



FAULHABER

ファームウェア TMCL MCST 3601

取扱説明書

WE CREATE MOTION

インプリント

バージョン

第 1 版、01.10.2014

Copyright

FAULHABER PRECISTEP SA

Rue des Gentianes 53 · 2300 La Chaux-de-Fonds · Switzerland

翻訳を含む転載禁止

本書のいかなる部分についても、書面による **FAULHABER PRECISTEP SA** の事前の許諾なくして、情報システムへ複製、再生、保存、あるいは他の形式に加工または転送することは禁止されています。

本技術説明書は細心の注意を払って作成されています。

FAULHABER PRECISTEP SA は本取扱説明書のあらゆる誤記および誤記より生じる結果に関して何ら責任を負いかねます。同様に、機器の不適切な使用に起因する直接的および結果的損害に関しましても何ら責任を負いかねます。

本ソフトウェアを使用される際には、安全工学および干渉抑制にまつわる関連規制ならびに本技術説明書に記載される要件に留意し遵守する必要があります。

仕様は予告なしに変更されることがあります。

本技術説明書の最新版は **FAULHABER** のウェブサイトより入手できます。

www.faulhaber.com

ファームウェアバージョン V1.33

TMCL™ファームウェア説明書



MCST3601

1 軸ステップ

コントローラ／ドライバ

3 軸コントローラ

マスタ／スレーブ動作

最大 1A／36V

インクリメンタルエンコーダ入力

GPIO

本機の特徴：

- ・ PRECIsstep[®]製ステッパモータの全レンジとの互換性
- ・ コンパクトかつ完全プログラマブル
- ・ ASIC 設計



POWERED BY:TRINAMIC MOTION CONTROL

目次

1 特徴	7
2 概要	8
3 モジュールの運転	9
3.1 基本設定.....	10
3.1.1 モジュールの接続.....	10
3.1.2 TMCL-IDE ソフトウェア開発環境の始動.....	13
3.1.3 TMCL™ダイレクトモードの使用.....	14
3.1.4 重要モータ設定.....	15
4 TMCL™および TMCL-IDE 17	18
4.1 バイナリコマンド形式.....	18
4.2 回答形式.....	19
4.2.1 状態コード.....	19
4.3 スタンドアロンアプリケーション.....	20
4.3.1 簡易 TMCL™プログラムを用いたテスト.....	20
4.4 TMCL™コマンド概要.....	20
4.4.1 TMCL™コマンド.....	20
4.4.2 対象領域別コマンドリスト.....	22
4.5 コマンド.....	26
4.5.1 ROR（右回転）.....	26
4.5.2 ROL(左回転).....	27
4.5.3 MST（モータ停止）.....	28
4.5.4 MVP（指定位置へ移動）.....	29
4.5.5 SAP（軸パラメータの設定）.....	31
4.5.6 GAP（軸パラメータの取得）.....	32
4.5.7 STAP（軸パラメータの保存）.....	33
4.5.8 RSAP（軸パラメータの復元）.....	34
4.5.9 SGP（グローバルパラメータの設定）.....	35
4.5.10 GGP（グローバルパラメータの取得）.....	36
4.5.11 STGP（グローバルパラメータの保存）.....	37
4.5.12 RSGP（グローバルパラメータの復元）.....	38
4.5.13 RFS（基準点サーチ）.....	39
4.5.14 SIO（設定出力）.....	40
4.5.15 GIO（入力／出力の取得）.....	42
4.5.16 CALC（計算）.....	44
4.5.17 COMP（比較）.....	45
4.5.18 JC（条件付きジャンプ）.....	46
4.5.19 JA（常にジャンプ）.....	47
4.5.20 CSUB（サブルーチン呼び出し）.....	48
4.5.21 RSUB（サブルーチンから復帰）.....	49

4.5.22 WAIT (イベント発生待機)	50
4.5.23 STOP (TMCL™プログラム実行の停止)	52
4.5.24 SCO (座標設定)	53
4.5.25 GCO (座標取得)	54
4.5.26 CCO (座標キャプチャ)	55
4.5.27 ACO (アキュムレータ→座標)	56
4.5.28 CALCX (X レジスタを使用した計算)	57
4.5.29 AAP (アキュムレータ→ 軸パラメータ)	58
4.5.30 AGP (アキュムレータ→グローバルパラメータ).....	59
4.5.31 CLE (エラーフラグのクリア)	60
4.5.32 VECT (割り込みベクトルの設定)	61
4.5.33 EI (割り込み有効化)	62
4.5.34 DI (割り込み無効化)	63
4.5.35 RETI (割り込みから復帰)	64
4.5.36 お客様固有の TMCL™コマンド拡張 (UF0～UF7 - ユーザ機能)	64
4.5.37 目標位置到達イベント	65
4.5.38 TMCL™制御機能	66
5 カスタム指定機能	67
6 軸パラメータ	68
6.1 基準点サーチ.....	75
6.1.1 基準点サーチモード (軸パラメータ 193)	76
6.2 エンコーダ	78
6.2.1 エンコーダのプリスケール値の変更	79
6.3 計算：速度および加速 vs. マイクロステップ/フルステップ周波数.....	80
6.3.1 マイクロステップ周波数.....	81
6.3.2 フルステップ周波数.....	81
7 グローバルパラメータ	83
7.1 バンク 0.....	83
7.2 バンク 1.....	85
7.3 バンク 2.....	85
7.4 バンク 3.....	86
8 TMCL™プログラミング技術および構成	87
8.1 初期化.....	87
8.2 メインループ..... エラー! ブックマークが定義されていません。	
8.3 記号定数の使用..... エラー! ブックマークが定義されていません。	
8.4 変数の使用	88
8.5 サブルーチンの使用.....	88
8.6 ダイレクトモードとスタンドアロンモードの混合	89
9 製品ライフサポートポリシー	90
10 変更履歴	91
10.1 ファームウェア変更.....	91
10.2 文書改訂	91

11 参考文献.....	92
--------------	----

1 特徴

MCST3601 は 2 相バイポーラスステップモータ向け単軸コントローラ／ドライバモジュールです。本機は DC36V までの電源電圧および 1A RMS までのモータ電流に対応しています (ソフトウェア上の設定と 2 個のジャンパスイッチによる設定でそれぞれモータ電流の設定ができます)。TMCL™ファームウェアにより、スタンドアロン運転および直接モードの両方が可能になります。モジュールは、1 個のオンボードドライバに加え最大 2 個の外部ドライバを制御するマスタ (コントローラ+ドライバ) またはステップ／方向／イネーブル入力を受け付けるスレイブ (ドライバのみ) として構成できます。

主な特徴

モーションコントローラ

- ・リアルタイムでのモーションプロファイル算出
- ・運転中のモータパラメータの変更 (例: 位置、速度、加速度)
- ・全体システム制御およびシリアル通信プロトコル処理用高性能マイクロコントローラ

バイポーラスステップモータ用ドライバ

- ・フルステップ当たり最大 256 個のマイクロステップ
- ・高効率の動作、低い電力散逸
- ・動的電流制御
- ・内蔵保護機能

インターフェース

- ・USB 機器インターフェース (オンボード mini USB コネクタ)

オープンドレイン出力 6 個 (24V 互換)

- ・REF_L／REF_R／HOME スイッチ入力 (プログラマブルプルアップにより 24V 互換)
- ・オンボードドライバ用 S/D 入力×1 (オンボードモーションコントローラの無効化可能)
- ・二つの独立した外部ドライバ用ステップ／方向出力×2 (オンボード出力は含まず)
- ・インクリメンタル A/B/I エンコーダ用エンコーダ入力×1
- ・汎用デジタル入力×3 (24V 互換)
- ・アナログ入力×1 (0～10V)

コネクタピンが共有されていますので、全ての機能が同時に使用できるわけではありませんのでご注意ください。

ソフトウェア

- ・TMCL: スタンドアロン運転または遠隔操作、最大 2048 個の TMCL コマンドを記憶可能なプログラムメモリ (不揮発性)、および無償で提供されるアプリケーション開発 PC ソフトウェア TMCL-IDE

電気／機械仕様

- ・電源電圧: 公称 DC+24V (DC9～36V)
- ・モータ電流: 最大 1A RMS／1.5A ピーク (プログラマブル)
- ・ボード寸法: 68mm×47.5mm

2 概要

MCST3601 のマイクロプロセッサ上で実行されるソフトウェアは、ブートローダおよびファームウェア自身の二つの部分から構成されます。ブートローダが **TRINAMIC** での製造およびテスト時にインストールされ、そのライフサイクル中は手を加えられることがない一方で、ファームウェアはユーザによりアップデートできます。

プロトコルとコマンドに関しては、ファームウェアは標準 **TMCL™** ファームウェアと関連があります。同様に、本モジュールは **TMC429** ステップモータおよび **TMC260** パワードライバをベースにしており、特別な値の範囲で標準 **TMCL™** をサポートしています。

TMC260 は、エネルギー効率がよい、高電流・高精度のバイポーラステップモータ用マイクロステップドライバ IC です。

本ユニットで使用可能なコマンドおよびパラメータは全て以下のページで説明しています。

3 モジュールの運転

本章では、モジュールを運転するための基本情報を説明します。基本情報には、TMCL™プログラムの簡単な例およびダイレクトモードでのモジュールの運転に関する短い説明が含まれます。

MCST3601 は最大 3 個のモータの制御が可能です。本章では、1 個のモータ（モータ番号 0）のみで運転を開始する手順を説明します。2 個以上のモータの制御にモジュールを使用する場合は、ハードウェア説明書を参照してください。ハードウェア説明書には拡張に関する情報が記載されています。

必要なもの

- MCST3601 および適切なステッパモータ
- 公称電源電圧+DC24V (DC+9〜+36V)のモジュール用電源
- USB インターフェース付き PC
- TMCL-IDE プログラム (www.trinamic.com から無償でダウンロード可能。TMCL-IDE 取扱説明書も参照してください。)
- ケーブル類 - 少なくとも電源ケーブル、通信ケーブル、およびモータ用ケーブル

注意事項

コネクタと短絡ピンを混同しないでください。

I/O 配線とモータ電源配線を絡めないでください。

DC+36V の最大電源供給を超えないようにしてください！

通電中にモータと接続・切断しないでください！

通電していない状態で作業を開始してください！

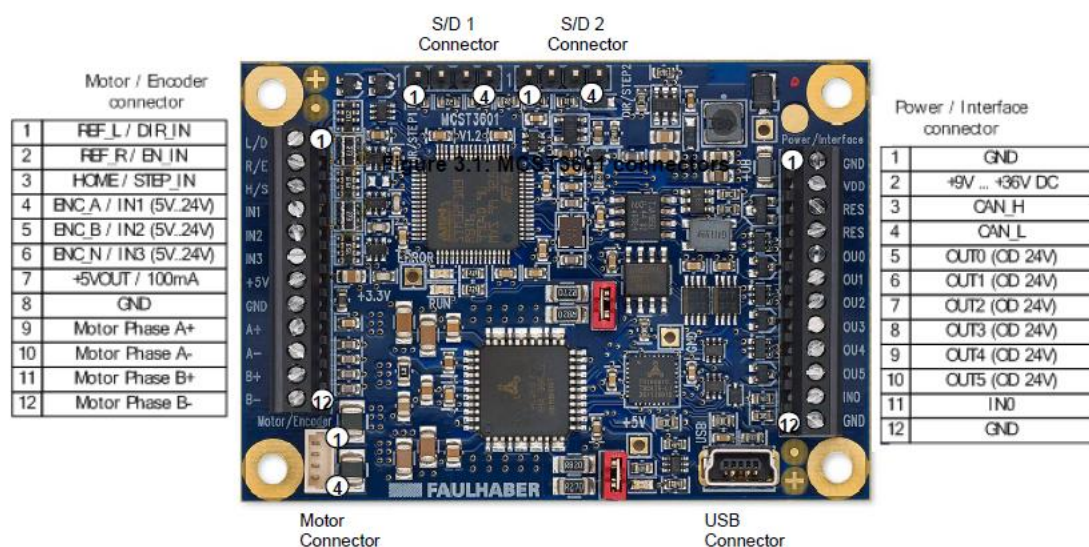


図 3.2 MCST3601 コネクタ概観

3.1 基本設定

以下のセクションでは、ユニットの接続およびモータの初回運転の手順について説明します。

3.1.1 モジュールの接続

まず、電源を接続し、通信用に PC と MCST3601 の USB インターフェースを接続する必要があります。

3.1.1.1 通信

3.1.1.1.1 USB

USB インターフェースを使用する前に、デバイスドライバをインストールする必要があります。

ラベル	コネクタ種類	相手コネクタ種類
Mini USB コネクタ	モレックス 500075-1517 B タイプ Mini USB、縦向きポート	標準 Mini USB プラグ全種

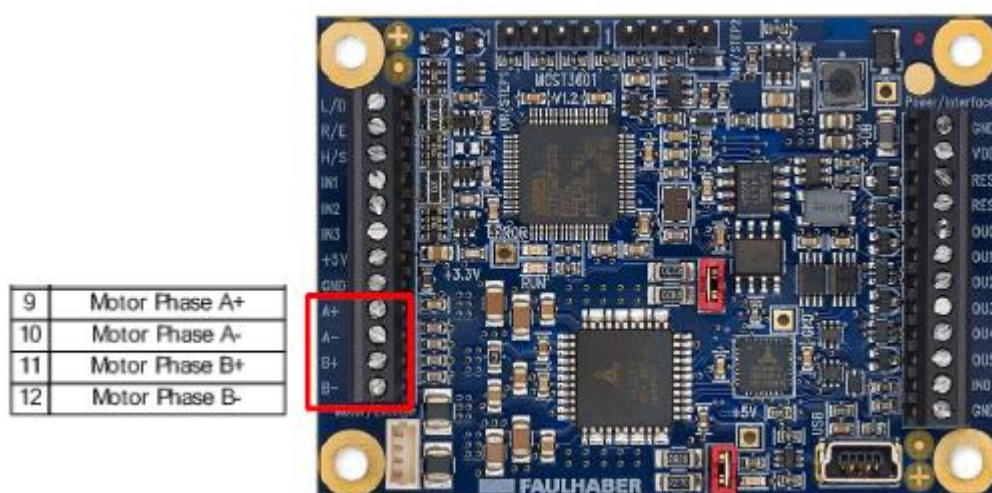
3.1.1.2 モータ

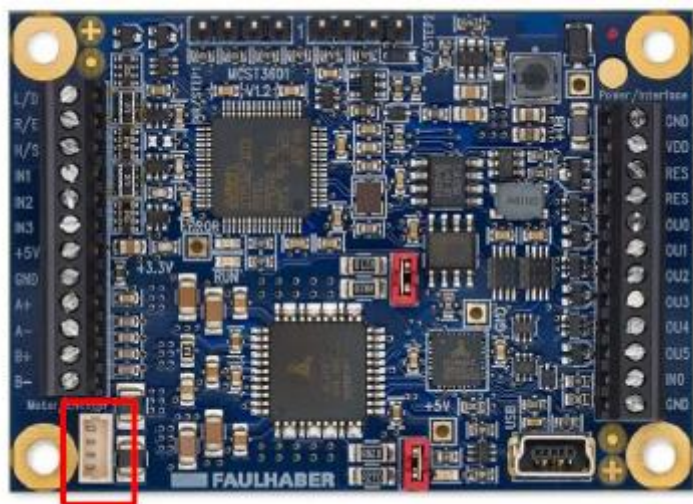
MCST3601 は 1 個目の 2 相ステップモータを直接制御し、駆動します (2 個目および 3 個目のモータは追加外部駆動機器によります)。モータのコイルの一つを「A+」および「A-」と表示のある端子に、別のコイルを「B+」および「B-」と表示のある端子にそれぞれ接続します。

モータを接続する前に、どのケーブルがどのコイルのものか、確認してください。接続を間違えると、ドライバのチップやモータが損傷する恐れがあります！

MCST3601 はモータ接続用に二つの接続オプションを提供しています。どちらか片方のオプションだけを同時に使用してください！

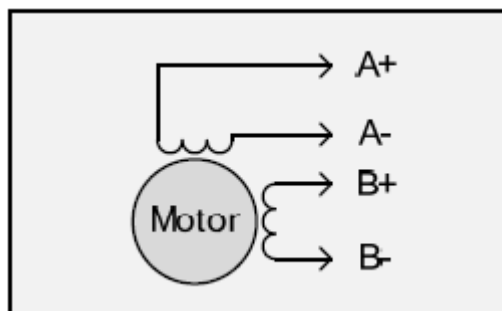
モータ接続オプション 1 (ネジ止め端子の使用)





	ピン	ラベル	方向	内容
	1	モータ 位相 A +	出力	モータドライバ出力、コイル A
	2	モータ 位相 A -	出力	モータドライバ出力、コイル A
	3	モータ 位相 B +	出力	モータドライバ出力、コイル B
	6	モータ 位相 B -	出力	モータドライバ出力、コイル B

図 3.3 モータ接続



3.1.1.3 電源

電源を電源端子に接続します（図 3.1 参照）。ただし、作業は電源オフの状態を開始してください。

極性に注意してください。極性が間違っていると、ボードが破損する恐れがあります。

DC+36V の最大電源供給を超えないようにしてください！



3.1.2 TMCL-IDE ソフトウェア開発環境の始動

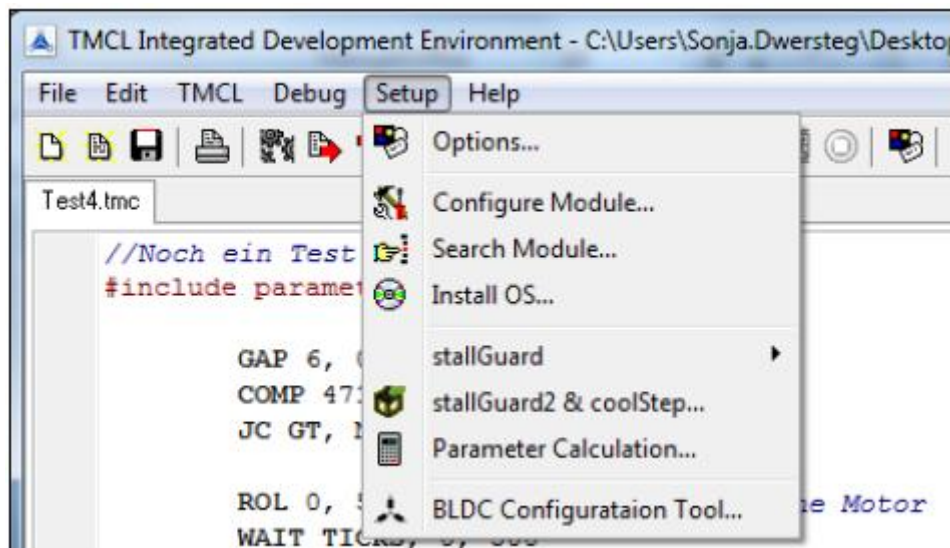
TMCL-IDE は www.trinamic.com からダウンロードできます。

TMCL-IDE のインストール:

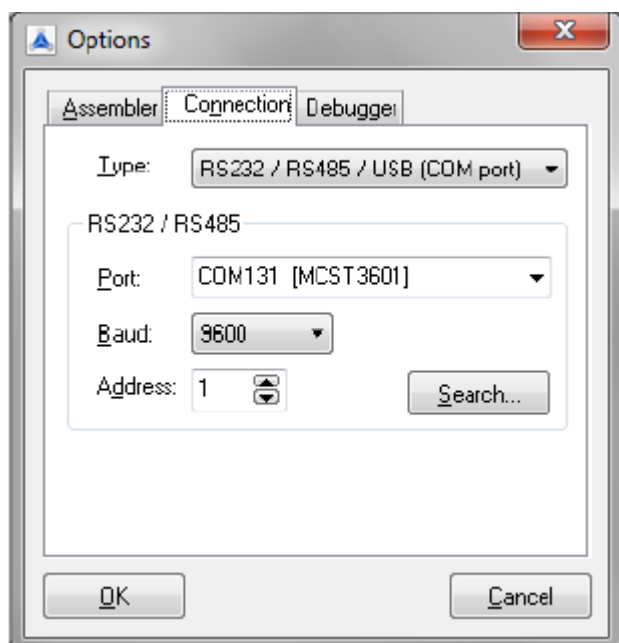
使用する COM ポートが他のプログラムによりブロックされていないことを確認してください。

TMCL.exe をクリックして TMCL-IDE を開きます。

[Setup]および[Options]を選択し、それから[Connection]タブを選択します。



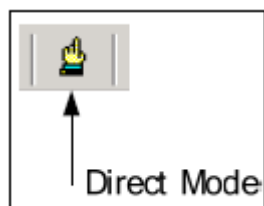
USB は、以下の設定で「COM port」と「Type」を選択します。[OK]をクリックします。



他のインターフェースへの接続に関する詳細な情報は、TMCL-IDE 取扱説明書を参照してください (www.TRINAMIC.com)。

3.1.3 TMCL™ダイレクトモードの使用

TMCL™ダイレクトモードを開始します。



Direct Mode	ダイレクトモード
-------------	----------

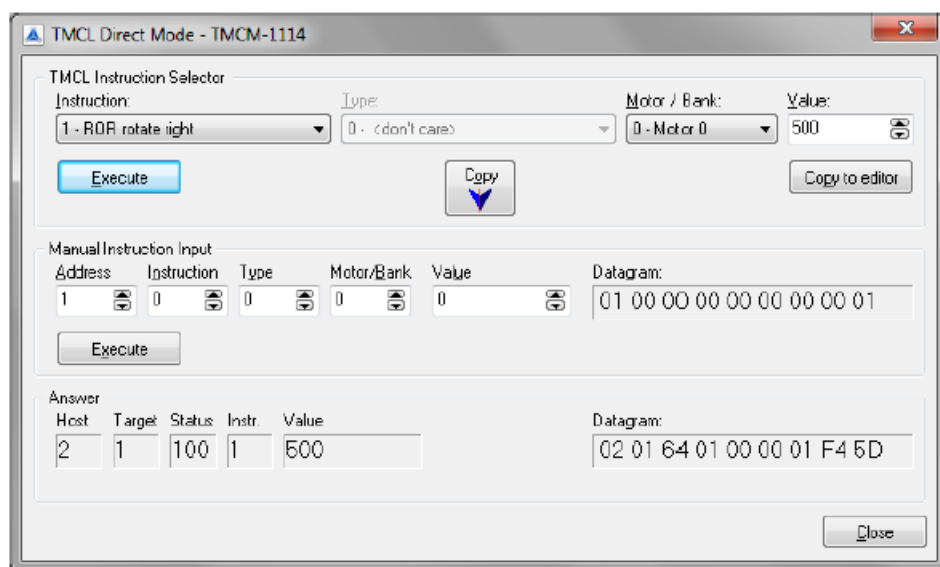
通信が確立されていれば、MCST3601 は自動的に検出されます (TMCL-IDE の最新バージョンを使用した場合)。

モジュールが検出されない場合は、ケーブル、インターフェース、電源、COM ポートおよびボーレートを確認してください。

「Instruction」、「Type」(必要なら)、「Motor」、および「Value」を選択してコマンドを発行し、[Execute]をクリックしてコマンドをモジュールに送信します。

注意

MCST3601 は最大 3 個のモータまで制御できますので、3 個のモータのモータ番号は 0、1、および 2 となります。モータを 1 個しか使用しない場合は、モータ番号は常に 0 になります。



例：

- ROR rotate right、motor 0、value 500 -> 「Execute」をクリック。第 1 モータが回転します。
- MST motor stop、motor 0 -> 「Execute」をクリック。第 1 モータが停止します。

TMCL Direct Mode ウィンドウの右上には「Copy to editor」ボタンがあります。ここをクリックすると、選択されたコマンドがコピーされ、独自の TMCL™プログラムを作成できます。コマンドは直ちにエディタ上に表示されます。

3.1.4 重要モータ設定

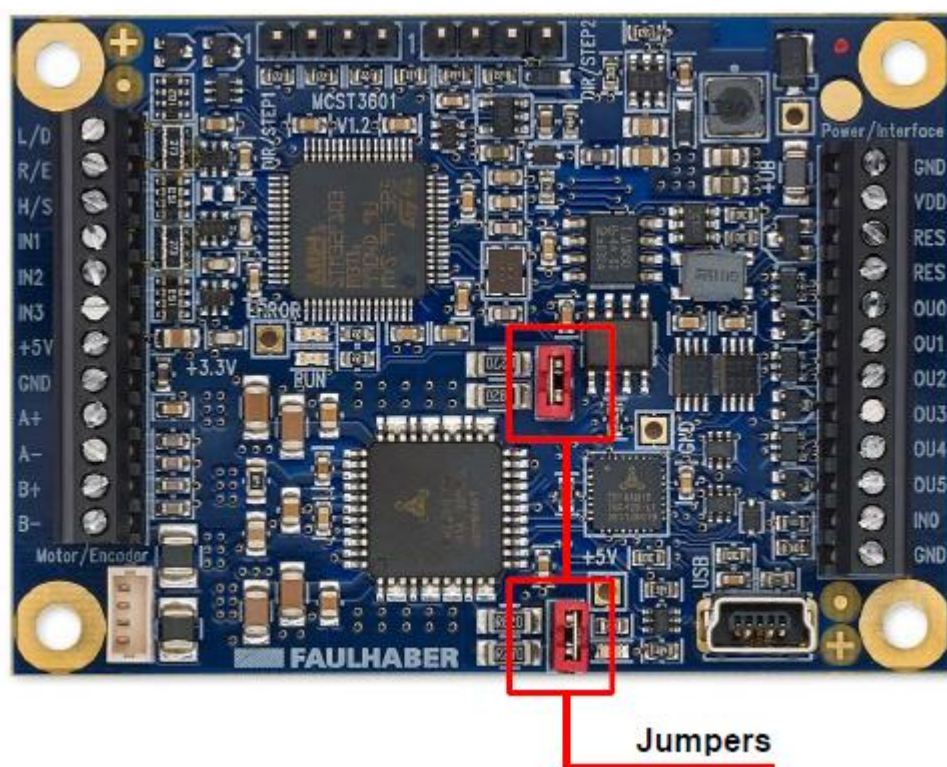
軸パラメータの中には、モジュールを取り付け後、最初に調整しなければいけないものがあります。速度の上限値（軸パラメータ 4）、加速度の上限値（軸パラメータ 5）、および電流の上限値（軸パラメータ 6）を設定してください。その後、待機電流（軸パラメータ 7）を設定し、パラメータ 140 でお使いのマイクロステップ分解能を選択してください。

これらを調整するために、SAP（軸パラメータ設定）コマンドを使用してください。SAP コマンドは 4.5.5 項で説明しています。TMCM-IDE ダイレクトモードを使用して、モジュールを簡単に構成することができます。

注意

最も重要なモータ設定は、最大モータ電流絶対値設定です。その理由は、この値が高すぎると、モータが損傷するためです。モータの電流範囲を設定するために、ソフトウェア上の設定に加え、ボード上の 2 個のジャンプスイッチも正しく設定してください。

ボード上の 2 個のジャンプスイッチによるモータ電流範囲の選択



ジャンプ スイッチ	内容
クローズ	最大モータ電流 1A RMS／1.5A ピーク（VSENSE=0 時（プログラマブル）） 最大モータ電流 0.57A RMS／0.8A ピーク（VSENSE=1 時（プログラマブル））
オープン	最大モータ電流 0.26A RMS／0.37A ピーク（VSENSE=0 時（プログラマブル）） 最大モータ電流 0.14A RMS／0.20A ピーク（VSENSE=1 時（プログラマブル））

モータ設定に重要な軸パラメータ

番号	軸パラメータ	内容	範囲 [単位]																																								
4	最大位置決め速度	物理的に可能な最大値を超えてはいけません。速度値が非常に低い (<50) または上限を超える場合は、パルス除数 (軸パラメータ 154) を調整してください。物理単位計算用 TMC 429 データシートを参照するか、TMCL-IDE 計算ツールを使用してください。	0... 2047 16MHz / 65536 · 2 ^{PD} マイクロステップ/秒																																								
5	最大加速度	加速度および減速度の限界です。パラメータを変更するためには、加速度係数および加速度除数の再計算が必要です。そのため、ランプ波除数 (軸パラメータ 153) を慎重に調整してください (調整単位=1)。物理単位計算用 TMC 429 データシートを参照するか、TMCL-IDE 計算ツールを使用してください。	0... 2047 ¹																																								
6	絶対最大電流 (CS/電流スケール)	<p>最大値は=255 です。これは、モジュールの最大電流の 100%を意味します。電流調整は 0~255 の範囲で行われ、32 段階で調整できます。</p> <table border="1"> <tr><td>0... 7</td><td>79... 87</td><td>160... 167</td><td>240... 247</td></tr> <tr><td>8... 15</td><td>88... 95</td><td>168... 175</td><td>248... 255</td></tr> <tr><td>16... 23</td><td>96... 103</td><td>176... 183</td><td></td></tr> <tr><td>24... 31</td><td>104... 111</td><td>184... 191</td><td></td></tr> <tr><td>32... 39</td><td>112... 119</td><td>192... 199</td><td></td></tr> <tr><td>40... 47</td><td>120... 127</td><td>200... 207</td><td></td></tr> <tr><td>48... 55</td><td>128... 135</td><td>208... 215</td><td></td></tr> <tr><td>56... 63</td><td>136... 143</td><td>216... 223</td><td></td></tr> <tr><td>64... 71</td><td>144... 151</td><td>224... 231</td><td></td></tr> <tr><td>72... 79</td><td>152... 159</td><td>232... 239</td><td></td></tr> </table> <p>これは、選択したモータに合わせて行う必要のある最も重要な設定です。その理由は、この値が高すぎると、モータが損傷するためです。</p>	0... 7	79... 87	160... 167	240... 247	8... 15	88... 95	168... 175	248... 255	16... 23	96... 103	176... 183		24... 31	104... 111	184... 191		32... 39	112... 119	192... 199		40... 47	120... 127	200... 207		48... 55	128... 135	208... 215		56... 63	136... 143	216... 223		64... 71	144... 151	224... 231		72... 79	152... 159	232... 239		<p>0... 255</p> <p>ジャンパススイッチが設定され、Vsense=0 の場合 (パラメータ 179 参照) :</p> $I_{peak} = <値> \times 1.5A/255$ $I_{RMS} = <値> \times 1A/255$ <p>ジャンパススイッチが設定され、Vsense=1 の場合 (パラメータ 179 参照) :</p> $I_{peak} = <値> \times 0.8A/255$ $I_{RMS} = <値> \times 0.57A/255$ <p>ジャンパススイッチが設定されず、Vsense=0 の場合 (パラメータ 179 参照) :</p> $I_{peak} = <値> \times 0.37A/255$ $I_{RMS} = <値> \times 0.26A/255$ <p>ジャンパススイッチが設定されず、Vsense=1 の場合 (パラメータ 179 参照) :</p> $I_{peak} = <値> \times 0.20A/255$ $I_{RMS} = <値> \times 0.147A/255$
0... 7	79... 87	160... 167	240... 247																																								
8... 15	88... 95	168... 175	248... 255																																								
16... 23	96... 103	176... 183																																									
24... 31	104... 111	184... 191																																									
32... 39	112... 119	192... 199																																									
40... 47	120... 127	200... 207																																									
48... 55	128... 135	208... 215																																									
56... 63	136... 143	216... 223																																									
64... 71	144... 151	224... 231																																									
72... 79	152... 159	232... 239																																									
7	待機電流	<p>モータ停止 2 秒後の電流限界です。設定とモータ電流との間の変換は、軸パラメータ 6 と同様です。</p> <p>Vsense (軸パラメータ 179) の値とジャンパ設定は、軸パラメータ 6 とこのパラメータと同じですのでご注意ください。</p>	<p>0... 255</p> <p>パラメータ 6 と同じ変換</p>																																								

番号	軸パラメータ	内容	範囲 [単位]
140	マイクロステップ分解能	0 フルステップ	0... 8
		1 ハーフステップ	
		2 4 マイクロステップ	
		3 8 マイクロステップ	
		4 16 マイクロステップ	
		5 32 マイクロステップ	
		6 64 マイクロステップ	
		7 128 マイクロステップ	
		8 256 マイクロステップ	
179	Vsense	<p>センサ抵抗電圧に基づく電流スケール</p> <p>0 : フルスケールのセンサ抵抗電圧は、最大 1A RMS / 1.5A ピーク (ジャンパクローズ時) または最大 0.26A RMS / 0.37A ピーク (ジャンパオープン時) です。</p> <p>1 : フルスケールのセンサ抵抗電圧は、最大 0.57A RMS / 0.8A ピーク (ジャンパクローズ時) または最大 0.14A RMS / 0.24A ピーク (ジャンパオープン時) です。</p>	0/1

*1 加速度単位 : $16\text{MHz}^2/536870912 \cdot 2^{\text{パルス除数} + \text{ランプ波除数}}$ マイクロステップ/秒²

4 TMCL™および TMCL-IDE 17

MCST3601 は TMCL™ダイレクトモード（バイナリコマンド）およびスタンドアロン TMCL™プログラム実行に対応しています。MCST3601 に最大 2048 個の TMCL™指示を保存できます。

ダイレクトモード時およびほとんどの場合で、USB を介した TMCL™通信は厳格なマスタ／スレーブ関係に従います。これは、インターフェースバスマスタの役割を果たすホストコンピュータ（PC／PLC など）が MCST3601 にコマンドを送信することを意味します。それから、モジュールの TMCL™インタプリタはこのコマンドを解釈し、モーションコントローラの初期化を行い、指令されたコマンドにしたがって入力を読み込んだり、出力を書きだしたり、その他の必要な動作を行います。この手順が実行されると、モジュールは直ちに USB を介してバスマスタに回答を送り返します。それからようやくマスタは次のコマンドを送信します。通常は、モジュールは回答用に送信モードに切り替わり、バスを占有します。それ以外は受信モードのままです。モジュールは、コマンドをまず受信しない限り、インターフェースを介してデータを送信することはありません。このようにして、一つのバスに二つ以上のノードが接続されている場合に、バス上のいかなる衝突も回避できます。

Tynamic Motion Control Language [TMCL™]は構造化されたモーションコントロールコマンドのセットを提供します。各モーションコントロールコマンドはホストコンピュータから送信されるか、あるいはモジュールにおいてスタンドアロンで実行されるプログラムを形成するために TCMC モジュールの EEPROM に保存されます。この目的のため、モーションコントロールコマンドだけでなく、プログラム構造を制御するコマンド（条件付きジャンプ、比較、計算など）もあります。

全てのコマンドはバイナリ表現とニモニックを有します。バイナリ形式はダイレクトモードでホストからモジュールへコマンドを送信するために使用され、ニモニック形式は TMCL-IDE（IDE は統合開発環境（Integrated Development Environment）を意味します）を使用したスタンドアロン TMCL™アプリケーションの開発時にコマンドを簡単に使うために使用されます。

また、軸用およびグローバルパラメータ用構成変数のセットもあり、それによりモジュールのほぼ全ての機能を個々に構成できます。本書では TMCL™コマンドおよびその使用法の全てを詳細に説明しています。

4.1 バイナリコマンド形式

ホストからモジュールにコマンドが送信される際は、バイナリ形式を使用する必要があります。各コマンドは、1 バイトコマンドフィールド、1 バイト種類フィールド、1 バイトモータ／バンクフィールド、および 4 バイト値フィールドにより構成されます。つまり、バイナリ表現では、コマンドは常に 7 バイトを有しています。コマンドが USB インターフェースを介して送信される時には、先頭にアドレスバイト、末尾にチェックサムバイトを付ける必要があります。この場合、コマンドは 9 バイトで構成されます。

USB 用バイナリコマンド形式は以下のとおりです。

バイト	意味
1	モジュールアドレス
1	コマンド番号
1	種類番号
1	モータまたはバンク番号
4	値（最上位バイトが先頭！）
1	チェックサム

- チェックサムは、追加の 8 ビットを使用して、それ以外のバイトを合計することで計算されます。

チェックサム計算

上記のように、チェックサムは、追加の 8 ビットを使用して、それ以外のバイト（モジュールアドレスバイトを含む）を合計することで計算されます。以下に二つのチェックサム計算例を示します。

C ドライブで

```
unsigned char i, Checksum;
```

```
unsigned char Command[9];
```

```
//Set the "Command" array to the desired command
```

```
Checksum = Command[0];
```

```
for(i=1; i<8; i++)
```

```
Checksum+=Command[i];
```

```
Command[8]=Checksum; //insert checksum as last byte of the command
```

```
//Now, send it to the module
```

4.2 回答形式

コマンドがモジュールに送信される度に、モジュールは回答を送信します。

USB 用回答形式は以下のとおりです。

バイト	意味
1	回答アドレス
1	モジュールアドレス
1	状態（例：100 は「エラーなし」の意味）
1	コマンド番号
4	値（最上位バイトが先頭！）
1	チェックサム

- チェックサムも、追加の 8 ビットを使用して、それ以外のバイトを合計することで計算されます。
- 回答を受信していない状態では、次のコマンドを送信しないでください！

4.2.1 状態コード

回答には状態コードが含まれます。状態コードは下記の値のいずれか一つを持つことができます。

コード	意味
100	実行の成功、エラーなし
101	コマンドの TMCL™ プログラム EEPROM へのロード
1	不正チェックサム
2	無効コマンド
3	不正種類
4	無効値
5	構成 EEPROM のロック
6	コマンド使用不可

4.3 スタンドアロンアプリケーション

モジュールは、TMCL™アプリケーション保存用 EEPROM を備えています。TMCL-IDE を使用してスタンドアロン TMCL™アプリケーションを開発することができます。アプリケーションは EEPROM にロードし、モジュール上で実行できます。TMCL-IDE にはエディタおよび TMCL™アセンブラが含まれ、ここではコマンドはニモニック形式を用いて入力できます。コマンドはバイナリ表現に自動的にアセンブルされます。その後、本コードはモジュールにダウンロードされ、そこで実行されます。

4.3.1 簡易 TMCL™プログラムを用いたテスト

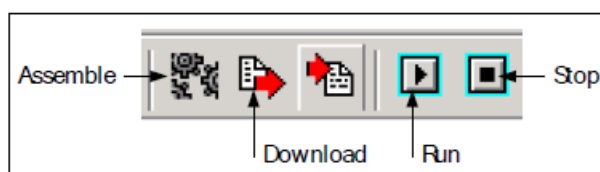
TMCL-IDE のファイル test2.tmc を開いてください。3 個のモータ用にテストプログラムが書き込まれています。モータが 1 個だけ接続されている場合は、モータ番号を 0 に変更してください。

テストプログラムは以下のように表示されます。

```
//A simple example for using TMCL™ and TMCL-IDE

ROL 0, 500           //Rotate motor 0 with speed 500
WAIT TICKS, 0, 500
MST 0
ROR 0, 250           //Rotate motor 0 with 250
WAIT TICKS, 0, 500
MST 0

SAP 4, 0, 500        //Set max. Velocity
SAP 5, 0, 50         //Set max. Acceleration
Loop: MVP ABS, 0, 10000 //Move to Position 10000
WAIT POS, 0, 0       //Wait until position reached
MVP ABS, 0, -10000   //Move to Position -10000
WAIT POS, 0, 0       //Wait until position reached
JA Loop              //Infinite Loop
```



- 1.アセンブルアイコンをクリックして、TMCL™コードを機械コードに変換します。
- 2.ダウンロードアイコンをクリックして、プログラムを MCST3601 モジュールにダウンロードします。
- 3.実行アイコンをクリックします。目的のプログラムが実行されます。
- 4.停止ボタンをクリックして、プログラムを停止します。

4.4 TMCL™コマンド概要

このセクションでは、TMCL™コマンドの概要を簡単に説明します。

4.4.1 TMCL™コマンド

コマンド	番号	パラメータ	内容
ROR	1	<モータ番号>, <速度>	指定した速度で右回転
ROL	2	<モータ番号>, <速度>	指定した速度で左回転
MST	3	<モータ番号>	モータ動作の停止
MVP	4	ABS REL COORD, <モータ番号>, <位置 オフセット>	指定位置に移動（絶対または相対）

コマンド	番号	パラメータ	内容
SAP	5	<パラメータ>, <モータ番号>, <値>	軸パラメータの設定（モーションコントロール固有の設定）
GAP	6	<パラメータ>, <モータ番号>	軸パラメータの取得（モーションコントロール固有の設定の読み出し）
STAP	7	<パラメータ>, <モータ番号>	軸パラメータを永続的に保存（非揮発性）
RSAP	8	<パラメータ>, <モータ番号>	軸パラメータの復元
SGP	9	<パラメータ>, <バンク番号>, <値>	グローバルパラメータの設定（通信設定または TMCL™ ユーザ変数などのモジュール固有の設定）
GGP	10	<パラメータ>, <バンク番号>	グローバルパラメータの取得（通信設定または TMCL™ ユーザ変数などのモジュール固有の設定の読み出し）
STGP	11	<パラメータ>, <バンク番号>	グローバルパラメータの保存（TMCL™ ユーザ変数のみ）
RSGP	12	<パラメータ>, <バンク番号>	グローバルパラメータの復元（TMCL™ ユーザ変数のみ）
RFS	13	START STOP STATUS、<モータ番号>	基準点サーチ
SIO	14	<ポート番号>, <バンク番号>, <値>	デジタル出力 j を指定値に設定
GIO	15	<ポート番号>, <バンク番号>	アナログ／デジタル入力値の取得
CALC	19	<演算>, <値>	アキュムレータおよび値の処理
COMP	20	<値>	アキュムレータ<->値の比較
JC	21	<条件>, <ジャンプアドレス>	条件付きジャンプ
JA	22	<ジャンプアドレス>	ジャンプ絶対値
CSUB	23	<サブルーチンアドレス>	サブルーチンの呼び出し
RSUB	24		サブルーチンからの復帰
EI	25	<割り込み番号>	割り込み有効化
DI	26	<割り込み番号>	割り込み無効化
WAIT	27	<条件>, <モータ番号>, <時間>	以降のプログラム実行の待機
STOP	28		プログラム実行の停止
SCO	30	<座標番号>, <モータ番号>, <位置>	座標設定
GCO	31	<座標番号>, <モータ番号>	座標の取得
CCO	32	<座標番号>, <モータ番号>	座標のキャプチャ
CALCX	33	<演算>	アキュムレータおよび X レジスタの処理
AAP	34	<パラメータ>, <モータ番号>	アキュムレータ→軸パラメータ
AGP	35	<パラメータ>, <バンク番号>	アキュムレータ→グローバルパラメータ
VECT	37	<割り込み番号>, <ラベル>	割り込みベクトルの設定
RETI	38		割り込みからの復帰
ACO	39	<座標番号>, <モータ番号>	アキュムレータ→座標

4.4.2 対象領域別コマンドリスト

4.4.2.1 モーションコマンド

これらのコマンドはモータの動作を制御します。これらは最も重要なコマンドであり、ダイレクトモードまたはスタンダロンモードで使用できます。

ニモニック	コマンド番号	意味
ROL	2	左回転
ROR	1	右回転
MVP	4	指定位置に移動
MST	3	モータ停止
RFS	13	基準点サーチ
SCO	30	座標の保存
CCO	32	座標のキャプチャ
GCO	31	座標の取得

4.4.2.2 パラメータコマンド

これらのコマンドは、軸パラメータやグローバルパラメータを設定、読み込み、および保存するために使用されます。軸パラメータは軸ごとに個々に設定でき、グローバルパラメータはモジュール自体の動作を制御します。これらのコマンドは、ダイレクトモードおよびスタンダロンモードで使用できます。

ニモニック	コマンド番号	意味
SAP	5	軸パラメータの設定
GAP	6	軸パラメータの取得
STAP	7	軸パラメータの EEPROM への保存
RSAP	8	軸パラメータの EEPROM からの復元
SGP	9	グローバルパラメータの設定
GGP	10	グローバルパラメータの取得
STGP	11	グローバルパラメータの EEPROM への保存
RSGP	12	グローバルパラメータの EEPROM からの復元

4.4.2.3 制御コマンド

これらのコマンドは、プログラムの流れ（ループ、条件、ジャンプなど）を制御するために使用されます。これらをダイレクトモードで使用することはありません。これらはスタンダロンモードのみでの使用を目的としています。

ニモニック	コマンド番号	意味
JA	22	常にジャンプ
JC	21	条件付きジャンプ
COMP	20	アキュムレータと定数値を比較
CSUB	23	サブルーチンの呼び出し
RSUB	24	サブルーチンからの復帰
WAIT	27	特定イベントの待機
STOP	28	TMCL™プログラムの終了

4.4.2.4 入出力ポートコマンド

これらのコマンドは外部入出力を制御し、ダイレクトモードおよびスタンドアロンモードで使用できます。

ニモニック	コマンド番号	意味
SIO	14	出力の設定
GIO	15	入力取得

4.4.2.5 計算コマンド

これらのコマンドは TMCL™アプリケーション内での計算における使用を目的としています。これらはダイレクトモードでも使用できますが、あまり意味がありません。

ニモニック	コマンド番号	意味
CALC	19	アキュムレータおよび定数値を使用した計算
CALCX	33	アキュムレータおよび X レジスタを使用した計算
AAP	34	アキュムレータの軸パラメータへのコピー
AGP	35	アキュムレータのグローバルパラメータへのコピー
ACO	39	アキュムレータの座標へのコピー

計算目的では、アキュムレータ（または **accu** または **A レジスタ**）および **X レジスタ**があります。値を読み込む全ての TMCL™コマンドは、TMCL™プログラム内で実行された場合（スタンドアロンモード）、結果をアキュムレータに保存します。X レジスタは、計算実行時に追加メモリとして使用できます。X レジスタはアキュムレータからロードできます。

値を読み込むコマンドがダイレクトモードで実行された場合、アキュムレータは影響を受けません。これは、TMCL™プログラムがモジュールで実行中に（スタンドアロンモード）、ホストはモジュールで実行中の TMCL™プログラムの流れに影響を及ぼすことなく、GAP や GGP と同様にまだモジュールに（例えばモータの実位置をクエリするために）コマンドを送信できることを意味します。

4.4.2.6 割り込みコマンド

お客様のご要望により、ARM ベースのモジュール用に割り込み処理が TMCL™ファームウェアに導入されています。

ニモニック	コマンド番号	意味
EI	25	割り込み有効化
DI	26	割り込み無効化
VECT	37	割り込みベクトルの設定
RETI	38	割り込みからの復帰

4.4.2.6.1 割り込み種類：

TMCL™では、タイマ割り込み、ストップスイッチ割り込み、位置達成割り込み、入力ピン変更割り込みなど、数多くの異なる割り込みがあります。各割り込みは独自の割り込みベクトルを有しています。

各割り込みベクトルはその割り込み番号により識別されます。割り込み番号の記号定数には、TMCL™に含まれるファイル `Interrupts.inc` を使用してください。

4.4.2.6.2 割り込み処理：

割り込みが発生し、割り込みが有効化され、その割り込み用に有効な割り込みベクトルが定義されると、通常の TMCL™プログラムの流れは中断され、割り込み処理ルーチンが呼び出されます。割り込み処理ルーチンが呼び出される前に、通常プログラムのコンテキストは自動的に保存されます（アキュムレータレジスタ、X レジスタ、TMCL™フラグなど）。

割り込みネスティングはありません。つまり、割り込み処理ルーチンの実行時には、他の割り込みは全て無効化されます。

割り込み処理ルーチンから復帰すると、直ちに通常プログラムのコンテキストが自動的に復元され、通常プログラムの実行が継続されます。

4.4.2.6.3 割り込みベクトル：

以下の表は全ての使用可能な割り込みベクトルを示しています。

割り込み番号	割り込み種類
0	タイマ 0
1	タイマ 1
2	タイマ 2
3	目標位置到達 0
4	目標位置到達 1
5	目標位置到達 2
15	stallGuard™軸 0
21	偏差軸 0
27	左停止スイッチ 0
28	右停止スイッチ 0
29	左停止スイッチ 1
30	右停止スイッチ 1
31	左停止スイッチ 2
32	右停止スイッチ 2
39	入力変更 0
40	入力変更 1
41	入力変更 2
42	入力変更 3
255	グローバル割り込み

4.4.2.6.4 割り込みの追加構成

割り込みには追加構成が必要なものがあります（タイマ割り込みの時間間隔など）。これはパラメータバンク 3（SGP <種類>, 3, <値>）を有する SGP コマンドを使用して構成できます。詳細については、SGP コマンド (4.5.9 項) を参照してください。

4.4.2.6.5 TMCL™での割り込みの使用

割り込みを使用するには、以下の作業を行う必要があります。

- VECT コマンドを使用して割り込み処理ルーチンを定義する。
- 必要に応じ、SGP <種類>, 3, <値> コマンドを使用して割り込みを構成する。
- EI <割り込み> コマンドを使用して割り込みを有効化する。
- EI 255 コマンドを使用して割り込みをグローバルに有効化する。
- 割り込み処理ルーチンは必ず RETI コマンドで終わる必要があります。

下記の例はタイマ割り込みの使用を示しています。

```
VECT 0, Timer0Irq //define the interrupt vector
SGP 0, 3, 1000 //configure the interrupt: set its period to 1000ms
EI 0 //enable this interrupt
EI 255 //globally switch on interrupt processing

//Main program: toggles output 3, using a WAIT command for the delay
Loop:
SIO 3, 2, 1
WAIT TICKS, 0, 50
SIO 3, 2, 0
WAIT TICKS, 0, 50
JA Loop
```



```
//Here is the interrupt handling routine
```

```
Timer0Irq:
```

```
GIO 0, 2 //check if OUT0 is high
```

```
JC NZ, Out0Off //jump if not
```

```
SIO 0, 2, 1 //switch OUT0 high
```

```
RETI //end of interrupt
```

```
Out0Off:
```

```
SIO 0, 2, 0 //switch OUT0 low
```

```
RETI //end of interrupt
```

上記の例では、割り込み番号は直接使用されます。プログラムをより読みやすくするには、提供されているインクルードファイル **Interrupts.inc** を使用してください。このファイルは、全割り込みコマンドで利用できる全ての割り込み番号用の記号定数を定義します。上記プログラムの冒頭は以下のようになります。

```
#include Interrupts.inc
```

```
VECT TI_TIMER0, Timer0Irq
```

```
SGP TI_TIMER0, 3, 1000
```

```
EI TI_TIMER0
```

```
EI TI_GLOBAL
```

他のサンプルプログラムも参照してください。

4.5 コマンド

以下のページでモジュール固有のコマンドを詳細に説明しています。これらのコマンドはコマンド番号の順に列記されています。

4.5.1 ROR（右回転）

モータは指定の速度で右に回転するように命令されます（位置カウンタ増加）。

内部機能：まず、速度モードが選択されます。次に、速度値が軸パラメータ#2 に転送されます（目標速度）。

本モジュールは TMC429 ステップモータコントローラおよび TMC262 パワードライバをベースにしています。これにより、速度を 0～2047 の範囲で選択できます。

関連コマンド：ROL、MST、SAP、GAP

ニモニック：ROR <モータ番号>, <速度>

バイナリ表現：

命令番号	種類	モータ／バンク	値
1	無関係	<モータ番号> 0... 2	<速度> 0... 2047

ダイレクトモードでの回答：

状態	値
100～OK	無関係

例：

モータ 0 を速度 350 で右に回転する場合

ニモニック：ROR 0, 350

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$01	\$00	\$00	\$00	\$00	\$01	\$5e

4.5.2 ROL(左回転)

このコマンドにより、モータは指定の速度で回転するように命令されます（ROR とは反対方向、位置カウンタ減少）。

内部機能：まず、速度モードが選択されます。次に、速度値が軸パラメータ#2 に転送されます（目標速度）。

本モジュールは TMC429 ステップモータコントローラおよび TMC262 パワードライバをベースにしています。これにより、速度を 0～2047 の範囲で選択できます。

関連コマンド：ROR、MST、SAP、GAP

ニモニック：ROL <モータ番号>, <速度>

バイナリ表現：

命令番号	種類	モータ／バンク	値
2	無関係	<モータ番号> 0... 2	<速度> 0... 2047

ダイレクトモードでの回答：

状態	値
100～OK	無関係

例：

モータ 0 を速度 1200 で左に回転する場合

ニモニック：ROL 0, 1200

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値 (16 進数)	\$01	\$02	\$00	\$00	\$00	\$00	\$04	\$b0

4.5.3 MST（モータ停止）

モータは停止するように命令されます。

内部機能：軸パラメータ目標速度が 0 に設定されます。

関連コマンド：ROL、ROR、SAP、GAP

ニモニック：MST <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
3	無関係	<モータ番号> 0... 2	無関係

ダイレクトモードでの回答：

状態	値
100～OK	無関係

例：

モータ 0 を停止

ニモニック：MST 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16 進数）	\$01	\$03	\$00	\$00	\$00	\$00	\$00	\$00

4.5.4 MVP（指定位置へ移動）

モータは、特定の相対または絶対位置もしくは事前にプログラムされた座標へ移動するように命令されます。本コマンドは、ユニットにプログラムされた加速度／減速度ランプ波および位置決め速度を使用します。本コマンドは非ブロックです。つまり、コマンドの解釈およびモーションコントローラの初期化後、直ちに回答が送信されます。これ以降のコマンドは、モータが終了位置に到達するのを待つことなく実行されます。最大速度および加速度は、軸パラメータ#4 および#5 により定義されています。

MVP コマンドの範囲は符号付き 32 ビットです (-2,147,483,648～ +2,147,483,647)。位置決めは、MST、ROL、または ROR コマンドを使用して中断できます。

注意：

- 実位置と新しい設定値との間には、2,147,483,647 ($2^{31}-1$) マイクロステップ以上開いてはいけませんのでご注意ください。これ以上開いた場合、モータは短い距離を選択するために反対方向に回ります。

2 種類の運転が使用できます：

- -2,147,483,648～+2,147,483,647 ($-2^{31}\sim 2^{31}-1$) の範囲での絶対位置への移動
- オフセットを用いた実位置への相対移動の開始。この場合、新しい結果の位置値も上記の範囲を超えてはいけません。

内部機能：新しい位置値は軸パラメータ#0 目標位置へ転送されます。

関連コマンド：SAP、GAP、SCO、CCO、GCO、MST

ニモニック：MVP <ABS|REL|COORD>, <モータ番号>, <位置|オフセット|座標番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
4	0 ABS - 絶対	<モータ番号> 0... 2	<位置>
	1 REL - 相対		<オフセット>
	2 COORD - 座標		<座標番号> 0... 20

ダイレクトモードでの回答：

状態	値
100～OK	無関係

例：

モータ 0 を絶対位置 90000 へ移動

ニモニック：MVP ABS, 0, 9000

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値 (16 進数)	\$01	\$04	\$00	\$00	\$00	\$01	\$5f	\$90

例：

モータ 0 を現在位置から 1000 ステップ後方に移動（相対的に-1000 移動）

ニモニック：MVP REL, 0, -1000

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ パ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16 進数）	\$01	\$04	\$01	\$00	\$ff	\$ff	\$fc	\$18

例：

モータ 0 を前回保存した座標#8 へ移動

ニモニック：MVP COORD, 0, 8

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ パ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16 進数）	\$01	\$04	\$02	\$00	\$00	\$00	\$00	\$08

座標へ移動させる場合は、SCO、CCO、または ACO コマンドを利用して座標を事前に適切に設定しておく必要があります。

4.5.5 SAP（軸パラメータの設定）

モジュールのほとんどのモーションコントロールパラメータは **SAP** コマンドを使用して指定できます。設定は **SRAM** に保存され、そのため揮発性です。これは、電源切断後に情報が失われることを意味します。設定を永続的に保存するには、**STAP**（軸パラメータの保存）コマンドを使用してください。

内部機能：パラメータ形式は冒頭の **0**（または負の値用の **0**）を無視して変換されます。

パラメータは適切な機器の正しい位置に転送されます。

関連コマンド：GAP、STAP、RSAP、AAP

ニモニック：SAP <パラメータ番号>, <モータ番号>, <値>

バイナリ表現：

命令番号	種類	モータ／バンク	値
5	<パラメータ番号>	<モータ番号> 0... 2	<値>

ダイレクトモードでの回答：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータや値を表に列記していますので、第 5 章を参照してください。

例：

モータの最大絶対電流を 200mA に設定する。

電流単位 $I_{RMS} = \text{<値>} \times 1A/255$ *)から、200mA の設定は<値>に 51 を持ちます（電流設定値の範囲：0～255）。電流の設定値は、この特別 SAP コマンドを使用する前に計算しておく必要があります。

ニモニック：SAP 6, 0, 47

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$05	\$06	\$00	\$00	\$00	\$00	\$2f

*) モータ電流がジャンパで選択できるため、その他の単位も使用可能です。電流単位の詳細については第 5 章を、ジャンパの使用についてはハードウェア説明書を参照してください。

4.5.6 GAP（軸パラメータの取得）

MCST3601 のほとんどのパラメータは、軸ごとに個別に調整できます。このパラメータを使用して、パラメータを読み出せます。スタンドアロンモードでは、要求された値は以降の処理（条件付きジャンプなど）の目的のために、アキュムレータレジスタに送信されます。ダイレクトモードでは、読み込まれた値は（アキュムレータに影響を及ぼすことなく）回答の値フィールドにのみ出力されます。

内部機能：パラメータは適切な機器の正しい位置から読み出されます。パラメータ形式は冒頭に 0（または負の値用の 0）を追加して変換されます。

関連コマンド：SAP、STAP、AAP、RSAP

ニモニック：GAP <パラメータ番号>, <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
6	<パラメータ番号>	<モータ番号> 0... 2	無関係

ダイレクトモードでの回答：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータや値を表に列記していますので、第 5 章を参照してください。

例：

モータの最大電流の取得

ニモニック：GAP 6, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値（16 進数）	\$01	\$06	\$06	\$00	\$00	\$00	\$00	\$00

回答：

バイトインデックス	0	1	2	3	4	5	6	7
機能	ホストアドレス	目標アドレス	状態	命令	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値（16 進数）	\$02	\$01	\$64	\$06	\$00	\$00	\$02	\$80

⇒ 状態＝エラーなし、値＝128

4.5.7 STAP（軸パラメータの保存）

軸パラメータ設定コマンド（SAP）を使用して以前に設定された軸パラメータは永続的に保存されます。パラメータのほとんどは電源投入後に自動的に復元されます。

内部機能：SRAM に保存された軸パラメータの値は EEPROM に転送され、次回電源が投入された際に EEPROM からロードされます。

関連コマンド：SAP、RSAP、GAP、AAP

ニモニック：STAP <パラメータ番号>, <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
7	<パラメータ番号>	<モータ番号> 0... 2	無関係*

* この機能の値オペランドは効果がありません。その代わりに、現在使用されている値（SAP により選択された値など）が保存されます。

ダイレクトモードでの回答：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータや値を表に列記していますので、第 5 章を参照してください。

例：

モータの最大速度の保存

ニモニック：STAP 4, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値（16 進数）	\$01	\$07	\$04	\$00	\$00	\$00	\$00	\$00

注記：構成 EEPROM がロックされている場合は（7.1 参照）、STAP コマンドは効果がありません。ダイレクトモードでは、この場合はエラーコード 5（構成 EEPROM のロック、セクション 0 も参照）が返されます。

4.5.8 RSAP（軸パラメータの復元）

全ての構成関連の軸パラメータ用に、非揮発性メモリ内の位置が用意されています。デフォルトでは、パラメータのほとんどは電源投入後に自動的に復元されます。以前に変更された一つのパラメータは、この命令を使用してもリセットできます。

内部機能：指定されたパラメータは構成 EEPROM メモリからその RAM 位置へコピーされます。

関連コマンド：SAP、STAP、GAP、AAP

ニモニック：RSAP <パラメータ番号>, <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
8	<パラメータ番号>	<モータ番号> 0... 2	無関係

ダイレクトモードでの回答構成：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータや値を表に列記していますので、第 5 章を参照してください。

例：

モータの最大電流を復元する場合

ニモニック：RSAP 6, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16 進数）	\$01	\$08	\$06	\$00	\$00	\$00	\$00	\$00

4.5.9 SGP（グローバルパラメータの設定）

モーションコントロールに直接関係のないモジュール固有パラメータはほとんどが指定でき、TMCL™ユーザ変数は変更できます。グローバルパラメータは、ホストインターフェース、周辺機器、またはその他のアプリケーション固有の変数と関連があります。これらのパラメータの異なるグループはバンクとして体系化され、将来の製品用により多い総数を可能にしています。現在は、グローバルパラメータ用にバンク 0 とバンク 1 が使用されています。バンク 2 はユーザ変数に使用され、バンク 3 は割り込み構成に使用されています。

モジュール設定は全て自動的に非揮発性メモリ（プロセッサの内部 EEPROM）に保存されます。TMCL™ユーザ変数は EEPROM に自動的に保存されませんが、STGP コマンドを使用することで保存できます。

内部機能：パラメータ形式は冒頭の 0（または負の値用の 0）を無視して変換されます。
 パラメータは適切な（オンボード）機器の正しい位置に転送されます。

関連コマンド：GGP、STGP、RSGP、AGP

ニモニック：SGP <パラメータ番号>, <バンク番号>, <値>

バイナリ表現：

命令番号	種類	モータ／バンク	値
9	<パラメータ番号>	<バンク番号>	<値>

ダイレクトモードでの回答：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータやバンク番号を表に列記していますので、第0章を参照してください。

例：

目標機器のシリアルアドレスを 3 に設定する場合

ニモニック：SGP 66, 0, 3

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ド レ ス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16 進数）	\$01	\$09	\$42	\$00	\$00	\$00	\$00	\$03

4.5.10 GGP（グローバルパラメータの取得）

グローバルパラメータは全てこの機能を使用して読み出されます。グローバルパラメータは、ホストインターフェース、周辺機器、またはアプリケーション固有の変数に関連があります。これらのパラメータの異なるグループはバンクとして体系化され、将来の製品用により多い総数を可能にしています。現在は、グローバルパラメータ用にバンク 0 とバンク 1 が使用されています。バンク 2 はユーザ変数に使用され、バンク 3 は割り込み構成に使用されています。

内部機能：パラメータは適切な機器の正しい位置から読み出されます。パラメータ形式は冒頭に 0（または負の値用の 0）を追加して変換されます。

関連コマンド：SGP、STGP、RSGP、AGP

ニモニック：GGP <パラメータ番号>, <バンク番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
10	<パラメータ番号>	<バンク番号>	無関係

ダイレクトモードでの回答：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータやバンク番号を表に列記していますので、第0章を参照してください。

例：

目標機器のシリアルアドレスを取得する場合

ニモニック：GGP 66, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値 (16 進数)	\$01	\$0a	\$42	\$00	\$00	\$00	\$00	\$00

回答：

バイトインデックス	0	1	2	3	4	5	6	7
機能	ホ ス ト ア ド レ ス	目 標 ア ドレス	状 態	命 令	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値 (16 進数)	\$02	\$01	\$64	\$0a	\$00	\$00	\$00	\$01

-> 状態=エラーなし、値=1

4.5.11 STGP（グローバルパラメータの保存）

本コマンドは TMCL™ ユーザ変数をモジュールの EEPROM に永続的に保存するために使用されます。グローバルパラメータの中には RAM メモリ内にあるものもあり、保存を行わないと電源切断後に修正は失われます。この命令により、保存が持続されます。パラメータのほとんどは電源投入後に自動的に復元されます。

内部機能：指定されたパラメータはその RAM 位置から構成 EEPROM へコピーされます。

関連コマンド：SGP、GGP、RSGP、AGP

ニモニック：STGP <パラメータ番号>, <バンク番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
11	<パラメータ番号>	<バンク番号>	無関係

ダイレクトモードでの回答：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータやバンク番号を表に列記していますので、第0章を参照してください。

例：

ユーザ変数#42 を保存する場合

ニモニック：STGP 42, 2

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$0b	\$2a	\$02	\$00	\$00	\$00	\$00

4.5.12 RSGP（グローバルパラメータの復元）

このコマンドを使用すると、TMCL™ユーザ変数の内容を EEPROM から復元できます。全ての構成関連の軸パラメータ用に、非揮発性メモリ内の位置が用意されています。デフォルトでは、パラメータのほとんどは電源投入後に自動的に復元されます。以前に変更された一つのパラメータは、この命令を使用してリセットできます。

内部機能：指定されたパラメータは構成 EEPROM メモリからその RAM 位置へコピーされます。

関連コマンド：SGP、STGP、GGP、AGP

ニモニック：RSGP <パラメータ番号>, <バンク番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
12	<パラメータ番号>	<バンク番号>	無関係

ダイレクトモードでの回答構成：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータやバンク番号を表に列記していますので、第0章を参照してください。

例：

ユーザ変数#42 を復元する場合

ニモニック：RSGP 42, 2

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16進数）	\$01	\$0c	\$2a	\$02	\$00	\$00	\$00	\$00

4.5.13 RFS（基準点サーチ）

MCST3601 は基準点サーチアルゴリズムを内蔵しており、使用できます。基準点サーチアルゴリズムは、切り替え点校正機能および三つの切り替えモードを提供します。基準点サーチの状態はクエリでき、サーチが終了したかどうか、確認できます（TMCL™プログラムでは、基準点サーチの終了を待つには WAIT コマンドを使用する方が賢明です）。軸パラメータ表で適切なパラメータを参照し、ご要望に沿った基準点サーチアルゴリズムを構成できます（第 5 章）。基準点サーチは開始および停止でき、基準点サーチの実際の状態を確認できます。

内部機能：基準点サーチはステートマシンとして実行されますので、実行中のインタラクションが可能です。

関連コマンド：WAIT

ニモニック：RFS <START|STOP|STATUS>, <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
13	0 開始 - 基準点サーチの開始 1 停止 - 基準点サーチの停止 2 状態 - 状態の取得	<モータ番号> 0... 2	下記参照

ダイレクトモードでの回答：

種類 0（開始）または 1（停止）使用時：

状態	値
100～OK	無関係

種類 2（状態）使用時：

状態	値	
100～OK	0	基準点サーチ有効
	他の値	基準点サーチ無効

例：

モータ 0 の基準点サーチを開始する場合

ニモニック：RFS START, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値（16 進数）	\$01	\$0d	\$00	\$00	\$00	\$00	\$00	\$00

このモジュールを使用して、基準点サーチの代わりにストール検出を使用することができます。詳細は 6.1 節を参照してください。

4.5.14 SIO（設定出力）

このコマンドは一般デジタル出力の状態を低（0）または高（1）に設定します。

内部機能：渡された値は指定された出力ラインへ転送されます。

関連コマンド：GIO、WAIT

ニモニック：SIO <ポート番号>, <バンク番号>, <値>

バイナリ表現：

命令番号	種類	モータ／バンク	値
14	<ポート番号>	<バンク番号> 2	<値> 0/1

回答構成：

状態	値
100～OK	無関係

例：

OUT1 を高（バンク 2、出力 1）に設定する場合 -> 出力 1 はアクティブに低にプルされます。

ニモニック：SIO 1, 2, 1

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$0e	\$01	\$02	\$00	\$00	\$00	\$01

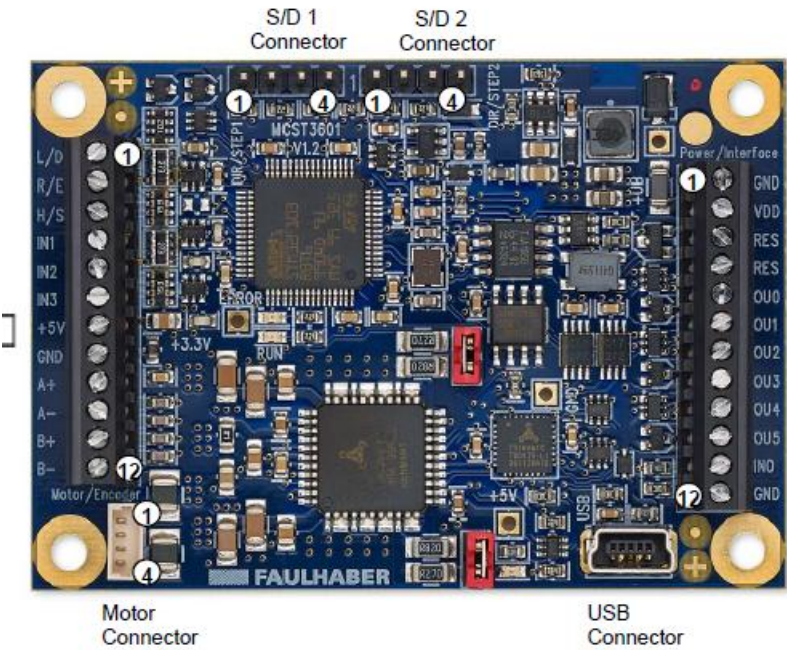


図 4.1：MCST3601 のプログラマブル汎用出力

バンク 2 は汎用デジタル出力（OD=オープンドレイン出力）の状態を低（低=0、出力ピンのフロート）または高（高=1、出力ピンの低へのプル）に設定するために使用されます。

ピン	入出力ポート	コマンド	範囲
5	OUT0 (OD 24V)	SIO 0, 2, <n>	1/0
6	OUT1 (OD 24V)	SIO 1, 2, <n>	1/0
7	OUT2 (OD 24V)	SIO 2, 2, <n>	1/0
8	OUT3 (OD 24V)	SIO 3, 2, <n>	1/0
9	OUT4 (OD 24V)	SIO 4, 2, <n>	1/0
10	OUT5 (OD 24V)	SIO 5, 2, <n>	1/0

SIO コマンドは、外部回路（モータのピン 7／エンコーダコネクタ）用+5V 電源出力をオン（値=1）またはオフ（値=0）するために使用できます。この+5V 出力は外部エンコーダへの電源供給にも使えます。デフォルト設定では、この出力はオンになっており、内部 DC／DC コンバータから+5V を供給します。

ピン	入出力ポート	コマンド	範囲
7	+5VOUT / 100mA	SIO 6, 2, <n>	1/0

4.5.15 GIO（入力／出力の取得）

このコマンドにより、モジュールの使用可能な汎用入力の状態が読み出せます。本機能はデジタルまたはアナログの入力ポートを読み出します。デジタルラインは0および1を読み出し、ADCチャンネルはその12ビットの結果を0～4095の範囲で送信します。

スタンドアロンモードでは、要求された値は以降の処理（条件付きジャンプなど）の目的のために、アキュムレータ（accu）にコピーされます。

ダイレクトモードでは、値はアキュムレータに影響を及ぼすことなく回答の値フィールドにのみ出力されます。デジタル出力ラインの実際の状態も読み出せます。

内部機能：指定されたラインが読み出されます。

関連コマンド：SIO、WAIT

ニモニック：GIO <ポート番号>, <バンク番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
15	<ポート番号>	<バンク番号>	無関係

ダイレクトモードでの回答：

状態	値
100～OK	<ポートの状態>

例：

IN0のアナログ値を取得する場合

ニモニック：GIO 0, 1

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$0f	\$00	\$01	\$00	\$00	\$00	\$00

回答：

バイトインデックス	0	1	2	3	4	5	6	7
機能	ホストアドレス	目標アドレス	状態	命令	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$02	\$01	\$64	\$0f	\$00	\$00	\$01	\$2e

状態＝エラーなし、値＝46

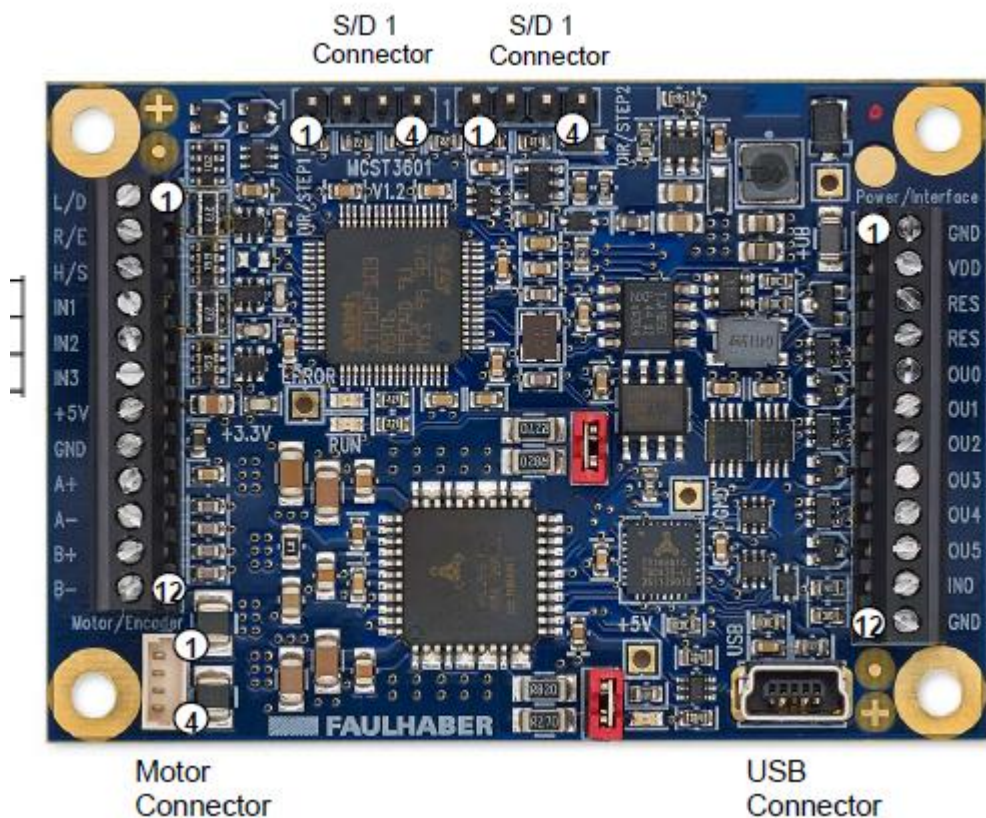


図 4.2 : MCST3601 のプログラマブル汎用入力および出力

4.5.15.1 入出力バンク 0 – デジタル入力

入力 IN0～IN3 の電圧はデジタル値として読み込めます。

ピン	入出力ポート	コマンド	範囲
11	IN0	GIO 0, 0	0/1
4	IN1	GIO 1, 0	0/1
5	IN2	GIO 2, 0	0/1
6	IN3	GIO 3, 0	0/1

4.5.15.2 入出力バンク 1 – アナログ入力

入力 IN0 の電圧はバンク 1 を介してアナログ値として読み込めます。

ピン	入出力ポート	コマンド	範囲／単位
11	IN0	GIO 0, 1	0～4095 の範囲で値を読み込みます。入力電圧の場合はおよそ DC0～10.3V です。
2	電圧	GIO 8, 1	供給電圧を×100mV で読み込みます。例えば、240 という値は DC24.0V の供給電圧を示します。

4.5.15.3 入出力バンク 2 – デジタル出力の状態

オープンドレイン出力 OUT0～OUT5（SIO コマンドにより設定）の状態はバンク 2 を使用して読み込めます。

ピン	入出力ポート	コマンド	範囲
5	OUT0	GIO 0, 2	1/0
6	OUT1	GIO 1, 2	1/0
7	OUT2	GIO 2, 2	1/0
8	OUT3	GIO 3, 2	1/0
9	OUT4	GIO 4, 2	1/0
10	OUT5	GIO 5, 2	1/0

4.5.16 CALC（計算）

GAP（軸パラメータ取得）などの機能により以前に読み込まれたアキュムレータ変数内の値は、この命令により修正できます。九つの異なる演算機能を選択できます。定数オペランド値を一つ指定する必要があります。結果は、比較やデータ転送といったこれ以降の処理用に、アキュムレータに書き込まれます。

関連コマンド：CALCX、COMP、JC、AAP、AGP、GAP、GGP、GIO

ニモニック：CALC <計算>, <オペランド>

バイナリ表現：

命令番号	種類<計算>	モータ／バンク	値
19	0 ADD - accu に加算 1 SUB - accu から減算 2 MUL - accu を乗算 3 DIV - accu を除算 4 MOD - モジュロ除算 5 AND - accu の論理積 6 OR - accu の論理和 7 XOR - accu の排他的論理和 8 NOT - accu の論理反転 9 LOAD - accu へのオペランドのロード	無関係	<オペランド>

例：

accu を-5000 で乗算する場合

ニモニック：CALC MUL, -5000

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値 (16進数)	\$01	\$13	\$02	\$00	\$FF	\$FF	\$EC	\$78

回答：

バイトインデックス	0	1	2	3	4	5	6	7
機能	ホストアドレス	目標アドレス	状態	命令	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値 (16進数)	\$02	\$01	\$64	\$13	\$ff	\$ff	\$ec	\$78

状態＝エラーなし、値＝-5000

4.5.17 COMP（比較）

指定された数値はアキュムレータレジスタ内の数値と比較されます。比較の結果は、例えば条件付きジャンプ(JC)命令などにより使用されます。このコマンドはスタンドアロン運転での使用のみを目的としています。

ホストアドレスと回答は、プログラムがダウンロード実行中に、命令を TMCL™ プログラムメモリに送るためのみに使用されます。

内部機能: 指定された値は、先行する取得または計算命令の値を有する内部アキュムレータと比較されます (GAP / GGP / GIO / CALC / CALCX 参照)。内部演算状態フラグは比較結果にしたがって設定されます。

関連コマンド : JC（条件付きジャンプ）、GAP、GGP、GIO、CALC、CALCX

ニモニック : COMP <比較値>

バイナリ表現 :

命令番号	種類	モータ／バンク	値
20	無関係	無関係	<比較値>

例 :

モータ位置が 1000 かそれより大きいとき、ラベルにより指定されるアドレスにジャンプする場合

GAP 1, 2, 0 //get axis parameter, type: no. 1 (actual position), motor:0, value:0 don't care

COMP 1000 //compare actual value to 1000

JC GE, Label //jump, type:5 greater/equal, the label must be defined somewhere else in the program

COMP 1000 コマンドのバイナリ形式 :

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値 (16 進数)	\$01	\$14	\$00	\$00	\$00	\$00	\$03	\$e8

4.5.18 JC (条件付きジャンプ)

JC 命令により、指定された条件が満たされた場合、TMCL™プログラムメモリ内の固定アドレスへの条件付きジャンプが実行されます。この条件とは、先行する比較の結果のことです。用例は **COMP** 命令を参照してください。この機能はスタンドアロン運転のみで使用できます。

ホストアドレスと回答は、プログラムがダウンロード実行中に、命令を TMCL™プログラムメモリに送るためのみに使用されます。

内部機能：TMCL™プログラムカウンタは、演算状態フラグが適切な状態であれば、渡された値に設定されます。

関連コマンド：JA、COMP、WAIT、CLE

ニモニク：JC <条件>, <ラベル>

ここで、<条件>=ZE|NZ|EQ|NE|GT|GE|LT|LE|ETO|EAL|EDV|EPO

バイナリ表現：

命令番号	種類	モータ／バンク	値
21	0 ZE - ゼロ 1 NZ - ゼロ以外 2 EQ - 等しい 3 NE - 不等 4 GT - より大きい 5 GE - 以上 6 LT - より小さい 7 LE - 以下 8 ETO - タイムアウトエラー 9 EAL - 外部アラーム 10 EDV - 偏差エラー 11 EPO - 位置エラー	無関係	<ジャンプアドレス>

例：

モータ位置が 1000 かそれより大きいとき、ラベルにより指定されるアドレスにジャンプする場合

GAP 1, 0, 0 //get axis parameter, type: no. 1 (actual position), motor:0, value:0 don't care

COMP 1000 //compare actual value to 1000

JC GE, Label //jump, type:5 greater/equal

...

...

Label:ROL 0, 1000

JC GE のバイナリ形式、ラベルがアドレス 10 の時のラベル

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値 (16進数)	\$01	\$15	\$05	\$00	\$00	\$00	\$00	\$0a

4.5.19 JA（常にジャンプ）

TMCL™プログラムメモリ内の固定アドレスへジャンプします。このコマンドはスタンドアロン運転での使用のみを目的としています。

ホストアドレスと回答は、プログラムがダウンロード実行中に、命令を TMCL™プログラムメモリに送るためのみに使用されます。

内部機能：TMCL™プログラムカウンタは渡された値に設定されます。

関連コマンド：JC、WAIT、CSUB

ニモニック：JA <ラベル>

バイナリ表現：

命令番号	種類	モータ／バンク	値
22	無関係	無関係	<ジャンプアドレス>

例：TMCL™内の無限ループ

Loop:MVP ABS, 0, 10000

WAIT POS, 0, 0

MVP ABS, 0, 0

WAIT POS, 0, 0

JA Loop //Jump to the label Loop

ラベルループがアドレス 20 にあるとした場合の JA ループのバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16進数）	\$01	\$16	\$00	\$00	\$00	\$00	\$00	\$14

4.5.20 CSUB（サブルーチン呼び出し）

本機能は **TMCL™** プログラムメモリ内のサブルーチンを呼び出します。このコマンドはスタンドアロン運転での使用のみを目的としています。

ホストアドレスと回答は、プログラムがダウンロード実行中に、命令を **TMCL™** プログラムメモリに送るためのみに使用されます。

内部機能： **TMCL™** プログラムカウンタ実値は内部スタックに保存され、その後渡された値により上書きされます。内部スタックのエントリ数は **8** に限定されています。これはサブルーチン呼び出しのネスティングも **8** に限定しています。スタックのスペースが残されていない場合は、本コマンドは無視されます。

関連コマンド： **RSUB**、**JA**

ニモニック： **CSUB** <ラベル>

バイナリ表現：

命令番号	種類	モータ／バンク	値
23	無関係	無関係	<サブルーチンアドレス>

例：サブルーチンの呼び出し

Loop:MVP ABS, 0, 10000

CSUB SubW //Save program counter and jump to label SubW

MVP ABS, 0, 0

JA Loop

SubW:WAIT POS, 0, 0

WAIT TICKS, 0, 50

RSUB //Continue with the command following the CSUB command

ラベル **SubW** がアドレス **100** にあるとした場合の **CSUB SubW** コマンドのバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$17	\$00	\$00	\$00	\$00	\$00	\$64

4.5.21 RSUB（サブルーチンから復帰）

本命令により、CSUB コマンド後に、サブルーチンからコマンドに復帰します。このコマンドはスタンドアロンモードでの使用のみを目的としています。

ホストアドレスと回答は、プログラムがダウンロード実行中に、命令を TMCL™ プログラムメモリに送るためのみに使用されます。このコマンドはダイレクトモードでは使用できません。

内部機能：TMCL™ プログラムカウンタはスタックの最後の値に設定されます。スタックが空だと、このコマンドは無視されます。

関連コマンド：CSUB

ニモニック：RSUB

バイナリ表現：

命令番号	種類	モータ／バンク	値
24	無関係	無関係	無関係

例：CSUB の例（4.5.20 項）をご覧ください。

RSUB のバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$18	\$00	\$00	\$00	\$00	\$00	\$00

4.5.22 WAIT (イベント発生待機)

この命令は、指定された条件が満たされるまで、TMCL™プログラムの実行を中断します。このコマンドはスタンドアロン運転での使用のみを目的としています。

ホストアドレスと回答は、プログラムがダウンロード実行中に、命令を TMCL™プログラムメモリに送るためのみに使用されます。このコマンドはダイレクトモードでは使用できません。

5 種類の使用可能な待機条件があります。

- **TICKS:**<時間カウント>パラメータにより指定されたタイマ時間に達するまで待機します。
- **POS:**<モータ>パラメータにより指定されたモータの目標位置に達するまで待機します。オプションのタイムアウト値 (0 はタイムアウトなし) は、<時間カウント>パラメータにより指定される必要があります。
- **REFSW:**<モータ>パラメータにより指定されたモータの基準点スイッチがトリガされるまで待機します。オプションのタイムアウト値 (0 はタイムアウトなし) は、<時間カウント>パラメータにより指定される必要があります。
- **LIMSW:**<モータ>パラメータにより指定されたモータのリミットスイッチがトリガされるまで待機します。オプションのタイムアウト値 (0 はタイムアウトなし) は、<時間カウント>パラメータにより指定される必要があります。
- **RFS:**<モータ>フィールドにより指定されたモータの基準点サーチが達成されるまで待機します。オプションのタイムアウト値 (0 はタイムアウトなし) は、<時間カウント>パラメータにより指定される必要があります。

タイムアウト限度に達すると、タイムアウトフラグ (ETO) が設定されます。それから、そのようなエラーを確認するため **JC ETO** コマンドを使用でき、または **CLE** コマンドを使用してエラーをクリアできます。

内部機能：指定した条件が満たされるまで、TMCL™プログラムカウンタは保持されます。

関連コマンド：JC、CLE

ニモニック：WAIT <条件>, <モータ番号>, <時間カウント>

バイナリ表現：

命令番号	種類<条件>	モータ／バンク	値
27	0 TICKS - タイマ時間カウント* ¹	無関係	<時間カウント数*>
	1 POS - 目標位置到達	<モータ番号> 0... 2	<タイムアウト用時間カウント数*>, 0=タイムアウトなし
	2 REFSW - 基準点スイッチ	<モータ番号> 0... 2	<タイムアウト用時間カウント数*>, 0=タイムアウトなし
	3 LIMSW - リミットスイッチ	<モータ番号> 0... 2	<タイムアウト用時間カウント数*>, 0=タイムアウトなし
	4 RFS - 基準点サーチ完了	<モータ番号> 0... 2	<タイムアウト用時間カウント数*>, 0=タイムアウトなし

* 時間カウント間隔=10 ミリ秒

例：

タイムアウトなしに、モータ 0 の目標到達を待機する場合

ニモニック：WAIT POS, 0, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7	8
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0	チ ェ ッ ク サ ム
値 (16 進 数)	\$01	\$1b	\$01	\$00	\$00	\$00	\$00	\$00	\$1d

4.5.23 STOP (TMCL™プログラム実行の停止)

本機能は TMCL™プログラムの実行を停止します。ホストアドレスと回答は、命令を TMCL™プログラムメモリに転送するためのみに使用されます。

スタンドアロン TMCL™プログラムは STOP コマンドで終了してください。これはダイレクトモードでは使用されません。

内部機能：TMCL™命令のフェッチが停止します。

関連コマンド：なし

ニモニク：STOP

バイナリ表現：

命令番号	種 類	モ ー タ / バ ン ク	値
28	無関係	無関係	無関係

例：

ニモニク：STOP

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値 (16 進数)	\$01	\$1c	\$00	\$00	\$00	\$00	\$00	\$00

4.5.24 SCO（座標設定）

最大 20 個の位置値（座標）が、MVP COORD コマンドでの使用用に軸ごとに保存できます。

このコマンドは座標を特定の値に設定します。グローバルパラメータ 84 によっては、座標は RAM のみに保存されるかまたは EEPROM にも保存され、起動時にコピーして戻されます（デフォルト設定では座標は RAM のみに保存されます）。

座標番号 0 は常に RAM のみに保存されていますのでご注意ください。

内部機能：渡された値は内部位置アレイに保存されます。

関連コマンド：GCO、CCO、MVP

ニモニック：SCO <座標番号>, <モータ番号>, <位置>

バイナリ表現：

命令番号	種類	モータ／バンク	値
30	<座標番号> 0... 20	<モータ番号> 0... 2	<位置> - 2^{23} ... $+2^{23}$

ダイレクトモードでの回答：

状態	値
100～OK	無関係

例：

モータの座標#1 を 1000 に設定する場合

ニモニック：SCO 1, 0, 1000

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$1e	\$01	\$00	\$00	\$00	\$03	\$e8

このコマンドの二つの特別な機能が導入され、それにより全座標もしくは一つの選択された座標を EEPROM にコピーできます。

以下の特別な SCO コマンド形式を使用することで、これらの機能にアクセスできます。

SCO 0, 255, 0 全座標（座標番号 0 以外）を RAM から EEPROM.へコピーします。

SCO <座標番号>, 255, 0 <座標番号>によって選択された座標を EEPROM へコピーします。座標番号は 1～20 の範囲の値でなければなりません。

4.5.25 GCO（座標取得）

このコマンドにより、以前に保存された座標を読み出すことができます。スタンダロンモードでは、要求された値は以降の処理（条件付きジャンプなど）の目的のために、アキュムレータレジスタにコピーされます。ダイレクトモードでは、値はアキュムレータに影響を及ぼすことなく回答の値フィールドにのみ出力されます。グローバルパラメータ **84** によっては、座標は **RAM** のみに保存されるかまたは **EEPROM** にも保存され、起動時にコピーして戻されます（デフォルト設定では座標は **RAM** のみに保存されます）。

座標番号 **0** は常に **RAM** のみに保存されていますのでご注意ください。

内部機能：目的の値は内部座標アレイから読み出され、アキュムレータレジスタにコピーされ、そしてダイレクトモードでは回答の値フィールドに戻されます。

関連コマンド：SCO、CCO、MVP

ニモニック：GCO <座標番号>, <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
31	<座標番号> 0... 20	<モータ番号> 0... 2	無関係

ダイレクトモードでの回答：

状態	値
100～OK	無関係

例：

座標 **1** のモータ値を取得する場合

ニモニック：GCO 1, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$1f	\$01	\$00	\$00	\$00	\$00	\$00

回答：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	目標アドレス	状態	命令	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$02	\$01	\$64	\$0a	\$00	\$00	\$00	\$00

⇒ 値：0

このコマンドの二つの特別な機能が導入され、それにより全座標もしくは一つの選択された座標を **EEPROM** から **RAM** にコピーできます。

以下の特別な **GCO** コマンド形式を使用することで、これらの機能にアクセスできます。

GCO 0, 255, 0 全座標（座標番号 **0** 以外）を **EEPROM** から **RAM** へコピーします。

GCO <座標番号>, 255, 0 <座標番号>によって選択された座標を **EEPROM** から **RAM** へコピーします。座標番号は **1～20** の範囲の値でなければなりません。

4.5.26 CCO（座標キャプチャ）

このコマンドにより、軸の実位置は選択された座標変数へコピーされます。グローバルパラメータ **84** によっては、座標は **RAM** のみに保存されるかまたは **EEPROM** にも保存され、起動時にコピーして戻されます（デフォルト設定では座標は **RAM** のみに保存されます）。**RAM** と **EEPROM** の間の座標のコピーについては、**SCO** コマンドおよび **GCO** コマンドを参照してください。

座標番号 **0** は常に **RAM** のみに保存されていますのでご注意ください。

内部機能：選択された位置値（**24** ビット）は、**20×3** バイト幅の座標アレイに書き込まれます。

関連コマンド：SCO、GCO、MVP

ニモニック：CCO <座標番号>, <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
32	<座標番号> 0... 20	<モータ番号> 0... 2	無関係

ダイレクトモードでの回答：

状態	値
100～OK	無関係

例：

軸 **0** の現在位置を座標 **3** に保存します。

ニモニック：CCO 3, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16 進数）	\$01	\$20	\$03	\$00	\$00	\$00	\$00	\$00

4.5.27 ACO（アキュムレータ→座標）

ACO コマンドにより、アキュムレータの実値が選択されたモータ座標にコピーされます。グローバルパラメータ 84 によっては、座標は RAM のみに保存されるかまたは EEPROM にも保存され、起動時にコピーして戻されます（デフォルト設定では座標は RAM のみに保存されます）。

座標番号 0 は常に RAM のみに保存されていることにもご留意ください。座標の保存の詳細は、SCO コマンドを参照してください。

内部機能：アキュムレータの実値は内部位置アレイに保存されます。

関連コマンド：GCO、CCO、MVP COORD、SCO

ニモニック：ACO <座標番号>, <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
39	<座標番号> 0... 20	<モータ番号> 0... 2	無関係

ダイレクトモードでの回答：

状態	値
100～OK	無関係

例：

アキュムレータの実値をモータ 0 の座標 1 にコピーする場合

ニモニック：ACO 1, 0

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目 標 ア ドレス	命 令 番 号	種 類	モ ー タ ／ バ ン ク	オ ペ ラ ン ド バ イト 3	オ ペ ラ ン ド バ イト 2	オ ペ ラ ン ド バ イト 1	オ ペ ラ ン ド バ イト 0
値（16 進数）	\$01	\$27	\$01	\$00	\$00	\$00	\$00	\$00

4.5.28 CALCX (Xレジスタを使用した計算)

この命令は **CALC** に非常に類似していますが、2 番目のオペランドは **X** レジスタからのものです。**X** レジスタは本命令の種類 **LOAD** または **SWAP** によりロードできます。結果は、比較やデータ転送といったこれ以降の処理用に、アキュムレータに書き戻されます。

関連コマンド : **CALC**、**COMP**、**JC**、**AAP**、**AGP**

ニモニック : **CALCX** <演算>

バイナリ表現 :

命令番号	種類<計算>	モータ／バンク	値
33	0 ADD - accu に X レジスタを加算 1 SUB - accu から X レジスタを減算 2 MUL - accu を X レジスタで乗算 3 DIV - accu を X レジスタで除算 4 MOD - accu を X レジスタでモジュロ除算 5 AND - accu と X レジスタの論理積 6 OR - accu と X レジスタの論理和 7 XOR - accu と X レジスタの排他的論理和 8 NOT - X レジスタの論理反転 9 LOAD - accu の X レジスタへのロード 10 SWAP - accu の X レジスタとのスワップ	無関係	無関係

例 :

accu を **X** レジスタで乗算する場合

ニモニック : **CALCX MUL**

バイナリ :

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値 (16 進数)	\$01	\$21	\$02	\$00	\$00	\$00	\$00	\$00

4.5.29 AAP（アキュムレータ→ 軸パラメータ）

このコマンドにより、アキュムレータレジスタの内容が指定された軸パラメータへ転送されます。実際の使用では、アキュムレータは先行する **GAP** 命令などでロードされている必要があります。アキュムレータは、**CALC** や **CALCX**（計算）命令などで修正されている場合があります。

関連コマンド：AGP、SAP、GAP、SGP、GGP、GIO、GCO、CALC、CALCX

ニモニック：AAP <パラメータ番号>, <モータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
34	<パラメータ番号>	<モータ番号> 0... 2	<無関係>

ダイレクトモードでの回答：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータや値を表に列記していますので、第 5 章を参照してください。

例：

モータをアナログ入力#0 に接続されているポテンショメータを使用して位置決めする場合：

Start:GIO 0,1 // get value of analogue input line 0

CALC MUL, 4 // multiply by 4

AAP 0,0 // transfer result to target position of motor 0

JA Start // jump back to start

AAP 0,0 コマンドのバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値（16 進数）	\$01	\$22	\$00	\$00	\$00	\$00	\$00	\$00

4.5.30 AGP（アキュムレータ→グローバルパラメータ）

このコマンドにより、アキュムレータレジスタの内容が指定されたグローバルパラメータへ転送されます。実際の使用では、アキュムレータは先行する **GAP** 命令などでロードされている必要があります。アキュムレータは、**CALC** や **CALCX**（計算）命令などで修正されている場合があります。バンク **0** のグローバルパラメータは **EEPROM** のみであり、そのためスタンドアロンアプリケーションにより自動的に修正されてはいけません。

関連コマンド：AAP、SGP、GGP、SAP、GAP、GIO

ニモニック：AGP <パラメータ番号>, <バンク番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
35	<パラメータ番号>	<バンク番号>	無関係

ダイレクトモードでの回答：

状態	値
100～OK	無関係

本コマンドとともに使用できるパラメータやバンク番号を表に列記していますので、第0章を参照してください。

例：

アキュムレータを **TMCL™** ユーザ変数 **#3** にコピーする場合

ニモニック：AGP 3, 2

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$23	\$03	\$02	\$00	\$00	\$00	\$00

4.5.31 CLE（エラーフラグのクリア）

このコマンドは内部エラーフラグをクリアします。これはスタンドアロンモードでの使用のみを目的としており、ダイレクトモードでは使用してはいけません。

以下のエラーフラグはこのコマンドによりクリアされます（<フラグ>パラメータにより決定）：

- ALL：全エラーフラグのクリア
- ETO：タイムアウトフラグのクリア
- EAL：外部アラームフラグのクリア
- EDV：偏差フラグのクリア
- EPO：位置エラーフラグのクリア

関連コマンド：JC

ニモニック：CLE <フラグ>

バイナリ表現：

命令番号	種類<フラグ>	モータ／バンク	値
36	0－(ALL) 全フラグ 1－(ETO) タイムアウトフラグ 2－(EAL) アラームフラグ 3－(EDV) 偏差フラグ 4－(EPO) 位置フラグ 5－(ESD) シャットダウンフラグ	無関係	無関係

例：

タイムアウトフラグをリセットする場合

ニモニック：CLE ETO

バイナリ：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$24	\$01	\$00	\$00	\$00	\$00	\$00

4.5.32 VECT（割り込みベクトルの設定）

VECT コマンドは割り込みベクトルを定義します。このコマンドには割り込み番号とラベルがパラメータとして必要です（JA、JC および CSUB コマンドと同様）。

このラベルは割り込み処理ルーチンのエントリポイントである必要があります。

関連コマンド：EI、DI、RETI

ニモニック：VECT <割り込み番号>, <ラベル>

バイナリ表現：

命令番号	種類	モータ／バンク	値
37	<割り込み番号>	無関係	<ラベル>

以下の表は使用可能な全ての割り込みベクトルを示しています。

割り込み番号	割り込み種類
0	タイマ 0
1	タイマ 1
2	タイマ 2
3	目標位置到達 0
4	目標位置到達 1
5	目標位置到達 2
15	stallGuard™ 軸 0
21	偏差軸 0
27	左停止スイッチ 0
28	右停止スイッチ 0
29	左停止スイッチ 1
30	右停止スイッチ 1
31	左停止スイッチ 2
32	右停止スイッチ 2
39	入力変更 0
40	入力変更 1
41	入力変更 2
42	入力変更 3
255	グローバル割り込み

例：目標位置 500 で割り込みベクトルを定義する場合

VECT 3, 500

VECT のバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値（16 進数）	\$01	\$25	\$03	\$00	\$00	\$00	\$01	\$F4

4.5.33 EI（割り込み有効化）

EI コマンドは割り込みを有効にします。このコマンドには割り込み番号がパラメータとして必要です。割り込み番号 255 は全体的に割り込みを有効化します。

関連コマンド：DI、VECT、RETI

ニモニック：EI <割り込み番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
25	<割り込み番号>	無関係	無関係

以下の表は使用可能な全ての割り込みベクトルを示しています。

割り込み番号	割り込み種類		割り込み番号	割り込み種類
0	タイマ 0		29	左停止スイッチ 1
1	タイマ 1		30	右停止スイッチ 1
2	タイマ 2		31	左停止スイッチ 2
3	目標位置到達 0		32	右停止スイッチ 2
4	目標位置到達 1		39	入力変更 0
5	目標位置到達 2		40	入力変更 1
15	stallGuard™ 軸 0		41	入力変更 2
21	偏差軸 0		42	入力変更 3
27	左停止スイッチ 0		255	グローバル割り込み
28	右停止スイッチ 0		29	左停止スイッチ 1

例：

全体的に割り込みを有効にする場合

EI, 255

EI のバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値 (16 進数)	\$01	\$19	\$FF	\$00	\$00	\$00	\$00	\$00

目標位置到達時に割り込みを有効にする場合

EI, 3

EI のバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値 (16 進数)	\$01	\$19	\$03	\$00	\$00	\$00	\$00	\$00

4.5.34 DI（割り込み無効化）

DI コマンドは割り込みを無効にします。このコマンドには割り込み番号がパラメータとして必要です。割り込み番号 255 は全体的に割り込みを無効化します。

関連コマンド：EI、VECT、RETI

ニモニック：DI <割り込み番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
26	<割り込み番号>	無関係	無関係

以下の表は使用可能な全ての割り込みベクトルを示しています。

割り込み番号	割り込み種類		割り込み番号	割り込み種類
0	タイマ 0		29	左停止スイッチ 1
1	タイマ 1		30	右停止スイッチ 1
2	タイマ 2		31	左停止スイッチ 2
3	目標位置到達 0		32	右停止スイッチ 2
4	目標位置到達 1		39	入力変更 0
5	目標位置到達 2		40	入力変更 1
15	stallGuard™ 軸 0		41	入力変更 2
21	偏差軸 0		42	入力変更 3
27	左停止スイッチ 0		255	グローバル割り込み
28	右停止スイッチ 0		29	左停止スイッチ 1

例：

全体的に割り込みを無効にする場合

DI, 255

DI のバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値 (16 進数)	\$01	\$1A	\$FF	\$00	\$00	\$00	\$00	\$00

目標位置到達時に割り込みを無効にする場合

DI, 3

DI のバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト 3	オペランドバイト 2	オペランドバイト 1	オペランドバイト 0
値 (16 進数)	\$01	\$1A	\$03	\$00	\$00	\$00	\$00	\$00

4.5.35 RETI（割り込みから復帰）

このコマンドは割り込み処理ルーチンを終了させ、通常プログラムの実行を継続させます。

割り込み処理ルーチンの最後には、RETI コマンドを実行する必要があります。

内部機能：保存されたレジスタ（A レジスタ、X レジスタ、フラグ）はコピーして戻されます。通常プログラムの実行が継続します。

関連コマンド：EI、DI、VECT

ニモニック：RETI

バイナリ表現：

命令番号	種類	モータ／バンク	値
38	無関係	無関係	無関係

例：割り込み処理を終了させ、通常プログラムの実行を継続させる場合

RETI

RETI のバイナリ形式：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	命令番号	種類	モータ／バンク	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$01	\$26	\$00	\$00	\$00	\$00	\$01	\$00

4.5.36 お客様固有の TMCL™ コマンド拡張（UF0～UF7 - ユーザ機能）

ユーザ定義機能 UF0～UF7 は、ユーザ固有の目的のためのトピックのない定義済み機能です。ユーザ機能（UF）コマンドは三つのパラメータを使用します。お客様固有のプログラミングについては、TRINAMIC にお問い合わせください。

内部機能：TRINAMIC により C で実装されたユーザ固有機能を呼び出す場合

関連コマンド：なし

ニモニック：UF0～UF7 <パラメータ番号>

バイナリ表現：

命令番号	種類	モータ／バンク	値
64... 71	ユーザ定義	ユーザ定義	ユーザ定義

ダイレクトモードでの回答：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	目標アドレス	状態	命令	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値（16進数）	\$02	\$01	ユーザ定義	64... 71	ユーザ定義	ユーザ定義	ユーザ定義	ユーザ定義

4.5.37 目標位置到達イベント

このコマンドは **TMCL™** プロトコルの唯一の例外であり、二つの回答を送信します。一つはコマンド実行直後に送信され（他の全てのコマンドと同様）、もう一つの追加回答はモータが目標位置に到達したときに送信されます。この命令はダイレクトモードでのみ使用でき（スタンドアロンモードでは **WAIT** コマンドによりカバーされています）、そのためニモニックを持ちません。

内部機能：モータが目標位置に到達したときに追加回答を送信

ニモニック：---

バイナリ表現：

命令番号	種類	モータ／バンク	値
138	0/1	(無関係)	モータビットマスク

コマンド **138** では、値フィールドはビットベクトルです。それは、どのモータ用に目標位置到達メッセージを設定するかを示します。値フィールドはビットマスクを含み、各ビットは一つのモータを表わしています。

モータビットマスク

ビット	選択されたモータ
0	0
1	1
2	2

種類を表す値

値	内容
0	次の MVP コマンド用に位置到達メッセージを設定
1	全 MVP コマンド用に位置到達メッセージを設定

ダイレクトモードでの回答（本コマンド実行直後）：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	目標アドレス	状態	命令	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値 (16 進数)	\$02	\$01	100	138	\$00	\$00	\$00	モータビットマスク

追加回答は、少なくとも最初のモータが目標位置に達したときに送信されます。**MCST3601** は最大 3 個のモータの制御が可能です。

ダイレクトモードでの追加回答（モータが目標位置に到達後）：

バイトインデックス	0	1	2	3	4	5	6	7
機能	目標アドレス	目標アドレス	状態	命令	オペランドバイト3	オペランドバイト2	オペランドバイト1	オペランドバイト0
値 (16 進数)	\$02	\$01	128	138	\$00	\$00	\$00	モータビットマスク

4.5.38 TMCL™制御機能

TMCL™制御機能はいくつかありますが、ユーザにはコマンド **136** および **137** のみに関係します。その他の制御機能は軸パラメータとともに使用できます。

命令番号	種類	コマンド	内容
136	0 - 文字列 1 - バイナリ	ファームウェアバージョン	モジュール種類およびファームウェアバージョンを文字列またはバイナリ形式で取得します (モータ/バンクおよび値は無視されます)。
137	無関係	工場出荷時の設定にリセット	EEPROM に保存されている設定を全て工場出荷時の設定にリセットします。このコマンドは回答を送り返しません。値は 1234 である必要があります。

コマンド 136 に関する追加情報

- 種類設定 0 - 文字列として回答：

バイトインデックス	内容
1	ホストアドレス
2... 9	バージョン文字列 (3601V133 などの 8 文字)

本回答形式ではチェックサムはありません！

- 種類設定 1 - バイナリ形式のバージョン番号

標準回答形式を使用してください。バージョン番号は、以下のように回答の値フィールドに出力されます。

値フィールドのバイトインデックス	内容
1	バージョン番号、低バイト
2	バージョン番号、高バイト
3	種類番号、低バイト (現在は未使用)
4	種類番号、高バイト (現在は未使用)

5 カスタム指定機能

現行の標準 **TMCL** 機能とは対照的に、**MCST3601** モジュールの軸 **0** に接続されたステッパモータは、電源投入後は自動的に次のフルステップ位置に移動します。

6 軸パラメータ

以下のセクションでは、**SAP**、**GAP**、**AAP**、**STAP** および **RSAP** コマンドで使用可能な全ての軸パラメータについて説明します。

注意

モジュールはドライバ IC を一つしか持っていないので、下記の軸パラメータは軸 0 のみに使用できます。

#6、#7

#140

#160～#184

#204～#254

アクセス欄の文字の意味

アクセス 種類	関連コマンド	内容
R	GAP	読み取り可能パラメータ
W	SAP, AAP	書き込み可能パラメータ
E	STAP, RSAP	リセットまたは電源投入後に自動的に EEPROM から復元されるパラメータ これらのパラメータは STAP コマンドを使用して EEPROM に永続的に保存でき、RSAP コマンドを使用して明確に復元することもできます (EEPROM から RAM へコピーして戻れます)。

モジュールを適切に運転するには、基本パラメータをモータ／アプリケーションに合わせて調整する必要があります。

熟練ユーザ用パラメータ - 操作に自信がもてない場合は変更しないでください。

番号	軸パラメータ	内容	範囲 [単位]	アクセス
0	目標 (次) 位置	位置モードでの目標位置 (ランプ波モード、No.138 参照)	-2,147,483,648 ～ +2,147,483,647 [マイクロステップ]	RW
1	実位置	モータの現在位置。基準点設定用にのみ書き	-2,147,483,648 ～ +2,147,483,647 [マイクロステップ]	RW
2	目標 (次) 速度	速度モードでの目標速度 (ランプ波モード、No.138 参照) 位置モードでは、本パラメータはハードウェアにより、加速時には最高速度に、減速時および停止時にはゼロに設定されます。	±2047 16MHz / 65536 · 2 ^{PD} マイクロステップ / 秒	RW
3	実速度	現在の回転速度	±2047 16MHz / 65536 · 2 ^{PD} マイクロステップ / 秒	RW
4	最大位置決め速度	物理的に可能な最大値を超えてはいけません。速度値が非常に低い (<50) または上限を超える場合は、パルス除数 (軸パラメータ 154) を調整してください。物理単位計算用 TMC 429 データシートを参照するか、TMCL-IDE 計算ツールを使用してください。	0... 2047 16MHz / 65536 · 2 ^{PD} マイクロステップ / 秒	RWE

番号	軸パラメータ	内容	範囲 [単位]	アクセス																																								
5	最大加速度	加速度および減速度の限界です。 パラメータを変更するためには、加速度係数および加速度除数の再計算が必要です。そのため、ランプ波除数（軸パラメータ 153）を慎重に調整してください（調整単位=1）。 物理単位計算用 TMC 429 データシートを参照するか、TMCL-IDE 計算ツールを使用してください。	0... 2047 ^{*1}	RWE																																								
6	絶対最大電流 （CS／電流スケール）	最大値は=255 です。これは、モジュールの最大電流の 100%を意味します。電流調整は 0～255 の範囲で行われ、32 段階で調整できます。 <table><tr><td>0... 7</td><td>79...87</td><td>160... 167</td><td>240... 247</td></tr><tr><td>8... 15</td><td>88... 95</td><td>168... 175</td><td>248... 255</td></tr><tr><td>16... 23</td><td>96... 103</td><td>176... 183</td><td></td></tr><tr><td>24... 31</td><td>104... 111</td><td>184... 191</td><td></td></tr><tr><td>32... 39</td><td>112... 119</td><td>192... 199</td><td></td></tr><tr><td>40... 47</td><td>120... 127</td><td>200... 207</td><td></td></tr><tr><td>48... 55</td><td>128... 135</td><td>208... 215</td><td></td></tr><tr><td>56... 63</td><td>136... 143</td><td>216... 223</td><td></td></tr><tr><td>64... 71</td><td>144... 151</td><td>224... 231</td><td></td></tr><tr><td>72... 79</td><td>152... 159</td><td>232... 239</td><td></td></tr></table> 電流の単位は選択されたモータ電流に合わせて最適に設定されます（ジャンパを使用してまたは使用せずに）。 最も重要なモータ設定です。その理由は、この値が高すぎると、モータが損傷するためです！	0... 7	79...87	160... 167	240... 247	8... 15	88... 95	168... 175	248... 255	16... 23	96... 103	176... 183		24... 31	104... 111	184... 191		32... 39	112... 119	192... 199		40... 47	120... 127	200... 207		48... 55	128... 135	208... 215		56... 63	136... 143	216... 223		64... 71	144... 151	224... 231		72... 79	152... 159	232... 239		0... 255 ジャンパスイッチが設定され、Vsense=0 の場合（パラメータ 179 参照）： $I_{peak} = <値> \times 1.5A/255$ $I_{RMS} = <値> \times 1A/255$ ジャンパスイッチが設定され、Vsense=1 の場合（パラメータ 179 参照）： $I_{peak} = <値> \times 0.8A/255$ $I_{RMS} = <値> \times 0.57A/255$ ジャンパスイッチが設定されず、Vsense=0 の場合（パラメータ 179 参照）： $I_{peak} = <値> \times 0.37A/255$ $I_{RMS} = <値> \times 0.26A/255$ ジャンパスイッチが設定されず、Vsense=1 の場合（パラメータ 179 参照）： $I_{peak} = <値> \times 0.20A/255$ $I_{RMS} = <値> \times 0.147A/255$	RWE
0... 7	79...87	160... 167	240... 247																																									
8... 15	88... 95	168... 175	248... 255																																									
16... 23	96... 103	176... 183																																										
24... 31	104... 111	184... 191																																										
32... 39	112... 119	192... 199																																										
40... 47	120... 127	200... 207																																										
48... 55	128... 135	208... 215																																										
56... 63	136... 143	216... 223																																										
64... 71	144... 151	224... 231																																										
72... 79	152... 159	232... 239																																										
7	待機電流	モータ停止 2 秒後の電流限界です。 電流の単位は選択されたモータ電流に合わせて最適に設定されます（ジャンパを使用してまたは使用せずに）。	0... 255 パラメータ 6 と同じ変換	RWE																																								
8	目標位置到達	実位置が目標位置と等しいことを示しています。	0/1	R																																								
9	基準点スイッチ状態	基準（左）スイッチの論理状態です。 異なるスイッチモードに関しては、TMC 429 データシートを参照してください。デフォルトでは二つのスイッチモードがあります。基準点スイッチとしての左スイッチと、リミット（停止）スイッチとしての右スイッチです。	0/1	R																																								
10	右リミットスイッチ状態	（右）リミットスイッチの論理状態です。	0/1	R																																								

番号	軸パラメータ	内容	範囲 [単位]	アクセス	
11	左リミットスイッチ状態	左リミットスイッチの論理状態です (3 スイッチモード時)。	0/1	R	
130	最小速度	目標位置に正確に到達するように、常に 1 に設定しておく必要があります。変更しないでください。	0... 2047 16MHz / 65536 · 2 ^{PD} マイクロステップ / 秒	RWE	
135	実加速度	現在の加速度 (リードオンリー)	0... 2047 ¹	R	
138	ランプ波モード	ROR、ROL、MST および MVP コマンドを使用したときに自動で設定されます。 0 : 位置モード。実位置パラメータと目標位置パラメータが異なる場合は、ステップが発生します。台形速度ランプ波が提供されます。 2 : 速度モード。パラメータ目標速度が変更されている場合は、モータは連続して回転し、速度は一定 (最高) 加速度に合わせて変化します。 特別な目的には、速度の指数関数的な減少を伴うソフトモード (値 1) を選択できます。	0/1/2	RWE	
140	マイクロステップ分解能	0	フルステップ	0... 8	RWE
		1	ハーフステップ		
		2	4 マイクロステップ		
		3	8 マイクロステップ		
		4	16 マイクロステップ		
		5	32 マイクロステップ		
		6	64 マイクロステップ		
		7	128 マイクロステップ		
		8	256 マイクロステップ		
153	ランプ波除数	ランプ波発生器のスケール係数の指数です。慎重に増加・減少させる必要があります (調整単位=1)。	0... 13	RWE	
154	パルス除数	パルス (ステップ) 発生器のスケール係数の指数です。慎重に増加・減少させる必要があります (調整単位=1)。	0... 13	RWE	
160	ステップ補間有効	ステップ補間は、16 マイクロステップ設定でのみサポートされています。本設定では、入力での各ステップパルスにより、16×1/256 マイクロステップが実行されます。このように、256 マイクロステップ分解能で動作のようなスムーズなモータの動作が実現できます。 0 – ステップ補間オフ 1 – ステップ補間オン	0/1	RW	
161	二重ステップ有効	サイクルの各エッジでステップ／マイクロステップを解放します。内部使用にこのパラメータを有効にすることは意味がありません。 二重ステップ有効はステップ／方向インターフェースで使用できます。 0 – 二重ステップオフ 1 – 二重ステップオン	0/1	RW	

番号	軸パラメータ	内容	範囲 [単位]	アクセス
162	チョップブラン ク時間	比較器ブランク時間を選択します。この時間 は、切り替えイベントとセンス抵抗のリンギ ング時間を安全にカバーできる必要がありま す。	0... 3	RW
163	チョップモード	チョップモードの選択： 0 – 拡散周期 1 – 従来の一一定オフ時間	0/1	RW
164	チョップヒステ リシスデクリメ ント	ヒステリシスデクリメント設定。この設定は、 オン時間および高速減衰期間中のヒステリシ スの傾斜を決定します。 0 – 高速デクリメント 3 – 超低速デクリメント	0... 3	RW
165	チョップヒステ リシス終了	ヒステリシス終了設定。複数のデクリメント 後のヒステリシス終了値を設定します。デク リメント間隔時間は軸パラメータ 164 により 制御されます。 - 3... - 1 負ヒステリシス終了設定 0 ゼロヒステリシス終了設定 1... 12 正ヒステリシス終了設定	- 3... 12	RW
166	チョップヒステ リシス開始	ヒステリシス開始設定。この値はヒステリシ ス終了値のオフセットですのでご注意ください。 。	0... 8	RW
167	チョップオフ時 間	オフ時間設定は、最小チョップ周期を制御し ます。5 μ s～20 μ s の間のオフ時間が適切です。 定数 t_{off} チョップ用オフ時間設定 $N_{CLK} = 12 + 32 \cdot t_{OFF}$ (最小は 64 クロック) このパラメータをゼロに設定することによ り、ドライバトランジスタを全て完全に無効 にし、モータは惰性で回転します。	0 / 2... 15	RW
168	スマートエナジ ー 最 小 電 流 (SEIMIN)	CS (電流スケール、軸パラメータ 6 参照) 値 をスケールリングすることにより、coolStep™ 運転用の最小モータ電流を設定します。 最小モータ電流： 0 - CS の 1/2 1 - CS の 1/4	0/1	RW
169	スマートエナジ ー電流減少ステ ップ	モータ電流の各電流デクリメントのために必 要な、上限しきい値より上の stallGuard2™ の 読み取り数を設定します。 デクリメントあたりの stallGuard2™ 計測数： スケールリング：0～3：32、8、2、1 0：低速デクリメント 3：高速デクリメント	0... 3	RW
170	スマートエナジ ーヒステリシス	stallGuard2™ 読み取りの下限しきい値と上 限しきい値の間の距離を設定します。上限し きい値より上では、モータ電流は減少します。 ヒステリシス： (スマートエナジーヒステリシス値+1) \times 32 上限 stallGuard2™ しきい値： (スマートエナジーヒステリシス開始+スマ ートエナジーヒステリシス+1) \times 32	0... 15	RW

番号	軸パラメータ	内容	範囲 [単位]	アクセス						
171	スマートエナジー電流増加ステップ	電流増加ステップを設定します。下限しきい値未満では、各 stallGuard2™ 計測値に対して電流は増大します（スマートエナジーヒステリシス開始参照）。 電流インクリメントステップサイズ： スケーリング：0～3：1、2、4、8 0：低速インクリメント 3：高速インクリメント／立ち上がり負荷に対する高速応答	1... 3	RW						
172	スマートエナジーヒステリシス開始	stallGuard2™ 値の下限しきい値です（スマートエナジー電流増加ステップ参照）。	0... 15	RW						
173	stallGuard2™ フィルタ有効化	計測をより正確にするため、 stallGuard2™ フィルタを有効にします。本パラメータが設定されると、計測周期を 4 回のフルステップあたり一回の計測に減らします。 ほとんどの場合、 coolStep™ の使用前にフィルタモードを設定した方が便利です。 ステップ喪失を検出するには、標準モードを使用してください。 0 – 標準モード 1 – フィルタモード	0/1	RW						
174	stallGuard2™ しきい値	この符号付き値は、ストール出力用 stallGuard2™ しきいレベルを制御し、読み出しのため最適な計測範囲を設定します。 値が低いほど、感度は高くなります。ゼロは開始値です。値が高いほど stallGuard2™ の感度は低くなり、ストールを表示させるためにより大きなトルクが必要となります。 <table><tr><td>0</td><td>無関係</td></tr><tr><td>1... 63</td><td>低感度</td></tr><tr><td>- 1 - 64</td><td>高感度</td></tr></table>	0	無関係	1... 63	低感度	- 1 - 64	高感度	- 64... 63	RW
0	無関係									
1... 63	低感度									
- 1 - 64	高感度									
175	傾斜制御 - 高側	モータドライバ出力の傾斜を決定します。 本モジュールでは2または3を設定するか、むしろデフォルト値を使用してください。 0: 最遅傾斜 3: 最速傾斜	0... 3	RW						
176	傾斜制御 - 低側	モータドライバ出力の傾斜を決定します。 「傾斜制御 - 高側」と同じ設定をしてください。	0... 3	RW						
177	短絡保護無効化	0：アースへの短絡保護のオン 1：アースへの短絡保護のオフ デフォルト値を使用してください。	0/1	RW						
178	短絡検出タイマ	0：3.2μs 1：1.6μs 2：1.2μs 3：0.8μs デフォルト値を使用してください！	0... 3	RW						
179	Vsense	センス抵抗電圧に基づく電流スケール 0：フルスケールのセンス抵抗電圧は、最大 1A RMS／1.5A ピーク（ジャンパクローズ時）または最大 0.26A RMS／0.37A ピーク（ジャンパオープン時）です。 1：フルスケールのセンス抵抗電圧は、最大 0.57A RMS／0.8A ピーク（ジャンパクローズ時）または最大 0.14A RMS／0.24A ピーク（ジャンパオープン時）です。	0/1	RW						

番号	軸パラメータ	内容	範囲 [単位]	アクセス
180	スマートエナジー 実電流	この状態値は、coolStep™により制御される実モータ電流設定です。値は CS 値まで上昇し、SEIMIN で指定される CS の部分まで減少します。 <u>実モータ電流スケール係数</u> 0 ... 31: 1/32, 2/32, ... 32/32	0... 31	RW
181	ストールにて停止	この速度未満では、モータは停止しません。この速度より大きいと、stallGuard2™負荷値がゼロに達したときにモータが停止します。	0... 2047 16MHz / 65536 · 2 ^{PD} マイクロステップ / 秒	RW
182	スマートエナジー しきい値速度	この速度より大きいと、coolStep™が有効になります。	0... 2047 16MHz / 65536 · 2 ^{PD} マイクロステップ / 秒	RW
183	smartEnergy 低速 電流	しきい値速度未満で 사용되는モータ電流を設定します。 電流の単位は選択されたモータ電流に合わせて最適に設定されます（ジャンプを使用してまたは使用せずに）。	0... 255 パラメータ 6 と同じ変換	RW
193	基準点サーチモード	<div> <div>1 左停止スイッチのみをサーチ</div> <div>2 右停止スイッチ、次に左停止スイッチをサーチ</div> <div>3 両側から右停止スイッチ、次に左停止スイッチをサーチ</div> <div>4 両側から左停止スイッチをサーチ</div> <div>5 原点復帰スイッチを負方向でサーチ、左停止スイッチ到達時に方向反転</div> <div>6 原点復帰スイッチを正方向でサーチ、右停止スイッチ到達時に方向反転</div> <div>7 原点復帰スイッチを正方向でサーチ、終了スイッチを無視</div> <div>8 原点復帰スイッチを負方向でサーチ、終了スイッチを無視</div> </div> これらの値に 128 を加算すると、原点復帰スイッチ入力 of 極性を反転します。	1... 8	RWE
194	基準点サーチ速度	基準点サーチでは、この値が直接サーチ速度を指定します。	0... 2047	RWE
195	基準点スイッチ速度	パラメータ No.194 と同じように、切り替え点校正用速度を選択できます。	0... 2047	RWE
196	終了スイッチ間距離	本パラメータは、RFS コマンド（モード 2 または 3）実行後の終了スイッチ間の距離を指定します。	0... 2147483647	R
204	惰性回転	モータの速度がゼロに達したときの、モータへの電源を切断するまでの時間を指定します。	0... 65535 0 = 切断なし [msec]	RWE
206	実負荷値	ストール検出（stallGuard2™）に使用される実負荷値を読み出します。	0... 1023	R

番号	軸パラメータ	内容	範囲 [単位]	アクセス
208	TMC262 ドライバ エラーフラグ	ビット 0 stallGuard2™状態 (1: しきい値到達) ビット 1 過温度 (1: 過温度によるドライバのシャットダウン) ビット 2 過温度事前警告 (1: しきい値の超越) ビット 3 アース A への短絡 (1: 短絡条件の検出、ドライバは現在シャットダウン) ビット 4 アース B への短絡 (1: 短絡条件の検出、ドライバは現在シャットダウン) ビット 5 負荷 A のオープン (1: 固定コイル極性での前回期間中に、チョッパイベント発生せず) ビット 6 負荷 B のオープン (1: 固定コイル極性での前回期間中に、チョッパイベント発生せず) ビット 7 静止 (1: 前回の 2~20 クロックサイクル期間中に、ステップ入力にステップパルス発生せず) 詳細は TMC262 データシートを参照してください。	0... 255	R
209	エンコーダ位置	エンコーダレジスタの値は読み出されるか書き込まれます。	[エンコーダステップ数]	RW
210	エンコーダプリスケアラ	エンコーダ用プリスケアラ	6.2.1 項参照	RWE
212	最大エンコーダ偏差	実位置 (パラメータ 1) とエンコーダ位置 (パラメータ 209) との差がここで設定した値以上であれば、モータは停止します。この機能は、最大偏差がゼロに設定されたときにオフします。	0... 65535 [エンコーダステップ数]	RWE
214	動力低下遅延	電流が待機電流へ低下する前の静止期間です。標準値は 200 (2000 ミリ秒に相当する値) です。	1... 65535 [10 ミリ秒]	RWE
254	ステップ/方向モード	0 標準モード。ステップ/方向モードはオフ。 1 静止時に自動的に電流が低下するステップ/方向モード。静止時の電流低下が好ましくない場合は、軸パラメータ #6 および #7 に同じ値を選択してください。	0/1	RWE

*1 加速度単位 : $16\text{MHz}^2/536870912 \cdot 2$ パルス除数+ランブ波除数 マイクロステップ/秒²

6.1 基準点サーチ

内蔵された基準点サーチ機能により、切り替え点校正が行え、一つまたは二つの基準点スイッチをサポートします。内部動作は、開始、停止および監視可能なステートマシンをベースにしています（RFS 命令 No.13）。基準点スイッチは左リミットスイッチと直列で接続されています。左リミットスイッチと原点復帰スイッチの区別はソフトウェアにより行われます。オープン接点（常閉型）が使用されます。モジュールのアナログ入力 AIN_0 は原点復帰スイッチとして使用できます。

基準点サーチのヒント：

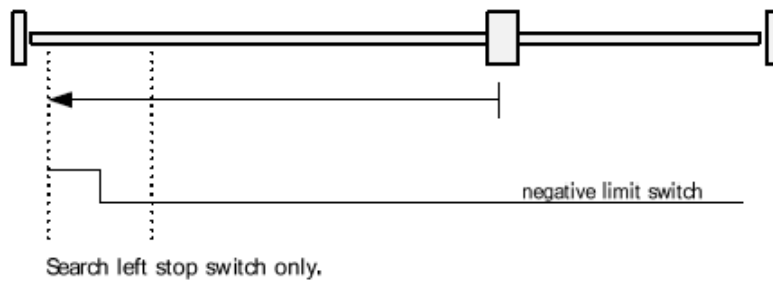
- スイッチ（軸パラメータ 12 および 13）に対応する自動停止機能の設定は、基準点サーチには影響を及ぼしません。
- 基準点スイッチが最初に検出されるまで、サーチ速度は軸パラメータ 194 で減速されていない限り最大位置決め速度と同じです（軸パラメータ 4）。
- 基準点スイッチに当たると、モータはスイッチが解放されるまで低速で動作します。最終的にスイッチは反対方向に再度オンし、基準点を二つの切り替え点の中間に設定します。この低速校正速度はデフォルトでは最大位置決め速度の 1/4 です（軸パラメータ 195）。
- 基準点サーチモードを選択するには、軸パラメータ 193 に値の一つを設定します。

基準点サーチに必要なパラメータ

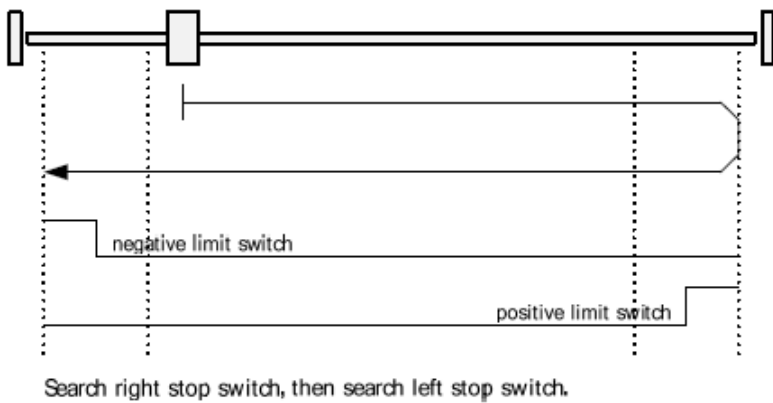
番号	軸パラメータ	内容
9	基準点スイッチ状態	基準（左）スイッチの論理状態です。 異なるスイッチモードに関しては、TMC 429 データシートを参照してください。デフォルトでは二つのスイッチモードがあります。基準点スイッチとしての左スイッチと、リミット（停止）スイッチとしての右スイッチです。
10	右リミットスイッチ状態	（右）リミットスイッチの論理状態です。
11	左リミットスイッチ状態	左リミットスイッチの論理状態です（3 スイッチモード時）。
12	右リミットスイッチ無効化	設定されると、右スイッチの停止機能を無効にします。
13	左リミットスイッチ無効化	設定されると、左スイッチ（基準点スイッチ）の停止機能を無効にします。
141	基準点スイッチ許容値	3 スイッチモード用。追加スイッチ（REFL 入力に接続）がモータを停止させない位置範囲です。
149	ソフト停止フラグ	本パラメータがクリアされると、基準点スイッチまたはリミットスイッチに当たったときに、モータが直ちに停止します（モータ限度は無視）。
193	基準点サーチモード	<ol style="list-style-type: none"> 1 左停止スイッチのみをサーチ 2 右停止スイッチ、次に左停止スイッチをサーチ 3 両側から右停止スイッチ、次に左停止スイッチをサーチ 4 両側から左停止スイッチをサーチ 5 原点復帰スイッチを負方向でサーチ、左停止スイッチ到達時に方向反転 6 原点復帰スイッチを正方向でサーチ、右停止スイッチ到達時に方向反転 7 原点復帰スイッチを正方向でサーチ、終了スイッチを無視 8 原点復帰スイッチを負方向でサーチ、終了スイッチを無視 これらの値に 128 を加算すると、原点復帰スイッチ入力の極性を反転します。
194	基準点サーチ速度	基準点サーチでは、この値が直接サーチ速度を指定します。
195	基準点スイッチ速度	パラメータ No.194 と同じように、切り替え点校正用速度を選択できます。
196	終了スイッチ間距離	本パラメータは、RFS コマンド（モード 2 または 3）実行後の終了スイッチ間の距離を指定します。

6.1.1 基準点サーチモード (軸パラメータ 193)

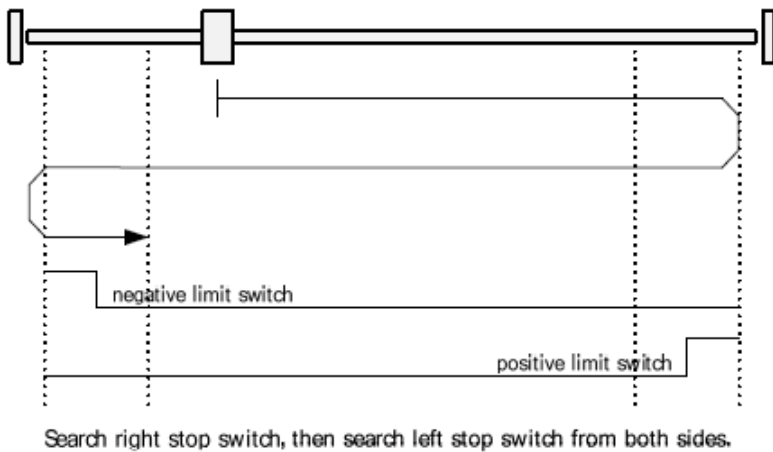
SAP 193, Q, 1



SAP 193, Q, 2



SAP 193, Q, 3



SAP 193, Q, 4

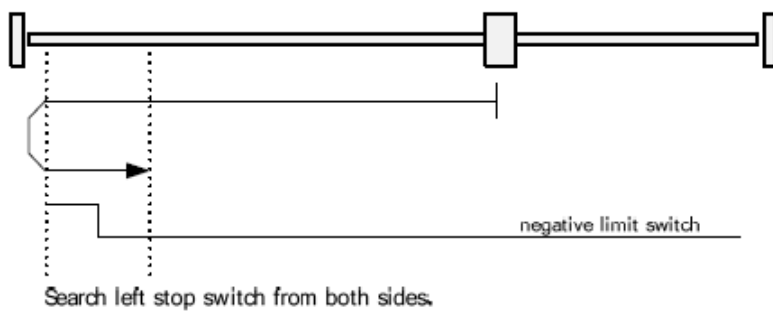
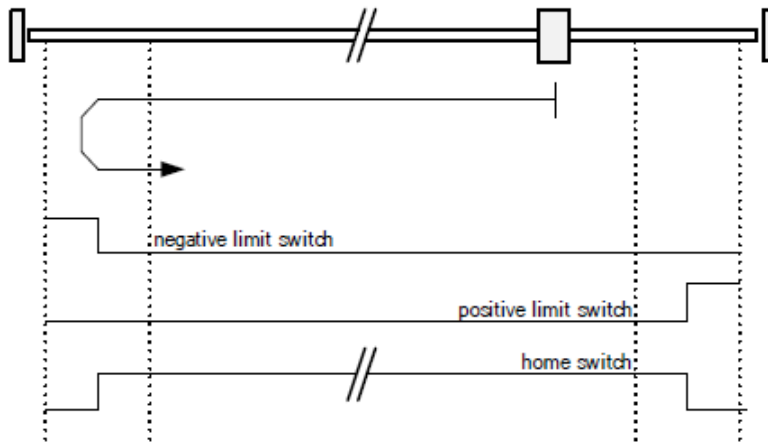


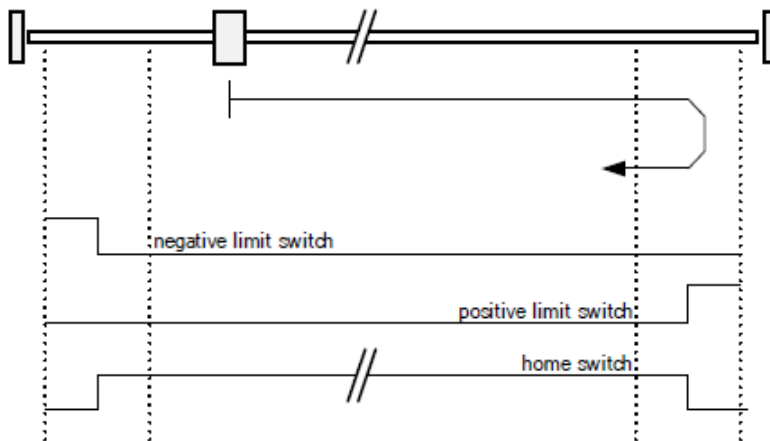
Figure 6.1: Reference search modes 1-4

SAP 193, 0, 5



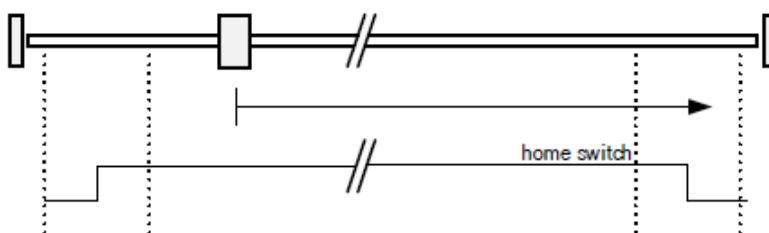
Search home switch in negative direction, reverse the direction when left stop switch reached.

SAP 193, 0, 6



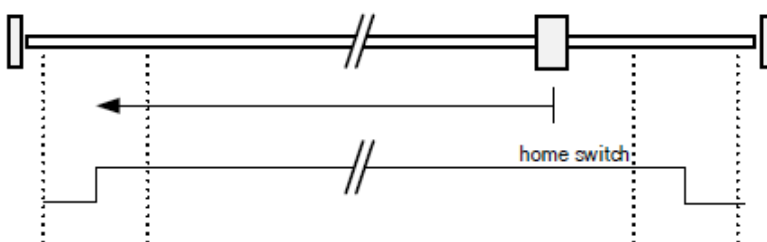
Search home switch in positive direction, reverse the direction when right stop switch reached.

SAP 193, 0, 7



Search home switch in positive direction, ignore end switches.

SAP 193, 0, 8



Search home switch in negative direction, ignore end switches.

Figure 6.2: Reference search modes 5-8

6.2 エンコーダ

MCST3601 は TTL (5V) 出力付き単一端インクリメンタルエンコーダ用インターフェースを提供します。エンコーダを使用して運転する場合は、以下の点を考慮してください。

- エンコーダカウンタはソフトウェアにより読み出され、モータの正確な位置を監視するために使用されます。これによりクローズドループ運転も可能になります。
- エンコーダチャンネル **ENC_I** はエンコーダカウンタをゼロリセットするためのものです。これは高または低で有効になるかを選択でき、正確に基準点に移動するためチャンネル **A** およびチャンネル **B** 入力と並行して自動的に確認されます。
- エンコーダの位置値を読み出しおよび変更するには、パラメータ **209** が使用されます。お使いのエンコーダ **0** の位置を読み出すには、**GAP 209, 0** を使用してください。位置値は **SAP 209, 0, <n>** コマンドを使用しても変更できます。n の値は **-2,147,483,648~+2,147,483,647** の範囲です。
- エンコーダの設定を変更するために、軸パラメータ **210** が使用されます。エンコーダ **0** のプリスケアラを変更するには、**SAP 210, 0, <p>** を使用してください。
- 偏差エラーによるモータの自動停止も使用できます。これは軸パラメータ **212** (最大偏差) を使用して設定できます。この機能は、最大偏差がゼロに設定されたときにオフします。

エンコーダの使用に必要なパラメータ

番号	軸パラメータ	内容	
209	エンコーダ位置	エンコーダレジスタの値は読み出されるか書き込まれます。	[エンコーダステップ数]
210	エンコーダプリスケアラ	エンコーダ用プリスケアラ	6.2.1 項参照
212	最大エンコーダ偏差	実位置 (パラメータ 1) とエンコーダ位置 (パラメータ 209) との差がここで設定した値以上であれば、モータは停止します。この機能は、最大偏差がゼロに設定されたときにオフします。	0... 65535 [エンコーダステップ数]

6.2.1 エンコーダのプリスケアラ値の変更

以下の表は選択可能なプリスケアラサブセットを示しています。この表で示す値の間の数値が使用できます。ビット 2～4 は、特殊エンコーダ機能を選択するために必要ですので、プリスケアラ用には使用できません。

プリスケアラを選択するためには、<P>に下記の値が使用可能です。

<p>用の値	結果プリスケアラ	モータ 0 用 SAP コマンド SAP 210, M0, <p>	400 ライン (4 通倍カウントで 1600) エンコーダ用の 1 回転あたりの結果ステップ
64	0.125	SAP 210, 0, 64	200
128	0.25	SAP 210, 0, 128	400
256	0.5	SAP 210, 0, 256	800
512	1	SAP 210, 0, 512	1600
1024	2	SAP 210, 0, 1024	3200
2048	4	SAP 210, 0, 2048	6400
4096	8	SAP 210, 0, 4096	12800
8192	16	SAP 210, 0, 8192	25600
16384	32	SAP 210, 0, 16384	51200
32768	64	SAP 210, 0, 32768	102400

1 回転当たりの結果ステップ用数式

1 回転当たりのステップ=エンコーダライン数×4×プリスケアラ

「プリスケアラ= p/512」の計算においては、以下の数式を考慮してください。

例 : <p>=6400

$6400/512=12.5$ (プリスケアラ)

これらの数値を使用して構成できる特別な機能がいくつかあります。これらの機能を選択するには、単純に以下の値を<p>に加算してください。

ビット	<p>へ加算する数値	コマンド : SAP 210, <モータ番号>, <p>
2	4	設定されると、エンコーダは次の位置指標チャンネルイベント発生時にゼロリセットされます。
3	8	ビット 2 との組み合わせで設定された場合 : エンコーダは各位置指標チャンネルイベント発生時にゼロリセットされます。
4	16	エンコーダクリア用チャンネル Z 極性 : 0 - 低 1 - 高

これらの表を参照して両方の<p>値を加算していき、SAP 210 コマンドに必要な値を取得します。

結果プリスケアラは値/512 です。

6.3 計算：速度および加速 vs. マイクロステップ／フルステップ周波数

TMC429 に送信される軸パラメータの値は、速度の RPM という通常のモータ数値を持ちません。しかし、本書で示すように、モータ数値などは TMC429 パラメータから計算できます。

TMC429 速度パラメータ

TMC429 速度パラメータ	MCST3601 関連軸パラメータ		範囲 (TMC429 および MCST3601)
速度	軸パラメータ 2	目標（次）速度	0... 2047
	軸パラメータ 3	実速度	
	軸パラメータ 4	最大位置決め速度	
	軸パラメータ 13	最小速度	
	軸パラメータ 194	基準点サーチ速度	
	軸パラメータ 195	基準点スイッチ速度	
a_max / 最大加速度	軸パラメータ 5		0... 2047
μsrs / マイクロステップ分解能 フルステップあたりの マイクロステップ = 2 ^{USRS}	軸パラメータ 140 は以下の設定を提供します：		0... 8
	0	フルステップ	
	1	ハーフステップ	
	2	4 マイクロステップ	
	3	8 マイクロステップ	
	4	16 マイクロステップ	
	5	32 マイクロステップ	
	6	64 マイクロステップ	
	7	128 マイクロステップ	
	8	256 マイクロステップ	
ramp_div / ランプ波除数	軸パラメータ 153：加速度用除数。値が高いほど、最大加速度は低くなります。 デフォルト：0		0... 13
pulse_div / パルス除数	軸パラメータ 153：速度用除数 1 ずつ値を増加するごとに加速度が半減し、1 ずつ値を減少するごとに加速度が 2 倍になります。 デフォルト：0		0... 13
f _{CLK} / クロック周波数	--		16MHz

6.3.1 マイクロステップ周波数

ステップモータのマイクロステップ周波数は以下の数式で計算されます。

$$\mu sf[\text{Hz}] = f_{\text{CLK}}[\text{Hz}] \cdot \text{速度} / 2^{\text{pulse_div}} \cdot 2048 \cdot 32 \quad \mu sf : \text{マイクロステップ周波数}$$

6.3.2 フルステップ周波数

マイクロステップ周波数からフルステップ周波数を計算するには、マイクロステップ周波数をフルステップ当たりのマイクロステップ数で除算する必要があります。

$$fsf[\text{Hz}] = \mu sf[\text{Hz}] / 2^{\mu srs} \quad fsf : \text{フルステップ周波数}$$

時間単位あたりのパルスレートの変化（ a : 1 秒あたりのパルス周波数の変化）は以下の数式で得られます。

$$a = f_{\text{CLK}}^2 \cdot a_{\text{max}} / 2^{\text{pulse_div} + \text{ramp_div} + 29}$$

その結果得られるフルステップでの加速度は以下で表わせます。

$$af = a / 2^{\mu srs} \quad af : \text{フルステップでの加速度}$$

例 :

信号	値
f_{CLK}	16MHz
速度	1000
a_{max}	1000
pulse_div	1
ramp_div	1
μsrs	6

$$\mu sf = 16\text{MHz} \cdot 1000 / 2^1 \cdot 2048 \cdot 32 = 122070.31 \text{ Hz}$$

$$fsf[\text{Hz}] = 122070.31 / 2^6 = 1907.34\text{Hz}$$

$$a = (16\text{Mhz})^2 \cdot 1000 / 2^{1+1+29} = 119.21 \text{ MHz} / \text{s}$$

$$af = 119.21 \text{ MHz} / \text{s} / 2^6 = 1.863 \text{ MHz} / \text{s}$$

6.3.2.1 回転数の計算：

例えば、ステッパモータが回転あたり **72** フルステップを持つ場合、その回転数は以下のとおりです。

$$\text{RPS} = \text{fsf} / 1 \text{ 回転当たりのフルステップ数} = 1907.34 / 72 = 26.49$$

$$\text{RPM} = \text{fsf} \cdot 60 / 1 \text{ 回転当たりのフルステップ数} = 1907.34 \cdot 60 / 72 = 1589.46$$

7 グローバルパラメータ

グローバルパラメータは4つのバンクにグループ分けされます：

- バンク 0 (モジュールのグローバル構成)
- バンク 1 (ユーザ C 変数)
- バンク 2 (ユーザ TMCL™変数)
- バンク 3 (割り込み構成)

グローバルパラメータの書き込みおよび読み出しには、SGP および GGP コマンドを使用してください！

7.1 バンク 0

パラメータ 0～38

0～38 までのパラメータは、説明の目的のみでここに記載しています。これらは TMCL-IDE の内部処理に使用され、マイクロステップおよびドライバ表のロードの役割を担います。通常はこれらのパラメータは手を加えられません。PC ソフトウェアを使用して特定の値をロードするためにこれらのパラメータを使用する場合は、TRINAMIC まで使用方法についてお問い合わせください。さもないと、モータドライバに損傷を与える恐れがあります。

番号	パラメータ
0	データグラム低ワード (リードオンリー)
1	データグラム高ワード (リードオンリー)
2	データグラム位置
3	データグラム長
4	データグラム内容
5	基準点スイッチ状態 (リードオンリー)
6	TMC429 SMGP レジスタ
7... 22	ドライバチェーン構成長ワード 0～15
23... 38	マイクロステップ表長ワード 0～15

パラメータ 64～132

番号 64 以降のパラメータは、モジュールの RS485 ボーレートのシリアルアドレスなどの要素を構成します。必要に応じてこれらのパラメータを変更してください。変更を行う最適かつ最も簡単な方法は、TMCL-IDE の適切な機能を使用することです。番号 64～128 のパラメータは EEPROM にのみ保存されています。

そのようなパラメータに対する SGP コマンドはパラメータを永続的に保存し、さらに STGP コマンドを必要とすることはありません。これらのパラメータを変更する際は注意を払い、TMCL-IDE の適切な機能をインタラクティブな方法で使用してください。

アクセス欄の文字の意味

アクセス種類	関連コマンド	内容
R	GGP	読み取り可能パラメータ
W	SGP, AGP	書き込み可能パラメータ
E	SGP, AGP	EEPROM に永続的に保存されるパラメータ

番号	グローバルパラメータ	内容		範囲	アクセス	
64	EEPROM マジック	このパラメータを\$E4 とは異なる値に設定すると、次回電源投入時に軸パラメータおよびグローバルパラメータが再初期化（工場出荷時設定）されます。構成に間違いがあった場合にこの機能は有効です。		0... 255	RWE	
65	シリアルインターフェースボーレート	0	9600 ボー	デフォルト	0... 11	RWE
		1	14400 ボー			
		2	19200 ボー			
		3	28800 ボー			
		4	38400 ボー			
		5	57600 ボー			
		6	76800 ボー	Windows は非対応！		
		7	115200 ボー			
		8	230400 ボー			
		9	250000 ボー	Windows は非対応！		
		10	500000 ボー	Windows は非対応！		
		11	1000000 ボー	Windows は非対応！		
66	シリアルアドレス	RS485 用モジュール（目標）アドレス		0... 255	RWE	
68	シリアルハートビート	USB インターフェース用シリアルハートビート この時間制限を超え、コマンドがそれ以降認識されない場合、モータは停止します。 0 – パラメータの無効化		[ms]	RWE	
69	予約					
70	予約					
71	予約					
73	構成 EEPROM ロックフラグ	書き込み：1234 で EEPROM をロック、4321 で解除 読み込み：1 で EEPROM をロック、0 で解除		0/1	RWE	
76	シリアルホストアドレス	USB を介して送り返される回答信号で使用されるホストアドレス		0... 255	RWE	
77	自動スタートモード	0：電源投入後に TMCL™アプリケーションを起動しない（デフォルト） 1：電源投入後に TMCL™アプリケーションを起動		0/1	RWE	
81	TMCL™コード保護	TMCL™を逆アセンブルや上書きから保護します。 0 – 保護なし 1 – 逆アセンブルに対して保護 2 – 上書きに対して保護 3 – 逆アセンブルと上書きに対して保護 逆アセンブルに対する保護をオフにすると、まずプログラムが消去されます。 この値を 1 または 3 から 0 または 2 に変更すると、TMCL™プログラムが消去されます。		0,1,2,3	RWE	
82	予約					
83	予約					
84	座標保存	0 – 座標は RAM にのみ保存されます（ただし RAM と EEPROM との間で明確にコピーできます）。 1 – 座標は常に EEPROM にのみ保存されます。		0 または 1	RWE	
番号	グローバルパラメータ	内容		範囲	アクセス	
85	ユーザ変数の復元不可	0 – ユーザ変数は復元されます（デフォルト）。 1 – ユーザ変数は復元されません。		0/1	RWE	
86	ステップパルス長	ステップパルスの長さ（ステップ／方向インターフェース用）：(1 + x) μs デフォルト設定：0 (1μs) 本設定は三つのモータ軸全てに有効です。		0... 15	RWE	
128	TMCL™アプリケーション状態	0 – 停止 1 – 実行		0... 3	R	

		2 - ステップ 3 - リセット		
129	ダウンロードモード	0 - 標準モード 1 - ダウンロードモード	0/1	R
130	TMCL™プログラムカウンタ	現在実行されている TMCL™命令の指標です。		R
132	タイマ	ミリ秒ごとに 1 ずつカウントアップする 32 ビットカウンタです。任意の開始値にリセットすることもできます。		RW
133	ランダム番号	ランダム番号を選択します。	0... 2147483647	R

7.2 バンク 1

グローバルパラメータバンク 1 は通常は使用できません。ファームウェアのユーザ固有拡張に使用できる場合があります。ユーザ定義コマンド (6.3 節参照) とともに、これらの変数はファームウェアの拡張 (C 言語記述) および TMCL™アプリケーションの間のインターフェースを形成します。

7.3 バンク 2

バンク 2 は、TMCL™アプリケーションでの使用用の汎用 32 ビット変数を含みます。これらの変数は RAM 内に存在し、EEPROM に保存できます。起動後は、これらの値は自動的に RAM に復元されます。

最大 56 個のユーザ変数が使用可能です。

アクセス欄の文字の意味

アクセス種類	関連コマンド	内容
R	GGP	読み取り可能パラメータ
W	SGP, AGP	書き込み可能パラメータ
E	SGP, AGP	EEPROM に永続的に保存されるパラメータ

番号	グローバルパラメータ	内容	範囲	アクセス
0... 55	汎用変数#0～#55	TMCL™アプリケーションでの使用	$-2^{31} \dots +2^{31}$	RWE
56... 255	汎用変数#56～#255	TMCL™アプリケーションでの使用	$-2^{31} \dots +2^{31}$	RW

7.4 バンク 3

バンク 3 は割り込みパラメータを含みます。割り込みには構成が必要なものがあります（タイマ割り込みの時間間隔など）。これはパラメータバンク 3 とともに SGP コマンド（SGP <種類>, 3, <値>）を使用して構成できます。割り込みの優先度順位はその番号により決まります。低い番号を有する割り込みには高い優先度順位があります。

以下の表は設定可能な全ての割り込みパラメータを示しています。

アクセス欄の文字の意味

アクセス種類	関連コマンド	内容
R	GGP	読み取り可能パラメータ
W	SGP, AGP	書き込み可能パラメータ
E	SGP, AGP	EEPROM に永続的に保存されるパラメータ

番号	グローバルパラメータ	内容	範囲	アクセス
0	タイマ 0 期間 (ms)	二つの割り込み間の時間 (ms)	符号なし 32 ビット [ms]	RWE
1	タイマ 1 期間 (ms)	二つの割り込み間の時間 (ms)	符号なし 32 ビット [ms]	RWE
2	タイマ 2 期間 (ms)	二つの割り込み間の時間 (ms)	符号なし 32 ビット [ms]	RWE
27	左停止スイッチ 0 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
28	右停止スイッチ 0 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
29	左停止スイッチ 1 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
30	右停止スイッチ 1 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
31	左停止スイッチ 2 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
32	右停止スイッチ 2 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
39	入力 0 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
40	入力 1 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
41	入力 2 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE
42	入力 3 エッジ種類	0=オフ、1=低-高、2=高-低、3=両方	0... 3	RWE

8 TMCL™プログラミング技術および構成

8.1 初期化

TMCL™プログラムで行う最初の仕事（他のプログラムでも同様）は、デフォルト値とは異なる値が必要なパラメータを全て初期化することです。この目的のため、**SAP** および **SGP** コマンドが使用されます。

8.2 メインループ

組み込まれたシステムは通常は無限に実行されるメインループを使用します。これは、スタンドアロンで実行される **TMCL™**でも同様です。通常は、モジュールの自動起動モードがオンします。

電源投入後、モジュールは **TMCL™**プログラムを起動します。**TMCL™**プログラムは必要な初期化を全て行い、それからメインループに入ります。メインループは必要なタスクを全て実行し、そして決して終了しません（モジュールの電源が切断されるかモジュールがリセットされた際にのみ終了します）。

例外もあります。例えば、ダイレクトモードで **TMCL™**ルーチンがホストから呼び出された場合などです。

そのため、ほとんどの（全てではありません）スタンドアロン **TMCL™**プログラムは以下のように構成されています。

```
//Initialization
SAP 4, 0, 500 //define max. positioning speed
SAP 5, 0, 100 //define max. acceleration

MainLoop:
//do something, in this example just running between two positions
MVP ABS, 0, 5000
WAIT POS, 0, 0
MVP ABS, 0, 0
WAIT POS, 0, 0
JA MainLoop //end of the main loop => run infinitely
```

8.3 記号定数の使用

お使いのプログラムを読みやすくかつ分かりやすくするため、プログラム内で使用される重要な数値には全て記号定数を代用すべきです。**TMCL-IDE** は、全ての重要な軸パラメータおよびグローバルパラメータ用の記号名を含んだインクルードファイルを提供しています。

例：

```
//Define some constants
#include TMCLParam.tmc
MaxSpeed = 500
MaxAcc = 100
Position0 = 0
Position1 = 5000

//Initialization
SAP APMaxPositioningSpeed, Motor0, MaxSpeed
SAP APMaxAcceleration, Motor0, MaxAcc

MainLoop:
MVP ABS, Motor0, Position1
WAIT POS, Motor0, 0
MVP ABS, Motor0, Position0
WAIT POS, Motor0, 0
JA MainLoop
```

TMCL-IDE に含まれるファイル **TMCLParam.tmc** を参照してください。それには全ての重要なパラメータを定義する記号定数が含まれます。

他の値に定数を使用することで、それらがプログラムで 2 回以上使用された場合に変更を簡単にできます。定数の定義を変更できるため、プログラム内で定数が出現する度に変更を行う必要がありません。

8.4 変数の使用

お使いのプログラムで変数が必要な場合には、ユーザ変数を使用することができます。ユーザ変数は一時的な値を保存できます。

SGP、**GGP** および **AGP** コマンドは、ユーザ変数とともに使用されます。

SGP は変数を定数に設定するために使用されます（初期化段階中など）

GGP はユーザ変数の内容を読み込み、以降の使用のためにアキュムレータレジスタに保存するために使用されます。

AGP は、例えば計算結果を保存するためなどに、アキュムレータレジスタの内容をユーザ変数にコピーするために使用されます。

例：

```
MyVariable = 42
```

```
//Use a symbolic name for the user variable
```

```
//(This makes the program better readable and understandable.)
```

```
SGP MyVariable, 2, 1234 //Initialize the variable with the value 1234
```

```
GGP MyVariable, 2 //Copy the contents of the variable to the accumulator register
```

```
CALC MUL, 2 //Multiply accumulator register with two
```

```
AAP MyVariable, 2 //Store contents of the accumulator register to the variable
```

さらに、これらの変数はモジュールで実行中の **TMCL™** プログラムとホストの間の強力な通信方法を提供できます。ホストはダイレクトモード **SGP** コマンドを発行することで変数を変更できます（**TMCL™** プログラムが実行中に、ダイレクトモードコマンドは実行中のプログラムを妨げることなく実行できることにご留意ください）。もし **TMCL™** プログラムが定期的にこの変数をポーリングした場合、プログラムは変数の内容への変更に反応します。

ホストはダイレクトモードで **GGP** を使用しても変数をポーリングでき、それが **TMCL™** プログラムにより変更されたかどうか、確認できます。

8.5 サブルーチンの使用

CSUB コマンドと **RSUB** コマンドはサブルーチンを使用するためのメカニズムを提供します。 **CSUB** コマンドは指定されたラベルに枝分かれします。**RSUB** コマンドが実行されると、制御はサブルーチンを呼び出した **CSUB** コマンドの後続のコマンドに戻ります。

このメカニズムはネスティングすることもできます。 **CSUB** コマンドにより呼び出されたサブルーチンから、他のサブルーチンを呼び出すことができます。 **TMCL™** の現行バージョンでは、ネスティングされたサブルーチン呼び出しは八つのレベルまで可能です。

8.6 ダイレクトモードとスタンドアロンモードの混合

ダイレクトモードとスタンドアロンモードは混合することもできます。TMCL™プログラムがスタンドアロンモードで実行されているときは、ダイレクトモードコマンドも処理されています（スタンドアロンモードで実行されているプログラムの流れは妨げません）。そのため、TMCL™プログラムの実行中に、ダイレクトモードのモータの実位置などをクエリすることもできます。

スタンドアロンモードで実行中のプログラムとホストとの通信は、TMCL™ユーザ変数を使用して行うことができます。そこでは、ホストはTMCL™プログラムにより定期的にポーリングされる（メインループ内などで）ユーザ変数の値を変更でき（ダイレクトモードSGP コマンドを使用して）、そのためTMCL™プログラムはそのような変更にも反応できます。逆に、TMCL™プログラムはホストにより（ダイレクトモードGGP コマンドを使用して）ポーリングされているユーザ変数を変更できます。

TMCL™プログラムは、ホストによりダイレクトモードで実行コマンドを使用して起動できます。このように、ホストに呼び出されるTMCL™ルーチンのセットも定義できます。この場合、特定のルーチンにジャンプするTMCL™の先頭にはJA コマンドを挿入することをお勧めします。これにより、たとえTMCL™ルーチンが変更されても、（TMCL™ルーチンの変更時にホストプログラムを変更する必要があるように）ルーチンのエントリアドレスが変更されないようにできます。

例：

```
//Jump commands to the TMCL™ routines
```

```
Func1:JA Func1Start
```

```
Func2:JA Func2Start
```

```
Func3:JA Func3Start
```

```
Func1Start:MVP ABS, 0, 1000
```

```
WAIT POS, 0, 0
```

```
MVP ABS, 0, 0
```

```
WAIT POS, 0, 0
```

```
STOP
```

```
Func2Start:ROL 0, 500
```

```
WAIT TICKS, 0, 100
```

```
MST 0
```

```
STOP
```

```
Func3Start:
```

```
ROR 0, 1000
```

```
WAIT TICKS, 0, 700
```

```
MST 0
```

```
STOP
```

本例は、三つの非常にシンプルなTMCL™ルーチンです。これらは、ホストが実行コマンドを発行することで呼び出すことができ、アドレスが0の場合は最初の機能、アドレスが1の場合は2番目の機能、アドレスが2の場合は3番目の機能を呼び出します。TMCL-IDEの記号ファイル生成機能を使用することで、TMCL™ラベル（コマンドの実行に必要な）のアドレスを確認することができます。

TMCL-IDEに関する詳細な情報は、TMCL-IDE取扱説明書を参照してください。

9 製品ライフサポートポリシー

TRINAMIC Motion Control GmbH & Co. KG は、TRINAMIC Motion Control GmbH & Co. KG の書面による同意無しに、同社のいかなる製品もライフサポートシステム上で使用することを許可も保証もいたしません。

ライフサポートシステムは生命を援助または維持するための装置であり、記載された指示通りに適切に使用したうえでの性能障害は、傷害または死亡事故を引き起こすことが合理的に予想されます。



© TRINAMIC Motion Control GmbH & Co. KG 2014

本データシート情報は正確かつ信頼性があるものと考えられています。しかし、その使用の結果、または本シートの使用の結果生じる第三者の特許およびその他の権利のいかなる侵害に関しても、弊社は責任を負いかねます。

仕様は予告なしに変更されることがあります。

10 変更履歴

10.1 ファームウェア変更

バージョン	日付	作者 OK - Olav Kahlbaum	内容
1.33	2014 年 4 月 29 日	OK	新ハードウェアバージョン V1.1 の初版

10.2 文書改訂

バージョン	日付	作者	内容
0.90	2014 年 5 月 27 日	GE	TMCM-1110 stepRocker TMCL™ ファームウェア説明書をベースにした初版
0.91	2014 年 7 月 9 日	DWI	CAN 情報の省略

11 参考文献

[MCST3601] MCST3601 ハードウェア説明書

[TMC262] TMC262 データシート

[TMC429] TMC429 データシート

[TMCL-IDE] TMCL-IDE ユーザ説明書

www.trinamic.com をご覧ください。

Memo:

〒140-0013

東京都品川区南大井6-20-8

マルイト大森第2ビル8F

新光電子株式会社

TEL.03-6404-1003

FAX.03-6404-1005

e-mail. motor-info@shinkoh-elecs.co.jp

www.shinkoh-faulhaber.jp

7000.55002 English, 1st edition, 10.2014

© FAULHABER PRECISTEP SA

仕様は予告なしに変更されることがあります。

FAULHABER PRECISTEP SA