

AC サーボモータ用 ドライバ  
TAD881x(RoHS 指令対応品)  
簡易コントロール操作マニュアル

DS' S

CH' D

APP' D

# Motortronics®

---



AC サーボモータ用ドライバ  
TAD881x シリーズ

簡易コントロール機能

---

RoHS 指令対応品

 TAMAGAWA SEIKI CO.,LTD

---

## 1. はじめに

本資料はTAD881×シリーズの簡易コントロール機能についてまとめています。

初めて簡易コントロール機能を使う場合の初期設定方法など、初回導入時の基本的な使い方を中心に説明しています。

簡易コントロール機能に使用するI/Oコネクタやケーブル配線についての詳細は取扱説明書(MNL000404W00)を参照下さい。

簡易コントロール機能命令の詳細はUSB通信仕様書(SPC009257W00)を参照下さい。

## 2. 概要

TAD8811シリーズのACサーボドライバではドライバ単体の機能として簡単な運転プログラムを設定する事が出来ます。

これを簡易コントロール機能と言います。

簡易コントロール機能では、最大128ステップのプログラムを組むことが出来、それぞれの命令では移動命令、移動中の指令変更、I/Oによる条件分岐、I/O出力、原点復帰、アラームリセット、ポジションリセット、サーボON/OFF、パラメータ変更などが行えます。

この機能を用いる事により、簡易な単軸動作であれば、PLCやコントローラを使わずに大抵の動作が可能になります。

プログラムの書き替えや、プログラム内のパラメータ変更には、USB接続、SV-NET通信、RS232通信、RS485通信を利用します。

無料のPCアプリケーションソフト「Motion Adjuster」からプログラミングする事も可能です。

本資料では主に「Motion Adjuster」を使用して簡易コントロール機能を使用する方法を解説しています。

### 3. 簡易コントロール機能のしくみ

簡易コントロール機能は、あらかじめプログラムした動作を実行しますが、そのプログラムを実行するためにはまずパラメータ ID31(制御モード)を”14”に設定しておき、且つ I / O 入力や通信によるサーボコマンドでサーボ ON 指令を入力する必要があります。

通常の制御モードと異なり、サーボ ON 入力は実際にモータをサーボ ON する為の入力としては使われない事にご注意下さい。

この時のサーボ ON 入力は「プログラムスタート指令」として扱われます。

またプログラム実行中でもサーボ ON 入力を OFF する事により、プログラムを中断しサーボ OFF する事が出来ます。

その概念を図 1 に示します。

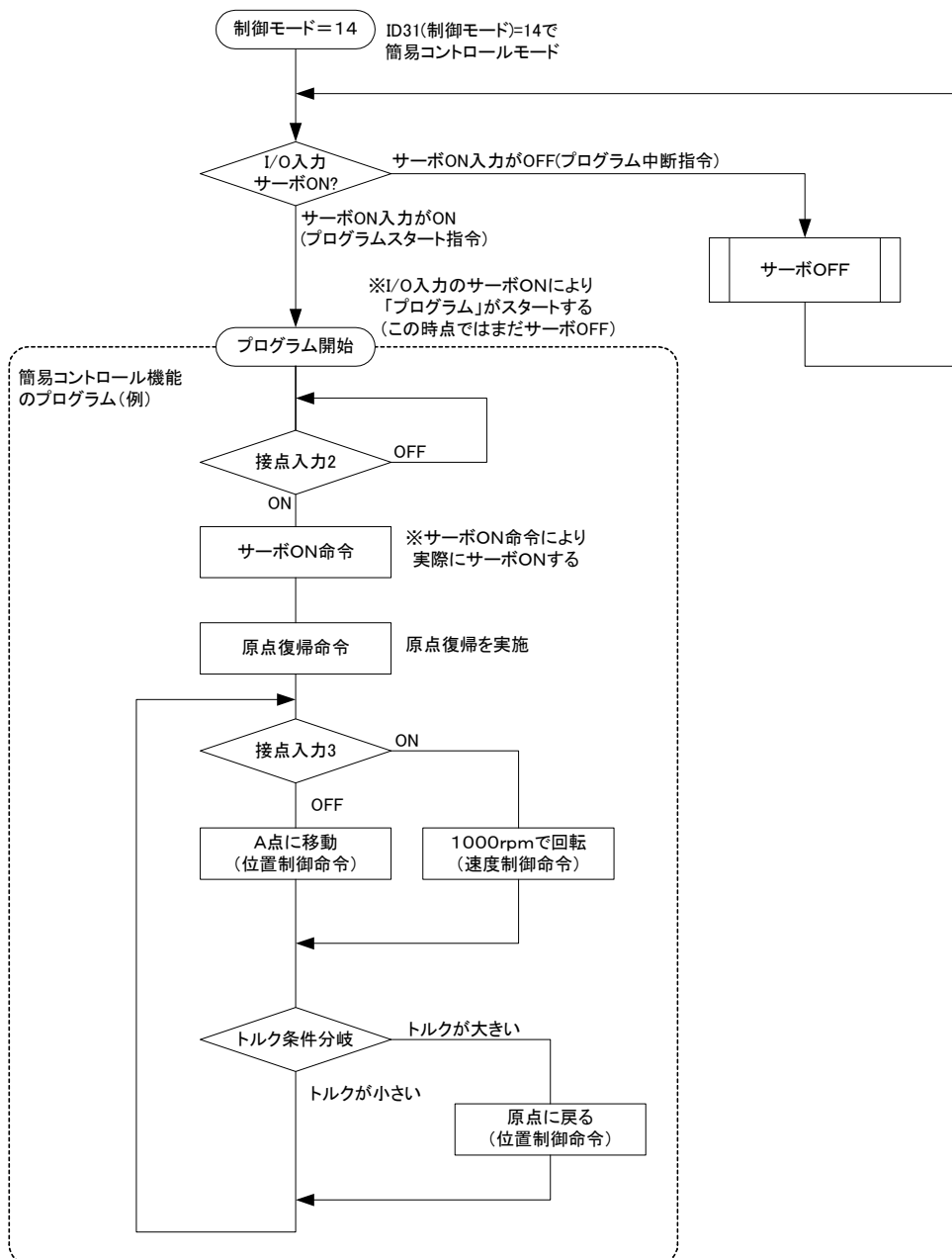


図 1 簡易コントロールモードの概念図

I/O入力によるサーボON入力とは、ID100~107(I/O 入力 1~8)の設定を“サーボオン指令”とした入力の事です (ID100 の場合は”0”or”1”、ID101~107 の場合は”1”と設定した入力)。

例1) I/O 入力 1 (IN1)をプログラムスタート指令に設定する場合  
ID100(I/O 入力 1 の設定) を” 0x00 (0)”に設定する

例2) I/O 入力 3 (IN3)をプログラムスタート指令に設定する場合  
ID102(I/O 入力 3 の設定) を” 0x01 (1)”に設定する

プログラムスタート指令として I/O を使用したくない場合は、ID100~107 の何れかをサーボオン指令に設定した上で負論理の設定(Bit7=1)とし常時オンの状態にします。

例1) I/O 入力 1 (IN1)をプログラムスタート指令 (常時 ON) に設定する場合  
ID100(I/O 入力 1 の設定) を” 0x80 (128)”に設定する

例2) I/O 入力 3 (IN3)をプログラムスタート指令 (常時 ON) に設定する場合  
ID102(I/O 入力 3 の設定) を” 0x81 (129)”に設定する

※この設定の場合はドライバが電源ONすると自動的にプログラムが実行されますので、非常停止等のサーボOFF処理はプログラム内で準備する必要があります。

上記の通り簡易コントロール機能では、プログラムスタート指令として通常必ず I/O 入力を 1 チャンネル占有します。

TAD 8 8 1 1 シリーズでは I/O 入力が 8 チャンネルありますが、プログラム中で使用できるのはプログラムスタート指令の 1 チャンネルを引いた 7 チャンネルとなります。

I/O 入力を使わない他の方法として、I/O の代わりにパラメータ ID30 (サーボコマンド) の Bit0 を “1” にセット (通信によるサーボON) する方法もあります。

この場合も、サーボON指令の代わりにプログラムスタート指令がONとなります。

この方法の場合は SV-NET,RS232C,RS485 などの通信を使って簡易コントロールプログラムをスタートさせる事になります。

## 4. I/O設定

簡易コントロール機能上のプログラムでI/O入出力を使用する場合は、I/Oを簡易コントロール機能の接点として使用できるように、あらかじめ設定しておく必要があります。

### 4.1 入力接点の設定

プログラム上で入力接点を使用する場合は、該当するI/O入力を「簡易コントロールプログラム入力」に設定する必要があります。

使用するI/O入りに該当するID100~107(I/O入力1~8)の設定を”0x0F”と設定する事で、そのI/O入力は簡易コントロール機能のプログラム上で、接点入力として使用できます。

例) IN5をプログラム上で接点入力5として使用する場合  
ID104(I/O入力5の設定)を”0x0F(15)”に設定する

また、I/O入力設定を”0x0F”以外に設定した場合の機能も、いくつかの機能は簡易コントロールプログラム動作中に使うことができます。

#### ※簡易コントロール機能中でも使用できるI/O入力設定

- 2:正方向リミット指令
- 3:負方向リミット指令
- 4:アラームリセット指令
- 7:原点センサ入力
- 8:外部アラーム入力
- 9:ゲイン切り替え指令
- 10 (A):アナログ入力ゼロ点調整指令
- 11 (B):第2電流リミット切り替え指令
- 14 (E):アナログ入力指令の強制0指令
- 15 (F):簡易コントロールプログラム入力1
- 17 (11):ハードストップ
- 18 (12):スムーズストップ
- 99 (63):入力無視

注意：上記の機能は簡易コントロールプログラムとは無関係に実行されますので、プログラム上の動作と矛盾が生じないように使用する必要があります。

### 4.2 出力接点の設定

プログラム上で出力接点を使用する場合は、該当するI/O出力を「簡易コントロールプログラム出力」に設定する必要があります。

使用するI/O出力に該当するID110~114(I/O出力1~5)の設定を”0xFFFFFFFF”と設定する事で、そのI/O出力は簡易コントロール機能のプログラム上で、接点出力として使用できます。

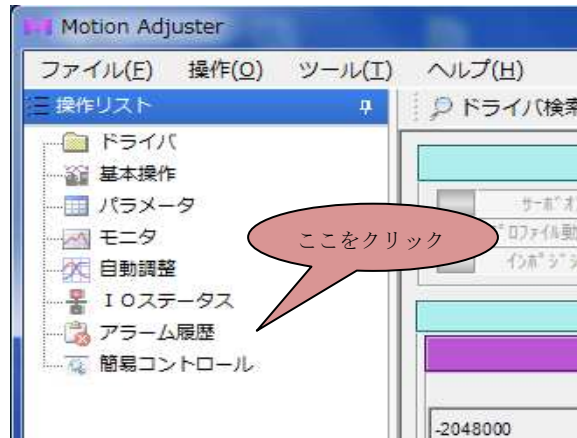
例) OUT3をプログラム上で接点出力3として使用する場合  
ID112(I/O出力3の設定)を”0xFFFFFFFF”に設定する

I/O出力を0xFFFFFFFFと設定しない場合は、簡易コントロールプログラム動作中でも、通常のスレータス出力として使用できます。

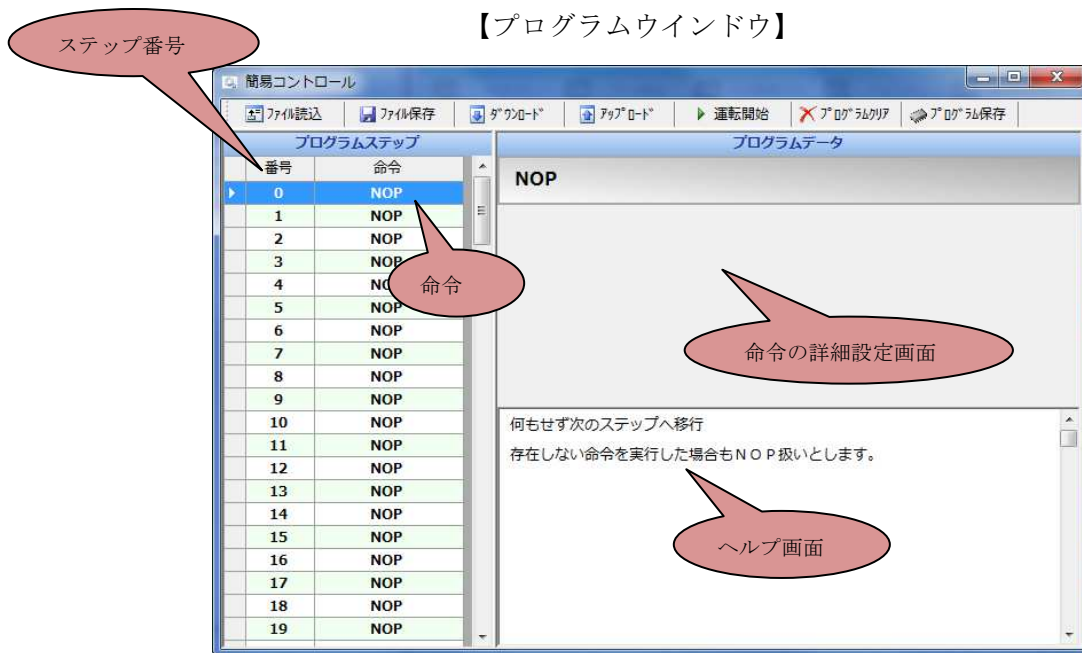
## 5. プログラムウインドウ

簡易コントロール機能では最大128ステップのプログラムを作成できます。

「Motion Adjuster」上でプログラムを作成する場合、まず画面左側の操作リスト上から「簡易コントロール」をクリックし、簡易コントロール機能のプログラムウインドウを表示します。

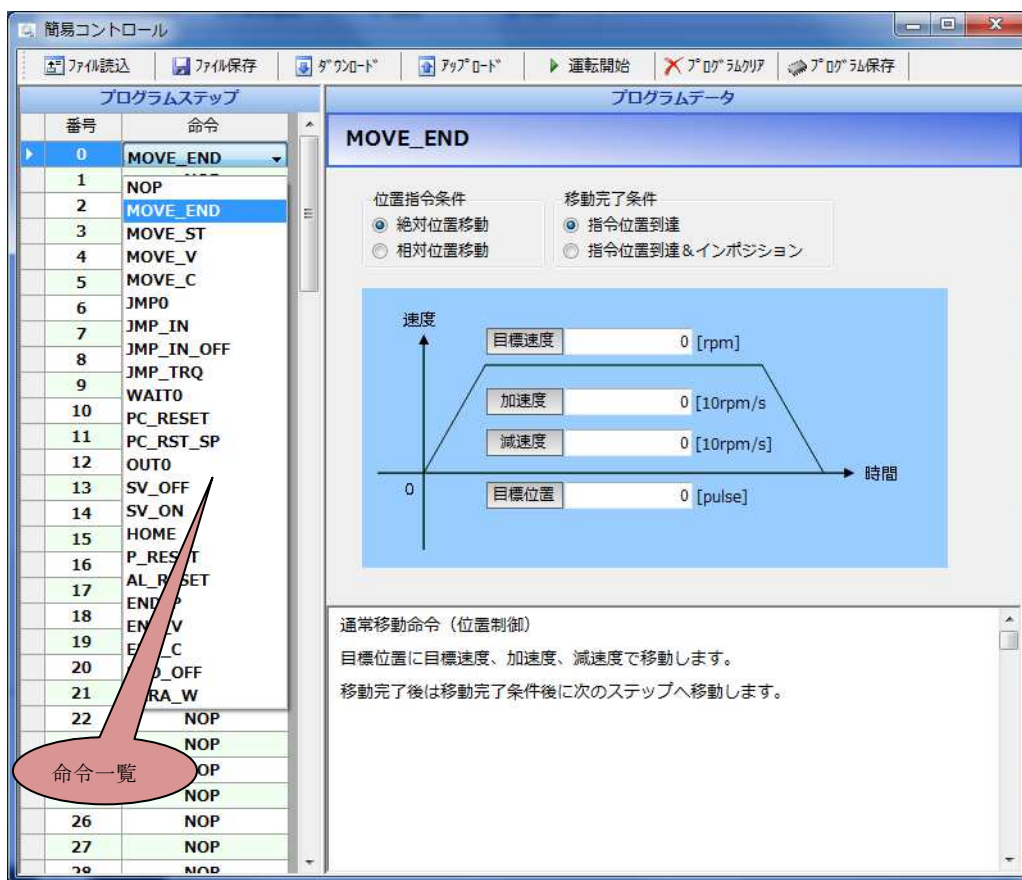


【プログラムウインドウ】



プログラムウインドウ左側のプログラムステップ欄にプログラムを作成していきます。プログラムはステップ0から実行されますので、通常はステップ0からプログラムを作成していきます。右側のプログラムデータ画面では、各命令のパラメータ等の詳細設定を行います。プログラムデータ画面の下は、選択した命令の説明が表示されるヘルプ画面になっています。

設定したいステップの命令欄をクリックすると、設定できる全命令の一覧が表示されます。



表示されている命令の内、一つを選択すると、右側のプログラムデータ画面に命令の詳細が表示されます。

上記は通常移動命令（MOVE\_END）を選択した場合の画面です。

選択した命令にパラメータがある場合は、プログラムデータ画面で設定します。

ステップ番号の周辺にマウスポインタを置いて右クリックをすると、プログラム編集サブウィンドウが開きます。一般的な表計算ソフトと同じように、ステップ毎に“コピー”、“切り取り”、“貼り付け”、“行挿入”、“行削除”、“元に戻す”、“やり直し”が行えます。但し編集できるのは1行づつになります（数行まとめて挿入等は出来ません）。

“ダウンロード”、“アップロード”は後述する編集済みのプログラムをドライバに書き込み、読み込みする機能です。





プログラムウインドウ上部にあるボタンはそれぞれ以下の通りの機能になります。



#### ファイル読込：

既に作成済みのプログラムのファイルを読み込みます。  
プログラムファイルは拡張子「.prg」のファイルです。

#### ファイル保存：

プログラムウインドウ上で作成されたプログラムをファイルに保存します。  
任意の名前で保存すると、拡張子「.prg」のプログラムファイルが生成されます。

#### ダウンロード：

プログラムウインドウ上に表示されているプログラムをドライバのRAM上に書き込みます。  
この操作はドライバと MotionAdjuster とのUSB通信が接続されている時に実行できます。  
ダウンロードしただけでは、ドライバの電源をOFFするとプログラムは消えてしまいます。  
プログラムを保存しておく場合は「プログラム保存」を行い、不揮発メモリに保存する必要があります。

#### アップロード：

ドライバのRAM上のプログラムを読み出し、プログラムウインドウ上に展開します。

#### 運転開始：

プログラム画面上でプログラムを実行する試運転機能です。  
この機能ではプログラムスタート指令を無視してプログラムが実行されます。  
試運転中は、実行中のステップが黄色で表示されます。

注意) モータを制御しますので、安全に注意して実行して下さい

#### プログラムクリア：

ドライバの不揮発メモリ上に保存されているプログラムを全てクリアします。  
過去に作成したプログラムが勝手に動かないようにしたい場合等に行ってください。

---

### プログラム保存：

ドライバのRAM上のプログラムを不揮発メモリに保存します。

不揮発メモリに保存しておくことで、ドライバの電源をOFFしても、次回ON時にプログラムが自動的にRAM上に展開されプログラムを実行する事が出来ます。

プログラム保存は必ずドライバがサーボOFFしている事を確認してから行って下さい。

サーボオン中にプログラム保存するとドライバが誤動作する可能性があります。

**注意) プログラム保存は必ずサーボオフ中に行ってください**

## 6. 命令

簡易コントロール機能のプログラムでは、以下の命令が用意されています。

位置制御命令：MOVE\_END、MOVE\_ST  
速度制御命令：MOVE\_V  
電流制御命令：MOVE\_C  
分岐命令：JMP0、JMP\_IN、JMP\_IN\_OFF、JMP\_TRQ  
待機命令：WAIT0  
カウンタクリア命令：PC\_RESET、PC\_RST\_SP  
接点出力命令：OUT  
サーボコマンド命令：SVON、SVOFF、P\_RESET、AL\_RESET  
原点復帰命令：HOME  
完了命令：END\_P、END\_V、END\_C、END\_OFF  
パラメータ変更命令：PARA\_W  
その他：NOP

それぞれの命令の詳細は以下の通りです。

### 6. 1 MOVE\_END (通常移動命令[位置制御])

指定した目標位置に移動する命令です。

本命令では目標位置に到達してから次のステップに進みます。

プログラムデータ画面上の台形図で、「目標位置」「目標速度」「加速度」「減速度」を設定します。

位置指令の条件は、「絶対位置移動」と「相対位置移動」を選択できます。

絶対位置移動では設定した目標位置に移動します。

相対位置移動では、現在の位置に、目標位置設定値を加算した位置に移動します。



次のステップへ進む条件としては目標位置に到達して移動完了となる必要があります。

その移動完了の判断条件として「指令位置到達」と「指令位置到達&インポジション」を選択できます。

指令位置到達では、実際のモータ軸位置が到達していなくても、ドライバ内部の位置指令が到達していれば移動完了と判断して次のステップに進みます。

指令位置到達&インポジションでは、ドライバ内部の指令位置が到達しているだけでなく、実際のモータ軸位置も目標位置に到達している場合に移動完了と判断して次のステップに進みます

実際のモータ軸位置が到達しているという判断は、パラメータ ID77 (インポジション[位置決め完了]信号範囲) によって調整出来ます。

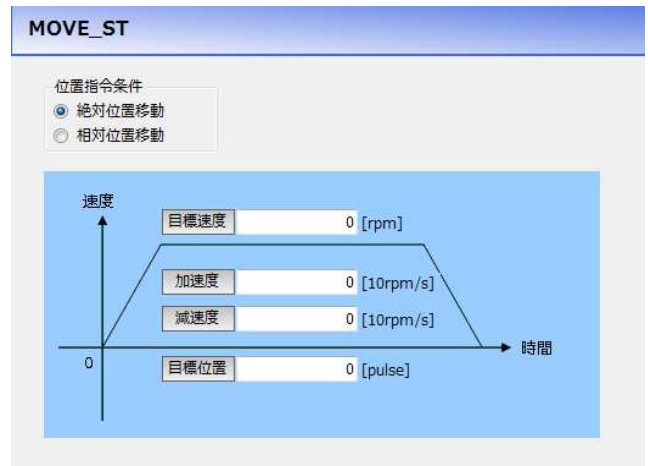
指令位置と現在位置の偏差が ID77 の設定範囲に入ると位置決め完了 (インポジション) となります。

## 6. 2 MOVE\_ST (移動開始命令[位置制御])

指定した目標位置に移動する命令です。  
プログラムデータ画面上の台形図で、「目標位置」  
「目標速度」「加速度」「減速度」を設定します。

位置指令の条件は、「絶対位置移動」と「相対位置移動」を選択できます。

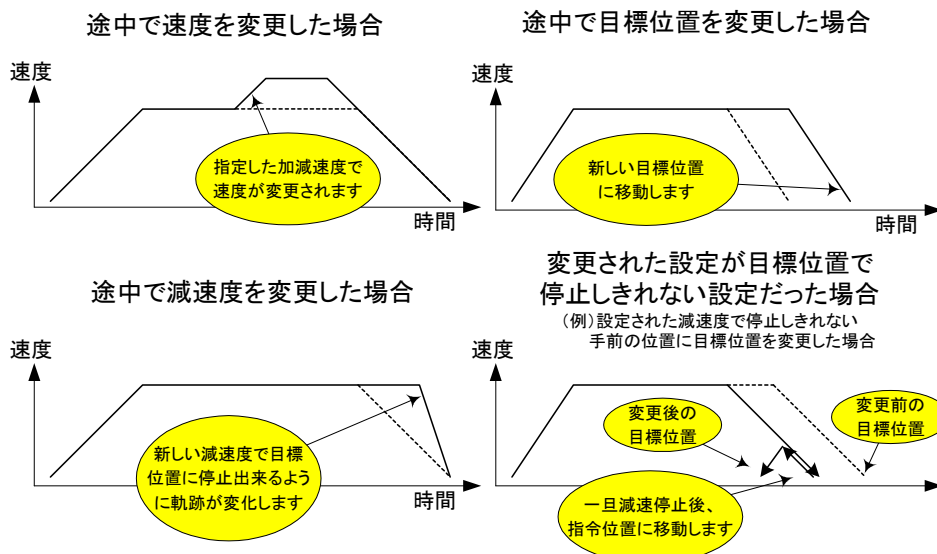
絶対位置移動では設定した目標位置に移動します。  
相対位置移動では、現在の位置に、目標位置設定値を加算した位置に移動します。



MOVE\_END とは異なり、本命令では命令実行後（移動開始時）に次のステップに進みます。  
移動中に条件分岐などで動作を変更したい場合はこちらの命令を使用します。

MOVE\_ST 命令を実行中に別の MOVE\_END、MOVE\_ST 命令を実行する事により、移動中でも新しい目標位置、目標速度、加速度、減速度に連続して変更できます。

但し移動中に目標位置や速度等を変更した場合、減速停止位置が目標位置を超えてしまう場合がありますが、その場合は減速停止後、指定した加速度・減速度で速やかに目標位置に戻る動作になります。



### 【ワンポイント】

MOVE\_END、MOVE\_ST 命令では、パラメータ ID78/79 (スムージング時間 1/2) の機能を併用することが出来ます。

スムージング時間を設定する事により、位置移動中の速度軌跡を通常台形から、S字にすることが出来ます。

### 6. 3 MOVE\_V (速度制御命令)

指定した速度で制御する命令です。

プログラムデータ画面上的の台形図で、「目標速度」

「加速度」「減速度」を設定します。

本命例は命令実行後（加速開始時）に次のステップに進みますので、速度制御中に条件分岐等で動作を変更することが出来ます。

また速度指令はアナログ入力指令を使うことも出来ます。

速度指令選択の条件を「パラメータ指令」に設定する

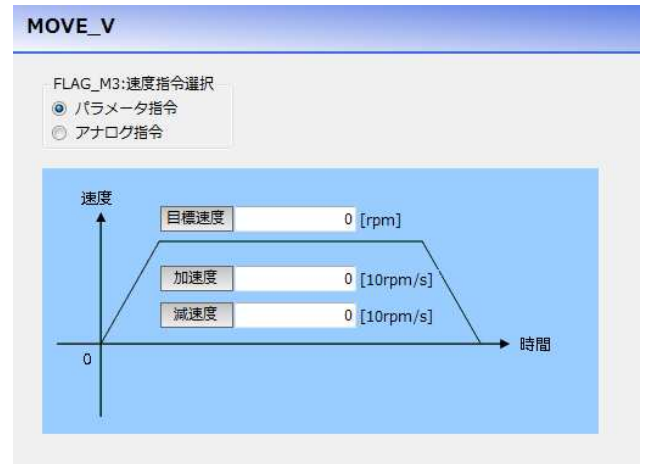
と、プログラムデータ画面上で指定する「目標速度」「加速度」「減速度」で制御します。

速度指令選択の条件を「アナログ指令」に設定すると、ドライバのアナログ入力指令で制御します。

アナログ指令の場合は画面上的の「加速度」「減速度」設定も無視され、アナログ指令の軌跡に沿って速度制御されます。

アナログ指令を使う場合の速度指令スケールは、パラメータ ID130（アナログ入力信号の速度換算スケール）に設定します。

またパラメータ ID133（アナログ入力ゼロクランプ）、ID134（アナログ入力フィルタ）の機能も使用することが出来ます。



### 6. 4 MOVE\_C (電流制御命令)

指定した電流で制御する命令です。

プログラムデータ画面上で「目標電流」を設定します。

本命例は命令実行後に次のステップに進みますので、電流制御中に条件分岐等で動作を変更することが出来ます。

また電流指令はアナログ入力指令を使うことも出来ます。

電流指令選択の条件を「パラメータ指令」に設定すると、プログラムデータ画面上で指定する「目標電流」で制御します。

電流指令選択の条件を「アナログ指令」に設定すると、ドライバのアナログ入力指令で制御します。

アナログ指令を使う場合の電流指令スケールは、パラメータ ID131（アナログ入力信号の電流換算スケール）に設定します。

またパラメータ ID133（アナログ入力ゼロクランプ）、ID134（アナログ入力フィルタ）の機能も使用することが出来ます。



## 6.5 JMP0 (無条件分岐命令)

別のステップに分岐する命令です。

「分岐先ステップ」に分岐したいステップ番号を設定します。

「待機時間」に数値を設定すると分岐前に待機時間が入ります。

待機時間中は次のステップにも分岐先ステップにも進まず、  
本命令のステップで待機します。

「繰り返し回数」に分岐先ステップに分岐する回数を設定できます。

例えば同じ処理を10回繰り返したい場合には、「繰り返し回数」に10回を設定すると、このJMP0命令で10回分岐先ステップに分岐した後、11回目で次のステップに進みます。

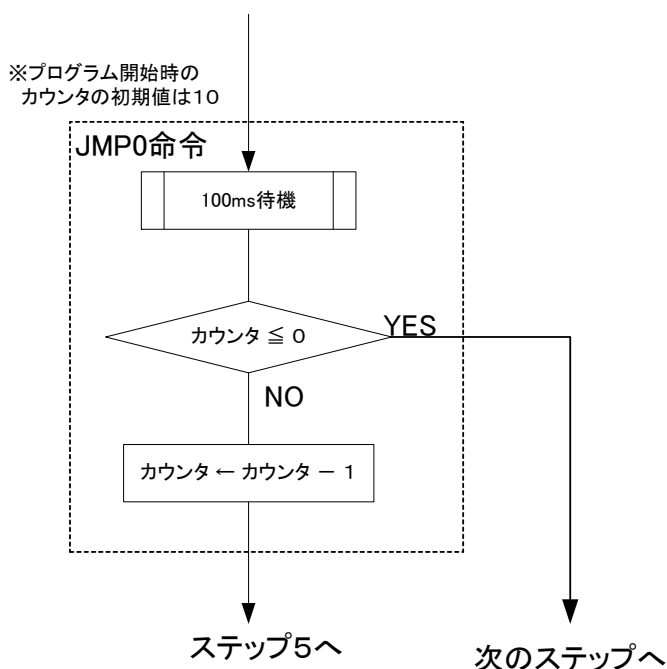
「繰り返し回数」に0を設定すると無限回になります。

繰り返しの残り回数は、プログラム実行中は自動的に初期化されませんので、全ての繰り返し回数を完了したJMP0命令ステップには、その後何度入っても分岐先ステップには分岐せず、次のステップに進む動きになります。

繰り返し回数を初期化したい場合は後述するPC\_RESET、PC\_RST\_SP命令を実行下さい。

JMP0	
分岐先ステップ	0
待機時間	0 [ms]
繰り返し回数	0 [回]

例) 分岐先ステップ=5、待機時間100ms、繰り返し回数10回に設定した場合の動作フロー



## 6. 6 JMP\_IN (待機判断&条件分岐命令)

指定した接点入力が ON した場合に、別のステップに分岐する命令です。

「分岐先ステップ」に分岐したいステップ番号を設定します。  
「接点入力番号」に判定用の接点入力番号 (1~8) を設定します。

もし接点入力が OFF だった場合、待機するか次のステップに進むのかは MODE\_J1 (待機条件) で設定できます。

「無条件待機」に設定した場合は、接点入力が OFF の間は無条件で現在ステップで待機し続けます。

「指令位置到達」に設定した場合は、過去に実行した MOVE\_ST 命令で、位置指令が目標位置に到達した場合は次のステップに進みます。

また MOVE\_ST 命令を実行していなければ無条件で次のステップに進みます。

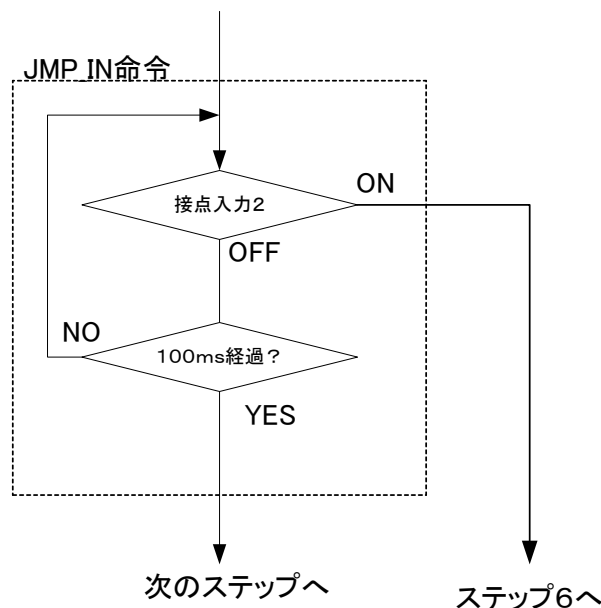
「指令位置到達&インポジション」に設定した場合は、過去に実行した MOVE\_ST 命令で、位置指令が目標位置に到達し、且つ実際のモータ軸位置も目標位置に到達している場合に次のステップに進みます。

また MOVE\_ST 命令を実行していなければ無条件で次のステップに進みます。

「時間待機」に設定した場合は、設定時間だけ待機し、待機時間を超えたら次のステップに進みます。

「待機時間」は MODE\_J1 (待機条件) を “時間待機” に設定した場合の待機時間を設定します。

例) 待機条件=時間待機、分岐先ステップ=6、接点入力番号2、待機時間100ms、に設定した場合の動作フロー



## 6. 7 JMP\_IN\_OFF (待機判断&条件分岐命令[OFF論理])

指定した接点入力が OFF した場合に、別のステップに分岐する命令です。

JMP\_IN との違いは接点論理が反転しているだけです。

「分岐先ステップ」に分岐したいステップ番号を設定します。

「接点入力番号」に判定用の接点入力番号 (1~8) を設定します。

もし接点入力が ON だった場合、待機するか次のステップに進むのかは MODE\_J2 (待機条件) で設定できます。

「無条件待機」に設定した場合は、接点入力が ON の間は無条件で現在ステップで待機し続けます。

「指令位置到達」に設定した場合は、過去に実行した MOVE\_ST 命令で、位置指令が目標位置に到達した場合は次のステップに進みます。

また MOVE\_ST 命令を実行していなければ無条件で次のステップに進みます。

「指令位置到達&インポジション」に設定した場合は、過去に実行した MOVE\_ST 命令で、位置指令が目標位置に到達し、且つ実際のモータ軸位置も目標位置に到達している場合に次のステップに進みます。

また MOVE\_ST 命令を実行していなければ無条件で次のステップに進みます。

「時間待機」に設定した場合は、設定時間だけ待機し、待機時間を超えたら次のステップに進みます。

「待機時間」は MODE\_J2 (待機条件) を “時間待機” に設定した場合の待機時間を設定します。

### JMP\_IN\_OFF

MODE\_J2

無条件待機

指令位置到達

指令位置到達&インポジション

時間待機

分岐先ステップ

接点入力番号

待機時間  [ms]

## 6. 8 JMP\_TRQ (トルクモニタ分岐)

モータが出しているトルク (電流フィードバック値) が、指定した条件に達した場合に分岐する命令です。

モータに掛かっている負荷を監視して動作を変更する場合などに使います。

「分岐先ステップ」に分岐したいステップ番号を設定します。

「トルク (電流) 閾値」に、判断条件の電流値を設定します。分岐論理の設定で「閾値以上」に設定すると、モータ電流が設定値より大きい場合に分岐します。

「閾値以下」に設定した場合は、モータ電流が設定値より小さい場合に分岐します。

### JMP\_TRQ

MODE\_J3

無条件待機

指令位置到達

指令位置到達&インポジション

時間待機

分岐先ステップ

トルク(電流)閾値  [0.01Arms]

待機時間  [ms]

分岐論理

閾値以上

閾値以下

もしトルク (電流) 閾値判定に達していない場合、待機するか次のステップに進むのかを MODE\_J3 (待機条件) で設定します。

「無条件待機」に設定した場合は無条件で現在ステップで待機し続けます。

「指令位置到達」に設定した場合は、過去に実行した MOVE\_ST 命令で、位置指令が目標位置に到達した場合は次のステップに進みます。

また MOVE\_ST 命令を実行していなければ無条件で次のステップに進みます。



「指令位置到達&インポジション」に設定した場合は、過去に実行した MOVE\_ST 命令で、位置指令が目標位置に到達し、且つ実際のモータ軸位置も目標位置に到達している場合に次のステップに進みます。

また MOVE\_ST 命令を実行していなければ無条件で次のステップに進みます。

「時間待機」に設定した場合は、設定時間だけ待機し、待機時間を超えたら次のステップに進みます。

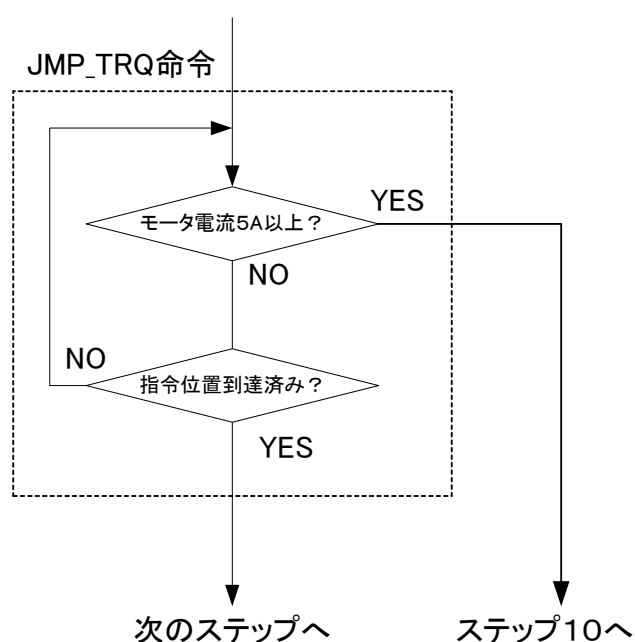
「待機時間」は MODE\_J3（待機条件）を“時間待機”に設定した場合の待機時間を設定します。

尚、トルク（電流）閾値判定に用いられる電流フィードバックは瞬時値を使用しています。

例) 待機条件=指令位置到達、分岐先ステップ=10、トルク（電流）閾値=5A、分岐論理

=

閾値以上に設定した場合の動作フロー



---

## 6. 9 JMP\_STS (ステータス判断分岐)

ID20 (サーボ状態表示) の bit 条件により分岐する命令です。

アラーム発生時やリミット検出時等を条件に分岐したい場合に使用します。

ServoStatus 判断 bit の設定で、判断したい ID20 の bit を”1”にする事で、その bit 条件で判断します。

例えばアラーム時に判断したい場合は 0x00000008 と設定します。

複数の bit を”1”にした場合は、それらの bit の OR で判断されます。

分岐論理の設定で「0」に設定すると、指定 bit が「0」の時に分岐します。

「1」に設定した場合は、指定 bit が「1」の時に分岐します。

もしステータス判断が分岐で無い場合、待機するか次のステップに進むのかを MODE\_J3 (待機条件) で設定します。

「無条件待機」に設定した場合は無条件で現在ステップで待機し続けます。

「指令位置到達」に設定した場合は、過去に実行した MOVE\_ST 命令で、位置指令が目標位置に到達した場合は次のステップに進みます。

また MOVE\_ST 命令を実行していなければ無条件で次のステップに進みます。

「指令位置到達&インポジション」に設定した場合は、過去に実行した MOVE\_ST 命令で、位置指令が目標位置に到達し、且つ実際のモータ軸位置も目標位置に到達している場合に次のステップに進みます。

また MOVE\_ST 命令を実行していなければ無条件で次のステップに進みます。

「時間待機」に設定した場合は、設定時間だけ待機し、待機時間を超えたら次のステップに進みます。

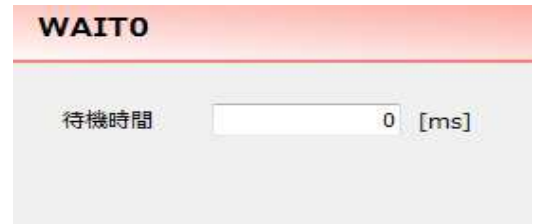
「待機時間」は MODE\_J3 (待機条件) を “時間待機” に設定した場合の待機時間を設定します。

## 6. 1 0 WAIT0 (無条件待機)

指定した待機時間だけ待機する命令です。

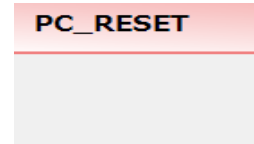
「待機時間」に設定した時間だけ、現在のステップで待機します。

待機時間を超えると次のステップに進みます。



## 6. 1 1 PC\_RESET (ジャンプ命令繰り返し回数リセット)

プログラム中で使われている JMP0 の繰り返し回数を初期化します。

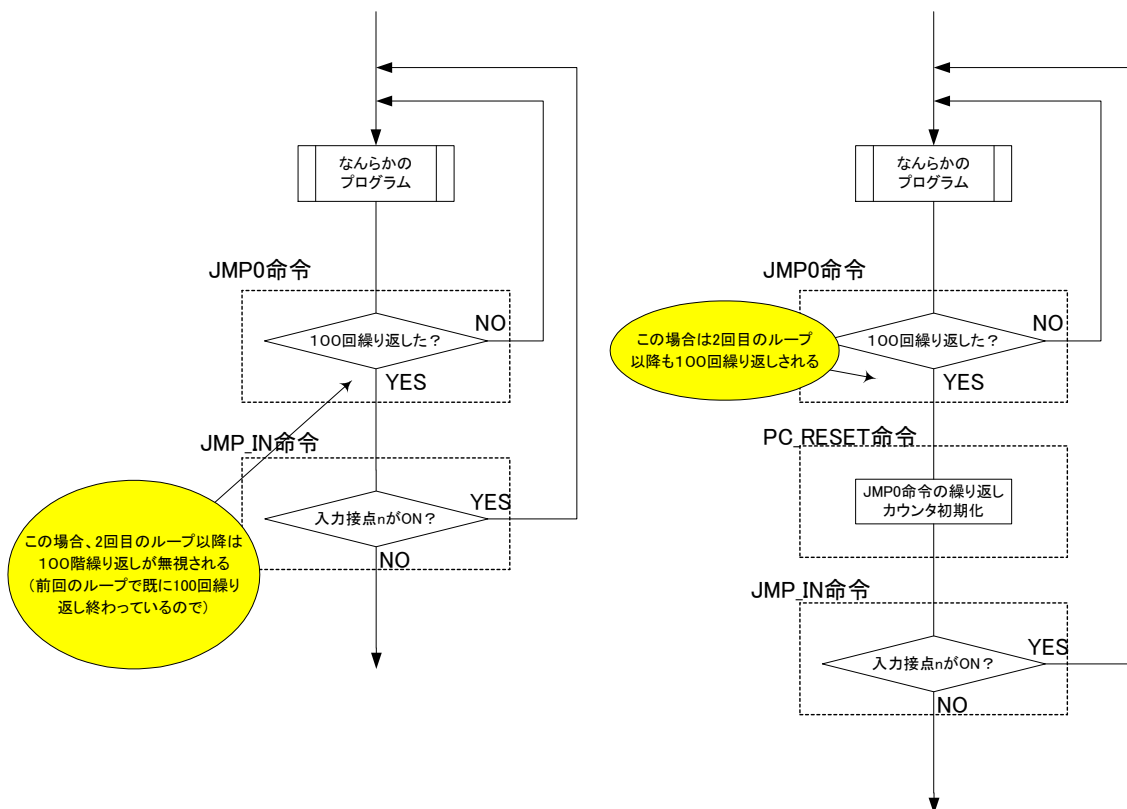


JMP0 命令の繰り返し回数を判定しているカウンタは、そのままだと減る一方で自動的に回復しません。

その為、全ての繰り返し回数が完了した JMP0 命令にもう一度処理すると、分岐せずに次のステップに移行します。

例えば 100 回処理を繰り返して JMP0 を抜けた後に、何らかの条件によりもう一度同じ処理を 100 回繰り返したい場合などは、JMP0 の繰り返し回数カウンタを初期化する必要があります。

本命令はプログラム中の全ての JMP0 命令の繰り返し回数カウンタを初期化します。

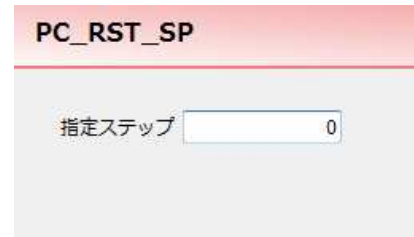


### 6. 1 2 PC\_RST\_SP (繰り返し回数指定リセット)

指定したステップの JMP0 命令の繰り返し回数を初期化します。  
「指定ステップ」に初期化したい JMP0 命令のステップを設定  
します。

PC\_RESET はプログラム中の全ての JMP0 命令を初期化するの  
に対して、PC\_RST\_SP は指定した 1 個の JMP0 命令のみ初期化  
する事が出来ます。

指定したステップが JMP0 命令でない場合は何もせず次のステップに進みます。



### 6. 1 3 OUT0 (接点出力命令)

指定した接点出力を ON/OFF します。

「接点出力番号」に、出力したい接点番号 (1~5) を設定  
します。

「出力論理」に “0” を設定すると接点出力は OFF します。

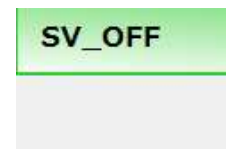
“1” を設定すると接点出力は ON します。



### 6. 1 4 SV\_OFF (サーボオフ命令)

モータをサーボ OFF して次のステップに進みます。

サーボ OFF してもプログラムスタート指令が入っている間は  
プログラムは実行されています。



### 6. 1 5 SV\_ON (サーボオン命令)

モータをサーボ ON して次のステップに進みます。

サーボ ON 時の制御モードは、直前に設定した制御モードです。  
位置制御系の制御モードでサーボ ON した場合は、位置指令 =  
現在位置 (偏差 0) でサーボ ON します。

速度・電流制御系の制御モードでサーボ ON した場合は、指令 = 0 でサーボ ON します。

本命令でサーボ ON 後、次のステップに進むまでには、自動的に 0.5 秒の待機時間が入ります。



### 6. 1 6 P\_RESET (現在位置リセット命令)

現在位置をパラメータ ID39 (ポジションリセット値) に初期化  
します。

位置制御でサーボ ON 中に本命令を行うと、自動的に一旦 0 速度

制御に切り替わり、位置リセット後に指令位置 = 現在位置 (偏差 0) にして位置制御に戻ります。



**注意) モータ回転中 (移動命令実行中) に現在位置リセットを行わないで下さい。  
モータが暴走する可能性があります。**

### 6. 1 7 AL\_RESET (アラームリセット命令)

アラームリセットを行って次のステップに進みます。

本命令はアラームリセット後、次のステップに移行するまでに自動的に0.2秒の待機時間が入ります。

AL\_RESET

注意) 過負荷アラーム (2 1)、過電流アラーム (1 1)、過回生アラーム (7 3、7 4) が発生した場合、冷却期間無しにアラームリセットを行い駆動し続けると、モータ焼損やドライバ故障の原因になります。  
アラームリセットをプログラム上で行う場合はご注意ください。

注意) アラームリセットを行ってもアラーム発生条件が解消されていない場合は再度アラームになります。  
その状態を繰り返すとアラーム履歴が同じアラームで何度も更新される事になりますので、繰り返し何度もアラームリセットを行わない様ご注意ください。

### 6. 1 8 HOME (原点復帰命令)

原点復帰動作を行います。

原点復帰はパラメータ ID90~96 の原点復帰関連の設定に従い、ID31 (制御モード) =4 (原点復帰モード) と同じ動作をします。

原点復帰が完了し、現在位置が初期化されると次のステップに移行します。

原点復帰命令を実行する前には必ずサーボ ON している必要があります。

サーボ OFF 状態で本命令を実行すると、何もせず次のステップに進みます。

HOME

原点復帰の設定や動作については TAD8811 取扱説明書 (MNL000404W00) を参照下さい。

原点復帰の種類によっては、原点復帰完了後にサーボ OFF するものがあります (突き当て方式等)。

その場合は原点復帰命令完了後もサーボ OFF されていますので、再度サーボ ON する必要があります。

### 6. 1 9 END\_P (プログラム終了[位置制御])

プログラムを終了し、ID31 (制御モード) = 1 (位置制御) に変更して簡易コントロール機能から通常的位置制御に移行します。  
サーボ ON の状態のまま通常的位置制御に移行する事も可能です。  
制御移行時の位置指令は、位置指令 = 現在位置 (偏差 0) の状態に自動的に初期化されます。

簡易コントロール機能による動作の後に、通常のパルス指令の位置制御に移行して引き続き制御したい場合等に使用できます。

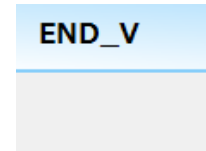
END\_P

## 6. 2 0 END\_V (プログラム終了[速度制御])

プログラムを終了し、ID31 (制御モード) = 2 (速度制御) に変更して簡易コントロール機能から通常速度制御に移行します。制御移行時の速度指令は一旦 0 になります。

サーボ ON の状態のまま通常速度制御に移行することも可能です。

簡易コントロール機能による動作の後に、通常のアナログ指令の速度制御に移行して引き続き制御したい場合等に使用できます。

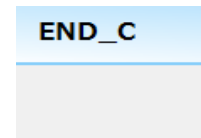


## 6. 2 1 END\_C (プログラム終了[電流制御])

プログラムを終了し、ID31 (制御モード) = 3 (電流制御) に変更して簡易コントロール機能から通常電流制御に移行します。制御移行時の電流指令は一旦 0 になります。

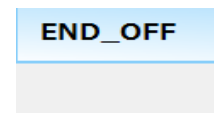
サーボ ON の状態のまま通常電流制御に移行することも可能です。

簡易コントロール機能による動作の後に、通常のアナログ指令の電流制御に移行して引き続き制御したい場合等に使用できます。



## 6. 2 2 END\_OFF (プログラム終了[サーボオフ])

プログラムを終了し、ID31 (制御モード) = 0 (サーボオフ) に変更して簡易コントロール機能から抜けます。



## 6. 2 3 PARA\_W (パラメータ変更命令)

指定したパラメータを変更し、次のステップに進みます。

「パラメータ ID」に変更したいパラメータの ID を設定します。

「書き込みデータ」に変更する値を 10 進数で設定します。

A screenshot of a parameter setting screen. It has an orange header with "PARA\_W". Below it are two input fields: "パラメータ ID" with a value of "0" and "書き込みデータ" with a value of "d".

PARA_W	
パラメータ ID	0
書き込みデータ	d

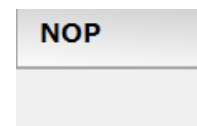
変更できないパラメータを指定した場合は何も行わず次のステップに進みます。

書き込みデータは ±2147483647 (符号付き 32bit) まで設定できますが、書き込むパラメータが 2 バイトや 1 バイトだった場合は、16 進数に変換後、上位バイトを無視して設定されます。

## 6. 2 4 NOP (無視命令)

何もせず次のステップに進みます。

プログラムを見やすくするため等で使用します。



## 7. プログラムの試運転

簡易コントロール機能のプログラムに問題ないか試すために試運転する場合は、プログラムウインドウ上部にある「ダウンロード」をクリックしてプログラムをドライバにダウンロードした後に「運転開始」をクリックします。

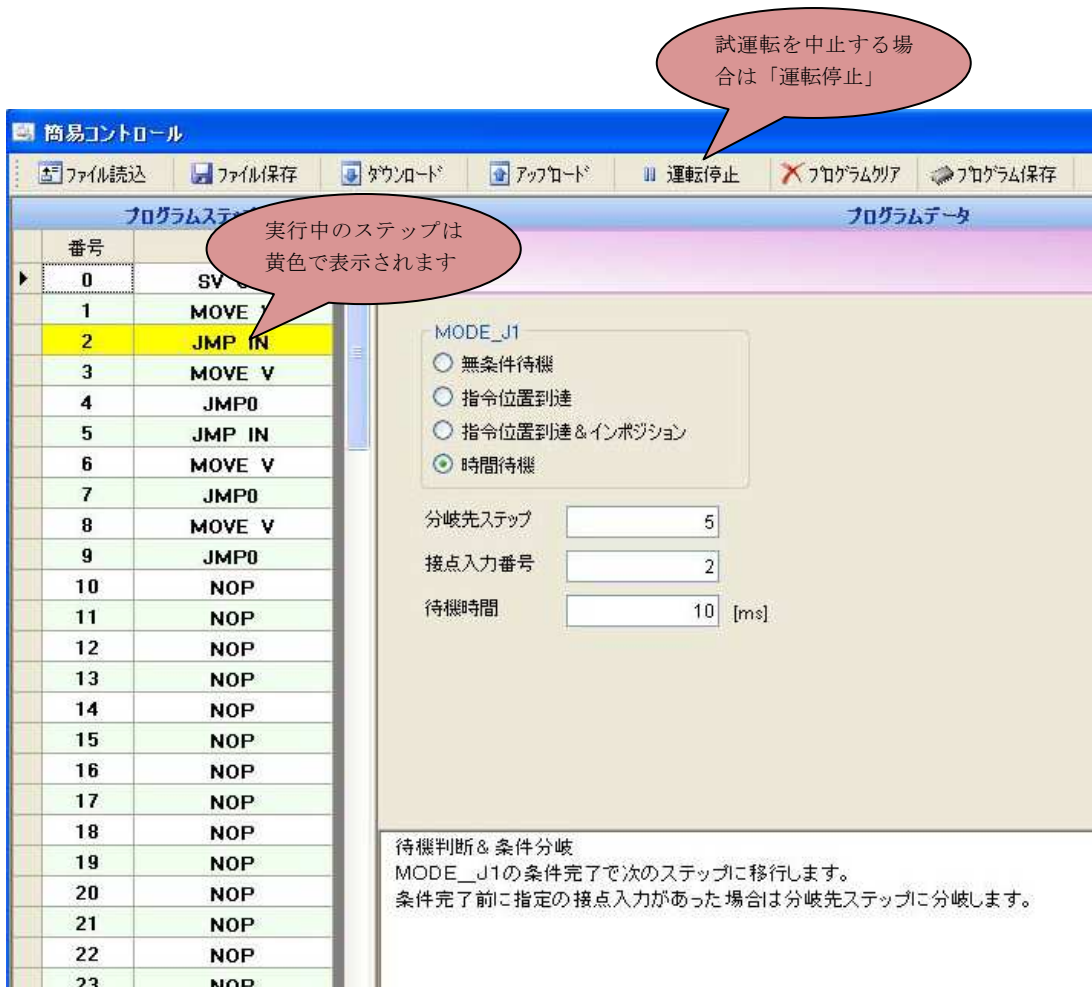


これによりプログラムを試運転させることが出来ます。

この試運転ではプログラムスタート指令がOFFでもプログラムを実行できます。

試運転中は現在実行中のステップが黄色で表示されます。

プログラムを中断する場合は「運転停止」をクリックします。

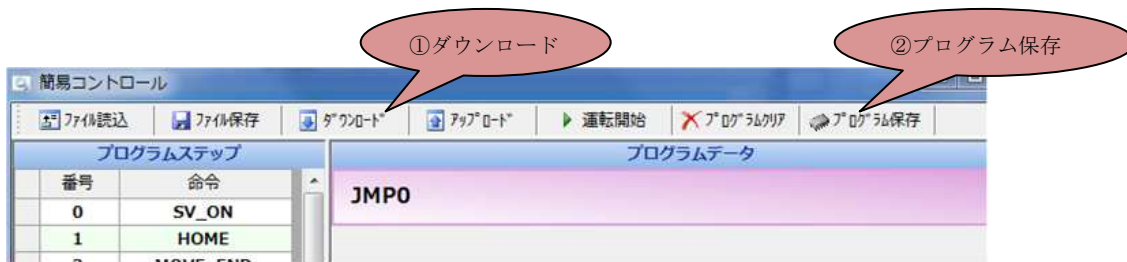


## 8. プログラム実行

簡易コントロール機能のプログラムが完成した場合は、プログラムウィンドウ上部にある「ダウンロード」をクリックします。

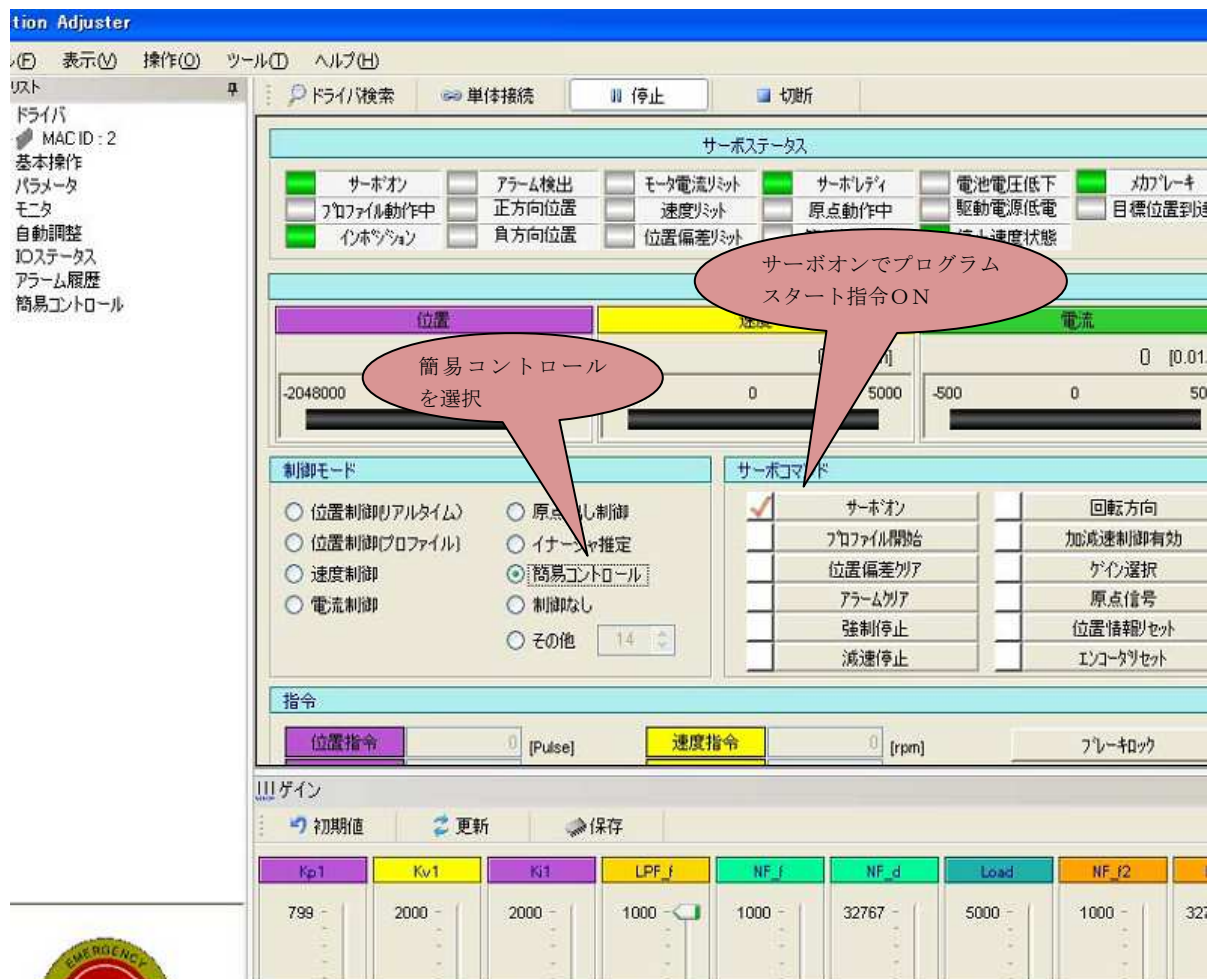
これによりプログラムがドライバにダウンロードされます。

ドライバの電源OFF→ON後もプログラムを有効にしておきたい場合はさらに「プログラム保存」をクリックし、プログラムをドライバの不揮発メモリに保存します。



その後、MotionAdjusterの基本操作画面に戻り、制御モードを「簡易コントロール」に設定し、サーボコマンドで「サーボオン」をチェックします。

これによりプログラムスタート指令がONになり、簡易コントロール機能のプログラムが実行されます。





尚、この方法は MotionAdjuster 上でプログラムを実行する場合の方法ですが、通常はパラメータ ID31（制御モード）＝”14”（簡易コントロールモード）に設定しておき、I/O 入力によりサーボ ON 指令を入れてプログラムを実行します。

I/O 入力を使わずに電源 ON で勝手にサーボ ON 指令を ON したい場合は、ID100～107 の何れかをサーボオン指令に設定した上で負論理の設定(Bit7=1)とし常時オンの状態にします。

この設定により電源 ON と同時に簡易コントロール機能のプログラムが実行されます。※3項参照

## 9. 通信によるプログラム更新

簡易コントロール機能は、MotionAdjuster によりプログラムを作成して動作させる方法の他に、ドライバに備わっている SV-NET 通信、RS 232C 通信、RS 485 通信などからプログラムを変更して動作させる事も可能です。

それぞれの通信プロトコルに関しては通信仕様書を参照下さい。

ここでは簡易コントロール機能のプログラム変更や参照がどの様に行われているか説明します。

簡易コントロールプログラムは通常のパラメータと同じ様にパラメータ ID が割り振られており、そのパラメータを読み書きする事により、プログラムの変更や参照が出来ます。

<p>ID176 : Program Pointer . . . . プログラムステップを設定します</p> <p>ID177 : Program[Step][0] . . . . ID176 で指定したステップの命令コード上位 32bit</p> <p>ID178 : Program[Step][1] . . . . ID176 で指定したステップの命令コード中位 32bit</p> <p>ID179 : Program[Step][2] . . . . ID176 で指定したステップの命令コード下位 32bit</p> <p>ID166 : ControlProgStepStatus . . 実行ステップ数等のステータス</p> <p style="padding-left: 40px;">下位 8bit=現在動作中のステップ数</p> <p style="padding-left: 40px;">12bit=全ステップ完了（127 ステップ完了）で”1”</p>
--

読み書きしたいプログラムステップのステップ数を ID176 にセットすると、ID177～179 はそのステップのプログラムデータ（命令コード）の RAM 情報が表示されます。

それを書き替えると、プログラムデータも書き換わります。

例1) ステップ10に MOVE\_END の命令コード(0x010000c1,0x00001000,0x00500050)を書き込む場合

①ID176 ← 10	(ステップ10を設定)
②ID177 ← 0x010000c1	(命令コード上位 32bit をセット)
ID178 ← 0x00001000	(命令コード中位 32bit をセット)
ID179 ← 0x00500050	(命令コード下位 32bit をセット)

例1) ステップ12に書かれている命令コードを読み出す場合

①ID176 ← 12	(ステップ12を設定)
②ID177 を読込	(命令コード上位 32bit を読込)
ID178 を読込	(命令コード中位 32bit を読込)
ID179 を読込	(命令コード下位 32bit を読込)

また、ID176 (Program Pointer) に10進数で“999”を設定すると、プログラムを不揮発メモリに保存します。

不揮発メモリへの保存が完了すると、ID176 は自動的に“0”に戻ります。

さらに ID176 (Program Pointer) に10進数で“888”を設定すると、RAM 上のプログラムと不揮発メモリ上のプログラムを全て消去します。

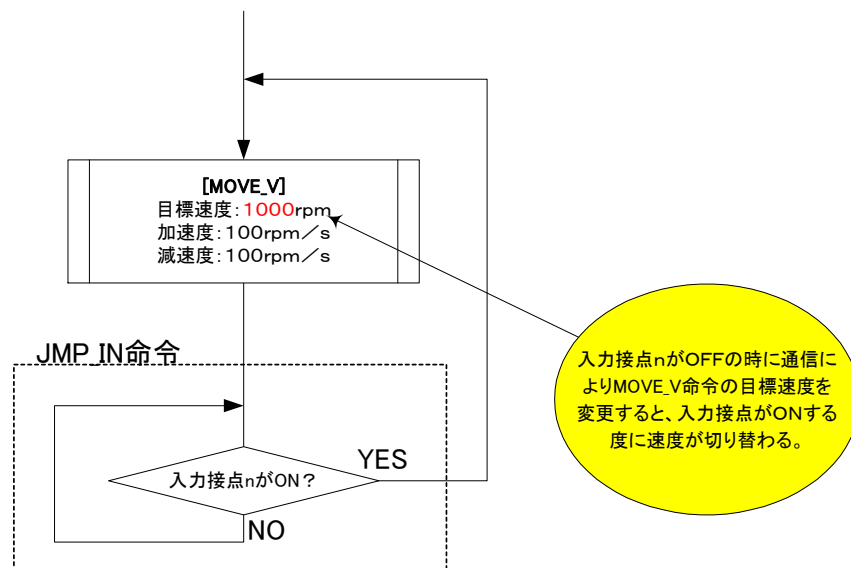
消去が完了すると ID176 は自動的に“0”に戻ります。

通信を使ってプログラムを書き替える場合、プログラム動作中に特定の命令コードを書き替える事も出来ます。

例えば、下図の様に速度制御命令の目標速度を通信により都度変更して動作させることも可能になります。

但しこの場合、確実に現在実行中でない命令のみ書き換え可能です。

実行中の命令を書き替えると、予期しない動作をする可能性があります。



10. 命令コード一覧

簡易コントロールプログラムの命令と命令コード一覧を以下に示します。

●簡易コントロール機能命令一覧表 1/2

命令	プログラムデータ0(ID:177)				プログラムデータ1(ID:178)				プログラムデータ2(ID:179)				説明	補足
	31~24bit	23~16bit	15~8bit	7~0bit	31~24bit	23~16bit	15~8bit	7~0bit	31~24bit	23~16bit	15~8bit	7~0bit		
NOP	0x00	**	**	**	**	**	**	**	**	**	**	**	何もせず次のステップへ移行 存在しない命令を実行した場合もNOP扱いとなります	
MOVE_END	0x01	FLAG_M1	目標速度[rpm]		目標位置[pulse]			加速度[10rpm/s]		減速度[10rpm/s]			通常移動命令(位置制御) 目標位置に目標速度、加速度、減速度で移動します 移動完了後はFLAG_M1の移動完了条件後に次のステップへ移行します	FLAG_M1: Bit0=位置指令条件 0:絶対値移動 1:相対値移動 Bit1=移動完了条件 0:指令位置到達 1:指令位置到達&インポジ
MOVE_ST	0x02	FLAG_M2	目標速度[rpm]		目標位置[pulse]			加速度[10rpm/s]		減速度[10rpm/s]			移動開始命令(位置制御) 目標位置に目標速度、加速度、減速度で移動開始し 次のステップへ移行します 移動中に条件分岐等を入れたい場合に使用します	FLAG_M2: Bit0=位置指令条件 0:絶対値移動 1:相対値移動
MOVE_V	0x05	FLAG_M3	目標速度[rpm]	**	**	**	**	加速度[10rpm/s]		減速度[10rpm/s]			速度制御命令 目標速度、加速度、減速度で速度制御開始し 次のステップへ移行します FLAG_M3の設定によりアナログ指令での制御も可能です アナログ指令時は加速度、減速度設定値は無視されます	FLAG_M3: Bit0=速度指令選択 0:パラメータ指令 1:アナログ指令
MOVE_C	0x06	FLAG_M4	目標電流[0.01Arms]	**	**	**	**	加速度[10rpm/s]		減速度[10rpm/s]			電流制御開始命令 目標電流で電流制御開始し次のステップへ移行します FLAG_M4の設定によりアナログ指令での制御も可能です	FLAG_M4: Bit0=電流指令選択 0:パラメータ指令 1:アナログ指令
JMP0	0x0a	**	分岐先STEP ※あり得ないSTEPの場合はEND_OFF命令となります	**	**	待機時間[ms]		(繰り返し残り回数) ※テンポラリーの為設定不要		繰り返し回数[回] ※0の場合は無限回			無条件分岐 待機時間経過後に繰り返し回数だけ分岐先STEPに分岐します	繰り返し回数
JMP_IN	0x0b	MODE_J1	分岐先STEP ※あり得ないSTEPの場合はEND_OFF命令となります	接点入力番号(1~8) 1~8以外は1として扱う	待機時間[ms] ※MODE_J1が時間待機時に使用			**	**	**	**		待機判断&IO条件分岐(ON論理) MODE_J1の条件完了で次のステップに移行します 条件完了前に指定の接点入力が入ONの場合は分岐先STEPに分岐します	MODE_J1: 0:無条件待機 1:指令位置到達 2:指令位置達&インポジション 3:時間待機
JMP_IN_OFF	0x0c	MODE_J1	分岐先STEP ※あり得ないSTEPの場合はEND_OFF命令となります	接点入力番号(1~8) 1~8以外は1として扱う	待機時間[ms] ※MODE_J1が時間待機時に使用			**	**	**	**		待機判断&IO条件分岐(OFF論理) MODE_J1の条件完了で次のステップに移行します 条件完了前に指定の接点入力が入OFFの場合は分岐先STEPに分岐します	MODE_J1: 0:無条件待機 1:指令位置到達 2:指令位置達&インポジション 3:時間待機
JMP_TRQ	0x0d	MODE_J1	分岐先STEP ※あり得ないSTEPの場合はEND_OFF命令となります	トルク閾値 [0.01Arms]	待機時間[ms] ※MODE_J1が時間待機時に使用			MODE_J2	**	**	**		待機判断&トルク条件分岐 MODE_J1の条件完了で次のステップに移行します 条件完了前に入力トルクがトルク閾値以上or以下になったら分岐先STEPに分岐します 閾値以上か以下かはMODE_J2により設定します	MODE_J1: 0:無条件待機 1:指令位置到達 2:指令位置達&インポジション 3:時間待機 MODE_J2: 0:閾値以上で分岐 1:閾値以下で分岐
JMP_STS	0x0e	MODE_J1	分岐先STEP ※あり得ないSTEPの場合はEND_OFF命令となります	ServoStatus判断bit ID20(サーボ状態表示)の条件判断したいBitを"1"に設定する 複数"1"を設定した場合はORで判断する 例)アラーム状態で分岐判断したい場合はBit3を"1"にする為"0x00000008"と設定	待機時間[ms] ※MODE_J1が時間待機時に使用			MODE_J2	**				ステータス判断分岐 MODE_J1の条件完了で次のステップに移行します 条件完了前にID20(サーボ状態表示)の指定Bitが「1」or「0」になったら分岐先STEPに分岐します 「1」か「0」かはMODE_J2により設定します	MODE_J1: 0:無条件待機 1:指令位置到達 2:指令位置達&インポジション 3:時間待機 MODE_J2: 0:「0」で分岐 1:「1」で分岐

## ●簡易コントロール機能命令一覧表 2/2

命令	プログラムデータ0(ID:177)				プログラムデータ1(ID:178)				プログラムデータ2(ID:179)				説明	補足
	31~24bit	23~16bit	15~8bit	7~0bit	31~24bit	23~16bit	15~8bit	7~0bit	31~24bit	23~16bit	15~8bit	7~0bit		
WAIT0	0x14	**	**	**	**	**	待機時間[ms]		**	**	**	**	無条件待機 待機時間経過後に次のSTEPに移行します	
PC_RESET	0x15	**	**	**	**	**	**	**	**	**	**	**	繰り返し回数全リセット プログラム中の全JMP0命令の繰り返し回数を初期値にリセットします。	
PC_RST_SP	0x16	**	指定STEP		**	**	**	**	**	**	**	**	繰り返し回数指定リセット 指定したステップのJPM0命令の繰り返し残り回数を初期値にリセットします 指定したステップがJMP0命令でない場合は何もしま	
OUT0	0x1e	**	1~5以外は1として扱う		**	**	出力論理 0:OFF		**	**	**	**	接点出力 指定した番号の接点出力をON/OFFします	
SV_OFF	0x28	**	**	**	**	**	**	**	**	**	**	**	サーボOFF サーボOFFします	
SV_ON	0x29	**	**	**	**	**	**	**	**	**	**	**	サーボON 現在位置でサーボONします	サーボON後は0.5secの待機時間が入ります
HOME	0x2a	**	**	**	**	**	**	**	**	**	**	**	原点復帰 ID90~96のパラメータに従い、原点復帰動作を行います(制御モード=4と同じ動作仕様)	
P_RESET	0x2b	**	**	**	**	**	**	**	**	**	**	**	現在位置をID:39(ポジションリセット値)に初期化します	セットを行うと、一旦0速度制御になり、リセット完了後位
AL_RESET	0x2c	**	**	**	**	**	**	**	**	**	**	**	アラームリセット アラームをリセットします	
END_P	0x2d	**	**	**	**	**	**	**	**	**	**	**	プログラム終了 プログラムを終了します サーボON/OFF状態は継続したままID31(制御モード)=1[位置制御]になります	位置指令は現在位置になります
END_V	0x2e	**	**	**	**	**	**	**	**	**	**	**	プログラム終了 プログラムを終了します サーボON/OFF状態は継続したままID31(制御モー	速度指令は0になります
END_C	0x2f	**	**	**	**	**	**	**	**	**	**	**	プログラム終了 プログラムを終了します サーボON/OFF状態は継続したままID31(制御モード)=3[電流制御]になります	電流指令は0になります
END_OFF	0x30	**	**	**	**	**	**	**	**	**	**	**	プログラム終了 プログラムを終了し、サーボOFFします ID31(制御モード)=0になります	
PARAM_W	0x31	**	**	**	**	**	**	**	**	**	**	**	指定したパラメータIDのパラメータ変更を行います。 書き込めないパラメータを指定した場合は何も行いません。	書込データは4byteですが、2byte/1byteのパラメータを変更する場合は上位byteは無視されます
PARAM_W	0x32	**	パラメータID (0~255)		書込データ				**	**	**	**		