

5Axisgui(1)	2
AT_PID(9)	3
AXIS(1)	7
axis-remote(1)	8
DEBUGLEVEL(1)	9
ENCODER(9)	10
GLADEVCP_DEMO(1)	13
GMOCCAPY(1)	14
GREMLIN_VIEW(1)	15
gs2_vfd(1)	17
GSCREEN(1)	20
HALUI(1)	21
IOCONTROL(1)	30
LinuxCNC(1)	32
MILLTASK(1)	34
MOTION(9)	37
PID(9)	51
SPINDLE(9)	55

**NAME**

5axisgui – Vismach Virtual Machine GUI

**DESCRIPTION**

5axisgui は、LinuxCNC 用のサンプル VismachGUI の 1 つです。

詳細については、LinuxCNC のメインドキュメントを参照してください。

<http://linuxcnc.org/docs/html/gui/vismach.html>

**SEE ALSO****LinuxCNC(1)**

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr/share/doc/LinuxCNC/ にあります。

**HISTORY****BUGS**

現時点では不明です。

**AUTHOR**

この man ページは、LinuxCNC Enhanced MachineController プロジェクトの一部として AlexJoni によって作成されました。

**REPORTING BUGS**

<https://github.com/LinuxCNC/linuxcnc/issues> でバグを報告してください

**COPYRIGHT**

Copyright © 2020 andypugh.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY

or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についてもまたは特定の目的への適合性。

**NAME**

at\_pid – proportional/integral/derivative controller with auto tuning

**SYNOPSIS**

**loadrt at\_pid [num\_chan=num | names=name1[,name2...]]**

**DESCRIPTION**

at\_pid は、古典的な比例/積分/微分コントローラーであり、サーボモーターやその他の閉ループアプリケーションの位置または速度のフィードバックループを制御するために使用されます。

at\_pid は、最大 16 個のコントローラーをサポートします。実際にロードされる数は、モジュールがロードされるときに num\_chan 引数によって設定されます。または、names = と一意の名前をコンマで区切って指定します。

num\_chan = および names = 指定子は相互に排他的です。num\_chan = も names = も指定されていない場合、デフォルト値は 3 です。

debug が 1 に設定されている場合（デフォルトは 0）、いくつかの追加の HAL パラメーターがエクスポートされます。これはチューニングに役立つ場合がありますが、それ以外の場合は不要です。

at\_pid には自動調整モードが組み込まれています。これは、プロセスを特徴付けるリミットサイクルを設定することによって機能します。このことから、Pgain / Igain / Dgain または Pgain / Igain / FF1 は、Ziegler-Nichols を使用して決定できます。FF1 を使用する場合は、出力がユーザー単位/秒になるようにスケーリングを設定する必要があります。

オートチューニング中は、コマンド入力を変更しないでください。リミットサイクルは、指令された位置の周りに設定されます。自動チューニングを開始するために初期チューニング値は必要ありません。オートチューニングを開始する前に設定する必要があるのは、**tune-cycles**、**tune-effort**、**tune-mode** のみです。オートチューニングが完了すると、チューニングパラメータが設定されます。LinuxCNC から実行している場合、次のエラーを回避するために、調整対象の軸の FERROR 設定をリミットサイクル振幅よりも大きくする必要があるので、緩める必要がある場合があります。

オートチューニングを行うには、以下の手順で行ってください。調整する軸を、移動の中心近くのどこかに移動します。tune-cycles（ほとんどの場合デフォルト値で問題ありません）と tune-mode を設定します。tune-effort を小さい値に設定します。enable を true に設定します。tune-mode を true に設定します。tune-start を true に設定します。振動が発生しない場合、または振動が小さすぎる場合は、tune-effort をゆっくりと増やします。オートチューニングは、enable または tune-mode を false に設定することでいつでも中止できます。

**NAMING**

ピン、パラメーター、および関数の名前には、次の接頭辞が付いています。

**pid.N.** for N=0,1,...,num-1 when using **num\_chan=num**

**nameN.** for nameN=name1,name2,... when using **names=name1,name2,...**

pid.N. 以下の説明にフォーマットを示します。

**FUNCTIONS**

**pid.N.do-pid-calcs** (uses floating-point)

Does the PID calculations for control loop N.

## PINS

**pid.N.command** float in

制御ループに必要な（コマンドされた）値。

**pid.N.feedback** float in

エンコーダーなどのセンサーからの実際の（フィードバック）値。

**pid.N.error** float out

コマンドとフィードバックの差。

**pid.N.output** float out

モーターなどのアクチュエーターに送られる PID ループの出力。

**pid.N.enable** bit in

true の場合、PID 計算を有効にします。false の場合、出力はゼロになり、すべての内部積分器などがリセットされます。

**pid.N.tune-mode** bit in

true の場合、自動調整モードを有効にします。false の場合、通常の PID 計算が実行されず。

**pid.N.tune-start** bit io

true に設定すると、自動チューニングを開始します。オートチューニングが完了するとクリアされます。

## PARAMETERS

**pid.N.Pgain** float in

比例ゲイン。結果は、エラーに Pgain を掛けたものである出力への寄与になります。

**pid.N.Igain** float in

積分ゲイン。結果は、エラーに Igain を掛けた積分である出力への寄与になります。たとえば、10 秒間続いた 0.02 のエラーは 0.2 の積分エラー（errorI）になり、Igain が 20 の場合、積分項は出力に 4.0 を追加します。

**pid.N.Dgain** float in

微分ゲイン。結果は、エラーの変化率（導関数）に Dgain を掛けたものである出力への寄与になります。たとえば、0.2 秒で 0.02 から 0.03 に変化したエラーは、0.05 のエラー導関数（errorD）になり、Dgain が 5 の場合、導関数項は出力に 0.25 を追加します。

**pid.N.bias** float in

バイアスは、出力に追加される一定量です。ほとんどの場合、ゼロのままにしておく必要があります。ただし、サーボアンプのオフセットを補正したり、垂直方向に移動するオブジェクトの重量のバランスをとったりすると便利な場合があります。出力の他のすべてのコンポーネントと同様に、PID ループが無効になると、バイアスはオフになります。PID ループが無効になっている場合でもゼロ以外の出力が必要な場合は、外部 HALsum2 ブロックを追加する必要があります。

**pid.N.FF0** float in

ゼロ次フィードフォワード項。FF0 にコマンド値を掛けたものが出力に寄与します。位置ループの場合、通常はゼロのままにしておく必要があります。速度ループの場合、FF0 は摩擦またはモーターカウンタ EMF を補正でき、適切に使用すればより良いチューニングが可能になる場合があります。

**pid.N.FF1** float in

一次フィードフォワード項。FF1にコマンド値の導関数を掛けた出力への寄与を生成します。位置ループの場合、寄与は速度に比例し、摩擦またはモーターCEMFを補正するために使用できます。速度ループの場合、加速度に比例し、慣性を補正できます。どちらの場合も、適切に使用すると、チューニングが向上する可能性があります。

**pid.N.FF2** float in

2次フィードフォワード項。FF2にコマンド値の2次導関数を掛けた出力への寄与を生成します。位置ループの場合、寄与は加速度に比例し、慣性を補正するために使用できます。速度ループの場合、寄与はジャークに比例し、通常はゼロのままにする必要があります。

**pid.N.deadband** float in

「許容可能な」エラーの範囲を定義します。エラーの絶対値が不感帯よりも小さい場合、エラーがゼロであるかのように扱われます。本質的に量子化されたエンコーダーなどのフィードバックデバイスを使用する場合、コマンドが2つの隣接するエンコーダー値の間にある場合に制御ループが前後にハンチングするのを防ぐために、不感帯は半分のカウントよりわずかに大きく設定する必要があります。エラーの絶対値が不感帯よりも大きい場合、不感帯のエッジでの伝達関数のステップを防ぐために、ループ計算を実行する前に不感帯の値がエラーから差し引かれます。(バグを参照してください。)

**pid.N.maxoutput** float in

出力制限。maxoutputがゼロでない限り、出力の絶対値がmaxoutputを超えることはできません。出力が制限されている場合、ウィンドアップとオーバーシュートを防ぐために、エラー積分器は積分する代わりに保持します。

**pid.N.maxerror** float in

P、I、およびDに使用される内部エラー変数の制限。エラーが大きい場合(コマンドがステップ変更を行う場合など)に、高いPgain値が大きな出力を生成しないようにするために使用できます。通常は必要ありませんが、非線形システムを調整するときに役立ちます。

**pid.N.maxerrorD** float rw

エラー導関数の制限。Dgain項で使用されるエラーの変化率は、値がゼロでない限り、この値に制限されます。Dgainの影響を制限し、コマンドやフィードバックのステップによる大きな出力スパイクを防ぐために使用できます。通常は必要ありません。

**pid.N.maxerrorI** float rw

エラー積分器の制限。Igain項で使用されるエラー積分器は、ゼロでない限り、この値に制限されます。インテグレータのウィンドアップと、エラーの発生中または発生後に発生するオーバーシュートを防ぐために使用できます。通常は必要ありません。

**pid.N.maxcmdD** float rw

コマンド派生物の制限。FF1で使用されるコマンド導関数は、値がゼロでない限り、この値に制限されます。コマンドにステップ変更がある場合に、FF1が大きな出力スパイクを生成するのを防ぐために使用できます。通常は必要ありません。

**pid.N.maxcmdDD** float rw

コマンドの二次導関数の制限。FF2で使用されるコマンドの2次導関数は、値がゼロでない限り、この値に制限されます。コマンドにステップ変更がある場合に、FF2が大きな出力スパイクを生成するのを防ぐために使用できます。通常は必要ありません。

**pid.N.tune-type** u32 rw

0 に設定すると、Pgain / Igain / Dgain が計算されます。1 に設定すると、Pgain / Igain / FF1 が計算されます。

**pid.N.tune-cycles** u32 rw

プロセスを特徴づけるために実行するサイクル数を決定します。tune-cycles は、実際には半サイクルの数を設定します。すべてのサイクルの平均が使用されるため、サイクルが多いほど、より正確な特性評価が得られます。

**pid.N.tune-effort** float rw

プロセスでリミットサイクルを設定する際に使用される効果を決定します。tune-effort は、maxoutput よりも小さい正の値に設定する必要があります。小さなものから始めて、最大モーター電流のかなりの部分を使用される結果になる値まで作業します。値が小さいほど、リミットサイクルの振幅は小さくなります。

**pid.N.errorI** float ro (only if debug=1)

エラーの積分。これは、出力の積分項を生成するために Igain を掛けた値です。

**pid.N.errorD** float ro (only if debug=1)

エラーの派生物。これは、出力の微分項を生成するために Dgain を掛けた値です。

**pid.N.commandD** float ro (only if debug=1)

コマンドの派生物。これは、出力の 1 次フィードフォワード項を生成するために FF1 を掛けた値です。

**pid.N.commandDD** float ro (only if debug=1)

コマンドの 2 次導関数。これは、出力の 2 次フィードフォワード項を生成するために FF2 を掛けた値です。

**pid.N.ultimate-gain** float ro (only if debug=1)

プロセスの特性から決定されます。アルティメットゲインは、リミットサイクル振幅に対するチューンエフォートの比率に 4.0 を掛けて Pi で割ったものです。pid.N.ultimate-period float ro (debug = 1 の場合のみ) プロセスの特性から決定されます。Ultimate-period は、リミットサイクルの期間です。

**BUGS**

一部の人は、デッドバンドは、エラーがデッドバンド内にある場合はゼロとして扱われ、デッドバンド外にある場合は変更されないように実装する必要があると主張します。これは、不感帯のサイズに等しい伝達関数のステップを引き起こすため、実行されませんでした。その振る舞いを好む人は、振る舞いを変更するパラメーターを追加するか、独自のバージョンの at\_pid を作成することを歓迎します。ただし、デフォルトの動作は変更しないでください。

**NAME**

axis – AXIS LinuxCNC Graphical User Interface

**SYNOPSIS**

**axis** *-ini* *INIFILE*

**DESCRIPTION**

axis は、LinuxCNC のグラフィカルユーザーインターフェイス（GUI）の 1 つです。通常、runscript によって実行されます。

**OPTIONS**

少し冗長になります。これにより、スクリプトは動作中に情報を出力します。

**INIFILE** ini ファイルは、LinuxCNC 構成の主要部分です。構成全体ではありません。それに付随する他のさまざまなファイルがあります（NML ファイル、HAL ファイル、TBL ファイル、VAR ファイル）。しかし、それは最も重要なのは、構成をまとめて保持するファイルだからです。調整できます多くのパラメータ自体がありますが、ロードして使用する他のファイルも linuxcnc に指示します。

**SEE ALSO**

**LinuxCNC(1)**

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr/share/doc/linuxcnc/ にあります。

**HISTORY****BUGS**

現時点では不明です。

**AUTHOR**

この man ページは、LinuxCNC Enhanced MachineController プロジェクトの一部として AlexJoni によって作成されました。

**REPORTING BUGS**

バグを alex\_joniAT ユーザーに報告する DOTsourceforge DOT net

**COPYRIGHT**

Copyright © 2006 Alex Joni.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についても または特定の目的への適合性。

**NAME**

axis-remote – AXIS Remote Interface

**SYNOPSIS**

**axis-remote** *OPTIONS* | *FILENAME*

**DESCRIPTION**

axis-remote は、実行中の AXISGUI でコマンドをトリガーする小さなスクリプトです。 axis-remote --- help を使用する 詳細については。

**OPTIONS**

**--ping, -n**

AXIS が実行されているかどうかを確認します。

**--reload, -r**

AXIS に現在ロードされているファイルをリロードさせます。

**--clear, -c**

AXIS にバックプロットをクリアさせます。

**--quit, -q**

AXIS を終了させます。

**--help, -h, -?**

axis-remote の有効なパラメータのリストを表示します。

**--mdi COMMAND, -m COMMAND**

MDI コマンド COMMAND を実行します。

**FILENAME**

G コードファイル FILENAME をロードします。

**SEE ALSO**

**axis(1)**

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr/share/doc/linuxcnc/ にあります。

**HISTORY****BUGS**

現時点では不明です。

**AUTHOR**

この man ページは、LinuxCNC プロジェクトの一部として AlexJoni によって作成されました。

**REPORTING BUGS**

バグを alex\_joniAT ユーザーに報告する DOTsourceforge DOT net

**COPYRIGHT**

Copyright © 2006 Alex Joni.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についてもまたは特定の目的への適合性。



**NAME**

debuglevel – sets the debug level for the userspace part of LinuxCNC

**SYNOPSIS**

**debuglevel** *-ini INIFILE*

**DESCRIPTION**

debuglevel は、LinuxCNC の一部の現在のデバッグレベルを選択するためのチェックボックス GUI を表示します。

**SEE ALSO**

halcmd(1) - debug subcommand **LinuxCNC(1)**

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr/share/doc/LinuxCNC/ にあります。

**HISTORY****BUGS**

現時点では不明です。

**AUTHOR**

この man ページは、LinuxCNC プロジェクトの一部として andypugh によって作成されました。

**REPORTING BUGS**

<https://github.com/LinuxCNC/linuxcnc/issues> でバグを報告してください

**COPYRIGHT**

Copyright © 2020 andypugh.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY

or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についても または特定の目的への適合性。

**NAME**

encoder – software counting of quadrature encoder signals

**SYNOPSIS**

**loadrt encoder [num\_chan=num | names=name1[,name2...]]**

**DESCRIPTION**

エンコーダは、直交エンコーダによって生成されたパルスをカウントすることによって位置を測定するために使用されます。ソフトウェアベースの実装として、ハードウェアよりもはるかに安価ですが、最大カウント率が制限されています。制限は、コンピューターの速度やその他の要因に応じて、10KHz から 50KHz の範囲です。より良いパフォーマンスが必要な場合は、ハードウェアエンコーダカウンターの方が適しています。一部のハードウェアベースのシステムは、MHz レートでカウントできます。

エンコーダは最大 8 チャンネルをサポートします。実際にロードされるチャンネルの数は、モジュールがロードされる時に num\_chan 引数によって設定されます。または、names = と一意の名前をコンマで区切って指定します。

num\_chan = および names = 指定子は相互に排他的です。num\_chan = も names = も指定されていない場合、または num\_chan = 0 が指定されている場合、デフォルト値は 3 です。

エンコーダには、カウンタと呼ばれる単相の単方向モードがあります。このモードでは、フェーズ B 入力は無視されます。カウントは、フェーズ A の立ち上がりエッジごとに増加します。このモードは、直角位相の耐ノイズ特性は失われますが、単一の入力ラインを持つ単方向スピンドルをカウントするのに役立つ場合があります。

**FUNCTIONS****encoder.update-counters** (no floating-point)

エンコーダ信号をサンプリングし、直交波形をデコードすることにより、実際のカウントを行います。できるだけ頻繁に呼び出す必要があります。できれば、必要な最大カウントレートの 2 倍の速度で呼び出す必要があります。すべてのチャンネルで一度に動作します。

**encoder.capture-position** (uses floating point)

update-counters から raw カウントをキャプチャし、スケーリングやその他の必要な変換を実行し、カウンタのロールオーバーなどを処理します。update-counters よりも頻繁に呼び出すことはできません（呼び出す必要があります）。すべてのチャンネルで一度に動作します。

**NAMING**

ピン、パラメーター、および関数の名前には、次の接頭辞が付いています。

**encoder.N.** for N=0,1,...,num-1 when using **num\_chan=num**

**nameN.** for nameN=name1,name2,... when using **names=name1,name2,...**

pid.N. 以下の説明にフォーマットを示します。

**PINS****encoder.N.counter-mode** bit i/o

カウンターモードを有効にします。true の場合、カウンタはフェーズ B の値を無視して、フェーズ A 入力の各立ち上がりエッジをカウントします。これは、単一チャンネル（非直交）

センサーの出力をカウントするのに役立ちます。 `false` (デフォルト) の場合、直交モードでカウントされます。

**encoder.N.counts** s32 out

エンコーダカウントでの位置。

**encoder.N.index-enable** bit i/o

`true` の場合、フェーズ Z の次の立ち上がりエッジでカウントと位置がゼロにリセットされます。同時に、インデックスイネーブルがゼロにリセットされ、立ち上がりエッジが発生したことを示します。

**encoder.N.min-speed-estimate** float in (default: 1.0)

速度が非ゼロとして推定され、位置補間が補間される最小速度を決定します。 `min-speed-estimate` の単位は、速度の単位と同じです。このパラメータの設定が低すぎると、エンコーダパルスの到着が停止した後、速度が 0 になるまでに長い時間がかかります。

**encoder.N.phase-A** bit in

エンコーダチャンネル N の A 相入力。

**encoder.N.phase-B** bit in

B 相入力。

**encoder.N.phase-Z** bit in

Z 相入力

**encoder.N.position** float out

スケールされた単位での位置 (`position-scale` を参照)

**encoder.N.position-interpolated** float out

エンコーダカウント間で補間された、スケールされた単位での位置。速度がほぼ一定で、最小速度の推定値を超えている場合にのみ有効です。位置制御には使用しないでください。

**encoder.N.position-scale** float i/o

長さの単位あたりのカウントでのスケール係数。たとえば、`position-scale` が 500 の場合、エンコーダの 1000 カウントは 2.0 単位の位置として報告されます。

**encoder.N.rawcounts** s32 out

`update-counters` によって決定された raw カウント。この値は、カウントや位置よりも頻繁に更新されます。また、リセットやインデックスパルスの影響を受けません。

**encoder.N.reset** bit in

`true` の場合、カウントと位置はすぐにゼロにリセットされます。

**encoder.N.velocity** float out

1 秒あたりのスケールされた単位での速度。エンコーダは、位置出力を単純に微分する場合と比較して、量子化ノイズを大幅に低減するアルゴリズムを使用します。真の速度の大きさが `min-speed-estimate` を下回る場合、速度出力は 0 になります。

**encoder.N.velocity-rpm** float out

1 分あたりのスケールされた単位での速度。便宜上、エンコーダーの速度を 60 倍にスケールしただけです。

**encoder.N.x4-mode** bit i/o

`times-4` モードを有効にします。 `true` (デフォルト) の場合、カウンタは直交波形の各エッジをカウントします (フルサイクルごとに 4 カウント)。 `false` の場合、フルサイクルご

とに1回だけカウントされます。カウンターモードでは、このパラメーターは無視されます。

**encoder.N.latch-input** bit in

**encoder.N.latch-falling** bit in (default: **TRUE**)

**encoder.N.latch-rising** bit in (default: **TRUE**)

**encoder.N.counts-latched** s32 out

**encoder.N.position-latched** float out

ラッチの立ち上がりとラッチの立ち下がりで見られるように、ラッチ入力の立ち上がりエッジおよび/または立ち下がりエッジでカウントラッチと位置ラッチを更新します。

**encoder.N.counter-mode** bit rw

カウンターモードを有効にします。trueの場合、カウンタはフェーズBの値を無視して、フェーズA入力の各立ち上がりエッジをカウントします。これは、単一チャンネル（非直交）センサーの出力をカウントするのに役立ちます。false（デフォルト）の場合、直交モードでカウントされます。ncoder.N.capture-position。tmax s32rw この関数の実行にかかったCPUサイクルの最大数。

## PARAMETERS

エンコーダコンポーネントにはHALパラメータがありません。

## NAME

gladevcp – used by sample configs to deonstrate Glade Virtual\_demo

## SYNOPSIS

**gladevcp\_demo** Control Panels

## DESCRIPTION

gladevcp\_demo はサンプルの GladeVCP ハンドラーです

## SEE ALSO

<https://linuxcnc.org/docs/html/gui/gladevcp.html> **LinuxCNC(1)**

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr / share / doc / linuxcnc /にあります。

## HISTORY

## BUGS

現時点では不明です。

## AUTHOR

この man ページは、LinuxCNC Enhanced MachineController プロジェクトの一部として AlexJoni によって作成されました。

## REPORTING BUGS

バグを alex\_joniAT ユーザーに報告する DOTsourceforge DOT net

## COPYRIGHT

Copyright © 2006 Alex Joni.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についても または特定の目的への適合性。

**NAME**

gmoccapy – TOUCHY LinuxCNC Graphical User Interface

**SYNOPSIS**

**gmoccapy** *-ini INIFILE*

**DESCRIPTION**

gmoccapy は、LinuxCNC のグラフィカルユーザーインターフェイス（GUI）の 1 つです。通常、runscript によって実行されます。

**OPTIONS****INIFILE**

ini ファイルは、LinuxCNC 構成の主要部分です。構成全体ではありません。それに付随する他のさまざまなファイルがあります（NML ファイル、HAL ファイル、TBL ファイル、VAR ファイル）。しかし、それは最も重要なのは、構成をまとめて保持するファイルだからです。調整できます多くのパラメータ自体がありますが、ロードして使用する他のファイルも linuxcnc に指示します。

**SEE ALSO****LinuxCNC(1)**

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr / share / doc / linuxcnc / にあります。

**HISTORY****BUGS**

現時点では不明です。

**AUTHOR**

この man ページは、LinuxCNC プロジェクトの一部として andypugh によって作成されました。

**REPORTING BUGS**

バグを alex\_joniAT ユーザーに報告する DOTsourceforge DOT net

**COPYRIGHT**

Copyright © 2006 Alex Joni.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についてもまたは特定の目的への適合性。

**NAME**

gremlin\_view – G-code graphical preview

**SYNOPSIS**

**gremlin\_view**

**DESCRIPTION**

gremlin\_view は、gremlinG コードのグラフィカルプレビュー用の Python ラッパーです。

linuxcnc が実行されている場合にスタンドアロン機能を簡単に埋め込むためのボタンを備えたグレムリンの PGremlinView デフォルトのボタン配置用にデフォルトの UI ファイル (gremlin\_view.ui) が提供されていますが、ユーザーが提供することもできます glade\_file 引数を指定することにより、独自のものにします。次のオブジェクトは必須です。

'gremlin\_view\_window' toplevel window

'gremlin\_view\_hal\_gremlin' hal\_gremlin

'gremlin\_view\_box' hal\_gremlin を含む HBox または VBox のオプションのラジオボタングループ名:

'select\_p\_view'

'select\_x\_view'

'select\_y\_view'

'select\_z\_view'

'select\_z2\_view' Optional Checkbuttons names:

'enable\_dro'

'show\_machine\_speed'

'show\_distance\_to\_go'

'show\_limits'

'show\_extents'

'show\_tool'

'show\_metric' 次のボタンアクションに対してコールバックが提供されます

on\_clear\_live\_plotter\_clicked

on\_enable\_dro\_clicked

on\_zoomin\_pressed

on\_zoomout\_pressed

on\_pan\_x\_minus\_pressed

on\_pan\_x\_plus\_pressed

on\_pan\_y\_minus\_pressed

on\_pan\_y\_plus\_pressed

on\_show\_tool\_clicked

on\_show\_metric\_clicked

on\_show\_extents\_clicked

on\_select\_p\_view\_clicked

on\_select\_x\_view\_clicked

on\_select\_y\_view\_clicked  
on\_select\_z\_view\_clicked  
on\_select\_z2\_view\_clicked  
on\_show\_distance\_to\_go\_clicked  
on\_show\_machine\_speed\_clicked  
on\_show\_limits\_clicked

## SEE ALSO

[http://wiki.linuxcnc.org/cgi-bin/wiki.pl?Gremlin LinuxCNC\(1\)](http://wiki.linuxcnc.org/cgi-bin/wiki.pl?Gremlin%20LinuxCNC(1))

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr/share/doc/linuxcnc/ にあります。

## HISTORY

## BUGS

現時点では不明です。

## AUTHOR

この man ページは、LinuxCNC プロジェクトの一部として andypugh によって作成されました。

## REPORTING BUGS

バグを alex\_joniAT ユーザーに報告する DOTsourceforge DOT net

## COPYRIGHT

Copyright © 2006 Alex Joni.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についてもまたは特定の目的への適合性。



## NAME

**gs2\_vfd** – HAL userspace component for Automation Direct GS2 VFD's

## SYNOPSIS

**gs2\_vfd** [OPTIONS]

## DESCRIPTION

このマニュアルページでは、**gs2\_vfd** コンポーネントについて説明しています。このコンポーネントは、modbus 接続を介して GS2 の読み取りと書き込みを行います。

**gs2\_vfd** は LinuxCNC で使用するためのものです

## OPTIONS

**-b, --bits <n>**

(デフォルトは 8) データビット数を<n>に設定します。n は 5 から 8 まででなければなりません。

**-d, --device <path>**

(デフォルト / dev / ttyS0) 使用するシリアルデバイスノードの名前を設定します。

**-v, --verbose**

詳細モードをオンにします。

**-g, --debug**

デバッグメッセージをオンにします。シリアルエラーがある場合、これは煩わしいものになる可能性があることに注意してください。デバッグモードでは、すべての modbus メッセージが端末に 16 進数で出力されます。

**-n, --name <string>**

(デフォルト **gs2\_vfd**) HAL モジュールの名前を設定します。HAL comp 名は <string> に設定され、すべてのピン名とパラメーター名は <string> で始まります。

**-p, --parity [even,odd,none]**

(デフォルトは奇数) シリアルパリティを偶数、奇数、またはなしに設定します。

**-r, --rate <n>**

(デフォルトは 38400) ボーレートを<n>に設定します。レートが次のいずれでもない場合はエラーです：  
110、300、600、1200、2400、4800、9600、19200、38400、57600、115200

**-s, --stopbits [1,2]**

(デフォルト 1) シリアルストップビットを 1 または 2 に設定します

**-t, --target <n>**

(デフォルト 1) MODBUS ターゲット (スレーブ) 番号を設定します。これは、で設定したデバイス番号と一致する必要があります GS2。

**-A, --accel-seconds <n>**

(デフォルトは 10.0) スピンドルを 0 から最大 RPM に加速する秒数

**-D, --decel-seconds <n>**

(デフォルトは 0.0) スピンドルを最大 RPM から 0 に減速する秒数。0.0 に設定すると、スピンドルは制御された減速なしで惰走停止することができます。

**-R, --braking-resistor**

この引数は、GS2 VFD に制動抵抗器が取り付けられている場合に使用する必要があります（GS2 マニュアルの付録 A を参照）。減速過電圧失速防止を無効にし（GS2 modbus パラメーター 6.05 を参照）、モーターが高電圧を再生している状況でも VFD がブレーキをかけ続けることができます。再生電圧は安全に制動抵抗器にダンプされま

**PINS****<name>.DC-bus-volts (float, out)**

from the VFD

**<name>.at-speed (bit, out)**

ドライブがコマンド速度のとき

**<name>.err-reset (bit, in)**

reset errors sent to VFD

**<name>.firmware-revision (s32, out)**

from the VFD

**<name>.frequency-command (float, out)**

from the VFD

**<name>.frequency-out (float, out)**

from the VFD

**<name>.is-stopped (bit, out)**

VFD が 0Hz 出力を報告するとき

**<name>.load-percentage (float, out)**

from the VFD

**<name>.motor-RPM (float, out)**

from the VFD

**<name>.output-current (float, out)**

from the VFD

**<name>.output-voltage (float, out)**

from the VFD

**<name>.power-factor (float, out)**

from the VFD

**<name>.scale-frequency (float, out)**

from the VFD

**<name>.speed-command (float, in)**

RPM で VFD に送信される速度 VFD で設定されているモーター最大 RPM よりも速い速度を送信するとエラーになります

**<name>.spindle-fwd (bit, in)**

VFD に送信される FWD の場合は 1、REV の場合は 0

**<name>.spindle-on (bit, in)**

VFD に送信されるオンの場合は 1、オフの場合は 0、実行中のみオン

**<name>.spindle-rev (bit, in)**

オンの場合は 1、オフの場合は 0、実行中のみオン

**<name>.status-1 (s32, out)**

Drive Status of the VFD (see the GS2 manual)

**<name>.status-2 (s32, out)**

VFD のドライブステータス (GS2 マニュアルを参照) 値は、オンになっているすべてのビットの合計であることに注意してください。したがって、ドライブが実行モードにあることを意味する 163 は、3 (実行) + 32 (シリアルによって設定された周波数) + 128 (シリアルによって設定された操作) の合計です。

## PARAMETERS

**<name>.error-count (s32, RW)**

**<name>.loop-time (float, RW)**

modbus がポーリングされる頻度 (デフォルトは 0.1)

**<name>.nameplate-HZ (float, RW)**

モーターの銘板 Hz (デフォルトは 60)

**<name>.nameplate-RPM (float, RW)**

モーターの銘板 RPM (デフォルトは 1730)

**<name>.retval (s32, RW)**

HAL のエラーの戻り値

**<name>.tolerance (float, RW)**

速度許容差 (デフォルトは 0.01)

**<name>.ack-delay (s32, RW)**

at-speed をチェックする前の読み取り/書き込みサイクル数 (デフォルトは 2)

## SEE ALSO

*GS2 Driver* in the LinuxCNC documentation for a full description of the **GS2** syntax

*GS2 Examples* in the LinuxCNC documentation for examples using the **GS2** component

## HISTORY

## BUGS

現時点では不明です。

## AUTHOR

John Thornton

## LICENSE

GPL

**NAME**

gscreen – TOUCHY LinuxCNC Graphical User Interface

**DESCRIPTION**

gscreen は、LinuxCNC のグラフィカルユーザーインターフェイス（GUI）の 1 つです。通常、runscript によって実行されます。

**OPTIONS****INIFILE**

ini ファイルは、LinuxCNC 構成の主要部分です。構成全体ではありません。それに付随する他のさまざまなファイルがあります（NML ファイル、HAL ファイル、TBL ファイル、VAR ファイル）。しかし、それは最も重要なのは、構成をまとめて保持するファイルだからです。調整できます多くのパラメータ自体がありますが、ロードして使用する他のファイルも linuxcnc に指示します。

**SEE ALSO****LinuxCNC(1)**

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr / share / doc / LinuxCNC / にあります。

**HISTORY****BUGS**

現時点では不明です。

**AUTHOR**

この man ページは、LinuxCNC プロジェクトの一部として andypugh によって作成されました。

**REPORTING BUGS**

<https://github.com/LinuxCNC/linuxcnc/issues> でバグを報告してください

**COPYRIGHT**

Copyright © 2020 andypugh.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についても または特定の目的への適合性。

**NAME**

halui – observe HAL pins and command LinuxCNC through NML

**SYNOPSIS**

**halui** [**-ini** <path-to-ini>]

**DESCRIPTION**

halui は、ハードウェアのノブとスイッチを使用してユーザーインターフェイスを構築するために使われます。多数のピンをエクスポートし、これらに変更されたときにそれに応じて動作します。

**OPTIONS****-ini name**

名前を構成ファイルとして使用します。注：halui は ini で指定された nml ファイルを見つける必要があります。通常、そのファイルは ini と同じフォルダーにあるため、そのフォルダーから halui を実行するのが理にかなっています。

**USAGE**

実行すると、halui は多数のピンをエクスポートします。ユーザーはそれらを自分の物理的なノブとスイッチと LED に接続でき、変化に気づいたときに halui が適切なイベントをトリガーします。

halui は信号がデバウンスされることを想定しているため、必要に応じて（ノブの接触が悪い）、最初に物理ボタンを HAL デバウンスフィルターに接続します。

**PINS****abort**

**halui.abort** bit in

ほとんどのエラーをクリアするためのピン

**tool**

**halui.tool.length-offset.a** float out

A 軸に現在適用されている工具長オフセット

**halui.tool.length-offset.b** float out

B 軸に現在適用されている工具長オフセット

**halui.tool.length-offset.c** float out

C 軸に現在適用されている工具長オフセット

**halui.tool.length-offset.u** float out

U 軸に現在適用されている工具長オフセット

**halui.tool.length-offset.v** float out

V 軸に現在適用されている工具長オフセット

**halui.tool.length-offset.w** float out

W 軸に現在適用されている工具長オフセット

**halui.tool.length-offset.x** float out

X 軸に現在適用されている工具長オフセット

**halui.tool.length-offset.y** float out

Y 軸に現在適用されている工具長オフセット

**halui.tool.length-offset.z** float out

Z軸に現在適用されている工具長オフセット

**halui.tool.diameter** float out

現在の工具径、または工具がロードされていない場合は0

**halui.tool.number** u32 out

現在選択されているツール

## spindle

**halui.spindle.N.brake-is-on** bit out

ブレーキがオンかどうかを示すステータスピン

**halui.spindle.N.brake-off** bit in

スピンドルブレーキを無効にするためのピン

**halui.spindle.N.brake-on** bit in

スピンドルブレーキを作動させるためのピン

**halui.spindle.N.decrease** bit in

このピンの立ち上がりエッジにより、現在のスピンドル速度が100低下します。

**halui.spindle.N.forward** bit in

このピンの立ち上がりエッジにより、スピンドルが前進します

**halui.spindle.N.increase** bit in

このピンの立ち上がりエッジにより、現在のスピンドル速度が100増加します。

**halui.spindle.N.is-on** bit out

スピンドルがオンかどうかを示すステータスピン

**halui.spindle.N.reverse** bit in

このピンの立ち上がりエッジにより、スピンドルが逆になります

**halui.spindle.N.runs-backward** bit out

スピンドルが逆回転しているかどうかを示すステータスピン

**halui.spindle.N.runs-forward** bit out

スピンドルが前進しているかどうかを示すステータスピン

**halui.spindle.N.start** bit in

このピンの立ち上がりエッジがスピンドルを開始します

**halui.spindle.N.stop** bit in

このピンの立ち上がりエッジでスピンドルが停止します

## spindle override

**halui.spindle.N.override.count-enable** bit in (default: **TRUE**)

TRUEの場合、カウントが変更されたときにスピンドルのオーバーライドを変更します。

**halui.spindle.N.override.counts** s32 in

カウント X スケール = スピンドルオーバーライドパーセンテージ

**halui.spindle.N.override.decrease** bit in

SOを下げるためのピン (- = スケール)

**halui.spindle.N.override.direct-value** bit in

直接スピンドルオーバーライド値入力を有効にするピン

**halui.spindle.N.override.increase** bit in

SOを増やすためのピン (+ = スケール)

**halui.spindle.N.override.scale** float in

SOのカウントのスケールを設定するためのピン

**halui.spindle.N.override.value** float out

現在の OF 値

## program

**halui.program.block-delete.is-on** bit out

ブロック削除がオンになっていることを示すステータスピン

**halui.program.block-delete.off** bit in

ブロック削除を要求するためのピンがオフになっている

**halui.program.block-delete.on** bit in

ブロック削除を要求するためのピンがオンになっている

**halui.program.is-idle** bit out

プログラムが実行されていないことを示すステータスピン

**halui.program.is-paused** bit out

プログラムが一時停止していることを示すステータスピン

**halui.program.is-running** bit out

プログラムが実行中であることを示すステータスピン

**halui.program.optional-stop.is-on** bit out

オプションストップがオンになっていることを示すステータスピン

**halui.program.optional-stop.off** bit in

オプションストップがオフであることを要求するピン

**halui.program.optional-stop.on** bit in

オプションストップがオンであることを要求するピン

**halui.program.pause** bit in

プログラムを一時停止するためのピン

**halui.program.resume** bit in

プログラムを再開するためのピン

**halui.program.run** bit in

プログラムを実行するためのピン

**halui.program.step** bit in

プログラムにステップインするためのピン

**halui.program.stop** bit in

プログラムを停止するためのピン (注: このピンは halui.abort と同じことを行います)

## mode

**halui.mode.auto** bit in

自動モードを要求するためのピン

**halui.mode.is-auto** bit out

自動モードのピンがオンになっている

**halui.mode.is-joint** bit out

ジョイントジョグモードによるジョイントを示すピンがオンになっている

**halui.mode.is-manual** bit out

手動モードのピンがオンになっている

**halui.mode.is-mdi** bit out

mdi モードのピンがオンになっている

**halui.mode.is-teleop** bit out

協調ジョグモードを示すピンがオンになっている

**halui.mode.joint** bit in

ジョイントジョグモードでジョイントを要求するためのピン

**halui.mode.manual** bit in

手動モードを要求するためのピン

**halui.mode.mdi** bit in

mdi モードを要求するためのピン

**halui.mode.teleop** bit in

協調ジョグモードを要求するためのピン

**mdi** (optional)

**halui.mdi-command-XX** bit in

**halui** looks for ini variables named [HALUI]MDI\_COMMAND, and exports a pin for each command it finds. When the pin is driven TRUE, **halui** runs the specified MDI command. XX is a two digit number starting at 00. If no [HALUI]MDI\_COMMAND variables are set in the ini file, no halui.mdi-command-XX pins will be exported by halui.

**mist**

**halui.mist.is-on** bit out

ミスト用ピンがオンになっています

**halui.mist.off** bit in

ミストを止めるためのピン

**halui.mist.on** bit in

ミストを開始するためのピン

**max-velocity**

**halui.max-velocity.count-enable** bit in (default: **TRUE**)

True の場合、halui.max-velocity.counts が変更されたときの最大速度を変更します。

**halui.max-velocity.counts** s32 in

.count-enable が True の場合、halui はこのピンの変更に応じて最大速度を変更します。通常、halui は操作パネルまたはジョグペンダントの MPG エンコーダーに接続されます。

.count-enable が False の場合、halui はこのピンを無視します。

**halui.max-velocity.direct-value** bit in

このピンが True の場合、halui は最大速度を直接 ( $\text{.counts} * \text{.scale}$ ) に命令します。このピンが False の場合、halui は相対的な方法で最大速度を指令します。最大速度を ( $\text{.counts} * \text{.scale}$  の変更) に等しい量だけ変更します。

**halui.max-velocity.increase** bit in

このピンのポジティブエッジ (False から True への遷移) は、.scale ピンの値だけ最大速度を増加させます。(halui は、.count-enable ピンとは関係なく、常にこのピンに応答する



ことに注意してください。)

**halui.max-velocity.decrease** bit in

このピンのポジティブエッジ (False から True への遷移) は、.scale ピンの値だけ最大速度を低下させます。(halui は、.count-enable ピンとは関係なく、常にこのピンに応答することに注意してください。)

**halui.max-velocity.scale** float in

このピンは、最大速度の変化のスケールを制御します。.counts の各単位の変更、および.increase と.decrease の各正のエッジは、最大速度を.scale だけ変更します。.scale ピンの単位は、1 秒あたりのマシン単位です。

**halui.max-velocity.value** float out

最大速度の現在の値 (マシン単位/秒)

**machine**

**halui.machine.units-per-mm** float out

inifile 設定に応じた 1mm あたりのマシンユニットのピン (インチ : 1 / 25.4、mm : 1) :  
[TRAJ]LINEAR\_UNITS

**halui.machine.is-on** bit out

機械のピンはオン/オフです

**halui.machine.off** bit in

機械をオフに設定するためのピン

**halui.machine.on** bit in

設定機用ピンオン

**lube**

**halui.lube.is-on** bit out

潤滑油のピンがオンになっています

**halui.lube.off** bit in

潤滑油を止めるためのピン

**halui.lube.on** bit in

潤滑を開始するためのピン

**joint (N = joint number (0 ... num\_joints-1))**

**halui.joint.N.select** bit in

ジョイント N を選択するためのピン

**halui.joint.N.is-selected** bit out

ジョイント N が選択されているステータスピン

**halui.joint.N.has-fault** bit out

ジョイント N に障害があることを示すステータスピン

**halui.joint.N.home** bit in

ホームジョイント用ピン N

**halui.joint.N.is-homed** bit out

ジョイント N がホームになったことを示すステータスピン

**halui.joint.N.on-hard-max-limit** bit out

ジョイント N が正のハードウェア制限にあることを示すステータスピン

**halui.joint.N.on-hard-min-limit** bit out

ジョイント N がハードウェアの負の制限にあることを示すステータスピン

**halui.joint.N.on-soft-max-limit** bit out

ジョイント N が正のソフトウェア制限にあることを示すステータスピン

**halui.joint.N.on-soft-min-limit** bit out

ジョイント N がソフトウェアの負の制限にあることを示すステータスピン

**halui.joint.N.override-limits** bit out

ジョイント N の制限が一時的に上書きされることを示すステータスピン

**halui.joint.N.unhome** bit in

ホーミング解除ジョイント用ピン N

**halui.joint.selected** u32 out

選択したジョイント番号 (0 ... num\_joints-1)

**halui.joint.selected.has-fault** bit out

ステータスピン選択ジョイントに障害があります

**halui.joint.selected.home** bit in

選択したジョイントをホーミングするためのピン

**halui.joint.selected.is-homed** bit out

選択したジョイントがホームになっていることを示すステータスピン

**halui.joint.selected.on-hard-max-limit** bit out

選択したジョイントが正のハードウェア制限にあることを示すステータスピン

**halui.joint.selected.on-hard-min-limit** bit out

選択したジョイントがハードウェアの負の制限にあることを示すステータスピン

**halui.joint.selected.on-soft-max-limit** bit out

選択したジョイントが正のソフトウェア制限にあることを示すステータスピン

**halui.joint.selected.on-soft-min-limit** bit out

選択したジョイントがソフトウェアの負の制限にあることを示すステータスピン

**halui.joint.selected.override-limits** bit out

選択したジョイントの制限が一時的に上書きされたことを示すステータスピン

**halui.joint.selected.unhome** bit in

選択したジョイントのホーミングを解除するためのピン

**joint jogging (N = joint number (0 ... num\_joints-1))**

**halui.joint.jog-deadband** float in pin for setting jog analog deadband (jog analog inputs smaller/slower)

これより (絶対値で) 無視されます)

**halui.joint.jog-speed** float in

プラス/マイナスジョギングのジョグ速度を設定するためのピン。

**halui.joint.N.analog** float in

フロート値 (ジョイスティックなど) を使用してジョイント N をジョグするためのピン。通常 0.0 から ±1.0 の間に設定される値は、ジョグ速度の乗数として使用されます。

**halui.joint.N.increment** float in

増分プラス/マイナスを使用する場合のジョイント N のジョグ増分を設定するためのピン

**halui.joint.N.increment-minus** bit in

立ち上がりエッジは、ジョイント N を増分量だけ負の方向にジョグします。

**halui.joint.N.increment-plus** bit in

立ち上がりエッジは、ジョイント N を増分量だけ正の方向にジョグします。

**halui.joint.N.minus** bit in

halui.joint.jog-speed 速度で負の方向にジョイント N をジョギングするためのピン

**halui.joint.N.plus** bit in

halui.joint.jog-speed 速度で正の方向にジョイント N をジョギングするためのピン

**halui.joint.selected.increment** float in

増分プラス/マイナスを使用する場合、選択したジョイントのジョグ増分を設定するためのピン

**halui.joint.selected.increment-minus** bit in

立ち上がりエッジにより、選択したジョイントが増分量だけ負の方向にジョグします。

**halui.joint.selected.increment-plus** bit in

立ち上がりエッジにより、選択したジョイントが増分量だけ正の方向にジョグします。

**halui.joint.selected.minus** bit in

選択したジョイントを halui.joint.jog-speed 速度で負の方向にジョギングするためのピン

**halui.joint.selected.plus**

選択したジョイントビットを halui.joint.jog-speed 速度で正の方向にジョグするためのピン

**axis jogging (L = axis letter (xyzabcuvw))****halui.axis.jog-deadband** float in

ジョグアナログ不感帯を設定するためのピン（これよりも小さい/遅いジョグアナログ入力（絶対値）は無視されます）

**halui.axis.jog-speed** float in

プラス/マイナスジョギングのジョグ速度を設定するためのピン。

**halui.axis.L.analog** float in

フロート値（ジョイスティックなど）を使用して軸 L をジョグするためのピン。通常 0.0 から ±1.0 の間に設定される値は、ジョグ速度の乗数として使用されます。

**halui.axis.L.increment** float in

増分プラス/マイナスを使用する場合、軸 L のジョグ増分を設定するためのピン

**halui.axis.L.increment-minus** bit in

立ち上がりエッジにより、軸 L が増分量だけ負の方向にジョグします。

**halui.axis.L.increment-plus** bit in

立ち上がりエッジにより、軸 L が増分量だけ正の方向にジョグします。

**halui.axis.L.minus** bit in

halui.axis.jog-speed 速度で負の方向に軸 L をジョギングするためのピン

**halui.axis.L.plus** bit in

halui.axis.jog-speed 速度で正方向に軸 L をジョギングするためのピン

**halui.axis.selected** u32 out

選択した軸（インデックス：0：x 1：y 2：z 3：a 4：b 5：cr 6：u 7：v 8：w）

**halui.axis.selected.increment** float in

増分プラス/マイナスを使用する場合、選択した軸のジョグ増分を設定するためのピン

**halui.axis.selected.increment-minus** bit in

立ち上がりエッジにより、選択した軸が増分量だけ負の方向にジョグします。

**halui.axis.selected.increment-plus** bit in

立ち上がりエッジにより、選択した軸が増分量だけ正の方向にジョグします。

**halui.axis.selected.minus** bit in

選択した軸を halui.axis.jog-speed 速度で負の方向にジョグするためのピン

**halui.axis.selected.plus**

選択した軸ビットを halui.axis.jog-speed 速度で正の方向にジョグするためのピン

## flood

**halui.flood.is-on** bit out

洪水用のピンがオンになっています

**halui.flood.off** bit in

洪水を止めるためのピン

**halui.flood.on** bit in

洪水を開始するためのピン

## feed override

**halui.feed-override.count-enable** bit in (default: **TRUE**)

TRUE の場合、カウントが変更されたときにフィードのオーバーライドを変更します。

**halui.feed-override.counts** s32 in

カウント X スケール=フィードオーバーライドパーセンテージ

**halui.feed-override.decrease** bit in

FO を下げるためのピン (-=スケール)

**halui.feed-override.direct-value** bit in

直接値フィードオーバーライド入力を有効にするピン

**halui.feed-override.increase** bit in

FO を上げるためのピン (+ =スケール)

**halui.feed-override.scale** float in

FO 変更時の目盛りを設定するためのピン

**halui.feed-override.value** float out

現在のフィードオーバーライド値

## rapid override

**halui.rapid-override.count-enable** bit in (default: **TRUE**)

TRUE の場合、カウントが変更されたときに RapidOverride を変更します。

**halui.rapid-override.counts** s32 in

カウント X スケール=高速オーバーライドパーセンテージ

**halui.rapid-override.decrease** bit in

ラピッドオーバーライドを減らすためのピン (-=スケール)

**halui.rapid-override.direct-value** bit in

直接値のラピッドオーバーライド入力を有効にするピン

**halui.rapid-override.increase** bit in

ラピッドオーバーライドを増やすためのピン (+ =スケール)

**halui.rapid-override.scale** float in

ラピッドオーバーライドの変更時にスケールを設定するためのピン

**halui.rapid-override.value** float out

現在のラピッドオーバーライド値

## **estop**

**halui.estop.activate** bit in

Estop を設定するためのピン (LinuxCNC 内部) オン

**halui.estop.is-activated** bit out

Estop 状態を表示するためのピン (LinuxCNC 内部) オン/オフ

**halui.estop.reset** bit in

Estop をリセットするためのピン (LinuxCNC 内部) オフ

## **home**

**halui.home-all** bit in

ホームオールを要求するためのピン (有効なホームイングシーケンスが指定されている場合にのみ使用可能)

## **SEE ALSO**

## **HISTORY**

## **BUGS**

現時点では不明です。

## **AUTHOR**

LinuxCNC プロジェクトの一部として AlexJoni によって書かれました。ジョンソートンによって更新されました

## **REPORTING BUGS**

バグを alex\_joniATユーザーに報告する DOTsourceforge DOT net

## **COPYRIGHT**

Copyright © 2006 Alex Joni.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性や特定の目的への適合性についてもそうではありません。

**NAME**

iocontrol – accepts NML I/O commands, interacts with HAL in userspace

**SYNOPSIS**

**loadusr io [-ini *inifile*]**

**DESCRIPTION**

これらのピンは、通常\$ LINUXCNC\_HOME / bin / ioにあるユーザースペース IO コントローラーによって作成されます。

信号はユーザースペースでオンとオフが切り替えられます。厳密なタイミング要件がある場合、または単により多くのI/Oが必要な場合は、代わりに motion (9) によって提供されるリアルタイム同期I/Oの使用を検討してください。

絶対パスが指定されていない限り、inifile は halcmd が実行されたディレクトリで検索されます。

**PINS****iocontrol.0.coolant-flood**

(Bit, Out) TRUE フラッドクーラントが要求されたとき

**iocontrol.0.coolant-mist**

(Bit, Out) TRUE ミストクーラントが要求されたとき

**iocontrol.0.emc-enable-in**

(Bit, In) 外部停止条件が存在する場合は、FALSE で駆動する必要があります。

**iocontrol.0.lube**

(Bit, Out) TRUE 潤滑油が要求されたとき。このピンは、コントローラーが E-stop から出たとき、および「LubeOn」コマンドがコントローラーに送信されたときに True に駆動されます。コントローラーが E-stop に入るとき、および「Lube Off」コマンドがコントローラーに送信されると、False に駆動されます。

**iocontrol.0.lube\_level**

(Bit, In) 潤滑タンクが空の場合は FALSE で駆動する必要があります。

**iocontrol.0.tool-change**

(Bit, Out) TRUE 工具交換が要求されたとき

**iocontrol.0.tool-changed**

(Bit, In) 工具交換が完了したら、TRUE で駆動する必要があります。

**iocontrol.0.tool-number**

(s32, Out) 現在の工具番号

**iocontrol.0.tool-prep-number**

(s32, Out) RS274NGCT ワードからの次のツールの番号

**iocontrol.0.tool-prep-pocket**

(s32, Out) これは、最新の T ワードによって要求されたツールのポケット番号（ツールストレージメカニズム内の場所）です。

**iocontrol.0.tool-prepare**

(Bit, Out) TRUE Tn ツールの準備が要求されたとき

**iocontrol.0.tool-prepared**

(Bit, In) ツールの準備が完了したら、TRUE を駆動する必要があります。

**iocontrol.0.user-enable-out**

(Bit, Out) FALSE 内部停止条件が存在する場合

**iocontrol.0.user-request-enable**

(Bit, Out) TRUE ユーザーが estop のクリアを要求したとき

**PARAMETERS****iocontrol.0.tool-prep-index**

(s32, RO) 最新の T ワードによって要求された準備済みツールの IO の内部配列インデックス。ツールが準備されていない場合は 0。ランダムツールチェンジャーマシンでは、これはツールのポケット番号です（つまり、tool-prep-pocket ピンと同じです）。非ランダムツールチェンジャーマシンでは、これはツールテーブルの内部表現でのツールの位置に対応する小さな整数です。このパラメータは、工具交換が成功すると 0 に戻ります (M6)。

**SEE ALSO**

**motion(9)**

## NAME

linuxcnc – LinuxCNC (The Enhanced Machine Controller)

## SYNOPSIS

**linuxcnc** [-v] [-d] [INIFILE]

## DESCRIPTION

linuxcnc は、LinuxCNC (Enhanced Machine Controller) を起動するために使用されます。リアルタイムシステムを起動し、次に、いくつかの LinuxCNC コンポーネント (IO、モーション、GUI、HAL など) を初期化します。最も重要なパラメータは INIFILE で、実行する構成名を指定します。INIFILE が指定されていない場合、linuxcnc スクリプトは、1つを選択できるグラフィカルウィザードを提供します。

## OPTIONS

- v**           少し冗長になります。これにより、スクリプトは動作中に情報を出力します。
  
- d**           たくさんのデバッグ情報を印刷します。実行されたすべてのコマンドが画面にエコーされます。このモードは何かが正常に機能していない場合に役立ちます。
  
- I**           プロンプトを表示せずに、最後に使用した INI ファイルを使用します。これは多くの場合、ショートカットコマンドに適していますまたはスタートアップアイテム。

## INIFILE

ini ファイルは、LinuxCNC 構成の主要部分です。構成全体ではありません。それに付随する他のさまざまなファイルがあります (NML ファイル、HAL ファイル、TBL ファイル、VAR ファイル)。しかし、それは最も重要なのは、構成をまとめて保持するファイルだからです。調整できます多くのパラメータ自体がありますが、ロードして使用する他のファイルも linuxcnc に指示します。

使用する構成を指定するには、いくつかの方法があります。

ini への絶対パスを指定します。例：

```
linuxcnc /usr/local/linuxcnc/configs/sim/sim.ini
```

現在のディレクトリからの相対パスを指定します。例：

```
linuxcnc configs/sim/sim.ini
```

それ以外の場合、INIFILE が指定されていない場合、動作は `--enable-run-in-place` を使用して linuxcnc を構成しました。もしそうなら、linuxcnc 設定チューザーはソースツリーの `configs` ディレクトリのみを検索します。そうでない場合 (またはパッケージバージョンを使用している場合) linuxcnc の)、それはいくつかのディレクトリを検索するかもしれません。構成チューザーは現在、検索するように設定されています

path :

```
~/linuxcnc/configs:/home/buildslave/emc2-buildbot/buster-rtpreempt-  
amd64/docs/build/configs
```



## EXAMPLES

### **Linuxcnc**

**linuxcnc** *configs/sim/sim.ini*

**linuxcnc** */etc/linuxcnc/sample-configs/stepper/stepper\_mm.ini*

## SEE ALSO

### **halcmd(1)**

LinuxCNC および HAL の詳細については、LinuxCNC および HAL のユーザーマニュアルを参照してください。 /usr/share/doc/linuxcnc/ にあります。

## HISTORY

### **BUGS**

現時点では不明です。

### **AUTHOR**

この man ページは、LinuxCNC Enhanced MachineController プロジェクトの一部として AlexJoni によって作成されました。

### **REPORTING BUGS**

バグを alex\_joniAT ユーザーに報告する DOTsourceforge DOT net

### **COPYRIGHT**

Copyright © 2006 Alex Joni.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

これは自由ソフトウェアです。コピー条件については、ソースを参照してください。保証はありません。商品性についてもまたは特定の目的への適合性。

## NAME

**milltask** – Userspace task controller for LinuxCNC

## SYNOPSIS

**milltask** is an internal process of LinuxCNC. It is generally not invoked directly but by an inifile setting:

**[TASK]TASK=milltask.** milltask プロセスは、以下にリストされ、inihal ユーザーコンポーネントが所有する ini。\* hal ピンを作成します。これらのピンは、LinuxCnC の実行中に変更され、通常は inifile で指定される値を変更できます。

## DESCRIPTION

初期ピンはすべてのタスクサイクルでサンプリングされますが、それらの値の影響を受けるコマンドは通常、コマンドが処理される時に存在する値を使用します。このようなコマンドには、インタープリターによって処理されるすべてのコード（Gcode プログラムおよび MDI コマンド）および GUI によって発行される NML ジョギングコマンド（halui を含む）が含まれます。ホイールジョギングはリアルタイムモーションモジュールに実装されているため、変更された値がモーションモジュールに伝播されるとすぐに、初期ピンの変更（ini.\*.max\_velocity、ini.\*.max\_acceleration など）が受け入れられます。

## PINS

### Per-joint pins (N == joint number)

#### **ini.N.backlash**

[JOINT\_N] BACKLASH の調整が可能

#### **ini.N.ferror**

[JOINT\_N] FERROR の調整が可能

#### **ini.N.min\_ferror**

[JOINT\_N] MIN\_FERROR の調整が可能

#### **ini.N.min\_limit**

[JOINT\_N] MIN\_LIMIT の調整が可能

#### **ini.N.max\_limit**

[JOINT\_N] MAX\_LIMIT の調整が可能

**ini.N.max\_velocity**

[JOINT\_N] MAX\_VELOCITY の調整が可能

**ini.N.max\_acceleration**

[JOINT\_N] MAX\_ACCELERATION の調整が可能

**ini.N.home**

[JOINT\_N] HOME の調整が可能

**ini.N.home\_offset**

[JOINT\_N] HOME\_OFFSET の調整が可能

**ini.N.home\_offset**

[JOINT\_N] HOME\_SEQUENCE の調整が可能

**Per-axis pins (L == axis letter)****ini.L.min\_limit**

[AXIS\_L] MIN\_LIMIT の調整が可能

**ini.L.max\_limit**

[AXIS\_L] MAX\_LIMIT の調整が可能

**ini.L.max\_velocity**

[AXIS\_L] MAX\_VELOCITY の調整が可能

**ini.L.max\_acceleration**

[AXIS\_L] MAX\_ACCELERATION の調整が可能

**Global pins****ini.traj\_default\_acceleration**

[TRAJ] DEFAULT\_ACCELERATION の調整が可能

**ini.traj\_default\_velocity**

[TRAJ] DEFAULT\_VELOCITY の調整が可能

**ini.traj\_max\_acceleration**

[TRAJ] MAX\_ACCELERATION の調整が可能

**motion.motion-enabled** OUT BIT

**ini.traj\_max\_velocity**

[TRAJ] MAX\_VELOCITY の調整が可能

**Global pins (arc\_blend trajectory planner)****ini.traj\_arc\_blend\_enable**

[TRAJ] ARC\_BLEND\_ENABLE の調整が可能

**ini.traj\_arc\_blend\_fallback\_enable**

[TRAJ] ARC\_BLEND\_FALLBACK\_ENABLE の調整が可能

**ni.traj\_arc\_blend\_gap\_cycles**

[TRAJ] ARC\_OPTIMIZATION\_DEPTH の調整を可能にします

**ini.traj\_arc\_blend\_optimization\_depth**

[TRAJ] ARC\_BLEND\_GAP\_CYCLES の調整が可能

ini.traj\_arc\_blend\_ramp\_freq

[TRAJ] ARC\_BLEND\_RAMP\_FREQ の調整を可能にします

**NOTES**

inihal ピンは、milltask が開始されるまで作成されないため、inifile [HAL] HALFILE アイテムで指定されたハーフファイルにリンクまたは設定することはできません。 inihal ピン値は、[APPLICATION] APP アイテムまたは[HAL] POSTGUI\_HALFILE をサポートする GUI で指定された独立した halcmd プログラムによって変更できます。

inifile は、inihal ピン設定によって変更された値で自動的に更新されませんが、[HAL] POSTGUI\_HALFILE を使用する場合は、キャリブレーションプログラム (emccalib.tcl) を使用して更新できます。

**NAME**

motion – accepts NML motion commands, interacts with HAL in realtime

**SYNOPSIS**

```
loadrt motmod [base_period_nsec=period] [base_thread_fp=0 or 1]  
[servo_period_nsec=period][traj_period_nsec=period] [num_joints=[1-9]] [num_dio=[1-64]] [num_aio=[1-64]] [num_spindles=[1-8]] [unlock_joints_mask=jointmask]
```

次の項目の制限は、コンパイル時の設定です。

Number of joints available (num\_joints) is set by EMCMOT\_MAX\_JOINTS.

Maximum number of digital inputs (num\_dio) is set by EMCMOT\_MAX\_DIO.

Maximum number of analog inputs (num\_aio) is set by EMCMOT\_MAX\_AIO.

Maximum number of spindles (num\_spindles) is set by EMCMOT\_MAX\_SPINDLES

**DESCRIPTION**

デフォルトでは、ベーススレッドは浮動小数点をサポートしていません。ソフトウェアステップング、ソフトウェアエンコーダカウント、およびソフトウェア pwm は浮動小数点を使用しません。base\_thread\_fp を使用して、ベーススレッドの浮動小数点を有効にすることができます（たとえば、ブラシレス DC モーター制御の場合）。

これらのピンとパラメーターは、リアルタイム motmod モジュールによって作成されます。このモジュールは、LinuxCNC のモーションプランナーに HAL インターフェースを提供します。基本的に、motmod はウェイポイントのリストを取り込んで、モータードライブに供給されるジョイント位置の適切にブレンドされた制約制限付きストリームを生成します。

オプションで、デジタル I/O の数は num\_dio で設定されます。アナログ I/O の数は num\_aio で設定されます。デフォルトはそれぞれ 4 です。

「joint」または「axis」で始まるピン名は、モーションコントローラ機能によって読み取られ、更新されます。

**MOTION PINS**

**motion-command-handler.time** OUT S32

モーションモジュール motion-command-handler の時間（CPU クロック単位）

**motion-controller.time** OUT S32

モーションモジュールモーションコントローラの時間（CPU クロック単位）

**motion.adaptive-feed** IN FLOAT

M52 P1 でアダプティブフィードを有効にすると、指令速度にこの値が乗算されます。この効果は、NML レベルのフィードオーバーライド値と `motion.feed-hold` で乗算されます。負の値は有効であり、G コードパスを逆に実行します。

**motion.analog-in-*NN* IN FLOAT**

これらのピンは、M66Enn の入力待機モードで使用されます。

**motion.analog-out-*NN* OUT FLOAT**

これらのピンは M67-68 で使用されます。

**motion.coord-error OUT BIT**

ソフト制限を超えるなど、モーションでエラーが発生した場合は TRUE

**motion.coord-mode OUT BIT**

モーションが「テレオペモード」ではなく「協調モード」の場合は TRUE

**motion.current-vel OUT FLOAT**

現在のデカルト速度

**motion.digital-in-*NN* IN BIT**

これらのピンは、M66Pnn 入力待機モードで使用されます。

**motion.digital-out-*NN* OUT BIT**

これらのピンは、M62～M65 ワードによって制御されます。

**motion.distance-to-go OUT FLOAT**

現在の移動で残っている距離

**motion.enable IN BIT**

このビットが FALSE に駆動されると、モーションが停止し、マシンは「マシンオフ」状態になり、オペレーターにメッセージが表示されます。通常の動作の場合、このビットを TRUE に駆動します。

**motion.eoffset-active OUT BIT**

外部オフセットがアクティブ（ゼロ以外）であることを示します

**motion.eoffset-limited** OUT BIT

外部オフセットのあるモーションがソフト制限制約 ([`AXIS_L`] `MIN_LIMIT`、`MAX_LIMIT`) によって制限されたことを示します。

**motion.feed-hold** IN BIT

M53 P1 で送り停止制御が有効になっていて、このビットが `TRUE` の場合、送り速度は 0 に設定されます。

注：フィードホールドは、ジョグではなく、`gcode` コマンドに適用されます。

**motion.feed-inhibit** IN BIT

このピンが `TRUE` の場合、`gcode` コマンドのマシンモーションは禁止されます。

このピンが `TRUE` になったときに機械が主軸同期移動を実行している場合、主軸同期動作は終了し、それ以降の移動は禁止されます（これは、機械、工具、またはワークの損傷を防ぐためです）。

このピンが `TRUE` になったときにマシンが（スピンドル同期されていない）移動の途中にある場合、マシンは最大許容加速速度で停止するまで減速します。

このピンが `FALSE` になると、モーションが再開されます。

注：`feed-inhibit` は、ジョグではなく、`gcode` コマンドに適用されます。

**motion.homing-inhibit** IN BIT

このビットが `TRUE` の場合、ジョイントホーミング移動（「ホームオール」を含む）の開始は許可されず、エラーが報告されます。デフォルトでは、モーションが有効になっている場合は常に、ジョイントモードでホーミングが許可されます。

**motion.in-position** OUT BIT

マシンが所定の位置にある場合（つまり、現在コマンド位置に向かって移動していない場合）は `TRUE`。

**motion.motion-enabled** OUT BIT**motion.motion-type** OUT S32

これらの値は、`src / emc / nml_intf / motion_types.h` からのものです。

- 0: Idle (no motion)
- 1: Traverse
- 2: Linear feed
- 3: Arc feed
- 4: Tool change
- 5: Probing
- 6: Rotary unlock for traverse

**motion.on-soft-limit** OUT BIT**motion.probe-input** IN BIT

G38.n は、このピンの値を使用して、プローブがいつ接触したかを判断します。プローブ接点が閉じている（接触している）場合は TRUE、プローブ接点が開いている場合は FALSE。

**motion.program-line** OUT S32

実行中の現在のプログラム行。実行されていない場合、またはシングルステップ中に行間でゼロ。

**motion.requested-vel** OUT FLOAT

現在要求されている速度（ユーザー単位/秒）。この値は、GコードファイルのFワード設定であり、マシンの速度と加速の制限に対応するために削減される可能性があります。このピンの値は、フィードオーバーライドまたはその他の調整を反映していません。

**motion.servo.last-period** OUT U32

サーボスレッドの呼び出し間の CPU クロックの数。通常、この数値を CPU 速度で割ると、時間が秒単位で示され、リアルタイムモーションコントローラーがタイミング制約を満たしているかどうかを判断するために使用できます。

**motion.teleop-mode** OUT BIT

モーションモードは teleop です（軸座標ジョギングが利用可能）。

**motion.tooloffset.L** OUT FLOAT

各軸の現在の工具オフセット（Lは軸の文字、次のいずれか：x y z a b c u v w）

**motion.tp-reverse** OUT BIT

軌道計画が逆になります（逆実行）



**AXIS PINS**

(**L** is the axis letter, one of: **x y z a b c u v w**)

**axis.L.eoffset** OUT FLOAT

現在の外部オフセット。

**axis.L.eoffset-clear** IN BIT

外部オフセット要求のクリア

**axis.L.eoffset-counts** IN S32

外部オフセットの入力をカウントします。 **eoffset-counts** は内部レジスタに転送されます。適用される外部オフセットは、レジスタカウントと **eoffset-scale** 値の積です。レジスタは、マシンの起動ごとにゼロにリセットされます。外部オフセットがアクティブな状態でマシンの電源がオフになっている場合は、再起動する前に **eoffset-counts** ピンをゼロに設定する必要があります。

**axis.L.eoffset-enable** IN BIT

外部オフセットを有効にします ([**AXIS\_L**] **OFFSET\_AV\_RATIO** の ini ファイル設定も必要です)

**axis.L.eoffset-request** OUT FLOAT

要求された外部オフセットのデバッグピン。

**axis.L.eoffset-scale** IN FLOAT

外部オフセットのスケール。

**axis.L.jog-accel-fraction** IN FLOAT

ホイールジョギングの加速度を、軸の **inimax\_acceleration** の一部に設定します。1 より大きい値または 0 より小さい値は無視されます。

**axis.L.jog-counts** IN S32

物理的なジョグホイールを使用するには、外部エンコーダの「カウント」ピンに接続します。

**axis.L.jog-enable** IN BIT

TRUE の場合（および手動モードの場合）、「ジョグカウント」を変更するとモーションが発生します。false の場合、「jog-counts」は無視されます。

**axis.L.jog-scale** IN FLOAT

「ジョグカウント」のカウンタごとに移動距離を機械単位で設定します。

**axis.L.jog-vel-mode** IN BIT

FALSE（デフォルト）の場合、ジョグホイールは位置モードで動作します。軸は、所要時間に関係なく、カウンタごとに正確にジョグスケールの単位で移動します。TRUE の場合、ホイールは速度モードで動作します。つまり、コマンドされたモーションが完了していない場合でも、ホイールが停止するとモーションが停止します。

**axis.L.kb-jog-active** OUT BIT

（無料プランナー軸ジョギングアクティブ（キーボードまたはハルイ））

**axis.L.pos-cmd** OUT FLOAT

軸が指示した位置。軸とモーターの座標の間には、バックラッシュ補正、ねじ誤差補正、ホームオフセットなどのオフセットがいくつかある場合があります。外部オフセットは個別に報告されます（axis.L.eoffset）。

**axis.L.teleop-pos-cmd** OUT FLOAT**axis.L.teleop-tp-enable** OUT BIT

この軸で「テレオペプランナー」が有効になっている場合は TRUE

**axis.L.teleop-vel-cmd** OUT FLOAT

軸の指令速度

**axis.L.teleop-vel-lim** OUT FLOAT

レオペプランナーの速度制限

**axis.L.wheel-jog-active** OUT BIT**JOINT PINS**

N はジョイント番号（0 ... num\_joints-1）

(注：(DEBUG)とマークされたピンはデバッグ支援として機能し、いつでも変更または削除される可能性があります。)

**joint.N.joint-acc-cmd** OUT FLOAT (DEBUG)

関節の命令された加速

**joint.N.active** OUT BIT (DEBUG)

このジョイントがアクティブな場合は TRUE

**joint.N.amp-enable-out** OUT BIT

このジョイントのアンプを有効にする必要がある場合は TRUE

**joint.N.amp-fault-in** IN BIT

このジョイントのアンプで外部障害が検出された場合は、TRUE で駆動する必要があります

**joint.N.backlash-corr** OUT FLOAT (DEBUG)

バックラッシュまたはねじ補正の生の値

**joint.N.backlash-filt** OUT FLOAT (DEBUG)

バックラッシュまたはねじ補正のフィルター値 (モーション制限を尊重)

**joint.N.backlash-vel** OUT FLOAT (DEBUG)

バックラッシュまたはねじ補正速度

**joint.N.coarse-pos-cmd** OUT FLOAT (DEBUG)**joint.N.error** OUT BIT (DEBUG)

このジョイントでリミットスイッチが閉じるなどのエラーが発生した場合は TRUE

**joint.N.f-error** OUT FLOAT (DEBUG)

実際の次のエラー

**joint.N.f-error-lim** OUT FLOAT (DEBUG)

次のエラー制限

**joint.N.f-errored** OUT BIT (DEBUG)

このジョイントが次のエラー制限を超えた場合は TRUE

**joint.N.faulted** OUT BIT (**DEBUG**)

**joint.N.free-pos-cmd** OUT FLOAT (**DEBUG**)

「フリーランナー」はこの関節の位置を命じました。

**joint.N.free-tp-enable** OUT BIT (**DEBUG**)

このジョイントで「フリーランナー」が有効になっている場合は TRUE

**joint.N.free-vel-lim** OUT FLOAT (**DEBUG**)

フリーランナーの速度制限

**joint.N.home-state** OUT S32 (**DEBUG**)

ホーミングステートマシンの状態

**joint.N.home-sw-in** IN BIT

このジョイントのホームスイッチが閉じている場合は、TRUE で駆動する必要があります

**joint.N.homed** OUT BIT (**DEBUG**)

ジョイントがホームになっている場合は TRUE

**joint.N.homing** OUT BIT

ジョイントが現在ホーミングしている場合は TRUE

**joint.N.in-position** OUT BIT (**DEBUG**)

ジョイントが「フリーランナー」を使用していて停止した場合は TRUE

**joint.N.index-enable** IO BIT

インデックスパルスへのホーミングを可能にするには、ジョイントのエンコーダのインデックスイネーブルピンに接続する必要があります

**joint.N.is-unlocked** IN BIT

ジョイントがロック解除されていることを示します (JOINT UNLOCK PINS を参照)。

**joint.N.jog-accel-fraction** IN FLOAT

ホイールジョギングの加速度を、ジョイントの `inimax_acceleration` の一部に設定します。1より大きい値または0より小さい値は無視されます。

#### **joint.N.jog-counts** IN S32

物理的なジョグホイールを使用するには、外部エンコーダの「カウント」ピンに接続します。

#### **joint.N.jog-enable** IN BIT

TRUE の場合（および手動モードの場合）、「ジョグカウント」を変更するとモーションが発生します。false の場合、「jog-counts」は無視されます。

#### **joint.N.jog-scale** IN FLOAT

「ジョグカウント」のカウントごとに移動距離を機械単位で設定します。

#### **joint.N.jog-vel-mode** IN BIT

FALSE（デフォルト）の場合、ジョグホイールは位置モードで動作します。ジョイントは、所要時間に関係なく、カウントごとに正確にジョグスケールの単位を移動します。TRUE の場合、ホイールは速度モードで動作します。つまり、コマンドされたモーションが完了していない場合でも、ホイールが停止するとモーションが停止します。

#### **joint.N.kb-jog-active** OUT BIT (DEBUG)

（無料プランナージョイントジョギングアクティブ（キーボードまたはハルイ））

#### **joint.N.motor-offset** OUT FLOAT (DEBUG)

ジョイントがホームになっているときに確立されるジョイントモーターオフセット。

#### **joint.N.motor-pos-cmd** OUT FLOAT

このジョイントのコマンド位置。

#### **joint.N.motor-pos-fb** IN FLOAT

このジョイントの実際の位置。

#### **joint.N.neg-hard-limit** OUT BIT (DEBUG)

ジョイントの負のハード制限

#### **joint.N.neg-lim-sw-in** IN BIT

このジョイントの負のリミットスイッチが作動した場合は、TRUE で駆動する必要があります。

#### **joint.N.pos-cmd** OUT FLOAT

ジョイント（モーターではなく）のコマンド位置。ジョイント座標とモーター座標の間には、バックラッシュ補正、ねじエラー補正、ホームオフセットなどのオフセットがいくつかある場合があります。

#### **joint.N.pos-fb** OUT FLOAT

ジョイントフィードバック位置。この値は、実際のモーター位置からジョイントオフセットを引いたものから計算されます。機械の視覚化に役立ちます。

#### **joint.N.pos-hard-limit** OUT BIT (DEBUG)

ジョイントの正のハード制限

#### **joint.N.pos-lim-sw-in** IN BIT

このジョイントの正のリミットスイッチが作動した場合は、TRUE で駆動する必要があります。

#### **joint.N.unlock** OUT BIT

軸がロックされたジョイント（通常はロータリー）であり、移動がコマンドされている場合はTRUE（JOINT UNLOCK PINS を参照）。

#### **joint.N.joint-vel-cmd** OUT FLOAT (DEBUG)

ジョイントの指令速度

#### **joint.N.wheel-jog-active** OUT BIT (DEBUG)

## JOINT UNLOCK PINS

ジョイントのロック解除に使用されるジョイントピン（`joint.N.unlock`、`joint.N.is-unlocked`）は、`motmod` の `unlock_joints_mask = jointmask` パラメーターに従って作成されます。これらのピンは、インデクサー（通常はロータリージョイント）をロックするために必要になる場合があります。

ジョイントマスクビットは次のとおりです：（lsb）0 : joint0、1 : joint1、2 : joint2、...

例：`loadrt motmod ... unlock_joints_mask = 0x38` は、ジョイントのロック解除ピンを作成します 3,4,5

**SPINDLE PINS**

(Mはスピンドル番号 (0 ... num\_spindles-1) )

**spindle.M.amp-fault-in** IN BIT

このスピンドルのアンプで外部障害が検出された場合は、TRUEで駆動する必要があります

**spindle.M.at-speed** IN BIT

次の条件下では、このピンがTRUEになるまでモーションが一時停止します。各スピンドルの開始または速度変更後の最初の送りの移動前。スピンドル同期移動のすべてのチェーンの開始前。CSSモードの場合、すべての高速->フィード遷移で。

**spindle.M.brake** OUT BIT

スピンドルブレーキをかける必要がある場合はTRUE

**spindle.M.forward** OUT BIT

スピンドルが前方に回転する必要がある場合はTRUE

**spindle.M.index-enable** I/O BIT

スピンドル同期移動を正しく動作させるには、この信号をスピンドルエンコーダのインデックスイネーブルピンにフックする必要があります。

**spindle.M.inhibit** IN BIT

TRUEの場合、スピンドル速度は0に設定および保持されます。

**spindle.M.is-oriented** IN BIT

スピンドル方向の確認ピン。オリентサイクルを完了します。主軸方向がアサートされたときに主軸方向が真の場合、主軸方向ピンがクリアされ、主軸ロックピンがアサートされます。また、スピンドルブレーキピンがアサートされます。

**spindle.M.locked** OUT BIT

スピンドルは完全なピンを方向付けます。M3、M4、M5のいずれかによってクリアされます。

**spindle.M.on** OUT BIT

スピンドルが回転する必要がある場合はTRUE

**spindle.M.orient** OUT BIT

スピンドルオリентサイクルの開始を示します。M19によって設定されます。  
M3、M4、M5のいずれかによってクリアされます。

スピンドルオリентがtrueのときにスピンドルオリентフォールトがゼロでない場合、  
M19 コマンドはエラーメッセージで失敗します。

**spindle.M.orient-angle** OUT FLOAT

M19に必要なスピンドルの向き。M19Rワードパラメータの値と[RS274NGC]  
ORIENT\_OFFSETiniパラメータの値。

**spindle.M.orient-fault** IN S32

オリентサイクルの故障コード入力。ゼロ以外の値を指定すると、オリентサイクルが中  
止されます。

**spindle.M.orient-mode** OUT BIT

必要なスピンドル回転モード。M19Pパラメータワードを反映します。

**spindle.M.reverse** OUT BIT

スピンドルが後方に回転する必要がある場合はTRUE

**spindle.M.revs** IN FLOAT

主軸同期移動を正しく動作させるには、この信号を主軸エンコーダの位置ピンに接続する必要  
があります。

**spindle.M.speed-cmd-rps** FLOAT OUT

毎秒回転数の単位での指令スピンドル速度

**spindle.M.speed-in** IN FLOAT

1秒あたりの回転数で表した実際のスピンドル速度フィードバック。G96（一定の表面速度）  
およびG95（1回転あたりの送り）モードに使用されます。

**spindle.M.speed-out** OUT FLOAT

1分あたりの回転数での望ましいスピンドル速度

**spindle.M.speed-out-abs** OUT FLOAT



1分あたりの回転数で表した望ましいスピンドル速度。スピンドルの方向に関係なく常に正です。

**spindle.M.speed-out-rps** OUT FLOAT

1秒あたりの回転数で表した望ましいスピンドル速度

**spindle.M.speed-out-rps-abs** OUT FLOAT

スピンドルの方向に関係なく常に正の、1秒あたりの回転数で表した望ましいスピンドル速度。

## MOTION PARAMETERS

パラメータの多くはデバッグの補助として機能し、いつでも変更または削除される可能性があります。

**motion-command-handler.tmax** RW S32

これらの HAL 関数の実行時間に関する情報を CPU クロックで表示します

**motion-command-handler.tmax-increased** RO S32

**motion-controller.tmax** RW S32

これらの HAL 関数の実行時間に関する情報を CPU クロックで表示します

**motion-controller.tmax-increased** RO BIT

**motion.debug-\***

これらの値は、デバッグの目的で使用されます。

## FUNCTIONS

通常、これらの機能は両方とも、示されている順序でサーボスレッドに追加されます。

**motion-command-handler**

ユーザースペースからのモーションコマンドを処理します。 `motion-command-handler` という名前のピン。時間とパラメータ `motion-command-handler.tmax`、`tmax-increased` がこの関数用に作成されます。

**motion-controller**

LinuxCNC モーションコントローラーを実行します。 `motion-controller.time` という名前のピンと `motion-controller.tmax`、`tmax-increased` パラメーターがこの関数用に作成されます。

**BUGS**

このマニュアルページは不完全です

(DEBUG) で分類されたピンの識別は疑わしいです。

**SEE ALSO**

[iocontrol\(1\)](#), [milltask\(1\)](#)

**NAME**

pid – proportional/integral/derivative controller

**SYNOPSIS**

**loadrt pid [num\_chan=num | names=name1[,name2...]] [debug=dbg]**

**DESCRIPTION**

pid は、位置または速度のフィードバックループを制御するために使用される、古典的な比例/積分/微分コントローラーです。サーボモーターおよびその他の閉ループアプリケーション用。pid は最大 16 個のコントローラーをサポートします。実際にロードされる数は、モジュールがロードされるときに num\_chan 引数によって設定されます。または、names = と一意の名前を区切って指定しますカンマで。

num\_chan = および names = 指定子は相互に排他的です。num\_chan = も names = も指定されていない場合、デフォルト値は 3 です。debug が 1 に設定されている場合（デフォルトは 0）、いくつかの追加の HAL パラメーターがエクスポートされます。これはチューニングに役立つ場合がありますが、それ以外の場合は不要です。

**NAMING**

ピン、パラメーター、および関数の名前には、次の接頭辞が付いています。

**pid.N.** for N=0,1,...,num-1 when using **num\_chan=num**

**nameN.** for nameN=name1,name2,... when using **names=name1,name2,...**

pid.N. 以下の説明にフォーマットを示します。

**FUNCTIONS**

**pid.N.do-pid-calcs** (uses floating-point) Does the PID calculations for control loop N.

**PINS**

**pid.N.command** float in

制御ループに必要な（コマンドされた）値。

**pid.N.Pgain** float in

比例ゲイン。結果は、エラーに Pgain を掛けたものである出力への寄与になります。

**pid.N.Igain** float in

積分ゲイン。結果は、エラーに Igain を掛けた積分である出力への寄与になります。たとえば、10 秒間続いた 0.02 のエラーは 0.2 の積分エラー（errorI）になり、Igain が 20 の場合、積分項は出力に 4.0 を追加します。

**pid.N.Dgain** float in

微分ゲイン。結果は、エラーの変化率（導関数）に Dgain を掛けたものである出力への寄与になります。たとえば、0.2 秒で 0.02 から 0.03 に変化したエラーは、0.05 のエラー導関数（errorD）になり、Dgain が 5 の場合、導関数項は出力に 0.25 を追加します。

**pid.N.feedback** float in

エンコーダーなどのセンサーからの実際の（フィードバック）値。

**pid.N.output** float out

モーターなどのアクチュエーターに送られる PID ループの出力。

**pid.N.command-deriv** float in

制御ループの目的の（コマンドされた）値の導関数。信号が接続されていない場合、導関数は数値的に推定されます。

#### **pid.N.feedback-deriv** float in

制御ループの実際の（フィードバック）値の導関数。信号が接続されていない場合、導関数は数値的に推定されます。フィードバックが量子化された位置ソース（エンコーダーフィードバック位置など）からのものである場合、エンコーダー（9）または hostmot2（9）の速度出力など、ここでより適切な速度推定を使用することにより、D項の動作を改善できます。

#### **pid.N.error-previous-target** bit in

モーションコントローラが期待するように、エラー計算には以前の呼び出しのターゲットと現在の位置を使用します。これにより、速度に依存する追従誤差を排除することにより、トルクモード位置ループおよび大きなIゲインを必要とするループの調整が容易になる場合があります。

#### **pid.N.error** float out

コマンドとフィードバックの差。

#### **pid.N.enable** bit in

true の場合、PID 計算を有効にします。false の場合、出力はゼロになり、すべての内部積分器などがリセットされます。

#### **pid.N.index-enable** bit in

index-enable の立ち下がりエッジで、pid は内部コマンド微分推定を更新しません。エンコーダインデックスパルスを使用するシステムでは、このピンをインデックスイネーブル信号に接続する必要があります。これが行われず、FF1 がゼロ以外の場合、入力コマンドのステップ変更により、PID 出力に 1 サイクルのスパイクが発生します。-deriv 入力のみを使用するシステムでは、これは D 項にも影響します。

#### **pid.N.bias** float in

バイアスは、出力に追加される一定量です。ほとんどの場合、ゼロのままにしておく必要があります。ただし、サーボアンプのオフセットを補正したり、垂直方向に移動するオブジェクトの重量のバランスをとったりすると便利な場合があります。出力の他のすべてのコンポーネントと同様に、PID ループが無効になると、バイアスはオフになります。PID ループが無効になっている場合でもゼロ以外の出力が必要な場合は、外部 HALsum2 ブロックを追加する必要があります。

#### **pid.N.FF0** float in

ゼロ次フィードフォワード項。FF0 にコマンド値を掛けたものが出力に寄与します。位置ループの場合、通常はゼロのままにしておく必要があります。速度ループの場合、FF0 は摩擦またはモーターカウンタ EMF を補正でき、適切に使用すればより良いチューニングが可能になる場合があります。

#### **pid.N.FF1** float in

一次フィードフォワード項。FF1 にコマンド値の導関数を掛けた出力への寄与を生成します。位置ループの場合、寄与は速度に比例し、摩擦またはモーター CEMF を補正するために使用できます。速度ループの場合、加速度に比例し、慣性を補正できます。どちらの場合も、適切に使用すると、チューニングが向上する可能性があります。

#### **pid.N.FF2** float in

2次フィードフォワード項。FF2にコマンド値の2次導関数を掛けた出力への寄与を生成します。位置ループの場合、寄与は加速度に比例し、慣性を補正するために使用できます。速度ループの場合、寄与はジャークに比例し、通常はゼロのままにする必要があります。

**pid.N.FF3 float in**

3次フィードフォワード項。FF3にコマンド値の3次導関数を掛けたものが出力に寄与します。位置ループの場合、寄与はジャークに比例し、加速中の残留誤差を補正するために使用できます。速度ループの場合、寄与はsnap (jounce) に比例し、通常はゼロのままにする必要があります。

**pid.N.deadband float in**

「許容可能な」エラーの範囲を定義します。エラーの絶対値が不感帯よりも小さい場合、エラーがゼロであるかのように扱われます。本質的に量子化されたエンコーダーなどのフィードバックデバイスを使用する場合、コマンドが2つの隣接するエンコーダー値の間にある場合に制御ループが前後にハンチングするのを防ぐために、不感帯は半分のカウントよりわずかに大きく設定する必要があります。エラーの絶対値が不感帯よりも大きい場合、不感帯のエッジでの伝達関数のステップを防ぐために、ループ計算を実行する前に不感帯の値がエラーから差し引かれます。(バグを参照してください。)

**pid.N.maxoutput float in**

出力制限。maxoutputがゼロでない限り、出力の絶対値がmaxoutputを超えることはできません。出力が制限されている場合、ウィンドアップとオーバーシュートを防ぐために、エラー積分器は積分する代わりに保持します。

**pid.N.maxerror float in**

P、I、およびDに使用される内部エラー変数の制限。エラーが大きい場合(コマンドがステップ変更を行う場合など)に、高いPgain値が大きな出力を生成しないようにするために使用できます。通常は必要ありませんが、非線形システムを調整するときに役立ちます。

**pid.N.maxerrorD float in**

エラー導関数の制限。Dgain項で使用されるエラーの変化率は、値がゼロでない限り、この値に制限されます。Dgainの影響を制限し、コマンドやフィードバックのステップによる大きな出力スパイクを防ぐために使用できます。通常は必要ありません。

**pid.N.maxerrorI float in**

エラー積分器の制限。Igain項で使用されるエラー積分器は、ゼロでない限り、この値に制限されます。インテグレータのウィンドアップと、エラーの発生中または発生後に発生するオーバーシュートを防ぐために使用できます。通常は必要ありません。

**pid.N.maxcmdD float in**

コマンド派生物の制限。FF1で使用されるコマンド導関数は、値がゼロでない限り、この値に制限されます。コマンドにステップ変更がある場合に、FF1が大きな出力スパイクを生成するのを防ぐために使用できます。通常は必要ありません。

**pid.N.maxcmdDD float in**

コマンドの二次導関数の制限。FF2で使用されるコマンドの2次導関数は、値がゼロでない限り、この値に制限されます。コマンドにステップ変更がある場合に、FF2が大きな出力スパイクを生成するのを防ぐために使用できます。通常は必要ありません。

**pid.N.maxcmdDDD float in**

コマンドの三階導関数の制限。FF3で使用されるコマンドの3次導関数は、値がゼロでない限り、この値に制限されます。コマンドにステップ変更がある場合に、FF3が大きな出力スパイクを生成するのを防ぐために使用できます。通常は必要ありません。

**pid.N.saturated** bit out

true の場合、現在の PID 出力は飽和しています。あれは、

**output = ± maxoutput.**

**pid.N.saturated-s** float out

**pid.N.saturated-count** s32 out

true の場合、PID の出力は、この数秒間 (saturated-s) または期間 (saturated-count) 継続的に飽和していました。

## PARAMETERS

**pid.N.errorI** float ro (only if debug=1)

エラーの積分。これは、出力の積分項を生成するために Igain を掛けた値です。

**pid.N.errorD** float ro (only if debug=1)

エラーの派生物。これは、出力の微分項を生成するために Dgain を掛けた値です。

**pid.N.commandD** float ro (only if debug=1)

コマンドの派生物。これは、出力の1次フィードフォワード項を生成するために FF1 を掛けた値です。

**pid.N.commandDD** float ro (only if debug=1)

コマンドの2次導関数。これは、出力の2次フィードフォワード項を生成するために FF2 を掛けた値です。

**pid.N.commandDDD** float ro (only if debug=1)

コマンドの3次導関数。これは、出力の3次フィードフォワード項を生成するために FF3 を掛けた値です。

## BUGS

一部の人は、デッドバンドは、エラーがデッドバンド内にある場合はゼロとして扱われ、デッドバンド外にある場合は変更されないように実装する必要があると主張します。これは、不感帯のサイズに等しい伝達関数のステップを引き起こすため、実行されませんでした。その振る舞いを好む人は、振る舞いを変更するパラメーターを追加するか、独自のバージョンの pid を作成することを歓迎します。ただし、デフォルトの動作は変更しないでください。

負のゲインは、望ましくない動作につながる可能性があります。状況によっては、負の FF ゲインが理にかなっている可能性があります。一般に、すべてのゲインは正である必要があります。一部の出力が間違った方向にある場合、それを修正するためにゲインを否定するのは間違いです。代わりに、他の場所でスケールリングを正しく設定してください。

**NAME**

spindle – Control a spindle with different acceleration and deceleration and optional gear change scaling

**SYNOPSIS**

**loadrt spindle [count=N|names=name1[,name2...]]**

**DESCRIPTION**

このコンポーネントは、調整可能な加速と減速でスピンドルを制御します。

注：このコンポーネントには残念ながら名前が付けられており、モーションコンポーネントによって作成された名前と非常によく似ていますほとんどの場合、これはあなたがいるドキュメントページではありません探している。代わりに、<http://linuxcnc.org/docs/html/man/man9/motion.9.html>を参照してください。

これは、DCドライブやインバーターなど、個別のフォワード/リバース入力を備えた非サーボスピンドルドライブで使用するよう設計されています。スピンドルエンコーダが利用可能な場合は、スピンドル負荷に合わせて加速と減速を調整するために使用されます。そうでない場合は、スピンドル速度がシミュレートされます。このコンポーネントでは、最大16個のギアを備えたギアボックスを使用できます。各ギアには、速度、加速、ドライバーのゲイン、方向を個別に制御できます。

**FUNCTIONS**

**spindle.N** (requires a floating-point thread)

**PINS**

**spindle.N.select-gear** u32 in

ギアを選択します。0->使用可能なギアの数-1の範囲内である必要があります。これを使用する場合は、select.x 入力ピンを使用しないでください。

**spindle.N.commanded-speed** float in

指令スピンドル速度 (RPM)

**spindle.N.actual-speed** float in

スピンドルエンコーダからの実際のスピンドル速度 (RPS) スピンドルエンコーダがない場合は、simulate\_encoder パラメータを1に設定します。

**spindle.N.simulate-encoder** bit in

エンコーダーがない場合は、これを 1 に設定します。

**spindle.N.enable** bit in

FALSE の場合、スピンドルはギアの最大減速度で停止します

**spindle.N.spindle-lpf** float in

速度が上がっているときは、スピンドル-rpm-abs 出力を滑らかにします。0 =無効。スピンドルの安定性に応じて、適切な値はおそらく 1~20 です。

**spindle.N.spindle-rpm** float out

現在のスピンドル速度 (RPM)。+ve =順方向、-ve =逆方向。可能な場合はエンコーダ入力を使用します。そうでない場合は、シミュレートされたエンコーダ速度を使用します。

**spindle.N.spindle-rpm-abs** float out

RPM 単位の絶対スピンドル速度。スピンドル速度表示に便利

**spindle.N.output** float out

スケーリングされた出力

**spindle.N.current-gear** u32 out

現在選択されているギア。

**spindle.N.at-speed** bit out

スピンドルが高速の場合は TRUE

**spindle.N.forward** bit out

正転の場合は TRUE

**spindle.N.reverse** bit out

逆回転の場合は TRUE。スピンドルが停止しているとき、順方向と逆方向の両方が偽です。

**spindle.N.brake** bit out

減速時に TRUE

**spindle.N.zero-speed** bit out

スピンドルが静止している場合は TRUE



**spindle.N.limited** bit out

指令されたスピンドル速度が > max または < min の場合に TRUE。

**NOTES**

次のピンは、「gears =」パラメータに応じて作成されます。

各ピンの 1 つは、ギアごとに作成されます。歯車が指定されていない場合は、1 つの歯車が作成されます。たとえば、コマンドラインに gears = 1 がある場合、2 つのスケールピンがあります。

**spindle.N.scale.0****spindle.N.scale.1****spindle.N.scale.x** float in

出力をスケールリングします。複数のギアの場合、ギアごとに異なるスケールを使用します。一部のギアの出力を逆にする必要がある場合は、負のスケールを使用してください。

**spindle.N.min.x** float in

許可される最小速度を設定します (RPM 単位)。指令速度が 0RPM から最小速度の間にある間、制限出力は TRUE になります。

**spindle.N.max.x** float in

許可される最大速度を設定します (RPM 単位)。指令速度がこの値を超えている間、制限出力は TRUE になります

**spindle.N.accel.x** float in

最大加速度を設定します。スピンドルエンコーダがない場合、これは RPM /秒です。エンコーダを使用している場合、出力は実際の速度にこの値を加えたものになります。このように、加速度はスピンドルの負荷に依存する可能性があります。

**spindle.N.decel.x** float in

最小減速度を設定します。スピンドルエンコーダがない場合、これは RPM /秒です。エンコーダを使用している場合、出力は実際の速度からこの値を引いたものになります。

**spindle.N.speed-tolerance.x** float in

「at-speed」信号の許容値（RPM）。指令速度のこの量内の実際のスピンドル速度により、at-speed 信号が TRUE になります。

**spindle.N.zero-tolerance.x** *float in*

「ゼロ速度」信号の許容値（RPM）。

**spindle.N.offset.x** *float in*

出力コマンドは、この量（RPM 単位）でオフセットされます。

**spindle.N.select.x** *bit in*

このギアを選択します。アクティブな選択入力がない場合は、ギア 0 が選択されます。複数の選択入力がアクティブな場合、最も高いものが選択されます。

## LICENSE

GPL