

スマートモータ入門編から使用方法と代表的なプログラムを紹介します。
 入門編には下記内容のプログラムと補足資料を準備しています。(抜粋した箇所=※)
 入門編が必要な場合は弊社もしくは正規代理店にお問い合わせください。

サンプルプログラム

- 3-1 入力指令と完了信号出力(I/O の設定と使用方法)
- 3-2 Z 相入力による原点サーチ
- 3-3 **押し当て原点サーチ + センサによる原点サーチ ※**
- 3-4 エラーハンドリング
- 3-5 前後繰り返し動作
- 3-6 繰り返し動作中の負荷を監視する
- 3-7 **アナログ入力による速度変更&制御 ※**
- 3-8 **デジタル入力による速度変更&制御 ※**
- 3-9 動作中に I/O から信号を出力する
エンコーダカウントが指定した位置に到達すると信号を I/O から出力する
- 3-10 指定電圧より実行電圧が下がると回転を停止する
- 3-11 不揮発性 (EEPROM) メモリと変数データの保存方法
- 3-12 往復動作中にプログラム設定により速度を変更する
- 3-13 **リニアアクチュエータ用プログラム例 ※**
入力 4 点による 16 点位置決め、ソフトリミット、押し当て原点サーチ
- 3-14 位置のティーチングとプレイバック制御
- 3-15 パルス列指令制御
- 3-16 電子ギヤ制御(追従)
- 3-17 外部エンコーダを使用した閉ループ制御
- 3-18 コンビトロニック通信(多軸制御)
 - 3-18-1 コロンによるモータの指定方法
 - 3-18-2 各モータの位置比較
 - 3-18-3 エンコーダモジュール
 - 3-18-4 **2 軸制御 ※**
 - 3-18-5 **直線補間(同期制御) ※**
 - 3-18-6 多軸制御で割り込みを使用する方法
 - 3-18-7 接続の注意&確認
 - 3-18-8 コンビトロニックにおける指令と違反例
 - 3-18-9 スレーブモータへの I/O 割り込み
- 3-19 乱数を使用して動作間の待ち を設定
- 3-20 ハードウェアリミット(センサー入力)を使用した原点サーチ
- 3-21 エンコーダモジュール機能

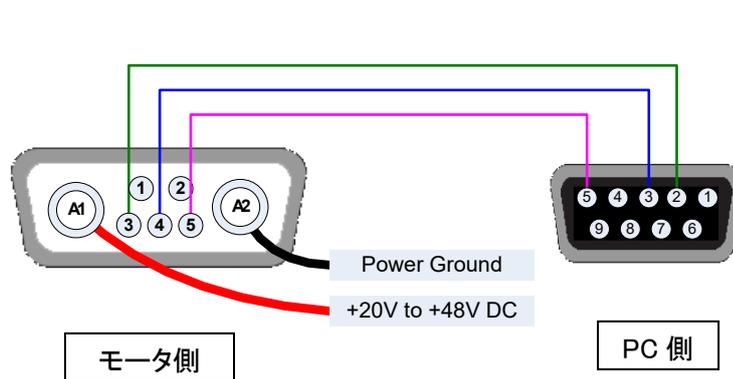
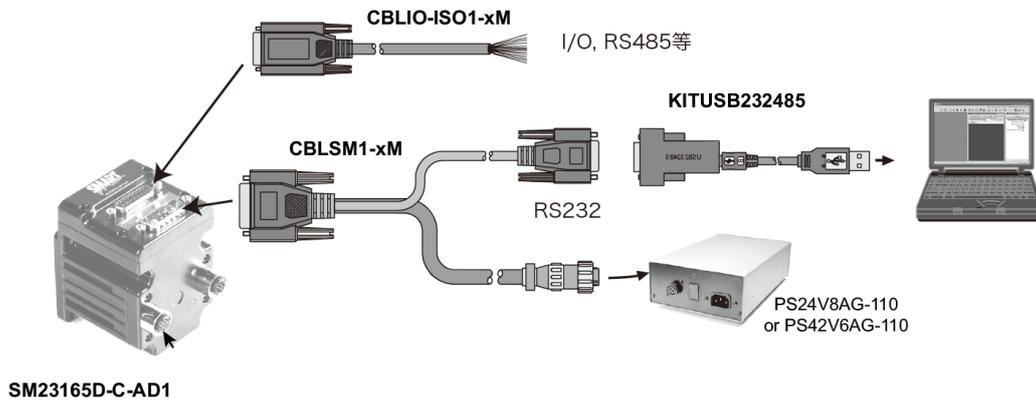
3-22 PLC・マイコンからのヘルシーシグナルプログラム

付録

-
- 4-1 I/O 仕様
 - 4-2 I/O タイプ
 - 4-3 PID 設定の目安
 - 4-4 PID 値の変更&保存
 - 4-5 DE オプション(制御電源入力分離オプション)
 - 4-6 ユーザ指定割り込み-最大 8 点使用可能
 - 4-7 トルクモードについて
 - 4-8 電源選択
 - 4-9 シャントについて
 - 4-10 RS232&RS485 接続方法
 - 4-11 RS485 通信する際の初期設定
 - 4-12 ASCII コマンド/文字コードによる通信
 - 4-13 通信コマンドタイミング
 - 4-14 ターミナルエミュレータ (Tera Term) 使用例
 - 4-15 シリアルデータアナライザ
 - 4-16 チャートデータをエクセルに出力
 - 4-17 LED 機能説明
 - 4-18 プログラム容量

1章 基本的な設定とサーボモータの制御

接続例（基本）



7ピンコンボDサブコネクタ

- A1 +20~+48DVC
- A2 電源 GND
- 1 I/O G
- 2 +5V 出力
- 3 RS232 送信 (Tx)
- 4 RS232 受信 (Rx)
- 5 信号 GND

9ピン RS232

- 2 RS232 受信 (Rx)
- 3 RS232 送信 (Tx)
- 5 RS232 GND

1-2 通信リンクの確立

PC を使用し SMI を立ち上げる

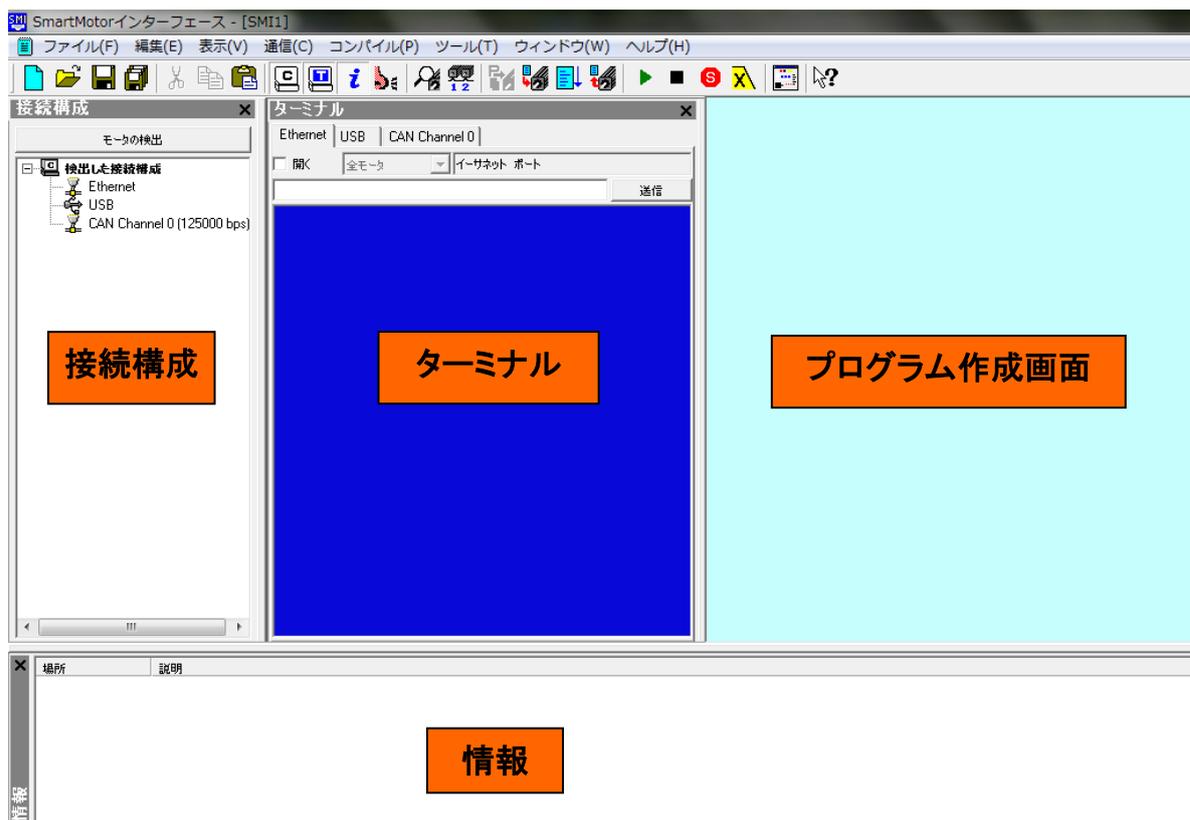


SMI を PC にインストールする際は、下記 URL からインストールしてください。(無償)

<https://www.moog.co.jp/products/motors-servomotors/smartmotor.html>



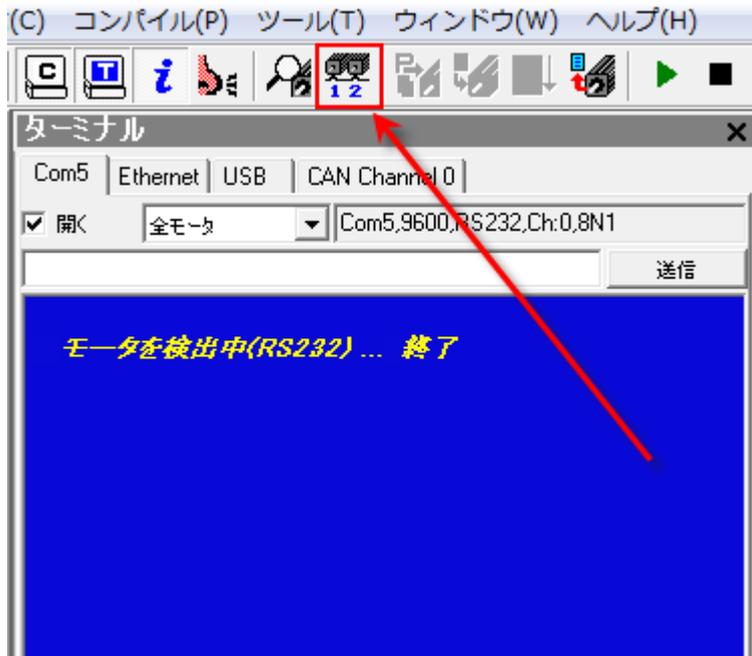
SMI インターフェース基本画面



*** シリアルデータアナライザ**

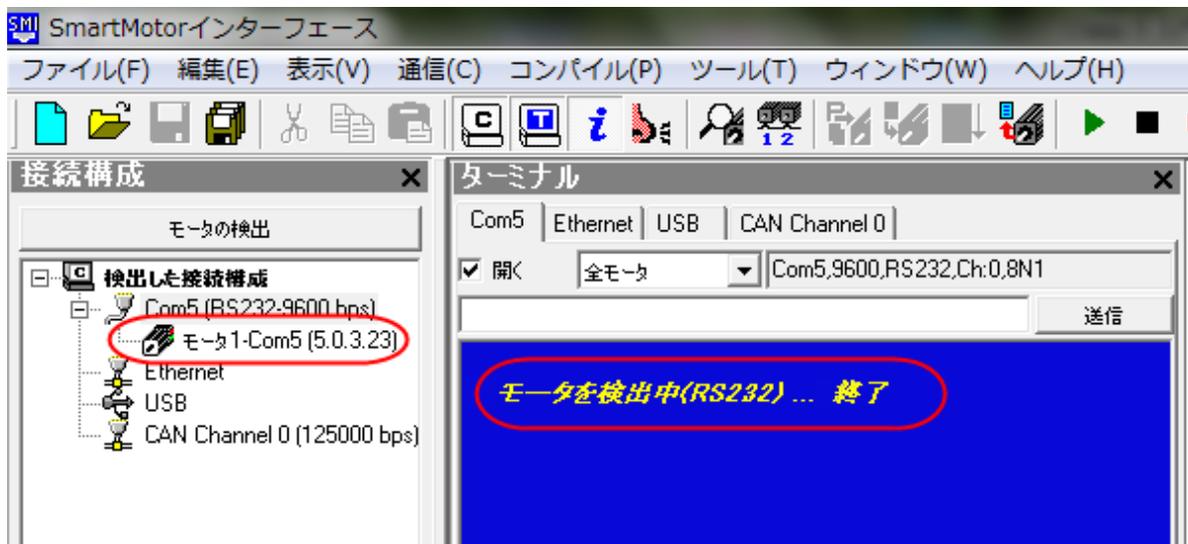
SMI とスマートモータ間のデータ情報を確認するのに便利な機能です。付録をご参考にしてください。

PC と一体型サーボモータの通信を確立する



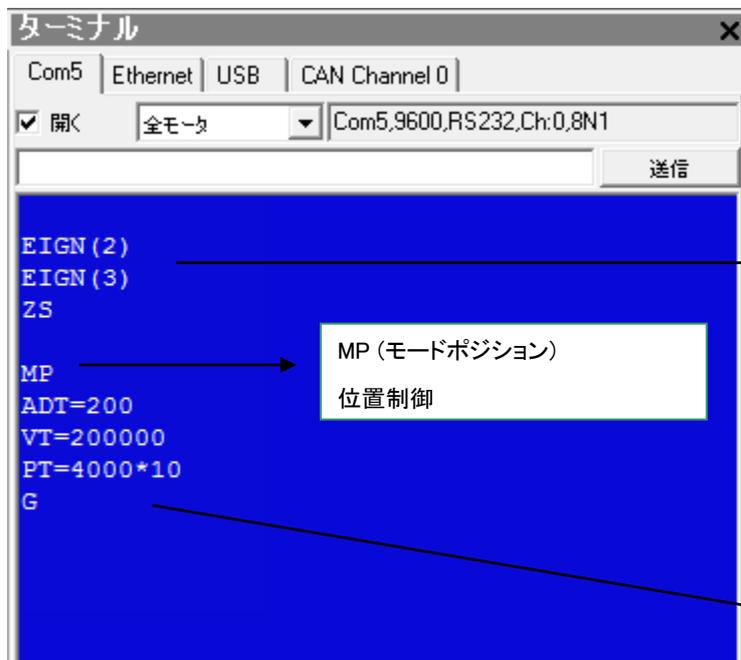
単軸で使用する場合、アドレスを指定する必要はありませんが、多軸ネットワークで使用する場合は、SMIから自動でアドレスを順番に割り当てないとターミナルウィンドウ（SMI青画面）から正確な制御指示ができません。（注：予め各軸にアドレスを書き込んでおき、そのアドレスを検出する際は、 をクリックしてください。）

通信の確立を確認する



1-3 正方向に 10 回転させる (位置制御)

ターミナル(青画面内)にコマンドを書き込みG(GO) + Enter で回転開始



ポート 2&3 はオーバトラベルリミット入力としてデフォルト設定になっています。リミットを無効にするか、リミットセンサー(ノーマリクローズ)を接続し、Low 状態にしないと動作しません。

EIGN(2): ポート2を入力に設定しリミット入力を無効にする

EIGN(3): ポート3を入力に設定しリミットを無効にする

ZS: ステータスビットのクリア

SM23XX: 4000 count/回転(エンコーダ分解能)

(SM34XX): 8000 count/回転(エンコーダ分解能)

絶対目標位置

PT=絶対目標到達位置

例: PT=4000 *10(回転数)=40000

(現位置にかかわらず、絶対値 40000 に移動する。0=>40000, 2500=>40000)

相対目標位置

PRT=相対位置移動量

例: PRT=4000 (現絶対位置が 40000 の場合、40000+4000=44000)

絶対位置を指定して動作



[到達速度値] VT

エンコーダ分解能 4000 のモータで PID 設定が PID2 の場合:

$$VT = \text{rev/sec} * 32768$$

$$\text{例: } 200,000 / 32,768 = 6.10 \text{ rps}$$

$$6.10 * 60 = 366 \text{ rpm}$$

エンコーダ分解能 8000 のモータで PID 設定が PID2 の場合:

$$VT = \text{rev/sec} * 65536$$

[加減速値] ADT (加速値:AT, 減速値:DT)

$$ADT = 4.096 \times \text{回転/秒}^2 \quad (\text{エンコーダ分解能 4000 のモータで PID 設定が PID2 の場合})$$

$$ADT = 8.192 \times \text{回転/秒}^2 \quad (\text{エンコーダ分解能 8000 のモータで PID 設定が PID2 の場合})$$

1-4 設定速度 100RPM で回転させる（速度制御）

```
EIGN (2)
EIGN (3)
ZS

MV
ADT=200
VT=54613
G
```

MV (モードベロシティ)
速度制御

エンコーダ分解能が 4000 の場合:

$$VT = \text{rev/sec} * 32768$$

Ex: $VT = 1.66\text{rps} * 32768 = 54613$

負方向に 100RPM で回転

```
VT=-54613
G
```

ヒント: 以前に設定したパラメータを継続して使用する場合は、再度パラメータを書く必要はありません。

正方向に 200RPM に回転

```
VT=109227
G
```

回転停止(減速停止)

```
X
```

1-5 パラメータのモニタリング

接続構成

モータの検出

- 検出した接続構成
 - Com5 (RS232-9600 bps)
 - モータ1-Com5 (5.0.3.23)
 - Ethernet
 - USB
 - CAN channel 0 (125000 bps)

モータ表示 "モータ1-Com5"

ステータス 情報 モニター ユーザ変数 フラッグ Calculations Trace

モニター 停止 リフレッシュ **オンライン** ADDR: 1

目盛り 16進数表示

Position: 1048211
Pos. Error: 0
Velocity: 0
モード: Velocity
温度: 32
電流: 0

入力/出力 赤 5v, グレー 0v

0 1 2 3 4 5 6

Status Word 0

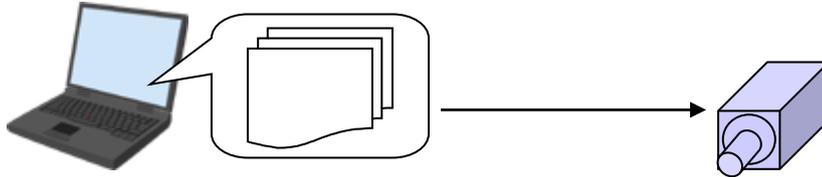
- Drive ready
- B0: Motor is off
- B1: Trajectory in progress
- B2: Bus voltage fault
- B3: Over-current occurred
- B4: Excessive temperature fault
- B5: Excessive position error
- B6: Velocity limit
- B7: Real-time temperature limit
- B8: de/dt error limit
- B9: Positive H/W limit enabled
- B10: Negative H/W limit enabled
- B11: Historical positive H/W limit
- B12: Historical negative H/W limit
- B13: Positive H/W limit asserted
- B14: Negative H/W limit asserted

1. ダブルクリック

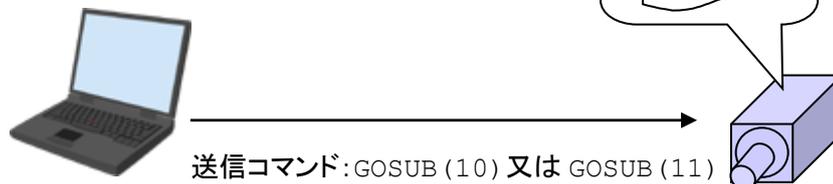
モニタリングをするパラメータを選択

2章 制御プログラムをモータにアップロード

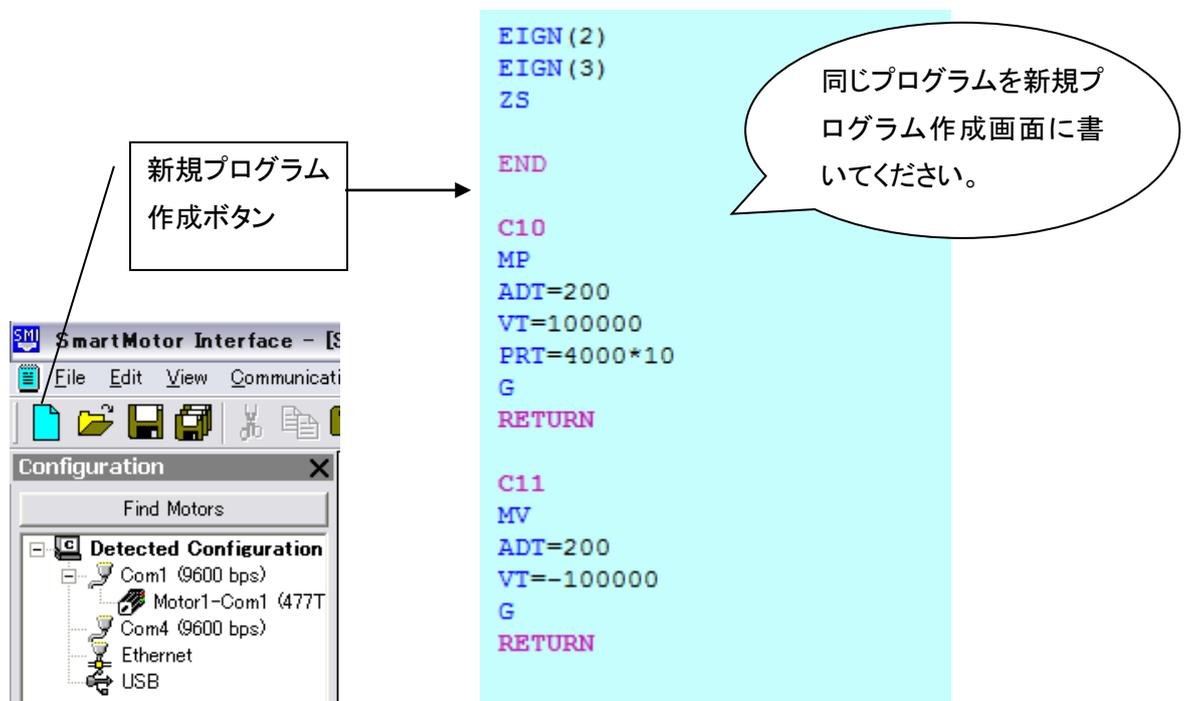
① 新規プログラム作成しモータにアップロード



② サブルーチンコマンドの指定と送信



2-1 新規プログラムの作成



新規プログラム作成ボタン

```

EIGN (2)
EIGN (3)
ZS

END

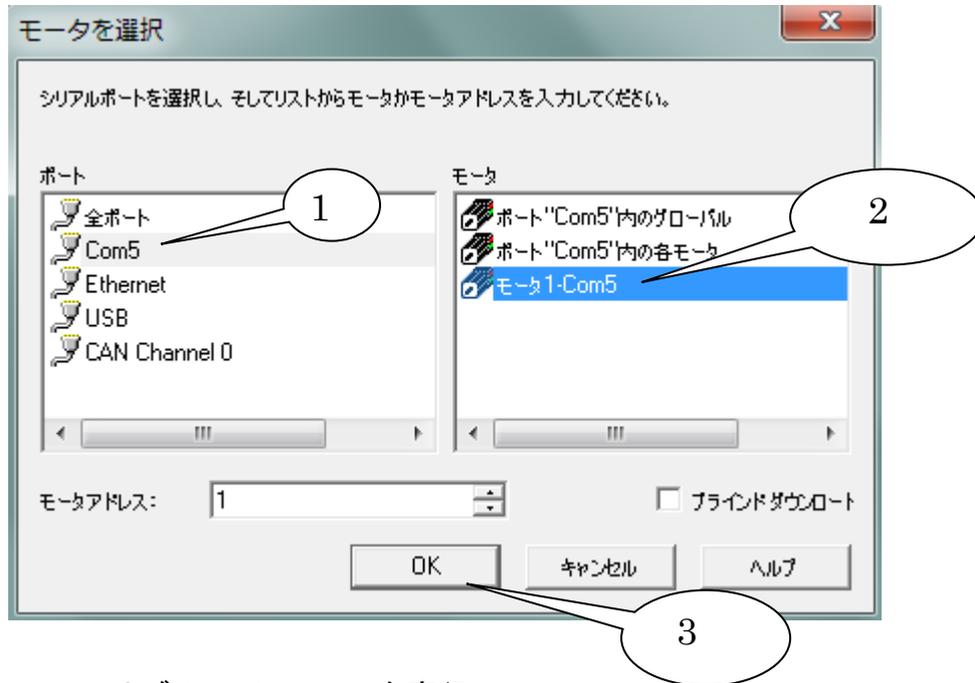
C10
MP
ADT=200
VT=100000
PRT=4000*10
G
RETURN

C11
MV
ADT=200
VT=-100000
G
RETURN

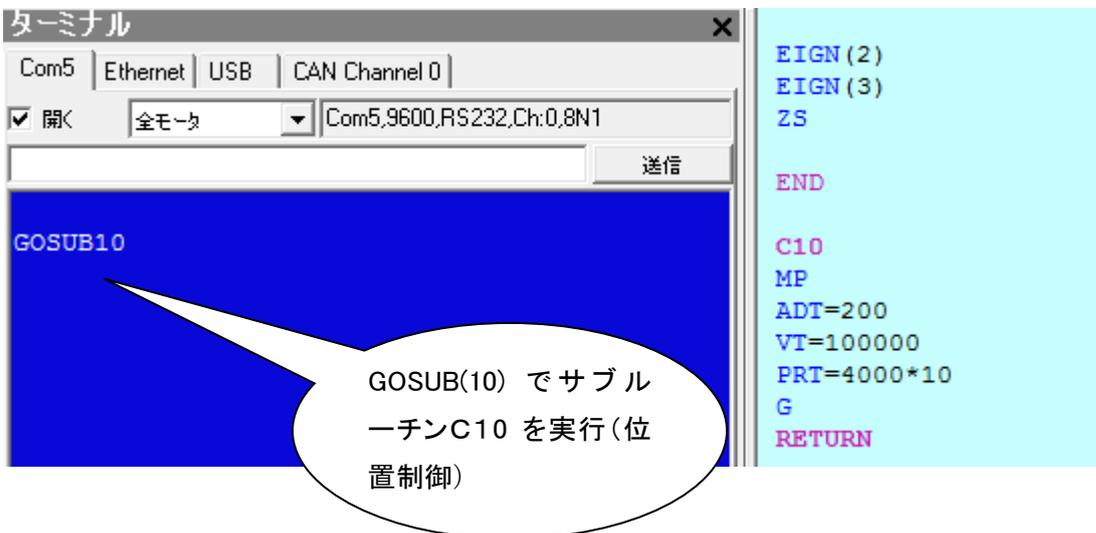
```

同じプログラムを新規プログラム作成画面に書いてください。

2-2 プログラムをモータにダウンロード



2-3 サブルーチン C10 を実行



3-8 デジタル入力による速度変更&制御

位置制御でスタートし、デジタル入力 A が Low になると速度が速くなります。

```

EIGN(W,0)           'ハードウェアリミット・I/Oを無効
KD=10010           'KDの値を変更
F                  'KDの値を有効
O=0                '原点設定
ADT=100            '加速設定
VT=10000           '速度設定
PT=40000           '到達地点設定
MP                 '位置モード設定
G                  'スタート
WHILE Bt           '動作時のループコマンド
  IF IN(0)==0       '0ポートがLowのとき
    IF VT==10000    'VTの値が10000になったとき
      VT=12000      '新しいVTの値を追加
    G                'スタート
  ENDIF
ENDIF
LOOP '動作中のループコマンド(WHILE Bt に戻る)
END

```

3-7 アナログ入力による速度変更&制御

アナログ信号に敏感に反応しないように、デットバンド機能を含んだプログラムです。オペレータがアナログ入力を変更しない限り、モータ速度は変わりません。

```

EIGN (W, 0)           'ハードウェアリミットを無効
KP=3020              'PID値(比例)をデフォルトから変更
KD=10010            'PID値(微分)をデフォルトから変更
F                   'PID設定バッファ値有効
ADT=100             '加速設定
MV                  '速度モード
d=10                'アナログデットバンド, 5000 = フォルトスケール
o=2500              'マイナス方向動作のためのオフセット
m=40                '速度(倍数)
w=10                '時間(10m秒)
b=0                 '速度
C10
a=INA (V1, 3) -o    '5Vアナログ値読み込み
x=a-b               'xを設定して入力値を変更する
IF x>d             'デットバンドを超えているか確認
    VT=b*m           '速度値の設定
    G                 '新しい速度でスタート
ELSEIF x<-d       'デットバンドを超えているか確認
    VT=b*m           '速度値の設定
    G                 '設定した速度でスタート
ENDIF
b=a                 'ハンチング防止アップデート
WAIT=w              'w秒待機
GOTO10              'ラベル10の呼び出し
END

```

D サブ 15 ピン I/Os (5V I/O)

- INA (A, exp) 生アナログ読込 : 10Bit
分解能, 0-32736=0-5VDC
- INA (V1, exp) 電圧スケール (ミリボルトで読込), 3456=3.456VDC

拡張 10点 24DC I/O

- INA (A, exp) 生アナログ読込 : 10Bit
分解能, 0-32736=0-41.25VDC
 - 分解能は 10bit で 32 毎増減,
 - 電圧値 0-41.25VDC はボード内レファレンス値。I/O 最大入力値は 24VDC
- INA (V, exp) 電圧スケール読込
(24000-0), 15500=15.5V
- INA (V1, exp) 電圧スケール読込
(5000-0), 550=0.55V

3-1 センサ入力からの原点出し

```
EILP          ' 正方向リミットスイッチ有効(デフォルト設定)
EILN          ' 負方向リミットスイッチ有効(デフォルト設定)
ZS
END
C0            ' 原点出しサブルーチン
EIGN (3)     ' ポート3の負方向用リミットスイッチを無効
ZS
MV            ' 速度モード設定
ADT=100      ' 原点復帰加速設定
VT=-70000    ' 原点復帰速度設定
G            ' 原点(リミットセンサ)方向に回転開始
WHILE IN(3) == 0 LOOP ' ポート3がLowのときループ ->Highになるとループを抜ける
MTB          ' トルクブレーキモードを使用し停止
WAIT=50      ' 50m秒待機
O=-8000      ' 現在位置をマイナス8000と設定
MP PT=0 G TWAIT ' モータを位置ゼロへスタート
EILN        ' ポート3をリミットスイッチ用に設定(負方向リミットスイッチ有効)
RETURN
```

3-3 押し当て原点サーチ

このプログラムから理解すること:

1. センサー入力を使用せずに押し当て原点復帰させる方法

注意: 押し当てる際は設定速度に注意し、偏差リミットは出来るだけ小さな値を使用してください。高速による衝突停止や、位置偏差の値を大きく設定して押し当てると、モータが破損する可能性がありますので、押し当て原点サーチはユーザの判断で使用してください。

```

' =====
' 原点復帰ルーチン (押し当て原点復帰)

EIGN(2) EIGN(3) ZS      'トラベルリミットを解除し、I/Oを通常入出力に設定する
' パラメータ設定
rr=-1                  '原点復帰回転方向
vv=70000               '原点復帰速度
aa=1000               '原点復帰加速
ee=300                 '原点復帰偏差リミット
tt=3000               '原点復帰トルクリミット
hh=4000               '原点復帰オフセット
ZS                     'エラービットをクリア

MV                     '速度モード設定
ADT=aa                 '原点復帰加速設定
VT=vv*rr              '原点復帰速度設定
G                      '押し当て方向に回転開始
WHILE ABS(EA) < ee LOOP '偏差リミット内の値はループ
MTB                    'トルクブレーキモードを使用し停止
MT                     '押し当て時、バウンスするためトルクモードで再度押し当てる
T=tt*rr               'トルク値を前設定
G
WAIT=50               '50m秒待機
O=hh*rr               '原点オフセット位置を原点に設定
MP PT=0 G TWAIT      'モータを位置ゼロに設定
END

```

3-4 エラーハンドリング

```

EIGN(W,0,12) ' トラベルリミットを解除し、I/Oを通常入出力に設定
              ' ビットは全ての入力をバイナリ値12でマスクされ、入力2と3を一般入力に設定す
              ' る。

ZS           ' フォルトビットをクリア(注:ブートアップ時にトラベルリミットがグランドされていな
              ' い場合)

              ' フォルト割り込み設定
              ' ITR(割り込み番号, ステータスワード, ステータスビット, ビットステータス, C
              ' ラベル呼び出し)
              ' 注:割り込みは8個設定可能, 0~7, 0は最高のプライオリティとする

ITR(0,0,0,0,0) ' 割り込みをゼロに設定
              ' ステータスワードゼロ
              ' ビットゼロ
              ' ビットステータスをゼロに設定(ドライブレディビット)
              ' サブルーチンゼロ(C0)呼び出し

EITR(0)      ' 割り込みゼロを有効にする

ITRE        ' グローバル割り込みスキャナーを有効にする

PAUSE      ' 一時停止コマンド(ENDコマンドにより、割り込みが無効になるのを防ぐ)

END

C0
' エラーコードを配置
' このルーチンはトラベルリミット、温度超過、位置偏差エラー、過電流時に割り込まれる。

IF Be
    PRINT("Position Error",#13)
ENDIF
IF Bh
    PRINT("Over Temp Error",#13)
ENDIF
IF Ba
    PRINT("Over Current Error",#13)
ENDIF
RETURNI

```

3-13 リニアアクチュエータ用プログラム例

入力 4 点による 16 点位置決め、ソフトリミット、押し当て原点サーチ

当社で販売している SmartBox BCD を使用すると便利です。

4点の入力をベースに16のサブルーチンを呼び出し、入力6 (G) を受けると動作します。

```
'ECHO                                'エコーオン
EIGN(W,0)                             '全てのI/Oを入力に設定
ZS                                     '全エラービットリセット
MDS                                    '正弦波駆動モード
' =====
OUT(4)=1                               'Eポートをビジー出力に設定
OUT(5)=0                               'Fポートをエラー出力に設定
' =====
'パラメータの設定
    rr=-1                               '原点復帰
    vv=100000                           '速度
    aa=100                               '加速度
    ee=300                               '電流エラー値
    tt=4000                              'トルクリミット
    hh=4000                              'オフセット
    nn=-300                              'ソフトウェアリミット (マイナス方向)
    pp=110000                            'ソフトウェアリミット (プラス方向)
' =====
SLE                                   'ソフトウェアリミットを有効
SLN=nn                               'マイナス方向回転リミット
SLP=pp                               'プラス方向回転リミット
SLM(1)                               'ソフトリミットモードに設定
' =====
ITR(0,0,0,0,100)                     'エラー時の割込み (4-6参照)
EITR(0)                               '割り込みゼロを有効にする
' =====
ITR(1,16,6,0,101)                   '割込み設定 (4-6参照)
ITRE                                  'グローバル割り込みスキャナーを有効にする
EITR(1)                               '割り込み1を有効にする
WHILE 1 LOOP
END
' =====
```

C100

```

OUT (5)=1                'Eポートをhigh
IF Be                    '位置偏差エラービット
    PRINT("Position Error",#13)
ENDIF
IF Bh                    '超過温度エラービット
    PRINT("Over Temp Error",#13)
ENDIF
IF Ba                    '過電流エラービット
    PRINT("Over Current Error",#13)
ENDIF

```

RETURNI

```
'=====
```

C101 'ポートFに入力があつたときに101を呼び出し

```

PRINT("GO PRESSED",#13)
x=15-(IN(W,0)&15)        '入力ポート 0,1,2,3 (A, B, C, D)を確認
GOSUB(x)                 'サブルーチン0-15を呼び出し
ITR(1,16,6,0,101)       '(4-6参照)

```

RETURNI

```
'=====
```

```
'押し当て原点復帰
```

C0

```

ZS                        '全エラービットリセット
SLD                       'ソフトリミット無効
OUT(4)=0                  'Dポート出力Low
VT=vv*rr                  '速度
ADT=aa                    '加速度
MV                         '速度モード
ZS                         'エラービットのクリア
G
WHILE ABS(EA)<ee LOOP     'ループ 偏差リミット内
MTB                       'トルクブレーキ
MT                         'トルクモード
T=tt*rr                   'トルク値設定
G
WAIT=500                  '500m秒待機
O=hh*rr                   '位置設定
MP PT=0 G TWAIT          '位置モード・原点復帰

```

```
OUT (4) =1          'Dポート出力High  
SLE                'ソフトリミット有効
```

```
RETURN
```

```
'=====
```

```
C1
```

```
OUT (4) =0
```

```
MP
```

```
ADT=300
```

```
VT=900000
```

```
PT=90000
```

```
G TWAIT
```

```
PT=0
```

```
G TWAIT
```

```
PT=90000
```

```
G TWAIT
```

```
PT=0
```

```
G TWAIT
```

```
OUT (4) =1
```

```
RETURN
```

```
'=====
```

```
C2
```

```
C3
```

```
C4
```

```
C5
```

```
C6
```

```
C7
```

```
C8
```

```
C9
```

```
C10
```

```
C11
```

```
C12
```

```
C13
```

```
C14
```

```
C15
```

```
PRINT ("Called SUBROUTINE ", x, #13)
```

```
PRINT ("EXITING SUBROUTINE ", x, #13)
```

```
RETURN
```

CAN&Combitronic 接続例(多軸制御)

1. 上位コントローラ(PC/PLC)を使用しない場合:

全スマートモータを CAN ネットワーク接続

コンビトロニック通信を使用

任意のスマートモータをマスタとして多軸モータ制御や各軸の I/O を制御する。



2. 上位コントローラ(PC/PLC)が必要で全てを CAN で接続する場合:

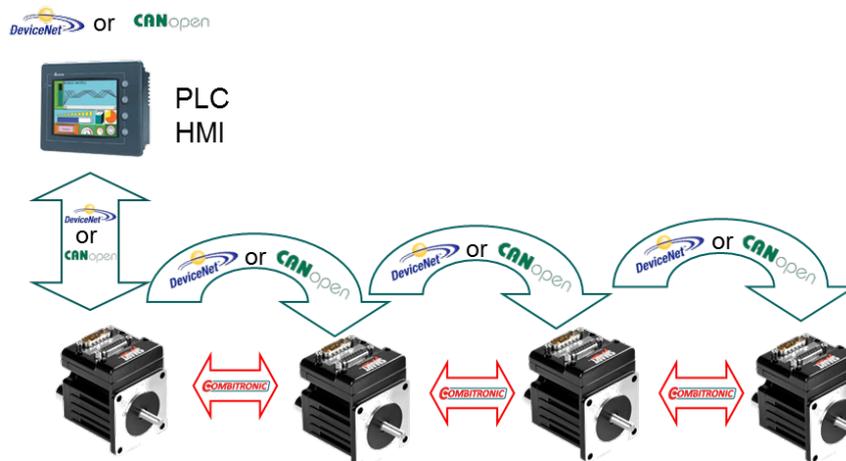
2-1 上位コントローラをメインマスタとする。

CANopen プロトコルを使用しスマートモータはスレーブ機器として制御される。(CANopen)

2-2 スマートモータをメインマスタとする。

上位コントローラは最小限のデータの送受信 (CANopen) をするだけで、スマートモータがマスタとなり制御する(Combitronic)。

コンビトロニック通信を併用した制御 (CANopen と Combitronic を併用)



3-18-4 2 軸制御

Master Motor(モータ1)

CADDR=1

'モータアドレス1であることを宣言

ADDR=CADDR

'CANアドレスとRS232アドレスを統一する

EIGN(2):0

'全モータ: 第二ポートを一般入力ポートに指定

EIGN(3):0

'全モータ: 第三ポートを一般入力ポートに指定

ZS:

'全モータリセット

MV:0

'全モータ: 速度モード

VT:1=64424*3

'モータ1 : 速度指定 (3rps)

ADT:0=1000

'全モータ: 加速度指定

G:1

'モータ1: 実行

'10秒待機

WAIT=1000*10

VT:2=64424*5

'モータ2: 速度指定 (5rps)

G:2

'モータ2: 実行

WHILE PA:1 > PA:2 LOOP

'ループ【モータ1の位置>モータ2の位置】

X:0

'全モータ: 停止

MP:0

'全モータ: 位置モード

PML:0=4000

'全モータ: エンコーダモジュールリミット指定

PMT:0=2000

'全モータ: エンコーダモジュールターゲット指定

G:0

'全モータ: 実行 (エンコーダモジュール機能)

END

Slave Motor(モータ2)

CADDR=2

'モータアドレス2であることを宣言する

ADDR=CADDR

'CANアドレスとRS232アドレスを統一する

END

3-18-5 直線補間（同期制御）

ホスト PC/PLC を使用せずに直線補間制御が可能です。

ADTS=100	＼同期指令加速度を設定
VTS=100000	＼同期指令到達速度を設定
PTS(30000;1,40000;2)	＼モータ1と2の到達位置を設定
	＼PTS(位置 1; 軸番号 1, 位置 2; 軸番号 2 [, 位置 3; 軸番号 3])
GS	＼スタート, 同期開始

Class5 コマンド表 (Ver1.10)

コマンド	説明
A	
a...z	ユーザ変数
a=...z=	ユーザ変数設定
aa...zz	ユーザ変数
aa=...zz=	ユーザ変数設定
aaa...zzz	ユーザ変数
aaa=...zzz=	ユーザ変数設定
ab[index]	8ビット変数
ab[index]=...	8ビット変数設定
af[index]	フロート変数
af[index]=...	フロート変数設定
al[index]	32ビット変数
al[index]=...	32ビット変数設定
aw[index]	16ビット変数
aw[index]=...	16ビット変数設定
Ai(0)	内部エンコーダZ相位置をライジングエッジで取得
Ai(1)	外部エンコーダZ相位置をライジングエッジで取得
Aij(0)	内部エンコーダZ相位置をライジングエッジとフォーリングエッジで取得
Aij(1)	外部エンコーダZ相位置をライジングエッジとフォーリングエッジで取得
Aj(0)	内部エンコーダZ相位置をフォーリングエッジで取得
Aj(1)	外部エンコーダZ相位置をフォーリングエッジで取得
Aji(0)	内部エンコーダZ相位置をフォーリングエッジとライジングエッジで取得
Aji(1)	外部エンコーダZ相位置をフォーリングエッジとライジングエッジで取得
ABS(...)	整数絶対値
AC	指令加速度
ACOS(...)	角度によるアークコサイン(コサインの逆関数)
ADDR	モータシリアルアドレス
ADDR=...	モータシリアルアドレス設定
ADT=...	加減速度設定
ADTS=...	同期運転加減速度設定
AMPS	最大PWMリミット
AMPS=...	最大PWMリミット設定
ASIN(...)	角度によるarc-sine(サインの逆関数)
AT	加速度
AT=...	加速度設定
ATS=...	同期運転加速度設定
ATAN()	角度によるアークタンジェント(タンジェントの逆関数)
ATOF()	ASCIIを取得しフロート型に変換
B	
B()	ステータスビット
Ba	過電流ステータスビット
BAUD(0)	チャンネル0ボーレート
BAUD(1)	チャンネル1ボーレート
BAUD#	チャンネル0ボーレート設定
BAUD(0)=...	チャンネル0ボーレート設定
BAUD(1)=...	チャンネル1ボーレート設定
Be	位置偏差エラーステータスビット

Bh	温度超過ステータスビット
Bi(0)	内部エンコーダZ相ステータスビットをライジングエッジで取得
Bi(1)	外部エンコーダZ相ステータスビットをライジングエッジで取得
Bj(0)	内部エンコーダZ相ステータスビットをフォーリングエッジで取得
Bj(1)	外部エンコーダZ相ステータスビットをフォーリングエッジで取得
Bk	EEPROMデータ完全性ステータスビット
Bl	左/負方向ハードウェアオーバートラベルリミット履歴ステータスビット
Bls	左/負方向ソフトウェアオーバートラベルリミット履歴ステータスビット
Bm	左/負方向ハードウェアオーバートラベルリミットステータスビット
Bms	左/負方向ソフトウェアオーバートラベルリミットステータスビット
Bo	モータOFFステータスビット
Bp	右/正方向ハードウェアオーバートラベルリミットステータスビット
Bps	右/正方向ソフトウェアオーバートラベルリミットステータスビット
Br	右/正方向ハードウェアオーバートラベルリミット履歴ステータスビット
Brs	右/正方向ソフトウェアオーバートラベルリミット履歴ステータスビット
Bs	構文エラーステータスビット
Bt	動作進行中ステータスビット
Bv	速度エラーステータスビット
Bw	32ビット位置カウンタ ラップ・アラウンドステータスビット
Bx(0)	リアルタイム内部インデックス入カステータスビット
Bx(1)	リアルタイム外部インデックス入カステータスビット
BREAK	プログラム実行フローコントロール
BRKENG	ブレーキ作動
BRKRLS	ブレーキ解除
BRKSRV	サーボ停止中ブレーキ作動
BRKTRJ	動作停止中ブレーキ作動
C	
C#	プログラム・サブルーチンのラベル
CADDR	CANアドレス
CADDR=	CANアドレス設定
CAN	CANエラー
CANCTL(...)	ネットワーク機能制御
CASE#	プログラムフロー命令
CBAUD	CANボーレート
CBAUD=	CANボーレート設定
CCHN()	シリアルチャンネル閉
CHN(0)	RS232 通信エラーフラッグ
CHN(1)	RS485 通信エラーフラッグ
CLK	1ミリ秒クロック変数
CLK=...	1ミリ秒クロック設定
COS(...)	コサイン(角度)
CP	カムポインタ
CTA(...)	カムテーブル追加
CTE(...)	カムテーブル削除
CTR(0)	プライマリエンコーダ/パルス列&方向カウンタ
CTR(1)	セカンドエンコーダ/パルス&方向カウンタ
CTT	EEPROM内カムテーブル数
CTW()	カムテーブル書き込み
D	
DEA	実行de/dt値

DEFAULT	Switch-case 構文要素
DEL	de/dtフォルトリミット
DEL=...	de/dtフォルトリミット設定
DFS(...)	32ビットIEEEフォーマット内、af[]変数
DITR(...)	割り込み停止
DT	減速度
DT=...	減速度設定
DTS=...	同期動作減速度設定
E	
EA	位置偏差エラー
ECHO	入力データを主チャンネルに出力
ECHO_OFF	主チャンネルエコー停止
ECHO1	入力データを2次チャンネルに出力
ECHO_OFF1	2次チャンネルエコー停止
EIGN(...)	I/Oピンを入力に設定
EILN	左/負側ハードウェアリミットスイッチオン
EILP	右/正側ハードウェアリミットスイッチオン
EIRE	インデックス信号取得ピンを外部エンコーダ用に設定
EIRI	インデックス信号取得ピンを内部エンコーダ用に設定
EISM(6)	ピン6をGコマンド入力に設定
EITR(...)	割り込み設定
EL	位置偏差エラーフォルトリミット
EL=...	位置偏差エラーフォルトリミット設定
ELSE	IF構文要素
ELSEIF	ELSE構文要素
ENC0	内部エンコーダ選択
ENC1	外部エンコーダ選択
END	プログラム終了
ENDIF	IF命令文終了
ENDS	スイッチ構文終了
EOBK(...)	I/O出力にブレーキ信号送信
EPTR	EEPROMポインタ
EPTR=...	EEPROMポインタ設定
ERRC	直近コマンドエラーコード
ERRW	直近コマンドエラー通信チャンネル
F	
F	バッファされたPID設定値有効
FABS(...)	浮動小数点絶対値エラー
FSA(...)	フォルトアクション設定
FSQRT(...)	浮動小数点平方根
FW	ファームウェアバージョン
G	
G	モーションスタート(GO)
G(...)	指定軌道モーションスタート(GO)
GS	同期動作モーションスタート(GO)
GETCHR	主通信チャンネルの文字列
GETCHR1	2次通信チャンネルの文字列
GOSUB(...)	数字又は変数によるサブルーチン呼び出し
GOSUB#	サブルーチン呼び出し
GOTO(...)	数字又は変数によるプログラムラベルにジャンプ

GOTO#	プログラムラベルにジャンプ
H	
HEX(...)	hex文字列変数
I	
I(0)(capital i)	エンコーダZ相入力位置変数(ライジングエッジ、内部エンコーダ)
I(1)(capital i)	エンコーダZ相入力位置変数(ライジングエッジ、外部エンコーダ)
IF...	IF 構文要素
IN(...)	I/O入力
INA(...)	アナログ入力
ITR(...)	ユーザ割り込み設定
ITRD	全ユーザ割り込み停止
ITRE	全ユーザ割り込み有効
J	
J(0)	エンコーダZ相位置変数取得(フォーリングエッジ、内部エンコーダ)
J(1)	エンコーダZ相位置変数取得(フォーリングエッジ、外部エンコーダ)
K	
KA	KAバッファPID値(加速)
KA=...	KAバッファPID値設定
KC	KC値
KC=...	KC値設定
KCS	KCS値
KCS=...	KCS値設定
KD	KDバッファPID値(微分)
KD=...	KDバッファPID値設定(微分)
KG	KGバッファPID値(重力)
KG=...	KGバッファPID値設定(重力)
KI	KIバッファPID値(積分)
KI=...	KIバッファPID値設定(積分)
KL	KLバッファPID値(積分リミット)
KL=...	KLバッファPID値設定(積分リミット)
KP	KPバッファPID値(比例)
KP=...	KPバッファPID値設定(比例)
KS	KSバッファPID値(積分フィルタ制御)
KS=...	KSバッファPID値設定(積分フィルタ制御)
KV	KVバッファPID値(速度フィードフォワード)
KV=...	KVバッファPID値設定(速度フィードフォワード)
L	
LEN	主通信チャンネルバッファ占有レベル(データモード)
LEN1	2次通信チャンネルバッファ占有レベル(データモード)
LFS(...)	32ビットIEEEフォーマットFloat値
LOAD	モータへプログラムをダウンロード
LOCKP	新プログラムがロードされるまでプログラムアップロード防止
LOOP	WHILE 構文要素
M	
MC	カムモード有効
MC(...)	カムモード有効(追加軌道)
MCE(...)	カムスプライン有効
MCW(...)	カムスタートポイント
MDB	TOBコミュニケーション有効
MDC	正弦波電流コミュニケーションモード

MDE	台形エンコーダコミュニケーションモード
MDS	正弦波電圧コミュニケーションモード
MDT	台形ホールセンサーコミュニケーションモード
MF0	CTR(1)を0に設定し、外部エンコーダを4通倍モードに選択
MFA(...)	追従モード、立上り
MFD(...)	追従モード、立下り
MFDIV	追従モード除数
MFDIV=...	追従モード除数設定
MFMUL	追従モード乗数
MFMUL=...	追従モード乗数設定
MFR	4通倍追従モード選択
MFR(...)	4通倍追従モード選択(追加軌道)
MFSDC(...)	追従モード (ストール - 一時停止 - 継続)
MFSLEW(...)	追従モード 一定
MINV(...)	逆コミュニケーション
MODE	動作モード
MODE(...)	動作モード(特定軌道)
MP	位置モード有効
MP(...)	位置モード有効(追加軌道)
MS0	CTR(1)を0に設定し、外部エンコーダ使用パルス/方向モードに選択
MSR	パルス/方向モードで追従モードを選択
MSR(...)	パルス/方向モードで追従モードを選択(追加軌道)
MT	トルクモード有効
MTB	モードトルクブレーキ
MV	速度モード有効
MV(...)	速度モード有効(追加軌道)
O	
O=...	原点設定
O(...)=...	特定軌道の原点設定
OC(...)	出力状態(24V I/O)
OCHN(...)	通信チャンネルを開く
OF(...)	出力フォルト(24V I/O)
OFF	サーボオフ
OR(...)	出力をLowに設定
OS(...)	出力をHighに設定
OSH=...	原点シフト
OSH(...)=...	特定原点シフト
OUT(...)=...	1つもしくは複数の出力を特定の状態に設定
P	
PA	実際位置
PAUSE	プログラム実行を一時停止
PC	指令位置
PC(...)	指令位置(特定軌道)
PI	π 値を取得
PID1	16,000 Hz PIDレート
PID2	8,000 Hz PIDレート(デフォルト)
PID4	4,000 Hz PIDレート
PID8	2,000 Hz PIDレート
PMA	実際位置モジュール
PML	位置モジュールリミット

PML=...	位置モジュールリミット設定
PMT	位置モジュールターゲット(位置動作)
PMT=...	位置モジュールターゲット設定(位置動作)
PRA	動作開始実測位置取得
PRC	動作開始指令位置
PRINT(...)	主通信チャンネルにデータ出力
PRINT1(...)	2次通信チャンネルにデータ出力
PRT	相対目標位置
PRT=...	相対目標位置設定
PRTS=(...)	同期相対目標位置設定
PRTSS=(...)	追加同期相対目標位置設定
PT	目標位置
PT=	目標位置設定
PTS=(...)	同期絶対目標位置設定
PTSS=(...)	追加同期絶対目標位置設定
PTSD	同期直線動作距離
R	
Ra...Rzzz	ユーザ変数取得
Ra...Rzzz	ユーザ変数報告
Raaa...Rzzz	ユーザ変数報告
Rab[index]	8ビット配列変数報告
Raf[index]	Float型変数報告
Ral[index]	32ビット変数報告
Raw[index]	16ビット変数報告
RABS(...)	整数型絶対値報告
RAC	指令加速値報告
RACOS(...)	角度によるarc-cosine(コサインの逆関数)値報告
RADDR	モータシリアルアドレス報告
RAMPS	最大PWMリミット報告
RANDOM	乱数取得 例:a=RANDOM
RANDOM=...	乱数設定
RASIN(...)	角度によるarc-sin(サインの逆関数)値報告
RAT	目標加速値報告
RATAN(...)	角度によるarc-tan(タンジェントの逆関数)値報告
RATOF(...)	ASCIIをFloat型に変換して報告
RB()	ステータスビット報告
RBa	過電流ステータスビット報告
RBAUD(0)	チャンネル0ポーレート報告
RBAUD(1)	チャンネル1ポーレート報告
RBe	位置偏差エラーステータスビット報告
RBh	温度超過ステータスビット報告
RBi(0)	エンコーダZ相ステータスビット(内部エンコーダ、ライジング信号)報告
RBi(1)	エンコーダZ相ステータスビット(外部エンコーダ、ライジング信号)報告
RBj(0)	エンコーダZ相ステータスビット(内部エンコーダ、フォーリング信号)報告
RBj(1)	エンコーダZ相ステータスビット(外部エンコーダ、フォーリング信号)報告
RBk	EEPROMデータ完全性ステータスビット報告
RBl	ハードウェア左/負方向オーバーラベルリミット履歴ビット報告
RBls	ソフトウェア左/負方向オーバーラベルリミット履歴ビット報告
RBm	左/負方向ハードウェアオーバーラベルリミットビット報告
RBms	左/正方向ソフトウェアオーバーラベルリミットビット報告

RBo	サーボオフステータスビット報告
RBp	右/正方向ハードウェアオーバートラベルリミットビット報告
RBps	右/正方向ソフトリミットオーバートラベルリミットビット報告
RBr	ハードウェア右/正方向オーバートラベルリミット履歴ビット報告
RBrS	ソフトウェア右/正方向オーバートラベルリミット履歴ビット報告
RBs	シンタックスエラーステータスビット報告
RBt	動作進行中ステータスビット報告
RBv	速度エラービット報告
RBw	32bit位置カウンタラップ・アラウンドステータスビット報告
RBx(...)	リアルタイム内部インデックス入力ステータスビット報告
RCADDR	CANアドレス報告
RCAN	CANエラー報告
RCBAUD	CANボーレート報告
RCHN(0)	RS-232通信エラーフラグ報告
RCHN(1)	RS485 通信エラーフラグ報告
RCKS	プログラムチェックサム報告
RCLK	1ミリ秒クロック変数報告
RCOS(...)	コサインを角度で報告
RCP	カムポインタ報告
RCTR(0)	プライマリエンコーダ/パルス列 & 方向カウンタ報告
RCTR(1)	セカンドエンコーダ/パルス & 方向カウンタ報告
RCTT	EEPROM内カムテーブル数報告
RDEA	実行de/dt値報告
RDEL	設定de/dtフォルトリミット報告
RDFS	32ビットIEEEフォーマット内、af[]変数報告
RDT	設定減速値報告
REA	位置偏差エラー報告
REL	位置偏差エラーフォルトリミット報告
REPTR	EEPROMポインタデータ報告
RERRC	直近コマンドエラーコード報告
RERRW	直近コマンドエラー通信チャンネル報告
RES	エンコーダ分解能報告 例: a=RES
RESUME	一時停止後プログラムを実行再開
RETURN	サブルーチンから戻る
RETURNI	割り込みルーチンから戻る
RFABS(...)	浮動小数点絶対値エラー報告
RFSQRT(...)	浮動小数点平方根報告
RFW	ファームウェアバージョン報告
RGETCHR	主通信チャンネルから文字列報告
RGETCHR1	2次通信チャンネルから文字列報告
RHEX(...)	変数をhex列で報告
RI(0)	エンコーダZ相入力位置変数報告(ライジングエッジ、内部エンコーダ)
RI(1)	エンコーダZ相入力位置変数報告(ライジングエッジ、外部エンコーダ)
RIN(...)	I/O入力報告
RINA(...)	アナログ入力報告
RJ(0)	エンコーダZ相入力位置変数報告(フォーリングエッジ、内部エンコーダ)
RJ(1)	エンコーダZ相入力位置変数報告(フォーリングエッジ、外部エンコーダ)
RKA	KAバッファPID値(加速フィードフォワード)報告
RKC	KC値報告
RKCS	KCS値報告

RKD	KDバッファPID値(微分)報告
RKG	KGバッファPID値(重力)報告
RKI	KIバッファPID値(積分)報告
RKL	KLバッファPID値(積分リミット)報告
RKP	KPバッファPID値(比例)報告
RKS	KSバッファPID値(積分フィルタ制御)報告
RKV	KVバッファPID値(速度フィードフォワード)報告
RLEN	主通信チャンネルバッファ占有レベル(データモード)報告
RLEN1	2次通信チャンネルバッファ占有レベル(データモード)報告
RLFS(...)	32ビットIEEEフォーマットFloat値報告
RMFDIV	追従モード除数報告
RMFMUL	追従モード乗数報告
RMODE	動作モード報告
RMODE(...)	動作モード(特定軌道)報告
ROC(...)	出力状態(24V I/O)報告
ROF(...)	出力フォルト(24V I/O)報告
RPA	実行位置報告
RPC	指令モータ位置報告
RPC(...)	指令モータ位置(特定軌道)報告
RPI	pi数値を報告
RPMA	実行位置モジュール報告
RPML	位置モジュールリミット報告
RPMT	位置モジュールターゲット(位置動作)報告
RPRA	動作開始実測位置報告
RPRC	動作開始指令位置報告
RPRT	相対目標位置報告
RPT	設定目標位置報告
RPTSD	同期直線動作距離報告
RPTST	同期動作時間報告(ms)
RRANDOM	乱数報告
RRES	エンコーダ分解能報告
RSAMP	サンプルレート(Hz)報告
RSIN(..)	サイン関数を角度で報告
RSLM	ソフトリミットモード報告
RSLN	ソフトリミット右/正方向設定報告
RSLP	ソフトリミット左/負方向設定報告
RSP	サンプルレート、ファームウェア番号報告
RSP1	ファームウェアコンパイル日時報告
RSP2	ブートローダーリビジョン報告
RSQRT(...)	整数平方根値報告
RT	リクエストトルク値報告
RTAN(...)	タンジェント関数を角度で報告
RTEMP	内部温度報告
RTH	温度限界設定値報告
RTHD	電流制限タイマー設定値報告
RTMR(...)	ユーザタイマー値報告
RTRQ	リアルタイムトルク値報告
RTS	トルク傾斜設定値報告
RUIA	電流値報告
RUJA	電圧値報告

RUN	プログラム実行
RUN?	電源投入後RUNコマンドが送られない限りプログラム停止
RVA	実行速度報告（フィルター値）
RVC	指令速度値報告
RVL	速度リミット報告
RVT	指定速度報告
RW(...)	特定のステータスワード報告
S	
S	急停止
S(...)	急停止(軌道指定)
SADDR#	モータのアドレス指定
SAMP	サンプリングレート(Hz)
SILENT	ポート1で受信するコマンドを無視
SILENT1	ポート2で受信するコマンドを無視
SIN(...)	サイン関数を角度で取得
SLD	ソフトウェアリミットを無効
SLE	ソフトウェアリミットを有効
SLEEP	ポート1でスリープモードを開始
SLEEP1	ポート2でスリープモードを開始
SLM	ソフトウェアリミット
SLM(...)	ソフトウェアリミットを設定
SLN	左方向ソフトウェアリミット
SLN=...	左方向ソフトウェアリミットを設定
SLP	右方向ソフトウェアリミット
SLP=...	右方向ソフトウェアリミット設定
SQRT(...)	整数平方根
SRC(...)	追従and/orカムエンコーダソース設定
STACK	ネスティングリセット
STDOUT=...	報告コマンドの出力先を設定
SWITCH...	プログラムフローコマンド
T	
T	設定トルク値
T=	トルク値設定
TALK	ポート1へのPRINTメッセージ出力を有効
TALK1	ポート2にPRINTメッセージ出力を有効
TAN(...)	タンジェント関数を角度で取得
TEMP	温度
TH	温度リミット
TH=...	温度リミット設定
THD	温度超過(Bh)時、駆動電源OFFまでのタイマー取得
THD=	温度超過(Bh)時、駆動電源OFFまでのタイマー設定
TMR(...)	特定タイマー設定時間、例: a=TMR(0)
TMR(...) (as cmd)	タイマー設定、例: TMR(0,1000) = タイマー0を一秒に設定
TS	トルク傾斜設定値
TS=...	トルク傾斜設定
TSWAIT	同期動作中 待機
TWAIT	動作中 待機
TWAIT(...)	動作中 待機 (軌道指定)
U	
UIA	モータ電流値

UJA	バス電圧
UO(...)=...	1個以上のステータスビットを特定の値に設定
UP	ユーザEEPROMプログラムをアップロード
UPLOAD	ユーザEEPROMリーダープログラムをアップロード
UR(...)	1個以上のステータスビットを0に設定
US(...)	1個以上のステータスビットを1に設定
V	
VA	実行速度
VAC(...)	速度フィルタ設定
VC	指定速度
VL	指定速度制限
VL=...	速度制限設定
VLD(...)	不揮発メモリからの読み出し
VST(...)	不揮発メモリへの書き込み
VT	目標速度
VT=...	目標速度を設定
VTS=...	同期動作の目標速度を設定
W	
W(...)	ステータスワードを報告
WAIT=...	指定時間待機 (msec単位で指定)
WAKE	ポート1のスリープモードを終了
WAKE1	ポート2のスリープモードを終了
WHILE...	無限ループ コマンド
X	
X	減速停止
X(...)	減速停止 (軌道を指定)
Z	
Z	全エラーリセット
Z(...)	特定のエラーをリセット
Za	電流エラーリセット
Ze	偏差エラーリセット
Zh	温度エラーリセット
Zl	ハードウェア(センサ入力)リミット履歴 左方向ステータスビット 解除
Zls	ソフトウェアリミット履歴 左方向ステータスビット 解除
Zr	ハードウェア(センサ入力)リミット履歴 右方向ステータスビット 解除
Zrs	ソフトウェアリミット履歴 右方向ステータスビット 解除
Zs	システックエラービット 解除
ZS	全エラー解除
Zv	速度エラー解除
Zw	エンコーダ値ラップラウンド ラッチビット リセット