio は、インチの寸法も、1/64 in 刻みまでの分数で表示します。工具の直径と長さ、素材の寸法、穴の直径と深さは、`1/4"` や `1 1/2"` のように表示されます。1/64 in の倍数でない寸法は、小数 4桁の `0.2010 in` で表示されます。座標、送り、速度、測定値は、常に小数です。

ミリメートルでは、分数は読み取られません。「ミリメートルの入力欄は分数を受け付けません：4.7625 と書くか、インチに切り替えてください」と表示されます。

工具の説明は、どちらの単位系でもインチの寸法を読み取ります。たとえば `flat 1/4in n2` です。

主軸回転数

主軸回転数は、1000 min^{-1} ずつ変更する代わりに、直接入力できます。

1. **実行** で、右の列にある **手動** を開きます。
2. **主軸回転数** の表示をダブルクリックするか、そこで Enter を押します。
3. 回転数を入力して、Enter を押します。Esc を押すか、ほかの場所をクリックすると、元の回転数のままになります。

主軸が手動で回転しているとき（プログラムが実行中でないとき）は、新しい回転数がすぐに主軸に送られます。主軸が止まっているときは、Studio は、次の **主軸正転** または **主軸逆転** のために回転数を保持します。プログラムの実行中は、プログラムが回転数を決めるため、入力欄は開きません。

主軸の範囲外の回転数は、入力欄が赤くなり、受け付けられません。MC-1 の範囲は 1000〜24000 min^{-1} です。

英語のままのもの

- G コード と、Studio が書き込み、読み込むファイル（ジョブ、段取り、レコーディング、ライセンスファイル）
- AI のツール、および AIクライアントとの間でやり取りされる内容
- Studio 自体から生成されるリファレンスの章：MCP tools、Command line、Machine profile schema
- プログラムに関する各監査記録のメッセージ（ログは、使用中の言語でも保持します）
- 1.0 のドイツ語、フランス語、日本語、スペイン語では、コントローラのエラーとアラームの説明、コントローラ設定のヘルプ、一部のライセンスメッセージ

検査レポートの PDF は、フォントに使用中の言語の文字がある場合はその言語で、ない場合は英語になります。CSV ファイルは常に使用中の言語です。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

キーボードとアクセシビリティ

機械、メニュー、パネルのボタン、エディタは、すべてキーボードで操作できます。3Dビューの回転とズーム、3Dビュー内の点の選択、パネルのサイズ変更にはマウスが必要です。Windows では、スクリーンリーダーがパネルのボタンと表示値を読み上げます。

ヘルプ > キーボードショートカット（または **?**）に、Studio のすべてのキーが一覧されます。

機械のキー

キー	動作
F1	**非常停止**：押して作動、もう一度押して解除
F2	**電源投入**
F3	**原点復帰**
F4	**ロック解除**
F5	**サイクルスタート**
F6 または Space	**フィードホールド**
F7	**停止**：フィードホールドし、機械が止まったらリセット。もう一度押すと、すぐにリセット
F8	**冷却** のオン・オフ
F9	**主軸正転**
F10	**主軸逆転**
F11	**主軸停止**
F12 または Ctrl+X	**リセット**

ジョグとハンドル：

キー	動作
← →	X をジョグ
↓ ↑	Y をジョグ
Page Down、Page Up	Z をジョグ
Shift とこれらのいずれか	キーを押している間ジョグ
1–5	ステップを 0.01、0.1、1、10、50 mm のいずれかにする
6	**押している間ジョグ**：すべてのジョグキーが、押している間ジョグする
Ctrl+1〜Ctrl+4	ジョグ速度を 100、300、1500、3000 mm/min のいずれかにする
Esc	**ジョグ停止**
X、Y、Z	ハンドルの軸を選ぶ
, と .	ハンドルを 1クリック分下げる、または上げる（Shift で 10クリック分）

オーバーライド：

キー	動作
Ctrl+↑、Ctrl+↓	送り +10 %、−10 %
Ctrl+→、Ctrl+←	主軸 +10 %、−10 %
Ctrl+0	送りと主軸を 100 % に戻す

ジョグキーを 1回押すごとに、1ステップ動きます。押し続けるジョグは、キーを離すか、ウィンドウがフォーカスを失うと止まります。インチでは、ステップは 0.0005〜2 in、速度は 4〜120 in/min です。

F キーは、どこにいても使えます。ほかの機械のキーは、テキストボックス、リスト、エディタに入力している間は働きません。3Dビューをクリックすると、キーの操作が機械に戻ります。

数字キー、および , と . のキーは、キーボード上の位置で働くため、どのキー配列でも使えます。

非常停止

F1 を押すか、機械バーの右端にある丸い赤いボタンをクリックします。ツインでは、機械のボタンと同じように、非常停止回路が開きます。軸のドライバと主軸インバータの電源が切れ、コントローラが ALARM:10 を発生させます。もう一度 F1 を押すと解除されます。その後、手順に従って作業に戻ります：**電源投入**、**リセット**、**原点復帰**。

> ⚠ **警告**
>
> F1 と赤いボタンが止めるのはツインです。実機の非常停止ではありません。実機では、機械自体の非常停止ボタンを使ってください。

ウィンドウのキー

キー	動作
Alt+1〜Alt+6	**準備**、**CAM**、**プログラム**、**実行**、**分析**、**診断** のいずれかへ移動
Ctrl+Tab、Ctrl+Shift+Tab	次または前のワークスペース
Ctrl+J	ドックを折りたたむ、または展開する
Ctrl+Shift+M	機械を切り替える
Ctrl+O	**プログラムを開く...**
Ctrl+,	**設定...**
Ctrl+L	**監査ログ...**
Ctrl+Q	**終了**
? または /	**キーボードショートカット**
Alt とメニューの文字	そのメニューを開く。Alt だけを押すと、メニュー名が強調表示される
Tab、Shift+Tab	パネルのボタン間を移動する。Space で、枠の付いたボタンを押す
Esc	ポップアップまたはサブウィンドウを閉じる

> **補足**
>
> F1 はヘルプではなく、非常停止です。マニュアルを開くには、**ヘルプ > ドキュメント** か、パネルの **?** ボタンを使います。

プログラム ワークスペースのエディタには、専用のキーがあります。

キー	動作
Ctrl+E	エディタに移動し、機械に戻る
Ctrl+K	**読み込んで検査**
Ctrl+S	**バージョン保存**
Ctrl+Shift+S	プログラムをファイルとして保存
Ctrl+D	保存したバージョンとの **比較**
Ctrl+F、Ctrl+G、Ctrl+H	検索、行へ移動、置換
Ctrl+B	その行の **ブレークポイント**
Ctrl+Shift+B	**条件で一時停止**
Ctrl+.	その行に対する検査の提案を適用
Alt+↓、Alt+↑	次または前の検出事項、または差分
Shift+F10 またはメニューキー	その行のメニュー
Ctrl+Shift+O	**最適化**
Ctrl+Shift+G	**生成...**
Ctrl+Shift+R	機械にあるプログラムを取り込む
Esc	比較を閉じる

マニュアルのウィンドウでは、Ctrl+F で検索、Alt+← と Alt+→ で戻る・進む、Esc で閉じます。

準備 でワークを配置している間は、矢印キーで移動し（Shift で細かく、Ctrl で粗く）、Q と E で回転し、Enter で配置し、Esc で元に戻します。

キーの完全な一覧

ヘルプ > キーボードショートカット には、上記のすべてのキーが 1つの一覧で表示されます。次のコマンドにはキーがありません。それぞれのメニューを使ってください。

- **シングルブロック** と **次のブロック**（**プログラム** メニューと機械バー）
- **プログラムを検査**
- 軸のゼロ設定とゼロ位置への移動（**機械 > ワーク原点**）
- ツールセッターとプローブの動作（**機械 > ツールセッターとプローブ**）
- 1000 min^{-1} ずつの主軸回転数の変更と、早送りオーバーライド（**機械**）
- シミュレーションの一時停止と速度（**シミュレーション**）

スクリーンリーダーを使う

Windows では、Studio は、システムのアクセシビリティインターフェースを通じてスクリーンリーダーと連携します。Studio の各パネルで、スクリーンリーダーは次の情報を受け取ります。

- 各ボタンの名前、ボタンの種類、オン・開いている・使用不可のどれか、ショートカット、ヒント

- 表示値のテキスト：機械バー（状態、プログラム、進捗、次の手順、時計）、トップバー（機械、状態、AI、レベル、エディション）、右の列（位置、主軸、ゲージ、モーダル行）
- Tab によるフォーカスの移動のひとつひとつ
- メニュー：各項目の文言、ショートカット、使用不可かチェック済みか、現在どの項目にいるか

フォーカスがどこにあっても、次の 2つは、発生した時点で通知されます。アラーム（次の手順を含む）と、AI による機械の変更の要求です。

マニュアルのウィンドウでは、ページのテキストが、見出しとリンクを含めてスクリーンリーダーに渡されます。

1.0 では次のとおりです。

- 機械の状態のその他の変化は、通知されません。聞くには、フォーカスを機械バーに移します。
- プログラムリスト、コンソール、メッセージ、エディタ、工具テーブルは、スクリーンリーダー自身の規則で読み上げられます。曲線は読み上げられません。
- Linux では、スクリーンリーダーはサポートされていません。

3Dビューで読み上げられるもの

3Dビュー自体は、スクリーンリーダーに何も渡しません。3Dビューに表示される数値は、右の列と機械バーにあり、そちらはスクリーンリーダーが読み上げます。ワークの点をクリックすると、ビューの下の文（プログラムの行と位置）が読み上げられます。

大きな文字と工場モード

Studio は、Windows で画面ごとに設定した表示スケールに従います。Studio 独自の文字サイズ設定はありません。

表示 > 外観 > 工場モード（大） を選ぶと、機械バー、そのボタン、非常停止、文字、プログラムリスト、位置表示が大きくなり、**実行** ワークスペースには、腕を伸ばした距離から読むためのレイアウトが適用されます（ライト、ダーク、工場モードを参照）。

システムのアニメーションがオフになっている場合、3Dビューは、滑らかに移動する代わりに、新しいビューへ瞬時に切り替わります。

トラブルシューティング

思いどおりに動かないときの対処法です。各節では、何が見えるか、通常は何を意味するか、詳しい答えがどこにあるかを説明します。どれにも当てはまらない場合は、診断情報を収集して、当社に連絡してください（最後の節）。

Studio が起動しない

- Studio には、64ビット版の Windows 10 または 11 が必要です。それ以外のバージョンでは、インストーラーが停止します。システム要件を参照してください。
- ウィンドウは開くものの、3Dビューが空のままか、利用できないと表示される場合：**ツール > 設定... > ファイル** で、**ソフトウェアレンダリング（グラフィックスカードに問題がある場合。次回起動時に反映）** にチェックを入れ、Studio をもう一度起動します。グラフィックスカードのドライバを更新すると、原因が解消することがよくあります。3Dビューが起動しない場合を参照してください。
- 起動中に Studio が閉じた場合、次回の起動時にレポートが提示されます（Studio が予期せず終了したを参照）。

機械がアラーム状態になっている

アラーム状態では、機械バーの状態表示が赤くなります。バーにはアラームの名前とその意味が表示され、それを実行するボタンとともに **次の手順** が示されます。手順が複数ある場合は、**すべての手順** に作業に戻るまでの流れが一覧されます。ボタンをクリックするか、**機械 > 次の手順を実行** を使ってください。Studio は、1つの手順を終えるたびに、手順を計算し直します。

- 何にも触れなかったプローブの動作は、ALARM:5 で停止します。手順（**リセット**、**ロック解除**）を実行してから、サイクルが測定面をどこに想定していたかを確認してください。
- 機械の可動範囲を超える動作は、ソフトリミットで停止します。プログラムリストで赤く示された行と、そのとき使っていたワーク原点を確認してください。
- アラームの後は、位置が信頼できなくなっていることがあります（**位置** の下の行にその旨が表示されます）。切削する前に、もう一度原点復帰してください。

それぞれの場合の手順は、アラームと対処にあります。すべてのアラーム、その原因、解除方法は、アラームにあります。

プログラムの行がエラーを返す

`error:` と番号が表示された場合、コントローラがその行を拒否したことを意味します。その行は実行されず、機械はその行のために動きませんでした。メッセージには、番号とその意味が示されます。

- **プログラム > プログラムを検査** は、機械を動かさずに、プログラム全体をコントローラのチェックモードに通し、拒否された最初の行を一覧します：「5行目：error:22 (...)」。それらの行を修正して、もう一度検査してください。
- よくある原因は、コントローラが受け付けないコード、ワードの欠落（中心または半径の指定がない円弧、`F` より前の切削移動）、または 1行に同時に書けない 2つのコードです。

すべてのエラーとその原因は、エラーにあります。

プログラムの検査で問題が見つかった

エディタは、入力中、読み込み時、実行後にプログラムを検査します。各検出事項は、その行の下に、規則の番号と種別（エラー、警告、注記）とともに表示されます。

- **エラー** は、コントローラが拒否する行、または段取りでは実行できない動作です。実行する前に修正してください。
- **警告** は、実行はされるものの、意図したとおりに動かない可能性がある行です。たとえば、ワークの上面より下での早送り、`S` を指定せずに始動した主軸、工具交換の後に、工具が開始位置の上に来る前にプログラムが下降する動作です。
- 検査のプレビュー部分には、機械の電源が入っていて原点復帰が済んでおり、プログラムが実行されていないことが必要です。それまでは待機します。

規則と検出事項の修正方法は、プログラム検査にあります。それ以外にツインでの実行で検査される内容は、ツインが検査するものにあります。

プローブサイクルが始まらない

プローブカードの検査項目が赤い間、または何かが足りない間、**プロービング開始...** はグレーのままです。その場合は、ボタンの下の行に、何が足りないかが表示されます（X と Y を指定する、リングの直径を指定する、実行中の呼び出しを待つ）。

- **主軸にタッチプローブがありません**：**準備** の **工具** ステップで、タッチプローブを選び、**工具交換** をクリックします。
- **主軸が回転している**：停止します（**主軸停止**、F11）。
- **素材の上を横切る余裕がありません**：この行には、Z を上端にしたときのプローブ先端の高さ、段取りの上面、不足している量が示されます。短いプローブを使うか、バイスの代わりにクランプや両面テープを使って素材をより低く固定してください。
- **ツインが実機に追従している**：プローブは、機械自身の制御装置で行ってください。
- 何にも触れなかったサイクルは、ALARM:5 で停止します（機械がアラーム状態になっているを参照）。

サイクルが止まるときを参照してください。

ライセンスとアクティベーション

ヘルプ > ライセンス には、適用中のエディション、ライセンスの状態、有効期限が表示されます。

- **オンラインで有効化**：受け取ったコードをそのまま入力します。コードとして読み取れるようになると、**有効化** が使えるようになります。サーバーの応答は、その下に表示されます。応答が、ライセンスが使われているコンピューターが多すぎる、終了している、一時停止中、未払い、返金済みのいずれかである場合は、**ライセンスのページを開く** で当社の Web サイトのライセンスのページが開きます。そこで、コンピューターの登録を解除したり、サブスクリプションを更新または管理したりできます。
- **インターネットなし**：同じダイアログに、オフラインの手順があります。Studio が要求コードを発行します。インターネットに接続できるコンピューターで、ライセンスのページからそのコードに対応するライセンスファイルを取得し、そのファイルをここにインポートします。
- 更新されなかったサブスクリプションは、さらに 14日間は使えます（ライセンスに **猶予期間中** と表示されます）。その後、またはライセンスが終了した場合、Studio は再び Learning エディションになります。プロジェクトとプログラムは影響を受けません。

ライセンスとライセンスが終了したときを参照してください。

AIアシスタントが応答しない

- 入力欄の下の行には、アシスタントが何をしているかが表示されます（サービスの応答待ち、ツインがツールを処理中）。
- **プロバイダー**、**ベース URL**、**モデル** を確認します。クラウドサービスでは、キーが保存されていること（**キーを保存**）も確認します。
- ローカルモデルの場合は、そのサーバー（LM Studio のサーバー、Ollama）が動作していて、モデルが選択されている必要があります（**モデル一覧**）。
- 応答に時間がかかることがあります。**停止** で応答を終了できます。
- クラウドサービスへの最初のメッセージは、確認を待ちます。Studio は、何がそのサービスに送られるかを一度表示し、同意した場合にだけ送信します（サービスごとに 1回確認します）。**ローカルモデルを使う...** を選ぶと、すべてがこのコンピューター上にとどまります。プライバシーを参照してください。
- AI がウィンドウの機械を変更しようとすると、ウィンドウ下部のバーで確認が求められます。60秒以内に答えがない場合、その要求は拒否されたものとして扱われます。

AIアシスタントを参照してください。

Studio が予期せず終了した

次回の起動時に、Studio は、何が起きたかのレポートをどうするかを尋ねます。そのフォルダーを開くか、ファイルとして保存するかを選びます。何も送信されません。レポートには、プログラム、ファイル名とフォルダー名、キー、コード、人名は含まれません。30日より古いレポートは、起動時に削除されます。プライバシーを参照してください。

サポートを受ける

ヘルプ > 診断情報の収集... は、サポートに必要な情報を、`ドキュメント\Twinwright Studio\Reports` にある 1つの ZIP ファイルにまとめます。内容は、バージョン、設定（キーは `***` として記録）、機械アーカイブ、コントローラの設定、監査ログの直近 7日分、クラッシュレポート、ライセンスの状態です。**現在のプログラムと段取り** は、チェックを入れない限り含まれません。パス、ファイル名、コンピューターのユーザー名は削除されます。

送る前に、ファイルを開いて内容を確認してください。Studio 自身は何も送信しません。連絡先のアドレスは **ヘルプ > バージョン情報** にあります。

Operations manual

システム要件

Twinwright Studio を動かすためにコンピューターに必要なものと、3Dビューがグラフィックスに求めるものを説明します。

Windows

- 64 ビット版の Windows 10 または Windows 11（x64）。これ以外のバージョンでは、インストーラーが停止します。
- ほかにインストールするものはありません。Studio は 1つの実行ファイルで、Windows に付属するライブラリだけを使います。

Linux

- x86-64 の Linux、glibc 2.38 以降：Ubuntu 24.04、Debian 13、Fedora 39 以降、およびそれらをもとにしたディストリビューション。
- X11 または Wayland のデスクトップ。
- AppImage には FUSE 2（Ubuntu では `libfuse2t64` パッケージ）。ない場合は `.tar.gz` を使います。
- オンラインでの有効化と試用：libcurl（Ubuntu では `libcurl4t64` パッケージ。デスクトップには通常入っています）。ない場合、Studio はオフラインで有効化します。
- AI サービスのキーには、デスクトップのキーリング（Secret Service：GNOME Keyring または KWallet）。ない場合、Studio はキーをあなただけが読めるファイルに保存します。

グラフィックス

3Dビューは Direct3D 11 を使います。Direct3D 11 に対応していれば、グラフィックスカードでも内蔵グラフィックスでも使えます（機能レベル 10.0 以上）。対応するものがない場合、Studio は Windows のソフトウェアレンダラー（WARP）で描画します。動作はしますが、ビューが遅くなり（最大で毎秒 30 フレーム）、工具のクローズアップはありません。

ヘルプ > バージョン情報 の **ディスプレイ** に、Studio がどちらを使っているかが表示されます。

3Dビューが起動しない場合

3Dビューがなくても、ウィンドウのそれ以外の部分は動作します。ビューが空のままのとき、または使用できないと表示されるときは、次のようにします。

1. **ツール > 設定... > ファイル** を開きます。
2. **ソフトウェアレンダリング（グラフィックスカードに問題がある場合。次回起動時に反映）** にチェックを入れます。
3. Studio を閉じて、もう一度起動します。

グラフィックスカードのドライバを更新すると、原因が解消することがよくあります。

Linux

Linux では、3D ビューは OpenGL 4.1 以上を使います。過去 10 年のグラフィックスなら Mesa ドライバーにあります。グラフィックスドライバーがない場合は、Mesa のソフトウェアレンダラーが（遅くなりますが）ビューを描きます。試すには、端末からプログラム名の前に `LIBGL_ALWAYS_SOFTWARE=1` を付けて Studio を起動します。

メモリとディスク

- メモリ：8 GB 以上。
- ディスク：Studio 本体とマニュアルに約 62 MB、それに加えてデータ用の空き容量が必要です。データは使うほど増えます。監査ログは、既定で 180日間、最大 2 GB まで保持され（**ツール > 設定... > ログ**）、プログラムの全バージョンとレポートはすべて保持されます（データとバックアップを参照）。

画面とスケーリング

- 1280 × 720 以上。Studio のレイアウトは 1600 × 1000 以上を想定しています。
- 表示スケールは任意です。Studio は画面ごとのスケールに従い、ウィンドウを別の画面に移すと、その画面のスケールに合わせ直されます。

実機

1.0 では、実機への接続はプレビュー機能で、コマンドラインから行います（実機を参照）。接続できるのは、115 200 ボーのシリアルポート（USB）か、ネットワーク上のコントローラです。

Studio がネットワークを使うのは、次の場合です。

- G コード送信ソフトは、このコンピューター上のツインにポート 2323 で接続します。
- ブラウザビューは、このコンピューター上のポート 8765 を使います。
- アドレスを受け付ける AIクライアントは、オンにしたとき、このコンピューター上のポート 8766 を使います。
- オンラインのアクティベートと更新は、ライセンスサーバーに接続します。

Studio のポートは、このコンピューターだけで待ち受けます。ほかのコンピューターからは接続できません。

Linux では、シリアルポートは `dialout` グループのものです。一度だけアカウントを追加し（`sudo usermod -aG dialout $USER`）、ログインし直してください。

インストールとアンインストール

Twinwright Studio のインストール方法（多数のコンピューターに確認なしで一度にインストールする方法を含みます）、各ファイルの置き場所、アンインストールの方法を説明します。

Linux については Linux の場合 を参照してください。

インストール

1. `TwinwrightStudio-<version>-setup.exe` を実行します。
2. ライセンス契約書を読み、**I Agree** をクリックします。
3. インストール先のフォルダー（`%LOCALAPPDATA%\Programs\Twinwright Studio`）をそのまま使うか、別のフォルダーを選びます。
4. 最後のページで、Studio を起動するかどうかと、デスクトップにショートカットを置くかどうかを選びます。

インストーラーの表示言語は英語か中国語で、システムの言語に従います。

自分用か、コンピューターの全員用か

インストーラーは、実行した人のために Studio をインストールします。管理者権限は不要で、スタートメニューの項目とデスクトップのショートカットは、その人のものになります。このコンピューターを使う人はそれぞれ、自分用に Studio をインストールします。

ライセンスは人ではなくコンピューターに属し、このコンピューターのすべてのユーザーが共有します（ライセンスフォルダーを参照）。

Studio は、ファイルの関連付け、`PATH`、ファイアウォールの設定を変更せず、サービスもインストールしません。

サイレントインストール

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

- `/S` は、ページを表示せずにインストールします。Studio は起動しません。
- `/DESKTOP` は、デスクトップのショートカットを追加します。
- `/D=` は、インストール先のフォルダーを指定します。必ず最後の引数にし、引用符は付けません。フォルダー名に空白が含まれていても同じです。

確認なしでアンインストールするには、Studio 本体のフォルダーで `uninstall.exe /S` を実行します。

置き場所

内容	場所
Studio 本体	`%LOCALAPPDATA%\Programs\Twinwright Studio`
作業内容：プログラム、バージョン、レポート、監査ログ	`ドキュメント\Twinwright Studio`
Studio 自身の状態：設定、機械、AI サービスのキー	`%LOCALAPPDATA%\Twinwright\Studio`
ライセンス	`%ProgramData%\Twinwright\licence`

Studio 本体のフォルダー

`TwinwrightStudio.exe`、マニュアル（`manual\`）、`README.txt`、`THIRD-PARTY-NOTICES.txt`、`uninstall.exe` が入っています。このフォルダーはアップデートしても変わらないので、接続した AIクライアントはそのまま動作し続けます。

ドキュメント

`ドキュメント\Twinwright Studio`（Windows が割り当てるドキュメントフォルダーで、リダイレクトされている場合も同じです）の中身は、次のとおりです。

- `programs\`：すべてのプログラムのバージョンと、CAM ジョブのバージョン。
- `Reports\`：検査レポート、段取りシート、診断パッケージ。
- `logs\`：監査ログ。

ツール > 設定... > ファイル と **ログ** で、プログラムとログの保存場所を別の場所に移せます（次回起動時に反映）。

Studio 自身の状態

`%LOCALAPPDATA%\Twinwright\Studio`：設定、ウィンドウの位置、機械ごとの「ワールド」（ワーク、工具、オフセット、コントローラの設定）、AI サービスのキー（お使いの Windows アカウント用に暗号化されます）、クラッシュレポート（30日間保持されます）。

ライセンスフォルダー

`%ProgramData%\Twinwright\licence` には、ライセンスファイル（試用を含む）が保存され、このコンピューターのすべてのユーザーで共有されます。Studio は、最初のアクティベートまたは試用のときに、このフォルダーを作成します。Studio がここに書き込めない場合は、管理者がフォルダーを作成し、ユーザーに書き込み権限を与えることができます。

特殊な構成では、次の環境変数でフォルダーを移せます。

環境変数	移動対象
`TWINWRIGHT_STATE_DIR`	Studio 自身の状態

環境変数	移動対象
`TWINWRIGHT_WORKDIR`	プログラムのフォルダー
`TWINWRIGHT_LOGDIR`	監査ログ
`TWINWRIGHT_LICENCE_DIR`	ライセンスフォルダー

新しいバージョンへのアップデート

Studio を閉じてから、新しいバージョンのインストーラーを実行します。同じフォルダーにインストールされ、作業内容、設定、機械、ライセンスはそのまま残ります。Studio は、アップデートがあるかどうかを自分で確認しません（アップデートを参照）。

Studio がバックグラウンドでまだ動作している場合（AIクライアントが起動したものの場合があります）、インストーラーは古い実行ファイルを退避させ、次回のアップデートまたはアンインストールのときに削除します。

アンインストール

Windows の **設定** > **アプリ** > **Twinwright Studio** > **アンインストール** を使うか、Studio 本体のフォルダーにある `uninstall.exe` を実行します。

アンインストール後に残るもの

アンインストールすると、Studio 本体、そのショートカット、**アプリ** の項目が削除されます。次のものは残ります。

- `ドキュメント\Twinwright Studio` にある作業内容。
- `%LOCALAPPDATA%\Twinwright\Studio` にある Studio 自身の状態。
- ライセンスフォルダー。
- 接続した AIクライアントの設定にある `twinwright` の項目。

すべてを削除するには、これらを手動で削除してください。ライセンスフォルダーを削除しても、試用を使い直すことはできません。ライセンスサーバーが、各コンピューターに各エディションの試用を 1 回ずつ与えるためです。

Linux の場合

インストール

Linux 版 Studio は 1 つのファイル、AppImage です。システムには何もインストールされません。

1. ファイルを確認します：`sha256sum TwinwrightStudio-<version>-x86_64.AppImage` の結果がダウンロードページの SHA-256 と一致すること。
2. 置いたままにできる場所に、バージョンを含まない名前で置きます（更新後も接続した AI クライアントがそのまま使えるように）。そして実行可能にします：

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

1. 起動します：`~/Applications/TwinwrightStudio.AppImage`、またはファイルマネージャーでダブルクリックします。

AppImage には FUSE 2 が必要です。そのために起動しない場合はインストールするか（Ubuntu 24.04 以降：`sudo apt install libfuse2t64`）、代わりに `.tar.gz` を使います：好きな場所に展開し、そのフォルダーの `TwinwrightStudio` を起動します。

Linux での保存場所

何	どこ
プログラム（マニュアル付き）	AppImage（置いた場所）
あなたの作業：プログラム、バージョン、レポート、監査ログ	`~/Documents/Twinwright Studio`
プログラム自身の状態：設定、機械	`~/.local/state/twinwright/studio`（`$XDG_STATE_HOME` が設定されていればその下）
AI サービスのキー	デスクトップのキーリング。ない場合は状態フォルダー内の、あなただけが読めるファイル
ライセンス	`/etc/twinwright/licence`、または `~/.local/share/Twinwright/licence`

ご自分のアカウントでライセンスフォルダーを作成または書き込みできる場合、フォルダーはそのコンピューターのもので、すべてのユーザーが共有します。できない場合、Studio はライセンスを `~/.local/share/Twinwright/licence`（`$XDG_DATA_HOME` が設定されていればその下）に保存します。あなただけのもので、引き続きこのコンピューターでのみ有効です。コンピューターのユーザー全員で 1 つのライセンスを共有するには、`sudo install -d -m 1777 /etc/twinwright/licence` でフォルダーを一度作成してください。

Linux での試用と有効化

試用、有効化、更新は Windows と同じくオンラインで行い、システムの libcurl を使います。libcurl がない場合やインターネットに接続できない場合は、オフラインで有効化します（ライセンス を参照）。

Linux での更新と削除

更新するには、Studio を終了し、新しい AppImage を古いものと同じ名前で置き換えます（上の手順 2）。削除するには、AppImage を削除します。どちらの場合も、ドキュメント、状態フォルダー、ライセンスフォルダー、キーリング内のキー、接続した AI クライアントの `twinwright` 項目は残ります。すべて消すには、これらを手で削除してください。

ライセンス、シート、アクティベーション

ライセンスがない場合、Studio は Learning エディションとして動作します。有料ライセンスを使うと、そのエディションの機能が加わります。この章は、ライセンスを購入、アクティベート、管理する担当者向けです。価格はウェブサイトに掲載しています。

この章をお読みの時点で有料ライセンスがまだ販売されていない場合は、ウェブサイトの価格ページにその旨を掲載しています。販売開始までは、Studio は Learning エディションとして動作し、Maker または Pro を 30 日間無料で試用できます（有料エディションを試すを参照）。アクティベーションに関する各節は、ライセンスの販売開始後に適用されます。

エディション

エディション	加わる機能	使える人
Learning	ツインのすべて、エディタ、実機、サンプル、AI 接続、CAM（ポケット、輪郭、穴あけ、彫刻、3D 走査線仕上げ）	だれでも無料。学習、教育、収入を伴わない個人のプロジェクト用
Maker	アダプティブ加工、すべての 3D 仕上げ加工、残り加工、両面加工と複数工程	1 人。その人が作ったものの販売を含む
Pro	Maker のすべてに加えて、ネスティング、プログラムオプティマイザ、プローブによる検査（設計との比較）とそのレポート	組織。あらゆる商用利用、シート単位

MC-1 オーナーライセンスは 1 シートの Pro で、ライセンスに記載されたシリアル番号の MC-1 と一緒に使います。機械を譲渡すると、ライセンスも次のオーナーに移ります。

有料ライセンスには次の種類があります。

- **サブスクリプション**：月単位または年単位で、解約するまで続きます。
- **買い切り**：永久にお客様のものです。購入日（またはアップデートの有料更新日）から 12 か月以内に公開されたすべてのバージョンが対象です。
- **MC-1 オーナー**：買い切りと同じで、機械に結び付きます。

シートとコンピューター

有料のシートは、同時に 1 人が最大 3 台のコンピューターで使えます。複数の人が共用する 1 台のコンピューターは 1 台と数えます。

アクティベートのたびに、ライセンスはコンピューターの指紋のハッシュでそのコンピューターに紐づきます。指紋そのものがコンピューターの外に出ることはありません。ライセンスがどのコンピューターで使われているかは、ライセンスのページ twinwright.com/ja/licence でアクティベーションコードを入力すると確認できます。

有料エディションを試す

有料エディションはそれぞれ 30 日間、コンピューター 1 台につき 1 回ずつ無料で試用できます。

1. 有料の機能を使います。たとえば CAM パネルのアダプティブ加工です。結果：その機能に必要なエディションが表示されます。
2. **30 日間試用...** を選びます。
3. 試用で使える機能と期限を確認してから、**試用を開始** を選びます。結果：Studio が Twinwright のライセンスサーバーに試用を申請し、表示された日付まで、そのエディションの機能が使えます。

試用のとき Studio が送信するのは、エディション、コンピューターの指紋のハッシュ、Studio のバージョンだけです。サーバーは、各コンピューターに各エディションの試用を 1 回ずつ与えます。試用期間中に Studio をインストールし直した場合、もう一度試用を開始すると、同じ終了日に戻ります。試用が終わると Studio は Learning エディションに戻り、プロジェクトとプログラムはそのまま残ります。

インターネットに接続していないコンピューターでは、Studio は代わりに要求コードを表示します。インターネットに接続できるコンピューターで、ライセンスのページ twinwright.com/ja/licence を開き、「インターネットのないコンピューターで試用」の欄でエディションを選び、要求コードを貼り付けて、試用ファイルをダウンロードします。そのあと、元のコンピューターでファイルを読み込みます。手順はインターネットに接続していないコンピューターをアクティベートすると同じです。

オンラインでアクティベートする

ライセンスと一緒に届いたメールのアクティベーションコード（`TW-XXXXX-XXXXX-XXXXX-XXXXX` の形式）と、インターネット接続が必要です。

1. **ヘルプ > ライセンス...** を選びます。
2. **オンラインで有効化...** を選びます。
3. アクティベーションコードを入力します。大文字と小文字、スペース、ハイフンは区別しません。
4. **有効化** を選びます。結果：ライセンスファイルがライセンスフォルダーに入り、ダイアログにエディションが表示されます。

アクティベートで送信するのは、アクティベーションコード、このコンピューターの指紋のハッシュ、Studio のバージョンだけです。

ライセンスが使える台数のコンピューターですでに使われている場合は、先にライセンスのページで 1 台の登録を解除してください（ライセンスを別のコンピューターに移すを参照）。

インターネットに接続していないコンピューターをアクティベートする

オンラインにならないラボや工場のコンピューターでは、この手順を使います。アクティベーションコード、インターネットに接続できる別のコンピューター、USB メモリーなど小さなファイルを運ぶ手段が必要です。

1. インターネットに接続していないコンピューターで、**ヘルプ > ライセンス...** を選びます。
2. **オフラインで有効化...** を選びます。
3. **コピー** を選びます。結果：要求コードがクリップボードにコピーされます。要求コードに含まれるのは、このコンピューターの指紋のハッシュと Studio のバージョンだけで、個人情報は含みません。
4. 要求コードをテキストファイルに貼り付けて、インターネットに接続できるコンピューターに運びます。
5. そのコンピューターで twinwright.com/ja/licence を開き、「インターネットに接続していないコンピューター」の欄にアクティベーションコードと要求コードを貼り付けます。

6. 「ライセンスファイルをダウンロード」を選びます。結果：`twinwright-TWL-….twlic` のような名前のファイルがダウンロードされます。
7. ファイルを元のコンピューターに持ち帰ります。Studio で **ヘルプ > ライセンス…**、**ライセンスファイルを読み込む…** の順に選び、ファイルを開きます。結果：ライセンスファイルがライセンスフォルダーに入ります。

インターネットに接続していないコンピューターは、ライセンスを自動では更新しません。次の場合はこの手順を繰り返してください。

- サブスクリプションの期間が終わる前
- 買い切りの場合、最初の 30 日（返金期間）が過ぎたあとに 1 回。これで永久ファイルに置き換わります
- アップデートの期間を追加で購入したあと。

ライセンスフォルダー

ライセンスフォルダーはコンピューターのもので、特定の Windows アカウントのものではありません。フォルダー内のライセンスは、そのコンピューターを使うすべての人に有効です。

システム	フォルダー
Windows	`%ProgramData%\Twinwright\licence`
Linux	`/etc/twinwright/licence`。書き込めない場合：`~/.local/share/Twinwright/licence`

環境変数 `TWINWRIGHT_LICENCE_DIR` で別のフォルダーを指定できます。フォルダーを開くには、**ヘルプ > ライセンス…**、**ライセンスフォルダーを開く** の順に選びます。

フォルダーには、ライセンスファイル（`*.twlic`。試用のファイルも含みます）と、Studio が最後に確認した時刻を記録したファイルがあります。ライセンスファイルは複数置けます。たとえば MC-1 オーナーライセンスとサブスクリプションです。有効なライセンスファイルのうち最も上のエディションが適用されます。

> **補足**
>
> これらのファイルは編集しないでください。変更されたライセンスファイルは無効になります。その場合はライセンスのページから再度ダウンロードして読み込んでください。

Studio がフォルダーを作成できない、または書き込めない場合は、管理者がフォルダーを作成し、ユーザーに書き込み権限を与えてください。Linux では root（`sudo`）で読み込むか、`TWINWRIGHT_LICENCE_DIR` を設定します。

更新と 14 日間の猶予期間

期限前に自動更新（送るのはライセンス番号、このコンピューターの指紋のハッシュ、Studio のバージョンだけです） がオンのとき、Studio はオンラインであればライセンスを自動で更新します。

- サブスクリプション：期間終了の 5 日前から更新します
- 買い切り：返金期間の終了時に更新します。アップデートの期間を追加で購入した場合は、アップデート期間が終わる 14 日前から更新します。

Studio は起動時に 1 回、1 日に 1 回まで試します。失敗しても作業を妨げることはありません。送信するのは、このオプションに書かれているものだけです。

サブスクリプションが更新されなかった場合も、すべての機能をさらに 14 日間使えます。ライセンスには **猶予期間中** と残りの日数が表示されます。その後、Studio は Learning エディションに戻ります。

ライセンスが終了したとき

期限切れ、解約、確認の失敗は、どれも Studio を Learning エディションに戻すだけです。

- プロジェクトはこれまでどおり開けます
- 作成済みの G コードは、書き出しも機械への送信もできます
- 有料の操作は、更新するまでお休みです。

Studio がお客様のファイルを削除したり、データをロックしたりすることはありません。

買い切りライセンスのアップデート

買い切りライセンスの対象は、購入日（またはアップデートの有料更新日）から 12 か月以内に公開されたバージョンです。それより後のバージョンは Learning エディションとして動作し、ライセンスには **アップデート期間終了** と表示されます。対象となる最後のバージョンを使い続けるか、ライセンスのページでアップデートを 1 年延長してください。Studio のアップデートを参照してください。

ライセンスを別のコンピューターに移す

1. ライセンスのページでアクティベーションコードを入力し、「コンピューターを表示」を選びます。
2. 使わなくなったコンピューターの横の「このコンピューターの登録を解除」を選びます。
3. 新しいコンピューターでライセンスをアクティベートします。

登録の解除は 30 日間に 3 台までです。それ以上必要な場合は hello@twinwright.com までご連絡ください。

ハードウェアの交換やシステムの再インストールで指紋が変わると、Studio に **このライセンスは別のコンピューターに紐づいています。** と表示されます。古いエントリーの登録を解除してから、再度アクティベートしてください。それまで Studio は Learning エディションとして動作します。

アクティベーションコードをなくした場合

ライセンスのページの **アクティベーションコードをなくした場合** を使います。メールアドレスの入力を求められたら、購入時のアドレスを入力してください。そのアドレスで購入した各ライセンスに、新しいコードをメールでお送りします。ご連絡を求められたら、アクティベーションコードを受け取ったメールアドレスから hello@twinwright.com までご連絡ください。内容を確認のうえ、新しいコードをお送りします。どちらの場合も、古いコードではアクティベートできなくなりますが、アクティベート済みのコンピューターはそのまま使えます。

コンピューターの時計

時計が 2 日以上戻されると、Studio に **このコンピューターの時計が前回より前の時刻になっています。** と表示されます。遅いほうの時刻で数えるため、ライセンスや試用の日数が戻ることはありません。日付と時刻を正しく設定してください。

Studio は最後に確認した時刻を 2 か所に記録します。遅いほうの時刻で数えるため、どちらか一方を削除しても日数は戻りません。

時計を一度未来の日付に設定したあと、ライセンスや試用が早めに終了と表示された場合は、日付と時刻を正しく設定し、hello@twinwright.com までご連絡ください。

管理者向け：コマンドライン

次のオプションを付けると、プログラムはウィンドウを開きません。フォルダーは省略でき、省略するとコンピューターのライセンスフォルダーを使います。

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

- `--licence-status`：ライセンスファイルと試用を 1 つずつ、有効かどうかとその理由を一覧表示します。`--json` を付けるとすべての項目を出力します。
- `--licence-request`：このコンピューターの要求コードを表示します。
- `--licence-import`：ライセンスファイルを確認して、ライセンスフォルダーにコピーします。

Linux ではプログラム名は `twinwright-studio` です。AIクライアントは、読み取り専用の MCP ツール `licence_status` で同じ状態を読めます。

Studio が送信するものと、当社のサーバーが保存するもの：プライバシーと Studio が送信するもの。

データとバックアップ

Studio は、作業内容をドキュメントフォルダーに、機械の状態を専用のフォルダーに保存します。この章では、何がどこにあるか、何をバックアップするか、別のコンピューターへ移行する方法を説明します。

保存場所

ユーザー自身のデータは、次の 2つのフォルダーにすべて入っています。

- `ドキュメント\Twinwright Studio`：作業内容。
- `%LOCALAPPDATA%\Twinwright\Studio`：Studio 自身の状態。

ジョブ、プログラム、そのバージョン

`ドキュメント\Twinwright Studio\programs\<program>\` には、プログラムのバージョンごとに 1つのファイル（`v0001.nc`）と、いつ、誰が、なぜ保存したかのメモ（`v0001.json`）が入っています。プログラムを作った CAM ジョブも、その隣にバージョン管理されます（`job-v0001.json`）。Studio が古いバージョンを削除することはありません。

ツール > 設定... > ファイル で、プログラムのフォルダーを別の場所に移せます（次回起動時に反映）。

レポート

`ドキュメント\Twinwright Studio\Reports\`：検査レポート（PDF と CSV）、段取りシート（HTML）、診断パッケージ（ZIP）。

機械とそのワールド

機械ごとに、状態フォルダーに「ワールド」があります。テーブル、切削の結果が反映されたワーク、工具と工具テーブル、ワーク座標オフセット、コントローラの設定、温度、稼働時間、タッチプローブの校正が入っています。Studio は、ウィンドウを閉じるときにこれを書き込みます。最初の機械のワールドは `world\` に、ほかの機械のワールドは `machines\<id>\world\` にあります。

設定

状態フォルダーの `settings.json` には、**ツール > 設定...** のすべての項目が保存されます。`window.json` には、ウィンドウの位置とサイズが保存されます。

監査ログ

`ドキュメント\Twinwright Studio\logs\` には、日ごとに 1つのファイル（`audit-YYYY-MM-DD.jsonl`）があります。記録は連鎖しているため、変更や欠落があると、ログを検証したときに分かります（**ツール > 設定... > ログ > 整合性を検証**）。既定では、ログは 180日分、最大 2 GB まで保持されます。これは **ツール > 設定... > ログ** で変更できます。連鎖を途切れさせないために、フォルダー全体をコピーしてください。

バックアップの対象

Studio を閉じてから、`ドキュメント\Twinwright Studio` と `%LOCALAPPDATA%\Twinwright\Studio` の両方のフォルダーをコピーします。

次の 2つは、コピーしても引き継がれません。

- AI サービスのキー：このコンピューター上のお使いの Windows アカウント用に暗号化されています。新しいコンピューターで入力し直してください。
- ライセンス：ライセンスはコンピューターに属します。新しいコンピューターで、もう一度アクティベートしてください（ライセンスを参照）。

別のコンピューターへの移行

1. 古いコンピューターで Studio を閉じ、両方のフォルダーをコピーします。
2. 新しいコンピューターに Studio をインストールし、一度起動してから閉じます。
3. 両方のフォルダーを、新しいコンピューターの同じ場所にコピーします。
4. 設定でプログラムまたはログのフォルダーを移している場合は、それらのフォルダーが存在することを確認します。
5. Studio を起動し、ライセンスをアクティベートして、AI サービスのキーをもう一度入力します。
6. AIクライアントをもう一度接続します（**AI > AIクライアントを接続...**）。

機械の初期状態へのリセット

工場出荷時へのリセットを行うと、ウィンドウに表示されている機械が出荷時の状態に戻ります。その機械のワールド（ワーク、工具、オフセット、コントローラの設定、タッチプローブの校正）が消去され、Studio が再起動します。

1. **ファイル > 工場出荷時にリセット...** を選びます（テクニシャンレベル以上）。
2. 5秒以内に、もう一度同じ項目を選んで確定します。

リセットされるのは、その機械だけです。プログラム、レポート、監査ログ、設定、ほかの機械はそのまま残ります。リセットは監査ログに記録されます。

Studio のアップデート

Studio は新しいバージョンを自分で確認せず、自分でアップデートすることもありません。アップデートの時期はお客様が決めます。

新しいバージョンがあるか確認する

1. Studio で **ヘルプ > バージョン情報** を選びます。結果：プログラム名の下の行にバージョンが表示されます。たとえば「バージョン 1.0.0 · 2026-10-10 ビルド · コミット 4f1c2ab」です。
2. ダウンロードページ twinwright.com/ja/download を開きます。結果：現在のバージョンと、インストーラーの SHA-256 チェックサムが表示されます。

バージョンが違う場合は、ダウンロードページのものが新しいバージョンです。

アップデートをインストールする

新しいインストーラーが必要です。管理者権限は不要です。Studio はお客様の Windows アカウントにインストールされます。

1. ダウンロードページからインストーラーをダウンロードします。
2. 任意：SHA-256 チェックサムがページのものと同じか確認します。PowerShell でファイル名を指定して `Get-FileHash` を実行します。
3. インストーラーを起動します。Windows が「発行元不明」と表示した場合は実行せず、hello@twinwright.com までご連絡ください。
4. Studio を開いている場合は、インストーラーに求められたら閉じます。
5. インストーラーの指示に従います。結果：同じフォルダーのプログラムが新しいバージョンに置き換わります。

設定、AI のキー、プログラムのバージョン、監査ログ、ライセンスはプログラムのフォルダーにはないため、そのまま残ります。プログラムのパスが変わらないので、接続済みの AIクライアントもそのまま使えます。

多数のコンピューターでは、インストーラーをサイレントモードで実行できます：`setup.exe /S`。Studio のインストールを参照してください。

ライセンスの対象となるバージョン

ライセンス	有料機能が使えるバージョン
Learning エディション	すべてのバージョン
サブスクリプション	サブスクリプション期間中のすべてのバージョン
買い切り、MC-1 オーナー	購入日（またはアップデートの有料更新日）から 12 か月以内に公開されたバージョン

ライセンスのアップデート期間が終わったあとに公開されたバージョンは、Learning エディションとして動作します。ライセンスには **アップデート期間終了** と表示されます。アップデートの前に **ヘルプ > ライセンス...** で日付を確認してください。ライセンスの対象となる最後のバージョンのインストーラーは保管しておいてください。

アップデートを続けるには、ライセンスのページでアップデートを 1 年延長してください。ライセンス、シート、アクティベーションを参照してください。

インターネットに接続していないコンピューター

1. インターネットに接続できるコンピューターでインストーラーをダウンロードします。
2. USB メモリーなどで、インターネットに接続していないコンピューターに運びます。
3. 上の手順でインストールします。

新しいバージョンにしてもライセンスファイルは変わりません。更新はこれまでどおりです：インターネットに接続していないコンピューターをアクティベートする。

以前のバージョンに戻す

1. アップデートの前にファイルをバックアップします。
2. Studio をアンインストールします（Windows の設定の「アプリ」）。
3. 以前のバージョンのインストーラーを実行します。

> **補足**
>
> 以前のバージョンでは、新しいバージョンで保存したプロジェクトや設定を開けないことがあります。問題がないと確認できるまで、バックアップは残しておいてください。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

プライバシーと Studio が送信するもの

Studio はお客様のコンピューター上で動作します。Studio 自身が送信するのは、試用の開始と、有料ライセンスのアクティベートおよび更新に必要なものだけです。テレメトリーも、新しいバージョンの確認も、クラッシュ情報の自動送信もありません。AIアシスタントをクラウドサービスに接続すると、お客様のメッセージと添付した内容がそのサービスに送られます（AIアシスタントとクラウドサービスを参照）。この章では、外に出るもの、コンピューターに残るもの、その削除方法を説明します。正式な内容はウェブサイトのプライバシーポリシー twinwright.com/ja/privacy です。

Studio が送信するもの

いつ	送信先	内容
有料エディションの試用を開始したとき	Twinwright のライセンスサーバー（HTTPS）	エディション、コンピューターの指紋のハッシュ、Studio のバージョン
有料ライセンスをアクティベートしたとき	Twinwright のライセンスサーバー（HTTPS）	アクティベーションコード、コンピューターの指紋のハッシュ、Studio のバージョン
Studio が有料ライセンスを更新するとき（更新を参照）	Twinwright のライセンスサーバー（HTTPS）	ライセンス番号、同じハッシュ、Studio のバージョン

Studio 自身が Twinwright に送信するのはこれだけです。指紋はそのままの形でコンピューターの外に出ることはなく、外に出るのはハッシュだけです。オフラインでアクティベートしたコンピューターは何も送信しません。

次の 3 つは、お客様が設定した場合にだけ外に出ます。

- **クラウドサービスに接続した AIアシスタント**：お客様のメッセージと添付した内容を、そのサービスへ送ります（下記を参照）。
- **MCP で接続した AIクライアント**（Claude、ChatGPT など）：クライアントがツインから読み取った内容を、そのクライアント自身のサービスへ送り、そのサービスの規約に従って扱われます。
- **Studio が開くウェブページ**（ライセンスのページなど）：お客様のブラウザが開きます。Studio は何も付け加えません。

AIアシスタントとクラウドサービス

AIアシスタントは、お客様がその中で選んだプロバイダーで動作します。Anthropic、OpenAI、DeepSeek などのクラウドサービス（お客様自身のアカウントと APIキーを使用）か、ローカルモデルです。

クラウドサービスにメッセージを送ると、Studio はお客様のコンピューターから、お客様のキーで、次のものをそのサービスに送ります。

- お客様のメッセージと、それまでの会話
- **ウィンドウのプログラム、段取り、検査結果を添付（変更があった場合）** にチェックが入っている間は、ウィンドウの機械、段取り、プログラム（行番号付き）と、検査で見つかった問題。このチェックは最初から入っています。Studio はこれらを最初のメッセージに添付し、その後は変更があったときだけ添付します。
- アシスタントが Studio で使うツールの結果（プログラムの検査やシミュレーション、機械の状態、ログの記録など）。画像を受け付けるサービスには、結果の画像も送ります。

最初のメッセージをクラウドサービスに送る前に、Studio は送る内容を一度表示します。

これらはすべて、そのサービス自身の規約に従って扱われます。Twinwright はそのどれも受け取りません。送る量を減らすには、送信前にチェックを外すか、**新しい会話** でやり直してください。

ローカルモデル（LM Studio、Ollama）を使う場合、これらはすべてコンピューターに残ります。

このコンピューターでは、監査ログが各メッセージの最初の 180 バイトを記録し（監査ログを参照）、APIキーは暗号化して保存されます（Studio がコンピューターに保存するものを参照）。

Studio がコンピューターに保存するもの

内容	場所（Windows）	見られる人
設定、ツインの状態、AI のキー	`%LOCALAPPDATA%\Twinwright\Studio`	お客様の Windows アカウント
プログラムのバージョン	`ドキュメント\Twinwright Studio`	お客様の Windows アカウント
監査ログ	`ドキュメント\Twinwright Studio\logs`（設定で別のフォルダーを指定した場合を除く）	お客様の Windows アカウント
検査レポート、段取りシート、診断ファイル	`ドキュメント\Twinwright Studio\Reports`	お客様の Windows アカウント
異常終了レポート	`%LOCALAPPDATA%\Twinwright\Studio\crash`（30 日後に削除）	お客様の Windows アカウント
ライセンスファイル（試用を含む）	`%ProgramData%\Twinwright\licence`	このコンピューターを使うすべての人

AI のキーは、お客様の Windows アカウント用に暗号化して保存されます（Windows DPAPI）。同じコンピューターの別のアカウントからは読めません。

監査ログ

監査ログは 1 日 1 ファイルで、`audit-YYYY-MM-DD.jsonl` としてログフォルダーに保存されます。記録する内容は次のとおりです。

- AI とお客様が機械に行わせた操作
- AI パネルの各メッセージの先頭 180 バイト
- ライセンスの確認：ライセンス番号、エディション、結果（ライセンス所有者の名前は記録しません）。

キー、トークン、パスワードは `***` と記録されます。どこまで記録し、どのくらい保存するかは設定で決めます。ログはコンピューターに残り、Studio がアップロードすることはありません。

Studio が予期せず終了したとき

障害が起きた瞬間、Studio は小さなレポートを書きます。内容は、エラーの発生箇所、そこに至るまでの呼び出しの順序、プログラムのモジュールの一覧です。メモリーの内容は書きません。

次に起動したとき、Studio はバージョン情報と、ログの最後の 200 行をクリーンにしてから加えます。

- パスとファイル名はプレースホルダーに置き換えます
- 名前とメールアドレスは削除します
- G コード、AI との会話、ライセンスコード、指紋は含めません。

そのあと Studio は 1 回だけ確認します。レポートのフォルダーを開く、レポートをファイルに保存する、メッセージを閉じる、のいずれかを選べます。何も送信しません。原因の調査にご協力いただける場合は、お客様ご自身でファイルを hello@twinwright.com にお送りください。

診断情報

サポートをご依頼いただいた際に、診断情報をお願いすることがあります。**ヘルプ** メニューで「診断情報を収集」を選ぶと、Studio はチェックした項目を ZIP ファイルにまとめてレポートフォルダーに保存します。

項目	最初からチェック
バージョン、システム、グラフィックス、言語と単位	はい
設定（AI のキーを除く）	はい
機械プロファイルとコントローラの `$` 設定	はい
直近 7 日間のログと異常終了レポート（上記と同じ方法でクリーンにしたもの）	はい
ライセンスの状態：エディションと終了日（コードと指紋を除く）	はい
現在のプログラムと段取り：お客様の部品のデータ	いいえ

送信する前にファイルを開いて内容をご確認ください。自動で送信されるものはありません。

ネットワークで待ち受けているもの

AIクライアント（MCP）と送信ソフト向けの Studio のサービスは、このコンピューター（`127.0.0.1`）でのみ待ち受けます。ほかのコンピューターからは接続できません。

コマンドラインオプション `--bind any` を使うと、ツインのネットワークポート（送信ソフト用のポートなど）がネットワークに開かれます。MCP サービスはローカルのままです。

> ⚠ **警告**
>
> シャドーモードでは、送信ソフト用のポートが実機にコマンドを渡します。`--bind any` を使うと、ネットワーク上でそのポートに届く人ならだれでも機械を動かせてしまいます。`--bind any` はツインだけに、ご自身が管理するネットワークでのみ使ってください。

当社のサーバーが保存するもの

有料ライセンスごとに、ライセンスサーバーは次のものを保存します。

- シートごとに最大 3 台のコンピューターのハッシュと、それぞれを最初と最後に確認した日付。ライセンスのページで登録を解除できるようにするためです
- アクティベーションコードはハッシュとしてのみ保存します。当社でも元のコードを読み出せません
- 90 日間、試用、アクティベート、更新、変更のログ。コンピューターのハッシュ、Studio のバージョン、ネットワーク部分までに短縮した IP アドレスを含みます。

試用ごとに、コンピューターのハッシュを、エディションと試用の日付とともに保存します。各コンピューターが各エディションの試用を 1 回ずつ使えるようにするためです。

お客様の作業に関するものは、何もサーバーに届きません。

Studio が保存したものを削除する

1. ライセンスのページでこのコンピューターの登録を解除し、台数に数えられないようにします（ライセンスを別のコンピューターに移すを参照）。
2. Studio をアンインストールします。
3. 上の表のフォルダーのうち、不要なものを削除します。結果：コンピューターに Studio のものは何も残りません。

アンインストールで削除されるのはプログラムだけで、お客様のファイルは削除されません。プロジェクト、プログラムのバージョン、ログ、レポートは、お客様が削除するまで残ります。

ラボ、教室、共用コンピューター

この章は、ラボ、工場、教室など、複数の人が使うコンピューターに Studio を設定する担当者向けです。

コンピューター全体で 1 つのライセンス

ライセンスフォルダーはコンピューターのもので、特定の Windows アカウントのものではありません（ライセンスフォルダーを参照）。コンピューターでアクティベートしたライセンスは、そのコンピューターにサインインするすべての人に有効です。

有料のシートは、同時に 1 人が最大 3 台のコンピューターで使えます。複数の人が共用する 1 台のコンピューターは 1 台と数えます。そのため、次のようにします。

- 機械の横のコンピューターのような共用コンピューターには、専用のシートを 1 つ割り当てます
- 自分のコンピューターでも有料機能を使う人には、そのコンピューター用に別のシートが必要です

共用コンピューターのアクティベートは 1 回だけで、管理担当者のアカウントで行います。

各自が持つもの

Studio は Windows アカウントごとにインストールされます。各自が次のものを持ちます。

- 設定と AI のキー：キーはそのアカウント用に暗号化され、ほかのアカウントからは読めません
- プログラムのバージョン、レポート、監査ログ：各自の `ドキュメント\Twinwright Studio` にあります

エディションの試用はコンピューター単位で 1 回と数え、そのコンピューターを使う全員で共有します。ライセンスサーバーが、各コンピューターに各エディションの試用を 1 回ずつ与えるためです。

多数のアカウントやコンピューターへのインストールについては、Studio のインストールを参照してください。

教室

Learning エディションは、学習と教育のためにすべてのコンピューターで無料で使えます。アクティベートは不要です。Studio をインストールすれば、ツインのすべて、エディタ、日常的な部品のための CAM が使えます。

可能であれば、学生ごとに Windows アカウントを用意してください。設定、プログラム、ログが分かれます。

有料機能を使って教える場合、学校や研修機関は Pro のシートを利用できます。hello@twinwright.com までご連絡ください。

起動のたびに元に戻るコンピューター

起動のたびに保存したイメージに戻るラボのコンピューターでは、ライセンスフォルダーも空になります。このようなコンピューターでは、次のようにします。

1. コンピューターを 1 回オフラインでアクティベートし（インターネットに接続していないコンピューターをアクティベートするを参照）、そのライセンスファイルを保管します。
2. ファイルをイメージのライセンスフォルダーに入れるか、起動のたびに読み込みます。

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

ライセンスファイルは 1 台のコンピューター専用です。教室のコンピューターごとに、それぞれの要求コードとファイルが必要です。

インターネットに接続していないコンピューター

オフラインでアクティベートし、手動で更新します。サブスクリプションの期間が終わる前と、買い切りの最初の 30 日が過ぎたあとに 1 回です。インターネットに接続していないコンピューターをアクティベートするを参照してください。

ライセンスフォルダーに書き込めないとき

Windows では、どのアカウントでも `%ProgramData%\Twinwright\licence` を作成できます。作成したアカウントはファイルを変更でき、ほかのアカウントは読むことができます。これで十分です。1 人がライセンスを読み込めば、全員が使えます。

組織のポリシーで許可されていない場合は、次のいずれかにします。

- 管理者がフォルダーを作成し、ユーザーに書き込み権限を与える
- システムの環境変数 `TWINWRIGHT_LICENCE_DIR` で、全員が読めるフォルダーを指定する

Linux では、ご自分のアカウントで `/etc/twinwright/licence` を作成または書き込みできる場合はそれがフォルダーになります。できない場合、Studio はご自分の `~/.local/share/Twinwright/licence`（`$XDG_DATA_HOME` が設定されていればその下）を使い、あなただけのものになります。コンピューターのユーザー全員で 1 つのライセンスを共有するには、`sudo install -d -m 1777 /etc/twinwright/licence` でフォルダーを一度作成してください。

共用コンピューターでの AI

各自が、自分のキーで自分の AI サービスまたは AIクライアントを接続します。各アカウントの監査ログには、その人と AI が機械に行わせた操作が記録されます。プライバシーと Studio が送信するものを参照してください。

Technical manual

ツインのモデルとその限界

ツインは、ソフトウェアとして作った MC-1 です。ツインの中では Twinwright 独自カーネルが動作し、コントローラのまわりにある機械の各部分は、物理モデルで表されます。ツインが物理世界について示す数値は、すべて物理モデルによる推定値です。モデルのパラメータは、機械の設計、部品のデータシート、公表されている値から得ています。それぞれの MC-1 は、コミッショニング時に測定し、そのツインを測定値に合わせます（計画中）。以下の各節の最後に、そのパラメータの根拠を示します。

草稿（ラウンド 14）：本文は執筆中で、節が増えることがあります。

コントローラ

ツインのコントローラは Twinwright 独自カーネルで、機械のファームウェアと同じソースからビルドされています。モデル化されているのは、そのまわりの基板です。タイマーとステップパルス、デバウンス付きの入力、コンピューターとの接続（USB シリアルポートとそのハンドシェイク）、コントローラのフォアグラウンド処理にかかる時間が含まれます。

根拠：カーネル自身のコード。フォアグラウンド処理の時間は、基板で測定するまでは推定値です。

モーションとドライバ

各軸は、クローズドループのステッピングドライバとボールねじで駆動されます。ツインはこれらを 4段階の詳細度でモデル化しており、既定は 3質量モデルです。

- ドライバの位置ループ：指令位置のまわりのモーターの磁気ばね、速度とともに低下するトルク、負荷が大きすぎるときの脱調、追従誤差が大きくなりすぎたときのドライバのアラーム。
- ガイド、ナット、軸受の摩擦と、Z 軸にかかる重力。
- モーター、ねじ、テーブルの間の伝達系のコンプライアンス、ナットのバックラッシ、X 軸の下のフレーム。
- ドライバの電源：非常停止回路、コンタクタ、ドライバが立ち上がるまでの時間。

根拠：設計（質量、慣性モーメント、手計算と有限要素解析による剛性）、ドライバとモーターのデータシート。摩擦とドライバの内部パラメータは、測定するまでは仮定です。

幾何精度モデル

オンにすると、ツインは工具先端を機械の幾何誤差の分だけ動かします。対象は、各軸のリード誤差とその周期成分、2つの真直度誤差と、アッベアームを介して作用する 3つの角度誤差、軸間の直角度、ナットのバックラッシ、そして熱によるねじと主軸の伸びです。リミットスイッチはスライドを、ツールセッターとタッチプローブは工具先端を検知します。これは実機と同じです。このモデルは既定ではオフです。公称のツインは理想状態です。

根拠：部品の等級から導いた、組み立て済みの標準的な機械の表。これらは機械の仕様ではありません。コミッショニング時に、その機械の ISO 230-2 による測定値に置き換えます（計画中）。

熱

主軸とボールねじは、使うと温まり、休止すると冷えます。ツインはセッションをまたいでそれらの温度を保持し、その温度に応じて工具先端とねじのリードを動かします。

根拠：1次の熱モデル。パラメータは、機械のウォームアップを測定するまでは仮定です。

主軸とインバータ（VFD）

主軸のインバータは、機械のものと同じようにモデル化されています。コントローラは RS-485 で、インバータ固有のプロトコルを使い、設定された通信速度で 1バイトずつ通信します。インバータは、そのマニュアルに書かれているとおりに応答します。これには、通信を受け付けるかどうかを決めるパラメータも含まれます。その背後には、すべりから求まるモーターのトルク、電流、インバータの過電流・過負荷・不足電圧のトリップ、電源が一瞬落ち込んでもインバータを動作させ続ける DC バス、ウォーターポンプがあります。

根拠：インバータとモーターのデータ。バスの静電容量、電流の励磁成分、摩擦は、測定するまでは仮定です。

切削

素材は高さフィールドで、指定した場所には、より細かい領域が置かれます。工具が動くたびに、工具の形状が掃引した部分が除去されるので、部品の形状はプログラムそのものから決まります。

切削抵抗は、材料の切削係数から求めます。切削係数は、切りくずの厚さと切削速度によって変わり、工具の実際の形状（フラット、ボール、ブルノーズ、V）に対して、刃ごとに計算されます。この切削抵抗から、ツインは主軸の負荷と動力、工具のたわみ、工具が折れる危険性を推定します。折れる危険性は、刃溝の付け根の曲げ応力を超硬合金の強度と比べて求めます。

根拠：材料ごとの公表されている切削データ（各材料の出典は機械プロファイルにあります）、切れ味の鋭い工具、一般的な工具形状。

びびり

切削ごとに、ツインは、使用中の主軸回転数でびびりが始まる切込み深さを、工具と機械の振動モードおよび切削の形状から計算し、切削が安定する主軸回転数を選びます。また、切削の強制振動から、壁面がどの位置に仕上がるかを推定します。

工具先端には、各方向に三つのモードがあります。機械の構造、コレットと主軸に取り付けたツールホルダー、そして工具自身のたわみです。たわみモードの周波数は、工具の形状（刃、首、シャンク、突き出し長さ）から決まります。突き出し 20 mm の 6 mm エンドミルで約 14 kHz、0.2 mm の工具では 100 kHz を超えます。約 1 mm 未満の工具は、びびるとすれば自身のたわみでびびり、その深さは主軸回転数にほとんど依存しません。ツインはこの深さを、切削ごとに時間領域で計算します。

根拠：有限要素解析によるフレームのモード、仮定したホルダーのモード、工具の形状からはり理論で推定したたわみ。三つとも減衰は仮定値（3 %）です。まだどれも測定していません。組み立てた機械でのタップテストで測定する予定で、それまでは安定な深さは推定値です。切削自体の減衰はモデル化していません。小さな工具では、びびり指標は安全側になります。

振れ

軸が主軸からずれた工具は、最も外側の刃で切削します。そのため、穴は大きく、ボスは小さく仕上がります（所見 187）。ツインはその包絡で切削し、CAM ははめあいの計算でこれを見込みます。工具ごとに、その仕様に振れを持たせることもできます（精密コレットの等級、または測定値。`tir3`：TIR をマイクロメートルで）。指定がなければ機械の振れを使います。

根拠：機械で測定するまでは、仮定した振れ。

工具摩耗

1.0 では、すべての工具が新品です。どれだけ長く切削しても、切削する形状と切削抵抗は、切れ味の鋭い工具のものです。プログラムが工具ごとに使う寿命を数える機能は、後のバージョンで計画されています。アルミニウムを削る工具は、逃げ面が摩耗したからではなく、アルミニウムが刃に付着して、部品にバリが出たり仕上がりが悪くなったりするために、たいていは先に交換されます。

タッチプローブ

タッチプローブは、ボールが部品に触れるとプローブ入力を切り替える工具です。ツインは、素材の形状と切削された面（傾斜面を含む）から、接触を厳密に求めます。モデルは 2種類あります。ボールが触れた瞬間に切り替わる理想的なプローブと、方向によって変わるプリトラベル、3つのローブ、ばらつき、押し込みすぎたときのスタイラスの破損を備えた一般的なプローブです。

根拠：タッチトリガープローブの一般的な値。測定に使う前に、リングゲージでタッチプローブを校正してください。

Z クリアランス

テーブル上の段取りについて、ツインは、越える必要がある最も高い点を求めます。また、主軸の工具と工具テーブルの各工具について、Z 軸の上端での工具先端の高さを求めます。段取りを越えられない工具は、段取りまたは工具が変わったときに報告されます（所見 188）。

根拠：機械の設計と、指定された工具の長さ。

衝突とワーク保持

ツインは、工具、そのホルダー、主軸が、部品、治具、機械に当たらないかを調べ、衝突しそうになる前に警告します。ワーク保持（クランプ、バイス、V ブロック、テープ、固定された部品）は、一定の力で部品を保持するので、それより強く押す切削では、部品が滑ることがあります。

根拠：設計に基づく機械と治具の形状。保持力は推定値です。

モデル化していないもの

- 壁面の位置以外の、表面のテクスチャと粗さ。
- 構成刃先、切りくず、切りくずの排出。
- 切削中に薄い壁がたわむこと。
- 部品自体の発熱。
- 工具が折れたあとに起こること。

組み上がった機械が必要な受け入れ試験（ドライバと音の測定、切削抵抗、表面、熱、フレームのモード、実機とのシャドーモード）は、ツインの設計ノート（docs/06、第7節）にあります。

Twinwright 独自カーネル

Twinwright 独自カーネルは、Twinwright が自社で開発した CNC コントローラのファームウェアです。Studio では、ツインの機械を動かします。同じコードが、MC-1 のコントローラ基板用にもビルドされます。

Grbl 1.1 シリアルプロトコルで通信するため、G コード送信ソフトがそのまま使えます。設定、アラーム、エラーは、送信ソフトが知っている番号のままです。

機能

- **モーション：** 100ブロックにわたる先読み計画、`$11` から決まるコーナー速度、真円に沿った円弧。モーションは 1 ms のスライスで進みます。各軸は、最大速度（`$110`–`$112`）と加速度（`$120`–`$122`）を守ります。
- **機械の機能：**
  - 原点復帰、ハードリミットとソフトリミット、プロービング、ツールセッター
  - Modbus 経由の主軸インバータ、クーラント
  - `$341` の設定に従う工具交換
  - パーキング付きの安全扉
  - オーバーライド、ジョグ、ハンドル（手動パルス発生器）
- **コントローラ上のプログラム：** コントローラのファイルシステム上のプログラムは、O ワードのサブルーチンとマクロを使って実行されます（O ワードを参照）。
- **G コード：** RS274/NGC と LinuxCNC に対応しています。工具テーブル、工具径補正、パスブレンディング、式、パラメータを含みます（G コードと M コードを参照）。

シリアルプロトコル、リアルタイムコマンド、`$` コマンドは、プロトコルにあります。すべての設定は設定に、すべてのアラームはアラームコードに、すべてのエラーはエラーコードにあります。

ほかの Grbl コントローラとの違い

Grbl 向けに作られた送信ソフトは、このコントローラでも使えます。この節では、ほかの Grbl コントローラに慣れている場合に気づく点を挙げます。

コントローラが停止する場面

次の場合、一部の Grbl コントローラは動作を続けますが、Twinwright 独自カーネルは停止するか、アラームを出します。

- **主軸インバータが自分で停止した場合**（トリップ、過負荷、またはインバータ自身の停止キー）：送りは 0.2 s 以内に止まり、コントローラは `ALARM:14` を出します。
- **主軸インバータが応答しない場合：** まず軸が減速して停止するので、位置は保たれます。そのあと、コントローラは `ALARM:19` を出します。
- **工具交換：**
  - 工具交換のときは、主軸は必ず停止します。
  - 確認なしで工具交換が続行される場合、主軸とクーラントは元の状態に戻ります。
- **フィードホールドまたは扉のあとの再開：** 主軸が回転数に達してから、工具が動き出します。
- **機械がパーキングから戻る途中で、扉がもう一度開いた場合：** 機械は停止し、もう一度パーキングの高さまで上がります。次のサイクルスタートを待ちます。
- **送信ソフトとの接続が切れた場合**（USB が抜かれた、ポートが閉じられた）：ジョブが停止します。これは `$1030` で制御され、既定ではオンです。
- **プローブスイッチと原点復帰スイッチ：** 配線上のグリッチは、接触として扱われません。
- **プログラム中の UTF-8 テキスト**（中国語、「Ø」、「–」）は、テキストとして読まれ、リアルタイムコマンドとして解釈されることはありません。
- **平面内に弦がない R 形式の円弧**、たとえば `G91 G2 X0 Z1 R5` は、`error:33` を返します。

より多くの G コード

一部の Grbl コントローラは、次のものを拒否するか無視します。Twinwright 独自カーネルは、規格の定義どおりに実行します。

- 工具テーブル（`G43 Hn`、`G43.2`、`G10 L1/L10/L11`）
- 工具径補正（`G41`、`G42`）
- パスブレンディング（`G64 P`）と正確停止（`G61.1`）
- O ワードのサブルーチンと制御フロー、`#` パラメータ、式
- `G52`、極座標（`@`、`^`）、`M99`
- `G85`、`G86`、`G89` の穴あけとボーリング（一部の Grbl コントローラはこれらを受け付けますが、動きません）
- CAM のポストプロセッサがよく出力するもの：名前付きの `%` 行、`G49 H0`、Fanuc のプログラム番号行（`O1000`）、Centroid の `M25`

プログラムにより忠実な動作

- **円弧** は、真円をたどります。一部の Grbl コントローラは円弧を弦に分割し、円からのずれは最大 4 × `$12` になります。
- **逆時間送りの円弧**（`G93`）は、プログラムされた時間をかけて動きます。一部の Grbl コントローラは、これを数倍の速さで実行します。
- **ドウェル**（`G4`）は、プログラムされた長さどおりに続きます。一部の Grbl コントローラは、50 ms 刻みに丸めます。
- **ステップ送り穴あけ：** 穴ごとに、必要な分のステップだけを行います。深さがステップ量のちょうど整数倍のとき、一部の Grbl コントローラは、リトラクト高さからもう 1ステップ分を丸ごと穴あけします。
- **ソフトリミット付近の大きな円弧** で、誤ったアラームは出ません。
- `M72` は、`M70` または `M73` で保存した工具長補正を元に戻します。保存した値が 0 で、補正モード（`G43.1`）が同じ場合、一部の Grbl コントローラは元に戻さず、後で設定した値を保持します。そのため `M72` の後、同じプログラムでもそのようなコントローラでは工具長補正が異なることがあります。
- **加速度：** すべての軸は、小さな円弧でも `$120`–`$122` を守ります。小さな円弧が非常に多いプログラムは、一部の Grbl コントローラより最大で約 2 % 長くかかることがあります。通常のプログラムは、約 0.1 % 短くなります。

プロトコルの詳細

- **ステータスレポート：**
  - フィードホールド中、または扉が開いているときは、`FS:` の送り速度は 0 です。軸は止まっています。
  - 残りの距離は、軸が止まるまで報告されます。

    - 重大なアラームの間、`Bf:` はプランナが空であることを示します。
    - `$10` でパーサの状態をオンにしている場合、`[GC:]` は変化ごとに 1回出力されます。
- **入力チェック：**
    - `$45x` のような不正な設定番号は、`error:3` を返します。
    - `$1=25.0` のように整数の設定を書くと、25 が保存されます。一部の GrbI コントローラでは、250 が保存されます。
    - `M6` と移動を 1行に書くと、`error:40` を返します。
    - `G65` または `G66` を、ほかのコマンドと 1行に書くと、`error:21` を返します。
- **アラームと起動：**
    - アラームのあとに 100 ms の待ちはありません。
    - 設定の保存領域が空の状態での最初の起動では、`error:7` を返しません。
- **テキストと設定：**
    - 設定、アラーム、エラーの説明文は、Twinwright 独自のものです。
    - コントローラには独自の設定があります。`$1030`（送信ソフトとの接続が切れたときに停止する）と、入力シェーピング用の `$1000` 以降（既定ではオフ）です。
- `$39=0`（従来のリアルタイム文字をオフ）：`?`、`!`、`~` は、`$` 行とコメントの中ではテキストで、それ以外の場所ではコマンドです。コントローラは、バイトが届くたびにその場で判断します。そのため、扉が開いているときやフィードホールド中でも `?` に応答します。一部の Grbl コントローラでは、応答しないことがあります。

非対応

- `G65 P7`（Modbus メッセージ）。
- Fanuc のシステム変数。`#4120` などは通常のパラメータとして扱われ、値は 0 です。
- 主軸エンコーダのないねじ切り（`G33`、`G76`、`G33.1`）。MC-1 にはエンコーダがないため、これらは `error:20` を返します。
- 旋盤の荒削りサイクル（ほかの Grbl コントローラにもありません）。
- ネットワーク機能、Trinamic ドライバ、ほかのメーカーの主軸インバータ（一部の Grbl コントローラがプラグインとして提供しています）。

送信ソフト

動作を確認した送信ソフトとその設定は、G コード送信ソフトにあります。実機とそのツインを比較するには、実機とシャドーモードを参照してください。

G コードと M コード

Twinwright 独自カーネルは、以下の G コードを実行します。コードは RS274/NGC と LinuxCNC に従います。コントローラが対応していないコードには `error:20` を返し、その行では何も起こりません。

リアルタイムコマンド（`?`、`!`、`~` など、1バイトで構成されるもの）については、プロトコルを参照してください。ほかの Grbl コントローラとの違いについては、Twinwright 独自カーネルを参照してください。

G コード

コード	意味	備考
`G0`、`G1`	早送り、直線補間	
`G2`、`G3`	円弧（時計回り、反時計回り。I J K または R 指定）、ヘリカル	工具は真円に沿って動きます。その平面内に弦がない R 指定の円弧には `error:33` を返します。
`G4 P`	ドウェル、P 秒	ドウェルはプログラムした時間のとおりに続きます。
`G5`、`G5.1`	3次スプライン、2次スプライン	I と J のない `G5` は、直前の移動が終わった方向に続きます。
`G7`、`G8`	旋盤の直径モード、半径モード	旋盤モードのみ（`$32=2`）。
`G10 L2`、`G10 L20`	ワーク座標オフセットの設定	
`G10 L1`、`G10 L10`、`G10 L11`	工具テーブルの設定	工具 32本。コントローラの電源を切っても、テーブルは保持されます。
`G17`、`G18`、`G19`	平面の選択	
`G20`、`G21`	インチ単位、ミリメートル単位	
`G28`、`G30`	中間点を経由して、記憶した位置へ移動	
`G28.1`、`G30.1`	現在位置の記憶	
`G33`、`G33.1`、`G76`	主軸同期移動、リジッドタッピング、ねじ切りサイクル	主軸エンコーダが必要です。MC-1 にはないため、`error:20` を返します。
`G38.2`–`G38.5`	プローブ	結果は、プローブが作動した点で、`[PRB:]` として報告されます。ゼロの設定には、軸が止まった位置ではなく、この点を使ってください。
`G40`	工具径補正の解除	
`G41`、`G42`、`G41.1`、`G42.1`	工具径補正：D は工具テーブルから、`.1` では直径を指定	G17 平面のみ。補正の開始と終了の移動は LinuxCNC と同じです。削り過ぎになる移動は拒否されます。
`G43 Hn`、`G43.2 Hn`	工具テーブルからの工具長補正、追加の補正	
`G43.1`	動的工具長補正	
`G49`	工具長補正の解除	
`G50`、`G51`	スケーリングのオフ、オン	
`G52`	ローカル座標系	オフセットを `G92` と共有します（LinuxCNC と同じ）。
`G53`	この行では機械座標を使う	
`G54`–`G59`、`G59.1`–`G59.3`	ワーク座標系	`$486` で `G59.1`–`G59.3` をロックできます。
`G61`	正確パス	
`G61.1`	正確停止	プログラムした移動の終わりごとに、軸が停止します。
`G64 P Q`	パスのブレンド	時間を短縮できる場合は、2つの直線移動の間の角が、角から P 以内の円弧になります。Q は受け付けますが、使われません。
`G65 P1`–`P6`	組み込みマクロ	設定、工具オフセット、パラメータ、機械の状態、プローブの選択、主軸の待機。`P7`（Modbus メッセージ）には対応していません。
`G65 P100` 以降、`G66`、`G66.1`、`G67`	ファイルマクロ、モーダルマクロ呼び出し	P は整数でなければなりません。同じ行に他のコマンドを置くことはできません（`error:21`）。
`G73`、`G81`、`G82`、`G83`	穴あけサイクル、ステップ送り穴あけ	各穴で、必要な回数のステップ送りを行います。
`G84`、`G85`、`G86`、`G89`	タッピングサイクルとボーリングサイクル	
`G80`	サイクルの解除	
`G90`、`G91`	絶対指令、増分指令	
`G91.1`	円弧中心の増分指令	
`G92`、`G92.1`、`G92.2`、`G92.3`	座標オフセット：設定、クリア、保留、復元	
`G93`、`G94`、`G95`	逆時間送り、毎分送り、毎回転送り	逆時間送りの円弧は、プログラムした時間をかけて動きます。
`G96`、`G97`	周速一定、主軸回転数一定	旋盤モードのみ。
`G98`、`G99`	サイクルのリトラクト先：開始高さ、R 平面	

次の G コードには対応していません（`error:20`）。

- `G5.2`、`G5.3`（NURBS）
- `G17.1`–`G19.1`（UVW 平面）
- `G70`–`G72`（旋盤サイクル）
- `G90.1`

M コード

コード	意味	備考
`M0`、`M1`	プログラムの一時停止、オプショナルストップ	リアルタイムコマンド `0x88` は、`M1` による一時停止のオンとオフを切り替えます。
`M2`、`M30`	プログラム終了	
`M3`、`M4`、`M5`	主軸：時計回り、反時計回り、停止	
`M6`	工具交換	`$341` で M6 の動作を設定します。M6 と移動を同じ行に書くと `error:40` を返します。
`M7`、`M8`、`M9`	クーラント：ミスト、フラッド、オフ	MC-1 では、M8 が主軸の水冷を動かします。
`M25`	Z を原点復帰した高さまで上げる	Centroid のコードで、一部の CAM ポストプロセッサが出力します。
`M48`、`M49`	オーバーライドのオン、オフ	
`M50`–`M53`	送りオーバーライド、主軸オーバーライド、アダプティブ送り、フィードホールド：オンまたはオフ	
`M56`	パーキングのオンまたはオフ	
`M60`	パレット交換による一時停止	
`M61`	工具交換をせずに現在の工具を設定	
`M62`–`M65`	デジタル出力のオンまたはオフ（次の移動に合わせる、または即時）	補助出力のあるボードで使えます。
`M66`	入力の待機	待機は、それより前の移動が終わった時点から始まります。
`M70`–`M73`	モーダル状態の保存と復元	
`M98 P L`、`M99`	サブルーチンの呼び出し、復帰	コントローラのファイルシステムから実行するプログラムで使えます。1行ずつ送った `M98` には `error:39` を返します。メインプログラム内の `M99` は、そのプログラムを終了させます。

次の M コードには対応していません（`error:20`）。

- `M19`（主軸のオリエンテーション）
- `M67`、`M68`（アナログ出力。MC-1 にはありません）
- `M100`–`M199`（ユーザー M コード）

ワード、構文、パラメータ

行の構造

- コメント `( … )` と `;` が使えます。
- `(MSG,…)` は、そのテキストをメッセージとして送信ソフトに送ります。
- `(PRINT,…)` は、指定したパラメータの値を含むメッセージを送ります。
- `(DEBUG,…)` は、`#5599` がオンの間、同じことを行います。
- ブロックスキップ `/`、行番号 `N`、`%` の行（名前の有無を問わず）が使えます。
- 大文字と小文字は区別されず、ワード内のスペースは無視されます。

パラメータと式

- 番号付きパラメータ `#1`–`#5000` は、プログラム自身のものです。プログラムが値を設定するまで、読み取った値は 0 です。
- `#5001` 以降のパラメータと、名前付きのシステムパラメータは、読み取りだけができます。
- 名前付きパラメータ `#<name>`：一度も設定されていない名前を読み取ると、`error:39` を返します。設定済みかどうかは `EXISTS[#<name>]` で調べられます。
- `[ ]` 内の式では、LinuxCNC の演算子と関数が使えます。角度の単位は度です。
- 行で設定した値を、同じ行の中で読み取ることはできません。

O ワード

- コントローラのファイルシステムから実行するプログラムでは、サブルーチン、`if`、`elseif`、`else`、`while`、`do`、`repeat` が使えます。
- 1行ずつ送られたサブルーチンとループは、いったん集められ、完結してから実行されます。
- サブルーチンは、独立したファイル（`<name>.ngc` または `<name>.macro`）にすることもできます。

極座標

`@` は半径、`^` は角度で、XY 平面内のワーク原点を中心にします。

行のエラー

- 同じモーダルグループのコードを 1行に 2つ書くと、`error:21` を返します。

- どちらも軸ワードを使う 2つのコマンドを 1行に書くと、`error:24` を返します。
- エラーコードの全一覧は、エラーコードにあります。

CAM の加工方式とパラメータ

各加工方式が何をするか、そのパラメータが何を意味するか、そしてツインが切削条件をどう決めるかを説明します。ツインが示す数値は、物理モデルによる推定値です。

以下は、ジョブの中でフィーチャが取るパラメータです（`program_generate` ツールのフィーチャ）。長さの単位はミリメートル、角度の単位は度です。

ポケットの削り取り：リング、ジグザグ、アダプティブ

ポケットの領域とは、その形状からアイランドを除き、工具の半径の分だけ内側にオフセットしたものです。領域は層ごとに削り取られ、各層の深さは切込み深さ（`ap`）以下です。層の中での経路は、加工方式（`strategy`）で決まります。

加工方式	経路	用途
`rings`（既定）	領域のオフセットを、最も内側から壁まで `ae` 間隔で並べたもの	ほとんどのポケット
`zigzag`	`angle` の向きに往復する直線パス。間隔は `ae`	開けた浅い領域
`adaptive`	工具の噛み込みをどこでも `ae` 付近に保つ経路	深いポケット、硬い材料

はめあいのあるポケットは、壁を専用のパスで寸法どおりに仕上げます。はめあいを参照してください。

アダプティブ加工は、小さな径方向の噛み込みで、より深い `ap` で切削します。リンクは材料から離れた位置で移動し、領域が狭すぎてアダプティブでは削れない箇所は、リングに切り替えます。

残り加工

`rest: true` を指定すると、ポケットは、前の工具が削り残した部分だけを削り取ります。大きな工具が届かなかった隅や狭い部分です。削り残しは、Studio が同じプログラム内の前のポケットから求めます。ツインにワークがあるときは、ツインのワークから求めます。

輪郭とタブ

輪郭（`side`：`outside`、`inside`、`on`）は、外形を 1層ずつたどり、ダウンカットで切削します。`outside` と `inside` は外形を工具の半径の分だけオフセットし、`on` は外形をそのままたどります。

タブ（`tabs`：数、幅、高さ）は、底に短いつなぎ部を残し、切り抜いた部品がその場に留まるようにします。タブの高さは、指定しない限り 1 mm です。切削の深さが 1 mm 以下の場合は、深さの半分の高さになります。

スロット：層ごとの切削とトロコイダル

スロット（`x0`、`y0`、`x1`、`y1`、`width`）の加工方式は、次のとおりです。

- `layers`（既定）：スロットの中央を工具の全幅で層ごとに削り、そのあと両側へスロットの幅まで広げます。
- `trochoidal`：スロットに沿った小さなループを、`step` ずつ先へ進めながら描きます。工具が噛み込むのは短い円弧の範囲だけなので、1回のパスでより深く削れます。スロットの幅は工具より広くなければなりません。

工具が材料に入る方法：ヘリカル、ランプ、プランジ

エンドミルは、材料が詰まった部分へまっすぐ突っ込むことはしません。各層は、次のいずれかで始まります。

- 領域に余地があるときは、ヘリカル（直径と角度は、応答の `entry` に示されます）
- 余地がないときは、経路に沿ったランプ
- まっすぐなプランジは、穴の中か、素材の外だけ

応答には、各加工工程について、最初にどのように進入したかが示されます。

穴

穴あけサイクル

`points` の位置にある、深さ `depth` の穴は、コントローラのサイクルで穴あけします。ドリル径の 3倍より深い穴は、ステップ送りで穴あけします。1回のステップ送りは、`peck` で別の値を指定しない限り、径の 1.5倍です。サイクルのリトラクト平面は、上面の 1 mm 上です。

ヘリカル補間によるボーリング

エンドミルより大きい直径 `d` の穴は、ヘリカル経路で切削し、底で寸法どおりに仕上げます。穴の径は、工具径の最大 2倍までです。それより大きい穴は、ポケットとして加工します（`shape: circle`）。

リーマ加工

穴のはめあいに合ったリーマ（`reamer 6 H7`）は、その送り速度で穴に入り、同じ送り速度で出ます。穴から早送りで抜くことはしません。穴は、同じジョブの中でリーマより前に、0.05–0.6 mm 小さく穴あけしておくか、`predrilled` と指定しておく必要があります。リーマの切削速度は、その材料でのエンドミルの切削速度の 4分の1、1回転あたりの送りはドリルの 2.5倍です。

スレッドミルによるねじ切り（管用テーパねじを含む）

スレッドミルは穴の中を底まで下がり、続いて底から上へ向かって、1周で 1ピッチ進むヘリカル経路でねじを切ります。右ねじは反時計回り（ダウンカット）、左ねじは時計回りです。ねじは、呼び（`thread`：`M8`、`M8x1`、`1/4-20 UNC`、`G1/4`）または `d` と `pitch` で指定します。管用テーパねじ（`1/4 NPT`、`Rc1/4`）は、円すいに沿って切削し、1周は 48個の円弧で構成します。

面取り、彫刻、V彫刻

- 面取り：Vビットが外形のエッジに沿って、`width` になる深さで動きます。幅は、円すい部で制限されます。
- 彫刻：工具は、線、またはシングルストロークフォントかアウトラインフォントの文字に沿って、一定の `depth` で動きます。
- V彫刻：Vビットは、形状の中軸（メディアル軸）に沿って動きます。各点での深さがその場所の幅を決めるため、形状の角はシャープに仕上がります。`max_depth` は、広い領域をその深さで平らに削ります。

3D の荒加工と仕上げ加工

メッシュは、レベルごとに荒加工します（`rough3d`）。各レベルはアダプティブに削り取ります。そのあと、次の加工方式で仕上げます（`finish3d`）。

加工方式	経路
`parallel`	部品全体を一方向に走るパス
`waterline`	各レベルの等高線。急な壁向け
`steep_shallow`（既定）	急な領域は等高線、緩い領域は走査線
`pencil`	内側の隅に沿う。ほかのパスのあとで行う

`rest_from` は、前の工具が削り残した部分だけを仕上げます。

ツインが決める切削条件

Studio は、工具の種類に対する材料のデータから出発します。切削速度の範囲と、1刃あたりの送りです。`cutting`（`rpm`、`feed`、`ap`、`ae`、`finish_ap`）に指定した値は、指定したとおりに保持されます。それ以外の値は、ツインのモデルで決めます。

安定性とびびり

ツインのびびりモデルは、ホルダーに取り付けた工具の安定限界線図を計算します。主軸の各回転数で安定を保てる切込み深さです。ローブは工具と機械のモデルが示す位置にあるにすぎないため、Studio はローブの頂点を狙いません。各回転数について、その前後 10 % の範囲で最も浅い安定切込み深さを取り、材料が示す回転数の 0.7〜1.8 倍の範囲で回転数を選び、その深さの 80 % 以下で切削します。

出力、切削抵抗、工具のたわみ

- 出力：切削が使うのは、その回転数での主軸出力の、最大でおよそ 80 % です。
- 切削抵抗：切削抵抗は、治具が横方向に保持できる力の半分未満に抑えられます。
- たわみ：仕上げパスは、ツインが推定する工具のたわみが小さく収まるよう、軽く抑えられます。
- 隅と円弧：円弧での送り速度は、機械の持つ加速度で制限されます。これは、コントローラが、切削を計画した送り速度より下まで減速することがないようにするためです。仕上げパスは、工具の噛み込みが増えるポケットの内側の隅で減速します。

クローズドループ：ツインでのプログラムの試運転

検証をオンにすると、Studio は新しいツインでプログラムを実行します。実行でびびり、過大な負荷、工具の大きすぎるたわみが見つかると、Studio はプログラムを変更して、もう一度試します。

- より浅い層
- 設定した回転数の 0.7〜1.8 倍の範囲で、ローブが多少ずれていても切削が安定する別の主軸回転数
- はめあいのある壁へのスプリングパス

試した各バージョンは保存されます。長いプログラムは、呼び出しの裏で試され、その報告は後から届きます（`run_status`）。応答には、何を変更したか、その理由が示されます。

複数の段取りとネスティング

- 段取り：Studio は各段取りのプログラムを書き出し、そのあと段取りを順にツインで確認します。各段取りは、それ以前の段取りが残したワークの上で確認します。そのワークが素材と異なる場合は、そのワークに合わせて、段取りのプログラムを書き直します。
- ネスティング：部品は、外形をもとに、余白を空けて板材の上に配置します。各部品はフィーチャを保ち、フィーチャは部品と一緒に回転・移動します。

制限事項

- 3軸です。4軸目も、工具の傾斜もありません。
- STEP ファイルは読み込めません。CAD ソフトから STL または 3MF をエクスポートしてください。
- 切削条件とツインのチェックは、機械、工具、材料の物理モデルによる推定値です。モデルと異なる工具や材料では、切れ方が変わります。
- エディション：アダプティブ加工、`parallel` 以外の 3D 仕上げ方式、残り加工、複数の段取りは、Maker エディション以上に含まれます。ネスティングとプログラムオプティマイザは、Pro エディションに含まれます。

ジョブの大きさの上限

この上限を超えると、Studio はプログラムを書き出しません。理由と、何を変更すればよいかを示します。

項目	上限
ジョブ内の段取り	8
ネスティング	部品 16種類、各部品 200個
メッシュ（STL、3MF、OBJ、glTF）	三角形 500万個。ファイルは 1 GB まで
SVG、DXF、テキストファイル	64 MB。DXF 図面は 200万セグメントまで
アダプティブ加工	経路探索のグリッドに入るのは、約 150万セルまでです。MC-1 の移動範囲いっぱいの大きさのポケットは、3 mm のエンドミルなら削り取れます。1 mm のエンドミルでは、リングかジグザグを使うか、ステップオーバーを大きくしてください。

1つのジョブには、フィーチャを最大 32000個まで入れられます。同種の穴は、位置をまとめて 1つのフィーチャとして扱います。加工工程の一覧、その理由、クローズドループによる修正の対象は、最初の 64個のフィーチャです。それ以降のフィーチャでびびりが出ても、報告されるだけで、修正はされません。段取りシートには、最初の 64個の加工工程が載り、そのことも書かれます。

プログラムのチェックと段取りシートが読み込めるのは、移動 200万回分までです。円弧は、短い弦に分けて数えます。半径 3 mm の全円で約 50本、半径 50 mm の全円で約 220本です。上限を超えた場合は、どこで止まったかが示されます。段取りシートには、その場合でもプログラム全体の時間は載りますが、その行より後の加工工程ごとの時間と、工具の高さは載りません。ツインでの試運転は、プログラム全体をチェックします。

Studio がプログラムを書き出すのにかかる時間を、デスクトップ PC の 2コアで測定した結果です（ツインでの試運転は、その後になります）。

ジョブ	時間	メモリ
180 x 130 mm の板材に、アイランド付きのポケット 20個	1 s 未満	18 MB
移動範囲いっぱいの大きさのポケット、アダプティブ、6 mm のエンドミル	12 s	47 MB
同じものを 3 mm のエンドミルで	37 s	70 MB
1000文字の彫刻	12 s	25 MB
ワッシャー 200個を板材にネスティング	5 s	30 MB
移動範囲いっぱいの大きさのレリーフ（三角形 50万個）の 3D 荒加工、6 mm のエンドミル	9 min	240 MB
同じレリーフを三角形 300万個から	20 min	1 GB
そのレリーフの 3D 走査線仕上げ、2 mm のボールエンドミル、カスプ高さ 0.01 mm（三角形 50万個）	6 min	270 MB
同じものを三角形 300万個から	18 min	1 GB

3D 加工について：

- メッシュは、部品に必要な細かさにとどめてください。仕上げにかかる時間は、メッシュの三角形が増えた割合とほぼ同じだけ、長くなります。
- 緩やかにうねった面は、`waterline` ではなく `parallel` か `steep_shallow` で仕上げてください。そのような面では、等高線は小さなループを数多く作ります。上のレリーフでは、材料の中よりも空中での移動のほうが多くなり、そのプログラムは 280万行になりました。

はめあい、振れ、検査方法

Studio が寸法をはめあいに合わせて切削する方法、工具の振れを見込む方法、タッチプローブで部品を測定する方法を説明します。ここでのツインの数値は、物理モデルによる推定値です。Studio は、機械の位置決め精度を主張しません。

はめあい（ISO 286）

寸法には、はめあいを指定できます。穴の直径（`"d": "Ø12 H7"`）、軸の直径（外側の輪郭に付ける `"fit": "g6"`）、スロットの幅、長方形の X と Y の寸法です。Studio は、その寸法に対する ISO 286-2 の表から、はめあいの寸法許容差を取ります。または、書かれたとおりの寸法許容差を使います（`"tol": "+0.015/0"`、単位はミリメートル）。

公差域の中央に合わせた切削

壁は、仕上げ代を残して荒加工し、そのあと専用のパスで、はめあいの公差域の中央まで仕上げます。公差等級 IT6 とそれより精密な等級のはめあいには、スプリングパスが入ります。これは、仕上げの周回を寸法どおりにもう一度行い、工具のたわみで残った分を取り除くものです。クローズドループは、ツインのたわみの推定値が求める箇所に、スプリングパスを追加します。

応答には、各はめあいについて、公差域、切削した直径、その理由が示されます。

よく使うはめあい

名前	組み合わせ	用途
`sliding`	H7/g6	回転または摺動し、すきまは小さい
`locating`	H7/h6	すきまばめの中で最もきついもの：手で組み立てられる
`transition`	H7/k6	わずかなすきま、またはわずかなしめしろ：位置決め、軽い圧入
`press`	H7/p6	しめしろ：圧入

組み合わせは、両方の部品を指定します。前の記号が穴、後の記号が軸です。

工具の振れ

振れのある工具は、自身の径より大きな穴を削ります。ツインは、ホルダーに取り付けた工具の振れを、エンベロープとしてモデル化します。はめあいのある寸法では、Studio は、そのエンベロープの分だけ、穴では内側へ、軸では外側へ、仕上げ経路を動かします。こうして、ツインが推定する寸法は、引き続き公差域の中央になります。リーマ加工した穴は変更しません。穴の寸法はリーマの寸法だからです。

エンベロープは、ツインが持つ工具とホルダーのモデルに対する推定値です。最初の部品を測定してください。

タッチプローブによる測定

接触とその補正

各接触は、一連のプローブ移動（G38）です。

1. 面の近くまで、保護移動で進みます。途中でプローブが作動すると、停止します。
2. 面まで、高速でプローブします。
3. プローブが復帰するまで戻り、さらに少しだけ戻ります。
4. 面まで、低速でプローブします。使うのは、この動作の作動点だけです。
5. 開始した位置まで戻ります。

作動点はボールの中心です。Studio は、接触の方向でのボールの実効半径を加えます。実効半径とプローブのロービングは、直径が既知のリングゲージでの校正から得ます。

穴、ボス、壁、底面

- 穴とボス：1つの深さで、壁の周りを 4回または 8回接触します。Studio は、ボールの中心点に最小二乗法で円を当てはめ、そのあと実効半径を加えます。
- 壁：壁に直角な接触を、壁に沿った複数の点で行います。
- 底面と上面：真下に向かう接触です。
- 位置は、ワーク原点を基準にした、プログラムの座標で示されます。

普通公差（ISO 2768-1）

個別の公差がない寸法には、指定された等級（f、m、c、v）の普通公差が適用されます。値は、その寸法区分に対する ISO 2768-1 の表から取ります。対象は、長さの寸法と、面取りの高さと角度です。

ツインのグリッドが公差に対して粗いときは、その検査でのツインの推定値は、粗すぎて判定できないものとされます。測定値は、それでも判定されます。

形状チェック（ISO 2768-2、第一段階）

ISO 2768-1 の等級のあとに付ける H、K、L の等級（たとえば「mK」）は、形状チェックを追加します。判定には、ISO 2768-2 の表を使います。

チェック	接触	表
平面度	底面の 3 × 3 のグリッド。点の、その平面に対するばらつき	1

チェック	接触	表
真直度	長い壁に沿った 3〜8点。点の、その直線に対するばらつき	1
直角度	壁の 2つ目の深さ。壁の高さにわたる傾き（機械の Z に対する）	2
真円度	穴またはボスの 8点。半径のばらつき	4（直径の公差）

壁への 2回の接触は、互いに十分離れているときにだけ、区別できます。それができないほど短い壁（たとえば、縁の面取りの下で短くなった壁）は、値の代わりに理由とともに一覧に載ります。

実効半径だけで校正したプローブには、固有のロービングがあります。それは真円度に現れます。レポートには、ツインのプローブモデルがそれをどの程度の大きさと見なしているかが示されます。

面取りと V溝

- 面取りは、斜面に沿って 2回接触します。外形からの位置は、面取り幅の 0.3 と 0.7 です。2つのボール中心を通る直線は斜面と平行になり、そこから面取りの幅、深さ、角度が得られます。
- V溝（彫刻または V彫刻）は、両側の斜面にそれぞれ 2回接触します。2本の直線から、溝の先端の深さ、上端の幅、開き角、そして溝を切った線からのオフセットが得られます。
- ねじは測定しません。ねじ山の頂はまっすぐな穴の面になり、ツインのプローブが触れるのはセルの頂面だけです。応答にもその旨が示されます。

ツインの推定値の意味

ツインは、シミュレーション上のワークを同じプログラムで削り、そのワークに、同じプローブモデルで同じプロービングを実行します。推定値は、その計算の結果です。推定値には、各寸法に対するプログラム自体の影響が、ツインがモデル化している範囲で表れます。工具のたわみ、振れ、切削抵抗です。

推定値は、機械の測定の不確かさではありません。レポートは、機械がどれだけ精密に測定や位置決めを行うかについて、何も述べていません。

制限事項

- プローブが測定するのは、触れたものだけです。ボールより狭いフィーチャや、ボールを直角に当てられない面は、測定されません。応答に理由が示されます。
- ツインのワークはグリッドです。推定値の細かさは、そのグリッドの細かさが限度です。公差に対してグリッドが粗すぎる場合、ツインの推定値は判定されません。
- 形状チェックは ISO 2768-2 の第一段階で、平面度、真直度、直角度、真円度を扱います。対称度と平行度は、まだチェックしていません。

機械プロファイル

草稿（ラウンド 14）：本文は執筆中で、節が増えることがあります。

機械プロファイルとは

機械プロファイルは、ツインが機械を組み立てる元になるファイルです。機械のモデルごとに 1つのファイルがあり、形式は JSON です。機械の形状と軸、ドライバとボールねじ、主軸とそのインバータ、センサとその配線、ワーク保持、材料ごとの切削データ、ツールセッターとタッチプローブ、音、コントローラのビルド内容が含まれます。コントローラの設定は、設定ファイルとして付属します。

プロファイルには、各値の出どころ（由来）が記録されます。設計、データシート、公表された資料、仮定、測定のいずれかです。

出どころ

MC-1 のプロファイルは、機械のパラメトリック設計から生成されます。図面と部品表を作るのと同じパラメータからプロファイルも作られるため、ツインと機械の設計が食い違うことはありません。プロファイルは、元になった設計ファイルのチェックサムを保持します。スキーマがその形式を検査します。リファレンスの章機械プロファイルのスキーマに、すべてのキーが載っています。

お使いの機械の値

それぞれの MC-1 は、コミッショニング時に測定し、そのツインを測定値に合わせます（計画中。手順とそのツールは作成済みで、ツイン上でテストされています）。

- リミットスイッチ：作動する位置、ヒステリシス、繰り返し精度。
- 各軸の ISO 230-2 による測定：バックラッシ、リード誤差の表、1 mm あたりのステップ数。
- 試験用の溝加工による切削係数。
- コーストダウンによる主軸の慣性モーメントと摩擦。
- リングゲージによるタッチプローブの測定。

測定値は、日付、測定に使ったツール、測定方法とともに記録されます。プロファイルはその値で作り直され、各値には測定値であることが記され、設計の値は履歴に残ります。測定値によって変わるコントローラの設定は、機械に送られます。

複数の機械

Studio は、同じモデルの機械を複数台保持できます。それぞれの機械は、セッションをまたいで自分のワールド（機械の位置、ワーク、工具、コントローラのメモリ）と、自分のウィンドウレイアウトを保持します。機械モデルのプロファイルは、その機械が測定されるまでは、すべての機械で共通です。

リファレンス

プロファイルのすべてのキーと、その型、単位、MC-1 での値は、機械プロファイルのスキーマにあります。

プレビュー：このページの安全上の注意は、まだネイティブの校閲を受けていません。

MC-1 のハードウェア

ドラフト（ラウンド 14）：本文は執筆中で、節が増えることがあります。

この章では、設計どおりの MC-1 を扱います。章内の数値は設計値です。組み立てた機械は、それぞれ立ち上げ時に測定し、そのツインを測定した値に合わせます（機械プロファイルを参照）。

機械

MC-1 は、アルミニウム、真鍮、エンジニアリングプラスチックを加工する、卓上の 3軸フライス盤です。ガントリーは固定で、テーブルが Y 方向、ガントリーのキャリッジが X 方向、主軸が Z 方向に動きます。

移動範囲 X / Y / Z	200 / 150 / 80 mm
テーブル	300 × 220 × 35 mm、25 mm ピッチのグリッド上に M6 のタップ穴
寸法（幅 × 奥行き × 高さ）	約 508 × 490 × 564 mm（モーター、ケーブルチェーン、レベリングフットを含み、制御盤は含まない）
質量	約 57 kg（制御盤を除く）
フレーム	6061-T6 アルミニウム板、ボルト締め
リニアガイド	HIWIN HGR15 レール（6本）、HGW15CC フランジブロック（12個）
ボールねじ	SFU1204（12 mm、リード 4 mm）、C7、BK10 / BF10 サポート
モーター	NEMA 23 クローズドループ式ステッピングモーター、1.2–1.3 N·m（3台）、ジョーカップリング付き
主軸	800 W、水冷、Ø65 × 195 mm、ER11 コレット
制御盤	機械とは別のボックスで、機械の右側に置く

軸とドライバ

各軸は、クローズドループ式のステッピングモーターが、ジョーカップリングを介してボールねじを直接駆動します。ドライバの異常は、1本の線で報告されます。3つのドライバのアラーム出力が、コントローラのモーター異常入力に直列に配線されているため、異常を起こしたドライバ、または電源を失ったドライバがあると、機械はアラームを出して停止します。

設計上の最大移動速度は、X と Y が 3000 mm/min、Z が 1500 mm/min です。

リミットスイッチと原点復帰

移動範囲の端は、M8 の誘導形近接スイッチ 5個で示します。X と Y は、それぞれ両端に 1個ずつ（それぞれ専用の入力）、Z は Z スライドにある 1個で Z の両端を示します。スイッチはノーマルクローズです。断線やプラグの緩みは、作動したスイッチとして読み取られます。そのためコントローラは、端のストッパーへ向かって動くのではなく、アラームを出して原点復帰を拒否します。

原点復帰は、Z を最初に、そのあと X と Y を、それぞれ正の端（X+、Y+、Z+）へ向かって行います。電源を入れたあとは、機械の原点復帰が必要です。

Z 軸と重力

Z スライドは主軸を載せています（約 6.35 kg）。ドライバに電源が入っている間は、ドライバが Z を保持します。電源が入っていないとき（非常停止、ドライバの異常、停電のあと）は、主軸を載せた Z が下がることがあります。

> ⚠ **警告**
>
> 非常停止またはドライバの異常のあとは、作業を続ける前に、もう一度機械の原点復帰を行ってください。ドライバの電源が切れている間は、主軸の下に手を入れないでください。

立ち上げには、ドライバの電源を切った状態での試験が含まれます。Z が下がる場合は、ブレーキ付きの Z モーターか、カウンターウェイトを機械に取り付けます。

主軸

主軸は、水冷の 800 W モーター主軸で、Ø65 × 195 mm、ER11 コレット付き、2極、最高 24 000 min^{-1} です。Huanyang HY01D523B のインバータ（VFD、単相 220 V 入力）が駆動します。コントローラは、絶縁モジュールを介した RS-485 で、回転数を設定し、状態を読み取ります。水ポンプで主軸を冷却します。

- インバータの異常リレーも、モーター異常の線に配線されています。インバータがトリップすると、コントローラは、止まった工具をワークに送り続ける代わりに、アラームを出してプログラムを停止します。
- 非常停止では、インバータはドライバと同時に電源を失い、主軸は惰性で止まります（ブレーキはかかりません）。

コントローラと配線

コントローラは、T41U5XBB キャリアボード上の Teensy 4.1 で、入力はフォトカプラで絶縁されています。Grbl 1.1 のシリアルプロトコルと互換性のある Twinwright 独自カーネルを実行し、USB でコンピューターに接続します。

入力	接続されているもの
リミット	X−、X+、Y−、Y+、Z の 5入力

入力	接続されているもの
プローブ	ツールセッターとタッチプローブ
非常停止	非常停止ボタンの 2つ目の接点
モーター異常	3つのドライバのアラーム出力と、主軸インバータの異常リレーを、直列に接続
サイクルスタート、フィードホールド	制御盤のボタン SB2 と SB3

電源

商用電源は、ブレーカーを通ります。24 V の電源は常時オンで、コントローラ、センサ、制御回路に給電します。そのため、コントローラはコンピューターに接続されたままで、非常停止を報告します。コンタクタ（KM1）が、ドライバ、主軸インバータ、ポンプの 48 V 電源を入り切りします。

制御盤のボタン

制御盤の扉にある Ø22 のボタン 4個です（正面から見た配置）。

	左	右
上段	SB1 電源オン（緑）	SB0 非常停止（赤）
下段	SB2 サイクルスタート（緑）	SB3 フィードホールド（黄）

- **SB1 電源オン** は KM1 を閉じ、KM1 はその後、自己保持します。非常停止や停電のあと、機械は自動では起動しません。非常停止を解除して、もう一度 SB1 を押してください。
- **SB3 フィードホールド** は、お使いの送信ソフトのフィードホールド（`!`）と同じ動作をします。軸は制御されながら停止し、主軸は回り続けます。このボタンは非常停止の真下にあるため、非常停止を押し損ねた手が当たるのは、サイクルスタートではなくフィードホールドです。
- **SB2 サイクルスタート** は、送信ソフトのサイクルスタート（`~`）と同じように、動作を続行します。

> **注意**
>
> フィードホールドは安全機能ではありません。安全のために機械を止めるときは、非常停止を使ってください。

ツールセッターとタッチプローブ

ツールセッターは、テーブルの左奥にあり、グリッドの穴に M6 ねじ 2本で固定されます。作業領域の角を、おおよそ X 0–57、Y 130–150 mm の範囲で占めます。ツールセッターとタッチプローブは、コントローラのプローブ入力を共有します。

Z が最上部にあるとき、コレットからの突き出しが 15.5 mm の工具の先端は、テーブルから 80 mm の高さにあり、主軸のノーズは 130 mm の高さです。突き出しがそれより長い工具は、先端が余分な長さの分だけ低くなります。ER コレットに保持したタッチプローブは、通常、ナットからスタイラス先端までの長さが 60–90 mm です。Z が最上部にあるとき、そのようなプローブが通過できるのは低い段取りだけで、70 mm のプローブならテーブルの約 20 mm 上までです。

Studio はこれを確認します。ワークを段取りしたとき、工具を交換したとき、工具テーブルを編集したときに、段取りの最も高い点を、Z が最上部にあるときの各工具の先端と比較します。そして、どの工具がどれだけ足りずに干渉するかを示します。

ワークの固定

テーブルの 25 mm ピッチの M6 穴には、クランプ、サブプレート、バイスを取り付けられます。ツインが扱えるのは、10 mm のアルミニウム製サブプレートに載せた 75 mm のセンタリングバイス、パラレル、ストラップクランプ、両面テープ、V ブロックです。

- バイスをサブプレートに載せると、ジョーの上端はテーブルからおよそ 66 mm の高さになるため、バイスで保持した部品は、テーブルからおよそ 71 mm 以上の高さに達します。
- プローブで測定したい部品には、低い段取りを使ってください。サブプレートにストラップクランプで固定するか（素材の上面がテーブルから 15–30 mm）、低いバイスを使います。75 mm のバイスで保持した部品の上には、タッチプローブが収まりません。
- 10 mm のサブプレートを外すと、工具には 10 mm 余裕ができます。タッチプローブには、それでも足りません。

安全

- **非常停止（SB0）**：コンタクタのコイル回路を開き、これによってドライバ、主軸インバータ、ポンプの電源が切れます。また、2つ目の接点でコントローラに伝えます。軸と主軸は電源を失い、主軸は惰性で止まります。Z が下がることがあります（Z 軸と重力を参照）。非常停止を解除しても、何も再起動しません。SB1 を押し、ドライバが起動するのを待ち、コントローラをリセットして、原点復帰を行ってください。
- **異常**：ドライバの異常、主軸インバータのトリップ、リミットスイッチの断線があると、機械はアラームを出して停止します。
- **フィードホールド** は、加工のための制御された停止であり、安全のための停止ではありません。
- MC-1 には囲いがありません。主軸が回っている間と、軸が動く可能性のある間は、近づかないでください。保護メガネを着用し、切りくずから守るため、X と Z のねじに蛇腹カバーまたはブラシカバーを取り付けてください。
- 機械の安全機能に AI を使うものはありません。Studio の AIアシスタントは安全機能ではありません。

立ち上げ時に測定する項目

上記の設計値は、組み立てた機械ごとに確認し、そのツインを測定した値に合わせます。

- 各リミットスイッチの作動点（誘導形近接スイッチは、側面から横切ると、図面より約 2 mm 遅れて作動することがあります）。続いて、原点復帰のプルオフ距離とソフトリミット
- ドライバのアラーム出力：1つを外すか、48 V を切ると、アラームになること

- ドライバの電源を切った状態での Z の試験
- インバータのパラメータ、インバータ本体のパネルでの無負荷の立ち上げ運転、そのあと RS-485 での制御
- 制御盤：上記のとおりの非常停止と SB1、すべてのリミットスイッチとツールセッターを 1つずつ外したとき（どれもアラームになること）、そして盤面のボタン

ツインが測定値をどう取り込むかは、機械プロファイルを参照してください。

Reference

このリファレンスはプログラムから生成したもので、英語版のみです。

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 31 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_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`
- Runs and results: `run_status`, `run_profile`, `result_compare`, `result_image`, `log_query`
- The licence: `licence_status`

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). 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`)
`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		

Name	Type	Default	Description
`frame`	string	`"work"`	(one of `work`, `machine`)
`feed`	number		mm/min (G1); omitted: rapid (G0)
`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_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 thin sheet cut through: fixture tape on a sacrificial plate (sacrificial) - the blank that much higher, the cuts through it go into the plate, not the table.

Parameters

Name	Type	Default	Description
`op`	string	`"place"`	(one of `place`, `remove`, `flip`)
`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`)
`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 only: the blank taped 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

Name	Type	Default	Description
`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). 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'
`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. 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)
`tol`	object		+/- mm (an angle: degrees) instead of ISO 2768-1
`tol.size`	number		
`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		

Name	Type	Default	Description
`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-S20 (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, ...) 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)
`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)
`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"`	
`version`	integer		with name, no gcode: that saved version's text (0: the last)
`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)
`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"`	
`line`	integer		file line (1-based)
`when`	object		conditions, e.g. {"force_n": 60, "cut": true}

Name	Type	Default	Description
`tools`	object of string		the tool table for M6: {"1": "flat 6 n3 L20", ...}
`until`	string	`"time"`	(one of `time`, `block`, `pause`)
`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"`	
`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`	

Name	Type	Default	Description
`goals.merge`	boolean	`true`	
`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

Name	Type	Default	Description
`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)
`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}; chamfer {an outline, width} (a V-bit); thread {points,d,pitch,depth[,hand]} (a thread mill); 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]} (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)
`verify`	boolean	`true`	

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`, `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`.

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`	
`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`)

Name	Type	Default	Description
`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`.

`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`.

設定リファレンス

コントローラのすべての設定です。`$<n>` で1つを表示し、`$<n>=<value>` で変更し、`$$` ですべてを一覧表示します。記載されている値はMC-1のものです。*再起動*と記された設定は、コントローラの再起動後に有効になります。

目次

- 全般
- 制御信号
- リミット
- 冷却
- 主軸
- 工具交換
- 原点復帰
- プロービング
- パーキング/安全扉
- ジョグ
- Modbus
- ステッピングモーター
- インバータ
- X軸
- Y軸
- Z軸

全般

設定	名前	単位	種類	範囲	MC-1	説明
`$10`	ステータスレポートのオプション		ビットフィールド		15871	リアルタイムのステータスレポートに含める内容です。パーサ状態は独立した行として、変化したときだけ送られます。ビット： 0 機械座標での位置, 1 バッファ状態, 2 シーケンス番号, 3 送り速度と主軸回転数, 4 ピン状態, 5 ワーク座標オフセット, 6 オーバーライド, 7 プローブ座標, 8 ワーク座標オフセット変更時のバッファ同期, 9 パーサ状態, 10 アラームのサブステータス, 11 実行中のサブステータス, 12 原点復帰中の有効化, 13 残移動量
`$11`	コーナー偏差	mm	小数		0.002	2つの移動の間のコーナーで、経路のずれをどこまで許容するかを表します。コーナーでの速度はこの値で決まります。小さいほど遅くなり、より正確になります。
`$12`	円弧公差	mm	小数		0.002	円弧をどれだけ正確にたどるかを表します。
`$13`	インチ単位でのレポート		ブール		0	レポートの位置と速度がインチ単位になります。
`$28`	G73の退避量	mm	小数		0.100	G73の各ステップ送りの間に行う、短い退避量です。
`$32`	動作モード		選択		0	フライス盤、レーザー、旋盤のいずれかです。選択肢： 0 通常, 1 レーザーモード, 2 旋盤モード
`$39`	従来のリアルタイムコマンドの有効化		ブール		1	「?」、「!」、「~」は、$コマンドやコメントの中も含め、どこでもリアルタイムコマンドとして動作します。
`$60`	オーバーライドの復元		ブール		0	プログラムの終了時に、オーバーライドが100 %に戻ります。
`$62`	スリープの有効化		ブール		0	アイドル状態が長く続くと、スリープ状態に入ります。
`$63`	フィードホールド時の動作		ビットフィールド		3	フィードホールドで、動作の停止に加えて行う処理です。ビット： 0 ホールド中のレーザー無効化, 1 再開時の主軸・冷却状態の復元, 2 主軸回転数の最低値設定（レーザーのM4を除く）
`$64`	起動時アラームの強制		ブール		0	電源投入後、アラーム状態で起動します。
`$384`	G92保持の無効化		ブール		0	電源を入れ直したとき、G92オフセットを保持しません。
`$398`	プランナバッファのブロック数 *(再起動)*		整数	30–1000	100	プランナが先読みする移動の数です。
`$481`	自動レポートの間隔 *(再起動)*	ms	整数	100–1000	0	この間隔で、要求がなくてもステータスレポートを送信します（0：オフ）。
`$484`	非常停止後のロック解除必須		ブール		1	非常停止の後、ロック解除 ($X) または原点復帰を行うまで、機械はロックされたままになります。

設定	名前	単位	種類	範囲	MC-1	説明
$486	座標系のロック		ビットフィールド		0	G10による変更を禁止する座標系です。ビット： 0 G59.1, 1 G59.2, 2 G59.3
$676	リセット時の動作		ビットフィールド		15	リセットでクリアされる内容です。ビット： 0 位置喪失時の原点復帰済み状態のクリア, 1 オフセットのクリア（G92を除く）, 2 早送りオーバーライドのクリア, 3 送りオーバーライドのクリア
$1030	送信ソフト切断時のジョブ停止		ブール		1	ジョブを実行している送信ソフトとの接続が0.1 s間途切れると（USBケーブルが抜かれた、ポートが閉じられたなど）、ジョブは停止 (0x19) を受けたときと同じように止まります。動作は減速して停止し、主軸と冷却がオフになり、送信済みでまだ実行されていないブロックは破棄され、位置は保持されます。接続が戻ると、送信ソフトに通知されます。ファイルシステムから実行するプログラムは、送信ソフトがなくてもそのまま実行が続きます。

制御信号

設定	名前	単位	種類	範囲	MC-1	説明
$14	制御入力の反転		ビットフィールド		0	反転させる制御入力です。ビット： 1 フィードホールド, 2 サイクルスタート, 6 非常停止, 8 モーター異常
$17	制御入力のプルアップ無効化		ビットフィールド		0	内部プルアップ抵抗を使わない制御入力です。ビット： 1 フィードホールド, 2 サイクルスタート, 6 非常停止, 8 モーター異常

リミット

設定	名前	単位	種類	範囲	MC-1	説明
$5	リミット入力の反転		軸マスク		7	反転させるリミット入力です。たとえば常時閉 (NC) のスイッチを使う場合に指定します。
$18	リミット入力のプルアップ無効化		軸マスク		0	内部プルアップ抵抗を使わないリミット入力です。
$20	ソフトリミットの有効化		ブール		1	ストローク ($130 ...) を超える動作を拒否します。位置が分かっている必要があるため、原点復帰が必要です。
$21	ハードリミットの有効化		ビットフィールド		1	リミットスイッチが作動すると、直ちに停止します。厳格モード：スイッチが作動している間は、一切動作できません。ビット： 0 有効, 1 厳格モード （ビット0で残りを有効化）

冷却

設定	名前	単位	種類	範囲	MC-1	説明
$15	冷却出力の反転		ビットフィールド		0	アクティブローの冷却出力です。ビット： 0 フラッド, 1 ミスト
$673	冷却オン遅延	s	小数	0.5–20	0.0	M7/M8の後、冷却が立ち上がるまで待つ時間です。

主軸

設定	名前	単位	種類	範囲	MC-1	説明
$30	主軸最高回転数	min^{-1}	小数		24000.000	主軸の最高回転数です。
$31	主軸最低回転数	min^{-1}	小数		6000.000	主軸の最低回転数です。
$340	主軸回転数到達の許容差	%	小数		5.0	M3/M4の後の動作は、主軸回転数が指令回転数に対してこの値（パーセント）以内になるまで待機します（0：待機しない）。
$394	主軸オン遅延	s	小数	0.5–60	15.0	M3/M4の後、主軸が回転数に達するまで待つ時間です（$340を設定した場合は、アラームが出るまでの最長の待ち時間）。
$539	主軸オフ遅延	s	小数	0.5–60	0.0	M5の後、主軸が停止するまで待つ時間です。

工具交換

設定	名前	単位	種類	範囲	MC-1	説明
$341	工具交換モード		選択		3	M6の動作を決めます。選択肢： 0 通常, 1 手動タッチオフ, 2 G59.3での手動タッチオフ, 3 G59.3での自動タッチオフ, 4 M6の無視
$342	工具交換時のプロービング距離	mm	小数		50.0	ツールセッターを探索する距離です。

設定	名前	単位	種類	範囲	MC-1	説明
$343	工具交換時の検出送り速度	mm/min	小数		25.0	ツールセッターの位置を正確に検出する、低速の送り速度です。
$344	工具交換時の探索速度	mm/min	小数		200.0	ツールセッターを探索する、高速の送り速度です。
$345	工具交換時のプローブプルオフ速度	mm/min	小数		200.0	ツールセッターから離れる送り速度です。
$346	工具交換オプション		ビットフィールド		0	工具交換のオプションです。ビット： 0 M6後の位置復元, 1 G30での工具交換, 2 プローブの高速プルオフ
$485	再起動後の工具番号保持		ブール		0	電源を入れ直しても、主軸に装着されている工具の番号を保持します。

原点復帰

設定	名前	単位	種類	範囲	MC-1	説明
$22	原点復帰サイクル		ビットフィールド		541	原点復帰サイクルとそのオプションです。ビット： 0 有効, 1 単軸原点復帰コマンドの有効化, 2 起動時の原点復帰必須, 3 機械原点の0設定, 4 2つのスイッチによる入力の共用, 5 手動の許可, 6 ロックのオーバーライド, 9 軸ごとの送り速度, 10 スタートアップブロックの実行を原点復帰完了後に限定 （ビット0で残りを有効化）
$23	原点復帰方向の反転		軸マスク		0	マイナス側の端に向かって原点復帰する軸です。
$26	原点復帰スイッチのチャタリング防止遅延	ミリ秒	整数		250	スイッチが作動した後、次の原点復帰動作を始めるまで待つ時間です。
$27	原点復帰スイッチのプルオフ距離	mm	小数		2.000	原点復帰でスイッチから戻る距離です。
$43	原点復帰のパス数		整数	1–128	1	スイッチの作動点を検出する回数です。
$44	原点復帰する軸（第1段階）		軸マスク		4	最初に原点復帰する軸です。
$45	原点復帰する軸（第2段階）		軸マスク		3	2番目に原点復帰する軸です。
$46	原点復帰する軸（第3段階）		軸マスク		0	3番目に原点復帰する軸です。

プロービング

設定	名前	単位	種類	範囲	MC-1	説明
$6	プローブ入力の反転		ビットフィールド		1	プローブ入力を反転します。たとえば常時閉 (NC) のツールセッターを使う場合に指定します。ビット： 0 プローブ
$19	プローブ入力のプルアップ無効化		ビットフィールド		0	内部プルアップ抵抗を使わないプローブ入力です。ビット： 0 プローブ
$65	プロービングオプション		ビットフィールド		0	プロービング動作の扱いを決めます。ビット： 0 送りオーバーライドの許可, 1 ソフトリミットの適用

パーキング/安全扉

設定	名前	単位	種類	範囲	MC-1	説明
$41	パーキングサイクル		ビットフィールド		0	安全扉が開いたときに、主軸を退避させます。ビット： 0 有効, 1 起動時の無効化, 2 パーキングオーバーライド制御の有効化 （ビット0で残りを有効化）
$42	パーキング軸		選択		2	パーキングを行う軸です。選択肢： 0 X, 1 Y, 2 Z
$56	パーキングの引き抜き距離	mm	小数		5.0	工具を最初にゆっくり引き抜く距離です。
$57	パーキングの引き抜き速度	mm/min	小数		100.0	この引き抜きの速度です。
$58	パーキング目標位置	mm	小数	≥ -100000	-5.0	パーキング軸が移動する位置を、機械座標で指定します。
$59	パーキングの高速移動速度	mm/min	小数		500.0	パーキング移動の残りの部分の速度です。
$61	安全扉オプション		ビットフィールド		0	安全扉が開いている間もオンのままにするものです。ビット： 1 扉開放時の冷却状態の保持
$392	主軸オン遅延	s	小数	0.5–60	4.0	扉が再び閉じたときに、主軸が回転数に達するまで待つ時間です。
$393	冷却オン遅延	s	小数	0.5–20	1.0	扉が再び閉じたときに、冷却が立ち上がるまで待つ時間です。

ジョグ

設定	名前	単位	種類	範囲	MC-1	説明
$40	ジョグコマンドの制限		ブール		0	ジョグは拒否されず、ソフトリミットの位置で停止します。

Modbus

設定	名前	単位	種類	範囲	MC-1	説明
$374	Modbusのボーレート		選択		3	Modbus通信のボーレートです（インバータ側の設定と一致させてください）。選択肢： 0 2400, 1 4800, 2 9600, 3 19200, 4 38400, 5 115200
$375	Modbus受信タイムアウト	ミリ秒	整数	50–250	50	Modbusの応答を待つ時間です。
$681	Modbusのシリアル形式		選択		0	Modbus通信のパリティです（インバータ側の設定と一致させてください）。選択肢： 0 8ビット、パリティなし, 1 8ビット、偶数パリティ, 2 8ビット、奇数パリティ

ステッピングモーター

設定	名前	単位	種類	範囲	MC-1	説明
$0	ステップパルス幅	マイクロ秒	小数	≥ 2.0	5.0	ステップパルスの長さ（マイクロ秒単位）です。ドライバには最小値があります（ドライバの取扱説明書を参照してください。MC-1のクローズドループドライバの最小値は2.5です）。
$1	ステップアイドル遅延	ミリ秒	整数	≤ 65535	255	動作が停止した後、ドライバがイネーブルのままでいる時間です。電流を切る前に軸が安定するようにします。255を指定すると、常にイネーブルのままになります。
$2	ステップパルスの反転		軸マスク		0	アクティブローのステップ出力です。
$3	ステップ方向の反転		軸マスク		5	方向出力を反転させる軸です（反転させないと軸が逆方向に動きます）。
$4	ステッピングモーターのイネーブル出力の反転		軸マスク		7	アクティブローのドライバイネーブル出力です（ほとんどのドライバが該当します）。イネーブル信号を1本で共用している場合は、Xの設定のみが有効です。
$29	パルス遅延	マイクロ秒	小数	≤ 20	5.0	方向出力を設定してから、ステップパルスを出すまでの時間です。
$37	イネーブルを維持するドライバ		軸マスク		7	機械がアイドル状態のときやアラーム中も、イネーブルのままにするドライバです。
$680	ステッピングモーターのイネーブル遅延	ms	整数	≤ 500	0	ドライバをイネーブルにしてから、最初のステップを出すまでの時間です。

インバータ

設定	名前	単位	種類	範囲	MC-1	説明
$460	主軸インバータのModbusアドレス		整数	≤ 255	1	インバータのModbusアドレスです。

X軸

設定	名前	単位	種類	範囲	MC-1	説明
$100	X軸の移動分解能	ステップ/mm	小数		800.00000	軸の1 mm当たりのステップ数です。モーター1回転当たりのステップ数にマイクロステップ数を掛け、ねじのリードで割った値です。
$110	X軸の最高速度	mm/min	小数		3000.000	軸の最高速度です。早送りの速度もこの値が上限になります。
$120	X軸の加速度	mm/s²	小数		300.000	軸の加速度です。
$130	X軸の最大ストローク	mm	小数		202.000	原点位置からの軸のストロークで、ソフトリミットに使われます。
$180	X軸の原点復帰検出送り速度	mm/min	小数		50.0	この軸のスイッチの作動点を正確に検出する、低速の送り速度です。
$190	X軸の原点復帰探索速度	mm/min	小数		1200.0	この軸のスイッチを探索する、高速の送り速度です。
$1000	X軸のインプットシェーパー		選択		0	軸の動作を整形し、下に示す周波数で機械の共振を励起しないようにします。ZVはその周波数を正確に打ち消し、ほかの方式はより広い帯域を打ち消しますが、精度は劣ります。動作は最大1.5周期遅れて静止します。選択肢： 0 なし, 1 ZV, 2 ZVD, 3 MZV, 4 EI, 5 2HUMP_EI

設定	名前	単位	種類	範囲	MC-1	説明
$1010	X軸のインプットシェーパー周波数	Hz	小数	15–500	100.0	この軸のインプットシェーパーが打ち消す共振の周波数です。
$1020	X軸のインプットシェーパー減衰比		小数	≤ 0.5	0.100	この共振の減衰比です。

Y軸

設定	名前	単位	種類	範囲	MC-1	説明
$101	Y軸の移動分解能	ステップ/mm	小数		800.00000	軸の1 mm当たりのステップ数です。モーター1回転当たりのステップ数にマイクロステップ数を掛け、ねじのリードで割った値です。
$111	Y軸の最高速度	mm/min	小数		3000.000	軸の最高速度です。早送りの速度もこの値が上限になります。
$121	Y軸の加速度	mm/s²	小数		300.000	軸の加速度です。
$131	Y軸の最大ストローク	mm	小数		152.000	原点位置からの軸のストロークで、ソフトリミットに使われます。
$181	Y軸の原点復帰検出送り速度	mm/min	小数		50.0	この軸のスイッチの作動点を正確に検出する、低速の送り速度です。
$191	Y軸の原点復帰探索速度	mm/min	小数		1200.0	この軸のスイッチを探索する、高速の送り速度です。
$1001	Y軸のインプットシェーパー		選択		0	軸の動作を整形し、下に示す周波数で機械の共振を励起しないようにします。ZVはその周波数を正確に打ち消し、ほかの方式はより広い帯域を打ち消しますが、精度は劣ります。動作は最大1.5周期遅れて静止します。選択肢： 0 なし, 1 ZV, 2 ZVD, 3 MZV, 4 EI, 5 2HUMP_EI
$1011	Y軸のインプットシェーパー周波数	Hz	小数	15–500	100.0	この軸のインプットシェーパーが打ち消す共振の周波数です。
$1021	Y軸のインプットシェーパー減衰比		小数	≤ 0.5	0.100	この共振の減衰比です。

Z軸

設定	名前	単位	種類	範囲	MC-1	説明
$102	Z軸の移動分解能	ステップ/mm	小数		800.00000	軸の1 mm当たりのステップ数です。モーター1回転当たりのステップ数にマイクロステップ数を掛け、ねじのリードで割った値です。
$112	Z軸の最高速度	mm/min	小数		1500.000	軸の最高速度です。早送りの速度もこの値が上限になります。
$122	Z軸の加速度	mm/s²	小数		200.000	軸の加速度です。
$132	Z軸の最大ストローク	mm	小数		82.000	原点位置からの軸のストロークで、ソフトリミットに使われます。
$182	Z軸の原点復帰検出送り速度	mm/min	小数		50.0	この軸のスイッチの作動点を正確に検出する、低速の送り速度です。
$192	Z軸の原点復帰探索速度	mm/min	小数		600.0	この軸のスイッチを探索する、高速の送り速度です。
$1002	Z軸のインプットシェーパー		選択		0	軸の動作を整形し、下に示す周波数で機械の共振を励起しないようにします。ZVはその周波数を正確に打ち消し、ほかの方式はより広い帯域を打ち消しますが、精度は劣ります。動作は最大1.5周期遅れて静止します。選択肢： 0 なし, 1 ZV, 2 ZVD, 3 MZV, 4 EI, 5 2HUMP_EI
$1012	Z軸のインプットシェーパー周波数	Hz	小数	15–500	100.0	この軸のインプットシェーパーが打ち消す共振の周波数です。
$1022	Z軸のインプットシェーパー減衰比		小数	≤ 0.5	0.100	この共振の減衰比です。

アラームコード

コントローラのアラームです。アラームが発生すると機械は停止し、解除するまでロックされます。解除はアラームの指示に従い、`$X`または原点復帰で行います。`$EA`を使うと、コントローラ自体でもアラームの一覧を表示できます。

コード	意味
`ALARM:1`	ハードリミット：動作中にリミットスイッチが作動しました。位置が失われた可能性が高いため、再度原点復帰してください
`ALARM:2`	ソフトリミット：移動の目標位置が機械のストローク外です。位置は保持されているため、ロック解除できます
`ALARM:3`	移動中にリセットまたは非常停止が行われました。位置が失われた可能性が高いため、再度原点復帰してください
`ALARM:4`	プローブ：プロービングサイクルの開始時に、プローブが想定した状態になっていませんでした
`ALARM:5`	プローブ：指令された移動範囲内で接触がありませんでした
`ALARM:6`	原点復帰：原点復帰サイクルがリセットされました
`ALARM:7`	原点復帰：原点復帰中に安全扉が開かれました
`ALARM:8`	原点復帰：プルオフ後もリミットスイッチが解除されませんでした。プルオフ距離 ($27) と配線を確認してください
`ALARM:9`	原点復帰：探索距離内にリミットスイッチが見つかりませんでした。ストロークの設定と配線を確認してください
`ALARM:10`	非常停止です。解除してから、リセットしてください
`ALARM:11`	原点復帰が必要です：原点復帰サイクルを実行してください ($H)
`ALARM:12`	リミットスイッチが作動しています。解除してから続行してください
`ALARM:13`	プローブ保護：プロービングサイクル以外でプローブが作動しました
`ALARM:14`	主軸が時間内に指令回転数に達しませんでした
`ALARM:15`	原点復帰：オートスクエアリング軸の2つ目のスイッチが見つかりませんでした
`ALARM:16`	電源投入時の自己診断に失敗しました
`ALARM:17`	モータードライバが異常を報告しています
`ALARM:18`	原点復帰：原点復帰の設定が無効です
`ALARM:19`	Modbus：機器（インバータなど）から応答がないか、応答が不正です
`ALARM:20`	I/Oエキスパンダが応答しません
`ALARM:21`	不揮発性メモリ（設定メモリ）に異常が発生しました
`ALARM:22`	バッファがオーバーフローしました

エラーコード

コントローラがブロックを拒否したときの応答 `error:N` です。そのブロックでは何も実行されません。`$EE` を使うと、コントローラ自体でもエラーの一覧を表示できます。

コード	意味
`error:1`	Gコードのワードに文字がありません：ワードはすべて、文字の後に数値が続く形式です
`error:2`	数値がないか、数値の形式が正しくありません
`error:3`	不明、または未対応の「$」コマンドです
`error:4`	正の値のみ使用できる箇所に、負の値が指定されました
`error:5`	設定で原点復帰が有効になっていません ($22)
`error:6`	ステップパルスの長さは2マイクロ秒以上にしてください
`error:7`	保存されている設定を読み込めませんでした：該当する設定を初期値に戻しました
`error:8`	この「$」コマンドは、機械がアイドル状態のときだけ受け付けます
`error:9`	アラーム中またはジョグ中は、Gコードを受け付けません
`error:10`	ソフトリミットを使うには原点復帰が必要です：先に原点復帰を有効にしてください ($22)
`error:11`	ブロックがコントローラの受け付けられる長さを超えているため、実行されませんでした
`error:12`	この値ではステップレートがコントローラの出力できる上限を超えます
`error:13`	安全扉が開いています
`error:14`	テキストが長すぎて保存できません（スタートアップブロックまたはビルド情報）
`error:15`	ジョグの目標位置が機械のストローク外です：ジョグは実行されませんでした
`error:16`	ジョグコマンドには「=」が必要で、ジョグ用のワードしか含められません
`error:17`	レーザーモードにはPWM主軸出力が必要です
`error:18`	コントローラがリセットされました
`error:19`	値はゼロより大きくする必要があります
`error:20`	ブロックに未対応または無効なGコード指令が含まれています
`error:21`	1つのブロックに同じモーダルグループの指令が2つあります
`error:22`	送り速度がまだ指定されていません
`error:23`	この指令には整数が必要です
`error:24`	1つのブロックに、どちらも軸ワードを使う指令が2つあります
`error:25`	同じワードが1つのブロックに2回あります
`error:26`	この指令には軸ワードが必要ですが、ブロックにありません
`error:27`	無効なシーケンス番号です
`error:28`	指令に必要な値ワードがありません
`error:29`	G59.1、G59.2、G59.3には対応していません
`error:30`	G53はG0とG1でのみ使用できます
`error:31`	どの指令も使わない軸ワードがブロックにあります
`error:32`	G2/G3円弧には、その平面内の軸ワードが少なくとも1つ必要です
`error:33`	移動先が無効です
`error:34`	円弧の半径が無効です
`error:35`	G2/G3円弧には、その平面内の中心オフセットワード（I、J、Kのいずれか）が少なくとも1つ必要です
`error:36`	どの指令も使わない値ワードがブロックにあります
`error:37`	G43.1には、設定された工具長軸が必要です
`error:38`	工具番号が対応範囲を超えているか、その工具がありません
`error:39`	値が範囲外です
`error:40`	工具交換の待機中は使用できません
`error:41`	この動作には主軸の回転が必要です（周速一定制御または主軸同期）
`error:42`	ねじ切りにはZX平面 (G18) が必要です
`error:43`	送り速度が最大値を超えています
`error:44`	主軸回転数が範囲外です
`error:45`	リミットスイッチが作動中です：原点復帰のみ可能です

コード	意味
error:46	先に機械の原点復帰を行ってください ($H)
error:47	現在の工具が不明です：M61で設定してください
error:48	値ワードが競合しています
error:49	電源投入時の自己診断に失敗しました：コントローラの電源を切って入れ直してください
error:50	非常停止中です
error:51	モータードライバが異常を報告しています
error:52	設定値が範囲外です
error:53	このコントローラにはその設定がありません
error:54	退避位置が穴あけ深さより下にあります
error:55	オートスクエアリングの2軸は同時に原点復帰できません
error:56	この座標系はロックされています ($486)
error:57	予期しないプログラム開始/終了マーク (%)
error:58	ポートを使用できません
error:60	SDカードをマウントできませんでした
error:62	ディレクトリの内容を一覧表示できませんでした
error:63	そのディレクトリは存在しません
error:64	SDカードがマウントされていません
error:65	ファイルシステムがマウントされていません（カードが入っていますか？）
error:66	ファイルシステムは読み取り専用です
error:71	式：不明な演算子または関数です
error:72	式：ゼロで除算しています
error:73	式：引数が範囲外です
error:74	式：引数が無効です
error:75	式：構文エラーです
error:76	式：結果が数値ではありません
error:77	認証が必要です
error:78	アクセスが拒否されました
error:79	重大イベントの発生中は実行できません
error:80	この O ワード文には、コントローラが読み直せるプログラムが必要です（コントローラのファイルシステムから実行）
error:81	不明な O ワード文か、O ワード文の位置が正しくありません
error:82	同時に開いている O ワードのブロックまたは呼び出しが多すぎます
error:83	O ワード文を格納する領域が残っていません
error:84	ファイルを開けませんでした
error:85	ファイルシステムをフォーマットできませんでした
error:86	このポートは使用できません
error:87	主軸に工具が装着されています
error:88	主軸に工具が装着されていません
error:89	ファイルを削除できませんでした
error:253	ユーザー定義のエラーです

このリファレンスはプログラムから生成したもので、英語版のみです。

Command line: mc1sim

The options of the twin's program `mc1sim`, as its usage text gives them (85 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)
`--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)
`--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

option	what it does
`--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)
`--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)

option	what it does
`--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)

このリファレンスはプログラムから生成したもので、英語版のみです。

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		"6528e49911ce4825f7555ab9e42991051ee85046b9ed9511f960cfbb…	
`generated.wiring_sha256`	string		"4f378185683bcbd33fb0a42e8fa2b79e93474c861a9a2f76da1e7ee1…	
`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		"2026-10-07T18:16:05"	

`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)		[-8.0, 208.0]	where the moving part hits the frame [min, max], machine mm

key	type	unit	value	what it is
axes[].moving_mass_kg	number	kg	13.83	

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	1274.0	
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	
`drives.drive.max_phase_advance_deg`	number	°	0.0	

key	type	unit	value	what it is
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	327.0	
drives.axes[].screw_inertia_kgm2	number	kg·m²	3.347015168018846e-06	

key	type	unit	value	what it is
drives.axes[].rotating_inertia_kgm2	number	kg·m²	3.594701516801885e-05	
drives.axes[].moving_mass_kg	number	kg	13.83	
drives.axes[].equivalent_mass_kg	number	kg	102.52570477707619	
drives.axes[].gravity_n	number	N	0.0	
drives.axes[].friction_n	number	N	57.06657900384689	
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	88.69570477707619	
drives.axes[].l2.table_mass_kg	number	kg	13.83	
drives.axes[].l2.k_axial_n_um	array (2)	N/µm		
drives.axes[].l2.friction_motor_n	number	N	47.12388980384689	
drives.axes[].l2.friction_table_n	number	N	9.9426892	
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	11.46605033855195	
drives.axes[].l2.coupling_n_um	number	N/µm	1189.2873303312676	
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	61.593853491210936	
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	31.87940238298543	
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	33.15	
drives.axes[].l2.frame.m_kg	number	kg	17.64	
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	
accuracy.axes.X.periodic_mm	number	mm	

key	type	unit	what it is
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)		
accuracy.axes.Z.backlash_um	number	µm	

key	type	unit	what it is
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		
accuracy.thermal.part.g_good_w_k	number	W/K	

key	type	unit	what it is
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	
spindle.vfd.factory.PD119	number		150	

key	type	unit	value	what it is
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	
spindle.motor.poles	number		2	

key	type	unit	value	what it is
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		-42.0	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
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.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, 256.0, 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, 465.0]	
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			
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"	

key	type	unit	value	what it is
`sound.sources.setter`	object			
`sound.sources.setter.at`	array (3)		[-75.0, 75.0, 97.0]	
`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, 0.0]	
`collision.table_items[].max`	array (3)		[-43.0, 84.0, 5.0]	
`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]	
`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.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)
`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)

key	type	unit	value	what it is
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…	
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.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	

key	type	unit	value	what it is
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	[2.3492, 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
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	
`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.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	65.75	

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		50	
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	c
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	c
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	c
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	c
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	d
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	d
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	d
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	d
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.typical	c

key	t
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
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

key	t
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
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

key	t
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	c
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	c
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	c
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	c
provenance.cutting.force_model.source	s
provenance.cutting.force_model.note	s
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	c
provenance.cutting.runout.source	s
provenance.cutting.runout.note	s

key	t
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.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
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

key	t
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.total_mass_kg	o
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	o
provenance.viewer.source	s

key	t
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	o
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
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 (1)		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		

key	type	unit	what it is
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)		
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 (0)		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 (1)		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		

1759 keys.

プロトコル

Twinwright 独自カーネルは、Grbl 1.1 シリアルプロトコルで通信し、現在の Grbl 送信ソフトが使う拡張にも対応しています。Grbl 用に書かれた送信ソフトは、このカーネルでも動作します。動作を確認した送信ソフトとその設定は、G コード送信ソフトにあります。

接続

接続先	方法
Studio のツイン	このコンピューターの TCP ポート 2323。送信ソフトは同時に 1つだけ接続できます。
MC-1	ネイティブ USB ポート。シリアルポート（Windows では COM*n*）として使います。ボーレートは関係ありません。送信ソフトが指定を求める場合は 115200 を使ってください。送信ソフトが DTR をオンにすると、コントローラが応答します。

行と応答

- **行の終わり：** 行は LF または CR で終わります。CR LF または LF CR の組は 1回と数え、LF が 2つ続けば 2行になります。
- **応答：** コントローラは、すべての行に `ok` または `error:N` で応答します。応答は、行が届いた順に返ります。N の値はエラーコードにすべて載っています。
- **行の長さ：** 256文字までです。これより長い行には `error:11` と応答します。
- **文字カウント方式：** 受信バッファは 1024 バイトです。送信ソフトは、応答がまだ返っていないバイトをその量まで送り出しておけます。空き容量は、ステータスレポートの `Bf:` で分かります。
- **プランナ：** 100 ブロックです。空きブロック数も `Bf:` で分かります。
- **待機する行：** フィードホールド中や安全扉が開いている間、行はバッファ内で待機します。サイクルが再び始まると、それらの行に応答します。リセット、`0x19`（ストップ）、`0x03`、`0x85`（ジョグキャンセル）のいずれかを受けると、まだ待機している行は破棄されます。
- **エラーの後：** 次の G コードの行は実行されず、同じエラーで応答します。これは、`$` コマンド、空行、またはリセットが来るまで続きます。

リアルタイムコマンド

リアルタイムコマンドは単一のバイトです。コントローラは、行の途中であっても、届いた時点で直ちに処理します。リアルタイムコマンドはどの行にも含まれず、`ok` も返りません。

バイト	コマンド	動作
`0x18`（Ctrl-X）	リセット	すべてを直ちに停止し、キューにあるものを消去します。軸が動いている間にリセットすると `ALARM:3` が発生します。原点復帰をやり直してください。非常停止中は受け付けません。
`0x03`（Ctrl-C）	リセット（非常停止中）	Ctrl-X と同じ動作で、非常停止中でも受け付けます。
`0x19`（Ctrl-Y）	ストップ	軸は減速して停止し、キューに残っている行は破棄されます。位置は保たれ、アラームは発生しません。
`ESC`、続けて `0x14`	再起動	自分で再起動できるボードで、コントローラを再起動します。
`?`	ステータスレポート	ステータスレポートを 1回返します。
`0x80`（または `0x05`）	ステータスレポート	ステータスレポートを 1回返します。ただし、自動レポートがオンの間は返しません。
`0x87`	フルステータスレポート	すべてのフィールドを含むステータスレポートを 1回返します。
`~`、`0x81`	サイクルスタート	フィードホールド、安全扉、`M0`/`M1`、工具交換のあとで再開します。
`!`、`0x82`	フィードホールド	軸は減速して停止し、そのまま保持されます。ジョグ中はジョグをキャンセルします。
`0x83`	パーサの状態	`$G` と同じく、モーダル状態を報告します。
`0x84`	安全扉	扉が開いたときと同じ動作をします。ホールドし、パーキングが設定されていればパーキングします。
`0x85`	ジョグキャンセル	ジョグを終了し、キューに残っているジョグの行を破棄します。
`0x88`	オプショナルストップ	`M1` による一時停止のオン／オフを切り替えます。
`0x89`	シングルブロック	シングルブロックのオン／オフを切り替えます。
`0x8B`	ハンドルモード	ハンドル（手動パルス発生器）が接続されている場合：そのモードを切り替えます。接続されていなければ無視されます。
`0x8C`	自動レポート	`$481` で間隔が設定されている場合に、自動ステータスレポートのオン／オフを切り替えます。設定されていなければ無視されます。
`0x90`	送りオーバーライド 100 %	
`0x91`、`0x92`	送りオーバーライド +10 %、−10 %	範囲：10–200 %。
`0x93`、`0x94`	送りオーバーライド +1 %、−1 %	範囲：10–200 %。
`0x95`、`0x96`、`0x97`、`0x98`	早送りオーバーライド	100 %、50 %、25 %、5 %。
`0x99`	主軸オーバーライド 100 %	
`0x9A`、`0x9B`	主軸オーバーライド +10 %、−10 %	範囲：10–200 %。
`0x9C`、`0x9D`	主軸オーバーライド +1 %、−1 %	範囲：10–200 %。

バイト	コマンド	動作
`0x9E`	主軸停止	フィードホールド中のみ：主軸を停止し、または再び始動します。
`0x9F`	ソフト非常停止	直ちに停止し、非常停止のアラームを発生させます。
`0xA0`	クーラント（フラッド）	フラッドクーラントのオン／オフを切り替えます。
`0xA1`	クーラント（ミスト）	ミストクーラントのオン／オフを切り替えます。
`0xA3`	工具交換完了	手動の工具交換を確認します。

コントローラは、これ以外の制御バイトと、コマンドが割り当てられていない `0x80–0xBF` のバイトを破棄します。

- **UTF-8 テキスト**（コメントやメッセージ内）は、テキストとして読み取られます。そのような文字に含まれる `0x80–0xBF` のバイトが、リアルタイムコマンドとして扱われることはありません。
- **`$39=0`**（従来のリアルタイム文字をオフ）：`?`、`!`、`~` は、`$` の行とコメントの中ではテキスト、それ以外の場所ではコマンドです。`$39=1`（デフォルト）では、常にコマンドです。

起動メッセージと識別情報

電源投入後と、リセットのたびに、コントローラは次の行を送ります。

```
Grbl 1.1f ['$' or '$HELP' for help]
```

送信ソフトは、「Grbl」で始まる行を再起動と見なします。

`$I` は次の行を返し、最後に `ok` を返します。

行	内容
`[VER:Twinwright 1.1f.<date>:<text>]`	プロトコルのバージョンとそのビルド日。`<text>` は `$I=<text>` で保存した文字列です。
`[OPT:…]`	オプションの文字、プランナのブロック数（100）、受信バッファ（1024）。
`[NEWOPT:…]`	オプション名：`ENUMS`、`RT+` または `RT-`（`$39`）、`HOME`、`ES`、`MPG`、`LATHE`、`REBOOT`、`TC`、`SS`（主軸同期）、`SED`、`RTC`、`SD`、`FS`、`YM`。
`[FIRMWARE:Twinwright]`	ファームウェアの名前。
`[SIGNALS:…]`	ボードが持つ制御入力を、文字で表したもの。
`[NVS STORAGE:…]`	設定の保存先。
`[DRIVER:…]`、`[DRIVER VERSION:…]`、`[DRIVER OPTIONS:…]`、`[BOARD:…]`	ボードの情報。
`[KERNEL:<version>+<build>]`	コントローラのバージョンとビルド：リリースのコミット、または `dev`。
`[KNCAP:…]`	ボードの機能（下記）。
`[PLUGIN:…]`	送信ソフトが認識する名前で示した機能：主軸インバータ用の `MODBUS` と `HUANYANG VFD`。

ボードの機能：KNCAP

`[KNCAP:]` は、`key=value` の組をカンマで区切って並べた 1行です。キーは末尾に追加されるだけで、意味が変わることはありません。行は位置ではなくキーで読んでください。欠けているキーは「不明」を意味します。

キー	値	意味
`port`	`teensy41`、`stm32h7`、`stm32f4`、`twin`、`host`、`sim`、`unknown`	コントローラのどのビルドか。
`pv`	数値	コントローラのポートインターフェースのバージョン。
`view`	数値	コントローラの状態ビューのバージョン。
`spindle`	`vfd`、`pwm`、`none`	主軸インバータ：Modbus、PWM 出力、またはなし。
`encoder`	`0`、`1`	主軸エンコーダが取り付けられています。
`sd`	`0`、`1`	ボードに SD カードスロットがあります。
`mpg`	`0`、`2`	ハンドルのポート。モード番号で示します。
`aux`	`di/do/ai/ao`	補助入出力：デジタル入力、デジタル出力、アナログ入力、アナログ出力。
`axes`	文字	軸。たとえば `XYZ`。
`probe`	`0`、`1`	プローブ入力。MC-1 では、ツールセッターもこの入力を使います。
`link`	`usb`、`uart`、`tcp`、`none`	送信ソフトとの接続。`usb` は、DTR がオンになると応答する USB シリアルポートです。

シャドーモードでは、実機とそのツインの間で `[KERNEL:]` と `[KNCAP:]` を比較します（実機とシャドーモードを参照）。

ステータスレポート

ステータスレポートは、`?`、`0x80`、`0x87` の後に返ります。自動レポートがオンの場合（`$481`、`0x8C`）は、コマンドなしでも自動的に返ります。次のような 1行です。

```
<Idle|MPos:0.000,0.000,0.000|Bf:100,1023|FS:0,0|Pn:XYZ>
```

先頭は状態です。

- `Idle`、`Run`、`Jog`、`Home`、`Check`（チェックモード）、`Sleep`、`Tool`（工具交換）。
- `Hold:0`（保持中）と `Hold:1`（減速中）。
- `Door:0`（閉じていて、再開できる状態）、`Door:1`（停止していて、まだ開いている）、`Door:2`（開いた：保持中またはパーキング中）、`Door:3`（閉じていて、復帰中）。
- `Alarm`。

続くフィールドは、`$10` と状況によって変わります。

フィールド	意味
`MPos:` または `WPos:`	機械位置またはワーク位置（`$10` のビット 0）。
`Bf:`	プランナの空きブロック数、受信バッファの空きバイト数。
`Ln:`	実行中のブロックの行番号。
`FS:`	送り速度、指令した主軸回転数、ドライバが報告する回転数。オンとオフだけの主軸は `F:` を報告します。フィードホールド中は送り速度が 0 になります。
`Pn:`	アクティブな入力：リミットは軸の文字、`P` プローブ、`D` 安全扉、`H` フィードホールド、`S` サイクルスタート、`R` リセット、`E` 非常停止。
`WCO:`	ワーク座標オフセット：数回のレポートごとに、また変更の後に。
`Ov:`	送り、早送り、主軸のオーバーライド（%）：数回のレポートごとに、また変更の後に。
`A:`	アクセサリ：`S` または `C` は主軸の時計回りまたは反時計回り、`F` はフラッド、`M` はミスト。
`Sl:`	主軸インバータの負荷。
`DTG:`	軸が停止するまでの残り距離（`$10` のビット 13）。
`H:`	原点復帰済みかどうか。
`T:`	工具交換中の工具。
`TLR:`	基準工具長が設定されています（`$TLR`）。
`MPG:`	ハンドルが制御権を持っている（1）か、持っていない（0）か。
`In:`	直近の `M66` の結果。
`WCS:`、`Sc:`、`D:`	ワーク座標系、スケーリング、旋盤の直径モード（フルレポート）。
`AR`、`AR:`	自動レポートがオフ、またはオンでその間隔（フルレポート）。
`FW:`	ファームウェアの名前（フルレポート）。
`SD:`	コントローラのファイルシステムから実行しているプログラムの進み具合。

コントローラが送るその他の行

行	タイミング
`ALARM:N`	アラーム。N の値はアラームコードにすべて載っています。
`[MSG:…]`	メッセージ：コントローラ自身のもの、またはプログラムの `(MSG,…)`、`(PRINT,…)`、`(DEBUG,…)`。
`[GC:…]`	モーダル状態。`$G` または `0x83` の後。`$10` のビット 9 を立てると、変更のたびに 1回。
`[PRB:x,y,z:1]`	プローブが作動した点と、接触したか（1）しなかったか（0）。
`[TLO:…]`	工具長補正。
`[PARAM:n=value]`	`$#=n` の後。
`[G54:…]` … `[G59.3:…]`、`[G28:…]`、`[G30:…]`、`[G92:…]`、`[T:…]`	`$#` の後：オフセット、保存された位置、工具テーブル。
`>line:ok`、`>line:error:N`	起動時に、スタートアップ行（`$N0`、`$N1`）を実行しました。

`$` コマンド

コントローラ自身も、`$HELP Commands` で一覧を表示します。

コマンド	動作
`$`、`$HELP`、`$HELP <group>`	ヘルプ、コマンド一覧、設定グループ（詳細付き）。
`$$`、`$<n>`、`$<n>=<value>`、`$$=<n>`、`$+`	すべての設定、1つの設定、その設定の変更、1つの設定の詳細、すべての設定をもう一度。
`$SED=<n>`	設定を説明します。

コマンド	動作
`$#`、`$#=<n>`、`$#=<name>`	オフセット、プローブの結果、保存された位置、1つのパラメータ。
`$G`	モーダル状態。
`$I`、`$I+`、`$I=<text>`	コントローラの識別情報（上記）、すべての情報、テキストを識別情報とともに保存。
`$N`、`$N0=<line>`、`$N1=<line>`	スタートアップ行の表示と保存。
`$C`	チェックモード：G コードは検査され、軸は何も動きません。リセットで終了します。
`$X`	アラームの後でロックを解除します。
`$H`、`$HX`、`$HY`、`$HZ`	全軸または 1軸の原点復帰。
`$J=<G-code>`	ジョグ。
`$B`、`$S`、`$O`	ブロックスキップ、シングルブロック、オプショナルストップ：オンまたはオフ。
`$SLP`	スリープ。
`$TLR`、`$TPW`	直近のプロービングを基準工具長として採用する。新しい工具をタッチプレートで測定する。
`$RST=$`、`$RST=#`、`$RST=*`	設定を MC-1 の工場出荷時の値に戻す（原点復帰とリミットが有効、MC-1 のストローク）、オフセットを消去する、その両方とスタートアップ行。
`$EA`、`$EAG`、`$EE`、`$EEG`、`$EG`、`$ES`、`$ESG`、`$ESH`、`$E*`	アラーム、エラーコード、設定グループ、設定を、行または CSV で一覧表示します。
`$SPINDLES`、`$SPINDLESH`	主軸の一覧。
`$PINS`、`$PINSTATE`、`$PORTS`	ボードのピン、補助入出力、シリアルポート。
`$LIM`、`$HSS`、`$LEV`	リミット入力、原点スイッチ、直近の入力イベント。
`$RTC`、`$RTC=<date and time>`	コントローラの時計（ISO 8601）。
`$REBOOT`	コントローラを再起動します。
`$F`、`$F+`、`$F=<file>`、`$F<=<file>`、`$FR`	コントローラのファイルシステム上のプログラムを一覧表示する、1つ実行する、1つ送り返す、最初からやり直す。
`$FD=`、`$FMD=`、`$FRD=`、`$FF=yes`、`$FI`、`$FM`、`$FU`、`$CWD=`、`$PWD`	削除、フォルダーの作成または削除、フォーマット、情報、マウント、アンマウント、作業フォルダー。

すべての設定は設定にあります。

ファイル形式

草稿（ラウンド 14）：本文は執筆中で、節が増える可能性があります。

プログラム

プログラムは、Twinwright 独自カーネルが扱う Grbl 1.1 方言の G コードテキストファイルです。Studio は、そのようなファイルであればどれでも読み込めます。Studio の CAM が書き出すプログラムは、次のような形をしています。

```
(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
```

要素	内容
ヘッダー	コメント：ジョブのタイトル、最初の工具・材質・主な切削条件、プログラムの原点の位置
モード	`G21 G90 G17 G94 G54`：ミリメートル、絶対指令、XY 平面、毎分送り、ワーク座標オフセット G54
`G53 G0 Z…`	機械座標での Z の上端への移動：開始時、工具交換のたび、終了時。その高さは、原点復帰を終えた時点での Z の位置です。
`T… M6`	工具交換。コントローラが新しい工具をツールセッターで測定します。
`M3 S…`、`G4 P…`	主軸をオンにし、そのあと回転数に達するまでドウェルします。水冷主軸のポンプ（`M8`）が先に入ります。
フィーチャのコメント	各フィーチャは、何を削るかを書いたコメントで始まります
移動	`G0`、`G1`、および `I` と `J` を使う円弧 `G2/G3`。穴あけには `G81`、またはペックを使う `G83` を使い、その `R` 平面は上面より 1 mm 上にあります。
終了	`M5`、`M9`、上端への移動、`M30`

コメントは括弧の中に書きます。行に番号は付きません。

Studio は、プログラムの各バージョンを、プログラムフォルダー（ドキュメント/Twinwright Studio/programs）の中の、プログラム名のフォルダーに保存します。

ファイル	内容
`v0001.nc`	保存されたときのままのプログラムのテキスト
`v0001.json`	そのバージョン、時刻（UTC）、作成者、理由、元になったバージョン、ハッシュ、行数
`v0001.ops.json`	各フィーチャが変換された結果（加工工程、加工方式、工具、行、回転数、送り）：段取りシートがこれを読み込みます

ジョブと段取り

ジョブは、CAM がそこからプログラムを作る、部品の記述です。JSON オブジェクトです。

キー	意味
`name`	プログラムの名前。たとえば `bracket.nc`
`title`、`description`	プログラムのヘッダーと段取りシートに入れる文言
`material`	材質。たとえば `6061-T6`。他の規格での名称も認識されます
`tool`	主軸に付いている工具。仕様の書式で書きます（`flat 6 n3 L20`）
`tools`	工具テーブル：`{"1": "flat 6 n3 L20", "2": "drill 3.4 118 L20"}`。省略すると、機械のものを使います。
`cutting`	選ばれた値を上書きする値：`rpm`、`feed`、`ap`、`ae`、`finish_ap`
`safe_z`	部品の上を横切って移動するときの高さ。ワーク原点からの高さをミリメートルで表します
`zero`	ワーク原点の位置：`front-left-top`（デフォルト）または `center-top`
`features`	フィーチャ。削る順に並べます。プログラムの作成を参照してください。
`setups`	`features` の代わりに使います：段取りごとに 1つのオブジェクト
`nest`	`features` の代わりに使います：板材にネスティングした部品

段取りには独自の `features` があり、前の段取りから部品がどのように渡されるかを指定します。

キー	意味
`turn`	`u`：X 軸まわりに裏返す（前と後ろが入れ替わる）。`v`：Y 軸まわり（左と右が入れ替わる）
`fixture`	この段取り以降、部品を別の方法で保持します（`vise`、`clamps`、`tape` …）
`zero`	ワーク原点を取り直します：`front-left-top` または `center-top`。省略した場合、裏返した後や治具を変えた後は `front-left-top`、それ以外は前の段取りの原点になります。
`name`、`tool`、`tools`、`cutting`、`safe_z`	段取り固有のもの。ジョブのものより優先されます

```
{"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}]}
```

ジョブの各バージョンは、それが作ったプログラムの隣に `job-v0001.json` として保存されます。内容は、そのバージョン、時刻（UTC）、作成者（`operator` または `ai:<client>`）、メモ、それが作ったプログラムのバージョン、そしてジョブそのものです。`setups` を使ったジョブには、各段取りのプログラム、作られたバージョン、そのハッシュも記録されます。

記録（MCAP）

`--mcap FILE` は、実行を MCAP ファイル（mcap.dev、フォーマットのバージョン 0）に記録します。データの種類ごとにチャンネルがあり、メッセージは JSON で、各チャンネルに JSON スキーマが付きます。チャンク化も圧縮もされておらず、末尾に統計と CRC を含むサマリーが付きます。Studio のテストは、記録したファイルを MCAP 仕様と突き合わせて読み戻します。PlotJuggler や Foxglove のような MCAP を読み込むツールで開けることを想定していますが、まだ試していません。

- `--rec-period MS` は、サンプリング対象のチャンネルを書き込む間隔を設定します（指定しなければ 1 ms）。
- 記録は実行に何も影響しません。記録した実行と、記録しない同じ実行は、ビット単位で同一です。
- `--play FILE.mcap` は、Studio で記録を再生します。タイムラインがあり、再生、一時停止、シーク、0.25–10倍の速度に対応します。再生中は何もシミュレートされません。

時刻

各メッセージのログ時刻は、ツインの仮想時間で、電源投入からのナノ秒です。ほとんどのチャンネルには、同じ時刻を秒で表した `t` も入っています。サンプリング対象のチャンネルは、その中の何かが変わったときだけメッセージを書き込みます。

ヘッダー

`mc1sim` という名前のメタデータレコードに、その実行が何であったかが書かれています。

キー	意味
`machine`	機械の名前
`drive_model`	実行に使ったドライバモデル：`L0`、`L1`、`L2`、`L2/3`（3つの質量）、`L3`
`f_cpu`	コントローラのクロック（Hz）
`nvs`	実行開始時のコントローラの保存設定のハッシュ（16進数 16桁）
`sample_period_ms`	サンプリング周期
`time`	`virtual, ns from power-on`

チャンネル

チャンネル	書き込み	フィールド	型	単位	意味
`/machine`	サンプリング	`pos`	number[3]	mm	各軸の実際の位置：機構そのものの位置で、コントローラが把握している位置ではありません
		`mpos`	number[3] または null	mm	コントローラの機械位置（MPos）。コントローラがオフの間は null
		`lost`	number[3]		各軸が失ったステップ数
		`feed`	number	mm/min	コントローラの現在の送り速度
		`tip`	number[3]	mm	ワークに対する工具先端の位置：ツインの誤差モデルがオンの場合は、その誤差の分だけずれた位置。オフの場合は `pos` と同じ
`/drives`	サンプリング（L1 以上のドライバモデルがある場合）	`ferr_um`	number[3]	µm	各ドライバの追従誤差
		`state`	string[3]		各ドライバの状態
		`km1`	boolean		コンタクタ KM1 が閉じている
		`bus48`	boolean		ドライバの 48 V 電源がオン

チャンネル	書き込み	フィールド	型	単位	意味
`/spindle`	サンプリング（主軸インバータがある場合）	`f_out`	number	Hz	ドライバの出力周波数
		`rpm`	number	min^{-1}	主軸の回転数
		`amps`	number	A	ドライバの出力電流
		`load_pct`	number	%	ドライバが報告する負荷
		`torque_nm`	number	N·m	モーターのトルク
		`bus_v`	number	V	ドライバの DC バス電圧
		`cmd_rpm`	number	min^{-1}	コントローラが指令する回転数
		`run`、`pump`	boolean		ドライバが運転中、ポンプがオン
		`power`	string		ドライバ自身の電源：`ready`、`boot`、`off`
		`fault`	string		ドライバのフォルト。ドライバの表示どおり
`/controller`	変化時	`state`	string		コントローラの状態（オフの間は `Off`）
		`alarm`	integer		発生中のアラーム（0：なし）
		`homed`	integer		原点復帰済みの軸。1 ビットずつ：X が 1、Y が 2、Z が 4
		`limits`	integer		アクティブなリミット入力。1 ビットずつ
		`probe`、`estop`、`motor_fault`、`door`	boolean		プローブ、非常停止、モーター異常、安全扉の各入力（MC-1 に安全扉はありません）
		`on`	boolean		コントローラに電源が入っている
`/power`	変化時	`mains`、`v24`、`mcu`、`km1`、`bus48`	boolean		商用電源、24 V 電源、コントローラの動作、KM1、48 V 電源
		`vfd_mains`、`vfd`、`pump`	boolean、string、boolean		主軸インバータの商用電源、その状態（`none`、`ready`、`boot`、`off`）、ポンプ
`/serial/in`	届いたとき	`text`	string		コントローラが受信したバイト。制御文字は `\u` エスケープで表します
`/serial/out`	届いたとき	`line`	string		コントローラが書き出した各行
`/events`	発生時	`text`	string		イベントログの各エントリ：マーク、電源、ドライバ、主軸インバータ、衝突、断線
`/cut`	サンプリング（素材がある場合）	`cutting`	boolean		材料が除去されている。除去されていない間は、`rpm` と `removed_mm3` 以外の下の値は 0
		`mrr_mm3_s`	number	mm^3/s	材料除去率
		`ap_mm`、`ae_mm`	number	mm	軸方向の切込み深さと、半径方向の切込み幅
		`fz_mm`	number	mm	1刃あたりの送り
		`rpm`	number	min^{-1}	主軸の回転数
		`fx_n`、`fy_n`、`fz_n`	number	N	機械の各軸方向の切削抵抗
		`torque_nm`、`power_w`	number	N·m、W	切削トルクと切削動力
		`kc_j_mm3`	number	J/mm^3	比切削エネルギー
		`removed_mm3`	number	mm^3	ここまでに除去された材料の量
`/input`	届いたとき	`line`	string		セッションの入力。`--record` が書き出すとおりに、1行ずつ（セッションログを参照）
`/shadow`	届いたとき（シャドーモード）	`host_s`	number	s	コンピューターの時刻
		`dir`	string		`t` は実機へ、`r` は実機から
		`data`	string		バイト列。各バイトを U+0000–U+00FF の 1文字で表します。1メッセージあたり最大 1 KB
`/ui/state`、`/sound`	プレーヤー用				ビューの状態フレーム（変化している間は毎秒 30回）と、サウンドのフレーム（変化したとき、仮想時間で 10 ms ごと）。Studio 独自のもので、フィールドはバージョンによって変わることがあります。

ピントレース（VCD）

`--vcd FILE` は、コントローラのピンを VCD ファイル（IEEE 1364）として書き出します。VCD は GTKWave や PulseView で読み込めます。スコープ `mcu` にはコントローラのピンが入り、各変化はそれが起きたクロックティックに記録されます（ステップピンは `$0` マイクロ秒のパルスとして記録）。スコープ `machine` には、コントローラの外にある信号が入ります：電源、KM1、主軸インバータ、ポンプ、非常停止ボタン、ツールセッター、5つのスイッチ。タイムスケールは 1 ns です。

セッションログ

`--record FILE` は、セッションの入力を、それぞれの仮想時間とともに書き出します。送信ソフトが送ったもの、リアルタイムコマンド、ビューのコマンドが対象です。`--replay FILE` は、同じバージョンの Studio と同じ機械ファイルを使って、そのセッションをビット単位で同一に再実行します。`--replay FILE --branch S` は、S 秒まで再生し、そこから先はライブで続けます。`--record` と併用すると、分岐は独自のログを持ち、そのログは親のログから始まります（`# branch <parent> <ticks>`）。ログはテキストで、その行は Studio 独自のものであり、バージョンによって変わることがあります。

ワールドアーカイブ

ツインがセッションの間も保持するもの。実機の電源を切って入れ直したときと同じです：軸の位置、切削済みの部分を含む素材、主軸の工具、コントローラのメモリとボードの時計、そして熱（機械の電源が切れている間に冷めます）。コントローラは、コールドスタートで起動します。機械は、あらためて原点復帰します。

ウィンドウは、機械ごとのワールドを状態フォルダー（Windows では `%LOCALAPPDATA%\Twinwright\Studio`）に保存します。最初の機械のものは `world`、それ以外の機械のものは `machines\<id>\world` です。`--world DIR` は別のフォルダーを使います。`--world-reset` は工場出荷時の状態から始めます。バッチ実行、テスト、リプレイ、再生される記録、AI のバックグラウンドの機械は、`--world` を指定しない限り、ワールドを保持しません。ワールドを開けるのは、一度に 1つのプロセスだけです。2つ目のプロセスは工場出荷時の状態から始まり、何も保存しません。

ファイル	内容
`world.json`	セッション終了時の状態（下記）。まず `world.json.tmp` に書き込み、その後で名前を変更します
`stock.snap`	素材のセル（削った部分も含む）（バイナリ、`TWSTOCK1`）
`eeprom.bin`	コントローラの保存メモリ：設定、オフセット、工具データ。変更のたびに即座に書き込まれます

`world.json`：

キー	意味
`world`	フォーマットのバージョン：1
`saved_at`、`saved_epoch`	保存された時刻：UTC、および 1970年からの秒数
`version`	保存した Studio のバージョン
`pos`	軸の位置、mm（X、Y、Z）
`tool`	主軸にある工具（仕様の書式で。`flat 6 n3 L20`）
`tool_table`	ジョブの工具テーブル：`{"1": "flat 6 n3 L20", ...}`
`tool_auto`	ツインが工具交換のときにオペレータの役を務めます：テーブルから工具を取り付け、サイクルスタートを押します
`setup`	ワークの保持方法と素材。ウィンドウの「段取りを保存」が書き出す形式（null：なし）
`cells`	`stock.snap` が素材のセルを保持している
`thermal`	熱の状態：`screws`（各ねじのフレームを超える分）、`frame`（20 °C を超える分）、`ambient`（室温の 20 °C を超える分）、`part_heat` と `part_offset`（ワークのフレームを超える分：切削で加わった熱と、与えられたオフセット）はいずれも K、`spindle_um` は主軸の伸びで µm
`rtc`	ボードの時計の読み値（秒）
`probe_cal`	最後に校正されたときのタッチプローブ：ボールの直径、有効半径、ロビングとその位相、リングゲージの直径、最大偏差、校正の日時と実施者（校正済みの場合のみ）
`odometer`	各軸の `travel_mm` と `reversals`、`spindle_h`（主軸の運転時間）と `on_h`（プログラムの稼働時間）を、セッションをまたいで累計した値

次の起動時には、軸は前回の位置にあります。同じ素材がセルごと戻り、工具と工具テーブルも戻ります。熱は、機械の電源が切れていた時間の分だけ冷め、各部分はそれぞれの時定数で平衡状態に向かいます。ボードの時計は進み続けています。`--world-off S` は、機械が電源オフだった時間を指定します（テスト用。指定しない場合は、保存からの経過時間）。起動時に指定した `--pos`、`--tool`、`--stock`、`--setup`、`--part` は、アーカイブより優先されます。

ライセンスファイル

ライセンスファイル（`.twlic`）は、UTF-8 の JSON オブジェクトで、Ed25519 で署名されています。Studio は、ネットワークを使わずに、コンピューター上でこれを検証します。ファイルの置き場所と使い方：ライセンス、シート、アクティベーション。

この例は、共有のテストベクターからのものです。テスト用の鍵（`kid` が `test`）で署名されており、リリースビルドはこれを受け付けません。

```
{"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"}
```

フィールド

フィールド	型	意味
`v`	整数	フォーマットのバージョン：1。Studio は、ほかのバージョンのファイルを無効として扱います。
`kid`	文字列	ファイルに署名した Twinwright の鍵のうちどれか

フィールド	型	意味
`id`	文字列	ライセンスの番号：`TWL-` に英数字 10文字が続きます。更新のときに送信されます。
`edition`	文字列	`maker` または `pro`
`kind`	文字列	`subscription`、`perpetual`（買い切り）、`owner`（MC-1 オーナー）
`seats`	整数	シート数
`holder`	文字列	ライセンスの名義人：個人または組織。メールアドレスは決して入りません。
`issued`	日時	ファイルが署名された時刻。UTC、`YYYY-MM-DDTHH:MM:SSZ`
`valid_until`	日時、または null	サブスクリプション：支払い済み期間の終わり。買い切り：最初のファイルでは返金期間の終わり、恒久的なファイルでは null。
`renew_after`	日時、または null	Studio がファイルの更新を試み始める時刻
`updates_until`	日付、または null	買い切り、MC-1 オーナー：この日までにリリースされたバージョンで、有料機能が使えます
`fp`	文字列の配列	ライセンスが紐づけられているコンピューター：そのフィンガープリントのハッシュ。1シートにつき最大 3台
`mc1_serial`	文字列、または null	MC-1 オーナー：機械のシリアル番号。Studio は、実行時に機械を確認しません。
`sig`	文字列	ほかのフィールドの Ed25519 署名。パディングなしの base64url

Studio が知らないフィールドも署名対象のバイト列に含まれます。Studio は、その意味を無視します。

署名されるバイト列

署名の対象は、`sig` を除いたファイルを、次の正規形で書き出したものです。

- キーは Unicode コードポイント順。
- 空白なし。
- 文字列は JSON の最短のエスケープ。
- 整数は小数点なし。それ以外の数値は使わず、同じキーを 2回使わない。

Studio は、署名を検証する前に、ファイルをこの形式で書き出します。そのため、ファイル内のスペース、改行、キーの順序は関係ありません。値が変更されると、検証は失敗します。

フィンガープリントのハッシュ

ハッシュは、`twinwright-fp-v1` というバイト列、ゼロバイト、コンピューターの ID をこの順に並べたものの SHA-256 の、先頭の 16進数 32桁です。

システム	コンピューターの ID
Windows	`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography` の下の値 `MachineGuid`
Linux	`/etc/machine-id`

どちらもない場合、Studio はランダムな ID を作り、ライセンスフォルダーに保存します（`machine.id`）。これを削除すると、そのコンピューターは別のコンピューターになります。ID 自体がコンピューターの外に出ることはありません。

たとえば、ID `8b1c3a0e-5f2d-4c7a-9e61-2d0f4b7a1c55` からは、ハッシュ `0a2d31bb82176516415efa7b3a340562` が得られます。

要求コード

オフラインアクティベーション用の要求コードは、正規形の UTF-8 JSON `{"fp": <the hash>, "version": <Studio's version>}` を、`=` パディングなしの base64url にしたものです。たとえば、ハッシュ `0123456789abcdef0123456789abcdef` とバージョン 1.0.0 からは、次のコードが得られます。

```
eyJmcCI6IjAxMjM0NTY3ODlhYmNkZWYwMTIzNDU2Nzg5YWJjZGVmIiwidmVyc2lvbiI6IjEuMC4wIn0
```

`TwinwrightStudio.exe --licence-request` は、このコンピューターの要求コードを出力します。

試用ファイル

試用はライセンスファイルの一種で、ほかのものと同じく Twinwright のライセンスサーバーが署名します：`"kind": "trial"`、`fp` にこのコンピューターのハッシュ、`valid_until` は試用の開始から 30 日後です。更新されることはありません。手で書き換えられた試用ファイルは、有効ではなくなります。

機械アーカイブ

機械のアーカイブのすべてのキーと、MC-1 の値：機械プロファイルのスキーマ。

Twinwright 独自カーネルで使う送信ソフト

このページは、Studio ではなく、自分で用意した G コード送信ソフト（bCNC、gSender、Universal G-code Sender）から MC-1 を動かしたい方のためのものです。MC-1 のコントローラは Twinwright 独自カーネルで、Grbl 1.1 シリアルプロトコルおよび現在の送信ソフトが使う拡張と互換性があります。そのため、これらの送信ソフトはこのコントローラで動作します。ただし、以下に述べるわずかな設定と注意点があります。

ここに書かれていることはすべて、MC-1 のデジタルツインでテストしました。デジタルツインは、実機と同じコントローラのコードを実行します。実機での確認は、これからです。テストしたバージョン：bCNC 0.9.16、gSender 1.6.4、UGS Platform 2.1.27。

接続

接続先	方法
Studio のウィンドウ内のツイン	Studio は `localhost:2323` で、送信ソフトを一度に 1つだけ受け付けます（このコンピューターからのみ。ネットワークから接続するには、Studio を `--bind any` で起動します）。bCNC：ポート `socket://localhost:2323`。gSender：インサネット、アドレス `127.0.0.1`、ポート `2323`。UGS：`tcp://localhost:2323`。
MC-1 本体	ネイティブ USB：コントローラはシリアルポートとして認識されます（Windows では COM*n*）。そのポートを選んでください。ネイティブ USB ではボーレートは関係ありませんが、指定を求める送信ソフトがあります。115200 で動作します。

コントローラ自身が伝えること：

- 起動メッセージは `Grbl 1.1f ['$' or '$HELP' for help]` で、従来の Grbl コントローラと同じです。送信ソフトは、「Grbl」で始まる起動メッセージを探して、コントローラが再起動したことを知ります（3節を参照）。
- 名前は `$I` の応答に含まれます：`[VER:Twinwright 1.1f.…]` と `[FIRMWARE:Twinwright]`。ファームウェアを表示する送信ソフトには、「Twinwright」と表示されます。

送信ソフトでのファームウェアの選択

送信ソフト	設定	理由
bCNC 0.9.16	Controller：**GRBL1**	bCNC はファームウェアの名前を読み取りません。GRBL1 でテストしました。
gSender 1.6.4	Default firmware：**Grbl** 以外の選択肢	gSender は、`$I` の中の名前でファームウェアを認識します。このコントローラは独自の名前を返すため、gSender は既定値として設定したファームウェアを使います。その選択肢でテストしました。
UGS 2.1.27	Firmware：**GRBL**	テストした内容：設定、プログラム、アラーム、リセット後の原点復帰が動作し、追加の Soft Reset は不要です（3節）。UGS のもう一方のファームウェア選択肢も動作しますが、リセットの後にもう一度 Soft Reset が必要です。ジョグはその選択肢でのみテストしました。

アラームまたはリセットの後

- **非常停止。** MC-1 の非常停止は、電源コンタクタを開きます。非常停止を解除し、機械の Start ボタンを押してから、送信ソフトでリセットし、原点復帰してください。Start を押すまで、アラーム（ALARM:10）は解除されません。
- **ALARM:11（原点復帰が必要）。** 原点復帰（`$H`）してください。`$X` によるロック解除は拒否されます（error:46）。
- **ALARM:2（ソフトリミット）。** プログラムが、移動範囲の外への移動を指示しました。原点復帰を終えると Z は移動範囲の上端（機械座標 Z 0）にあるため、それより上への移動（`G53 G0 Z5` など）は ALARM:2 を発生させます。リセットしてから、原点復帰またはロック解除を行ってください。
- **bCNC** は、どのリセットの後でも復帰します。「Grbl」で始まる起動メッセージを見て、送信途中だった行を忘れるため、`$H` が送信されます。
- **bCNC の Stop 後。** bCNC の Stop は、フィードホールドを行い、続けてリセットします。その後、bCNC は、それまでのモード（座標系、単位、工具長補正）を 1行か 2行でコントローラに送り返しますが、いくつかの場合には、ほかの Grbl コントローラと同じように、コントローラがそれを拒否します：工具長補正（`G43.1`）が有効なときは、その行に error:24 が返ります（bCNC は最初の Stop の後に自分で `G43.1Z...` を送るため、2回目の Stop 以降は必ずこうなります）。`G80` が有効なときは、`G43.1Z...` に error:31 が返ります。スケーリング（`G51`）が有効なときは、`G51:X` に error:1 が返ります。エラーの後は、空行、`$` コマンド、またはリセットが来るまで、続くすべての G コードの行に同じエラーが返ります。そのため、Stop の後は、ジョグしたり作業を続けたりする前に、`$G` と `$#` で座標系、単位、工具長補正を確認してください。誤っていた場合は、ワーク原点を設定し直し、工具長を測り直してください（ツインでテスト、2026-10-06）。
- **UGS のもう一方のファームウェア選択肢の場合。** UGS 自身が送ったものではないリセットの後（たとえば、Studio のウィンドウからのリセットや、コントローラが自分で再起動した場合）は、Home の前に、UGS で **Soft Reset をもう一度**押してください。それまでは、Home は何もしません。その選択肢に設定した UGS は、別の起動メッセージだけを再起動として扱います。UGS 自身の Soft Reset ボタンによるリセットは問題ありません。MC-1 本体では、このようなリセットはまれです：リセット入力がなく、USB ホストも 1つだけだからです。ファームウェアを「GRBL」に設定していれば、これは起こりません。そのようなリセットの後、UGS は CONNECTING と表示し、Home はすぐに動作します（ツインでテスト、2026-10-06）。

工具交換（M6）

3つの送信ソフトはどれも、デフォルトの設定のままでは、MC-1 で工具を交換して新しい工具を測定することができません。

送信ソフト	デフォルト	対処
bCNC	すべての M6 の行を飛ばします：次の工具による切削が、まだ主軸に付いている工具で行われます。	工具交換を **Manual tool change (TLO)** に設定し、MC-1 のツールセッターで測定します（5節）。
gSender	工具交換が「Ignore」：M6 はコメントとして送られ、プログラムは止まりません。	工具交換を、MC-1 のツールセッターでの **Fixed Tool Sensor** に設定します（5節）。
UGS	M6 を自分で処理します：退避し、主軸を止めて、工具の交換を求めますが、測定は一切行いません。そのため、長い工具では削り過ぎになり、実機では部品に衝突することがあります。	プログラム 1つにつき工具 1本で実行し、それぞれの前に工具長またはワーク原点を設定するか、工具交換を含むプログラムには Studio を使います。

UGS と M6 についての補足：

- コメントの中の「M6」でも止まります。FreeCAD の `( M6 T1 )` がその例です。
- 工具交換の直後に 2つ目の M6 が来ると、UGS の処理をすり抜けてコントローラに届くことがあります。すると、コントローラは Tool 状態のまま、UGS が決して送らない確認を待ち続け、UGS のウィンドウのボタンはどれも効かなくなります。MC-1 では、非常停止だけがこの状態を解除できます：非常停止を押し、解除し、Start を押してから、UGS で Soft Reset と Home を行います。
- UGS でプログラムを実行している間に、UGS が M6 に達する前にコントローラがリセットされると、UGS がその実行を完了と報告することがありますが、それでも工具交換のダイアログは表示されます。これを閉じると、プログラム全体が最初からもう一度始まることがあります。実行の途中でリセットした後は、先に進む前に、UGS でファイルを閉じてください。

MC-1 のツールセッターでの工具長の測定

ツールセッターはテーブルの左奥にあります。その真上の位置が `G59.3` に保存されています。

- **bCNC、Manual tool change (TLO)。** 工具交換位置はツールセッターの上。プローブ位置は `G59.3`。プロービング距離は 50 mm、シークは 200 mm/min、スローは 25 mm/min（MC-1 の `$342`–`$344`）、コマンドは G38.2。プログラムを実行する前に、「Calibrate」で最初の工具を測定してください。その後、bCNC は新しい工具ごとに `G43.1` を設定します。
- **gSender、Fixed Tool Sensor。** センサの位置は `G59.3`。ウィザードで最初の工具を測定してください。その後、gSender は工具長補正を使う代わりに、新しい工具ごとにワーク Z をずらします（`G10 L20`）。gSender は、コントローラが報告するトリガー点ではなく、接触の後に軸が止まった位置から長さを求めるため、交換のたびに少しずれることがあります。ツインでは、工具交換を 1× の速度で実行してください：gSender は決まった短い待ち時間の後でその位置を読み取るため、より速く動いているツインでは、停止前の位置を返してしまうことがあります。
- 工具交換を始めるときは、**Z の上端より 0.1 mm 以上下から始めてください**（機械座標 Z が −0.1 以下）。どちらの送信ソフトも、測定の後でその高さに戻ります。上端ちょうどから始めると、測定のわずかなばらつきによって、その戻り位置がソフトリミットを超えてしまい、切削を始める前に ALARM:2 で実行が止まります。

タッチプローブでのワーク原点の設定

- **UGS**（そのプローブモジュール）は、コントローラが報告するトリガー点（`[PRB:]`）から原点を求めます。これが正しい方法です。
- **gSender**（3D プローブ）と **bCNC**（プローブのページと、その Z=0 / X / Y ボタン）は、接触の後に軸が止まった位置に原点を設定します。機械は、トリガーの後、止まるまでに必ず少し行き過ぎます。最後の接触が速いほど、行き過ぎは大きくなります。最後の接触を遅くするか、トリガー点を使う Studio のプローブサイクルで原点を設定してください。
- **UGS の穴中心**（プローブモジュールの Hole Center）：プローブを置いた位置から、-X、+X、-Y、+Y の順に、それぞれ入力した穴径の半分の距離だけプロービングし、低速での接触のトリガー点の中間に中心を設定します。どこまで届くかはその径に依存します：開始位置は、中心から最大で（入力した径 - 実際の穴径）/ 2 + ボールの半径までです。穴の実際の径を入力した場合、それはボールの半径だけです（2 mm のボールなら 1 mm）。プローブを目分量で置くことを見込んで、数 mm 大きく入力してください（20 mm の穴なら 24 から 26）。
- **UGS の穴中心が壁に届かないとき**（所見 192、UGS 自身の動作）：そのプロービングは ALARM:5 で終わり、UGS は停止します。ただし、UGS はプロービングの前に、開始点に G54 の X0 Y0 を設定しており、サイクルが完了しないため、G54 はそこに残ります。ロック解除の後、切削の前に、ワーク原点を設定し直してください。プローブのサイクルも途中のまま残るため、もう一度プロービングする前に、切断してから再接続してください。また、コントローラが接触していないと報告した点（`[PRB:...:0]`）も、接触として扱います。
- 先にプローブを校正してください（Studio のプローブサイクル：リングゲージ）。校正していないプローブは、すべての穴を大きく、すべてのボスを小さく、プリトラベルの 2倍だけ読み取ります。

そのほかの注意点

- **ネットワーク経由の bCNC。** Windows では、bCNC は TCP 接続を約 2秒ごとにしか読み取らないため、表示が 2秒遅れていることがあります。bCNC の「Idle」は、機械が止まったことを意味しません。USB 接続には、この遅れはありません。
- **gSender、M0。** プログラムの最初の 10行以内にある M0 は、一時停止しません（gSender がコメントに変えてしまいます）。M1 は必ず一時停止します。
- **gSender、連続ジョグ。** ジョグボタンを押し続けると、MC-1 では拒否されます（error:15）：gSender は、移動範囲を超える長いジョグを要求するためです。ステップ単位でジョグしてください。一部の Grbl コントローラでは、`$40=1`（ジョグコマンドの制限）にすると、長いジョグは拒否されずにソフトリミットで終了します。MC-1 のコントローラは、この設定にまだ対応しておらず、引き続き拒否します（ツインで gSender を使って試しました、2026-10-06）。
- **UGS、長い行。** UGS は 127文字より長い行を送らず、実行はメッセージなしで止まります。行を短くしてください（UGS は先にコメントを取り除きます）。
- **UGS、M30。** UGS は M30 を取り除きます。M30 で主軸とクーラントを止めることに頼っているプログラムでは、それらが動いたままになります：プログラムは、M30 の前に M5 と M9 で終えてください。
- **UGS、コメント。** 1行に 2つのコメントがあると、UGS は、その間にあるコードも取り除きます。
- **プログラムの実行中の切断。** コントローラは、すでに受け取っている行を実行し続けます。gSender と UGS は、切断するときに停止を送らないため、機械は主軸を回したまま動き続けます。切断する前に、実行を止めてください。bCNC は、接続を閉じる前に、ホールドとリセットを行います。
- **ASCII 以外の文字を含むコメント**（中国語、「Ø」、「–」など）：3つの送信ソフトはすべて、送信前にコメントを取り除くため、これらの文字がコントローラに届くことはありません。コントローラ自体は、コメントや `(MSG,...)` の中の UTF-8 テキストを、テキストとして読み取り、リアルタイムコマンドとして扱うことはありません。