

インバータ

MX2/RXシリーズ用

DriveProgramming

ユーザーズマニュアル

形3G3MX2シリーズV1タイプ

形3G3MX2シリーズV2タイプ

形3G3RXシリーズV1タイプ

形CX-Drive



おことわり

- (1) 本マニュアルの内容の一部または全部を無断で複写、複製、転載することを禁じます。
- (2) 本マニュアルの内容に関しては、改良のため予告なしに仕様などを変更することがあります。
あらかじめご了承ください。
- (3) 本マニュアルの内容に関しては、万全を期しておりますが、万一ご不審な点や誤りなど、お気づきのことがありましたら、お手数ですが弊社支店または営業所までご連絡ください。
その際、巻末記載の**Man.No.** (マニュアル**No.**)も併せてお知らせください。

商標

- ・ **Sysmac**は、オムロン株式会社**FA**機器製品の日本およびその他の国における商標または登録商標です。
- ・ **Microsoft, Windows, Excel, Visual Basic, Microsoft Edge**は、米国**Microsoft Corporation**の米国およびその他の国における登録商標または商標です。
- ・ **EtherCAT®** は、ドイツ**Beckhoff Automation GmbH**によりライセンスされた特許取得済み技術であり登録商標です。
- ・ **ODVA, CIP, CompoNet, DeviceNet, EtherNet/IP**は**ODVA**の商標です。

その他、本文中に掲載しているシステム名および製品名は、それぞれ各社の商標または、登録商標です。

はじめに

このたびは、インバータ／サーボ用サポートツール CX-Drive または形 3G3MX2 シリーズ V1 タイプ／3G3MX2 シリーズ V2 タイプ／3G3RX シリーズ V1 タイプをお買い上げいただき、ありがとうございます。

このユーザーズマニュアルは、インバータ用 DriveProgramming の仕様、操作方法について記載しています。

なお、ご使用の際は、多機能型小型インバータ 形 3G3MX2 シリーズ V1 タイプ ユーザーズマニュアル (SBCE-376)、多機能型小型インバータ 形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル (SBCE-506)、高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル (SBCE-367)、および CX-Drive オペレーションマニュアル (SBCE-375) を併せてご覧ください。

対象となる読者の方々

このユーザーズマニュアルは、次の方を対象に記述しています。

電気の知識（電気工事士あるいは同等の知識）を有する方で、

- FA 機器の導入を担当される方
- FA システムを設計される方
- FA 機器を設置、接続される方
- FA 現場を管理される方

お願い

このユーザーズマニュアルは、DriveProgramming の使用方法について必要な情報を記載しています。

お使いになる前に、このユーザーズマニュアルをよく読んで、十分に理解してください。

また、お読みになった後もお使いになるときはいつでも見られるように、このユーザーズマニュアルを大切に保管してください。

なお、このユーザーズマニュアルは最終的にお使いになる方のお手元にも届けられるようお取り計らいをお願いします。

章構成

このユーザーズマニュアルは、お客様の用途に合わせて以下に示すように構成されています。

章		概要
第 1 章	概要	この章では、DriveProgramming の概要やシステム構成について説明しています。
第 2 章	仕様	この章では、DriveProgramming の仕様について説明します。
第 3 章	DriveProgramming の使用手順	この章では、DriveProgramming の使用手順および、関係するパラメータとプログラムの構造について説明します。
第 4 章	DriveProgramming エディタ	この章では、DriveProgramming エディタの起動、保存と読み出し、各部の詳細について説明します。
第 5 章	DriveProgramming のユーザ変数	この章では、DriveProgramming のユーザ変数について説明します。
第 6 章	DriveProgramming 命令	この章では、DriveProgramming の命令について説明しています。
第 7 章	DriveProgramming で使用するパラメータの注意事項	この章では、DriveProgramming で使用するパラメータの注意事項を説明します。
第 8 章	異常とその処理	この章では、異常発生時のプログラムの動作、DriveProgramming 機能固有の異常と原因、その処理方法について説明しています。

マニュアルの読み方

ページ構成

このユーザーズマニュアルの各ページの構成と記号は以下のようになっています。

章タイトル

節タイトル

項タイトル

8 異常とその処理

8-1 トラブルシューティング

8-1-1 異常発生時の DriveProgramming動作

異常発生時の DriveProgrammingの動作や、DriveProgramming機能固有の異常コードとその処理方法について説明しています。

エラー内容	ユーザトリップ E50 ~ E59	DriveProgramming関連 トリップ E43 ~ E45	その他のトリップ
on trip goto の有無			
なし	プログラム停止	プログラム停止	そのまま 動作継続
あり	指定ラベルに飛び、 動作継続	プログラム停止	指定ラベルに飛び、 動作継続

使用上の注意
多機能 I/O を使用中にプログラムが停止した場合、停止直前の状態を保持します。

8-1-2 異常リセット時の DriveProgramming動作

異常リセットした時の DriveProgrammingの動作は、リセット選択(C102)の設定により異なります。動作の状態を、以下の表に示します。

プログラム状態	停止	動作中
C102 の設定		
00 ~ 02	最初から動作	最初から動作
03	停止 ^{*1}	動作継続

^{*1} プログラムを運転するには、PRG 離す OFF ON (A017=01 秒) するが、DriveProgramming機能選択 (A017) を 00 (無効) に設定したあと、02 (常時有効) に設定してください。

8-1-3 異常コード一覧

DriveProgramming機能固有の異常コードと原因、その処理方法について説明しています。その他の異常は、『高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル』(SBCE-367) を参照してください。

異常コード	異常 (インバータのトリップ発生要因)	考えられる原因	チェック方法	処理方法
E43	無効な命令	プログラムがインバータにダウンロードされていないのに、PRG 離すが ON になりました。	プログラムをアップロードすることにより、実際にプログラムがインバータ内に存在するがチェックします。	プログラムを再作成して、インバータにダウンロードしてください。

8 - 2

DriveProgramming ユーザーズマニュアル (SBCE-369)

マニュアル名称

操作手順

操作手順を示します。

注記、補足、参照先

注記、補足、参照先などの各項目を、アイコンによって示しています。



4 DriveProgrammingエディタ

4-7 テキストエディタによるプログラムの作成

4-7-1 プログラムの作成

以下の手順に従って、ドライブプログラムを作成してください。

- 1 プログラムエディタを開きます。ドライブプログラミングの補助ウィンドウ (ツールボックス、ブロックパラメータ、プロパティ、および異常一覧) が自動的に表示されます。
- 2 コードの編集方法は3つあります。
(a) 手動入力
(b) コード一覧の呼び出し (Ctrl + スペース)
(c) ツールボックスウィンドウから箱をドラッグ&ドロップ
- 3 プログラムのコンパイルを実行し異常がなければ、ドライブへの転送、エクスポートなどを実行できます。

使用上の注意
テキストおよびフローチャートタスク/サブルーチンは、同じプログラム内で同時に使用できます。

4

DriveProgramming ユーザーズマニュアル (SBCE-369)

4 - 17

節タイトル

現在のページの節タイトルを示します。

章番号

現在のページの章番号を示します。

項タイトル

現在のページの項タイトルを示します。

(注) このページは、説明用の見本です。実際の内容とは異なります。

アイコン

このユーザーズマニュアルで使われているアイコンには、次のような意味があります。



安全上の要点

製品を安全に使用するために実施または回避すべきことを示します。



使用上の注意

製品の動作不能、誤動作、または性能・機能への悪影響を予防するために実施または回避すべきことを示します。



参考

必要に応じて読んでいただきたい項目です。
知っておくと便利な情報や、使用する上で参考となる内容について説明しています。



詳しい情報や、関連情報の記載されている場所を示します。



バージョン関連情報

対象インバータや CX-Drive の、バージョンによる性能や機能の違いについて説明しています。

目次構成

		1
1	概要	2
2	仕様	3
3	DriveProgramming の使用手順	4
4	DriveProgramming エディタ	5
5	DriveProgramming のユーザ変数	6
6	DriveProgramming 命令	7
7	DriveProgramming で使用するパラメータの注意事項	8
8	異常とその処理	

目次

はじめに	1
章構成	2
マニュアルの読み方	3
目次構成	5
目次	6
ご承諾事項	9
安全上の注意	10
安全上の要点	13
法規と規格	14
関連マニュアル	15
マニュアル改訂履歴	16

第 1 章 概要

1-1 DriveProgramming の概要	1-2
1-2 準備とシステム構成	1-4

第 2 章 仕様

2-1 仕様	2-2
--------------	-----

第 3 章 DriveProgramming の使用手順

3-1 使用手順	3-2
3-2 DriveProgramming に関連するパラメータ	3-5
3-3 プログラムの構造	3-12
3-3-1 タスクの機能	3-12
3-3-2 サブルーチンの機能	3-12
3-3-3 タスクの処理	3-12
3-3-4 DriveProgramming 機能の起動／停止とタスク動作	3-14
3-3-5 DriveProgramming 機能の再起動	3-16
3-3-6 トリップ発生時のタスク動作	3-17

第 4 章 DriveProgramming エディタ

4-1 DriveProgramming エディタの起動	4-2
4-2 各部の説明	4-6
4-2-1 DriveProgramming エディタ	4-6
4-2-2 ツールバー	4-6

4-2-3	DriveProgramming 領域.....	4-9
4-2-4	コマンドボックスウィンドウ.....	4-13
4-2-5	ユーザパラメータウィンドウ.....	4-14
4-2-6	プロパティウィンドウ.....	4-15
4-2-7	アウトプットウィンドウのエラーリストタブ.....	4-16
4-3	タスクの追加・削除と名前の変更.....	4-17
4-4	サブルーチンの挿入・削除と呼び出し.....	4-18
4-5	フローチャートプログラムの作成.....	4-19
4-6	テキストプログラムの作成.....	4-20
4-7	転送（アップロード）したプログラムの編集.....	4-21
4-8	プログラムの保存.....	4-22
4-9	プログラムの転送と照合.....	4-24
4-10	プログラムの実行（DriveProgramming 機能選択）.....	4-25
4-11	その他の便利な機能.....	4-28

第 5 章 DriveProgramming のユーザ変数

5-1	ユーザ用の変数とユーザパラメータ.....	5-2
5-2	入出力端子変数.....	5-5
5-3	タイマ変数.....	5-10
5-4	インバータ設定変数.....	5-12
5-5	インバータモニタ変数.....	5-14
5-6	多機能入力変数.....	5-17
5-7	多機能出力変数.....	5-20

第 6 章 DriveProgramming 命令

6-1	命令のカテゴリ.....	6-2
6-2	命令のフォーマット.....	6-3
6-3	命令一覧.....	6-4
6-4	プログラム制御命令.....	6-10
6-5	算術演算命令と論理演算命令.....	6-23
6-6	入出力制御命令.....	6-36
6-7	タイマ制御命令.....	6-46
6-8	パラメータ制御命令.....	6-52
6-9	インバータ制御命令.....	6-58

第 7 章 DriveProgramming で使用するパラメータの注意事項

7-1	設定順序で影響を受けるインバータパラメーター一覧.....	7-2
7-2	定格電流 (%) による影響を受けるパラメーター一覧.....	7-4
7-3	PID の有効／無効による影響を受けるパラメーター一覧.....	7-5

第 8 章 異常とその処理

8-1	トラブルシューティング	8-2
8-1-1	異常発生時の DriveProgramming 動作	8-2
8-1-2	異常リセット時の DriveProgramming 動作	8-3
8-1-3	アラームコード一覧	8-4

ご承諾事項

● 保証内容

(1) 保証期間

本ソフトウェアの保証期間は、ご購入後またはご指定の場所に納入後 1 年といたします。
ただし「カタログ等」に別途記載がある場合を除きます。

(2) 保証範囲

- ・ 本ソフトウェアの使用許諾を受けたお客様が、上記保証期間中にコンピュータ・プログラムの瑕疵（マニュアルとの重要な不一致）を発見し、当社に返却した場合は、当社は瑕疵（当該マニュアルとの重要な不一致）のないコンピュータ・プログラムを記録した媒体と交換いたします。{もしくは、当社の選択により、当社ホームページより瑕疵（当該マニュアルとの重要な不一致）のないコンピュータ・プログラムをダウンロードしていただく方法により提供いたします。} また、当社の責任によるコンピュータ・プログラムの記録媒体の不良を発見し、当社に返却した場合、当社は、無償で、良品の媒体に記録したコンピュータ・プログラムと交換いたします。
- ・ 万一、当社がコンピュータ・プログラムの瑕疵（マニュアルとの重要な不一致）を除去できないと判断した場合は、お客様が本ソフトウェア購入代金として支払った金額をお返しいたします。

● 責任の制限

- ・ 前条に定める交換または購入代金の返金は、本ソフトウェアの保証責任のすべてを定めるものであり、当社は本ソフトウェアの瑕疵により発生した、お客様の直接的、間接的あるいは波及効果による損害等いかなる損害に対しても一切の責任を負いません。
- ・ 当社は、本ソフトウェアを当社以外の第三者が変更、改造することにより生じた瑕疵につきましては、一切責任を負いません。
- ・ 当社は、本ソフトウェアに基づき、当社以外の第三者が開発したソフトウェアおよびそれにより生じた結果について一切の責任を負いません。

● 本ソフトウェアの用途

本ソフトウェアを本マニュアルに記載の用途以外の用途で使用しないでください。

● 仕様の変更

本ソフトウェアの仕様および付属品は改善またはその他の事由により、必要に応じて、変更される場合があります。

● 適用範囲

以上の内容は、日本国内での取引および使用を前提としております。日本国外での取引および使用に関しては、当社営業担当者までご相談ください。

安全上の注意

安全に使用していただくための表示と意味

このユーザーズマニュアルでは、DriveProgramming を安全に使用していただくために、注意事項を次のような表示と図記号で示しています。

ここで示した注意事項は、安全に関する重大な内容を記載しています。必ず守ってください。
表示と意味は次のとおりです。

警告表示の意味



警告

正しい取り扱いをしなければ、この危険のために、軽傷や中程度の傷害を負ったり、万一の場合には重傷や死亡に至るおそれがあります。また、同様に重大な物的損害を受けるおそれがあります。



注意

正しい取り扱いをしなければ、この危険のために、時に軽傷や中程度の傷害を負ったり、あるいは物的損害を受けるおそれがあります。

図記号の説明



⊘ 記号は禁止（してはいけないこと）を示します。
具体的な内容は、⊘ の中や近くに絵や文章で示します。
左図の場合は、「分解禁止」を示します。



⚡ 記号は危険、注意を示します。
具体的な内容は、⚡ の中や近くに絵や文章で示します。
左図の場合は、「感電による危険」を示します。



⚠ 記号は危険、注意を示します。
具体的な内容は、⚠ の中や近くに絵や文章で示します。
左図の場合は、「特定しない一般的な危険」を示します。



⚠ 記号は、注意（警告を含む）を意味しています。
具体的な内容は、⚠ の中と文章で示します。
左図の場合は、「高温注意」を表します。



● 記号は強制を意味しています。
具体的な内容は、● の中と文章で示します。
左図の場合は、「一般的な強制事項」を表します。



● 記号は強制（必ず守ること）を示します。
具体的な内容は、● の中と文章で示します。
左図の場合は、「必ずアース接地すること」を表します。

⚠ 警告

	万一の場合、感電による重度の傷害が起こるおそれがあります。 入力電源 OFF を確認してから正しく配線してください。
	万一の場合、感電による重度の傷害が起こるおそれがあります。 配線作業は、電気工事の専門家が行ってください。
	万一の場合、感電による重度の傷害が起こるおそれがあります。 配線変更、スイッチの変更、オプション類の脱着はインバータの入力電源を OFF にしてから行ってください。
	万一の場合、感電・発火による重度の傷害が起こるおそれがあります。 接地端子は必ずアースしてください。 (200V 級：D 種接地、400V 級：C 種接地)
	万一の場合、感電による重度の傷害が起こるおそれがあります。 通電中および電源遮断後 10 分以内はフロントカバーや端子台カバーを外さないでください。
	万一の場合、感電による重度の傷害が起こるおそれがあります。 濡れた手でオペレータ、スイッチ類を操作しないでください。
	万一の場合、感電による重度の傷害が起こるおそれがあります。 STO 機能が働いた状態になっても、主電源が遮断されたわけではありません。 製品の確認は、インバータの入力電源を OFF にしてから行ってください。
	やけどのおそれがあります。 通電中や電源遮断後のしばらくの間は、インバータの冷却フィン、制動抵抗器、モータなどは高温になる場合がありますので触れないでください。

⚠ 警告

	本ソフトウェアを使用するパソコンにウィルス対策ソフトを導入し、パソコンがウィルスに感染するリスクを低減させてください。 また、ウィルス対策ソフトは常に最新の状態にしてください。
	OS の脆弱性に起因するセキュリティリスクを回避するために、本ソフトウェアを使用するパソコンの OS を常に最新の状態にしてください。
	機能の追加や操作性の改善、セキュリティの強化のため、本ソフトウェアは、常に最新のバージョンにアップデートしてお使いください。
	本ソフトウェアで設定するユーザ名やパスワードについて、他人から不正に利用されないよう適切な設定・管理をしてください。
	制御システムや装置のネットワークはファイアウォール（未使用通信ポートの遮断、通信ホストの制限）を導入し、IT ネットワークからの分離を行い、本ソフトウェアの制御システムへの接続はファイアウォール内で行うようにしてください。
	本ソフトウェアから制御システムや装置へのリモートアクセスが必要な場合は仮想プライベートネットワーク (VPN) を使用してください。

 注意	
	軽度の発火、発熱、機器破損がまれに起こるおそれがあります。 端子 [+1, P/+2] と端子 [N/-] に抵抗器を直接接続しないでください。
	軽度の傷害がまれに起こるおそれがあります。 安全を確保するための停止装置を設置してください。 ※ 保持ブレーキは安全を確保するための停止装置ではありません。
	軽度の傷害がまれに起こるおそれがあります。 出力端子機能が出力中に、DriveProgramming が停止すると、出力状態を保持したままになりますので、周辺機器を停止させるなど安全対策を施してください。
	立上げ時、調整時、メンテナンス時、交換時にパラメータを誤って設定すると予期せぬ動作が起きるおそれがあります。十分な確認を行ったあと、運転に移行してください。
	制動抵抗器や回生制動ユニットの発熱により、中程度のやけどがまれに起こるおそれがあります。 必ず指定された制動抵抗器や回生制動ユニットを使用し、制動抵抗器を使用する場合には、抵抗器の温度を監視するサーマルリレーを設置してください。 また、制動抵抗器や回生制動ユニットの異常過熱時にインバータの電源を OFF するシーケンスを組んでください。
	製品内部には高電圧部分があり、短絡させると製品の破損や物的損害がまれに起こるおそれがあります。設置や配線時には切り粉やリード線くずなどの金属物が製品内部に入らないようにカバーをつけるなどの処置を行ってください。
	負荷の短絡により物的損害がまれに起こるおそれがあります。 インバータの電源側にはインバータ容量に応じた配線用遮断機 (MCCB) を設置するなどの安全対策を施してください。
	けがのおそれがあります。 分解や修理、改造は行わないでください。

安全上の要点

運転・調整について

- DriveProgramming で時刻機能を使用して制御を行う場合、LCD デジタルオペレータの電池消耗で予期せぬ動作が起きるおそれがあります。
電池消耗は、時刻データが初期値に戻ったことで検出し、インバータまたはプログラムを停止させるなどの対策を行ってください。
LCD デジタルオペレータが取り外されたり、断線したりした場合、DriveProgramming は時刻機能で待機状態になります。
- DriveProgramming を使用される場合は、プログラムが正常にダウンロードできたことを確認してから動作を開始してください。
- 本製品は低速から高速までの設定ができますので、使用するモータ設備の許容範囲を十分に確認したあとで運転を行ってください。

法規と規格

本製品のうち、外国為替および外国貿易管理法に定める輸出許可、承認対象貨物（または技術）に該当するものを輸出（または非居住者に提供）する場合は、同法に基づく輸出許可、承認（または役務取引許可）が必要です。

関連マニュアル

この商品进行操作するにあたり、接続する機器の情報が必要になります。
関連商品の情報は、下記マニュアルを参照してください。

名称	マニュアル番号
多機能型小型インバータ 形 3G3MX2 シリーズ V1 タイプ ユーザーズマニュアル	SBCE-376
多機能型小型インバータ 形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル	SBCE-506
高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル	SBCE-367
CX-Drive オペレーションマニュアル	SBCE-375



参考

インバータの動作に関しては、本体の □『多機能型小型インバータ 形 3G3MX2 シリーズ V1 タイプ ユーザーズマニュアル (SBCE-376)』、□『多機能型小型インバータ 形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル (SBCE-506)』、□『高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル (SBCE-367)』を参照してください。

マニュアル改訂履歴

マニュアル改訂記号は、表表紙・裏表紙に記載されている Man.No. の末尾にアルファベットで追記されています。

例

Man.No.	SBCE-369F
---------	------------------

↑ 改訂記号

改訂記号	改訂日	改訂内容と改訂ページ
A	2012 年 11 月	初版
B	2013 年 3 月	3G3MX2 シリーズ V1 タイプの仕様を追加
C	2014 年 5 月	・ 誤記修正 ・ 説明改善
D	2022 年 9 月	セキュリティに関する安全上の注意追記に伴う改訂
E	2023 年 10 月	3G3MX2 シリーズ V2 タイプを追加
F	2024 年 10 月	誤記修正

概要

この章では、DriveProgramming の概要やシステム構成について説明しています。

1-1 DriveProgramming の概要	1-2
1-2 準備とシステム構成	1-4

1-1 DriveProgramming の概要

DriveProgramming はインバータ内蔵の簡易シーケンス機能です。

シーケンスプログラム作成や状態確認にはインバータ／サーボ用サポートツール CX-Drive を使用します。

作成したプログラムを形 3G3MX2 シリーズ V1 タイプ／3G3MX2 シリーズ V2 タイプ／3G3RX シリーズ V1 タイプインバータに転送（ダウンロード）することで簡単なシーケンス制御をインバータ単体で実行することができます。

DriveProgramming の特長

DriveProgramming には、以下のような特長があります。

- DriveProgramming は、フローチャート方式およびテキスト言語方式でのプログラミングが可能です。
- プログラムは、タスク単位（最大 5 タスク）に分けて作成することができます。
- 5 つのタスクの並列処理が可能です。
- 多機能入力端子の設定で、外部からユーザプログラムを実行させることが可能です。
- 多機能入力端子、ならびに多機能出力端子をプログラムに割り付けて使用することが可能です。
- 周波数設定や加減速時間など、現場で調整が必要なプログラム定数（変数）は、ユーザパラメータ（P100～P131）を指定しておけば、パソコンをつながなくてもデジタルオペレータから変更できます。
- ユーザプログラムは、インバータ本体内部の EEPROM に記憶されるので、インバータの電源投入後すぐにプログラムを起動することも可能です。
- オプションの LCD デジタルオペレータ（形 3G3AX-OP05）には、時刻機能を内蔵しています。LCD デジタルオペレータをご購入いただき接続すれば、時刻機能を使用したプログラムを作成できます。

【MX2】

3G3MX2 シリーズは、LCD デジタルオペレータ（形 3G3AX-OP05）との接続ができません。このため、時刻機能は使用できません。



安全上の要点

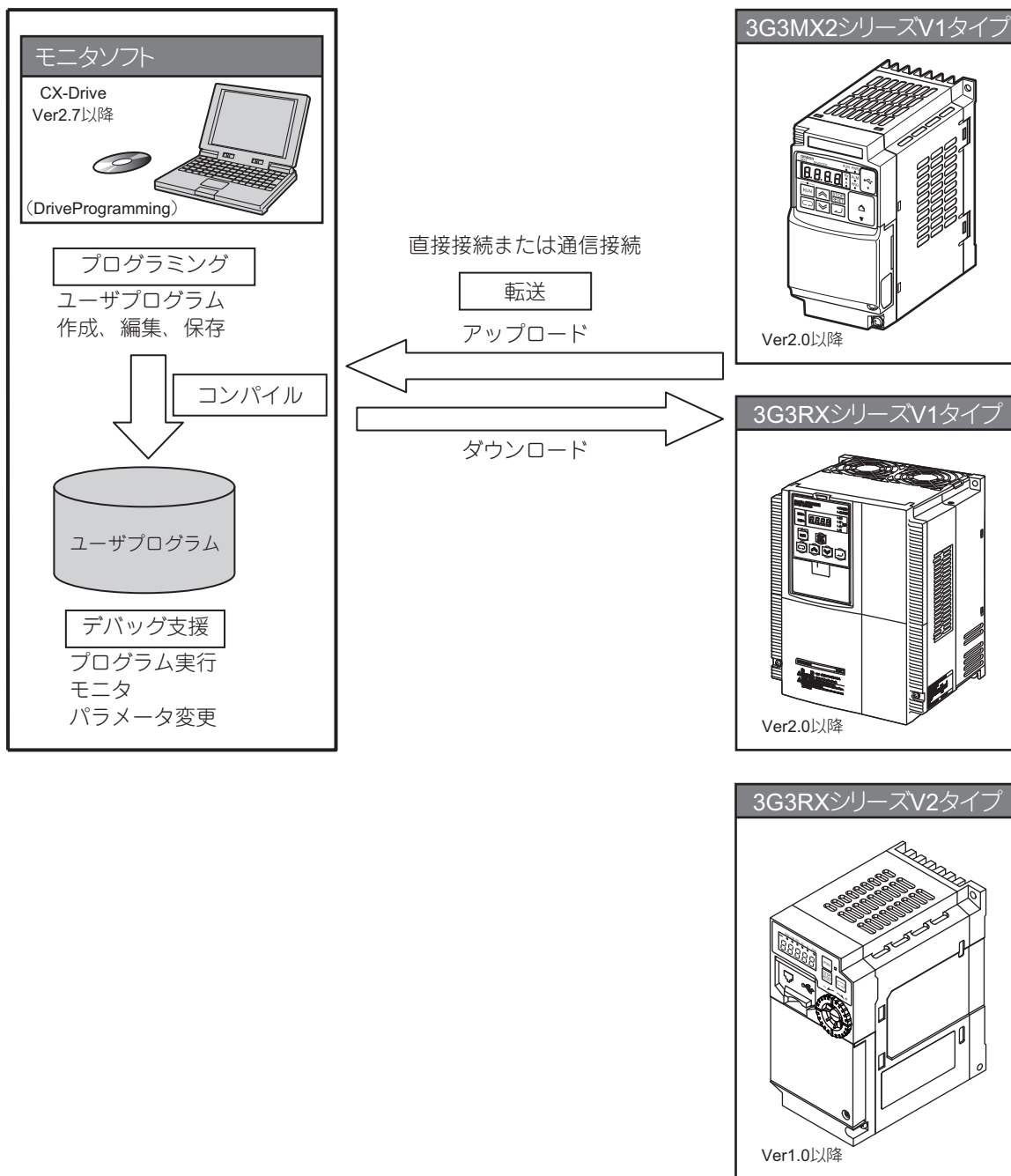
DriveProgramming で時刻機能を使用して制御を行う場合、LCD デジタルオペレータの電池消耗で予期せぬ動作が起きるおそれがあります。

電池消耗は、時刻データが初期値に戻ったことで検出し、インバータまたはプログラムを停止させるなどの対策を行ってください。

LCD デジタルオペレータが取り外されたり、断線したりした場合、DriveProgramming は時刻機能で待機状態になります。

以下に、CX-Drive の DriveProgramming エディタで実行できる主な機能を示します。

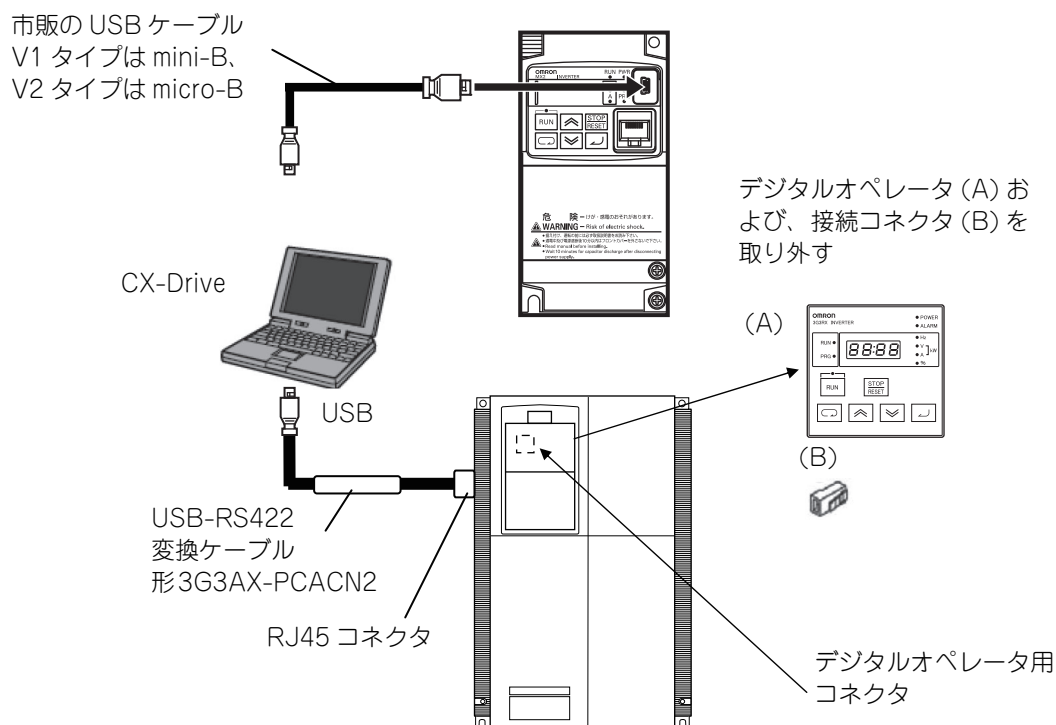
機能	説明
プログラミング	ユーザプログラムの作成、編集、保存、読込、印刷に対応しています。
コンパイル	ユーザプログラムのプログラムチェックと中間コード作成を行います。
転送	ユーザプログラムをインバータにダウンロードします。 または、ユーザプログラムをインバータからアップロードします。
デバッグ支援	プログラムの実行、停止を行います。 インバータ状態モニタなどを確認できます。



1-2 準備とシステム構成

CX-Drive の DriveProgramming 機能にてプログラムを作成し、形 3G3MX2 シリーズ V1 タイプ／3G3MX2 シリーズ V2 タイプ／3G3RX シリーズ V1 タイプインバータでこのプログラムを実行するためには、以下の準備が必要です。

- 形 3G3MX2 ／ 3G3RX シリーズ V1 タイプインバータ本体（Ver2.0 以降）／ 3G3MX2 シリーズ V2 タイプ
- パーソナルコンピュータ (PC) (Windows システム)
- CX-Drive は以下の Ver. が必要です。
 【MX2】 3G3MX2 シリーズ V1 タイプで Ver.2.8 以降、V2 タイプで Ver.3.02 以降
 【RX】 3G3RX シリーズ V1 タイプで Ver.2.7 以降
 (CX-Drive は、FA 総合ツールパッケージ CX-One に同梱されています。)
- PC- インバータ接続ケーブルとして以下のケーブルを準備してください。
 【MX2】 市販の USB ケーブル (V1 タイプは mini-B、V2 タイプは micro-B) が必要になります。
 【RX】 オプションの USB から RJ-45 への変換ケーブル (形 3G3AX-PCACN2) が必要になります。



バージョン関連情報

- 本 DriveProgramming 機能は形 3G3MX2 シリーズ V1 タイプ／ 3G3MX2 シリーズ V2 タイプ／ 3G3RX シリーズ V1 タイプに搭載しています。
従来機種 of 形 3G3MX2 ／ 3G3RX シリーズには搭載していません。
- 本 DriveProgramming 機能は CX-Drive の以下に示すバージョンから搭載します。
CX-One のバージョンアップを実行していただき、下記のバージョンで対応してください。
 【MX2】 V1 タイプは Ver.2.8 以降 (Ver.2.7 以前では操作できません。)、V2 タイプは Ver.3.02 以降
 【RX】 Ver.2.7 以降 (Ver.2.6 以前では操作できません。)

**参考**

CX-Drive のインストールおよびバージョンアップの方法については □『CX-Drive のオペレーションマニュアル』(SBCE-375) を参照してください。

2

仕様

この章では、DriveProgramming の仕様について説明します。

2-1 仕様	2-2
--------------	-----

2-1 仕様

DriveProgramming に関連する仕様を示します。

項目		仕様	
プログラム仕様	プログラミング言語	フローチャート方式およびテキスト言語方式	
	入力機器	Windows パersonalコンピュータ	
	プログラム容量	最大 1024 ステップ: 6k バイト (5 タスクトータルで最大 1024 ステップ)	
	プログラミングサポート機能	インバータ/サーボ用サポートツール CX-Drive でのサポート ・編集と表示 ・プログラムコンパイル (プログラム構文チェック) ・プログラムのダウンロード、アップロード、オールクリア	
	実行形式	・インタープリタ方式 ・実行周期: 2ms / ステップ (5 タスク並行処理で 5 命令実行可能) ・サブルーチンコールが可能 (最大 8 ネスト)	
入出力機能	外部入力	DriveProgramming 起動	DriveProgramming 機能選択 (A017) で選択 ・多機能入力 PRG 端子での起動/停止 (A017=01) 【MX2】 多機能入力 S1 ~ S7 選択 (C001 ~ C007) に割付。 【RX】 多機能入力 S1 ~ S8 選択 (C001 ~ C008) に割付。 ・電源投入/遮断による起動/停止 (A017=02)
		多機能入力	【MX2】 X(00) ~ X(07) / 8 点 ・多機能入力 S1 ~ S7 端子 ・X(07) はパルス列入力 RP 端子 (P003=02 の時だけ有効) 【RX】 X(00) ~ X(07) / 8 点 ・多機能入力 S1 ~ S8 端子
		周波数指令入力 (多機能アナログ入力)	XA(0): 0 ~ 10V (FV 端子)
			XA(1): 4 ~ 20mA (FI 端子)
			XA(2): 【MX2】 対応機能無し 【RX】 -10 ~ +10V (FE 端子)
	外部出力	多機能出力/多機能リレー出力	【MX2】 Y(00) ~ Y(02) / 3 点 ・多機能出力 P1、P2 端子 ・多機能リレー出力端子 (MA, MB) 【RX】 Y(00) ~ Y(05) / 6 点 ・多機能出力 P1 ~ P5 端子 ・多機能リレー出力端子 (MA, MB)
		モニタ出力 (多機能デジタル出力)	YA(0): PWM 出力 6.4ms 周期 (MP 端子)
		モニタ出力 (多機能アナログ出力)	YA(1): 0 ~ 10V (AM 端子)
			YA(2): 【MX2】 対応機能無し 【RX】 4 ~ 20mA (AMI 端子)

項目	仕様
命令	プログラム制御命令 ・ ループ (for) ・ 無条件分岐 (goto) ・ 時間制御 (wait) ・ 条件分岐 (if then, ifs then, select case, until, while) ・ サブルーチン (call, sub) ・ その他 (entry, end, inc, dec)
	演算命令 ・ 四則演算 (+, -, *, /) ・ 剰余 (mod) ・ 代入 (=) ・ 絶対値 (abs) ・ 論理演算 (or, and, xor, not)
	入出力制御 ・ 多機能入出力 (ビット入力、ワード入力、ビット出力、ワード出力) ・ インバータ入力端子の読み出し ・ インバータ出力端子の読み出し・書き込み
	タイマ制御 ・ タイマによる接点制御やタイマカウンタのコントロール制御
	パラメータ制御 ・ パラメータ No. を指定した設定データの変更 ・ パラメータ設定データを EEPROM へ保存 【RX だけの機能】 現在時刻の読み出し (1 回／連続)
	インバータ制御 ・ 正転運転と逆転運転の実行と停止 ・ DriveProgramming でのトリップ発生 (E50 ～ E59 / 10 点) ・ 周波数指令、加減速時間の設定
機能変数	ユーザパラメータ変数 ・ U(00) ～ U(31) / 32 点
	内部ユーザ変数 ・ UL(00) ～ UL(07) / 8 点
	周波数指令変数 ・ SET-Freq
	加速時間変数 ・ ACCEL
	減速時間変数 ・ DECEL
	インバータモニタ変数 ・ インバータのモニタ機能 (d001 ～ d102) を変数として使用可能。 【MX2・RX 共通機能】 - FM, Iout, Dir, PID-FB, F-CNV, Tmon, Vout, Power, RUN-Time, ON-Time, POS, STATUS, DCV, ERR-CNT, ERR(1) ～ ERR(6), UMon(0) ～ UMon(2)
	多機能入力変数 ・ インバータの多機能入力 S1 選択 (C001) で選択できる機能が使用可能。 【MX2・RX 共通機能】 - FW, RV, CF1, CF2, CF3, CF4, JG, DB, SET, TCH, FRS, EXT, USP, CS, SFT, AT, RS, STA, STP, F-R, PID, PIDC, UP, DWN, UDC, OPE, SF1, SF2, SF3, SF4, SF5, SF6, SF7, OLR, TL, TRQ1, TRQ2, BOK, LAC, PCLR, ADD, F-TM, ATR, KHC, AHD, CP1, CP2, CP3, ORL, ORG, SPD 【MX2 だけの機能】 - RS485, HLD, ROK, DISP 【RX だけの機能】 - SET3, CAS, PPI, ORT, STAT, SON, FOC, FOT, ROT, PCNT, PCC
	多機能出力変数 ・ インバータの多機能出力 P1 選択 (C021) で選択できる機能が使用可能。 【MX2・RX 共通機能】 - RUN, FA1, FA2, OL, OD, AL, FA3, OTQ, UV, TRQ, RNT, ONT, THM, BRK, BER, ZS, DSE, POK, FA4, FA5, OL2, ODc, OIDc, FBV, NDc, LOG1, LOG2, LOG3, WAC, WAF, FR, OHF, LOC, IRDY, FWR, RVR, MJA, WCO, WCOI 【MX2 だけの機能】 - FREF, REF, SETM, EDM 【RX だけの機能】 - IP, O2Dc, LOG4, LOG5, LOG6, WCO2
	入力端子変数 【MX2】 X(00) ～ X(07) / 8 点 ※X(07) はパルス列入力 RP 端子 (P003=02 の時だけ有効) 【RX】 X(00) ～ X(07) / 8 点

項目		仕様
機能変数	出力端子変数	【MX2】 Y(00) ~ Y(02) / 3 点 【RX】 Y(00) ~ Y(05) / 6 点
	内部ユーザ接点	UB(0) ~ UB(7) / 8 点
	タイマ出力接点	TD(0) ~ TD(7) / 8 点
	タイマカウンタ変数	TC(0) ~ TC(7) / 8 点
	アナログ入力端子変数	【MX2】 XA(0) ~ XA(1) 【RX】 XA(0) ~ XA(2)
	アナログ出力端子変数	【MX2】 YA(0) ~ YA(1) 【RX】 YA(0) ~ YA(2)

3

DriveProgramming の使用手順

この章では、DriveProgramming の使用手順および、関係するパラメータとプログラムの構造について説明します。

3-1	使用手順	3-2
3-2	DriveProgramming に関連するパラメータ	3-5
3-3	プログラムの構造	3-12
3-3-1	タスクの機能	3-12
3-3-2	サブルーチンの機能	3-12
3-3-3	タスクの処理	3-12
3-3-4	DriveProgramming 機能の起動／停止とタスク動作	3-14
3-3-5	DriveProgramming 機能の再起動	3-16
3-3-6	トリップ発生時のタスク動作	3-17

3-1 使用手順

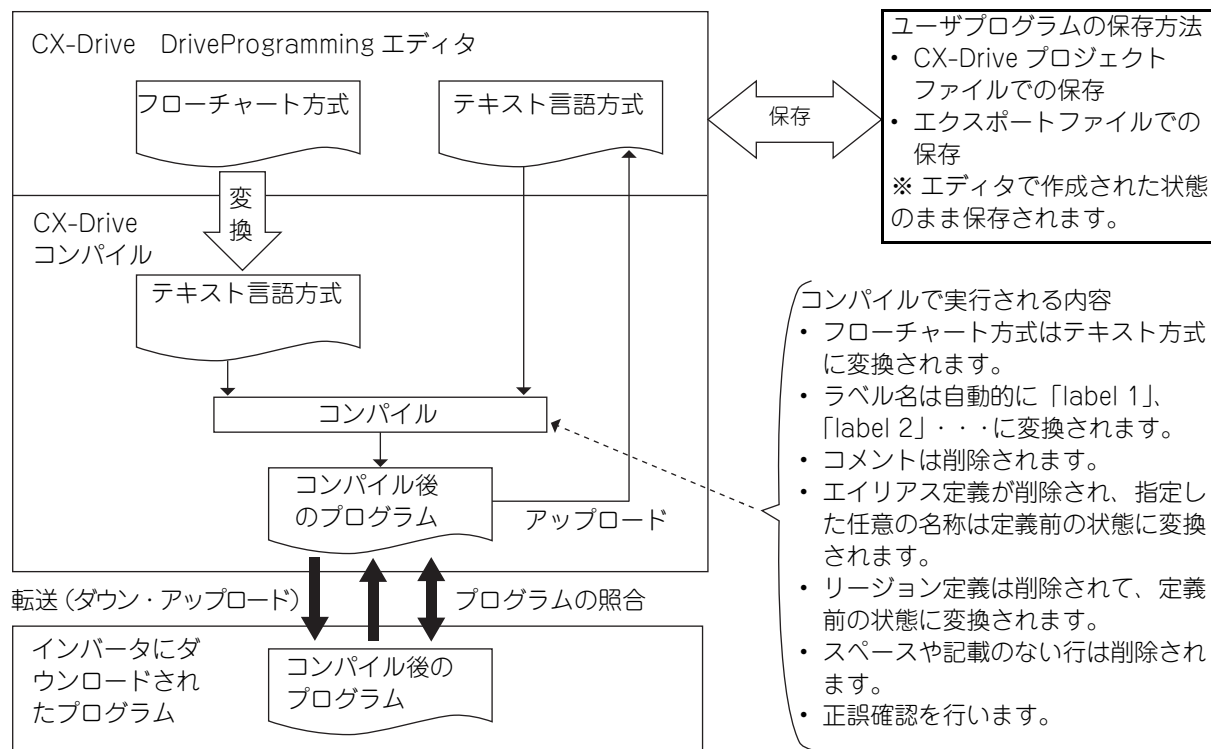
DriveProgramming のプログラム作成から実行までの流れは以下のようになります。

項 目	参照
プログラムの作成	P. 3-3
↓	
プログラムのコンパイル	P. 3-3
↓	
プログラムのダウンロード	P. 3-4
↓	
DriveProgramming 機能の選択	P. 3-4
↓	
プログラムの起動	P. 3-4

DriveProgramming のプログラムは、インバータ／サーボ用サポートツール CX-Drive の DriveProgramming エディタを使用して作成します。

以下にプログラム作成からインバータに転送（ダウンロード）するまでの流れを示します。

操作などの詳細については □「第 4 章 DriveProgramming エディタ」を参照してください。



プログラムの作成

ユーザプログラムは、フローチャート方式、およびテキスト言語方式で作成できます。タスクやサブルーチンごとにプログラミング方式を2つの方式から選択することも可能です。

ユーザプログラムの入力、CX-Drive 上の DriveProgramming エディタで行います。

DriveProgramming エディタは、DriveProgramming 領域、コマンドボックスウィンドウ、ユーザパラメータウィンドウ、プロパティウィンドウ、アウトプットウィンドウから構成されています。

- DriveProgramming 領域では、フローチャート方式または、テキスト言語方式で、プログラムを作成することができます。
- コマンドボックスウィンドウは、対応している命令のブロックがカテゴリ別にまとめて表示されます。
- ユーザパラメータウィンドウは、プログラム実行開始時のパラメータを設定することができます。
- プロパティウィンドウは、フローチャート方式で現在選択中のブロックのプロパティを編集できます。
- アウトプットウィンドウでは、コンパイル後のコンパイル異常および警告を表示します。

プログラムのコンパイル

DriveProgramming 領域で作成されたプログラムは、コンパイルを実行して最終的な「コンパイル後のプログラム」を作成し、インバータへ転送（ダウンロード）します。

コンパイル時には、プログラムの内容が正しいか、プログラムの文法、パラメータ入力制限、最大ステップ数などの確認が実施され、許可されない入力があるとコンパイルを中断し、エラーメッセージが表示されます。

さらに、コンパイル時には以下の操作も実行され、最終的な「コンパイル後のプログラム」を作成します。このため、インバータにいったん保存されたプログラムを CX-Drive に転送（アップロード）した場合、「コンパイル後のプログラム」が読み出されるので、動作は同じですが、その形式や内容は一部変更されています。

- フローチャート方式はテキスト言語方式に変換されます。
- プログラム内で任意の名称を設定したラベル名は自動的に「label 1」、「label 2」・・・に変換されます。
- プログラムに記載したコメントは削除されます。
- エイリアス定義が削除されて、指定した任意の名称は定義前の状態に変換されます。
- リージョン定義は削除されて、定義前の状態に変換されます。
- プログラム内のスペースや記載のない行は削除されます。
- 正誤確認を行います。



使用上の注意

- インバータへ転送（ダウンロード）またはプログラム変換する際のコンパイルで、設定したコメント、エイリアス定義、リージョン定義の内容は削除されます。

これらの内容を保存するには、インバータに転送（ダウンロード）する前、またはプログラム変換前のプログラムで保存し、管理してください。

CX-Drive のプロジェクトファイル全体の保存、またはプログラムのエクスポートファイルの保存でプログラムを保存できます。

- プログラムの照合は、「コンパイル後のプログラム」とインバータ内部のプログラムを比較しますので、コメント、エイリアス定義、リージョン定義などは照合されません。

プログラムのダウンロード

コンパイルしたプログラムをインバータにダウンロードし、インバータの EEPROM へ保存します。ツール (CX-Drive) がなくても、電源投入後、EEPROM に記憶したプログラムを起動することができます。

DriveProgramming 機能の選択

DriveProgramming 機能を使用するには、DriveProgramming 機能選択 (A017) を有効 (01 または 02) に設定してください。DriveProgramming 機能選択 (A017) は、運転中にも変更することができます。

パラメータ No.	機能名称	データ	説明
A017	DriveProgramming 機能選択	00: 無効 (初期データ)	DriveProgramming 機能を無効にします。プログラムを実行しません。プログラムの実行中にデータを「00: 無効」に変更すると、プログラムが停止します。
		01: 有効 (多機能入力 PRG 端子での起動/停止)	「82: PRG」機能に設定した多機能入力端子*1を入力すると、DriveProgramming 機能のプログラムを起動します。
		02: 有効 (電源投入/遮断による起動/停止)	DriveProgramming のプログラムをインバータ電源投入後、自動的に起動します。プログラムの停止中にデータを「02: 有効」に変更すると、プログラムが起動します。

*1. 多機能入力端子は、【MX2】S1～S7、【RX】S1～S8 が対象となります。

プログラムの起動

DriveProgramming 機能選択 (A017) に「01: 有効 (多機能入力 PRG 端子での起動/停止)」を設定した場合、多機能入力端子選択のいずれかに「82: PRG」を設定してください。PRG 端子が設定された多機能入力端子を「ON」するとプログラムが起動します。PRG 端子が「ON」中は実行を継続し、「OFF」で停止します。

DriveProgramming 機能選択 (A017) に「02」を設定すると、設定直後からプログラムを実行します。また、次の電源投入から自動的にプログラムを実行します。

プログラムが起動されたあと、ひと通りの処理を完了し、「end」命令に達したときは、再度プログラムを起動しなければ実行されません。

プログラムを繰り返すためには、プログラム内でループを構成して「end」命令に達しないように作成してください。

ダウンロードされた DriveProgramming 機能のプログラムはインバータ本体の EEPROM へ保存されますので、プログラムダウンロード後はサポートツールが無くてもプログラムを起動させることが可能です。

パラメータ No.	機能名称	データ	説明
C001 ～ C007	【MX2・RX 共通機能】 多機能入力 S1～S7 選択	82: PRG (DriveProgramming 起動)	DriveProgramming 機能選択 (A017) に「01: 有効 (多機能入力 PRG 端子での起動/停止)」を設定した場合、この設定を行った多機能入力でプログラムを起動します。
C008	【RX だけの機能】 多機能入力 S8 選択		

3-2 DriveProgramming に関連するパラメータ

以降に DriveProgramming に関連するインバータのパラメータについて説明します。

DriveProgramming 機能の選択

DriveProgramming 機能を使用するには、DriveProgramming 機能選択 (A017) を有効 (01 または 02) に設定してください。DriveProgramming 機能選択 (A017) は、運転中にも変更することができます。

パラメータ No.	機能名称	データ	説明
A017	DriveProgramming 機能選択	00: 無効 (初期データ)	DriveProgramming 機能を無効にします。 プログラムを実行しません。 プログラムの実行中にデータを「00: 無効」に変更すると、プログラムが停止します。
		01: 有効 (多機能入力 PRG 端子での起動/停止)	「82: PRG」機能に設定した多機能入力端子 ^{*1} を入力すると、DriveProgramming のプログラムを起動します。
		02: 有効 (電源投入/遮断による起動/停止)	DriveProgramming のプログラムをインバータ電源投入後、自動的に起動します。 プログラムの停止中にデータを「02: 有効」に変更すると、プログラムが起動します。
C001 ～ C007	【MX2・RX 共通機能】 多機能入力 S1 ～ S7 選択	82: PRG (DriveProgramming 起動)	DriveProgramming 機能選択 (A017) に「01: 有効 (多機能入力 PRG 端子での起動/停止)」を設定した場合、この設定を行った多機能入力でのプログラムを起動します。
C008	【RX だけの機能】 多機能入力 S8 選択		

*1. 多機能入力端子は、【MX2】S1 ～ S7、【RX】S1 ～ S8 が対象となります。

プログラムの初期化

インバータ本体にダウンロードした DriveProgramming 機能のプログラムを初期化する場合は、初期化範囲選択 (b084) で「04: 異常モニタクリア+データ初期化+ DriveProgramming クリア」を選択し、初期化実行 (b180) で初期化を実行してください。

パラメータ No.	機能名称	データ	説明
b084	初期化範囲選択	04: 異常モニタクリア+データ初期化+ DriveProgramming クリア	DriveProgramming のプログラムを含めて初期化するために、「04」の範囲を選択します。
b180	初期化実行	01: 初期化を実行	選択した範囲の初期化を実行します。

運転指令選択の設定

DriveProgramming 機能のプログラムから運転指令を制御するためには、FW（正転）変数と RV（逆転）変数を使用します。FW 変数または RV 変数を使用する場合は、必ず運転指令選択に「01: 制御回路端子台」を設定してください。

多機能入力端子の【MX2 だけの機能】「00: FW（正転）」または「01: RV（逆転）」の設定とは関係なく、プログラムから制御できます。

プログラムの運転指令と、【MX2】「00: FW（正転）」を設定した多機能入力端子／【RX】正転運転指令 FW 端子および「01: RV（逆転）」を設定した多機能入力端子は OR の関係となります。

【MX2】

パラメータ No.	機能名称	データ	説明
A002/ A202	第 1 / 第 2 運転指令選択	01: 制御回路端子台 (DriveProgramming) 02: オペレータ 03: Modbus 通信 04: オプション 1	プログラムから FW（正転）変数または RV（逆転）変数で運転指令を制御する場合は、「01: 制御回路端子台」を設定してください。 それ以外の設定では、FW 変数と RV 変数は無効です。
C001 ～ C007	多機能入力 S1 ～ S7 選択	00: FW（正転） 01: RV（逆転）	プログラムの運転指令と制御回路端子台の運転指令は OR の関係となります。

【RX】

- ・正転指令：正転運転指令 FW 端子（プログラムの運転指令と制御回路端子台の運転指令は OR の関係）

パラメータ No.	機能名称	データ	説明
A002	運転指令選択	01: 制御回路端子台 (DriveProgramming) 02: オペレータ 03: Modbus 通信 04: オプション 1 05: オプション 2	プログラムから FW（正転）変数または RV（逆転）変数で運転指令を制御する場合は、「01: 制御回路端子台」を設定してください。 それ以外の設定では、FW 変数と RV 変数は無効です。
C001 ～ C008	多機能入力 S1 ～ S8 選択	01: RV（逆転）	プログラムの運転指令と制御回路端子台の運転指令は OR の関係となります。

インバータ入出力機能の設定

DriveProgramming では、インバータの入出力機能（多機能入出力、アナログ入出力）をプログラムの入出力機能として使用することが可能です。

入出力機能を使用するためには、各入出力機能の設定を用途に合わせて設定していく必要があります。

ここでは、DriveProgramming 機能の入出力機能に設定する方法を説明します。

下記の設定を行うことで、DriveProgramming 機能の機能変数から自由に入出力機能を制御できます。

DriveProgramming 以外の入出力機能として設定する場合は、□『多機能型小型インバータ形 3G3MX2 シリーズ V1 タイプユーザズマニュアル (SBCE-376)』、□『多機能型小型インバータ形 3G3MX2 シリーズ V2 タイプユーザズマニュアル (SBCE-506)』、または □『高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプユーザズマニュアル (SBCE-367)』を参照してください。

【MX2】

パラメータ No.	機能名称	データ	説明
A001/ A201	第 1 / 第 2 周波数 指令選択	07: DriveProgramming	- インバータの周波数指令を DriveProgramming 機能の機能変数から指令する設定です。 - 設定を「01: 制御端子台」以外を設定すれば、アナログ入力 (FV, FI) は周波数指令と無関係となり DriveProgramming 用のアナログ入力 XA(00) ~ XA(01) として使用できます。
C001 ~ C007	多機能入力 S1 ~ S7 選択	56 ~ 62: MI1 ~ 7 (汎用入力 1 ~ 7)	「56 ~ 62: MI1 ~ MI7」を設定すると、DriveProgramming 機能の汎用入力となります。 - 各設定は機能変数 X(00) ~ X(06) に対応します。
P003	パルス列入力端子 RP 選択	02: MI8 (DriveProgramming 用汎 用入力 8)	「02: MI8 (DriveProgramming 用汎用入力 8)」を設定すると、RP 端子は DriveProgramming 機能の汎用入力となります。 - RP 端子は機能変数 X(07) に対応します。
C021, C022	多機能出力 P1、P2 選択	44 ~ 46: MO1 ~ 3 (汎用出力 1 ~ 3)	「44 ~ 46: MO1 ~ MO3」を設定すると、DriveProgramming 機能の汎用出力となります。 - 対応した機能変数 Y(00) ~ Y(02) を ON/OFF することで多機能出力端子を制御できます。
C026	多機能リレー出力 (MA, MB) 選択		
C027	MP 選択	12: DriveProgramming (YA(0))	「12: DriveProgramming」を設定すると、DriveProgramming 機能の汎用パルス出力 YA(0) として機能します。
C028	AM 選択	13: DriveProgramming (YA(1))	「13: DriveProgramming」を設定すると、DriveProgramming 機能の汎用アナログ出力 (電圧) YA(1) として機能します。
P031	加減速時間入力種別	03: DriveProgramming	加減速時間を DriveProgramming 機能から制御するときに設定します。



使用上の注意

- 多機能入力 S1 ~ S7 選択で MI1 ~ 7 (汎用入力 1 ~ 7) を選択した場合でも、多機能入力 S1 ~ S7 動作選択 (C011 ~ C017) で、NO (a 接点)、NC (b 接点) の選択が可能です。
- 多機能出力 P1、P2 選択、多機能リレー出力 (MA, MB) 選択で MO1 ~ 3 (汎用出力 1 ~ 3) を選択した場合でも、多機能出力 P1、P2 接点選択 (C031, C032)、多機能リレー出力 (MA, MB) 接点選択 (C036) で、NO (a 接点)、NC (b 接点) の選択が可能です。
- 3G3MX2 シリーズ V1 タイプのセーフティ機能切替スイッチを「ON」にし、セーフティ機能を有効に設定した場合、強制的に多機能入力 S3 選択 (C003) は「77: GS1 (セーフティ入力 1)」、多機能入力 S4 選択 (C004) は「78: GS2 (セーフティ入力 2)」に割り付けられ、多機能入力 S3 / S4 動作選択 (C013/C014) は NC (b 接点) に設定され変更できません。
多機能入力 S3、S4 に汎用入力を割り付ける場合は、セーフティ機能切替スイッチを OFF にしてください。
- DriveProgramming 機能では、アナログ入出力機能を XA(0) ~ XA(1) と YA(0) ~ YA(1) に割り付けてあります。A001、A201、C027、C028 の設定にかかわらず、プログラム上でアナログ入出力状態を各機能変数からモニタできます。
- DriveProgramming のプログラム上では、C001 ~ C007、C021、C022、C026 で汎用入出力を設定していない多機能入出力の状態をモニタできません。

【RX】

パラ メータ No.	機能名称	データ	説明
A001	周波数指令選択	07: DriveProgramming	- インバータの周波数指令を DriveProgramming 機能の機能変数から指令する設定です。 - 設定を「01: 制御端子台」以外を設定すれば、アナログ入力 (FV, FI, FE) は周波数指令と無関係となり DriveProgramming 用のアナログ入力 XA(00) ~ XA(02) として使用できます。
C001 ～ C008	多機能入力 S1 ~ S8 選択	56 ~ 63: MI1 ~ 8 (汎用入力 1 ~ 8)	「56 ~ 63: MI1 ~ MI8」を設定すると、DriveProgramming 機能の汎用入力となります。 - 各設定は機能変数 X(00) ~ X(07) に対応します。
C021 ～ C025	多機能出力 P1 ~ P5 選択	44 ~ 49: MO1 ~ 6 (汎用出力 1 ~ 6)	「44 ~ 49: MO1 ~ MO6」を設定すると、DriveProgramming 機能の汎用出力となります。 - 対応した機能変数 Y(00) ~ Y(05) を ON/OFF することで多機能出力端子を制御できます。
C026	多機能リレー出力 (MA, MB) 選択		
C027	MP 選択	12: DriveProgramming (YA(0))	「12: DriveProgramming」を設定すると、DriveProgramming 機能の汎用パルス出力 YA(0) となります。
C028	AM 選択	13: DriveProgramming (YA(1))	「13: DriveProgramming」を設定すると、DriveProgramming 機能の汎用アナログ出力 (電圧) YA(1) となります。
C029	AMI 選択	14: DriveProgramming (YA(2))	「14: DriveProgramming」を設定すると、DriveProgramming 機能の汎用アナログ出力 (電流) YA(2) となります。
P031	加減速時間入力種別	03: DriveProgramming	加減速時間を DriveProgramming 機能から制御するときに設定します。



使用上の注意

- 多機能入力 S1 ~ S8 選択で MI1 ~ 8 (汎用入力 1 ~ 8) を選択した場合でも、多機能入力 S1 ~ S8 動作選択 (C011 ~ 018) で、NO (a 接点)、NC (b 接点) の選択が可能です。
- 多機能出力 P1 ~ P5 選択、多機能リレー出力 (MA, MB) 選択で MO1 ~ 6 (汎用出力 1 ~ 6) を選択した場合でも、多機能出力 P1 ~ P5 接点選択 (C031 ~ C035)、多機能リレー出力 (MA, MB) 接点選択 (C036) で、NO (a 接点)、NC (b 接点) の選択が可能です。
- 3G3RX シリーズ V1 タイプの緊急遮断機能を有効 (SW1=「ON」) に設定した場合、強制的に多機能入力 S1 選択 (C001) は「18: RS」、多機能入力 S3 選択 (C003) は「64: EMR」に割り付けられ、多機能入力 S3 / S4 動作選択 (C011/C013) は NO (a 接点) と NC (b 接点) に設定され変更できません。
多機能入力 S1、S3 に汎用入力を割り付ける場合は、緊急遮断機能を無効にしてください。
- DriveProgramming では、アナログ入出力機能を XA(0) ~ XA(2) と YA(0) ~ YA(2) に割り付けてあります。A001、C027 ~ C029 の設定にかかわらず、プログラム上でアナログ入出力状態を各機能変数からモニタできます。
- DriveProgramming のプログラム上では、C001 ~ C008、C021 ~ C026 で汎用入出力を設定していない多機能入出力の状態をモニタできません。

DriveProgramming のモニタ機能

DriveProgramming の状態をモニタする機能として、以下の機能を用意しています。



使用上の注意

- 4 桁を超えるデータは、デジタルオペレータで以下のように上位 4 桁を表示します。
末尾に示されているドットは、小数点の位置を示していますので、ドットの位置で判断してください。
【0 ～ 9,999 のデータ表示】 0. ～ 9999.
【10,000 ～ 65,535 のデータ表示】 1000 ～ 6553
【-1,230,000、1,230,000 のデータ表示】 -123、1230
- デジタルオペレータは上位 4 桁しか表示できないため、末尾のデータまで確認・設定できません。
4 桁を超えるデータを確認・設定するためには CX-Drive をご使用ください。

パラメータ No.	機能名称	データ	説明
d023	プログラムカウンタ (DriveProgramming)	0 ～ 1,024	DriveProgramming の実行中、行番号を表示します。 行番号は、代表してタスク 1 (DriveProgramming エディタ上の一番左側タブのタスク) の行番号で示されます。 ・「コンパイル後のプログラム」の行番号で表示します。 ・サブルーチン実行中の行も含めて表示します。 ※ プログラム停止中は「0000」を示します。
d024	プログラム番号モニタ (DriveProgramming)	1	転送されたプログラム番号を表示します。
d025	ユーザモニタ 0 (DriveProgramming)	-2,147,483,647 ～ +2,147,483,647	機能変数の UMon(0) に対応したモニタ機能です。 プログラム中の任意のデータを表示することが可能です。
d026	ユーザモニタ 1 (DriveProgramming)	-2,147,483,647 ～ +2,147,483,647	機能変数の UMon(1) に対応したモニタ機能です。 プログラム中の任意のデータを表示することが可能です。
d027	ユーザモニタ 2 (DriveProgramming)	-2,147,483,647 ～ +2,147,483,647	機能変数の UMon(2) に対応したモニタ機能です。 プログラム中の任意のデータを表示することが可能です。

DriveProgramming ユーザパラメータ

DriveProgramming のユーザパラメータとして最大 32 個のパラメータを用意しています。プログラムの初期データ、調整パラメータ、演算結果の格納などに使用してください。



使用上の注意

- 4 桁を超えるデータは、デジタルオペレータで以下のように上位 4 桁を表示します。末尾に示されているドットは、小数点の位置を示していますので、ドットの位置で判断してください。
【0 ～ 9,999 のデータ表示】 0. ～ 9999.
【10,000 ～ 65,535 のデータ表示】 1000 ～ 6553
【-1,230,000、1,230,000 のデータ表示】 -123、1230
- デジタルオペレータは上位 4 桁しか表示できないため、末尾のデータまで確認・設定できません。
 4 桁を超えるデータを確認・設定するためには CX-Drive をご使用ください。

パラメータ No.	機能名称	データ	説明
P100	DriveProgramming ユーザパラメータ U00	0 ～ 65,535	- 機能変数 U(00) ～ U(28) に対応したユーザパラメータです。 - デジタルオペレータから変更が可能です。変更した場合は EEPROM に記憶されます。 - DriveProgramming で EEPROM 書込命令を実行した際に、EEPROM に設定データを保存します。
P101	DriveProgramming ユーザパラメータ U01		
P102	DriveProgramming ユーザパラメータ U02		
P103	DriveProgramming ユーザパラメータ U03		
P104	DriveProgramming ユーザパラメータ U04		
P105	DriveProgramming ユーザパラメータ U05		
P106	DriveProgramming ユーザパラメータ U06		
P107	DriveProgramming ユーザパラメータ U07		
P108	DriveProgramming ユーザパラメータ U08		
P109	DriveProgramming ユーザパラメータ U09		
P110	DriveProgramming ユーザパラメータ U10		
P111	DriveProgramming ユーザパラメータ U11		
P112	DriveProgramming ユーザパラメータ U12		
P113	DriveProgramming ユーザパラメータ U13		
P114	DriveProgramming ユーザパラメータ U14		
P115	DriveProgramming ユーザパラメータ U15		
P116	DriveProgramming ユーザパラメータ U16		

パラメータ No.	機能名称	データ	説明
P117	DriveProgramming ユーザパラメータ U17	0 ~ 65,535	- 機能変数 U(00) ~ U(28) に対応したユーザパラメータです。 - デジタルオペレータから変更が可能です。変更した場合は EEPROM に記憶されます。 - DriveProgramming で EEPROM 書込命令を実行した際に、EEPROM に設定データを保存します。
P118	DriveProgramming ユーザパラメータ U18		
P119	DriveProgramming ユーザパラメータ U19		
P120	DriveProgramming ユーザパラメータ U20		
P121	DriveProgramming ユーザパラメータ U21		
P122	DriveProgramming ユーザパラメータ U22		
P123	DriveProgramming ユーザパラメータ U23		
P124	DriveProgramming ユーザパラメータ U24		
P125	DriveProgramming ユーザパラメータ U25		
P126	DriveProgramming ユーザパラメータ U26		
P127	DriveProgramming ユーザパラメータ U27		
P128	DriveProgramming ユーザパラメータ U28		
P129	DriveProgramming ユーザパラメータ U29	0 ~ 65,535	- 機能変数 U(29) ~ U(31) に対応したユーザパラメータです。 - デジタルオペレータから変更が可能です。変更した場合は EEPROM に記憶されます。 - DriveProgramming で EEPROM 書込命令を実行した際に、EEPROM に設定データを保存します。 - インバータの電源遮断時に、遮断時の設定データを EEPROM に保存します。
P130	DriveProgramming ユーザパラメータ U30		
P131	DriveProgramming ユーザパラメータ U31		

3-3 プログラムの構造

形 3G3MX2 シリーズ V1 タイプ／3G3MX2 シリーズ V2 タイプ／3G3RX シリーズ V1 タイプの DriveProgramming では、最大 5 つのタスクを作成することが可能です。

作成された各タスクは並列処理されます。

1 つのアプリケーションを分割して、複数のタスクに処理を割り付けることで、それぞれの実行条件や実行順序などを調整することが可能です。さらに、各タスクが並列処理されるので、処理時間も短縮できます。

3-3-1 タスクの機能

タスクとは、DriveProgramming で実行されるプログラム単位です。

形 3G3MX2 シリーズ V1 タイプ／3G3MX2 シリーズ V2 タイプ／3G3RX シリーズ V1 タイプでは、最大 5 つのタスクが作成できますので、5 つのタスクを合わせて全体のプログラムとなります。

- 各タスクは同時に起動されます。
- 2ms の処理時間で、タスク 1 からタスク 5 の順に 1 命令（「コンパイル後のプログラム」の 1 行）が実行されます。
- 各タスクでユーザパラメータなどの機能変数を全て共用して使用します。タスク間の情報伝達はユーザパラメータや内部ユーザ接点などを使用してください。
- 2ms の処理時間の最後に、各タスクの演算結果をインバータ動作や外部出力に反映し、インバータ状態や外部入力の状態を取り込みます。
- 「end」命令を実行すると、そのタスクは終了し、次の起動待ち状態になります。

3-3-2 サブルーチンの機能

サブルーチンとは、呼び出された場合だけ実行される、分離されたプログラム処理です。

サブルーチンを使用すると、1 つのタスクで複数回実行される処理を 1 つのプログラムにまとめたり、他のプログラムで再利用できるプログラムにまとめることができます。

DriveProgramming 機能では、サブルーチンをタスクごとにプログラムしてください。タスク間で共用することはできません。

サブルーチン内で、次のサブルーチンを呼び出すことも可能です（ネスティング）。

ただし、サブルーチンのネスティングは最大 8 階層です。

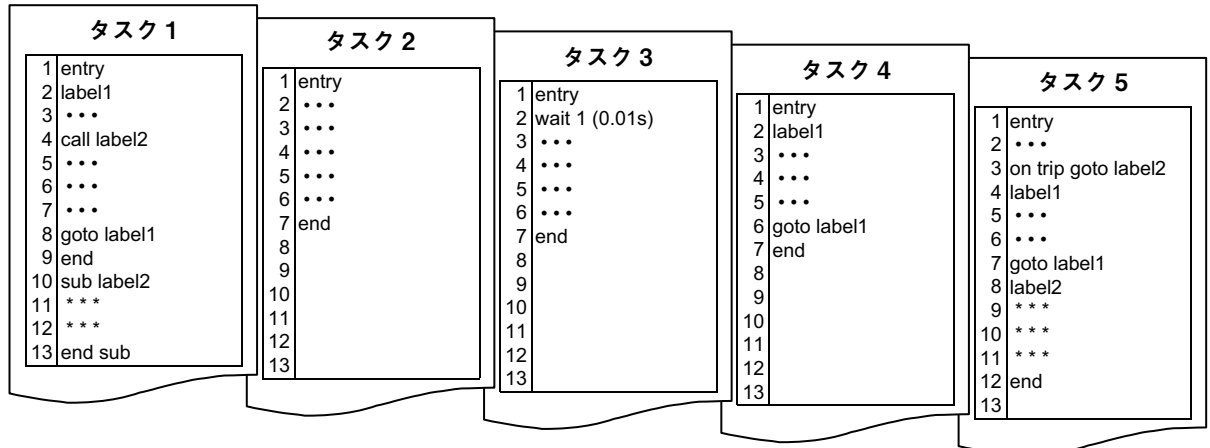
3-3-3 タスクの処理

最大 5 つのタスクは、同時に起動され、2ms の処理時間に各タスクの 1 命令（「コンパイル後のプログラム」の 1 行）が実行されます。そのプログラム処理の流れを下図の「コンパイル後のプログラム」で説明します。

- 各タスクは同時に起動されます。
- 2ms の処理時間で、タスク 1 からタスク 5 の順に 1 命令（「コンパイル後のプログラム」の 1 行）が実行されます。
- 各タスクでユーザパラメータなどの機能変数を全て共用して使用します。
- 2ms の処理時間の最後に、各タスクの演算結果をインバータの動作や外部出力に反映し、インバータの状態や外部入力の状態を取り込みます。
- タスク 1 の様に、「call」命令でサブルーチンを呼び出すと、次の処理からサブルーチンの行を実行します。
- タスク 2 の様に、「end」命令を実行すると、そのタスクは終了し、次の起動待ち状態になります。
- タスク 3 の様に、「wait」命令を実行すると、条件が成立するまで実行を繰り返します。
- タスク 4 の様に、「goto」命令でループを形成すると、常にそのタスク処理を繰り返します。

- ・ タスク 5 の様に、「on trip goto 」命令を 1 度実行すると、その内容はインバータ内部に記憶されます。
トリップが発生した時点で、すぐに「goto」先に分岐します。

● コンパイル後のプログラム (例)



● プログラム処理の流れ

処理 時間 [ms]	タスク 1	タスク 2	タスク 3	タスク 4	タスク 5
行 No.	行 No.	行 No.	行 No.	行 No.	行 No.
2	1 entry	1 entry	1 entry	1 entry	1 entry
4	2 label1	2 ...	2 wait 1 (0.01s)	2 label1	2 ...
6	3 ...	3 ...	2 wait 1 (0.01s)	3 ...	3 on trip goto label2
8	4 call label2	4 ...	2 wait 1 (0.01s)	4 ...	4 label1 【トリップ発生】
10	10 sub label2	5 ...	2 wait 1 (0.01s)	5 ...	5 label2
12	11 ***	6 ...	2 wait 1 (0.01s)	6 goto label1	6 ***
14	12 ***	7 end	3 ...	2 label1	7 ***
16	13 end sub		4 ...	3 ...	2 ***
18	5 ...		5 ...	4 ...	3 end
20	6 ...		6 ...	5 ...	8
22	7 ...		7 end	6 goto label1	9
24	8 goto label1			2 label1	10
26	2 label1			3 ...	11
28	3 ...			4 ...	12
30	4 call label2			5 ...	
32	10 sub label2			6 goto label1	
34	11 ***			2 label1	
36	12 ***			3 ...	
38	13 end sub			4 ...	

インバータトリップが発生しているため、Label2 を実行します。

プログラムカウンタ (d023) の表示

3-3-4 DriveProgramming 機能の起動／停止とタスク動作

DriveProgramming 機能のプログラムの起動／停止方法は、DriveProgramming 機能選択 (A017) で設定します。

● 起動

- DriveProgramming 機能選択 (A017) が「01: 有効 (多機能入力 PRG 端子での起動／停止)」の場合：
多機能入力に設定した PRG 端子 ON
- DriveProgramming 機能選択 (A017) が「02: 有効 (電源投入／遮断による起動／停止)」の場合：
インバータの電源投入

● 停止

- DriveProgramming 機能選択 (A017) が「01: 有効 (多機能入力 PRG 端子での起動／停止)」の場合：
多機能入力に設定した PRG 端子 OFF
- DriveProgramming 機能選択 (A017) が「02: 有効 (電源投入／遮断による起動／停止)」の場合：
インバータの電源遮断

また、A017 の設定に関係なくプログラム異常の E43 ～ E45 トリップを検出した場合と、全てのタスクが「end」命令を実行した場合にも DriveProgramming 機能のプログラムは停止します。



安全上の要点

DriveProgramming 機能のプログラムが停止すると、DriveProgramming で制御している多機能出力はプログラム停止直前の状態を保持します。

このため、インバータの DriveProgramming 機能のプログラム停止を DriveProgramming 起動信号とアラーム (トリップ) 信号で検出し、インバータ周辺機器を安全に停止させる配線を行ってください。



使用上の注意

DriveProgramming 機能のプログラムが停止した場合、各機能変数の状態は下記になりますので、各状態を認識して、次の対応を実施してください。

- 多機能出力 (実際のインバータ出力端子) の状態はプログラム停止前の状態を保持します。
ただし、DriveProgramming 機能の汎用出力 MO1 ～ MO6 を設定していない出力は、通常のインバータ多機能出力として動作します。
- 周波数指令、加減速時間、アナログ出力に DriveProgramming 機能を設定した場合、各設定値はプログラム停止前の状態を保持します。
- ユーザパラメータ変数、内部ユーザ変数、内部ユーザ接点はプログラム停止前の状態を保持します。
- 多機能入力やアナログ入力など実際のインバータ入力端子状態は、入力状態を保持せず常に状態を更新します。
- 多機能出力変数 (RUN、FA1、AL、・・・などの機能ビット) やインバータモニタ変数は、状態を保持せずインバータの状態に合わせて常に状態を更新します。
- DriveProgramming の多機能入力変数 (FW、RV、CF1、・・・などの機能ビット) とタイマ変数だけ、プログラム停止と同時にクリアされ、全ての状態がゼロとなります。

CX-Drive で DriveProgramming 機能を起動/停止するとき、CX-Drive はインバータのパラメータ DriveProgramming 機能選択 (A017) の値を、起動時は「02」に、停止時は「00」と一時的に (RAM 上のデータだけ) 変更しています。

CX-Drive で DriveProgramming 機能の起動/停止を行ったあとは、次の対応をしてください。

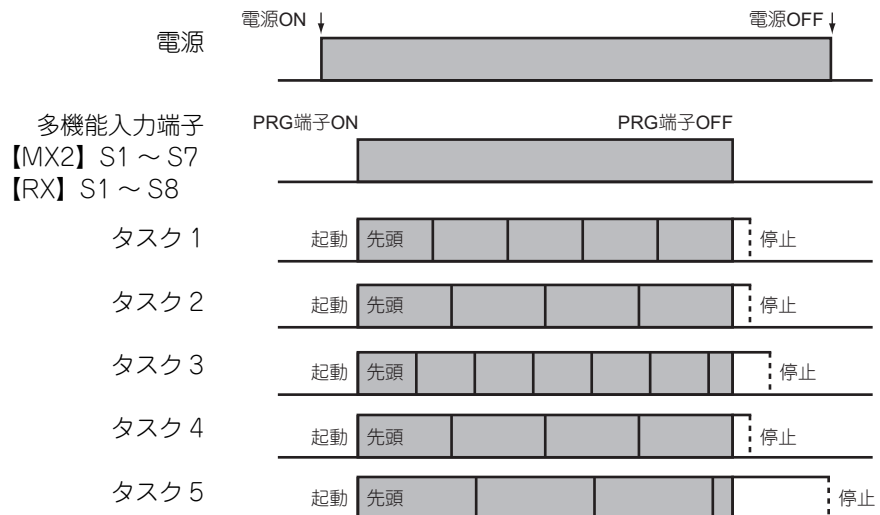
- ・インバータの電源を再投入して、A017 を EEPROM に保存されている値に戻してください。
- ・インバータの電源を再投入するまで、以下の EEPROM 保存操作を実行しないでください。
 - ・ CX-Drive からパラメータの部分転送 (ダウンロード) を行う。
 - ・ Modbus 通信または通信オプション経由で「enter」命令を発行する。

DriveProgramming 機能のプログラムを起動させると、5 つのタスクは同時に起動します。タスク内で「goto」命令などを使用してループ構成にした場合、そのタスクは起動後、常にそのループを繰り返します。

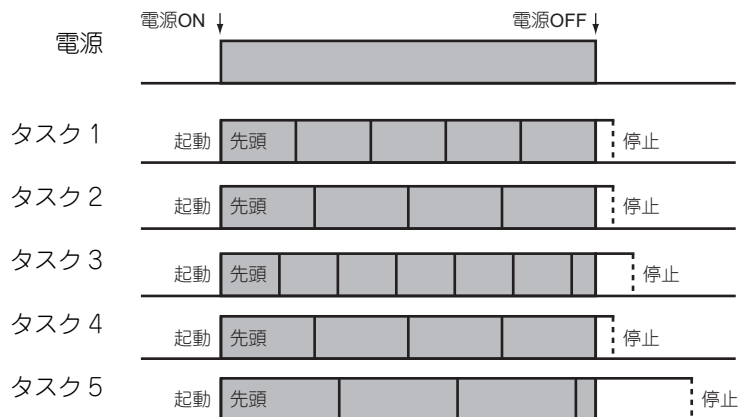
DriveProgramming 機能のプログラムを停止させると、5 つのタスクは同時に停止します。モータ運転時にプログラムを停止させた場合、停止方法選択 (b091) の設定に従ってモータを停止 (減速停止またはフリーランストップ) させます。

各設定のタスク動作を以下に示します。

● DriveProgramming 機能選択 (A017) に「01: 有効」を設定した場合



● DriveProgramming 機能選択 (A017) に「02: 有効」を設定した場合



3-3-5 DriveProgramming 機能の再起動

一度停止させたプログラムは以下の操作で再起動できます。再起動すると各タスクはプログラムの先頭から同時に起動します。

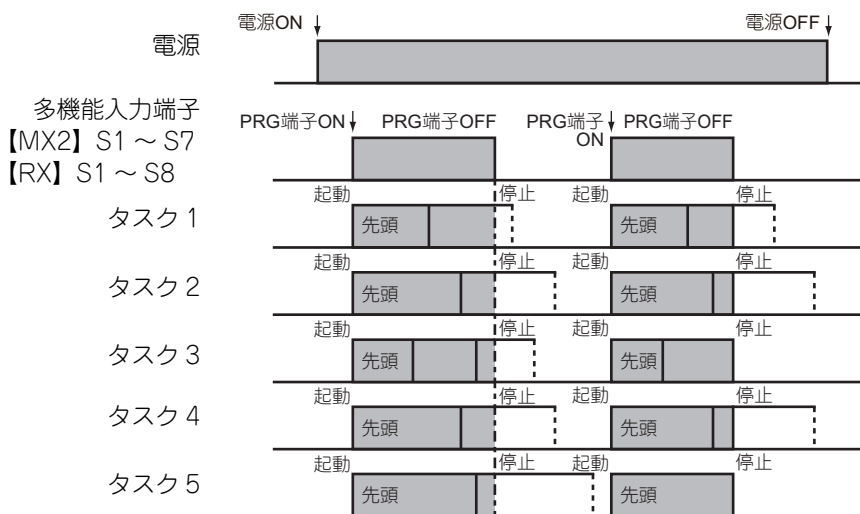
- DriveProgramming 機能選択 (A017) が「01: 有効 (多機能入力 PRG 端子での起動/停止)」の場合：PRG 端子が ON 状態で制御回路端子からのリセット入力、または、PRG 端子の再入力。
- DriveProgramming 機能選択 (A017) が「02: 有効 (電源投入/遮断による起動/停止)」の場合：制御回路端子からのリセット入力、または、インバータ電源再投入。



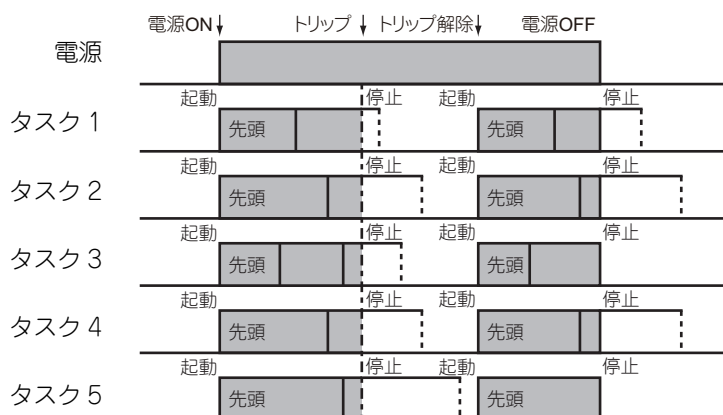
使用上の注意

- デジタルオペレータや LCD デジタルオペレータ (オプション) の STOP/RESET キーでは DriveProgramming 機能を再起動できません。多機能入力端子 (【MX2】 S1 ~ S7 / 【RX】 S1 ~ S8) に「18: RS (リセット)」を設定して、対応する端子を入力してください。
- DriveProgramming 機能をリセット入力で再起動させるためには、リセット選択 (C102) の設定で、03 (トリップだけ解除) 以外を設定してください。03 (トリップだけ解除) の設定では再起動しません。
- リセット選択 (C102) に、「00: ON 時トリップ解除」または「01: OFF 時トリップ解除」を設定した場合、トリップしていない状態でリセット入力を入力しても DriveProgramming 機能は再起動します。

● DriveProgramming 機能選択 (A017) に「01: 有効」を設定した場合の再起動



● DriveProgramming 機能選択 (A017) に「02: 有効」を設定した場合のトリップ再起動 (「on trip goto」命令なし)



3-3-6 トリップ発生時のタスク動作

DriveProgramming が動作している状態で、インバータがトリップを検出しても基本的に DriveProgramming は動作を継続します。ただし、DriveProgramming 関連のトリップ E43 ～ E45 を検出した場合は動作を停止します。

また、「on trip goto」命令を使用すると、トリップ発生後に別の処理に分岐することが可能です。

on trip goto の有無	エラー内容		
	ユーザトリップ E50 ～ E59	DriveProgramming 関連 トリップ E43 ～ E45	その他のトリップ
なし	そのまま動作継続	プログラム停止	そのまま動作継続
あり	「on trip goto」命令実行後、指定ラベルの行に移行し、動作継続	プログラム停止	「on trip goto」命令実行後、指定ラベルの行に移行し、動作継続

トリップ E43 ～ E59 の詳細内容は □「第 8 章 異常とその処理」を参照してください。

4

DriveProgramming エディタ

この章では、DriveProgramming エディタの起動、保存と読み出し、各部の詳細について説明します。

4

4-1	DriveProgramming エディタの起動	4-2
4-2	各部の説明	4-6
4-2-1	DriveProgramming エディタ	4-6
4-2-2	ツールバー	4-6
4-2-3	DriveProgramming 領域	4-9
4-2-4	コマンドボックスウィンドウ	4-13
4-2-5	ユーザパラメータウィンドウ	4-14
4-2-6	プロパティウィンドウ	4-15
4-2-7	アウトプットウィンドウのエラーリストタブ	4-16
4-3	タスクの追加・削除と名前の変更	4-17
4-4	サブルーチンの挿入・削除と呼び出し	4-18
4-5	フローチャートプログラムの作成	4-19
4-6	テキストプログラムの作成	4-20
4-7	転送（アップロード）したプログラムの編集	4-21
4-8	プログラムの保存	4-22
4-9	プログラムの転送と照合	4-24
4-10	プログラムの実行（DriveProgramming 機能選択）	4-25
4-11	その他の便利な機能	4-28

4-1 DriveProgramming エディタの起動

DriveProgramming はインバータ／サーボ用サポートツール CX-Drive の DriveProgramming エディタの機能を使用してプログラムを作成します。

CX-Drive の DriveProgramming エディタの構成と、DriveProgramming エディタを起動するまでの操作を説明します。

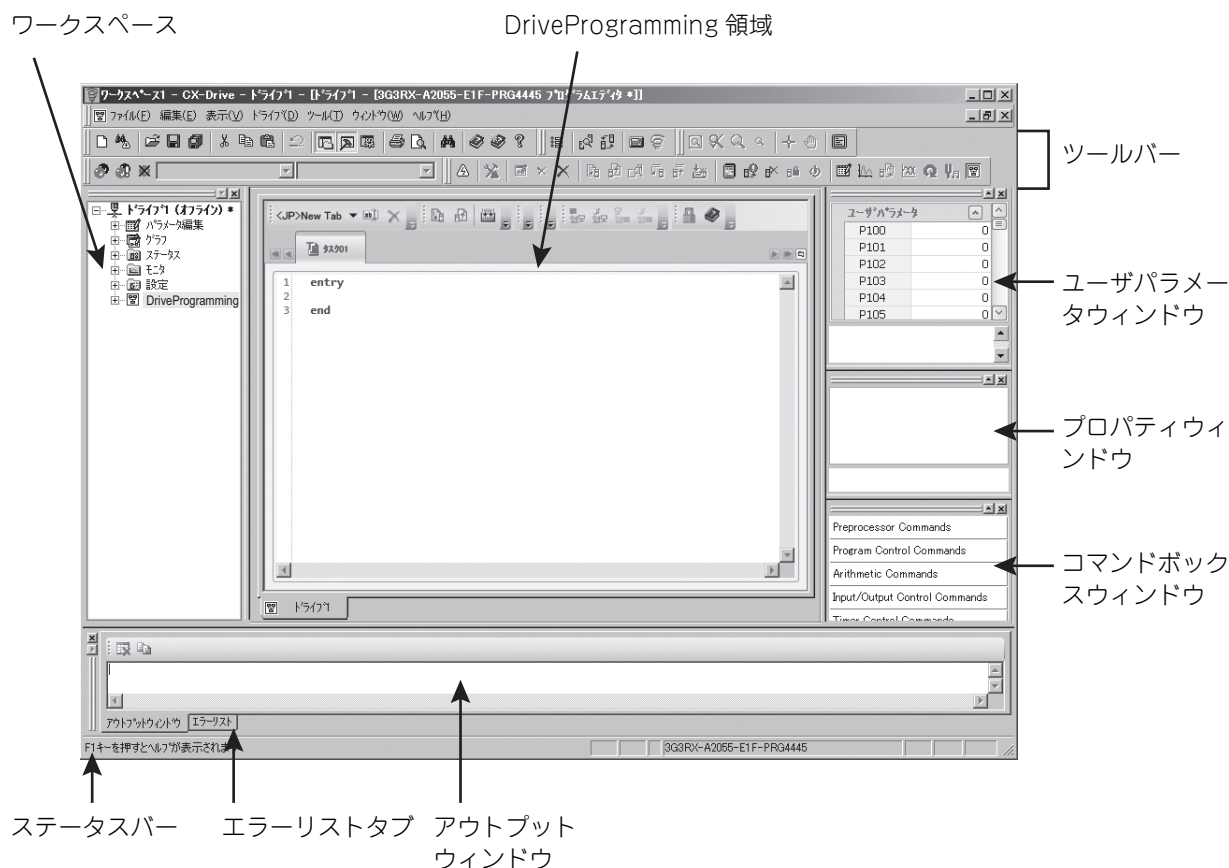


使用上の注意

- CX-Drive は、以下に示すバージョン以降から DriveProgramming エディタに対応しています。
以下のバージョン以前の場合、CX-Drive のバージョンアップが必要です。
【MX2】 V1 タイプは Ver.2.8 以降（Ver.2.7 以前では操作できません。）、V2 タイプは Ver.3.02 以降
【RX】 Ver.2.7 以降（Ver.2.6 以前では操作できません。）
- CX-Drive の DriveProgramming エディタを、最初に起動させるためには、パスワードが必要です。
当社営業担当者までご確認ください。

CX-Drive と DriveProgramming エディタ画面構成

DriveProgramming エディタはインバータ／サーボ用サポートツール CX-Drive の 1 つの機能として搭載されています。DriveProgramming エディタを起動すると CX-Drive は以下の画面構成になります。



CX-Drive の起動

CX-Drive を起動するには、Windows の [スタート] から [すべてのプログラム] を選び、[OMRON] - [CX-One] - [CX-Drive] のフォルダで CX-Drive のアイコンをクリックしてください。

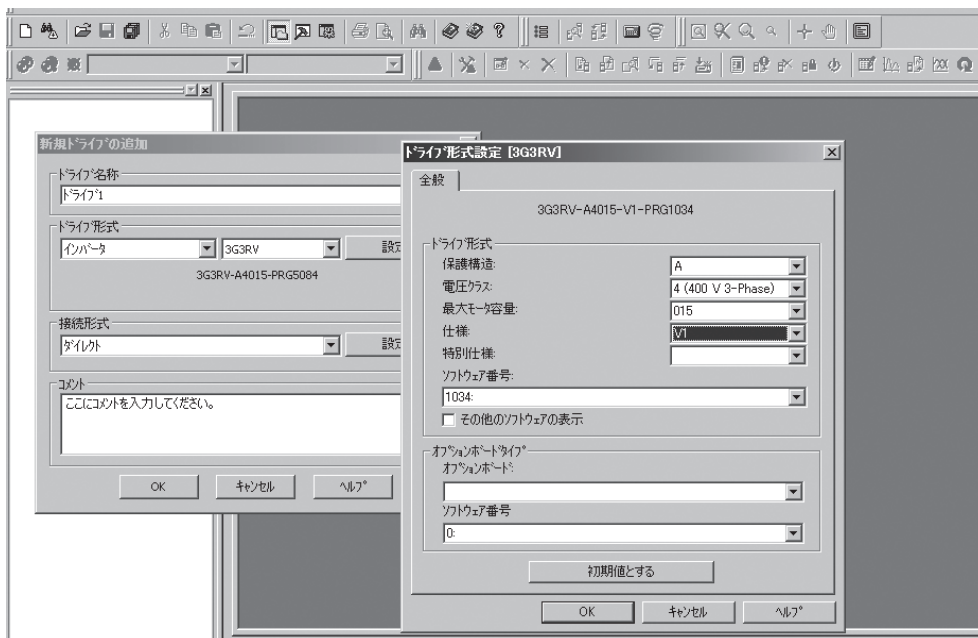
CX-Drive が起動したあと、CX-Drive のメニューから [ファイル] を選び [新規作成] をクリックして、新しい CX-Drive ファイルを作成してください。[新規ドライブの追加] ウィンドウが表示されます。

[ドライブ形式] のインバータシリーズから、3G3MX2 シリーズまたは 3G3RX シリーズを選択したあと、その右の [設定] ボタンを押してください。

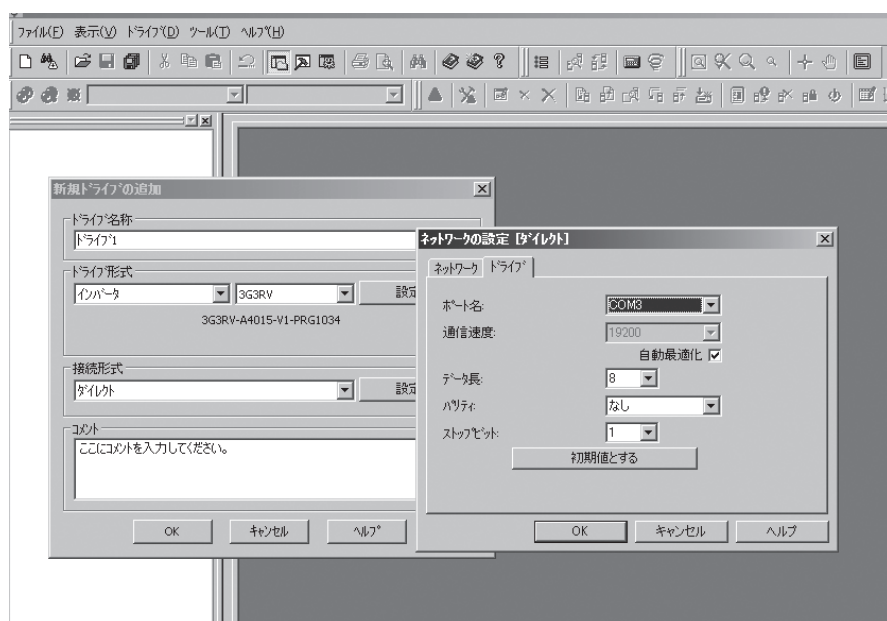
ドライブ形式設定ウィンドウで、インバータの [保護構造] / [電圧クラス] / [最大モータ容量] を設定してください。

3G3MX2 / 3G3RX シリーズで V1 タイプを選択するには、必ず [仕様] で [V1] を選択してください。

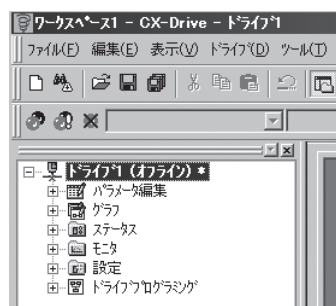
設定後に OK ボタンを押して、ドライブ形式設定ウィンドウを閉じてください。



次に新規ドライブの追加のウィンドウで CX-Drive とインバータとの接続方式を設定します。
 [接続方式] でダイレクトを選択したあと、その右の [設定] ボタンを押してください。
 [ドライブ] のタブで、CX-Drive で使用するパソコンのポート名を設定してください。



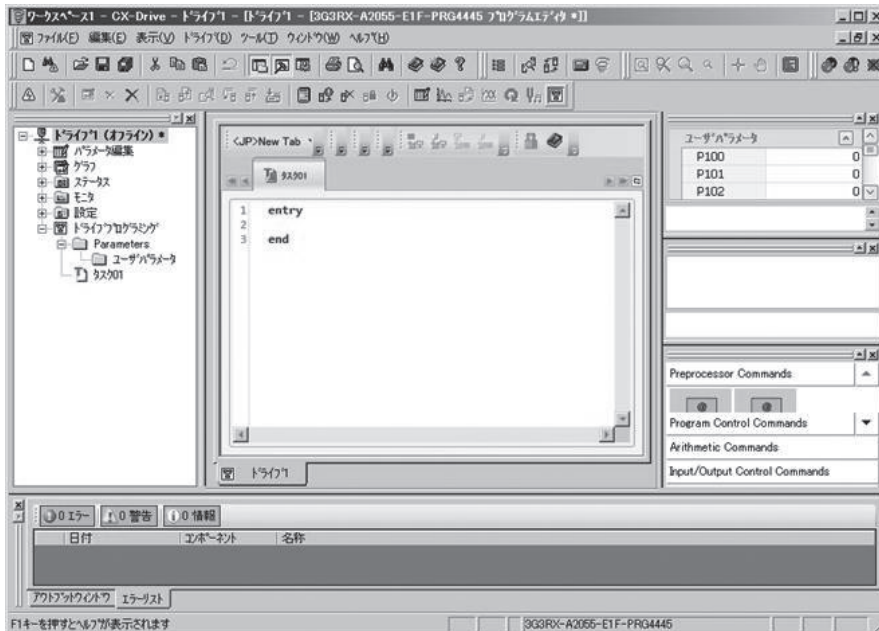
設定したあと、[OK] を押してすべてのウィンドウを閉じると、ワークスペースに新しいプロジェクトが登録されます。



DriveProgramming エディタの起動

DriveProgramming エディタを表示させるには以下の 3 つの方法があります。

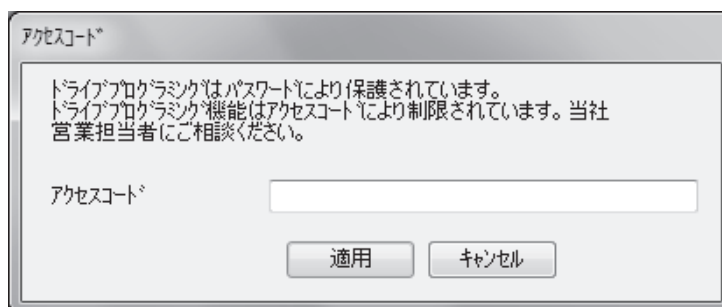
- ワークスペースの [ドライブプログラミング] をダブルクリックする。
- CX-Drive ツールバーから  ボタンをクリックする。
- メニューから [ドライブ] を選び、[プログラム] - [プログラムエディタ] を選択する。



使用上の注意

CX-Drive を立ち上げたあと、最初に DriveProgramming エディタを起動するにはパスワードが必要です。

当社営業担当者にパスワードをご確認いただき、下記の画面でパスワードを設定してください。

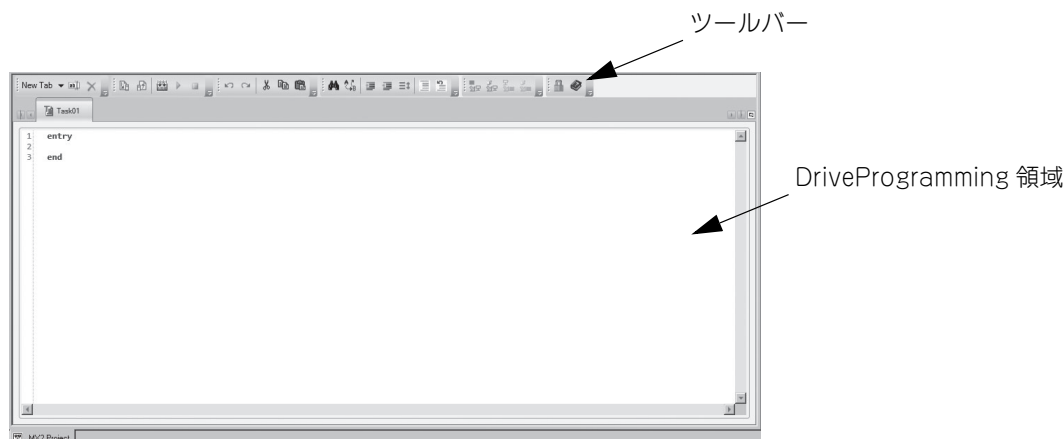


4-2 各部の説明

ここでは、CX-Drive の DriveProgramming 機能に関するエディタやウィンドウの機能について、1 つ 1 つ説明していきます。

4-2-1 DriveProgramming エディタ

DriveProgramming エディタは、DriveProgramming 機能のメインウィンドウです。



このウィンドウは、共通のコマンドを含むツールバーと、プログラムがテキストまたはフローチャートとして表示される DriveProgramming 領域で構成されます。

4-2-2 ツールバー

DriveProgramming エディタには、次のコマンドが用意されています。

● 共通のコマンド

コマンド	アイコン	説明
新規タスク（フローチャート）		プログラムに新しいフローチャートタスクを作成します。タスクとは、DriveProgramming で実行されるプログラム単位です。
新規タスク（テキスト）		プログラムに新しいテキストタスクを作成します。タスクとは、DriveProgramming で実行されるプログラム単位です。
新規サブルーチン（フローチャート）		新しいフローチャートサブルーチンを作成します。サブルーチンとは、呼出された場合だけ実行されるプログラム処理です。
新規サブルーチン（テキスト）		新しいテキストサブルーチンを作成します。サブルーチンとは、呼出された場合だけ実行されるプログラム処理です。
現在のタスク／サブルーチンの名前変更		現在のタスクやサブルーチンの名前を変更します。
現在のタブを削除		現在のタスクやサブルーチンを削除します。
元に戻す		直前に行った操作を元に戻します。
やり直し		「元に戻す」の操作をやり直します。

コマンド	アイコン	説明
ドライブに転送		プログラムをコンパイルし、異常がない場合はドライブ（インバータ）に転送（ダウンロード）します。
ドライブから転送		ドライブ（インバータ）から DriveProgramming エディタにプログラムを転送（アップロード）します。
コンパイル		DriveProgramming 領域のプログラムをコンパイルします。プログラム内で発生したコンパイル異常および警告は、アウトプットウィンドウのエラーリストタブに表示されます。
起動		インバータ内部のプログラムを起動します。CX-Drive は、インバータ内部と DriveProgramming 領域のプログラムを確認し、同じ場合だけプログラムを起動します。異なる場合はプログラムは起動しません。 また、このプログラムの起動は CX-Drive が接続された時だけ有効です。 インバータ単体で起動する場合は、必ず DrivProgramming 機能選択 (A017) を設定してください。
停止		インバータ内部のプログラムを停止します。この動作は、CX-Drive の DriveProgramming 領域のプログラムとは関係なく、実行されます。
プログラム番号の設定		プログラムに番号を設定します。(0 ~ 9999 の範囲で設定できます。) プログラム番号は、プログラムをドライブ（インバータ）に転送（ダウンロード）した後に有効になり、プログラム番号モニタ (DriveProgramming) (d024) で確認できます。
パスワード		プログラムのパスワードを設定、変更、または削除できます。
ヘルプ		CX-Drive のヘルプを表示します。

● フローチャートプログラムのコマンド

コマンド	アイコン	説明
拡大		拡大します。
縮小		縮小します。
ズームリセット		ズームを初期値に戻します。
選択		マウスカーソルでドラッグすることにより、プログラムのブロック ^{*1} を1つ以上選択できます。
フルスクロール		ドラッグすることにより、同じスケールを保ちながら、任意の方向にビューの範囲を移動できます。
左揃え		現在選択中のブロック ^{*1} を左に揃えます。
左右中央揃え		現在選択中のブロック ^{*1} を中央に揃えます。
右揃え		現在選択中のブロック ^{*1} を右に揃えます。
上揃え		現在選択中のブロック ^{*1} を上揃えます。
上下中央揃え		現在選択中のブロック ^{*1} を中央に揃えます。

コマンド	アイコン	説明
下揃え		現在選択中のブロック ^{*1} を下に揃えます。
方向	方向 ▼	ブロック ^{*1} を接続する方向を選択します。
自動配置		現在選択中の方向に、フローチャートのブロック ^{*1} を自動的に配置します。
接点を表示		ブロック ^{*1} の接点の表示または非表示を切り替えます。
表示	表示 ▼	プログラムの表示スタイルを以下から選択できます。 • テキストだけ • アイコンだけ • アイコンとテキスト • 名称、アイコン、引数

^{*1}. フローチャート上の1つの表示単位を「ブロック」と呼んでいます。ブロックは、1つの命令または、1つの命令にラベルが添付されて構成されます。

● テキストプログラムのコマンド

コマンド	アイコン	説明
検索		DriveProgramming エディタのテキストを対象に検索します。
置換		DriveProgramming エディタのテキストを対象に置き換えます。
インデント		選択中のテキストのインデントを右に移動します。
インデント解除		選択中のテキストのインデントを左に移動します。
テキストの整形		選択中のテキストを自動で整形します。
テキストのコメントアウト		選択中のテキスト行をコメントに変換します。
テキストのコメントアウト解除		選択したテキスト行のコメントを解除します。

● プログラム変換

フローチャートプログラムをテキストプログラムに変換、またはその逆の変換が可能です。ただし、変換時にいったんコンパイルしますので、コンパイル異常が発生した場合、変換できません。

また、変換時のコンパイルにより、動作は同じですが、コメント、エイリアス定義、リージョン定義などは削除され、その形式や内容は一部変更されます。

コマンド	アイコン	説明
フローチャート変換		現在のテキストのタスクやサブルーチンをフローチャートに変換します。
全体フローチャート変換		プログラム全体をフローチャートに変換します。
テキスト変換		現在のフローチャートのタスクやサブルーチンをテキストに変換します。
全体テキスト変換		プログラム全体をテキストに変換します。

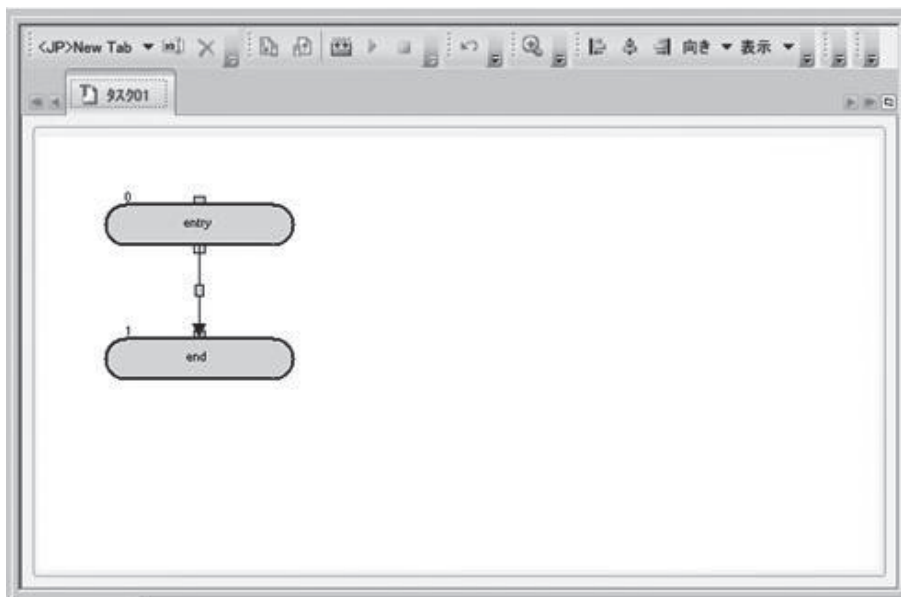
● ショートカットキー

DriveProgramming 領域では、次のキーボードショートカットを使用できます。

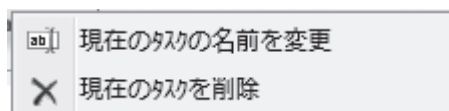
ショートカット	内容
Ctrl + X	切り取り
Ctrl + C	コピー
Ctrl + V	貼り付け
Ctrl + Z	元に戻す
Ctrl + Y	やり直す
Ctrl + A	すべて選択
Ctrl + F	検索・置換
Ctrl + スペース	コマンド一覧
Tab	次を選択（フローチャートだけ）
矢印キー	選択中のブロックを移動
Home	先頭までスクロール（フローチャートだけ）
End	最後尾までスクロール（フローチャートだけ）
Page Up	カーソルを行頭へ移動
Page Down	カーソルを行末へ移動
+	拡大
-	縮小

4-2-3 DriveProgramming 領域

DriveProgramming 領域には、プログラムの設計状態が表示されます。

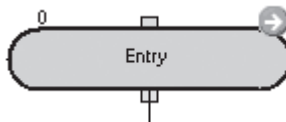


DriveProgramming 領域は、タブにまとめられたさまざまなページを含みます。各タブは、フローチャートまたはテキストのタスクおよびサブルーチンです。各タブのタブタイトルを右クリックすると、タスクまたはサブルーチンを削除したり、名前を変更したりすることができます。



フローチャートプログラム

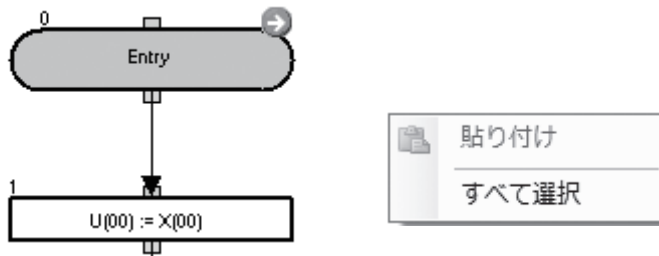
フローチャートプログラムでは、1つの表示単位を「ブロック」と呼んでいます。複数のブロックを領域上に配置し、そのブロック間の相互関係を設定してプログラムを作成します。フローチャートプログラムでは、正常にプログラムがコンパイルされていれば、緑地の白抜き矢印アイコンによって各フローチャートタスクの始点が強調表示されます。また、正常にプログラムがコンパイルされると、各ブロックの左上にブロック No. が表示されます。ブロック No. は、プログラム全体の各ブロックに 0 から始まる通し番号を示したものです。テキストプログラム変換時の行番号とは異なります。



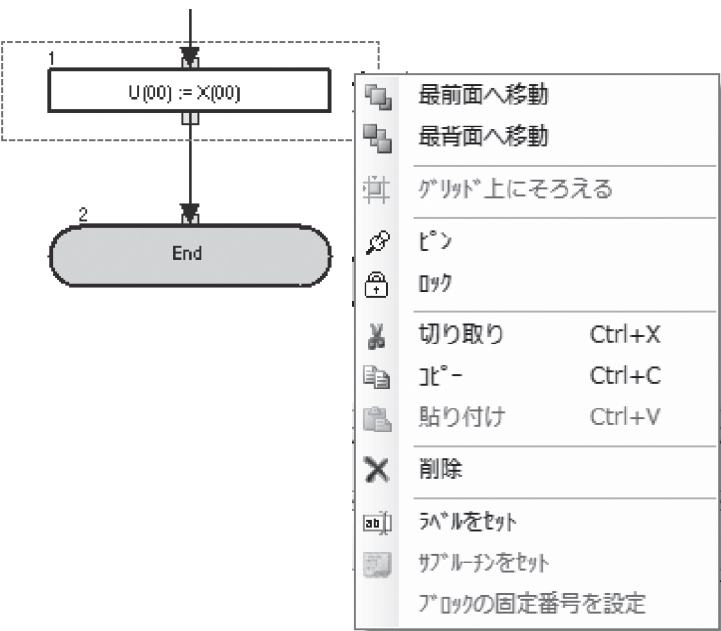
アウトプットウィンドウには、プログラムが正常にコンパイルされたかどうかが表示されます。コンパイル時に異常が発生したプログラムでは、赤の感嘆符アイコンにより、フローチャートプログラムで誤りのあるブロックを識別できます。異常アイコン上にマウスを移動させるとコンパイル異常が表示されますが、アウトプットウィンドウのエラーリストでも確認できます。



フローチャートプログラム内で何もブロックを選択せずに右クリックすると、直前にコピーしたブロックを貼り付けたり、全てのブロックを選択したりできます。



フローチャートプログラム内のブロックを右クリックすると、多数のオプションを含むポップアップメニューが表示されます。



以下に、フローチャートプログラムで利用できるメニューコマンドについて説明します。

コマンド	説明
[最前面に移動]	選択中のブロックをほかのブロックの前面に配置します。
[最背面に移動]	選択中のブロックをほかのブロックの背面に配置します。
[ピン]	選択中のブロックを図表内の現在の位置に固定します。これにより、ドラッグ操作で移動できなくなります。
[ロック]	[ピン]と同様に動作し、さらにブロックのプロパティを読み取り専用を設定します。
[切り取り]	選択中のブロックを削除し、あとで貼り付けることができるようにクリップボードに保存します。
[コピー]	選択中のブロックを、あとで貼り付けることができるようにクリップボードに保存します。
[貼り付け]	直前にクリップボードにコピーされた内容を貼り付けます。ほかのアプリケーションソフトに、画像データとして貼り付けることも可能です。
[ラベルをセット]	選択中のブロックにラベル名を設定します。
[サブルーチンをセット]	選択中のブロックにサブルーチン名を設定します。サブルーチンを設定できるブロックだけ有効表示になります。

テキストプログラム

テキストプログラムでは、テキスト言語を使用してプログラムを作成していきます。
テキストプログラムで正常にコンパイルできない場合は、プログラムの異常がアウトプットウィンドウに表示されます。また、異常が発生している行が赤の下線付きで示されます。

```
..... Dummy UL01 := A038 .....
```

選択中のテキストを右クリックすると、ポップアップメニューが表示されます。



以下に、テキストプログラムで利用できるメニューコマンドについて説明します。

コマンド	説明
[検索]	プログラム上で、選択中のテキストを検索します。
[置換]	プログラム上で、選択中のテキストを置き換えます。
[切り取り]	選択中のテキストを削除し、あとで貼り付けることができるようにクリップボードに保存します。
[コピー]	選択中のテキストを、あとで貼り付けることができるようにクリップボードに保存します。
[貼り付け]	直前にクリップボードにコピーされた内容を貼り付けます。他のアプリケーションソフトに画像データとして貼り付けることも可能です。
[サブルーチンに移動]	選択中のテキストサブルーチンにジャンプします。
[ラベルに移動]	選択中のテキストラベルにジャンプします。
[元に戻す]	最後に加えられた変更を元の状態に戻します。
[やり直す]	最後に元に戻された変更を回復します。
[ヘルプ]	CX-Drive のヘルプを表示します。

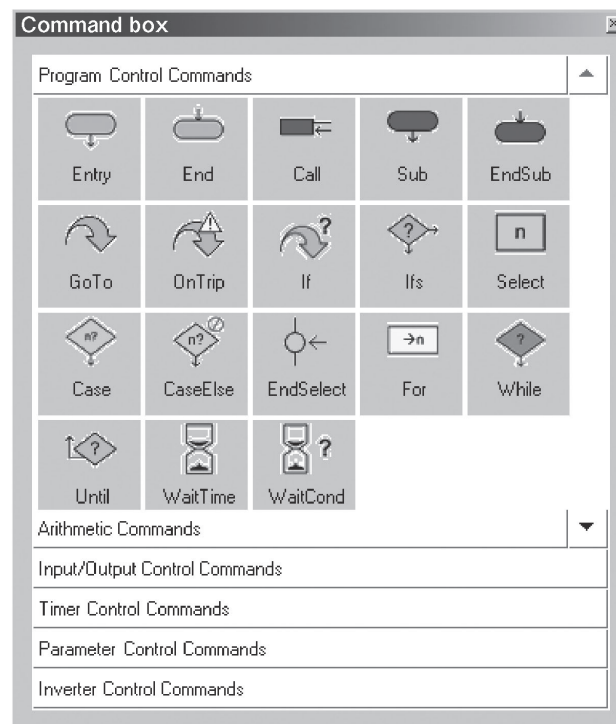
4-2-4 コマンドボックスウィンドウ

コマンドボックスウィンドウを使用すると、ドラッグ＆ドロップにより、ブロックを DriveProgramming 領域に追加できます。フローチャートのブロックが、命令カテゴリ別にまとめて表示されます。

コマンドボックスは、DriveProgramming エディタを起動すると表示されます。[表示] メニューの [ドライブプログラミング] から [コマンドボックス] をクリックして、表示／非表示を切り替えることもできます。

CX-Drive の初期設定で、コマンドボックスウィンドウは CX-Drive の右側に表示されます。

ウィンドウのタイトルバー（太くなった枠）をダブルクリックすると、独立したウィンドウ表示に切り替えることができます。



さらに、マウスで右クリックすることによって表示スタイルを選択できます。スタイルは、[大アイコン]、[小アイコン]、[一覧] の3つから選択できます。いずれのスタイルの場合でも、マウスカーソルをブロック上に移動すると、簡単なヘルプテキストが表示されます。

カテゴリに属するブロックを表示するには、そのカテゴリのタイトルをクリックします。

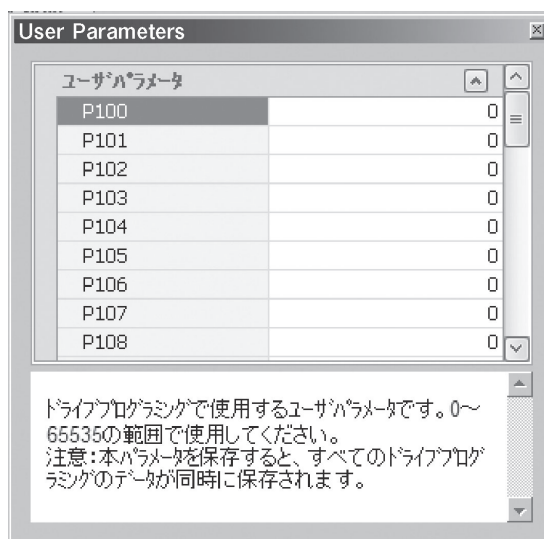
4-2-5 ユーザパラメータウィンドウ

ユーザパラメータウィンドウでは、プログラムの変数として動作する DriveProgramming ユーザパラメータを編集できます。このパラメータは、カテゴリに分けて表示されます。

ユーザパラメータは、DriveProgramming エディタを起動すると表示されます。[表示] メニューの [DriveProgramming] から [ユーザパラメータ] をクリックして、表示／非表示を切り替えることもできます。

CX-Drive の初期設定で、ユーザパラメータウィンドウは CX-Drive の右側に表示されます。

ウィンドウのタイトルバー（太くなった枠）をダブルクリックすると、独立したウィンドウ表示に切り替えることができます。



パラメータの値を変更するには、その行にカーソルを移動し、パラメータ No. の右にある編集ボックスをクリックして、新しい値を入力します。入力した値が有効な範囲を超えている場合は、警告が表示されます。

ウィンドウの下部に、パラメータに関する説明が表示されます。

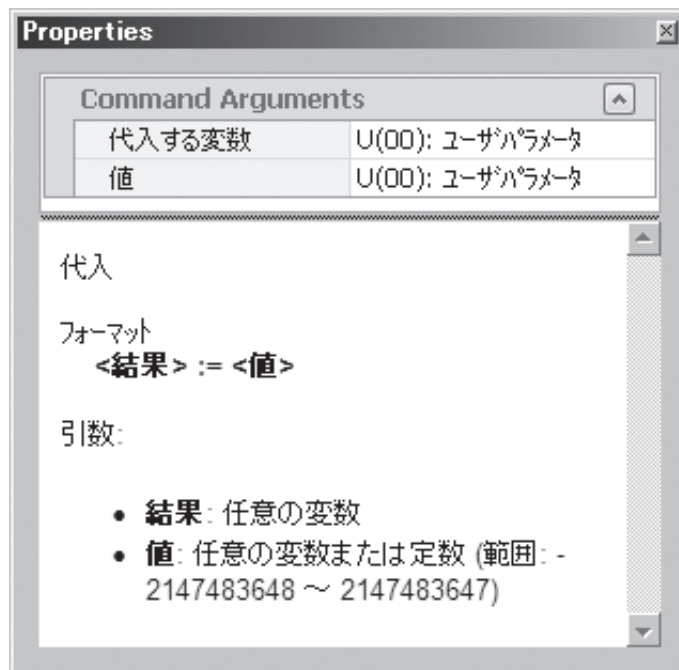
4-2-6 プロパティウィンドウ

プロパティウィンドウでは、フローチャートプログラムでブロックを選択した場合に、そのブロックのプロパティを編集できます。

プロパティは、DriveProgramming エディタを起動すると表示されます。[表示] メニューの [ドライブプログラミング] から [プロパティ] をクリックして、表示／非表示を切り替えることもできます。

CX-Drive の初期設定で、プロパティウィンドウは CX-Drive の右側に表示されます。

ウィンドウのタイトルバー（太くなった枠）をダブルクリックすると、独立したウィンドウ表示に切り替えることができます。



ブロックの変数を変更するには、フローチャートプログラムのブロックを選択し、プロパティウィンドウ内で入力したい箇所にカーソルを持ていきます。

- ブロックの変数にオプションがある場合は、現在の値をクリックすると、右端に  のアイコンが表示されます。 をクリックするか、現在の設定値をダブルクリックすると、プルダウンメニューが表示され、選択可能なオプションが展開されます。
- ブロックの変数にオプションがない場合は、現在の値をクリックし、新しい値を入力することによって値を変更できます。入力した値が有効な範囲を超えている場合は、警告が表示されます。
- ブロックの変数にオプションと任意設定の両方を設定できる場合は、現在の設定値をクリックして、直接数値を設定できます。また、現在の設定値をダブルクリックすると選択できるオプションがプルダウンメニューで表示されます。

4-2-7 アウトプットウィンドウのエラーリストタブ

アウトプットウィンドウの中にあるエラーリストのタブをクリックすると、DriveProgramming のエラーリストが表示されます。

DriveProgramming エディタで作成したプログラムをコンパイルした際に、発生したコンパイルの異常、および警告を表示します。

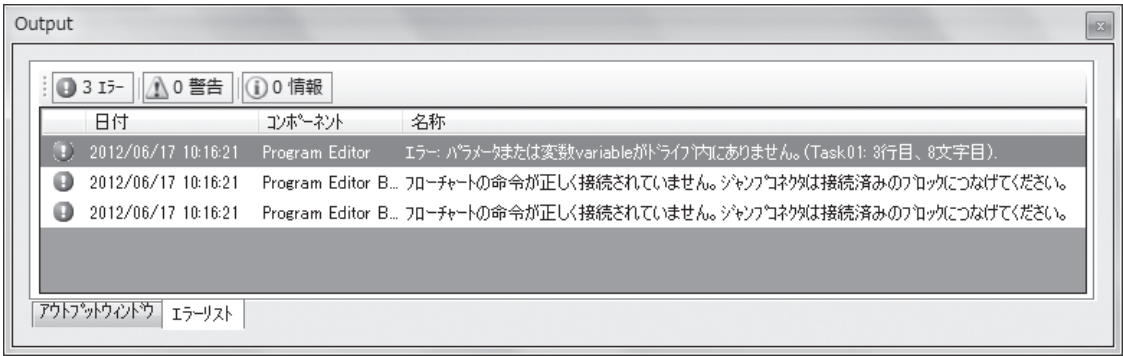
コンパイルの異常が表示されている場合、プログラムは正しくコンパイルされていません。このため、インバータへの転送やフローチャートプログラムとテキストプログラムの変換ができません。

コンパイルが正しく完了しても、プログラムに修正すべき内容がある場合に警告内容が表示されます。

エラーリストは、コンパイルが完了するたびに自動的に更新され、解決した異常や警告はクリアされます。

CX-Drive の初期設定で、アウトプットウィンドウは CX-Drive の下側に表示されます。

ウィンドウのタイトルバー（太くなった枠）をダブルクリックすると、独立したウィンドウ表示に切り替えることができます。



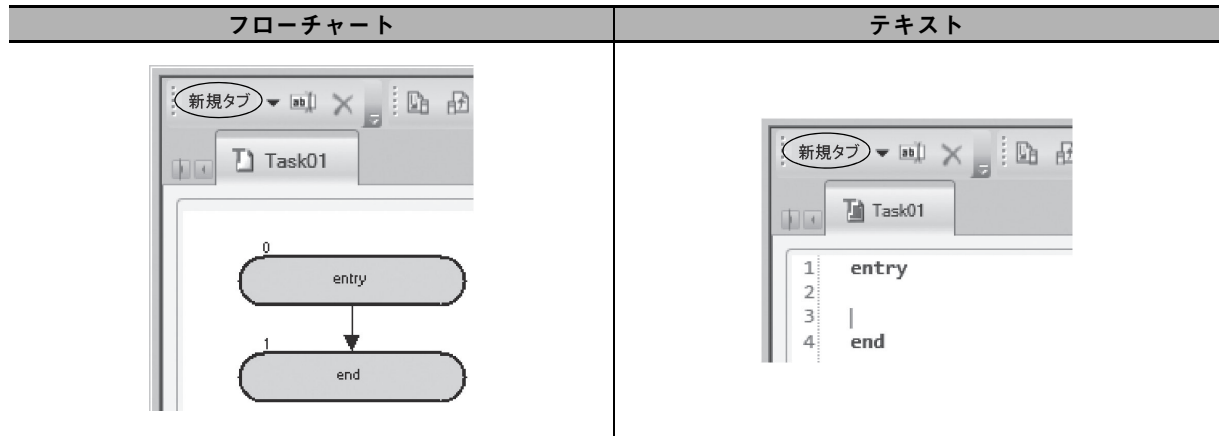
- 3 エラー [エラー] ボタンは、一覧への異常の表示／非表示を切り替えます。
- 0 警告 [警告] ボタンは、一覧への警告の表示／非表示を切り替えます。
- 0 情報 [情報] ボタンは、一覧への情報メッセージの表示／非表示を切り替えます。

一覧のメッセージには、以下の情報が示されます。

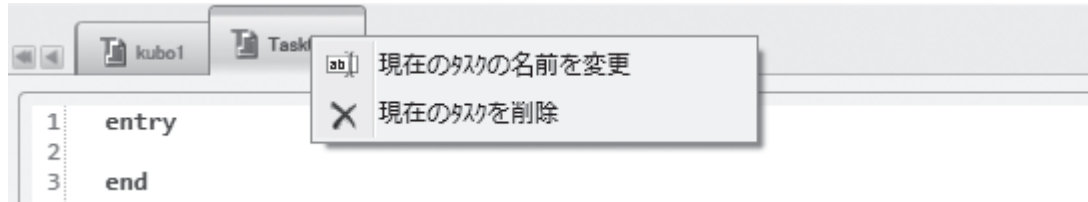
情報	説明
日付	異常が発生した日時
コンポーネント	異常が発生しているブロックを示します。
名称	異常または警告メッセージの名称や内容を説明します。

4-3 タスクの追加・削除と名前の変更

CX-Drive の初期設定で、DriveProgramming エディタを起動すると空白のタスクが表示されます。新たなタスクを追加する場合、DriveProgramming エディタのツールバーで「新規タブ」を選択し、「新規タスク（フローチャート）」または「新規タスク（テキスト）」を選択してください。選択すると、DriveProgramming エディタに新しいタスクが表示されます。表示されたタスクは、左側から優先順位が高く、2ms の処理時間の中で先に処理されます。また、全てのタスクの中では、「entry」命令で始まり、「end」命令で終わる必要があります。



DriveProgramming エディタのタスクのタブ部分で右クリックすると、選択しているタスクを削除したり、名前を変更したりすることができます。



4-4 サブルーチンの挿入・削除と呼び出し

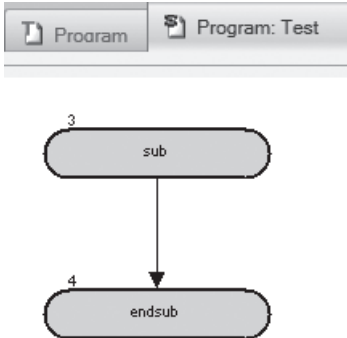
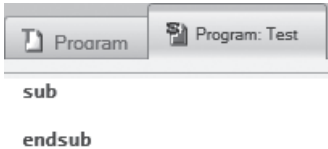
サブルーチンを挿入するには、サブルーチンを挿入したいタスクのタブを選択したあと、DriveProgramming エディタのツールバーで「新規タブ」を選択し、「新規サブルーチン（フローチャート）」または「新規サブルーチン（テキスト）」を選択してください。

選択すると、DriveProgramming エディタに新しいサブルーチンが表示されます。

タスクと同様、サブルーチンのタブ上で右クリックすると、削除または名前の変更が可能です。

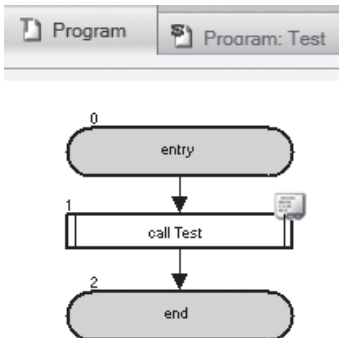
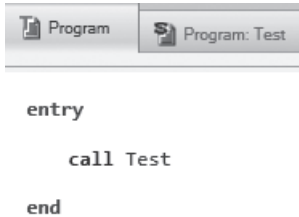
サブルーチンの名前は、所属するタスク名が先頭に表示され「タスク名：サブルーチン名」と表示されます。

すべてのサブルーチンは、「sub」命令で始まり、「endsub」命令で終わる必要があります。

フローチャート	テキスト
	

サブルーチンを実行するには、「call」命令でサブルーチン名を指定します。

各タスクで呼び出せるサブルーチンは、各タスクに所属するサブルーチンだけです。タスク間で共用することはできません。複数のタスクで同じサブルーチン処理を行う場合は、同じサブルーチンを各タスクに作成してください。

フローチャート	テキスト
	

4-5 フローチャートプログラムの作成

DriveProgramming のプログラムを作成する場合、タスクやサブルーチン単位で、フローチャートプログラムまたはテキストプログラムを選択できます。

フローチャートプログラムは、以下の手順に従って作成してください。

- 1 DriveProgramming エディタを開いてください。**
DriveProgramming の補助ウィンドウ（コマンドボックス、ユーザパラメータ、プロパティ、およびエラーリストタブ）が自動的に表示されます。
- 2 ツールバーで、[新規タブ] から [新規タスク（フローチャート）] または [新規サブルーチン（フローチャート）] を選択してください。**
- 3 コマンドボックスウィンドウから命令を選択し、DriveProgramming エディタにドラッグ＆ドロップしてください。**
エディタ上に配置された各ブロックの左上には、コンパイル後、ブロック No. が表示されます。
ブロック No. は、プログラム全体の各ブロックに 0 から始まる通し番号を示したものです。テキストプログラム変換時の行数とは異なります。
- 4 ブロックをクリックしてプロパティを編集してください。**
プロパティウィンドウでブロックに関係する変数を編集します。
- 5 各ブロックをプログラム順に接続してください。**
表示されたブロックのオレンジの接点から別のブロックの緑の接点にドラッグすると接続できます。
- 6 ユーザパラメータウィンドウで、ユーザパラメータを編集してください。**
- 7 プログラムのコンパイル、インバータへの転送、プログラムの保存などを行ってください。**
プログラムのコンパイルを行って、プログラムにコンパイル異常が発生しないか確認します。コンパイルが正常終了すれば、インバータへ転送することが可能です。
プログラムを保存するにはプロジェクト全体を保存します。また、プログラムのエクスポート機能でプログラムだけを分離して保存することも可能です。

4-6 テキストプログラムの作成

DriveProgramming のプログラムを作成する場合、タスクやサブルーチン単位で、フローチャートプログラムまたはテキストプログラムを選択できます。

テキストプログラムは、以下の手順に従って作成してください。

- 1 DriveProgramming エディタを開いてください。**
DriveProgramming の補助ウィンドウ（コマンドボックス、ユーザパラメータ、プロパティ、およびエラーリストタブ）が自動的に表示されます。
- 2 ツールバーで、[新規タブ] から [新規タスク（テキスト）] または [新規サブルーチン（テキスト）] を選択してください。**
- 3 テキストプログラムの編集方法は 3 つあります。**
 - ・ 手動入力
 - ・ テキストコマンド一覧の呼び出し（Ctrl + スペース）
 - ・ コマンドボックスウィンドウから命令をドラッグ & ドロップ
- 4 各命令の変数を設定していきます。**
コマンドボックスウィンドウからドラッグ & ドロップされた命令と、テキスト命令一覧（Ctrl + スペース）で選択した命令には、緑の背景で必要な変数が表示されます。
各変数を設定して、命令を完成させてください。
手動入力で命令を入力する場合、入力ごとに表示されるサポート表示を参考にして命令を完成させてください。各命令の詳細内容は □「第 6 章 DriveProgramming 命令」を参照してください。
コンパイル時に不要な行やスペースが削除されますので、行やスペースを意識せずプログラムを作成してください。
- 5 プログラムのコンパイル、インバータへの転送、プログラムの保存などを実行してください。**
プログラムのコンパイルを行って、プログラムにコンパイル異常が発生しないか確認します。
コンパイルが正常終了すれば、インバータへ転送することが可能です。
プログラムを保存するにはプロジェクト全体を保存します。また、プログラムのエクスポート機能でプログラムだけを分離して保存することも可能です。

4-7 転送（アップロード）したプログラムの編集

プログラムを持つインバータからプログラムを転送（アップロード）し、編集することができます。以下の手順に従って、プログラムを編集してください。

- 1 DriveProgramming エディタを開いてください。**
DriveProgramming の補助ウィンドウ（コマンドボックス、ユーザパラメータ、およびプロパティ）が自動的に表示されます。
- 2 CX-Drive をオンライン接続してください。**メニューから【ドライブ】を選び、【オンライン接続】を選択するか、CX-Drive のツールバーから【オンライン接続】アイコンをクリックしてください。
- 3 DriveProgramming エディタのツールバーにある【ドライブから転送】アイコンをクリックします。**
プログラムがドライブ（インバータ）から転送され、DriveProgramming エディタの DriveProgramming 領域に自動的に表示されます。
- 4 転送（アップロード）してきたプログラムを編集します。**
インバータ内部には、「コンパイル後のプログラム」がダウンロードされています。そのため、転送（アップロード）されたプログラムはテキストプログラムで表示されます。フローチャートプログラムで表示するには、DriveProgramming エディタのツールバーにある【全プログラムをフローチャートに変換】をクリックし、フローチャートプログラムに変換してください。
- 5 プログラムのコンパイル、インバータへの転送、プログラムの保存などを実行してください。**
プログラムのコンパイルを行って、プログラムにコンパイル異常が発生しないか確認します。コンパイルが正常終了すれば、インバータへ転送することが可能です。プログラムを保存するにはプロジェクト全体を保存します。また、プログラムのエクスポート機能でプログラムだけを分離して保存することも可能です。

DriveProgramming 機能のプログラムが存在する場合、CX-Drive ツールバーの【転送【パソコン → ドライブ】】／【転送【ドライブ → パソコン】】アイコンを使用しても、インバータとの間でプログラムを転送することができます。この場合、パラメータ、プログラム、またはその両方を転送するかどうかをたずねるメッセージダイアログでプログラムを選択してください。

4-8 プログラムの保存

DriveProgramming 機能で作成したプログラムは、以下に示す2つの方法で保存できます。
用途に合わせて適切な方法を選んで、保存してください。

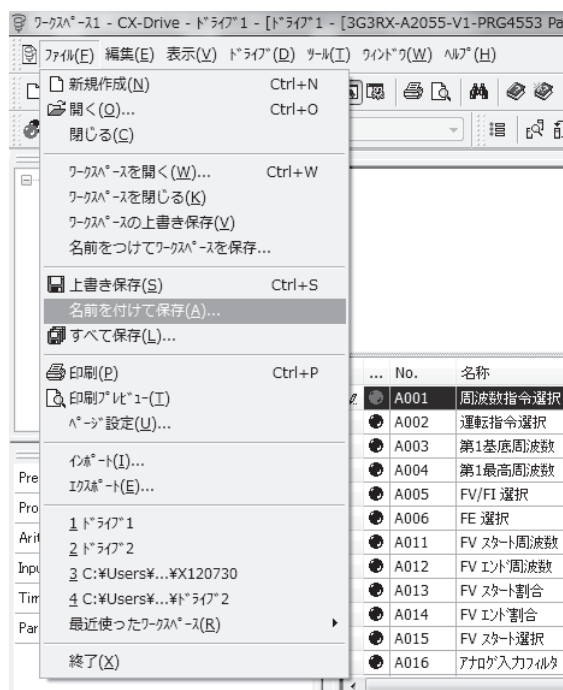
● CX-Drive のプロジェクト全体の保存

CX-Drive で作成したプロジェクト全体を保存すれば、DriveProgramming のプログラムを含むドライブデータの全体を保存することができます。

メニューバーの「ファイル」をクリック、「名前を付けて保存」を選択し、ファイル名を入力してください。

保存したプロジェクトを開くと、そのプロジェクトに含まれる DriveProgramming が自動的にロードされます。

プログラムは、ワークスペースにある DriveProgramming をダブルクリックして DriveProgramming エディタを起動すると表示されます。

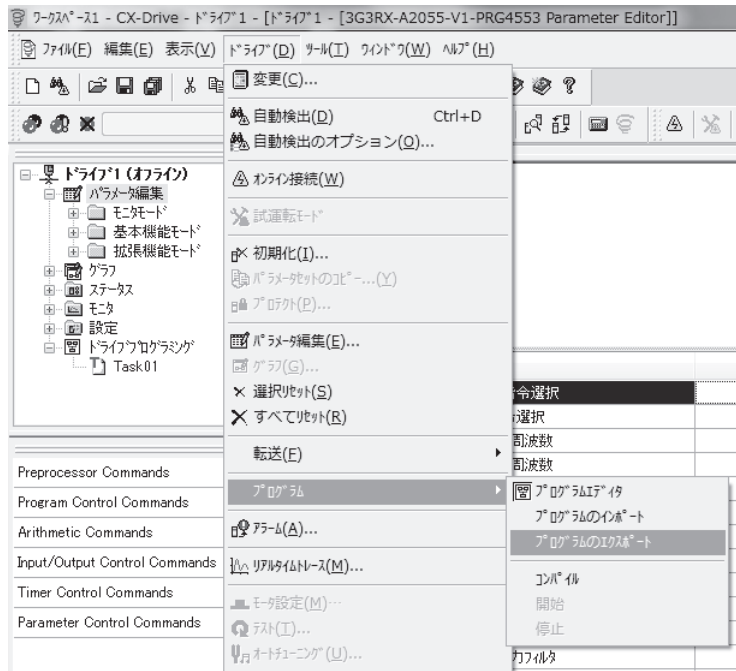


● DiveProgramming のエクスポート保存

DriveProgramming 機能のプログラムだけを分離して保存することも可能です。

メニューバーの「ドライブ」をクリック、「プログラム」の中から「プログラムのエクスポート」を選択し、ファイル名を入力してください。他のドライブ情報と分離してプログラムだけを保存します。

エクスポートしたプログラムファイルを CX-Drive にインポートするには、メニューバーの「ドライブ」をクリック、「プログラム」の中から「プログラムのインポート」を選択し、インポートするファイルを選択してください。



4-9 プログラムの転送と照合

CX-Drive の DriveProgramming 機能 とインバータ本体の間でプログラムの転送と照合が可能です。同時にパラメータの転送と照合も実行することができます。

転送（パソコンからインバータ）

作成したプログラムをコンパイル（プログラムチェック）し、異常がなければインバータに転送（ダウンロード）します。

〔ドライブ〕メニューで〔転送〕から〔転送〔パソコン→ドライブ〕〕を選択することによって転送（ダウンロード）することができます。

転送（インバータからパソコン）

インバータにあるプログラムやパラメータを DriveProgramming に転送（アップロード）します。

〔ドライブ〕メニューで〔転送〕から〔転送〔ドライブ→パソコン〕〕を選択することによって転送（アップロード）することができます。

照合（パソコンとインバータ）

DriveProgramming のプログラムやパラメータをインバータ本体のデータと照合します。

〔ドライブ〕メニューで〔転送〕から〔照合〔ドライブ-パソコン〕〕を選択すれば照合することができます。

部分転送（パソコンからインバータ）

〔ドライブ〕メニューで〔転送〕から〔部分転送〔パソコン→ドライブ〕〕を選択することによって部分転送（ダウンロード）することができます。

部分転送（インバータからパソコン）

〔ドライブ〕メニューで〔転送〕から〔部分転送〔ドライブ→パソコン〕〕を選択することによって部分転送（アップロード）することができます。

4-10 プログラムの実行 (DriveProgramming 機能選択)

プログラムをインバータに転送（ダウンロード）後、プログラムを実行するには以下の2つの方法があります。

CX-Drive からのプログラム実行

作成したプログラムをデバッグするために、プログラムを実行するには、CX-Drive からのプログラム実行が便利です。

DriveProgramming エディタのツールバーにある  アイコンでプログラムを起動し、 アイコンでプログラムを停止できます。ただし、CX-Drive からのプログラム実行は以下の場合に有効です。

- CX-Drive が接続されている場合に有効です。
プログラムを起動したあと、CX-Drive の接続が切れても、停止しませんので、必ず電源 OFF など、インバータを直ぐに停止できる状態でご使用ください。
- CX-Drive のプログラムとインバータのプログラムがコンパイル後の照合で一致する場合に、起動が有効です。
- プログラムの停止は、CX-Drive のプログラムに関係なく実行されます。

実際のアプリケーションでのプログラム実行

インバータのパラメータを DriveProgramming 機能選択 (A017) で設定してプログラムを実行します。

プログラムがひと通りの処理を完了し、「end」命令に達したときは、再度プログラムを起動しなければ実行されません。

プログラムを繰り返すためには、プログラム内でループを構成して「end」命令に達しないように作成してください。

モータ運転時に DriveProgramming 機能を停止させた場合、停止方法選択 (b091) の設定に従ってモータを停止（減速停止またはフリーランストップ）させます。

パラメータ No.	機能名称	データ	説明
A017	DriveProgramming 機能選択	00: 無効（初期データ）	DriveProgramming 機能を無効にします。 プログラムを実行しません。 プログラムの実行中にデータを「00: 無効」に変更すると、プログラムが停止します。
		01: 有効（多機能入力 PRG 端子での起動／停止）	「82: PRG」機能に設定した多機能入力端子*1を入力すると、DriveProgramming のプログラムを起動します。
		02: 有効 （電源投入／遮断による起動／停止）	DriveProgramming のプログラムをインバータ電源投入後、自動的に起動します。 プログラムの停止中にデータを「02: 有効」に変更すると、プログラムが起動します。

パラメータ No.	機能名称	データ	説明
C001 ～ C007	【MX2・RX 共通機能】 多機能入力 S1 ～ S7 選択	82: PRG (DriveProgramming 起動)	DriveProgramming 機能選択 (A017) に「01: 有効 (多機能入力 PRG 端子での起動/停止)」を設定した場合、この設定を行った多機能入力でプログラムを起動します。
C008	【RX だけの機能】 多機能入力 S8 選択		

*1. 多機能入力端子は、【MX2】 S1 ～ S7、【RX】 S1 ～ S8 が対象となります。



安全上の要点

DriveProgramming 機能のプログラムが停止すると、DriveProgramming で制御している多機能出力はプログラム停止直前の状態を保持します。

このため、インバータの DriveProgramming 機能のプログラム停止を DriveProgramming 起動信号とアラーム (トリップ) 信号で検出し、インバータ周辺機器を安全に停止させる配線を行ってください。



使用上の注意

DriveProgramming 機能のプログラムが停止した場合、各機能変数の状態は下記のようになりますので、各状態を認識して、次の対応を実施してください。

- 多機能出力 (実際のインバータ出力端子) の状態はプログラム停止前の状態を保持します。ただし、DriveProgramming 機能の汎用出力 MO1 ～ MO6 を設定していない出力は、通常のインバータ多機能出力として動作します。
- 周波数指令、加減速時間、アナログ出力に DriveProgramming 機能を設定した場合、各設定値はプログラム停止前の状態を保持します。
- ユーザパラメータ変数、内部ユーザ変数、内部ユーザ接点はプログラム停止前の状態を保持します。
- 多機能入力やアナログ入力など実際のインバータ入力端子状態は、入力状態を保持せず常に状態を更新します。
- 多機能出力変数 (RUN、FA1、AL、・・・などの機能ビット) やインバータモニタ変数は、状態を保持せずインバータの状態に合わせて常に状態を更新します。
- DriveProgramming の多機能入力変数 (FW、RV、CF1、・・・などの機能ビット) とタイマ変数だけ、プログラム停止と同時にクリアされ、全ての状態がゼロとなります。

CX-Drive で DriveProgramming 機能を起動/停止するとき、CX-Drive はインバータのパラメータ DriveProgramming 機能選択 (A017) の値を、起動時は「02」に、停止時は「00」と一時的に (RAM 上のデータだけ) 変更しています。

CX-Drive で DriveProgramming 機能の起動/停止を行ったあとは、次の対応をしてください。

- インバータの電源を再投入して、A017 を EEPROM に保存されている値に戻してください。
- インバータの電源を再投入するまで、以下の EEPROM 保存操作を実行しないでください。
 - CX-Drive からパラメータの部分転送 (ダウンロード) を行う。
 - Modbus 通信または通信オプション経由で「enter」命令を発行する。

A017=02 を設定している場合に、CX-Drive からプログラムを再起動する場合は、いったん電源起動時に起動されたプログラムを停止してから実行してください。

電源起動時に起動されたプログラムが起動されたままでは、CX-Drive から再起動しません。

既にインバータが運転している状態で、CX-Drive のプログラム起動ボタンをクリックしていくと、最終的に下記のメッセージが表示されます。

「ドライブと CX-Drive の初期化データが異なります。」

この表示で、「はい」をクリックすると、インバータの運転状態にかかわらず、強制的にプログラムを起動します。

起動する際は、装置の状態を確認し、安全を確保してから起動してください。

4-11 その他の便利な機能

フローチャートからテキストへの変換

フローチャートからテキストに変換する場合、2つの方法があります。

指令	アイコン	説明
テキスト変換		現在のフローチャートのタスクやサブルーチンをテキストに変換します。
全体テキスト変換		プログラム全体をテキストに変換します。



使用上の注意

- ・フローチャートプログラムをテキストプログラムに変換する際、プログラムをいったんコンパイルします。このため、コンパイル異常が発生する場合は変換できません。
- ・フローチャートプログラムの各ブロック左上には、コンパイル後、ブロック No. が表示されます。
ブロック No. は、プログラム全体の各ブロックに 0 から始まる通し番号を示したものです。テキストプログラム変換時の行数とは異なります。

テキストからフローチャートへの変換

テキストからフローチャートに変換する場合、2つの方法があります。

指令	アイコン	説明
フローチャート変換		現在のフローチャートタスクやサブルーチンをフローチャートに変換します。
全体フローチャート変換		プログラム全体をフローチャートに変換します。



使用上の注意

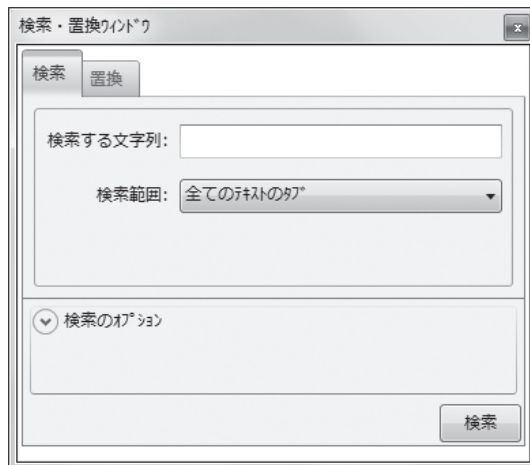
- ・テキストプログラムをフローチャートプログラムに変換する際、プログラムをいったんコンパイルします。このため、コンパイル異常が発生する場合は変換できません。
- ・フローチャートプログラムの各ブロック左上には、コンパイル後にブロック No. が表示されます。
ブロック No. は、プログラム全体の各ブロックに 0 から始まる通し番号を示したものです。テキストプログラム変換時の行数とは異なります。
- ・テキストプログラムで作成したコメント、エイリアス定義、リージョン定義などは、変換時のコンパイルにより削除され、動作は同じですがその形式や内容は一部変更されます。

検索と置換機能

検索と置換機能は、テキストプログラムだけで使用できます。検索と置換機能を使用することで、テキストプログラム内の任意の文字列を検索、または置換できます。

検索機能を使用するには、 アイコンをクリックするか、ショートカットキー Ctrl + F を押して、[検索] タブを選択します。

置換機能を使用するには、 アイコンをクリックするか、ショートカットキー Ctrl + F を押して、[置換] タブを選択します。



検索機能



置換機能

コメントの追加（テキストプログラム）

コメントは、テキストプログラムのタスクまたはサブルーチンに追加できます。フローチャートプログラムではコメントを追加できません。テキスト行にコメントを追加するには、「」に続けてコメントを入力します。コメントは緑色で表示されます。

● 例

```
#alias global Time as U(10)           ' Timer time
#alias global AppTimer as TD(0)        ' Timer TD(0)
#alias global Temp as UL(05)           ' Internal use
```



使用上の注意

- テキストプログラムで作成したコメントは、インバータへ転送（ダウンロード）時またはプログラム変換時のコンパイルで削除されます。作成したコメントを保存するには、コンパイル前のプログラムで保存し、管理してください。CX-Drive のプロジェクトファイル全体の保存、または、プログラムのエクスポートファイルの保存でプログラムを保存できます。
- プログラムの照合は、「コンパイル後のプログラム」とインバータ内部のプログラムを比較しますので、コメント、エイリアス定義、リージョン定義などの照合は含まれません。

エイリアス定義（テキストプログラム）

エイリアス定義は、テキストプログラムのタスク内で「entry」指令の前に定義できます。サブルーチン内やフローチャートプログラムでは、エイリアスを定義できません。

エイリアス定義とは、パラメータ、変数、命令、および数値定数に指定した任意の名称をつけることです。

エイリアス定義を使用することで、プログラム内で指定した名称をパラメータ、変数、命令および数値と同様にあつかうことができます。プログラムの視認性を高めたい場合に役立ちます。

- ・ **ローカルエイリアス**：このエイリアス定義は、現在のタスクやサブルーチン内で使用できます。プログラム内のほかのタスクやサブルーチンでは使用できません。以下は、タスク内でのローカルエイリアス定義のフォーマットです。

```
#alias local alias as replacement
```

● 例

```
#alias local ON_ as 1
#alias local OFF_ as 0
#alias local Monitor_1 as UMon(0)
#alias local MaxFrequency as A004
#alias local Count as U(00)
#alias local Dummy_1 as UL(00)

entry
```

- ・ **グローバルエイリアス**：このエイリアス定義は、プログラム内のすべてのタスクやサブルーチンで使用できます。最大5つのタスクのうち、どのタスクでも設定できます。以下は、グローバルエイリアス定義のフォーマットです。

```
#alias global alias as replacement
```

● 例

```
#alias global const_100 as 100
#alias global Acceleration as F002
#alias global Deceleration as F003
#alias global Time as U(10)
#alias global AppTimer as TD(0)
#alias global Temp as UL(05)

entry
```



使用上の注意

- ・ テキストプログラムで作成したエイリアス定義は、インバータへ転送（ダウンロード）時またはプログラム変換時のコンパイルで削除され、指定した任意の名称は通常名称または数値に変換されます。作成したエイリアス定義を保存するには、コンパイル前のプログラムで保存し、管理してください。CX-Driveのプロジェクトファイル全体の保存、または、プログラムのエクスポートファイルの保存でプログラムを保存できます。
- ・ プログラムの照合は、「コンパイル後のプログラム」とインバータ内部のプログラムを比較しますので、コメント、エイリアス定義、リージョン定義などの照合は含まれません。
- ・ すでに使用している変数や命令と同じものをエイリアス定義に使用できません。コンパイル異常が表示されます。

リージョン定義 (テキストプログラム)

リージョン定義は、テキストプログラムのタスクやサブルーチン内に定義できます。フローチャートプログラムでは、リージョンを定義できません。

リージョン定義とは、領域を指定することでプログラムをグループ化することです。

リージョン定義を使用することで、プログラムを任意に分割して表示でき、分割した領域ごとの折り畳みができるので、プログラムを見やすくしたい場合に役立ちます。

● 例

```
#region Alias
#alias global const_100 as 100
#alias global Acceleration as F002
#alias global Deceleration as F003
#alias global Time as U(10)
#alias global AppTimer as TD(0)
#alias global Temp as UL(05)
#endregion

entry

#region Start
    Acceleration := const_100
    Deceleration := const_100
    Time := 500
    Temp := 10000
    set-freq := 1000
    Fw := 1
#endregion
#endregion
```



使用上の注意

- テキストプログラムで作成したリージョン定義は、インバータへ転送（ダウンロード）時またはプログラム変換時のコンパイルで削除されます。
作成したリージョン定義を保存するには、コンパイル前のプログラムで保存し、管理してください。CX-Drive のプロジェクトファイル全体の保存、または、プログラムのエクスポートファイルの保存でプログラムを保存できます。
- プログラムの照合は、「コンパイル後のプログラム」とインバータ内部のプログラムを比較しますので、コメント、エイリアス定義、リージョン定義などの照合は含まれません。

5

DriveProgramming のユーザ変数

この章では、DriveProgramming のユーザ変数について説明します。

5-1 ユーザ用の変数とユーザパラメータ	5-2
5-2 入出力端子変数	5-5
5-3 タイマ変数	5-10
5-4 インバータ設定変数	5-12
5-5 インバータモニタ変数	5-14
5-6 多機能入力変数	5-17
5-7 多機能出力変数	5-20

5-1 ユーザ用の変数とユーザパラメータ

プログラムを作成するために、ユーザパラメータ変数、内部ユーザ変数、内部ユーザ接点を用意しています。

プログラムのユーザインターフェース、演算の開始データ、演算途中のデータ格納や退避、データの保存などに使用してください。

ユーザパラメータ変数 U(00) ～ U(31)

DriveProgramming 機能のユーザパラメータ変数 U(00) ～ U(31) は、符号無 1 ワードの変数です。インバータ本体の DriveProgramming ユーザパラメータ U00 ～ U31 (P100 ～ P131) と DriveProgramming 機能のユーザパラメータ変数 U(00) ～ U(31) は、それぞれが対応しています。以下の用途で使用してください。

用途	説明
ユーザインターフェース	演算結果を調整するためのパラメータとして使用します。アプリケーションの状態に合わせて、インバータのパラメータ (P100 ～ P131) から調整することが可能です。
演算の開始データ	ユーザパラメータ変数 U(00) ～ U(31) は演算用の変数として使用できます。EEPROM に保存された P100 ～ P131 のデータは演算の開始データとして使用できます。
演算途中のデータ格納や退避	演算途中のデータを格納したり、退避させたりする変数として使用できます。
データの保存	演算結果を電源遮断しても保存しておくために、「eepwrt」命令を使用して設定データを EEPROM に保存することができます。U(29) ～ U(31) を使用すれば、インバータ電源遮断時の演算結果を自動的に EEPROM に保存し、次の電源投入後に開始データとして使用することが可能です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
U(00) ～ U(31)	ユーザパラメータ変数 (P100 ～ P131 に対応した変数)	0 ～ 65,535	0	—	符号無し 1 ワード	R/W

- パラメータ P100 ～ P131 の EEPROM に保存されたデータは、電源投入時にユーザパラメータ変数 U(00) ～ U(31) に設定されます。
- プログラム起動後のユーザパラメータ変数 U(00) ～ U(31) の状態は、パラメータ P100 ～ P131 をモニタすることで確認できます。ただし、デジタルオペレータで表示させたパラメータデータは、表示させた瞬間のデータです。データを更新するには、いったんパラメータ No. を表示させたあと、再度データを表示させてください。
- パラメータ P100 ～ P131 をデジタルオペレータから変更しエンターキーを押すと、変更されたデータは各パラメータの EEPROM に保存され、現在のユーザパラメータ変数 U(00) ～ U(31) に反映されます。EEPROM に保存されたデータは、次の電源投入時から U(00) ～ U(31) に設定されます。
- ユーザパラメータ変数 U(00) ～ U(31) のデータが、プログラムで変更されてもパラメータ P100 ～ P131 の EEPROM には保存されません。
- プログラムからユーザパラメータ変数 U(00) ～ U(31) に設定するデータを EEPROM に保存する場合は、「eepwrt」命令のあとに「ChgParam」命令を使用して、保存したいデータを設定します。データは対応するパラメータ P100 ～ P131 の EEPROM に保存されます。
- ユーザパラメータ変数 U(29) ～ U(31) だけ、インバータ電源遮断時のデータを自動保存します。データは対応するパラメータ P129 ～ P131 の EEPROM に保存されます。ユーザパラメータ変数 U(00) ～ U(28) は、電源遮断時のデータを保存しません。

**使用上の注意**

DriveProgramming 機能のプログラムが停止した場合、ユーザパラメータ変数はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。

内部ユーザ変数 UL(00) ～ UL(07)

DriveProgramming 機能の内部ユーザ変数 UL(00) ～ UL(07) は、符号付き 2 ワードの変数です。四則演算結果など、演算中のデータ格納や退避に使用してください。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
UL(00) ～ UL(07)	内部ユーザ変数	-2,147,483,648 ～ 2,147,483,647	0	—	符号付き 2 ワード	R/W

- 電源投入時、内部ユーザ変数 UL(00) ～ UL(07) はゼロにクリアされます。EEPROM などでのデータ保存機能はありません。
- 開始データを設定したい場合は、開始データを UL(00) ～ UL(07) に設定するプログラムを作成してください。
- 内部ユーザ変数 UL(00) ～ UL(07) の上位ワードデータを、1 ワードデータのユーザパラメータ変数 U(00) ～ U(31) に移行するには、以下のような演算を使用します。
 正の値の上位ワードデータを代入する：U(00)=UL(00)/65536
 負の値の上位ワードデータを代入する：U(00)=UL(00)/65535
 「U(00)=UL(00)」を実行すると下位ワードデータが U(00) に移行します。

**使用上の注意**

DriveProgramming 機能のプログラムが停止した場合、内部ユーザ変数はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。

内部ユーザ接点 UB(0) ～ UB(7)

DriveProgramming 機能の内部ユーザ接点 UB(0) ～ UB(7) は、ビットアクセスの変数です。ビット演算のデータ格納や退避に使用してください。

UBw 変数を使用すれば、内部ユーザ接点 UB(0) ～ UB(7) を下位バイトとしたワードアクセスデータとして使用することもできます。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
UB(0) ～ UB(7)	内部ユーザ接点 (ビットアクセス)	0: OFF 1: ON	0	—	ビット	R/W

- 内部ユーザ接点 UB(0) ～ UB(7) は、ビット単位の変数として使用できます。
- 電源投入時、UB(0) ～ UB(7) はゼロにクリアされます。EEPROM などでのデータ保存機能はありません。
- 開始データを設定したい場合は、開始データを UB(0) ～ UB(7) に設定するプログラムを作成してください。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
UBw	内部ユーザ接点 (ワードアクセス)	0 ~ 255	0	—	符号無し 1ワード	R/W

- 内部ユーザ接点（ワードアクセス）UBw は、内部ユーザ接点 UB(0) ~ UB(7) をワード単位の変数として使用できる機能です。
- 内部ユーザ接点 UB(0) ~ UB(7) は下位バイトに配置されます。上位バイトのデータはゼロが読み出され、上位バイトに書き込んだ場合はその設定を無視します。

ビット	8 ~ 15	7	6	5	4	3	2	1	0
機能	未使用	UB(7)	UB(6)	UB(5)	UB(4)	UB(3)	UB(2)	UB(1)	UB(0)



使用上の注意

DriveProgramming 機能のプログラムが停止した場合、内部ユーザ接点はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。

5-2 入出力端子変数

インバータの多機能入力端子、多機能出力端子、多機能リレー出力、アナログ入力端子、アナログ出力端子を DriveProgramming 機能で使用するための変数について説明します。インバータの周辺機器と DriveProgramming 機能とのインターフェースに使用してください。

入力端子変数 X(00) ～ X(07)

インバータの多機能入力に汎用入力 MI1 ～ MI8 を設定すると、DriveProgramming 機能の入力端子変数 X(00) ～ X(07) として使用することができます。

Xw 変数を使用すれば、入力端子変数 X(00) ～ X(07) を下位バイトとしたワードアクセスデータとして使用することもできます。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
X(00) ～ X(07)	入力端子変数 (ビットアクセス)	0: OFF 1: ON	0	—	ビット	R

【MX2】

- 多機能入力 S1 ～ S7 端子とパルス列入力 RP 端子の状態をビット単位で変数に取り込みます。この変数は読出専用です。
- 多機能入力 S1 ～ S7 選択 (C001 ～ C007) に「56 ～ 62: 汎用入力 (MI1 ～ MI7)」を設定し、パルス列入力端子 RP 選択 (P003) に「02: MI8 (DriveProgramming 用汎用入力 8)」を設定すると、入力端子 S1 ～ S7 と RP 端子の状態を DriveProgramming の入力端子変数 X(00) ～ X(07) として取り込みます。
入力端子変数の番号 X(00) ～ X(07) は、S1 ～ S7 の端子番号ではなく、設定した汎用入力 MI1 ～ 7 の番号順で設定されます。

【RX】

- 多機能入力 S1 ～ S8 端子の状態をビット単位で変数に取り込みます。この変数は読出専用です。
- 多機能入力 S1 ～ S8 選択 (C001 ～ C008) に「56 ～ 63: 汎用入力 (MI1 ～ MI8)」を設定すると、入力端子 S1 ～ S8 の状態を DriveProgramming の入力端子変数 X(00) ～ X(07) として取り込みます。
入力端子変数の番号 X(00) ～ X(07) は、S1 ～ S8 の端子番号ではなく、設定した汎用入力 MI1 ～ 8 の番号順で設定されます。

機能変数	多機能入力選択／パルス列入力端子 RP 選択の設定 (【MX2】 C001 ～ C007, P003 / 【RX】 C001 ～ C008)
X(00)	56: MI1
X(01)	57: MI2
X(02)	58: MI3
X(03)	59: MI4
X(04)	60: MI5
X(05)	61: MI6
X(06)	62: MI7
X(07)	【MX2】 P003 = 02 / 【RX】 63: MI8

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
Xw	入力端子変数 (ワードアクセス)	0 ~ 255	0	—	符号無し 1ワード	R

- 入力端子変数 (ワードアクセス) Xw は、入力端子変数 X(00) ~ X(07) をワード単位の変数として使用できる機能です。この変数は読出専用です。
- 入力端子変数 X(00) ~ X(07) は下位バイトに配置され、上位バイトや使用していない入力端子変数はゼロが読み出されます。

ビット	8 ~ 15	7	6	5	4	3	2	1	0
機能	未使用	X(07)	X(06)	X(05)	X(04)	X(03)	X(02)	X(01)	X(00)



使用上の注意

- DriveProgramming 機能のプログラムが停止した場合、入力端子変数は状態を保持せず、常に入力端子の状態を更新します。
- 多機能入力端子 (【MX2】 S1 ~ S7 / 【RX】 S1 ~ S8) で MI1 ~ 8 (汎用入力 1 ~ 8) を選択した場合でも、多機能入力動作選択 (【MX2】 C011 ~ C017 / 【RX】 C011 ~ C018) で、NO (a 接点)、NC (b 接点) の選択が可能です。

出力端子変数 【MX2】 Y(00) ~ Y(02) / 【RX】 Y(00) ~ Y(05)

インバータの多機能出力に汎用出力 【MX2】 MO1 ~ MO3 / 【RX】 MO1 ~ MO6 を設定すると、DriveProgramming 機能の出力端子変数 【MX2】 Y(00) ~ Y(02) / 【RX】 Y(00) ~ Y(05) として使用することができます。

Yw 変数を使用すれば、出力端子変数 Y(00) ~ Y(05) を下位バイトとしたワードアクセスデータとして使用することもできます。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
【MX2】 Y(00) ~ Y(02)	出力端子変数 (ビットアクセス)	0: OFF 1: ON	0	—	ビット	R
【RX】 Y(00) ~ Y(05)						

【MX2】

- 多機能出力 P1、P2 端子、多機能リレー出力 (MA, MB) の状態をビット単位で変数として制御できます。
- 多機能出力 P1、P2 選択 (C021, C022)、または多機能リレー出力 (MA, MB) 選択 (C026) に「44 ~ 46: MO1 ~ MO3 (汎用出力)」を設定すると、出力端子 P1、P2 およびリレー出力 (MA, MB) を DriveProgramming 機能の出力端子変数 Y(00) ~ Y(02) として制御できます。出力端子変数の番号 Y(00) ~ Y(02) は、P1、P2、MA、MB の端子番号ではなく、設定した汎用出力 MO1 ~ 3 の番号順で設定されます。
- 3G3MX2 シリーズでは、汎用出力は MO1 ~ MO3 の最大 3 点です。

【RX】

- 多機能出力 P1 ~ P5 端子、多機能リレー出力 (MA, MB) の状態をビット単位で変数として制御できます。
- 多機能出力 P1 ~ P5 選択 (C021 ~ C025)、または多機能リレー出力 (MA, MB) 選択 (C026) に「44 ~ 49: MO1 ~ MO6 (汎用出力)」を設定すると、出力端子 P1 ~ P5 およびリレー出力 (MA, MB) を DriveProgramming 機能の出力端子変数 Y(00) ~ Y(05) として制御できます。出力端子変数の番号 Y(00) ~ Y(05) は、P1 ~ P5、MA、MB の端子番号ではなく、設定した汎用出力 MO1 ~ 6 の番号順で設定されます。

機能変数	多機能出力選択の設定 ([MX2] C021, C022, C026 / [RX] C021 ~ C026)
Y(00)	【MX2・RX 共通機能】 44: MO1
Y(01)	【MX2・RX 共通機能】 45: MO2
Y(02)	【MX2・RX 共通機能】 46: MO3
Y(03)	【RX だけの機能】 47: MO4
Y(04)	【RX だけの機能】 48: MO5
Y(05)	【RX だけの機能】 49: MO6

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
Yw	出力端子変数 (ワードアクセス)	0 ~ 255	0	—	符号無し 1 ワード	R

- 出力端子変数（ワードアクセス）Yw は、出力端子変数 【MX2】 Y(00) ~ Y(02) / 【RX】 Y(00) ~ Y(05) をワード単位の変数として使用できる機能です。
- 出力端子変数 Y(00) ~ Y(05) は下位バイトに配置されます。上位バイトや使用していない出力端子変数のデータはゼロが読み出され、データを書き込んだ場合はその設定を無視します。

機種	ビット	8 ~ 15	7	6	5	4	3	2	1	0
【MX2】	機能	未使用	未使用	未使用	未使用	未使用	未使用	Y(02)	Y(01)	Y(00)
【RX】	機能	未使用	未使用	未使用	Y(05)	Y(04)	Y(03)	Y(02)	Y(01)	Y(00)



安全上の要点

DriveProgramming 機能のプログラムが停止すると、DriveProgramming で制御している多機能出力はプログラム停止直前の状態を保持します。
このため、インバータの DriveProgramming 機能のプログラム停止を DriveProgramming 起動信号とアラーム（トリップ）信号で検出し、インバータ周辺機器を安全に停止させる配線を行ってください。



使用上の注意

- DriveProgramming 機能のプログラムが停止した場合、出力端子変数はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。
ただし、DriveProgramming 機能の汎用出力 MO1 ~ MO6 を設定していない出力は、プログラムとは無関係でインバータの多機能出力として制御されます。
- 複数のタスクで同じ多機能端子を使用すると、最後に実行されたタスクの出力状態が有効になります。制御の安定性から、複数のタスクからの制御は避けることを推奨します。
- 多機能出力端子（【MX2】 P1、P2 / 【RX】 P1 ~ P5）、または多機能リレー出力 (MA, MB) 選択で、MO1 ~ MO6（汎用出力 1 ~ 6）を選択した場合でも、多機能出力接点選択（【MX2】 C031、C032 / 【RX】 C031 ~ C035）、多機能リレー出力 (MA, MB) 接点選択 (C036) で、NO (a 接点)、NC (b 接点) の選択が可能です。

アナログ入力端子変数 XA(0) ～ XA(2)

インバータの周波数指令入力（アナログ電圧入力）FV 端子、周波数指令入力（アナログ電流入力）FI 端子を DriveProgramming 機能のアナログ入力端子変数 XA(0)、XA(1) として使用できます。3G3RX シリーズ V 1 タイプでは、周波数指令入力補助（アナログ電圧入力）FE 端子も DriveProgramming 機能のアナログ入力端子変数 XA(2) として使用できます。パラメータの設定にかかわらず、常にアナログ入力の状態をモニタすることができます。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
XA(0)	【MX2・RX 共通機能】 アナログ入力端子変数 (FV-FC 端子間: 0 ～ 10V 入力)	0 ～ 10,000	0	0.01%	符号無し 1 ワード	R
XA(1)	【MX2・RX 共通機能】 アナログ入力端子変数 (FI-FC 端子間: 4 ～ 20mA 入力)				符号付き 1 ワード	
XA(2)	【RX だけの機能】 アナログ入力端子変数 (FE-FC 端子間: -10V ～ 10V 入力)	-10,000 ～ 10,000			符号付き 1 ワード	

- アナログ入力端子変数 XA(0)、XA(1) は符号無し 1 ワード、XA(2) は符号付き 1 ワードの変数です。この変数は読出専用です。
- アナログ入力の最大入力 10V または 20mA の値を 100% として、0.01% 単位で表示されます。
- アナログ入力端子は、以下に示すようにインバータ本体の複数のパラメータで、インバータとしての機能を設定できます。各パラメータで機能を選択してください。
DriveProgramming 機能だけのアナログ入力として使用する場合は、以下のパラメータでアナログ入力端子を設定しないでください。
【MX2】 A001, A201, A005, A076, A079, A141, A142, b040, P033
【RX】 A001, A005, A006, A076, A079, A141, A142, b040, P033
- アナログ入力の調整は、インバータ本体のパラメータ（FV: A011 ～ A016、FI: A101 ～ A105、FE: A111 ～ A114）で行ってください。



使用上の注意

DriveProgramming 機能のプログラムが停止した場合、アナログ入力端子変数は状態を保持せず、常に入力端子の状態を更新します。

アナログ出力端子変数 YA(0) ～ YA(2)

インバータの多機能デジタル出力（PWM 出力）MP 端子、多機能アナログ出力（電圧出力）AM 端子を、DriveProgramming 機能のアナログ出力端子変数 YA(0) ～ YA(1) として使用できます。3G3RX シリーズ V 1 タイプでは、多機能アナログ出力（電流出力）AMI 端子も DriveProgramming 機能のアナログ出力端子変数 YA(2) として使用できます。パラメータの設定にかかわらず、常にアナログ出力の状態をモニタすることができます。DriveProgramming 機能からアナログ出力を制御する場合は、インバータ本体のパラメータ、MP 選択 (C027)、AM 選択 (C028)、AMI 選択 (C029) で DriveProgramming を設定してください。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
YA(0)	【MX2・RX 共通機能】 アナログ出力端子変数 (MP 端子: 6.4ms 周期 PWM 出力)	0 ~ 10,000	0	0.01%	符号無し 1 ワード	R/W
YA(1)	【MX2・RX 共通機能】 アナログ出力端子変数 (AM 端子: 0 ~ 10V 出 力)					
YA(2)	【RX だけの機能】 アナログ出力端子変数 (AMI 端子: 4 ~ 20mA 出力)					

- アナログ出力端子変数 YA(0) ~ YA(2) は、符号無し 1 ワードの変数です。
- アナログ出力の最大出力デューティ 100% または 10V、20mA の値を 100% として、0.01% 単位で設定します。
- インバータ本体のパラメータ、MP 選択 (C027)、AM 選択 (C028)、AMI 選択 (C029) で DriveProgramming を設定すると、DriveProgramming 機能からアナログ出力端子を制御できます。パラメータに DriveProgramming を設定していない場合でも、アナログ出力端子の状態はモニタすることができます。

機能変数	パラメータ設定
YA(00)	MP 選択 (C027) に「12: DriveProgramming (YA(0))」
YA(01)	AM 選択 (C028) に「13: DriveProgramming (YA(1))」
YA(02)	AMI 選択 (C029) に「14: DriveProgramming (YA(2))」

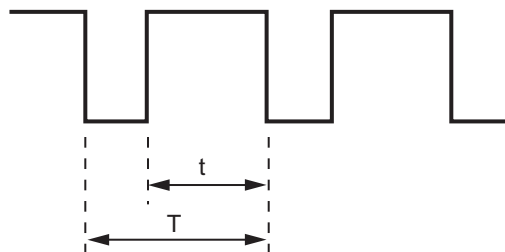
- アナログ出力の調整は、インバータ本体のパラメータ (C105 ~ C107、C109、C110)で行ってください。



使用上の注意

- DriveProgramming 機能のプログラムが停止した場合、アナログ出力端子変数はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。ただし、DriveProgramming を設定していない出力は、プログラムとは無関係でインバータのアナログ出力として制御されます。
- 多機能デジタル出力 (PWM 出力) MP 端子の出力は、PWM 信号で出力されます。6.4ms 周期のパルス幅 (デューティ t/T) を、0.00 ~ 100.00% で可変して出力します。

PWM 出力



周期 T: 一定 (6.4ms)
デューティ t/T : 変化

5-3 タイマ変数

DriveProgramming 機能のタイマ制御命令で使用するタイマ変数について説明します。

タイマ変数 TC(0) ～ TC(7)、TD(0) ～ TD(7)

DriveProgramming 機能のタイマ制御命令で使用するタイマカウンタ変数とタイマ出力接点です。タイマ制御命令の「delay on/off」命令、「timer set」命令でタイマ出力接点を指定すると、指定されたタイマ出力接点に対応するタイマカウンタ変数がカウントを開始します。指定したカウント値に達するとタイマ出力接点が出力されます。

タイマ制御命令で使用しない場合は、フリーランニングカウンタとして動作します。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
TC(0) ～ TC(7)	タイマカウンタ変数	0 ～ 2,147,483,647	0	10ms	符号無し 2ワード	R

- ・タイマカウンタ変数 TC(0) ～ TC(7) は、符号無し 2 ワードの変数です。この変数は読出専用です。
- ・タイマカウンタ変数 TC(0) ～ TC(7) は、それぞれタイマ出力接点 TD(0) ～ TD(7) に対応しています。
- ・タイマカウンタ変数 TC(0) ～ TC(7) は、プログラム起動と同時にゼロから 10ms 周期でカウントアップする 31 ビットのフリーランニングカウンタとして動作します。
- ・「delay on/off」命令、「timer set」命令でタイマ出力接点を指定されると、対応するタイマカウンタ変数 TC(0) ～ TC(7) がタイマ制御のカウンタとして動作します。
命令が実行されると、TC(0) ～ TC(7) はゼロにクリアされ、10ms 周期でカウントアップし、指定したカウント値に達するとカウントは停止します。
- ・「timer off」命令が実行された場合、対応するタイマカウンタ変数はゼロにクリアされ、10ms 周期でカウントアップする 31 ビットのフリーランニングカウンタとして動作を再開します。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
TD(0) ～ TD(7)	タイマ出力接点 (ビットアクセス)	0: OFF 1: ON	0	－	ビット	R

- ・タイマ出力接点 TD(0) ～ TD(7) は、ビット単位の変数です。この変数は読出専用です。
- ・プログラム起動時、タイマ出力接点 TD(0) ～ TD(7) はゼロにクリアされます。
- ・タイマ出力接点 TD(0) ～ TD(7) は、「delay on/off」命令、または「timer set」命令で指定されると動作を開始します。対応するタイマカウンタ変数がゼロにクリアされると同時に TD(0) ～ TD(7) に「0(OFF)」が設定されます。タイマカウンタ変数が指定したカウント値に達すると「1(ON)」に設定され、その状態を保持します。「timer off」命令が実行されると、TD(0) ～ TD(7) の状態はゼロにクリアされます。
- ・タイマカウンタ変数 TC(0) ～ TC(7) がフリーランニングカウンタとして動作している場合は、対応するタイマ出力接点 TD(0) ～ TD(7) の状態は変化せずに保持されます。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
TDw	タイマ出力接点 (ワードアクセス)	0 ～ 255	0	0.01Hz	符号無し 1ワード	R

- ・タイマ出力接点（ワードアクセス）TDw は、タイマ出力接点 TD(0) ～ TD(7) をワード単位の変数として使用できる機能です。この変数は読出専用です。
- ・タイマ出力接点 TD(0) ～ TD(7) は下位バイトに配置されます。上位バイトのデータはゼロが読み出されます。

ビット	8 ～ 15	7	6	5	4	3	2	1	0
機能	未使用	TD(7)	TD(6)	TD(5)	TD(4)	TD(3)	TD(2)	TD(1)	TD(0)

**使用上の注意**

DriveProgramming 機能のプログラムが停止した場合、タイマカウンタ変数とタイマ出力接点は状態を保持せず、ゼロにクリアします。

5-4 インバータ設定変数

インバータの周波数指令や加減速時間を設定する変数について説明します。
DriveProgramming 機能のプログラムからインバータを制御するために使用してください。

周波数指令変数 SET-Freq

DriveProgramming 機能から、周波数指令を直接制御する場合は、インバータ本体の周波数指令選択に「07: DriveProgramming」を設定してください。周波数指令変数 SET-Freq が有効になります。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
SET-Freq	周波数指令変数	0 ~ 40,000	0	0.01Hz	符号無し 1ワード	R/W

- 周波数指令変数 SET-Freq は、符号無し 1 ワードの変数です。
- 周波数指令選択に「07: DriveProgramming」を設定した場合だけ有効です。
 - 【MX2】 第 1 / 第 2 周波数指令選択 (A001/A201) に設定します。
多機能入力 S1 ~ S7 選択 (C001 ~ C007) に「08: SET (第 2 制御)」を割り付けて切り替えます。
 - 【RX】 周波数指令選択 (A001) に設定します。第 1 / 第 2 / 第 3 制御で切替できません。
- 電源投入時、周波数指令変数 SET-Freq はゼロにクリアされます。
EEPROM などでのデータ保存機能はありません。
- 開始データを設定したい場合は、開始データを SET-Freq に設定するプログラムを作成してください。
- 実際にインバータが出力できる周波数は、始動周波数 (b082) ~ 最高周波数です。
範囲外のデータを設定した場合は、以下のように動作します。
 - 始動周波数 (b082) 未満の設定
 - 【MX2・RX 共通機能】 0.00Hz として動作します。
 - 【RX だけの機能】 制御方式選択 (A044) が「04: 0SLV (0Hz 域センサレスベクトル)」、
「05: V2 (センサ付きベクトル制御)」を選択している場合は、始動周波数 (b082) は無効で、指令した周波数を出力します。
 - 最高周波数を超えた設定
 - 【MX2】 第 1 / 第 2 最高周波数 (A004/A204) に設定された周波数で周波数指令を制限します。
 - 【RX】 第 1 / 第 2 / 第 3 最高周波数 (A004/A204/A304) に設定された周波数で周波数指令を制限します。
- 周波数指令変数 SET-Freq で設定した周波数指令は、インバータ本体の出力周波数設定 / モニタ (F001) でもモニタすることができます。



使用上の注意

DriveProgramming 機能のプログラムが停止した場合、周波数指令変数はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。

加速・減速時間変数 ACCEL、DECEL

DriveProgramming 機能から周波数指令の加減速時間を直接制御する場合は、インバータ本体の加減速時間入力種別 (P031) に「03: DriveProgramming」を設定してください。

加速時間変数 ACCEL と減速時間変数 DECEL が有効になります。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
ACCEL	加速時間変数	【MX2】 0 ~ 360,000	パラメータ	0.01 秒	符号無し	R/W
DECEL	減速時間変数	【RX】 1 ~ 360,000	設定		2 ワード	

- 加速時間変数 ACCEL と減速時間変数 DECEL は、符号無し 2 ワードの変数です。
- 加減速時間入力種別 (P031) に「03: DriveProgramming」を設定した場合だけ加速時間変数 ACCEL と減速時間変数 DECEL が有効です。
- 電源投入したあとの初回プログラム起動時だけ、インバータ本体の加減速時間のパラメータ設定が DriveProgramming 機能の加速時間変数 ACCEL と減速時間変数 DECEL に設定されます。
 - 【MX2】 第 1 / 第 2 制御の選択に従って、第 1 / 第 2 加速時間 1 (F002/F202)、第 1 / 第 2 減速時間 1 (F003, F203) の加減速時間が設定されます。
 - 【RX】 第 1 / 第 2 / 第 3 制御の選択に従って、第 1 / 第 2 / 第 3 加速時間 1 (F002/F202/F302)、第 1 / 第 2 / 第 3 減速時間 1 (F003/F203/F303) の加減速時間が設定されます。
- 加速時間変数 ACCEL と減速時間変数 DECEL に設定されたデータは EEPROM に保存されません。
保存したい場合は、「eepwrt」命令を使用して、ユーザパラメータ変数またはインバータ本体の加減速時間パラメータの EEPROM に保存してください。
- 加減速時間の内部処理は 40ms 周期です。プログラムで加速時間変数 ACCEL と減速時間変数 DECEL を変更しても、反映するには最大 40ms の時間が必要です。
- プログラムで加速時間変数 ACCEL と減速時間変数 DECEL に、データ範囲外のデータや、3G3RX シリーズ V1 タイプで「0」を設定すると、その設定データは反映せず、それ以前に設定したデータで動作します。



使用上の注意

DriveProgramming 機能のプログラムが停止した場合、加速時間変数と減速時間変数はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。

5-5 インバータモニタ変数

インバータ内部のモニタ機能とステータスモニタを、DriveProgramming 機能で変数として使用できます。

各モニタ機能の詳細については、□□『多機能型小型インバータ形 3G3MX2 シリーズ V1 タイプ ユーザーズマニュアル (SBCE-376)』、□□『多機能型小型インバータ形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル (SBCE-506)』、または □□『高機能型汎用インバータ形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル (SBCE-367)』を参照してください。

ただし、データ単位はインバータ本体と異なる場合がありますので、必ず以下に示す単位で使用してください。

●【MX2・RX 共通機能】

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
FM	出力周波数モニタ (d001)	0 ~ 40,000	—	0.01Hz	符号無し 1 ワード	R

インバータの出力周波数をモニタすることができます。モニタ内容は出力周波数モニタ (d001) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
Iout	出力電流モニタ (d002)	0 ~ 9,999	—	0.1%	符号無し 1 ワード	R

インバータの出力電流をモニタすることができます。モニタ内容は出力電流モニタ (d002) に相当します。インバータの定格電流を 100% として、0.1% 単位で示します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
Dir	運転方向モニタ (d003)	0: 停止 1: 正転 2: 逆転	—	—	符号無し 1 ワード	R

インバータの運転方向をモニタすることができます。モニタ内容は運転方向モニタ (d003) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
PID-FB	PID フィードバック値モニタ (d004)	0 ~ 133,333	—	0.01%	符号無し 2 ワード	R

PID フィードバック値をモニタすることができます。モニタ内容は PID フィードバック値モニタ (d004) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
F-CNV	出力周波数モニタ (換算後) (d007)	0 ~ 3,999,600	—	0.01	符号無し 2 ワード	R

出力周波数モニタ (換算後) をモニタすることができます。モニタ内容は出力周波数モニタ (換算後) (d007) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
Tmon	出力トルクモニタ (d012)	-200 ~ 200	—	%	符号無し 1 ワード	R

出力トルクモニタをモニタすることができます。モニタ内容は、出力トルクモニタ (d012) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
Vout	出力電圧モニタ (d013)	0 ~ 6,000	-	0.1V	符号無し 1ワード	R

出力電圧をモニタすることができます。モニタ内容は、出力電圧モニタ (d013) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
Power	入力電力モニタ (d014)	0 ~ 9,999	-	0.1kW	符号無し 1ワード	R

入力電力をモニタすることができます。モニタ内容は、入力電力モニタ (d014) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
RUN-Time	RUN 中累積時間モニタ (d016)	0 ~ 999,999	-	時間	符号無し 2ワード	R

RUN 中累積時間モニタをモニタすることができます。モニタ内容は RUN 中累積時間モニタ (d016) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
ON-Time	電源 ON 累積時間モニタ (d017)	0 ~ 999,999	-	時間	符号無し 2ワード	R

電源 ON 時間がモニタすることができます。モニタ内容は電源 ON 時間モニタ (d017) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
UMon(0) ~ UMon(2)	ユーザモニタ 0 ~ 2 (d025 ~ d027)	-2,147,483,647 ~ 2,147,483,647	0	-	符号付き 2ワード	R/W

DriveProgramming 機能のユーザモニタ UMon(0) ~ UMon(2) は符号付き 2 ワードの変数です。DriveProgramming 機能のプログラムで、ユーザモニタ UMon(0) ~ UMon(2) にデータを設定すると、インバータ本体のユーザモニタ (d025、d026、d027) に表示することができます。プログラムの演算状態を外部に表示する場合に使用してください。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
POS	現在位置モニタ (d030)	-268,435,455 ~ 268,435,455 (-1,073,741,823 ~ 1,073,741,823) *1	-	1	符号付き 2ワード	R

現在位置をモニタすることができます。モニタ内容は現在位置モニタ (d030) に相当します。

*1.() 内の表示範囲は、【RX だけの機能】です。

V2 制御モード選択 (P012) に「03: 高分解能絶対位置制御」を設定した際の表示範囲です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
ERR-CNT	異常回数モニタ (d080)	0 ~ 65,535	-	回	符号無し 1ワード	R

インバータの異常累積回数がモニタすることができます。モニタ内容は、異常回数モニタ (d080) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
ERR(1)～ ERR(6)	異常モニタ 1～6 (d081 ～ d086)	0～127	－	－	符号無し 1ワード	R

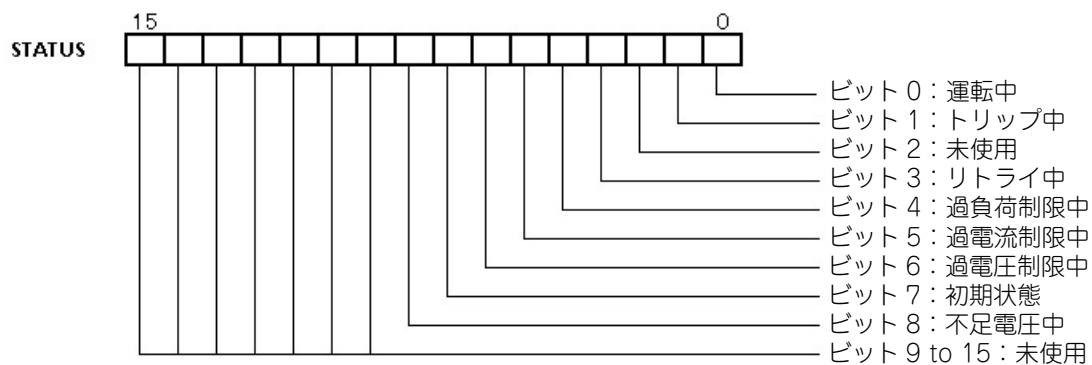
インバータの過去 6 回のトリップ内容をモニタすることができます。モニタ内容は異常モニタ 1～6 (d081～d086) に相当します。この変数は読出専用です。トリップ内容は、インバータ本体の Modbus 通信で使用している「インバータ異常モニタ要因一覧」と同じコードで示されます。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
DCV	直流電圧モニタ (d102)	0～10,000	－	0.1V	符号無し 1ワード	R

インバータの内部直流電圧をモニタすることができます。モニタ内容は、直流電圧モニタ (d102) に相当します。この変数は読出専用です。

機能変数	説明	データ範囲	初期データ	単位	データサイズ	R/W
STATUS	インバータステータスモ ニタ	－	－	－	符号無し 1ワード	R

インバータのステータス情報をモニタするために使用できます。
各ビットの内容は以下の内容に対応します。この変数は読出専用です。



5-6 多機能入力変数

多機能入力端子に設定できる機能を、DriveProgramming 機能のプログラムから実行することが可能です。以下に示す変数は多機能入力に設定できる機能にそれぞれ対応しています。

各変数に「1: ON」を設定すると、多機能入力が ON されたのと同じように各機能が動作します。

各変数に「0: OFF」を設定すると、その動作を停止します。多機能入力選択（【MX2】C001～C007 / 【RX】C001～C008）で各機能の設定をしなくても、プログラム上で実行できます。下記の表の参考に、インバータ本体の各機能設定データを示しました。各機能の詳細については、[□『多機能型小型インバータ形 3G3MX2 シリーズ V1 タイプ ユーザーズマニュアル \(SBCE-376\)』](#)、[□『多機能型小型インバータ形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル \(SBCE-506\)』](#)、または [□『高機能型汎用インバータ形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル \(SBCE-367\)』](#) で確認してください。

● 例

FW（正転）変数に「1」を設定すると、正転運転命令が実行されます。

FW: =1 インバータが正転運転を開始します。

FW: =0 インバータが正転運転を停止し、減速を開始します。



使用上の注意

- DriveProgramming 機能のプログラムが停止した場合、多機能入力変数は状態を保持せず、ゼロにクリアします。
- FW（正転）変数または RV（逆転）変数は、インバータ本体の運転指令選択（A002）で「01: 制御回路端子」を選択した場合だけ有効です。それ以外の設定では動作しません。
- インバータ電源投入直後、FW（正転）変数または RV（逆転）変数に「1」を設定してもその設定は無視し、正転運転または逆転運転を実行しません。いったん、「0」を設定し、再度「1」を設定してください。この動作を回避するためには、電源投入後だけ「wait」命令などを使用して 1 秒間の待ち時間をプログラムしてください。
- AT（アナログ入力切替）変数だけ、多機能入力選択（【MX2】C001～C007 / 【RX】C001～C008）での設定が必要です。AT（アナログ入力切替）変数を使用する場合は、多機能入力選択のいずれかに、「16: AT（アナログ入力切替）」を設定してください。設定していない場合は、プログラムで AT（アナログ入力切替）変数に「1」を設定しても動作しません。
- 多機能入力選択に設定したインバータ本体の多機能入力機能と多機能入力変数は OR の関係になります。

● 【MX2・RX 共通機能】

機能変数	説明	R/W	参考
FW	正転	R/W	【MX2】C001～C007 = 00 【RX】FW 端子
RV	逆転	R/W	C001～C008 = 01
CF1～CF4	多段速設定バイナリ 1～4	R/W	C001～C008 = 02～05
JG	ジョギング	R/W	C001～C008 = 06
DB	外部直流制動	R/W	C001～C008 = 07
SET	第 2 制御	R/W	C001～C008 = 08
TCH (2CH) ^{*1}	2 段加減速	R/W	C001～C008 = 09
FRS	フリーランストップ	R/W	C001～C008 = 11
EXT	外部トリップ	R/W	C001～C008 = 12
USP	復電再始動防止機能	R/W	C001～C008 = 13

機能変数	説明	R/W	参考
CS	商用切替	R/W	C001 ~ C008 = 14
SFT	ソフトロック	R/W	C001 ~ C008 = 15
AT	アナログ入力切替	R/W	C001 ~ C008 = 16
RS	リセット	R/W	C001 ~ C008 = 18
STA	3 ワイヤ起動	R/W	C001 ~ C008 = 20
STP	3 ワイヤ停止	R/W	C001 ~ C008 = 21
F-R (F/R)*1	3 ワイヤ正転／逆転	R/W	C001 ~ C008 = 22
PID	PID 無効	R/W	C001 ~ C008 = 23
PIDC	PID 積算リセット	R/W	C001 ~ C008 = 24
UP	遠隔操作加速	R/W	C001 ~ C008 = 27
DWN	遠隔操作減速	R/W	C001 ~ C008 = 28
UDC	遠隔データクリア	R/W	C001 ~ C008 = 29
OPE	強制オペレータ	R/W	C001 ~ C008 = 31
SF1 ~ SF7	多段速設定ビット 1 ~ 7	R/W	C001 ~ C008 = 32 ~ 38
OLR	過負荷制限切替	R/W	C001 ~ C008 = 39
TL	トルク制限有無	R/W	C001 ~ C008 = 40
TRQ1 ~ 2	トルクリミット切替 1 ~ 2	R/W	C001 ~ C008 = 41 ~ 42
BOK	ブレーキ確認	R/W	C001 ~ C008 = 44
LAC	LAD キャンセル	R/W	C001 ~ C008 = 46
PCLR	位置偏差クリア	R/W	C001 ~ C008 = 47
ADD	周波数 A145 加算	R/W	C001 ~ C008 = 50
F-TM	強制端子台	R/W	C001 ~ C008 = 51
ATR	トルク命令入力許可	R/W	C001 ~ C008 = 52
KHC	積算電力クリア	R/W	C001 ~ C008 = 53
AHD	アナログ命令保持	R/W	C001 ~ C008 = 65
CP1 ~ 3	位置命令選択 1 ~ 3	R/W	C001 ~ C008 = 66 ~ 68
ORL	原点復帰リミット信号	R/W	C001 ~ C008 = 69
ORG	原点復帰起動信号	R/W	C001 ~ C008 = 70
SPD	速度／位置切替え	R/W	C001 ~ C008 = 73

*1. 機能変数と多機能入力の選択機能短縮名が異なります。() の中に選択機能短縮名を示しました。

●【MX2 だけの機能】

機能変数	説明	R/W	参考
RS485 (485)*1	インバータ通信起動	R/W	C001 ~ C007 = 81
HLD	加減速停止	R/W	C001 ~ C007 = 83
ROK	運転許可信号	R/W	C001 ~ C007 = 84
DISP	表示固定	R/W	C001 ~ C007 = 86

*1. 機能変数と多機能入力の選択機能短縮名が異なります。() の中に選択機能短縮名を示しました。

● 【RX だけの機能】

機能変数	説明	R/W	参考
SET3	第 3 制御	R/W	C001 ~ C008 = 17
CAS	制御ゲイン切替	R/W	C001 ~ C008 = 26
PPI	P/PI 切り替え	R/W	C001 ~ C008 = 43
ORT	オリエンテーション	R/W	C001 ~ C008 = 45
STAT	パルス列位置命令入力許可	R/W	C001 ~ C008 = 48
SON	サーボ ON	R/W	C001 ~ C008 = 54
FOC	予備励磁	R/W	C001 ~ C008 = 55
FOT	正転駆動停止	R/W	C001 ~ C008 = 71
ROT	逆転駆動停止	R/W	C001 ~ C008 = 72
PCNT	パルスカウンタ	R/W	C001 ~ C008 = 74
PCC	パルスカウンタクリア	R/W	C001 ~ C008 = 75

5-7 多機能出力変数

多機能出力端子に設定できる機能を、DriveProgramming 機能のプログラムに使用することが可能です。以下に示す変数は、多機能出力に設定できる機能にそれぞれ対応しています。

各多機能出力機能が多機能出力を ON または OFF するのと同じように、各変数に「1: ON」または「0: OFF」を設定します。多機能出力選択（【MX2】 C021、C022 / 【RX】 C021 ~ C025）または多機能リレー出力（MA、MB）選択（C026）で各機能を設定しなくても、プログラム上でモニタできます。下記の表の参考に、インバータ本体の各機能設定データを示しました。各機能の詳細については、『多機能型小型インバータ形 3G3MX2 シリーズ V1 タイプ ユーザーズマニュアル (SBCE-376)』、『多機能型小型インバータ 形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル (SBCE-506)』、または 『高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル (SBCE-367)』を参照してください。



使用上の注意

DriveProgramming 機能のプログラムが停止した場合、多機能出力変数は状態を保持せず、常に各機能の状態を更新します。

● 【MX2・RX 共通機能】

機能変数	説明	R/W	参考
RUN	運転中信号	R	C021 ~ C026 = 00
FA1	定速到達信号	R	C021 ~ C026 = 01
FA2	設定周波数以上信号	R	C021 ~ C026 = 02
OL	過負荷予告	R	C021 ~ C026 = 03
OD	PID 偏差過大	R	C021 ~ C026 = 04
AL	アラーム出力	R	C021 ~ C026 = 05
FA3	設定周波数だけ信号	R	C021 ~ C026 = 06
OTQ	オーバアンダートルク信号	R	C021 ~ C026 = 07
UV	不足電圧中信号	R	C021 ~ C026 = 09
TRQ	トルク制限中	R	C021 ~ C026 = 10
RNT	RUN 時間オーバー	R	C021 ~ C026 = 11
ONT	電源 ON 時間オーバー	R	C021 ~ C026 = 12
THM	サーマル警告	R	C021 ~ C026 = 13
BRK	ブレーキ開放	R	C021 ~ C026 = 19
BER	ブレーキ異常	R	C021 ~ C026 = 20
ZS	0Hz 検出信号	R	C021 ~ C026 = 21
DSE	速度偏差過大	R	C021 ~ C026 = 22
POK	位置決め完了	R	C021 ~ C026 = 23
FA4	設定周波数以上信号 2	R	C021 ~ C026 = 24
FA5	設定周波数だけ信号 2	R	C021 ~ C026 = 25
OL2	過負荷予告 2	R	C021 ~ C026 = 26
ODc	アナログ FV 断線検出	R	C021 ~ C026 = 27
OIDc	アナログ FI 断線検出	R	C021 ~ C026 = 28
FBV	PID フィードバック比較信号	R	C021 ~ C026 = 31
NDc	通信断線検出	R	C021 ~ C026 = 32
LOG1	論理演算結果 1	R	C021 ~ C026 = 33
LOG 2	論理演算結果 2	R	C021 ~ C026 = 34
LOG 3	論理演算結果 3	R	C021 ~ C026 = 35

機能変数	説明	R/W	参考
WAC	コンデンサ寿命予告信号	R	C021 ~ C026 = 39
WAF	冷却ファン寿命予告信号	R	C021 ~ C026 = 40
FR	起動接点信号	R	C021 ~ C026 = 41
OHF	フィン過熱予告	R	C021 ~ C026 = 42
LOC	低電流信号	R	C021 ~ C026 = 43
IRDY	運転準備完了	R	C021 ~ C026 = 50
FWR	正転運転中信号	R	C021 ~ C026 = 51
RVR	逆転運転中信号	R	C021 ~ C026 = 52
MJA	重故障信号	R	C021 ~ C026 = 53
WCO	ウインドウコンパレータ FV	R	C021 ~ C026 = 54
WCOI	ウインドウコンパレータ FI	R	C021 ~ C026 = 55

● 【MX2 だけの機能】

機能変数	説明	R/W	参考
FREF	周波数指令オペレータ	R	C001 ~ C007 = 58
REF	運転指令オペレータ	R	C001 ~ C007 = 59
SETM	第 2 モータ選択中	R	C001 ~ C007 = 60
EDM	セーフティモニタ	R	C001 ~ C007 = 62

● 【RX だけの機能】

機能変数	説明	R/W	参考
IP	瞬停中信号	R	C021 ~ C026 = 08
O2Dc	アナログ FE 断線検出	R	C021 ~ C026 = 29
LOG 4	論理演算結果 4	R	C021 ~ C026 = 36
LOG 5	論理演算結果 5	R	C021 ~ C026 = 37
LOG 6	論理演算結果 6	R	C021 ~ C026 = 38
WCO2	ウインドウコンパレータ FE	R	C021 ~ C026 = 56

6

DriveProgramming 命令

この章では、DriveProgramming の命令について説明しています。

6-1 命令のカテゴリ	6-2
6-2 命令のフォーマット	6-3
6-3 命令一覧	6-4
6-4 プログラム制御命令	6-10
6-5 算術演算命令と論理演算命令	6-23
6-6 入出力制御命令	6-36
6-7 タイマ制御命令	6-46
6-8 パラメータ制御命令	6-52
6-9 インバータ制御命令	6-58

6-1 命令のカテゴリ

各命令は以下のカテゴリに分類されます。

- プログラム制御命令
- 算術演算命令と論理演算命令
- 入出力制御命令
- タイマ制御命令
- パラメータ制御命令
- インバータ制御命令

6-2 命令のフォーマット

各命令は、命令とその引数（0 ～最大 5 個）で構成されます。
以下に例を示します。

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
entry						プログラムの先頭を意味します

引数がない場合は、空欄です。

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
wait	<待機時間>					<待機時間> (×0.01) s 間待機します。

wait 命令の場合、引数は 1 つで、待機時間（変数または定数）を入力します。

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
while	<条件>					<条件>成立中<命令セット>を実行します。
	<命令セット>					<条件>成立中の命令セット。
wend						while にループして、終了します。

while 命令の場合、引数に条件を設定します。（詳細については、□「条件 (P.6-6)」を参照してください。）

while ～ wend の間に<条件>成立中に実行する命令セットを記述します。

6-3 命令一覧

プログラム制御命令

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
entry						タスクの始まりを示します。
end						タスクの終わりを示します。
call	<サブルーチン名>					<サブルーチン名>に分岐します。
sub	<サブルーチン名>					サブルーチンの始まりを示します。
end sub						サブルーチンの終わりを示します。
goto	<ラベル名>					無条件で<ラベル名>に分岐します。
on trip goto	<ラベル名>					トリップ発生時、<ラベル名>に分岐します。
if	<条件>			goto	<ラベル名>	条件成立時<ラベル名>に分岐します。
if	<条件>			then		構造 if を開始します。 <条件>成立時は、直下から else 命令までの<命令セット 1>を実行し、endif の次に移行します。 <条件>不成立時は、else 命令直下から endif 命令までの<命令セット 2>を実行し、endif 命令の次に移行します。
<命令セット 1>						<条件>成立時の命令セット。
else						<条件>不成立時の命令を開始します。
<命令セット 2>						<条件>不成立時の命令セット。
endif						構造 if を終了します。

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
select	<条件変数>					<条件変数>が<条件値>となった時 case 以下の命令を実行します。
case	<条件値 1 >					条件値の実行命令を開始します。
<命令セット 1 >						<条件値 1 >の命令セット。
[case 条件値 n] [<命令セット n >]						case による条件設定は複数回可能です。
case else						<条件値>以外の命令開始
<命令セット>						<条件値>以外の命令セット。
end select						select case 構文を終了します。
for	<変数>	<開始値>	<終了値>	<増分値>		<開始値>から<終了値>まで<刻み幅>でループします。
<命令セット>						繰り返し命令セット。
next						for にループして、終了します。
while	<条件>					<条件>成立中<命令セット>を実行します。
<命令セット>						<条件>成立中の命令セット。
wend						while にループして、終了します。
until	<条件>					<条件>成立まで<命令セット>を実行します。
<命令セット>						<条件>不成立中に実行する命令セット。
loop						until にループして、終了します。
wait	<待機時間>					<待機時間> (×0.01) s 間待機します。
wait	<条件>					<条件>成立まで待機します。



参考

<条件>のフォーマットについては、□「条件 (P.6-6)」を参照してください。

条件

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
条件	<変数 1 / 定数>	=	<変数 2 / 定数>			<変数 1 / 定数>と<変数 2 / 定数>が等しい時 TRUE
	<変数 1 / 定数>	<	<変数 2 / 定数>			<変数 1 / 定数>が<変数 2 / 定数>より小さい時 TRUE
	<変数 1 / 定数>	< =	<変数 2 / 定数>			<変数 1 / 定数>が<変数 2 / 定数>以下の時 TRUE
	<変数 1 / 定数>	>	<変数 2 / 定数>			<変数 1 / 定数>が<変数 2 / 定数>より大きい時 TRUE
	<変数 1 / 定数>	> =	<変数 2 / 定数>			<変数 1 / 定数>が<変数 2 / 定数>以上の時 TRUE
	<変数 1 / 定数>	< >	<変数 2 / 定数>			<変数 1 / 定数>と<変数 2 / 定数>が等しくない時 TRUE

算術演算命令と論理演算命令

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
算術演算	<変数 1>	: =	<変数 2 / 定数>			<変数 2 / 定数>を<変数 1>に代入します。
	<変数 1>	: =	<変数 2 / 定数>	+	<変数 3 / 定数>	<変数 2 / 定数>と<変数 3 / 定数>を加算して<変数 1>に代入します。
	<変数 1>	: =	<変数 2 / 定数>	-	<変数 3 / 定数>	<変数 2 / 定数>から<変数 3 / 定数>を減算して<変数 1>に代入します。
	<変数 1>	: =	<変数 2 / 定数>	*	<変数 3 / 定数>	<変数 2 / 定数>と<変数 3 / 定数>を乗算して<変数 1>に代入します。
	<変数 1>	: =	<変数 2 / 定数>	/	<変数 3 / 定数>	<変数 2 / 定数>を<変数 3 / 定数>で除算して<変数 1>に代入します。
除算した余り	<変数 1>	: =	<変数 2 / 定数>	mod	<変数 3 / 定数>	<変数 2 / 定数>を<変数 3 / 定数>で除算した余りを<変数 1>に代入します。
絶対値	<変数 1>	: = abs	<変数 2 / 定数>			<変数 2 / 定数>の絶対値を<変数 1>に代入します。

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
論理演算	<変数 1>	:=	<変数 2 / 定数>	and	<変数 3 / 定数>	<変数 2 / 定数> と <変数 3 / 定数> の論理積をとって <変数 1> に代入します。
	<変数 1>	:=	<変数 2 / 定数>	or	<変数 3 / 定数>	<変数 2 / 定数> と <変数 3 / 定数> の論理和をとって <変数 1> に代入します。
	<変数 1>	:=	<変数 2 / 定数>	xor	<変数 3 / 定数>	<変数 2 / 定数> と <変数 3 / 定数> の排他的論理和をとって <変数 1> に代入します。
	<変数 1>	:= not	<変数 2 / 定数>			<変数 2 / 定数> をビット反転して <変数 1> に代入します。

インクリメント命令デクリメント命令

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
インクリメント	inc	<変数>				<変数> の内容を 1 加算します。
デクリメント	Dec	<変数>				<変数> の内容を 1 減算します。

入出力制御命令

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
入力端子変数代入	<変数>	:=	X(i)			<変数> に入力端子変数の情報を代入します。 0=off 1=on
	<変数>	:=	Xw			<変数> に入力端子変数の情報をワード単位で代入します。
出力端子変数出力	Y(**)	:=	<変数 / 定数>			出力端子変数にビット単位で出力します。 0=off 1=on
	Yw	:=	<変数 / 定数>			出力端子変数にワード単位で出力します。
多機能入力変数出力	<多機能入力変数>	:=	<変数 / 定数>			多機能入力変数に出力します。 0=off 1=on
多機能出力変数代入	<変数>	:=	<多機能出力変数>			多機能出力変数の状態を <変数> に代入します。 0=off 1=on

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
内部ユーザ接点制御	<変数>	:=	UB(i)			<変数> に内部ユーザ接点情報を代入します。 0=off 1=on
	<変数>	:=	UBw			<変数> に内部ユーザ接点情報をワード単位で代入します。
	UB(**)	:=	<変数/定数>			内部ユーザ接点にビット単位で出力します。 0=off 1=on
	UBw	:=	<変数/定数>			内部ユーザ接点にワード単位で出力します。

■ タイマ制御命令

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
delay	on / off	<変数 1>	TD(i)	<変数 2 / 定数>		指定タイマに<変数 2 / 定数>を設定しスタートさせます。 <変数 2 / 定数>経過後に<変数 1>を ON または OFF させます。 同時に TD(i) を ON します。
timer set	TD(*)	<変数/定数>				指定タイマに<変数/定数>を設定しスタートさせます。 <変数/定数>経過後に TD(i) を ON します。
timer off	TD(*)					指定タイマを停止します。

■ パラメータ制御命令

命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
ChgParam	<パラメータ>	:=	<変数/定数>			<パラメータ>の内容を<変数/定数>に書換えます。
MonParam	<変数>	:=	<パラメータ>			<パラメータ>の内容を<変数>に代入します。
eepwrt						次に続くパラメータ書込命令実行時（1回だけ）に、インバータのEEPROMにも書き込む要求を発行します。
ChgParam	<パラメータ>	:=	<変数/定数>			<パラメータ>の内容を<変数/定数>に書換え、同時にEEPROMに保存します。
rtcset ^{*1}	オン/オフ	<変数>				LCD デジタルオペレータの時刻情報を<変数>に代入します。

*1. rtcset 命令は、3G3RX シリーズ V1 タイプだけの機能です。

3G3MX2 シリーズは、LCD デジタルオペレータ（形 3G3AX-OP05）との接続ができないため、rtcset 命令を使用できません。

インバータ制御命令

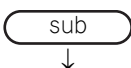
命令	引数 1	引数 2	引数 3	引数 4	引数 5	内容
FW	: =	1				インバータを正転します。
RV	: =	1				インバータを逆転します。
stop						インバータの運転を減速停止します。 ※ インバータがトリップ中はリセットとして動作します。
trip	<変数/定数>					<変数/定数>に対応したユーザトリップを発行します。
SET-Freq	: =	<変数/定数>				インバータの周波数指令変数に<変数/定数> (×0.01Hz) を設定します。
ACCEL	: =	<変数/定数>				インバータの加速時間変数に<変数/定数> (×10ms) を設定します。
DECEL	: =	<変数/定数>				インバータの減速時間変数に<変数/定数> (×10ms) を設定します。

6-4 プログラム制御命令

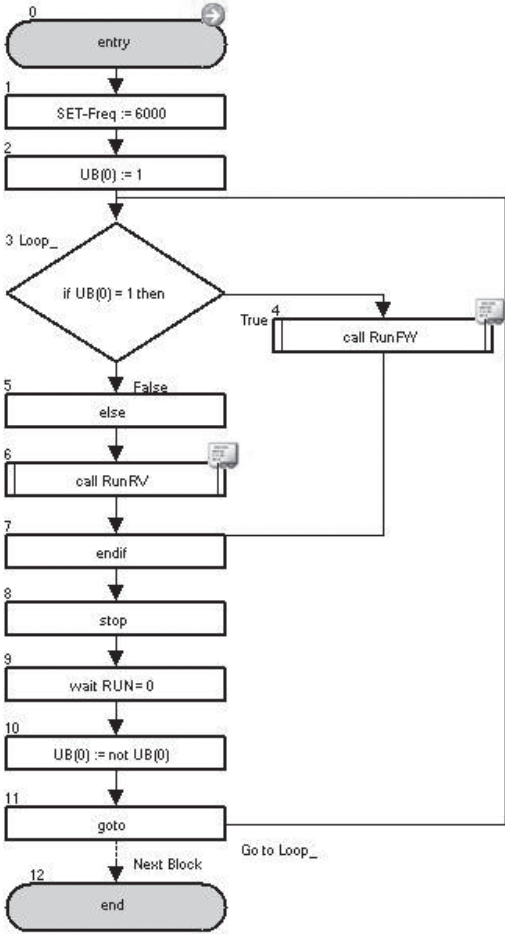
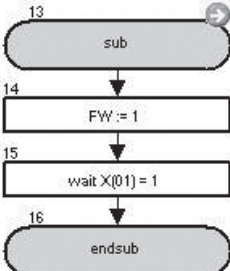
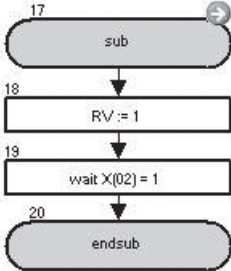
Entry		
命令	説明	引数
 Entry	タスクの始まりを示します。	---
フォーマット		
フローチャート方式	テキスト言語方式	
entry ↓	entry	
(注) 各タスクの先頭には、必ず必要です。		

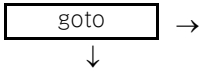
End		
命令	説明	引数
 End	タスクの終わりを示します。	---
フォーマット		
フローチャート方式	テキスト言語方式	
end ↓	end	
(注) 各タスクの末尾には、必ず必要です。		

Call		
命令	説明	引数
 Call	<サブルーチン名>に分岐します。	サブルーチン名 : サブルーチン名は、ユーザーが定義した名前またはエイリアスによって識別されます。
フォーマット		
フローチャート方式	テキスト言語方式	
call <サブルーチン名> ↓	call <サブルーチン名>	
(注) 1. サブルーチンの実行が終了すると、call 命令の次の命令行が実行されます。 2. フローチャート方式で call 命令にサブルーチン名を設定する場合は、命令を右クリックしてメニューから [サブルーチンをセット] を選択してください。中の選択オプションから選んで設定します。		

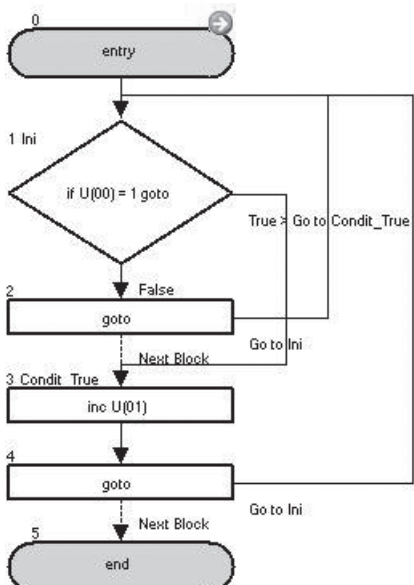
Sub		
命令	説明	引数
	サブルーチンの始まりを示します。	---
フォーマット		
フローチャート方式		テキスト言語方式
		sub
(注) 各サブルーチンの先頭には、必ず必要です。		
End Sub		
命令	説明	引数
	サブルーチンの終わりを示します。	---
フォーマット		
フローチャート方式		テキスト言語方式
		endsub
(注) 各サブルーチンの末尾には、必ず必要です。		

● 例

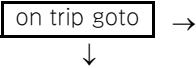
フローチャート		テキスト
Main		Main
		<pre>entry set-freq := 6000 UB(0) := 1 :Loop_ if UB(0) = 1 then call RunFW else call RunRV endif stop wait RUN = 0 UB(0) := not UB(0) goto Loop_ end</pre>
Main: RunFW		Main: RunFW
		<pre>sub FW := 1 wait X(01) = 1 endsub</pre>
Main: RunRV		Main: RunRV
		<pre>sub RV := 1 wait X(02) = 1 endsub</pre>
ブロック No.	動作	
1, 2	インバータの出力周波数を 60.00Hz にし、内部ユーザ接点 (UB(0)) を ON にします。	
3 ~ 7	UB(0) が ON のとき、サブルーチン (RunFW) を実行し、UB(0) が OFF のときサブルーチン (RunRV) を実行します。	
8 ~ 11	サブルーチンが終了すると、インバータを停止し、RUN が OFF するのを待ち、UB(0) を反転し、3 に戻ります。	
13 ~ 16	インバータを正転し、X(01) が ON するまでその wait 命令で待機します。(サブルーチン: RunFW)	
17 ~ 20	インバータを逆転し、X(02) が ON するまでその wait 命令で待機します。(サブルーチン: RunRV)	

Go To		
命令	説明	引数
	<ラベル名>へ無条件で分岐します。	ラベル名 : タスク内の特定の機能ブロックを識別するために使用される名前
フォーマット		
フローチャート方式		テキスト言語方式
		goto <ラベル名>
(注) 1. この命令も次に実行される命令に接続されている必要があります。これは、プログラムのフローを明確にするために必要です。 2. フローチャート方式でのラベル名のつけかたは、命令を右クリックしてメニューから「ラベルをセット」を選択してください。任意の名称を設定してください。		

● 例

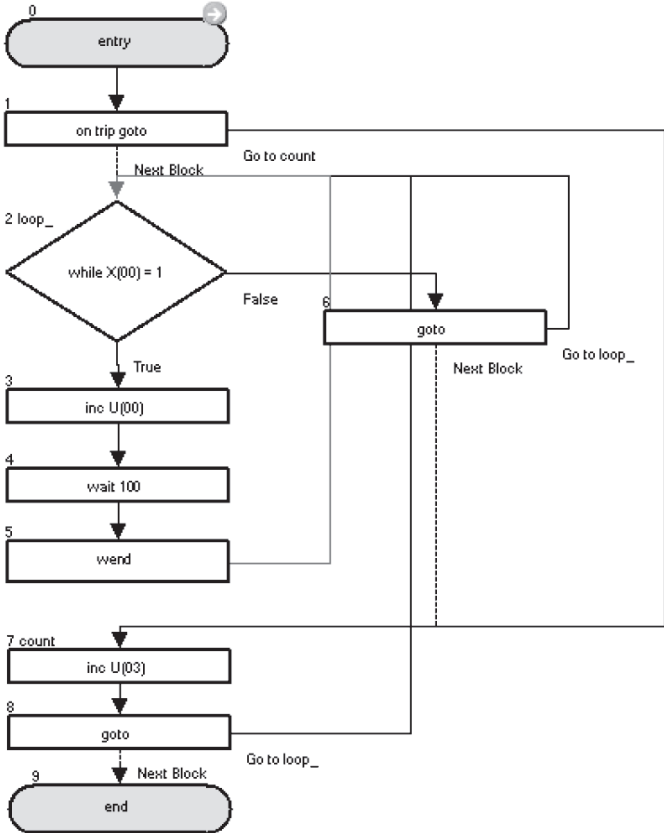
フローチャート	テキスト
	<pre> entry :Ini if U(00) = 1 goto Condit_True goto Ini :Condit_True inc U(01) goto Ini end </pre>

ブロック No.	動作
1	U(00) が 1 のとき 3: Condit_True に分岐し、U(00) が 0 のとき次の 2 に移行します。
2	無条件に 1: Ini に分岐します。
3	U(01) を +1 します。
4	無条件に 1: Ini に分岐します。

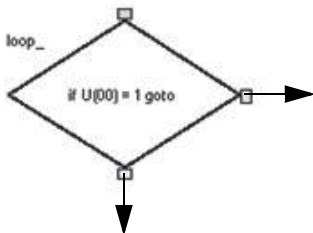
On Trip		
命令	説明	引数
	インバータでトリップ発生時、＜ラベル名＞に分岐します。	ラベル名 : タスク内の特定の機能ブロックを識別するために使用される名前
フォーマット		
フローチャート方式		テキスト言語方式
		on trip goto <ラベル名>

- (注) 1. on trip goto 命令は、インバータにトリップが発生すると分岐する機能です。
2. on trip goto 命令を 1 度実行すると、その内容はインバータ内部に記憶され、毎スキャントリップ状態を確認します。トリップが発生した次の処理で goto 先に分岐します。トリップが発生していなければ何も実行しません。
3. on trip goto 命令を実行しても、そのタスクの end 命令が実行されると on trip goto 命令は解除され、トリップが発生しても分岐されません。
4. on trip goto 命令の分岐が 1 度実行されると、どのような場合でも 2 度目の分岐は実行されません。on trip goto 命令の分岐が 1 度実行されたあとは、必ずプログラムを再起動してください。
5. フローチャート方式でのラベル名のつけかたは、命令を右クリックしてメニューから [ラベルをセット] を選択してください。任意の名称を設定してください。

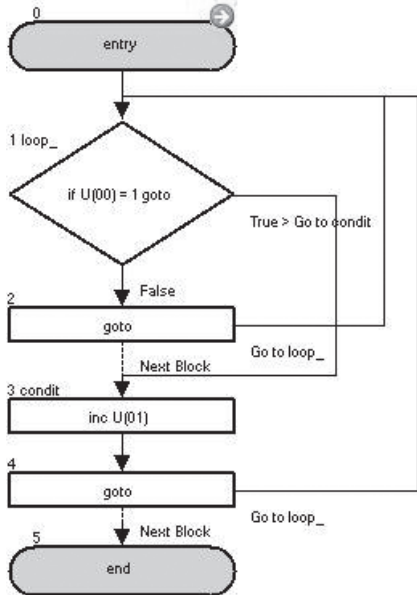
● 例

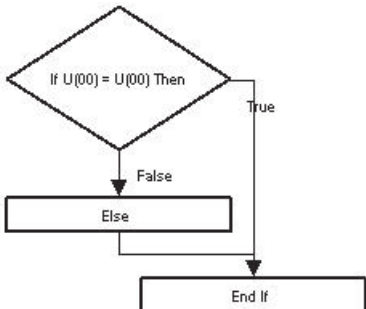
フローチャート	テキスト
	<pre>entry on trip goto count :loop_ while X(00) = 1 inc U(00) wait 100 wend goto loop_ :count inc U(03) goto loop_ end</pre>

ブロック No.	動作
1	on trip goto 命令を 1 回実行し、トリップ発生時に 7: count へ分岐するように設定します。
2～6	X(00) が 1 のとき、U(00) を +1 し、1.00 秒間 wait 命令で待機し、2: loop_ に戻ります。X(00) が 0 のとき無条件で 2: loop_ に分岐します。
7～8	U(03) を +1 し、無条件で 2: loop_ に分岐します。

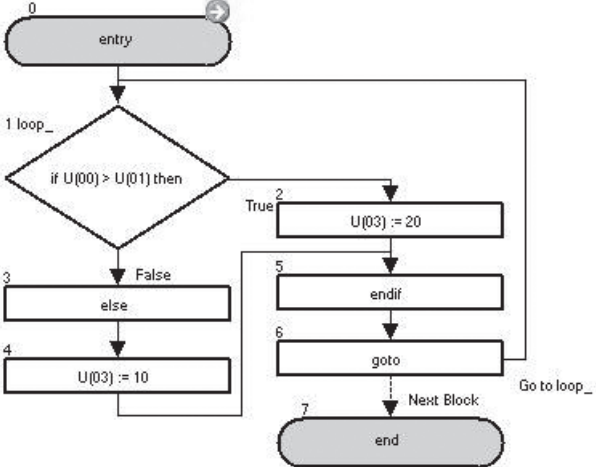
If		
命令	説明	引数
	<条件>が満たされていれば、<ラベル名>に分岐します。	条件: 「<左辺値><比較演算子><右辺値>」のフォーマットで、2つの、変数または定数を比較 ・ 左辺値: 任意の変数または定数 (-128 ~ 127 の範囲) ・ 比較演算子: =, <, >, <=, >=, <> ・ 右辺値: 任意の変数または定数 (-128 ~ 127 の範囲) ラベル名: タスク内の特定の機能ブロックを識別するために使用される名前
フォーマット		
フローチャート方式		テキスト言語方式
		if <条件> goto <ラベル名>
(注) フローチャート方式でのラベル名のつけかたは、命令を右クリックしてメニューから「ラベルをセット」を選択してください。任意の名称を設定してください。		

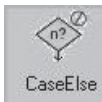
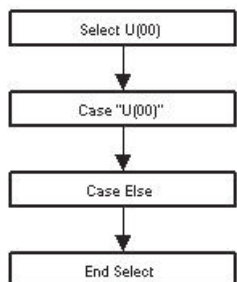
● 例

フローチャート		テキスト
		<pre> entry :loop_ if U(00) = 1 goto condit goto loop_ :condit inc U(01) goto loop_ end </pre>
ブロック No.	動作	
1	U(00) が 1 のとき、3: condit に分岐し、U(00) が 1 以外のとき次の 2 に移行します。	
2	無条件で 1: loop_ に分岐します。	
3 ~ 4	U(01) を +1 し、無条件で 1: loop_ に分岐します。	

Ifs/Else/End If		
命令	説明	引数
 Ifs	<条件>成立時は、直下から else 命令までの<命令セット 1>を実行し、endif 命令の次に移行します。 <条件>不成立時は、else 命令直下から endif 命令までの<命令セット 2>を実行し、endif 命令の次に移行します。	条件: 「<左辺値><比較演算子><右辺値>」のフォーマットで、2 つの、変数または定数を比較 • 左辺値: 任意の変数または定数 (−128 ~ 127 の範囲) • 比較演算子: =, <, >, <=, >=, <> • 右辺値: 任意の変数または定数 (−128 ~ 127 の範囲) 命令セット 1: else 命令までの 1 つ以上の命令。ネストされた命令 (最大 8 階層) を含むことができます。 命令セット 2: endif 命令までの 1 つ以上の命令。ネストされた命令 (最大 8 階層) を含むことができます。
 Ifs ...		
 Else		
 EndIf		
フォーマット		
フローチャート方式		テキスト言語方式
		<pre>if <条件> then <命令セット 1> else <命令セット 2> endif</pre>

● 例

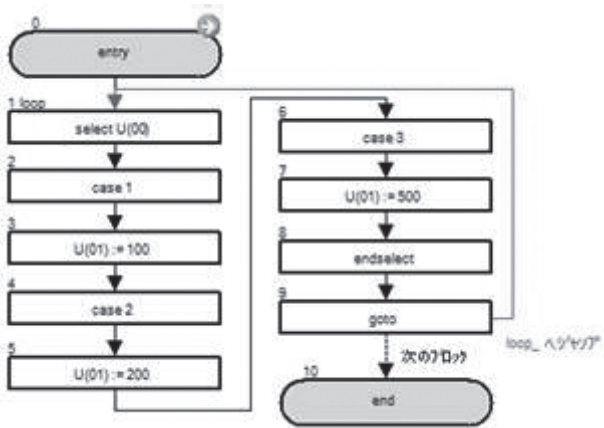
フローチャート	テキスト
	<pre>entry :loop_ if U(00) > U(01) then U(03) := 20 endif goto loop_ else U(03) := 10 endif goto loop_ end</pre>
ブロック No.	動作
1 ~ 5	U(00) > U(01) のとき、直下の U(03) に 20 を代入し、endif 命令に分岐します。U(00) > U(01) 以外の場合、else 命令直下の U(03) に 10 を代入し、endif 命令に移行します。
6	無条件で 1: loop_ に分岐します。

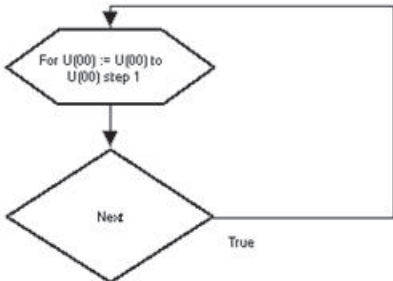
Select/Case/Case Else/EndSelect		
命令	説明	引数
 Select	変数の値に基づいて、複数のプログラムセクションを実行できるようにします。特定の case 命令の<条件変数>が<条件値 n>と一致していれば、<命令セット n>を実行します。 <条件変数>が case 命令のいずれの条件値とも一致していない場合は、<条件に一致しない場合の命令セット> (case else) を実行します。 この命令は、1 つの<条件変数>によって複数の選択を行う必要がある場合に適しています。この命令は、サブルーチンの call 命令と組み合わせて使用することによりプログラムを整理する場合にも適しています。	条件変数：使用する条件変数 条件値 n: 任意の変数または定数（-128 ～ 127 の範囲） 命令セット n: 次の case または endselect 命令までの 1 つ以上の命令。ネストされた命令（最大 8 階層）を含むことができます。
 Case		
 Case Else		
 EndSelect		
フォーマット		
フローチャート方式	テキスト言語方式	
	<pre>select <条件変数> case <条件値 1> <命令セット 1> ... case <条件値 n> <命令セット n> ... case else <条件に一致しない場合の命令セット> endselect</pre>	

(注) select は 1 ネストと数えます。

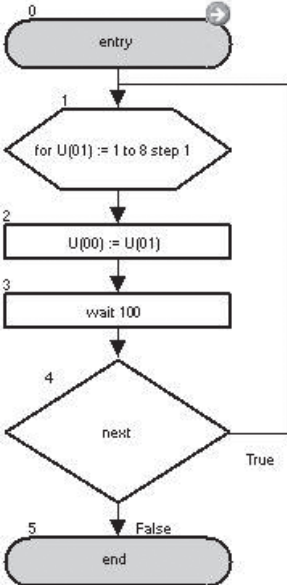
サブルーチン内での select ~ endselect を使用した場合は 2 ネストとなります。

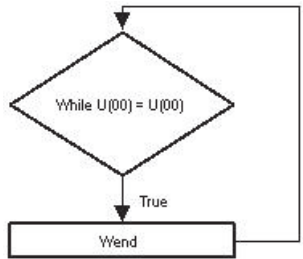
● 例

フローチャート		テキスト
		<pre> :loop_ select U(00) case 1 U(01) := 100 case 2 U(01) := 200 case 3 U(01) := 500 endselect goto loop_ end </pre>
ブロック No.	動作	
1 ~ 8	U(00) の値に基づいて、U(00) が、1 のとき U(01) に 100 を代入し、2 のとき U(01) に 200 を代入し、3 のとき U(01) に 500 を代入し、endselect 命令に分岐します。	
9	無条件に 1: loop_ に分岐します。	

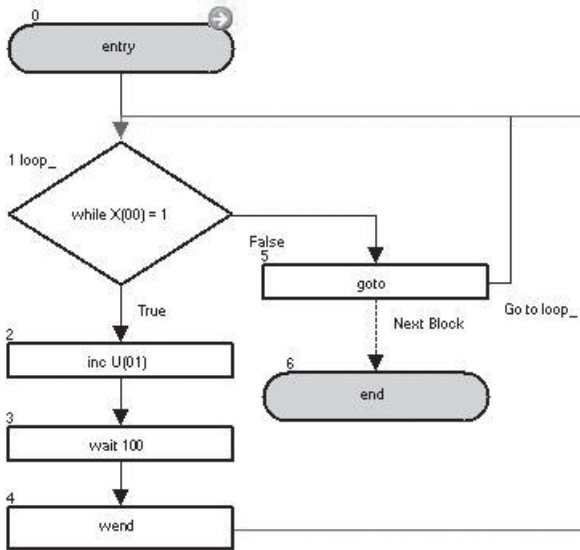
For/Next		
命令	説明	引数
 For ...	<命令セット>を実行し、<変数>を<開始値>から<増分値>ずつ増加し、<終了値>に達するまで、<命令セット>を繰り返し実行します。next 命令実行時に、<終了値>の判定を実施し、<変数>に<増分値>が加算されます。	変数：任意の変数 開始値：初期値。最初のループの変数に割り当てられている値です。 定数（-128 ～ 127 の範囲） 終了値：ループを終了する値 定数（-128 ～ 127 の範囲） 増分値：変数は、ループごとに、この値ずつ増加します。 定数（-128 ～ 127 の範囲） 命令セット：next 命令までの 1 つ以上の命令。ネストされた命令（最大 8 階層）を含むことができます。
 For		
 Next		
フォーマット		
フローチャート方式		テキスト言語方式
		<pre>for <変数> := <開始値> to <終了値> step <増分値> <命令セット> next</pre>

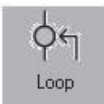
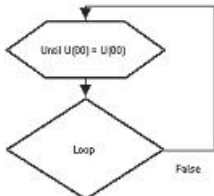
● 例

フローチャート		テキスト
		<pre>entry for U(01) := 1 to 8 step 1 U(00) := U(01) wait 100 next end</pre>
ブロック No.	動作	
1 ～ 4	U(01) に 1 を代入し、U(00) に U(01) の値を代入し、1.00 秒間 wait 命令で待機し、U(01) が 8 未満のときは U(01) を +1 して for 命令の次の命令を実行します。U(01) が 8 以上のとき next 命令の次に移行します。	

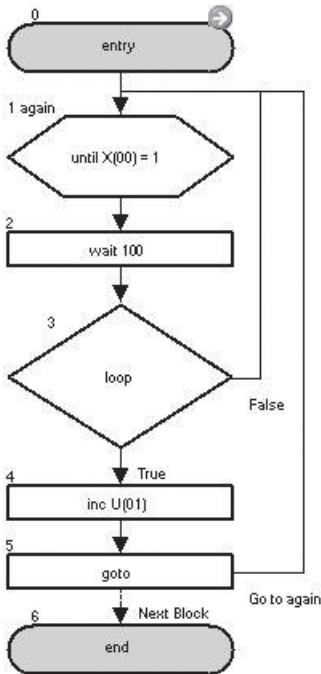
While/Wend		
命令	説明	引数
 While ...  While  Wend	条件が満たされている間、＜命令セット＞を実行します。	条件: 「＜左辺値＞＜比較演算子＞＜右辺値＞」のフォーマットで、2つの、変数または定数を比較 - 左辺値: 任意の変数または定数 (−128 ~ 127 の範囲) - 比較演算子: =, <, >, <=, >=, <> - 右辺値: 任意の変数または定数 (−128 ~ 127 の範囲) 命令セット: wend 命令までの 1 つ以上の命令。ネストされた命令 (最大 8 階層) を含むことができます。
フォーマット		
フローチャート方式		テキスト言語方式
		<pre>while <条件> <命令セット> wend</pre>

● 例

フローチャート		テキスト
		<pre>entry :loop_ while X(00) = 1 inc U(01) wait 100 wend goto loop_ end</pre>
ブロック No.	動作	
1 ~ 4	X(00) が 1 のとき、U(01) を +1 し、1.00 秒間 wait 命令で待機し、while 命令に分岐します。X(00) が 1 以外のとき、wend 命令の次の命令を実行します。	
5	無条件で 1: loop_ に分岐します。	

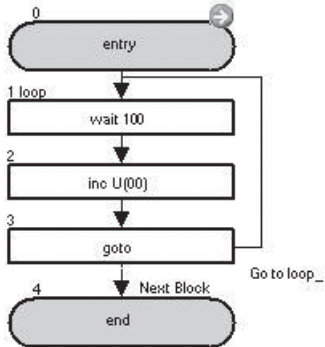
Until/Loop		
命令	説明	引数
 Until ...  Until  Loop	<条件>が満たされるまで、<命令セット>を実行します。	条件：「<左辺値><比較演算子><右辺値>」のフォーマットで、2つの、変数または定数を比較 • 左辺値：任意の変数または定数（-128 ～ 127 の範囲） • 比較演算子：=, <, >, <=, >=, <> • 右辺値：任意の変数または定数（-128 ～ 127 の範囲） 命令セット：loop 命令までの 1 つ以上の命令。ネストされた命令（最大 8 階層）を含むことができます。
フォーマット		
フローチャート方式		テキスト言語方式
		until <条件> <命令セット> loop

● 例

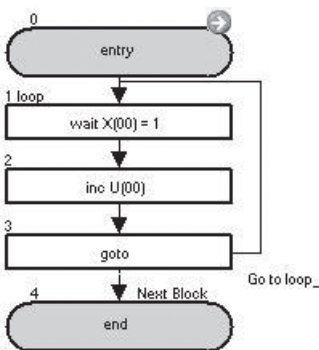
フローチャート	テキスト
	<pre> entry :again until X(00) = 1 wait 100 loop inc U(01) goto again end </pre>
ブロック No.	動作
1 ~ 3	X(00) が 1 以外るとき、1.00 秒間 wait 命令で待機し、until 命令に分岐します。X(00) が 1 のとき、loop 命令まで実行後、次の命令に移行します。
4 ~ 5	U(00) を +1 し、無条件で 1: again に分岐します。

Wait		
命令	説明	引数
 WaitTime  WaitCond	一定の秒数、または条件が満たされるまで、プログラムを待機させます。	値: 任意の変数または定数 (10ms で指定された時間) 待ち時間値 (0 ~ 32767×10ms) 条件: 「<左辺値><比較演算子><右辺値>」のフォーマットで、2 つの、変数または定数を比較 - 左辺値: 任意の変数または定数 (0 ~ 127 の範囲) - 比較演算子: =, <, >, <=, >=, <> . - 右辺値: 任意の変数または定数 (0 ~ 127 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
wait <値>または<条件> ↓		wait <値>または<条件>
(注) 1. wait 命令は、設定された待ち時間または設定された条件が成立するまで、wait 命令で待機し、次の命令に移行しません。このため、wait 命令の待機時間中、そのタスクの処理は実行されません。定期的なサイクルで監視を行わないといけない処理は、別タスクに分けて、wait 命令を使用せず定期的なサイクルで処理されるようにプログラムを構成してください。 2. wait 命令で使用する定数に負の値を設定しないでください。 待ち時間や条件成立を判断できず、プログラムが wait 命令で停止したまま動作しません。 3. wait 命令は、時間を測定するための正確な手段ではありません。正確な時間測定を行うには、内部タイマまたは LCD デジタルオペレータの時刻機能を使用してください。		

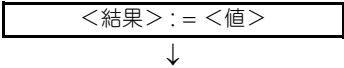
- <値> で待ち時間を設定した例：設定された<値>の時間、この命令で待機します。

フローチャート		テキスト
		<pre> entry :loop_ wait 100 inc U(00) goto loop_ end </pre>
ブロック No.	動作	
1	1.00 秒間 wait 命令で待機します。	
2 ~ 3	U(00) を +1 し、再度 1 秒間待機するため、無条件で 1: loop_ に分岐します。	

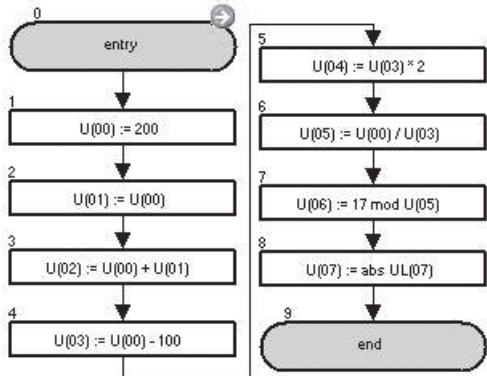
- <条件>で待機解除条件を設定した例：条件が満たされるまで待機します。

フローチャート		テキスト
		<pre>entry :loop_ wait X(00) = 1 inc U(00) goto loop_ end</pre>
ブロック No.	動作	
1	X(00) が 1 になるまで待機します。	
2 ~ 3	U(00) を +1 し、無条件で 1: loop_ に分岐します。	

6-5 算術演算命令と論理演算命令

= (代入)		
命令	説明	引数
	<値>を<結果>に代入します。	結果：任意の変数 値：任意の変数または定数 (-2,147,483,648 ~ 2,147,483,647 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
		<結果> := <値>
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。オーバーフローやアンダーフローが発生しないように、アプリケーションで処理する必要があります。		

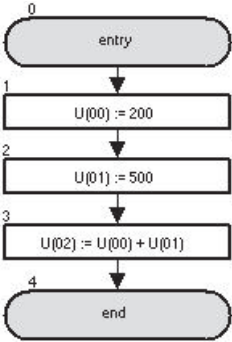
● 例

フローチャート	テキスト
	<pre> entry U(00) := 200 U(01) := U(00) U(02) := U(00) + U(01) U(03) := U(00) - 100 U(04) := U(03) * 2 U(05) := U(00) / U(03) U(06) := 17 mod U(05) U(07) := abs UL(07) end </pre>

ブロック No.	動作
1	U(00) に 200 を代入します。
2	U(01) に U(00) の値を代入します。
3	U(02) に U(00) と U(01) の和を代入します。
4	U(03) に U(00) と 100 の差を代入します。
5	U(04) に U(03) と 2 の積を代入します。
6	U(05) に U(00) と U(03) の商を代入します。
7	U(06) に 17 割る U(05) の余りを代入します。
8	U(07) に UL(07) の絶対値を代入します。

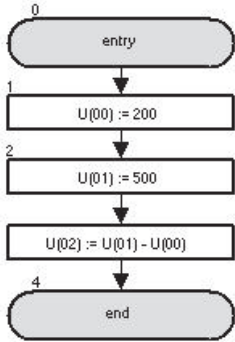
+（加算）		
命令	説明	引数
	<値 1 >と<値 2 >を加算します。	結果 ：任意の変数 値 1 ：任意の変数または定数（-128 ～ 127 の範囲） 値 2 ：任意の変数または定数（-2,147,483,648 ～ 2,147,483,647 の範囲）
フォーマット		
フローチャート方式		テキスト言語方式
<結果> := <値 1> + <値 2> ↓		<結果> := <値 1> + <値 2>
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。オーバーフローやアンダーフローが発生しないように、アプリケーションで処理する必要があります。		

● 例

フローチャート		テキスト
		<pre>entry U(00) := 200 U(01) := 500 U(02) := U(00) + U(01) end</pre>
ブロック No.	動作	
1	U(00) に 200 を代入します。	
2	U(01) に 500 を代入します。	
3	U(02) に U(00) と U(01) の和を代入します。	

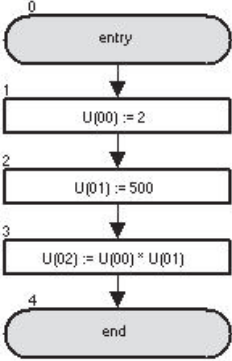
－（減算）		
命令	説明	引数
	<値 1 > から <値 2 > を減算します。	結果 ： 任意の変数 値 1 ： 任意の変数または定数（-128 ～ 127 の範囲） 値 2 ： 任意の変数または定数（-2,147,483,648 ～ 2,147,483,647 の範囲）
フォーマット		
フローチャート方式		テキスト言語方式
<結果> := <値 1> - <値 2> ↓		<結果> := <値 1> - <値 2>
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。オーバーフローやアンダーフローが発生しないように、アプリケーションで処理する必要があります。		

● 例

フローチャート		テキスト
		<pre>entry U(00) := 200 U(01) := 500 U(02) := U(01) - U(00) end</pre>
ブロック No.	動作	
1	U(00) に 200 を代入します。	
2	U(01) に 500 を代入します。	
3	U(02) に U(01) と U(00) の差を代入します。	

* (乗算)		
命令	説明	引数
	<値 1 >に<値 2 >を乗算します。	結果 : 任意の変数 値 1 : 任意の変数または定数 (-128 ~ 127 の範囲) 値 2 : 任意の変数または定数 (-2,147,483,648 ~ 2,147,483,647 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
<結果> := <値 1> * <値 2> ↓		<結果> := <値 1> * <値 2>
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。オーバーフローやアンダーフローが発生しないように、アプリケーションで処理する必要があります。		

● 例

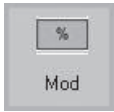
フローチャート		テキスト
		<pre>entry U(00) := 2 U(01) := 500 U(02) := U(00) * U(01) end</pre>
ブロック No.	動作	
1	U(00) に 2 を代入します。	
2	U(01) に 500 を代入します。	
3	U(02) に U(00) と U(01) の積を代入します。	

／（除算）		
命令	説明	引数
	<値 1> を <値 2> で除算します。	結果 : 任意の変数 値 1 : 任意の変数または定数 (−128 ～ 127 の範囲) 値 2 : 任意の変数または定数 (−2,147,483,648 ～ 2,147,483,647 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
<結果> := <値 1> / <値 2> ↓		<結果> := <値 1> / <値 2>
(注) DriveProgramming は、オーバーフローやアンダーフローまたは「0」による割り算が発生した場合、エラーとします。発生しないように、アプリケーションで処理する必要があります。		

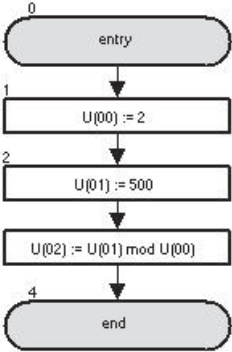
● 例

フローチャート	テキスト
<pre> graph TD 0([entry]) --> 1[U(00) := 2] 1 --> 2[U(01) := 500] 2 --> 3[U(02) := U(01) / U(00)] 3 --> 4([end]) </pre>	<pre> entry U(00) := 2 U(01) := 500 U(02) := U(01) / U(00) end </pre>

ブロック No.	動作
1	U(00) に 2 を代入します。
2	U(01) に 500 を代入します。
3	U(02) に U(01) と U(00) の商を代入します。

Mod（剰余）		
命令	説明	引数
	<値 1 > を <値 2 > で除算した余り	結果 : 任意の変数 値 1 : 任意の変数または定数（-128 ～ 127 の範囲） 値 2 : 任意の変数または定数（-2,147,483,648 ～ 2,147,483,647 の範囲）
フォーマット		
フローチャート方式		テキスト言語方式
<結果> := <値 1> mod <値 2> ↓		<結果> := <値 1> mod <値 2>
(注) DriveProgramming は、オーバーフローやアンダーフローまたは「0」による割り算が発生した場合、エラーとします。発生しないように、アプリケーションで処理する必要があります。		

● 例

フローチャート		テキスト
		<pre>entry U(00) := 2 U(01) := 500 U(02) := U(01) mod U(00) end</pre>
ブロック No.	動作	
1	U(00) に 2 を代入します。	
2	U(01) に 500 を代入します。	
3	U(02) に U(01) と U(00) の除算の余りを代入します。	

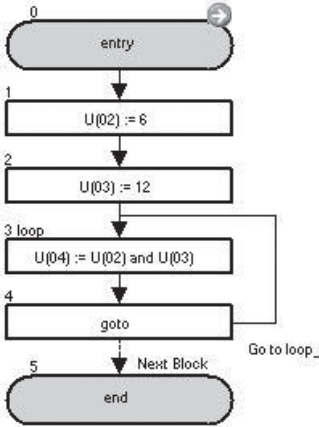
Abs（絶対値）		
命令	説明	引数
	<値>の絶対値をとります	結果 : 任意の変数 値 : 任意の変数または定数 (-2,147,483,647 ~ 2,147,483,647 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
<結果> := abs <値> ↓		<結果> := abs <値>
(注) 1. DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。発生しないように、アプリケーションで処理する必要があります。 2. 変数または定数が負の最大値 -2,147,483,648 の場合、絶対値に変換できません。-2,147,483,648 をそのままの負の値として認識します。		

● 例

フローチャート		テキスト
<pre> graph TD 0([entry]) --> 1[UL[00] := -200] 1 --> 2[UL[01] := abs UL[00]] 2 --> 4([end]) </pre>		<pre> entry UL(00) := -200 UL(01) := abs UL(00) end </pre>
ブロック No.	動作	
1	UL(00) に -200 を代入します。	
2	UL(01) に UL(00) の絶対値を代入します。	

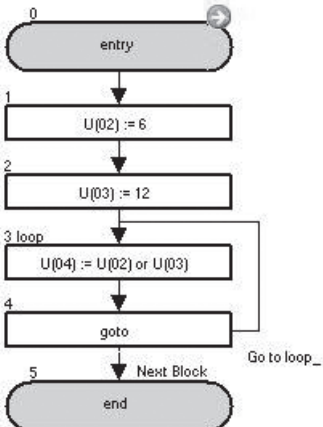
And（論理積）				
命令	説明		引数	
	<値 1> と <値 2> の 2 進数での論理積 and（論理積）		結果 ： 任意の変数 値 1 ： 任意の変数または定数 （-128 ～ 127 の範囲） 値 2 ： 任意の変数または定数 （-2,147,483,648 ～ 2,147,483,647 の範囲）	
				値 1
	0	0		0
	0	1		0
	1	0		0
	1	1		1
フォーマット				
フローチャート方式		テキスト言語方式		
<結果> := <値 1> and <値 2> ↓		<結果> := <値 1> and <値 2>		
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。発生しないように、アプリケーションで処理する必要があります。				

● 例

フローチャート		テキスト
		<pre>entry U(02) := 6 U(03) := 12 :loop_ U(04) := U(02) and U(03) goto loop_ end</pre>
ブロック No.	動作	
1	U(02) に 6 を代入します。	
2	U(03) に 12 を代入します。	
3	U(04) に U(02) の 6（2 進数：00000110）と U(03) の 12（2 進数：00001100）の論理積 4（2 進数：00000100）を代入します。	
4	無条件で 3: loop_ に分岐します。	

Or（論理和）			
命令	説明		引数
	<値 1>と<値 2>の2進数での論理和 or（論理和）		結果：任意の変数
			値 1: 任意の変数または定数 (-128 ～ 127 の範囲)
			値 2: 任意の変数または定数 (-2,147,483,648 ～ 2,147,483,647 の範囲)
	値 1	値 2	結果
	0	0	0
	0	1	1
1	0	1	
1	1	1	
フォーマット			
フローチャート方式		テキスト言語方式	
<結果> := <値 1> or <値 2> ↓		<結果> := <値 1> or <値 2>	
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。発生しないように、アプリケーションで処理する必要があります。			

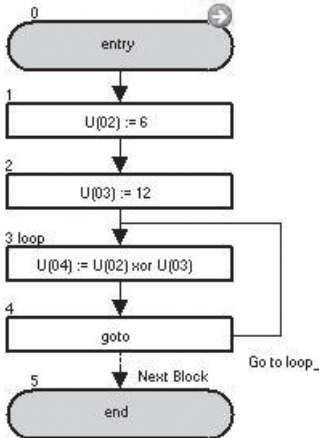
● 例

フローチャート	テキスト
	<pre> entry U(02) := 6 U(03) := 12 :loop_ U(04) := U(02) or U(03) goto loop_ end </pre>

ブロック No.	動作
1	U(02) に 6 を代入します。
2	U(03) に 12 を代入します。
3	U(04) に U(02) の 6 (2進数: 00000110) と U(03) の 12 (2進数: 00001100) の論理和 14 (2進数: 00001110) を代入します。
4	無条件で 3: loop_ に分岐します。

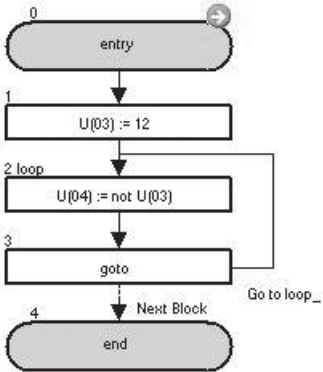
XOr（排他的論理和）																
命令	説明	引数														
	<値 1>と<値 2>の 2 進数での排他的論理和 xor（排他的論理和）	結果 ： 任意の変数 値 1 ： 任意の変数または定数 （-128 ～ 127 の範囲） 値 2 ： 任意の変数または定数 （-2,147,483,648 ～ 2,147,483,647 の範囲）														
	<table border="1"> <thead> <tr> <th>値 1</th><th>値 2</th><th>結果</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	値 1	値 2	結果	0	0	0	0	1	1	1	0	1	1	1	0
値 1	値 2	結果														
0	0	0														
0	1	1														
1	0	1														
1	1	0														
フォーマット																
フローチャート方式		テキスト言語方式														
<結果> := <値 1> xor <値 2> ↓		<結果> := <値 1> xor <値 2>														
（注） DriveProgramming は、 オーバーフローやアンダーフローが発生した場合、 エラーとします。 発生しないように、 アプリケーションで処理する必要があります。																

● 例

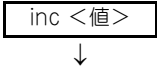
フローチャート		テキスト
		<pre> entry U(02) := 6 U(03) := 12 :loop_ U(04) := U(02) xor U(03) goto loop_ end </pre>
ブロック No.	動作	
1	U(02) に 6 を代入します。	
2	U(03) に 12 を代入します。	
3	U(04) に U(02) の 6（2 進数：00000110）と U(03) の 12（2 進数：00001100）の排他的論理和 10（2 進数：00001010）を代入します。	
4	無条件で 3: loop_ に分岐します。	

Not (2 進数での否定)								
命令	説明	引数						
	<値 1 >の 2 進数での否定 (ビット反転) not (否定)	結果 : ビットデータサイズの変数を除く任意の変数 *1 値 : ビットデータサイズの変数を除く任意の変数または定数 *1 (-2,147,483,648 ~ 2,147,483,647 の範囲)						
	<table><tr><th>値 1</th><th>結果</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>		値 1	結果	0	1	1	0
	値 1		結果					
	0		1					
1	0							
フォーマット								
フローチャート方式		テキスト言語方式						
<結果> := not <値> ↓		<結果> := not <値>						
*1. UB(1) = not UB(0) のような場合では、正しい結果を取得できません。 ビットデータサイズの変数を反転させる場合は、次の例に示すように、xor 命令を使用してください。 - 例 1: UB(1) = UB(0) xor 1 - 例 2: UB(2) = X(00) xor 1								
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。発生しないように、アプリケーションで処理する必要があります。								

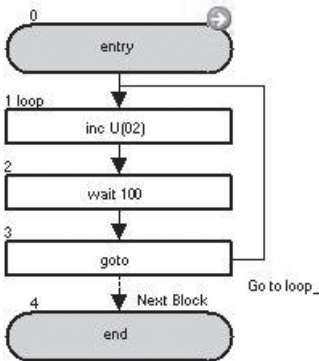
● 例

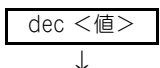
フローチャート	テキスト
	<pre> entry U(03) := 12 :loop_ U(04) := not U(03) goto loop_ end </pre>

ブロック No.	動作
1	U(03) に 12 を代入します。
2	U(04) に U(03) の 12 (2 進数: 00000110) の否定 (ビット反転) した 65523 (2 進数: 111110011) を代入します。
3	無条件で 3: loop_ に分岐します。

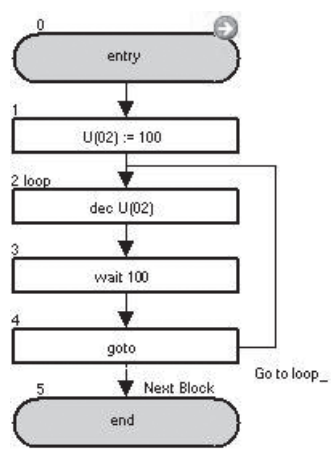
Inc (1 ずつ増加)		
命令	説明	引数
	<値>を 1 ずつ増やします。	値: 任意の変数
フォーマット		
フローチャート方式		テキスト言語方式
		inc <値>
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。発生しないように、アプリケーションで処理する必要があります。		

● 例

フローチャート		テキスト
		<pre>entry :loop_ inc U(02) wait 100 goto loop_ end</pre>
ブロック No.	動作	
1	U(02) の値を +1 します。	
2	1.00 秒間 wait 命令で待機します。	
3	無条件で 1: loop_ に分岐します。	

Dec (1 ずつ減少)		
命令	説明	引数
	<値>を 1 ずつ減らします。	値: 任意の変数
フォーマット		
フローチャート方式		テキスト言語方式
		dec <値>
(注) DriveProgramming は、オーバーフローやアンダーフローが発生した場合、エラーとします。発生しないように、アプリケーションで処理する必要があります。		

● 例

フローチャート	テキスト
	<pre>entry U(02) := 100 :loop_ dec U(02) wait 100 goto loop_ end</pre>

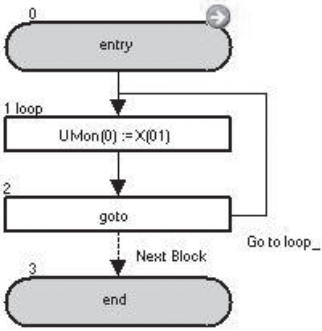
ブロック No.	動作
1	U(02) に 100 を代入します。
2	U(02) の値を -1 します。
3	1.00 秒間 wait 命令で待機します。
4	無条件で 2: loop_ に分岐します。

6-6 入出力制御命令

入出力制御には、この入出力制御命令を使用します。＝（代入）命令でも可能ですが、入出力制御命令の方が引数のデータサイズが小さく、プログラム容量を効率的に使用できます。

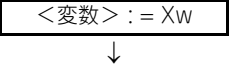
var = X(i)		
命令	説明	引数
	入力端子変数の状態を 1 ビット分<変数>に代入します。	変数 ：任意の変数（変数の値は 0 または 1） i ：入力端子変数の番号（00 ～ 07 の範囲）
フォーマット		
フローチャート方式		テキスト言語方式
<変数> := X(i) ↓		<変数> := X(i)
（注）入力端子変数はインバータ本体の入力端子の状態を認識する変数です。以下のように設定する必要があります。入力端子変数の番号順は、設定した汎用入力番号順に従います。 【MX2】 多機能入力 S1 ～ S7 選択（C001 ～ C007）に、汎用入力 MI1 ～ 7（56 ～ 62）を設定。パルス列入力端子 RP 選択（P003）に、「02: MI8（DriveProgramming 用汎用入力 8）」を設定。 【RX】 多機能入力 S1 ～ S8 選択（C001 ～ C008）に、汎用入力 MI1 ～ 8（56 ～ 63）を設定。 <割付例> X(00) = MI1（機能番号：56） X(01) = MI2（機能番号：57） X(02) = MI3（機能番号：58） X(03) = MI4（機能番号：59） X(04) = MI5（機能番号：60） X(05) = MI6（機能番号：61） X(06) = MI7（機能番号：62） X(07) = MI8（【MX2】 P003=02 / 【RX】 機能番号：63） （注）詳細については、□「5-2 入出力端子変数（P.5-5）」を参照してください。		

● 例

フローチャート	テキスト
	<pre>entry :loop_ UMon(0) := X(01) goto loop_ end</pre>

上記の例では、入力端子 X(01) の状態をパラメータ UMon(0) [d025] でモニタします。

ブロック No.	動作
1	UMon(0) に X(01) を代入します。
2	無条件で 1: loop_ に分岐します。

var = Xw		
命令	説明	引数
	入力端子変数の状態をワード単位で<変数>に代入します。	変数：任意の変数
フォーマット		
フローチャート方式		テキスト言語方式
		<変数> := Xw

(注) 入力端子変数はインバータ本体の入力端子の状態を認識する変数です。以下のように設定する必要があります。入力端子変数の番号順は、設定した汎用入力の番号順に従います。

【MX2】 多機能入力 S1 ～ S7 選択 (C001 ～ C007) に、汎用入力 MI1 ～ 7 (56 ～ 62) を設定。パルス列入力端子 RP 選択 (P003) に、「02: MI8 (DriveProgramming 用汎用入力 8)」を設定。

【RX】 多機能入力 S1 ～ S8 選択 (C001 ～ C008) に、汎用入力 MI1 ～ 8 (56 ～ 63) を設定。

ワード単位で代入する場合、上位バイトや設定していない入力端子変数はゼロが読み出されます。

<割付例>

Xw = 1 (ビット 0) ← X(00) = MI1 (機能番号: 56)

Xw = 2 (ビット 1) ← X(01) = MI2 (機能番号: 57)

Xw = 4 (ビット 2) ← X(02) = MI3 (機能番号: 58)

Xw = 8 (ビット 3) ← X(03) = MI4 (機能番号: 59)

Xw = 16 (ビット 4) ← X(04) = MI5 (機能番号: 60)

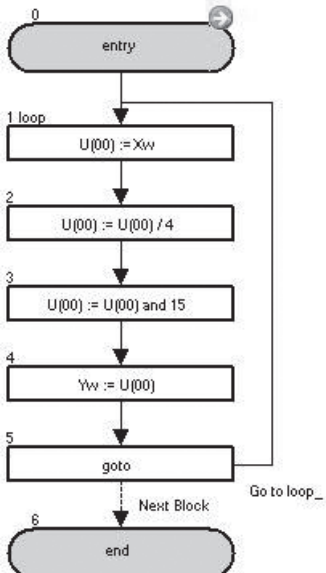
Xw = 32 (ビット 5) ← X(05) = MI6 (機能番号: 61)

Xw = 64 (ビット 6) ← X(06) = MI7 (機能番号: 62)

Xw = 128 (ビット 7) ← X(07) = MI8 (【MX2】 P003=02 / 【RX】 機能番号: 63)

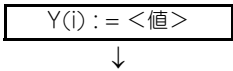
詳細については、□「5-2 入出力端子変数 (P.5-5)」を参照してください。

● 例

フローチャート	テキスト
	<pre> entry :loop_ U(00) := Xw U(00) := U(00) / 4 U(00) := U(00) and 15 Yw := U(00) goto loop_ end </pre>

上記の例では、入力端子 X(02) ～ X(05) の状態を取得して、出力端子 Y(00) ～ Y(03) に出力します。

ブロック No.	動作
1	U(00) に Xw (入力端子の値) を代入します。
2	U(00) の値を 4 で割って (2 ビット右シフト)、ビット 0 に X(02) が割り付くようにします。
3	U(00) と 15 (2 進数: 00001111) の論理積をとって、X(06) より上位のビットを 0 にします。
4	Yw に U(00) を代入します。
5	無条件で 1: loop_ に分岐します。

Y(i) = 値		
命令	説明	引数
	出力端子変数にビット単位で出力します。	i: 出力端子変数の番号 (0 ~ 6 の範囲) 値: 任意の変数または定数
フォーマット		
フローチャート方式		テキスト言語方式
		Y(i) := <値>

(注) 出力端子変数はインバータ本体の出力端子を制御する変数です。以下のように設定する必要があります。出力端子変数の番号順は、設定した汎用出力の番号順に従います。

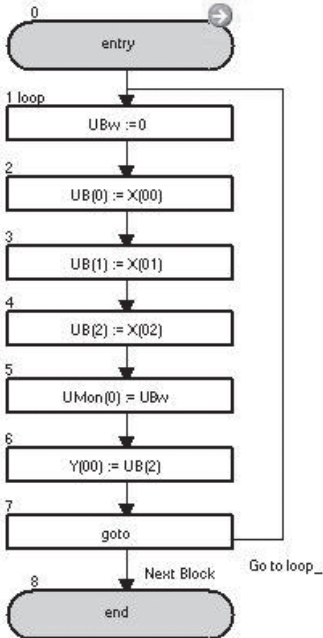
- 【MX2】 多機能出力 P1、P2 選択 (C021, C022)、多機能リレー出力 (MA, MB) 選択 (C026) に、汎用出力 MO1 ~ 3 (44 ~ 46) を設定。
- 【RX】 多機能出力 P1 ~ P5 選択 (C021 ~ C025)、多機能リレー出力 (MA, MB) 選択 (C026) に、汎用出力 MO1 ~ 6 (44 ~ 49) を設定。

<割付例>

Y(00) = MO1 (多機能番号: 44)
Y(01) = MO2 (多機能番号: 45)
Y(02) = MO3 (多機能番号: 46)
Y(03) = MO4 (【RX だけの機能】多機能番号: 47)
Y(04) = MO5 (【RX だけの機能】多機能番号: 48)
Y(05) = MO6 (【RX だけの機能】多機能番号: 49)

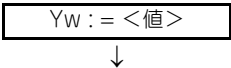
詳細については、□「5-2 入出力端子変数 (P.5-5)」を参照してください。

● 例

フローチャート	テキスト
	<pre>entry :loop_ ubw := 0 UB(0) := X(00) UB(1) := X(01) UB(2) := X(02) UMon(0) := ubw Y(00) := UB(2) goto loop_ end</pre>

上記の例では、X(00) ~ X(02) の状態をパラメータ UMon(0) [d025] でモニタし、X(02) を Y(00) に出力します。

ブロック No.	動作
1	UBw に 0 を代入します。
2 ~ 4	UB(0) ~ UB(2) に X(00) ~ X(02) を代入します。
5	UMon(0) に UBw を代入します。
6	Y(00) に UB(2) を代入します。
7	無条件で 1: loop_ に分岐します。

Yw = 値		
命令	説明	引数
	出力端子変数にワード単位で出力します。 各ビットを、それぞれの出力に反映します。	値: 任意の変数または定数
フォーマット		
フローチャート方式		テキスト言語方式
		Yw := <値>

(注) 出力端子変数はインバータ本体の出力端子を制御する変数です。以下のように設定する必要があります。出力端子変数の番号順は、設定した汎用出力の番号順に従います。

【MX2】 多機能出力 P1、P2 選択 (C021, C022)、多機能リレー出力 (MA, MB) 選択 (C026) に、汎用出力 MO1 ~ 3 (44 ~ 46) を設定。

【RX】 多機能出力 P1 ~ P5 選択 (C021 ~ C025)、多機能リレー出力 (MA, MB) 選択 (C026) に、汎用出力 MO1 ~ 6 (44 ~ 49) を設定。

ワード単位で代入する場合、上位バイトや設定していない出力端子変数はゼロが読み出され、設定した場合はその設定を無視します。

<割付例>

Yw = 1 (ビット 0) → Y(00) = MO1 (多機能番号: 44)

Yw = 2 (ビット 1) → Y(01) = MO2 (多機能番号: 45)

Yw = 4 (ビット 2) → Y(02) = MO3 (多機能番号: 46)

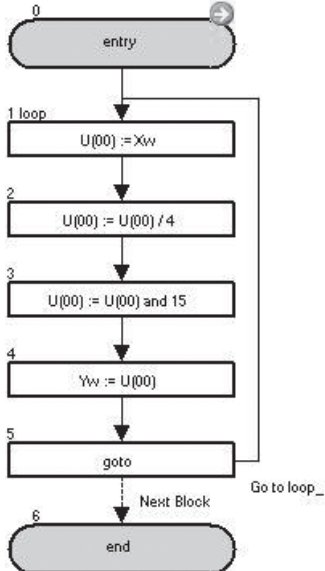
Yw = 8 (ビット 3) → Y(03) = MO4 (【RX だけの機能】多機能番号: 47)

Yw = 16 (ビット 4) → Y(04) = MO5 (【RX だけの機能】多機能番号: 48)

Yw = 32 (ビット 5) → Y(05) = MO6 (【RX だけの機能】多機能番号: 49)

詳細については、□「5-2 入出力端子変数 (P.5-5)」を参照してください。

● 例

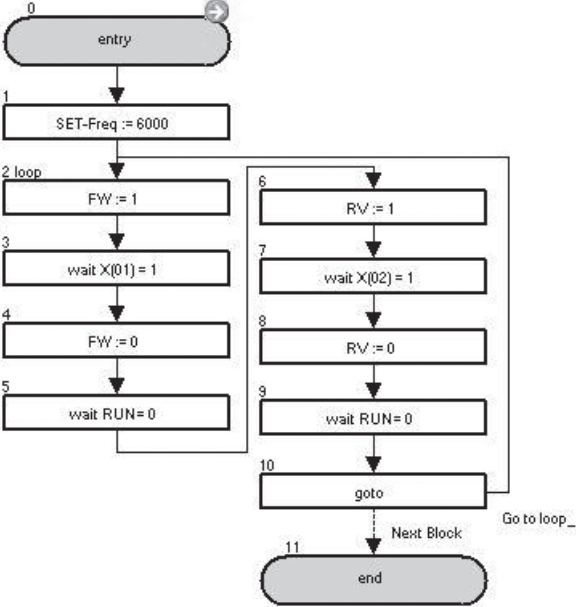
フローチャート	テキスト
	<pre> entry :loop_ U(00) := Xw U(00) := U(00) / 4 U(00) := U(00) and 15 yw := U(00) goto loop_ end </pre>

上記の例では、入力端子 X(02) ~ X(05) の状態を取得して、出力端子 Y(00) ~ Y(03) に出力します。

ブロック No.	動作
1	U(00) に Xw (入力端子の値) を代入します。
2	U(00) の値を 4 で割って (2 ビット右シフト)、ビット 0 に X(02) が割り付くようにします。
3	U(00) と 15 (2 進数: 00001111) の論理積をとって、X(06) より上位のビットを 0 にします。
4	Yw に U(00) を代入します。
5	無条件で 1: loop_ に分岐します。

func = 値		
命令	説明	引数
	<値>を多機能入力変数に代入します。	機能 ：多機能入力変数の任意の機能ビット（各ビットの詳細は □「5-6 多機能入力変数 (P.5-17)」を参照してください。） 値 ：任意の変数または定数
フォーマット		
フローチャート方式		テキスト言語方式
＜機能＞ := ＜値＞ ↓		<機能> := <値>

● 例

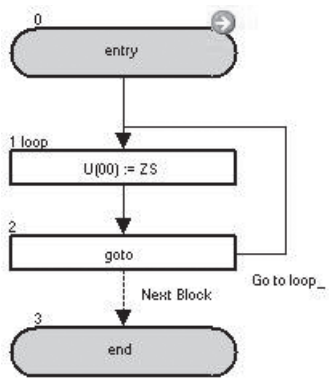
フローチャート	テキスト
	<pre>entry SET-Freq := 6000 :loop_ FW := 1 wait X(01) = 1 FW := 0 wait RUN = 0 RV := 1 wait X(02) = 1 RV := 0 wait RUN = 0 goto loop_ end</pre>

上記の例では、60.00Hz での X(01) が ON するまで正転し、X(02) が ON するまで逆転することが連続して繰り返されます。

ブロック No.	動作
1	出力周波数を 60.00Hz に設定します。
2	インバータを正転します。
3～5	X(01) が ON するまで待機し、インバータを停止し、停止するまで wait 命令で待機します。
6	インバータを逆転します。
7～9	X(02) が ON するまで待機し、インバータを停止し、停止するまで wait 命令で待機します。
10	無条件で 2: loop_ に分岐します。

var = func		
命令	説明	引数
	多機能出力変数のステータスを<変数>に代入します。	変数 : 任意の変数 機能 : 多機能出力変数の任意のビット
フォーマット		
フローチャート方式		テキスト言語方式
<変数> := <機能> ↓		<変数> := <機能>
詳細は □「5-7 多機能出力変数 (P.5-20)」を参照してください。		

● 例

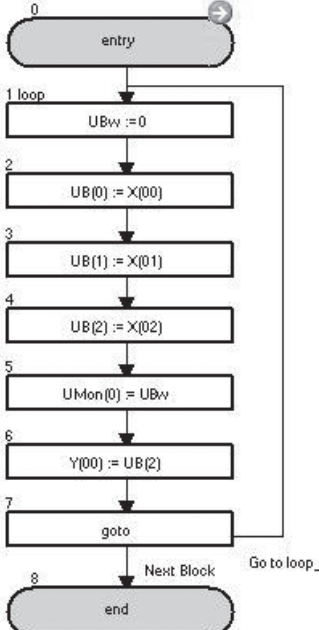
フローチャート	テキスト
	<pre> entry :loop_ U(00) := ZS goto loop_ end </pre>

上記の例では、U(00) に ZS（ゼロ速度信号）が ON である場合は 1、OFF である場合は 0 を代入します。

ブロック No.	動作
1	ZS の状態を U(00) に代入します。
2	無条件で 1: loop_ に分岐します。

var = UB(i)		
命令	説明	引数
	内部ユーザ接点の値を 1 ビット分<変数>に代入します。	変数 : 任意の変数 (変数の値は 0 または 1) i : 内部ユーザ接点の番号 (0 ~ 7 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
<変数> := UB(i) ↓		<変数> := UB(i)

● 例

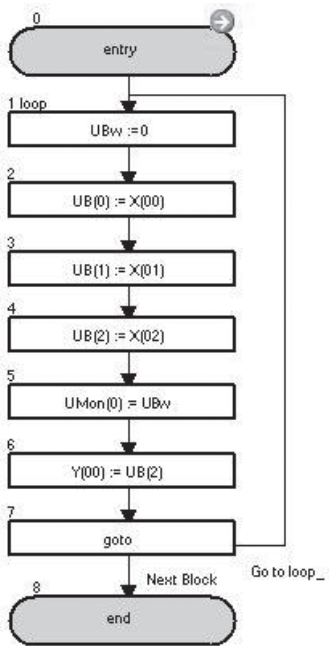
フローチャート	テキスト
	<pre>entry :loop_ ubw := 0 UB(0) := X(00) UB(1) := X(01) UB(2) := X(02) UMon(0) := ubw Y(00) := UB(2) goto loop_ end</pre>

上記の例では、X(00) ~ X(02) の状態をパラメータ UMon(0) [d025] でモニタし、X(02) を Y(00) に出力します。

ブロック No.	動作
1	UBw に 0 を代入します。
2 ~ 4	UB(0) ~ UB(2) に X(00) ~ X(02) を代入します。
5	UMon(0) に UBw を代入します。
6	Y(00) に UB(2) を代入します。
7	無条件で 1: loop_ に分岐します。

var = UBw		
命令	説明	引数
	内部ユーザ接点の値を、ワード単位で<変数>に代入します。	変数：任意の変数
フォーマット		
フローチャート方式		テキスト言語方式
<変数> := UBw ↓		<変数> := UBw
(注) ワード単位で代入する場合、使用していない上位バイトはゼロが読み出され、設定した場合はその設定を無視します。 UBw = 1 (ビット 0) ← UB(0) UBw = 2 (ビット 1) ← UB(1) UBw = 4 (ビット 2) ← UB(2) UBw = 8 (ビット 3) ← UB(3) UBw = 16 (ビット 4) ← UB(4) UBw = 32 (ビット 5) ← UB(5) UBw = 64 (ビット 6) ← UB(6) UBw = 128 (ビット 7) ← UB(7)		

● 例

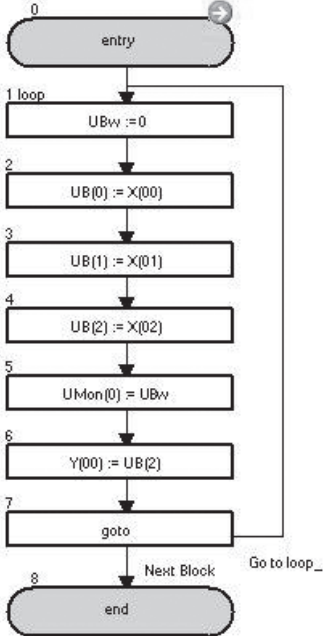
フローチャート	テキスト
 <pre> graph TD Entry([entry]) --> Loop1[1 loop] Loop1 --> UBw0[UBw := 0] UBw0 --> UB0[2 UB(0) := X(00)] UB0 --> UB1[3 UB(1) := X(01)] UB1 --> UB2[4 UB(2) := X(02)] UB2 --> UMon0[5 UMon(0) := UBw] UMon0 --> Y00[6 Y(00) := UB(2)] Y00 --> Goto[7 goto] Goto -- "Go to loop_" --> Loop1 Goto -- "Next Block" --> End([8 end]) </pre>	<pre> entry :loop_ ubw := 0 UB(0) := X(00) UB(1) := X(01) UB(2) := X(02) UMon(0) := ubw Y(00) := UB(2) goto loop_ end </pre>

上記の例では、X(00)～X(02)の状態をパラメータ UMon(0) [d025] でモニタし、X(02)をY(00)に出力します。

ブロック No.	動作
1	UBw に 0 を代入します。
2～4	UB(0)～UB(2) に X(00)～X(02) を代入します。
5	UMon(0) に UBw を代入します。
6	Y(00) に UB(2) を代入します。
7	無条件で 1: loop_ に分岐します。

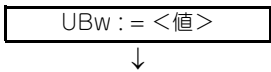
UB(i) = 値		
命令	説明	引数
	<値>を内部ユーザ接点に代入します。	i: 内部ユーザ接点の番号 (0 ～ 7 の範囲) 値: 任意の変数または定数
フォーマット		
フローチャート方式		テキスト言語方式
UB(i) := <値> ↓		UB(i) := <値>

● 例

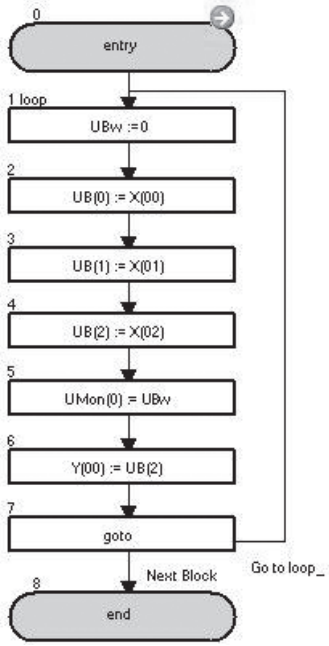
フローチャート	テキスト
	<pre>entry :loop_ ubw := 0 UB(0) := X(00) UB(1) := X(01) UB(2) := X(02) UMon(0) := ubw Y(00) := UB(2) goto loop_ end</pre>

上記の例では、X(00) ～ X(02) の状態をパラメータ UMon(0) [d025] でモニタし、X(02) を Y(00) に出力します。

ブロック No.	動作
1	UBw に 0 を代入します。
2 ～ 4	UB(0) ～ UB(2) に X(00) ～ X(02) を代入します。
5	UMon(0) に UBw を代入します。
6	Y(00) に UB(2) を代入します。
7	無条件で 1: loop_ に分岐します。

UBw = 値		
命令	説明	引数
	<値>をワード単位で内部ユーザ接点に代入します。	値 : 任意の変数または定数
フォーマット		
フローチャート方式		テキスト言語方式
		UBw := <値>
(注) ワード単位で代入する場合、使用していない上位バイトはゼロが読み出され、設定した場合はその設定を無視します。 UBw = 1 (ビット 0) → UB(0) UBw = 2 (ビット 1) → UB(1) UBw = 4 (ビット 2) → UB(2) UBw = 8 (ビット 3) → UB(3) UBw = 16 (ビット 4) → UB(4) UBw = 32 (ビット 5) → UB(5) UBw = 64 (ビット 6) → UB(6) UBw = 128 (ビット 7) → UB(7)		

● 例

フローチャート	テキスト
	<pre> entry :loop_ ubw := 0 UB(0) := X(00) UB(1) := X(01) UB(2) := X(02) UMon(0) := ubw Y(00) := UB(2) goto loop_ end </pre>

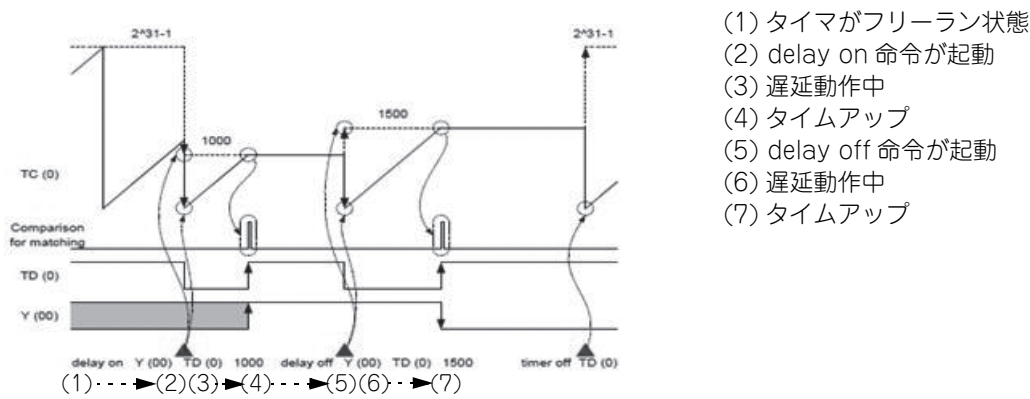
上記の例では、X(00)～X(02)の状態をパラメータ UMon(0) [d025] でモニタし、X(02)をY(00)に出力します。

ブロック No.	動作
1	UBw に 0 を代入します。
2～4	UB(0)～UB(2) に X(00)～X(02) を代入します。
5	UMon(0) に UBw を代入します。
6	Y(00) に UB(2) を代入します。
7	無条件で 1: loop_ に分岐します。

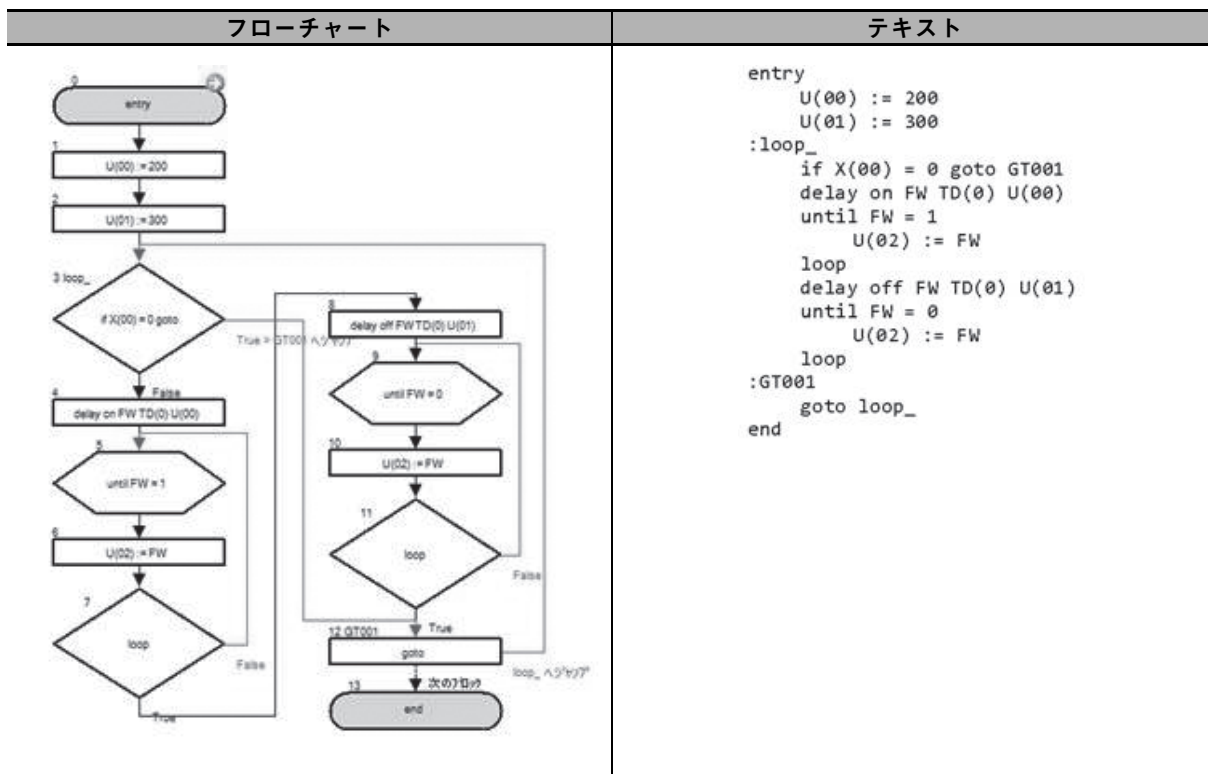
6-7 タイマ制御命令

Delay on/off		
命令	説明	引数
	タイマ (k) を起動します。タイマカウンタ TC(k) は 0 から開始し、<値 2>に達するまで 10ms ごとに増加し、<値 2>に達するとタイマ接点 TD(K) が ON に切り替わり、<値 1>についても指定した処理 (on または off) が実行されます。	on/off: 遅延時間後の動作設定 (on/off) 値 1: 任意の接点変数または変数 値 2: 任意の変数または定数 (×10ms で指定する時間) TD(k): 使用するタイマのタイマ出力接点 (k は 0 ~ 7 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
delay on/off <値 1> TD(k) <値 2> ↓		delay on/off <値 1> TD(k) <値 2>
(注) 1. delay on/off 命令を実行すると、タイマ (k) を起動します。TD(k) が ON 状態の場合は、タイマ (k) 起動時に TD(k) が OFF しますが、<値 1>の変数は元の値から変化しません。 タイマ (k) 起動したあと、プログラムの処理は次の命令に移行します。 2. delay on/off 命令を実行すると、実行時の<値 1>、TD(k)、<値 2>のデータが内部に保存されます。実行後に<値 1>、<値 2>に設定された変数の値を変更しても、delay on/off 命令動作は変わりません。 3. delay on/off 命令を実行したあと、その処理が完了する前に再度同じタイマ (k) を起動すると、先に実行された処理は取り消され、再度設定した実行内容でタイマ (k) を起動します。このため、タイマ (k) を起動したあとは、処理が完了するまで、再実行しないようにプログラムしてください。 4. 起動されたタイマ (k) のタイマデータは、タイマカウンタ変数 TC(k) でモニタできます。タイマ処理の完了はタイマ出力接点 TD(k) で確認します (完了時に ON します)。		

● タイミングチャート



● 例

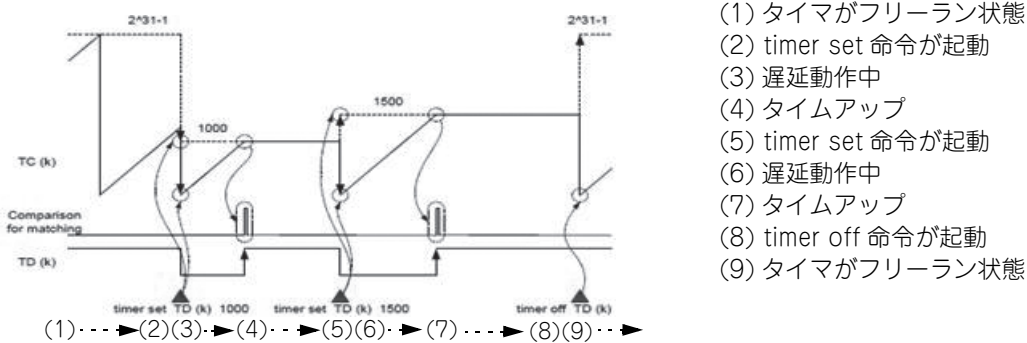


上記の例では、X(00)= 0 でないとき、delay on 命令で正転運転を開始し、delay off で運転を停止します。

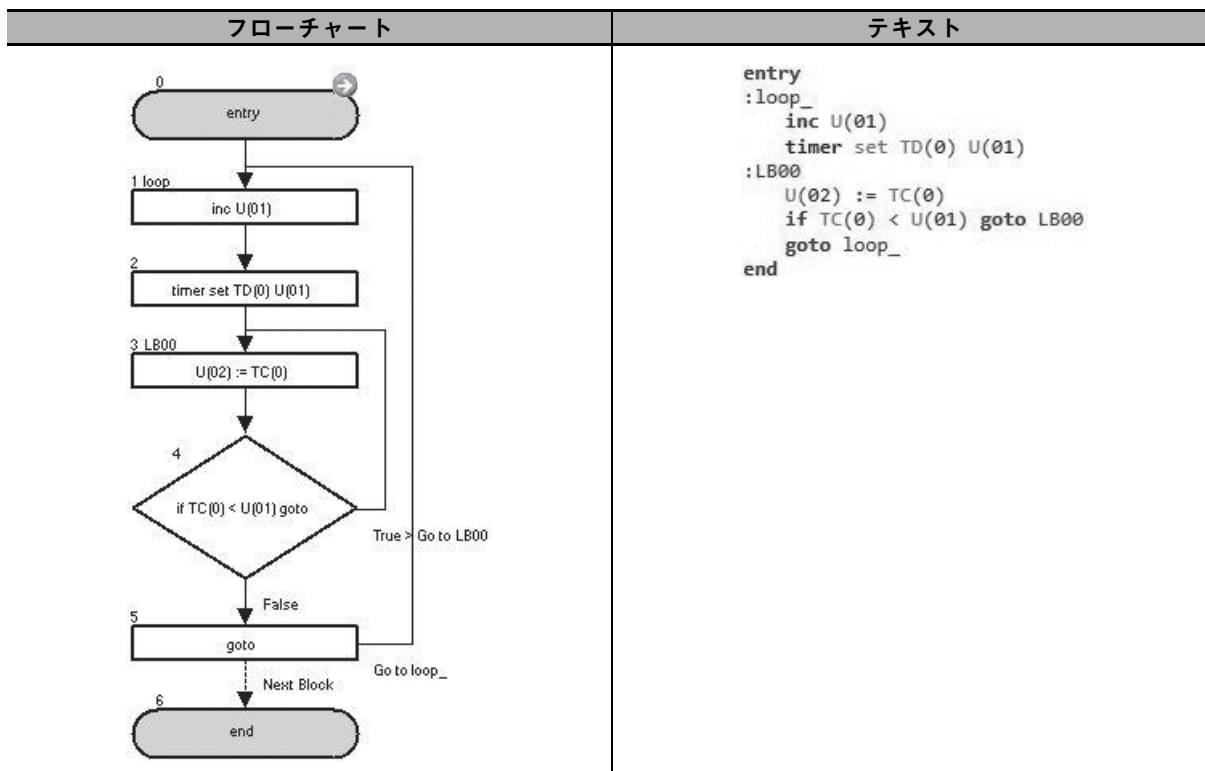
ブロック No.	動作
1	U(00) に 200 を代入します。
2	U(01) に 300 を代入します。
3	X(00) が 0 のとき、8: GT001 に分岐します。X(00) が 0 でないとき次へ移行します。
4	delay on 命令でタイマ (0) を設定値 U(00) で起動します。
5～7	FW が 1 になるまでの間、U(02) に FW の状態を代入します。FW が 1 になると次へ移行します。
8	delay off 命令でタイマ (0) を設定値 U(01) で起動します。
9～11	FW が 0 になるまでの間、U(02) に FW の状態を代入します。FW が 0 になると次へ移行します。
12	無条件で 3: loop_ に分岐します。

Timer Set		
命令	説明	引数
	タイマ (k) を起動します。タイマカウンタ TC(k) は 0 から開始し、<値>に達するまで 10ms ごとに増加し、<値>に達するとタイマ 接点 TD(k) を、ON します。	TD(k): 使用するタイマのタイマ出力接点 (k は 0 ~ 7 の範囲) 値: 任意の変数または定数 (×10ms で指定する時間)
フォーマット		
フローチャート方式		テキスト言語方式
timer set TD(k) <値> ↓		timer set TD(k) <値>
(注) 1. timer set 命令を実行すると、タイマ (k) を起動します。TD(k) が ON 状態の場合は、タイマ (k) 起動時に TD(k) が OFF します。タイマ (k) 起動したあと、プログラムの処理は次の命令に移行します。 2. timer set 命令を実行すると、実行時の TD(k)、<値>のデータが内部に保存されます。実行後に<値>に設定された変数の値を変更しても、timer set 命令動作は変わりません。 3. timer set 命令を実行したあと、その処理が完了する前に再度同じタイマ (k) を起動すると、先に実行された処理は取り消され、再度設定した実行内容でタイマ (k) を起動します。このため、タイマ (k) を起動したあとは、処理が完了するまで、再実行しないようにプログラムしてください。 4. 起動されたタイマ (k) のタイマデータは、タイマカウンタ変数 TC(k) でモニタできます。タイマ処理の完了はタイマ出力接点 TD(k) で確認します (完了時に ON します)。		

● タイミングチャート



● 例

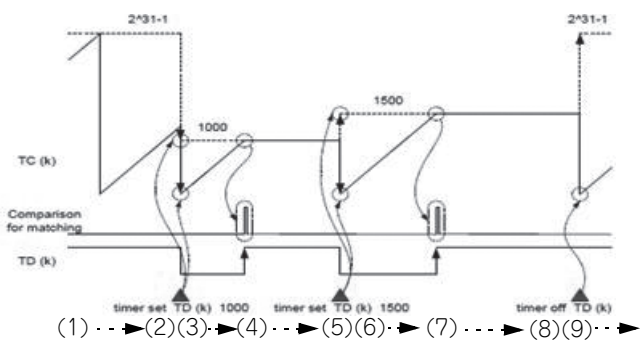


上記の例では、タイマの実行完了ごとに TD(0) の設定値が 1 増加し、各ループの所要時間が実行ごとに長くなります。タイマの現在値は U(02) に反映されます。

ブロック No.	動作
1	U(01) を +1 します。
2	timer set 命令でタイマ (0) を設定値 U(01) で起動します。
3	タイマ (0) の現在値 TC(0) を U(02) に代入します。
4	タイマ (0) 現在値 TC(0) が設定値 U(01) より小さいとき、3: LB00 に分岐します。それ以外の場合、次に移行します。
5	無条件で 1: loop_ に分岐します。

Timer Off		
命令	説明	引数
 TimerOff	タイマカウンタ TC(k) を 0 にリセットし、フリーランモードで起動します。	TD(k): 使用するタイマのタイマ出力接点 (k は 0 ~ 7 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
timer off TD(k) ↓		timer off TD(k)

● タイミングチャート



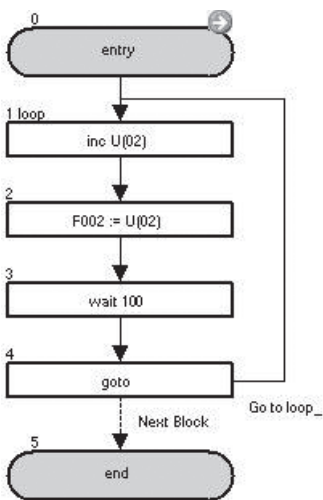
- (1) タイマがフリーラン状態
- (2) timer set 命令が起動
- (3) 遅延動作中
- (4) タイムアップ
- (5) timer set 命令が起動
- (6) 遅延動作中
- (7) タイムアップ
- (8) timer off 命令が起動
- (9) タイマがフリーラン状態

6-8 パラメータ制御命令

ChgParam		
命令	説明	引数
	<パラメータ>で指定されたインバータのパラメータデータを、<値>に変更します。 任意のインバータパラメータを変更できます。	パラメータ ：パラメータコード (Fxxx, Axxx, bXXX, Cxxx, Hxxx, Pxxx) 値 ：任意の変数または定数
フォーマット		
フローチャート方式		テキスト言語方式
<パラメータ> := <値> ↓		<パラメータ> := <値>

- (注) 1. 各パラメータには、インバータ本体のデジタルオペレータで設定する場合と同様に、以下のような制約事項があります。制約事項に示す状態が発生するとインバータが命令異常 1 (E45) を検出し、DriveProgramming 機能のプログラムが停止します。パラメータの詳細な制約事項については、□「第 7 章 DriveProgramming で使用するパラメータの注意事項」や □『多機能型小型インバータ形 3G3MX2 シリーズ V1 タイプユーザーズマニュアル (SBCE-376)』、□『多機能型小型インバータ形 3G3MX2 シリーズ V2 タイプユーザーズマニュアル (SBCE-506)』、または □『高機能型汎用インバータ形 3G3RX シリーズ V1 タイプユーザーズマニュアル (SBCE-367)』を参照してください。
- ・設定するパラメータの設定範囲を超えたデータを設定した。
 - ・設定するパラメータと関連するパラメータとの整合異常が発生した。
 - ・選択したモードで使用できない機能を設定した。
 - ・運転中に変更できないパラメータを、運転中に変更した。
 - ・ソフトロック機能 (b031) で、パラメータのデータ変更を禁止している状態で設定した。
2. ChgParam 命令でパラメータの設定データを変更しても、そのパラメータの EEPROM のデータは変更されません。EEPROM のデータを含めて変更する場合は、eepwrt 命令を先に実行し、直後に ChgParam 命令を実行してください。

● 例

フローチャート	テキスト
 <pre> graph TD entry([entry]) --> loop[1 loop inc U(02)] loop --> f002[F002 := U(02)] f002 --> wait[3 wait 100] wait --> goto[4 goto] goto -- "Go to loop_" --> loop goto -.-> end([5 end]) </pre>	<pre> entry :loop_ inc U(02) F002 := U(02) wait 100 goto loop_ end </pre>

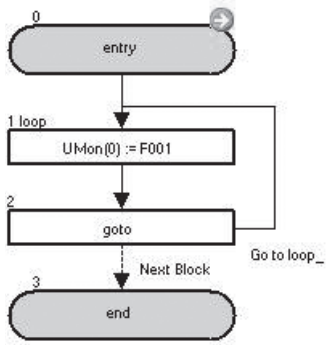
上記の例では、F002（第 1 加速時間 1）の値を、1 秒ごとに 1 ずつ増加します。

ブロック No.	動作
1	U(02) を +1 します。
2	パラメータ F002 に U(02) を代入します。
3	1.00 秒間 wait 命令で待機します。
4	無条件で 1: loop_ に分岐します。

MonParam

命令	説明	引数
	<パラメータ> で指定されたインバータパラメータの内容を <変数> に代入します。	パラメータ : パラメータコード (Fxxx, Axxx, bxxx, Cxxx, dxxx, Hxxx, Pxxx) 変数 : 任意の変数
フォーマット		
フローチャート方式		テキスト言語方式
<変数> := <パラメータ> ↓		<変数> := <パラメータ>

● 例

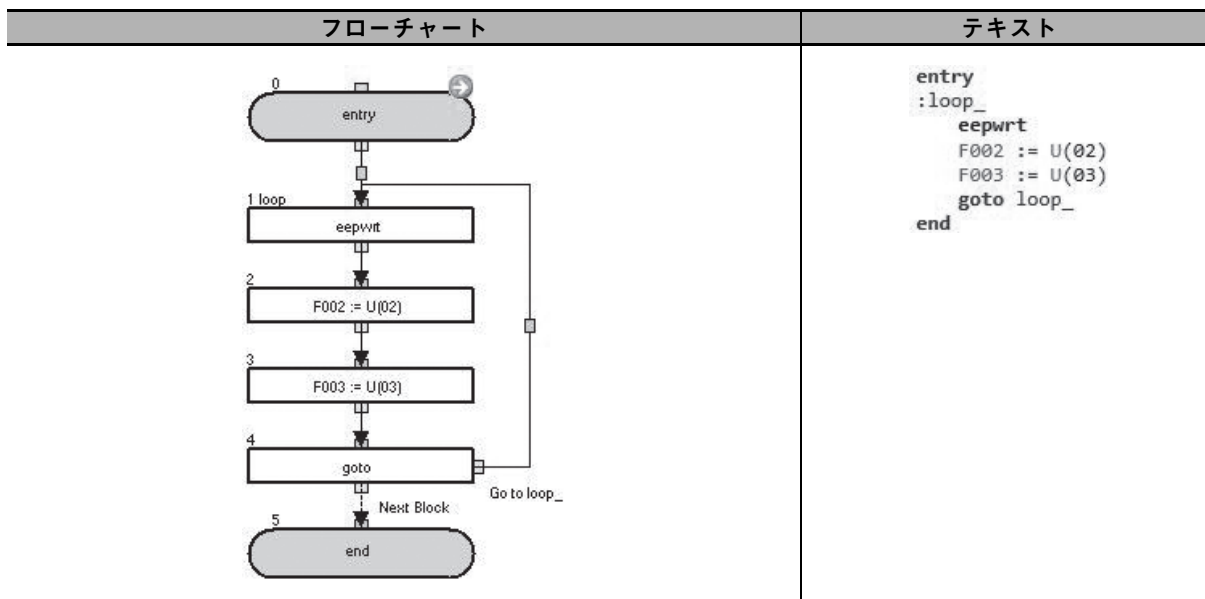
フローチャート	テキスト
 <pre> graph TD entry([entry]) --> loop[1 loop UMon(0) := F001] loop --> goto[2 goto] goto -- "Go to loop_" --> loop goto -- "Next Block" --> end([3 end]) </pre>	<pre> entry :loop_ UMon(0) := F001 goto loop_ end </pre>

上記の例では、パラメータ F001（出力周波数設定）の値をパラメータ UMon(0) [d025]（ユーザパラメータモニタ）に代入しモニタします。

ブロック No.	動作
1	UMon(0) にパラメータ F001 を代入します。
2	無条件で 1: loop_ に分岐します。

EepWrt		
命令	説明	引数
	この命令を実行した直後の ChgParam 命令実行時に EEPROM に書き込むことができるようにします。	---
フォーマット		
フローチャート方式		テキスト言語方式
eepwrt ↓		eepwrt
(注) 1. eepwrt 命令を実行した直後に、ChgParam 命令でパラメータのデータを設定してください。そのパラメータデータの変更は、プログラム上で変更され、さらに EEPROM に保存されます。次の電源投入時、開始データとして EEPROM に保存されたデータが使用できます。eepwrt 命令の直後に、ChgParam 命令が実行されない場合、EEPROM にデータは保存されません。また、eepwrt 命令の直後の 2 つ ChgParam 命令を実行しても、後側の ChgParam 命令は、EEPROM にデータが保存されません。 2. eepwrt 命令を実行した場合、その処理には最大 10ms の時間が必要です。eepwrt 命令を実行したタスクは eepwrt 命令の完了を待ってから、次の命令に移行します。 3. eepwrt 命令を、複数のタスクで同時に実行したり、連続で実行することを避けてください。1 つの eepwrt 命令の処理には最大 10ms が必要で、連続して eepwrt 命令を実行すると最初の eepwrt 命令が完了したあとに実行されます。その間、連続して eepwrt 命令を実行したタスクは次の命令に移行せず、待機状態となります。複数回の eepwrt 命令の実行が連続した場合、プログラム処理時間が不足し、他のタスク処理が遅れたり、異常を検出する場合があります。 4. 電源投入後 1 秒間の間に、eepwrt 命令を実行しても正常に処理できません。電源投入後、wait 命令などを使用して 1 秒間の待ち時間をプログラムしてください。 5. インバータの EEPROM メモリは書替え回数に限界（約 10 万回）があります。このため、eepwrt 命令を頻繁に使用すると、インバータの寿命を縮めるおそれがあります。プログラムの中で、以下のような対応を考慮してしてください。 - 電源遮断前の最終データを EEPROM に書き込むようにプログラムを構成する。 - パラメータなどを変更した場合だけ、EEPROM に書き込むようにプログラムを構成する。 - 電源遮断時のデータを保存するユーザパラメータ変数 U(29) ~ U(31) を使用する。 6. eepwrt 命令で複数のパラメータを EEPROM に連続して保存する必要がある場合、各 eepwrt 命令を実行する間に、少なくとも 30ms 以上の間隔が必要です。 次ページに考えられる例を示します。		

● 例：（EEPROM に F002 だけを保存する場合）



上記の例では、パラメータ F002 は U(02) の値に変更され EEPROM 内に保存されます。F003 は U(03) の値に変更されますが、EEPROM 内には保存されません。

ブロック No.	動作
1	次の ChgParam 命令で EEPROM に保存する設定にします。
2	パラメータ F002 に U(02) を代入します。直前の eepwrt 命令により、EEPROM にも保存します。
3	パラメータ F003 に U(03) を代入します。この値は EEPROM に保存されません。
4	無条件で 1: loop_ に分岐します。

【RX だけの機能】 RtcSet

命令	説明	引数
	LCD デジタルオペレータから送られてくる、6 バイトの時刻データを変数に設定します。このデータは、年、月、日、曜日、時間、分に対応します。 16 進数の変数値は、年、月、日、曜日、時間、分（10 進数）に対応します。 rtcset on: 6 バイトのデータを連続して更新します。 rtcset off: 6 バイトのデータを 1 回だけ更新します。	on/off: 連続 / 1 回 ユーザ変数: 任意のユーザパラメータ変数または内部ユーザ変数 (U(xx) または UL(xx))
フォーマット		
フローチャート方式		テキスト言語方式
rtcset on/off <ユーザ変数> ↓		rtcset on/off <ユーザ変数>

- (注) 1. LCD デジタルオペレータ（オプション）に搭載された時刻機能を使用した機能です。LCD デジタルオペレータが接続されていない場合に、rtcset on/off 命令を実行すると、その処理が完了せずに、全プログラムはその命令で待機状態となります。このため、プログラム起動直後に 1 度だけ rtcset on 命令を起動し、常に指定した変数へ時刻データを出力するプログラム構成を推奨します。
2. LCD デジタルオペレータの時刻は初期データ（2009 年 1 月 1 日 THU 00:00）で出荷されています。rtcset on/off 命令を使用する場合には、正確に時刻を設定してください。また、時刻機能には時刻誤差（月差：-1.5 分～+1.5 分）がありますので、考慮して使用してください。
3. rtcset on/off 命令を実行すると、バイト単位で BCD データを出力します。曜日データは日曜日から順に 00、土曜日は 06 で示します。しかし、ユーザパラメータ変数 U(k) や内部ユーザ変数は BCD データの認識ができません。DriveProgramming エディタやデジタルオペレータでは、BCD データを 16 進数として認識し、10 進数データに変換して表示しますので、プログラムを行う場合は注意してください。
4. rtcset on/off U(k) を実行すると、以下の順で上位バイトから設定されます。U(k) を含めて連続した 3 つのユーザパラメータ変数が必要ですので、確保してください。

ユーザパラメータ変数	上位バイト	下位バイト
U(k)	年 BCD データ	月 BCD データ
U(k+1)	日 BCD データ	曜日 BCD データ
U(k+2)	時間 BCD データ	分 BCD データ

5. rtcset on/off UL(k) を実行すると、以下の順で上位バイトから設定されます。UL(k) を含めて連続した 2 つの内部ユーザ変数が必要ですので、確保してください。

ユーザパラメータ変数	上位ワード		下位ワード	
	上位バイト	下位バイト	上位バイト	下位バイト
UL(k)	年 BCD データ	月 BCD データ	日 BCD データ	曜日 BCD データ
UL(k+1)	時間 BCD データ	分 BCD データ	0 (パディング)	0 (パディング)

6. LCD デジタルオペレータの取り付けや取り外しは、インバータ本体の電源を遮断した状態で行ってください。故障の原因になります。
- LCD デジタルオペレータとの接続が断線する異常が発生すると、断線後最小 1 分 45 秒間は断線時の時刻を保持します。このため、断線後最小 1 分 45 秒間は rtcset on/off 命令を実行しても処理可能です。しかし、1 分 45 秒を超えると時刻機能の処理が完了せず、全プログラムはその命令で待機状態となります。LCD デジタルオペレータとの接続が再開した時点で、プログラムは次の処理へ移行します。全プログラムの待機状態を回避するには、プログラム起動時に 1 度だけ rtcset on 命令を起動し、常に指定した変数を時刻データに出力するプログラム構成としてください。プログラムでの時刻確認は指定した変数で行い、rtcset on/off 命令を実行しないでください。断線後 1 分 45 秒を超えた場合、指定した変数に全てゼロのデータが出力されます。
7. LCD デジタルオペレータの電池が消耗すると、時刻機能のデータが維持できなくなります。インバータの電源投入中は、制御回路電源で維持できますが、次の電源投入時に時刻機能は初期データ（2009 年 1 月 1 日 THU 00:00）に戻ります。



安全上の要点

DriveProgramming で時刻機能を使用して制御を行う場合、LCD デジタルオペレータの電池消耗で予期せぬ動作が起きるおそれがあります。

電池消耗は、時刻データが初期値に戻ったことで検出し、インバータまたはプログラムを停止させるなどの対策を行ってください。

LCD デジタルオペレータが取り外されたり、断線したりした場合、DriveProgramming は時刻機能で待機状態になります。

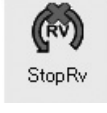
● 例

フローチャート	テキスト
<pre> graph TD 0([0]) --> entry([entry]) entry --> 1[1] 1 --> rtcset[rtcset off U(00)] rtcset --> 2([2]) 2 --> end([end]) </pre>	<pre> entry rtcset off U(00) end </pre>

このプログラム例を 2012 年 10 月 18 日（木曜日）午後 2 時 29 分に実行した場合、DriveProgramming エディタやデジタルオペレータには、U(00)、U(01)、および U(02) が以下のように表示されます。

ユーザパラメータ 変数	時刻機能 BCD データ (実際の 16 進数データ)	表示 (BCD データを 16 進数 として 10 進数に変換)	時刻機能 BCD データの意味
U(00)	1210	4624	「12」は 2012 年、「10」は 10 月
U(01)	1804	6148	「18」は 18 日、「04」は木曜日
U(02)	1429	5161	「14」は午後 2 時、「29」は 29 分

6-9 インバータ制御命令

Run FW		
命令	説明	引数
	インバータが正転運転を開始します。 この命令は、事前に FW := 1 として設定された func= 値の命令のショートカットです。	---
フォーマット		
フローチャート方式		テキスト言語方式
FW := 1 ↓		FW := 1
Run RV		
命令	説明	引数
	インバータが逆転運転を開始します。 この命令は、事前に RV := 1 として設定された func= 値の命令のショートカットです。	---
フォーマット		
フローチャート方式		テキスト言語方式
RV := 1 ↓		RV := 1
Stop FW		
命令	説明	引数
	インバータを正転運転から減速停止させます。 この命令は、事前に FW := 0 として設定された func= 値の命令のショートカットです。	---
フォーマット		
フローチャート方式		テキスト言語方式
FW := 0 ↓		FW := 0
Stop RV		
命令	説明	引数
	インバータを逆転運転から減速停止させます。 この命令は、事前に RV := 0 として設定された func= 値の命令のショートカットです。	---
フォーマット		
フローチャート方式		テキスト言語方式
RV := 0 ↓		RV := 0



バージョン関連情報

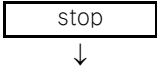
Stop FW 命令と Stop RV 命令は、CX-Drive Ver. 2.9 以降に追加されています。CX-Drive のソフトウェアをバージョンアップするか、代わりに下記の入出力制御命令 func = 値を使用してください。

停止 (func = 値)		
命令	説明	引数
 func=value	func = 値の命令を使用して、運転中のインバータを減速停止させます。 ☐ 「6-6 入出力制御命令 func = 値 (P.6-40)」を参照してください。 選択した機能>変数に<値>を代入します。	機能 ：以下の変数のどちらかを選択してください。 FW：Run FW で正転運転 RV：Run RV で逆転運転時 値 ：減速停止させるために、「0」（停止）を設定してください。
フォーマット		
フローチャート方式		テキスト言語方式
(Forward operation) (Reverse Operation) FW := 0 または RV := 0 ↓ ↓		(Forward operation) (Reverse Operation) FW := 0 RV := 0

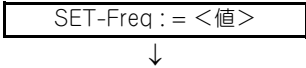


使用上の注意

- DriveProgramming 機能のプログラムが停止した場合、FW（正転）変数または RV（逆転）変数は状態を保持せず、ゼロにクリアします。
- FW（正転）変数または RV（逆転）変数は、インバータ本体の運転指令選択 (A002) で「01: 制御回路端子」を選択した場合だけ有効です。それ以外の設定では動作しません。
- インバータ電源投入直後、FW（正転）変数または RV（逆転）変数に「1」を設定してもその設定は無視し、正転運転または逆転運転を実行しません。いったん、「0」を設定し再度「1」を設定してください。この動作を回避するためには、電源投入後だけ wait 命令などを使用して 1 秒間の待ち時間をプログラムしてください。

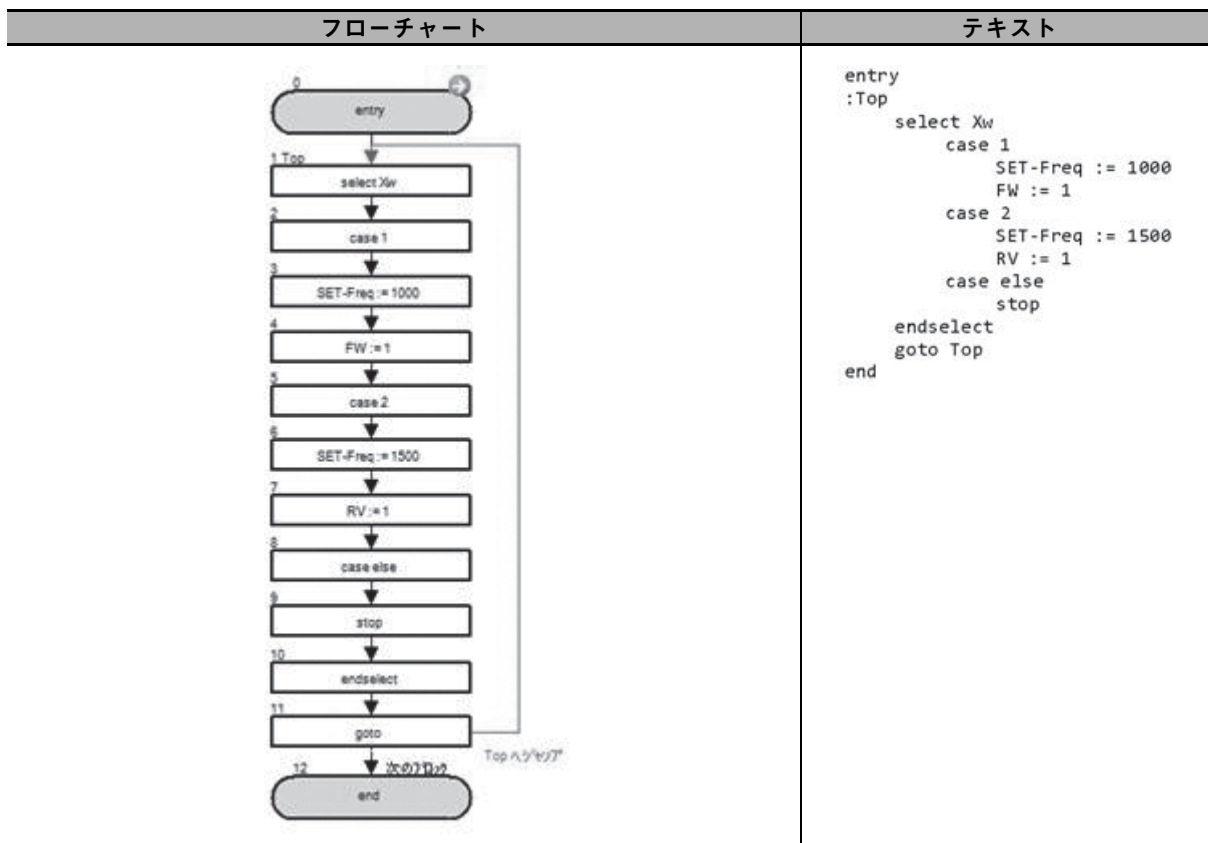
Stop		
命令	説明	引数
	運転中のインバータを減速停止させます。 また、インバータがトリップを検出している場合は、リセットとして機能します。	---
フォーマット		
フローチャート方式		テキスト言語方式
		stop

- (注) 1. stop 命令は、デジタルオペレータの STOP/RESET キーと同じ機能です。
 運転中のインバータを減速停止させます。また、インバータがトリップを検出している場合は、リセットとして機能します。
2. stop 命令がリセットとして動作しないようにするには、インバータ本体の停止キー選択 (b087) に「01: 無効」を設定してください。その設定で stop 命令はインバータを減速停止させますが、リセットとしては機能しません。ただし、デジタルオペレータの STOP/RESET キーは機能しなくなります。
3. stop 命令を使用すると、リセットとして機能したり、デジタルオペレータの STOP/RESET キーが機能しなくなりますので、前記の「停止 (func= 値)」で停止させることを推奨します。

Set Freq		
命令	説明	引数
	インバータの周波数を設定します。 この命令は、= (代入) の命令の左辺に周波数指令変数 SET-Freq を指定したものです。 単位: 0.01Hz	値: 任意の変数または定数 (0 ~ 40,000 の範囲)
フォーマット		
フローチャート方式		テキスト言語方式
		SET-Freq := <値>

- (注) 1. 実際にインバータが出力できる周波数は、始動周波数 (b082) ~ 最高周波数です。
 範囲外のデータを設定した場合は、以下のように動作します。
- ・ 始動周波数 (b082) 未満の設定
 - 【MX2・RX 共通機能】 0.00Hz として動作します。
 - 【RX だけの機能】 制御方式選択 (A044) が「04: 0SLV (0Hz 域センサレスベクトル)」、「05: V2 (センサ付きベクトル制御)」を選択している場合は、始動周波数 (b082) は無効で、指令した周波数を出力します。
 - ・ 最高周波数を越えた設定
 - 【MX2】 第 1 / 第 2 最高周波数 (A004/A204) に設定された周波数で周波数指令を制限します。
 - 【RX】 第 1 / 第 2 / 第 3 最高周波数 (A004/A204/A304) に設定された周波数で周波数指令を制限します。
2. DriveProgramming 機能のプログラムが停止した場合、周波数指令変数はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。

● 例

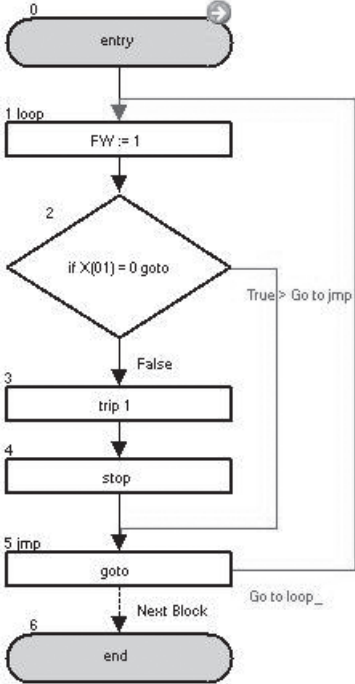


上記の例では、汎用入力接点 Xw が 1 であれば、モータは 10Hz で正転方向に回転します。汎用入力接点 Xw が 2 であれば、15Hz で逆方向に回転します。それ以外の値であれば、モータは停止します。

ブロック No.	動作
1	select 命令で Xw の値により以下の case 命令に分岐します。 Xw=1 のとき case 1、Xw=2 のとき case 2、それ以外のとき case else に分岐します。
2～4	Xw=1 のとき実行します。周波数指令変数を 10.00Hz にし、インバータを正転し、endselect 命令の次を実行します。
5～7	Xw=2 のとき実行します。周波数指令変数を 15.00Hz にし、インバータを逆転し、endselect 命令の次を実行します。
8～10	Xw が 1, 2 以外のとき実行します。インバータを停止し、endselect 命令の次を実行します。
11	無条件で 1: Top に分岐します。

Trip		
命令	説明	引数
	インバータをトリップさせます。 インバータはトリップを検出すると異常停止します。	値: 任意の変数または定数 (0 ~ 9 の範囲) 0 ~ 9 は E50 ~ E59 のアラームコードに対応します。
フォーマット		
フローチャート方式		テキスト言語方式
		trip <値>
(注) 1. 1 つのタスクでインバータをトリップさせても、DriveProgramming 機能の他のタスクは停止せず、継続します。 2. 任意の変数や定数（最大 127）で、10 以上を設定することも可能ですが、10 以上の設定でトリップさせても、全てエラーコードは E59 で表示されます。		

● 例

フローチャート	テキスト
	<pre>entry :loop_ FW := 1 if X(01) = 0 goto jmp trip 1 stop :jmp goto loop_ end</pre>

上記の例では、入力端子変数 X(01) が ON に設定されていると、インバータでユーザトリップが発生させます。

ブロック No.	動作
1	インバータを正転します。
2	X(01) が 0 のとき 5: jmp に分岐します。それ以外るとき、次に移行します。
3	インバータでユーザトリップ 1 を発生します。
4	インバータを停止します。
5	無条件で 1: loop_ に分岐します。

Accel		
命令	説明	引数
	インバータの加速時間を設定します。 この命令は、=（代入）の命令の左辺に加速時間変数 ACCEL を指定したものです。 単位：10ms	値：任意の変数または定数 （【MX2】 0 ～ 360,000 の範囲／【RX】 1 ～ 360,000 の範囲）
フォーマット		
フローチャート方式		テキスト言語方式
ACCEL := <値> ↓		ACCEL := <値>

Decel		
命令	説明	引数
	インバータの減速時間を設定します。 この命令は、=（代入）の命令の左辺に減速時間変数 DECEL を指定したものです。 単位：10ms	値：任意の変数または定数 （【MX2】 0 ～ 360,000 の範囲／【RX】 1 ～ 360,000 の範囲）
フォーマット		
フローチャート方式		テキスト言語方式
DECEL := <値> ↓		DECEL := <値>



使用上の注意

- ・加減速時間入力種別（P031）に「03: DriveProgramming」を設定した場合だけ加速時間変数 ACCEL と減速時間変数 DECEL が有効です。
- ・プログラムで加速時間変数 ACCEL と減速時間変数 DECEL に、データ範囲外のデータや、3G3RX シリーズ V1 タイプで「0」を設定すると、その設定データは反映せず、それ以前に設定したデータで動作します。
- ・DriveProgramming 機能のプログラムが停止した場合、加速時間変数と減速時間変数はプログラム停止前の状態を保持します。次にプログラムが起動されると、保持したデータから開始します。

● 例

フローチャート	テキスト
<pre>graph TD 0([entry]) --> 1{if X(00) = 1 then} 1 -- True --> 2[ACCEL := 100] 2 --> 3[DECEL := 2000] 3 --> 4[SET-Freq := 1000] 4 --> 5[FW := 1] 5 --> 6[endif] 6 --> 7[goto] 7 -- Go to loop_ --> 1 1 -- False --> 8[else] 8 --> 9[stop] 9 --> 10([end])</pre>	<pre>entry :loop_ if X(00) = 1 then accel := 100 decel := 2000 set-freq := 1000 fw := 1 else stop endif goto loop_ end</pre>

上記の例では、入力端子変数 X(00) が ON に設定されていると、加速時間を 1.00 秒、減速時間を 20.00 秒に設定します。

ブロック No.	動作
1 ～ 6	X(00) が 1 のとき、加速時間を 1.00 秒にし、減速時間を 20.00 秒にし、出力周波数を 10.00Hz にし、インバータを正転します。 X(00) が 1 以外のとき、else 命令の次に移行します。
7 ～ 8	インバータを停止します。
9	無条件で 1: loop_ に分岐します。

7

DriveProgramming で使用する パラメータの注意事項

この章では、DriveProgramming で使用するパラメータの注意事項を説明します。

7-1 設定順序で影響を受けるインバータパラメーター一覧	7-2
7-2 定格電流 (%) による影響を受けるパラメーター一覧	7-4
7-3 PID の有効／無効による影響を受けるパラメーター一覧	7-5

7-1 設定順序で影響を受けるインバータパラメータ一覧

パラメータの設定範囲が、他のパラメータの設定データの制約を受ける場合があります。

下記に代表的な例を示します。

詳細は、□『多機能型小型インバータ形 3G3MX2 シリーズ V1 タイプ ユーザーズマニュアル (SBCE-376)』、□『多機能型小型インバータ 形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル (SBCE-506)』、または □『高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル (SBCE-367)』の「10-2 ワーニング機能」を参照してください。

パラメータ No.	説明	データ
A003	第 1 基底周波数	30. ～第 1 最高周波数 [A004]
A004	第 1 最高周波数	【MX2】 第 1 基底周波数 / 【RX】 30. ～ 400.
A203	第 2 基底周波数	30. ～第 2 最高周波数 [A204]
A204	第 2 最高周波数	【MX2】 第 2 基底周波数 / 【RX】 30. ～ 400.
A303	【RX だけ】 第 3 基底周波数	30. ～第 3 最高周波数 [A304]
A304	【RX だけ】 第 3 最高周波数	30. ～ 400.
A013	FV スタート割合	0. ～ FV エンド割合 [A014]
A014	FV エンド割合	FV スタート割合 [A013] ～ 100.
A103	FI スタート割合	0. ～ FI エンド割合 [A104]
A104	FI エンド割合	FI スタート割合 [A103] ～ 100.
A113	【RX だけ】 FE スタート割合	-100. ～ FE エンド割合 [A114]
A114	【RX だけ】 FE エンド割合	FE スタート割合 [A113] ～ 100.
A020	第 1 多段速指令 0	0.00 / 始動周波数～第 1 最高周波数 [A004]
A220	第 2 多段速指令 0	0.00 / 始動周波数～第 2 最高周波数 [A204]
A320	【RX だけ】 第 3 多段速指令 0	0.00 / 始動周波数～第 3 最高周波数 [A304]
A021 ～ A035	多段速指令 1 ～ 15	0.00 / 始動周波数～第 1 最高周波数 [A004]
b015	自由電子サーマル周波数 1	0. ～自由電子サーマル周波数 2
b017	自由電子サーマル周波数 2	自由電子サーマル周波数 1 ～自由電子サーマル周波数 3
b019	自由電子サーマル周波数 3	自由電子サーマル周波数 2 ～ 400.
b051	瞬停・不足電圧減速停止 開始電圧	0.0 ～ 999.9/1000.
b052	瞬停・不足電圧減速停止 減速保留レベル	0.0 ～ 999.9/1000.
b060	ウインドウコンパレータ FV 上限レベル	上限レベルを設定します。 設定範囲：0. ～ 100. 設定下限値：下限レベル [b061]+ ヒステリシス幅 [b062] × 2
b061	ウインドウコンパレータ FV 下限レベル	下限レベルを設定します。 設定範囲：0. ～ 100. 設定上限値：上限レベル [b060]- ヒステリシス幅 [b062] × 2
b062	ウインドウコンパレータ FV ヒステリシス幅	上限・下限レベルに対して、ヒステリシス幅を設定します。 設定範囲：0. ～ 10. 設定上限値：(上限レベル [b060]- 下限レベル [b061]) / 2

パラメータ No.	説明	データ
b063	ウインドウコンパレータ FI 上限レベル	上限レベルを設定します。 設定範囲: 0. ~ 100. 設定下限値: 下限レベル [b064]+ ヒステリシス幅 [b065] × 2
b064	ウインドウコンパレータ FI 下限レベル	下限レベルを設定します。 設定範囲: 0. ~ 100. 設定上限値: 上限レベル [b063]- ヒステリシス幅 [b065] × 2
b065	ウインドウコンパレータ FI ヒステリシス幅	上限・下限レベルに対して、ヒステリシス幅を設定します。 設定範囲: 0. ~ 10. 設定上限値: (上限レベル [b063]- 下限レベル [b064]) / 2
b066	【RX だけ】ウインドウコンパレータ FE 上限レベル	上限レベルを設定します。 設定範囲: -100. ~ 100. 設定下限値: 下限レベル [b067]+ ヒステリシス幅 [b068] × 2
b067	【RX だけ】ウインドウコンパレータ FE 下限レベル	下限レベルを設定します。 設定範囲: -100. ~ 100. 設定上限値: 上限レベル [b066]- ヒステリシス幅 [b068] × 2
b068	【RX だけ】ウインドウコンパレータ FE ヒステリシス幅	上限・下限レベルに対して、ヒステリシス幅を設定します。 設定範囲: 0. ~ 10. 設定上限値: (上限レベル [b066]- 下限レベル [b067]) / 2
b100	自由 V / f 機能周波数 1	0. ~ 自由 V / f 機能周波数 2 [b102]
b102	自由 V / f 機能周波数 2	0. ~ 自由 V / f 機能周波数 3 [b104]
b104	自由 V / f 機能周波数 3	0. ~ 自由 V / f 機能周波数 4 [b106]
b106	自由 V / f 機能周波数 4	0. ~ 自由 V / f 機能周波数 5 [b108]
b108	自由 V / f 機能周波数 5	0. ~ 自由 V / f 機能周波数 6 [b110]
b110	自由 V / f 機能周波数 6	0. ~ 自由 V / f 機能周波数 7 [b112]
b112	自由 V / f 機能周波数 7	0. ~ 400.

7-2 定格電流 (%) による影響を受けるパラメータ一覧

電流を設定するパラメータは、その初期データや設定範囲がインバータ定格電流に影響されます。DriveProgramming 機能のプログラムから、設定する場合はインバータ定格電流に対する比率 (%) で設定してください。

下記に代表的な例を示します。

詳細は、□『多機能型小型インバータ形 3G3MX2 シリーズ V1 タイプユーザーズマニュアル (SBCE-376)』、□『多機能型小型インバータ 形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル (SBCE-506)』、または □『高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル (SBCE-367)』の「8-5 Modbus 通信レジスタ番号一覧」を参照してください。

パラメータ No.	説明	データ
b012	第 1 電子サーマルレベル	0.20 ~ 1.00 × 定格電流
b212	第 2 電子サーマルレベル	0.20 ~ 1.00 × 定格電流
b312	【RX だけ】第 3 電子サーマルレベル	0.20 ~ 1.00 × 定格電流
b022	過負荷制限レベル	<重負荷>
b222	【MX2 だけ】第 2 過負荷制限レベル	0.20 ~ 2.00 × 定格電流 (0.4 ~ 55kW)
b025	過負荷制限レベル 2	0.20 ~ 1.80 × 定格電流 (75 ~ 132kW)
b028	周波数引込再始動レベル	<軽負荷> 0.20 ~ 1.50 × 定格電流
b126	ブレーキ制御開放電流	0.20 ~ 2.00 × 定格電流 (0.4 ~ 55kW) 0.20 ~ 1.80 × 定格電流 (75 ~ 132kW)
C030	デジタル電流モニタ基準値	<重負荷> 0.20 ~ 2.00 × 定格電流 <軽負荷> 0.20 ~ 1.50 × 定格電流
C039	低電流検出レベル	<重負荷> 0.00 ~ 2.00 × 定格電流 (0.4 ~ 55kW) 0.00 ~ 1.80 × 定格電流 (75 ~ 132kW) <軽負荷> 0.00 ~ 1.50 × 定格電流
C041	過負荷予告レベル	0.0: 動作しません。 <重負荷> 0.00 ~ 2.00 × 定格電流 (0.4 ~ 55kW)
C241	【MX2 だけ】第 2 過負荷予告レベル	0.00 ~ 2.00 × 定格電流 (75 ~ 132kW) <軽負荷> 0.00 ~ 1.50 × 定格電流
C111	過負荷予告レベル 2	<重負荷> 0.00 ~ 2.00 × 定格電流 (0.4 ~ 55kW) 0.00 ~ 1.80 × 定格電流 (75 ~ 132kW) <軽負荷> 0.00 ~ 1.50 × 定格電流

7-3 PID の有効／無効による影響を受けるパラメータ一覧

設定値が A071/A075 の設定によってスケーリングされます。

パラメータ No.	説明	データ	
		A071=01 or 02	A071=00
A011	FV スタート周波数	変換後 = 設定値 × [A075]	設定値
A012	FV エンド周波数	変換後 = 設定値 × [A075]	設定値
A020	第 1 多段速指令 0	変換後 = 設定値 × [A075]	設定値
A021	多段速指令 1	変換後 = 設定値 × [A075]	設定値
A022	多段速指令 2	変換後 = 設定値 × [A075]	設定値
A023	多段速指令 3	変換後 = 設定値 × [A075]	設定値
A024	多段速指令 4	変換後 = 設定値 × [A075]	設定値
A025	多段速指令 5	変換後 = 設定値 × [A075]	設定値
A026	多段速指令 6	変換後 = 設定値 × [A075]	設定値
A027	多段速指令 7	変換後 = 設定値 × [A075]	設定値
A028	多段速指令 8	変換後 = 設定値 × [A075]	設定値
A029	多段速指令 9	変換後 = 設定値 × [A075]	設定値
A030	多段速指令 10	変換後 = 設定値 × [A075]	設定値
A031	多段速指令 11	変換後 = 設定値 × [A075]	設定値
A032	多段速指令 12	変換後 = 設定値 × [A075]	設定値
A033	多段速指令 13	変換後 = 設定値 × [A075]	設定値
A034	多段速指令 14	変換後 = 設定値 × [A075]	設定値
A035	多段速指令 15	変換後 = 設定値 × [A075]	設定値
A101	FI スタート周波数	変換後 = 設定値 × [A075]	設定値
A102	FI エンド周波数	変換後 = 設定値 × [A075]	設定値
A145	周波数加算量設定	変換後 = 設定値 × [A075]	設定値
A220	第 2 多段速指令 0	変換後 = 設定値 × [A075]	設定値
A320	【RX だけ】第 3 多段速指令 0	変換後 = 設定値 × [A075]	設定値
F001	出力周波数設定／モニタ	変換後 = 設定値 × [A075]	設定値
d004	PID フィードバック値モニタ	変換後 = フィードバック量 × [A075]	フィードバック量

8

異常とその処理

この章では、異常発生時のプログラムの動作、DriveProgramming 機能固有の異常と原因、その処理方法について説明しています。

8-1	トラブルシューティング	.8-2
8-1-1	異常発生時の DriveProgramming 動作	.8-2
8-1-2	異常リセット時の DriveProgramming 動作	.8-3
8-1-3	アラームコード一覧	.8-4

8-1 トラブルシューティング

異常発生時の DriveProgramming の動作や、DriveProgramming 機能固有のエラーコードとその処理方法について説明しています。

8-1-1 異常発生時の DriveProgramming 動作

DriveProgramming が動作している状態で、インバータがトリップを検出しても基本的に DriveProgramming は動作を継続します。ただし、DriveProgramming 関連のトリップ E43 ～ E45 を検出した場合は動作を停止します。また、「on trip goto」命令を使用すると、トリップ発生後に別の処理に分岐することが可能です。

on trip goto の有無	エラー内容		
	ユーザトリップ E50 ～ E59	DriveProgramming 関連 トリップ E43 ～ E45	その他のトリップ
なし	そのまま動作継続	プログラム停止	そのまま動作継続
あり	「on trip goto」命令実行後、指定ラベルの行に移行し、動作継続	プログラム停止	「on trip goto」命令実行後、指定ラベルの行に移行し、動作継続



安全上の要点

DriveProgramming 機能のプログラムが停止すると、DriveProgramming で制御している多機能出力はプログラム停止直前の状態を保持します。

このため、インバータの DriveProgramming 機能のプログラム停止を DriveProgramming 起動信号とアラーム（トリップ）信号で検出し、インバータ周辺機器を安全に停止させる配線を行ってください。

8-1-2 異常リセット時の DriveProgramming 動作

異常リセットした場合の DriveProgramming の動作は、そのリセット入力方法とリセット選択 (C102) の設定によって異なります。その動作状態を以下の表で示します。

● リセット入力方法

- DriveProgramming 機能を再起動するには多機能入力選択 (【MX2】 C001 ~ C007 / 【RX】 C001 ~ C008) に「18: RS (リセット)」を設定して、対応する入力端子を入力してください。
- デジタルオペレータや LCD デジタルオペレータ (オプション) の STOP/RESET キーでは DriveProgramming 機能を再起動できません。

● リセット選択 (C102)

- DriveProgramming 機能をリセット入力で再起動させるためには、リセット選択 (C102) の設定に、03 (トリップだけ解除) 以外を設定してください。03 (トリップだけ解除) の設定では再起動しません。
- リセット選択 (C102) に、00 (ON 時トリップ解除)、または 01 (OFF 時トリップ解除) を設定した場合、トリップしていない状態でリセット入力を入力しても、DriveProgramming 機能は再起動します。
- リセット選択 (C102) に、02 (トリップ時だけ有効) を設定すると、インバータがトリップを検出している場合だけ、再起動が可能です。インバータがトリップしていない状態では、リセット入力で再起動しません。

対象動作	プログラム状態	トリップ状態	制御端子のリセット			オペレータの STOP/RESET キー		
			C102 = 00, 01	C102 = 02	C102 = 03	C102 = 00, 01	C102 = 02	C102 = 03
DriveProgramming の動作 *1	動作中	正常時	再起動	動作継続	動作継続	動作継続	動作継続	動作継続
		トリップ中	再起動	再起動	動作継続	動作継続	動作継続	動作継続
	停止中	正常時	再起動	停止継続 *2	停止継続 *2	停止継続 *2	停止継続 *2	停止継続 *2
		トリップ中	再起動	再起動	停止継続 *2	停止継続 *2	停止継続 *2	停止継続 *2
インバータ本体のリセット動作		正常時	リセット	動作しない	動作しない	リセット	動作しない	動作しない
		トリップ中	リセット	リセット	リセット *3	リセット	リセット	リセット *3

*1. DriveProgramming 機能選択 (A017) で「01」を選択した場合、PRG 端子が ON した状態が表に示されています。PRG 端子が OFF の場合、プログラムが動作しないため全ての状態で「停止継続」となります。

*2. プログラムを再度起動するには、DriveProgramming 機能選択「00 (無効)」に設定したあと、「02 (常時有効)」に設定してください。また、A017=01 時は、PRG 端子を再投入でも再起動します。

*3. C102=03 を設定した場合、リセット動作はトリップだけを解除します。現在値などの内部演算データは初期化しません。

8-1-3 アラームコード一覧

DriveProgramming 機能固有のアラームコードと原因、その処理方法について説明しています。その他の異常については、□『多機能型小型インバータ形 3G3MX2 シリーズ V1 タイプユーザーズマニュアル (SBCE-376)』、□『多機能型小型インバータ 形 3G3MX2 シリーズ V2 タイプ ユーザーズマニュアル (SBCE-506)』、または □『高機能型汎用インバータ 形 3G3RX シリーズ V1 タイプ ユーザーズマニュアル (SBCE-367)』を参照してください。

アラームコード	アラーム (インバータのトリップの発生要因)	考えられる原因	チェック方法	処理方法
E43	無効な命令	プログラムがインバータにダウンロードされていないのに、PRG 端子が ON になりました。	プログラムをアップロードすることにより、実際にプログラムがインバータ内に存在するかチェックします。	プログラムを再作成して、インバータにダウンロードしてください。
E44	ネスティング カウント異常	サブルーチンが 8 階層を超えてネストされています。 「for next」命令が 8 階層を超えてネストされています。 「if」命令が 8 階層を超えてネストされています。	プログラムをアップロードして、ネスティングの階層数をチェックします。	ネスティングの階層数が 8 階層以下になるようにプログラムを修正してください。
E45	命令異常 1	「goto」命令のジャンプ先が、「for」またはその他のループを終了させる「next」命令になっている。 U (i) など参照している変数が見つかりません。 四則演算命令により、次の異常が発生しました。オーバーフロー、アンダーフロー、またはゼロ除算。 「ChgParam」命令にて、次の異常が発生しました。 ・存在しないパラメータへの変更。 ・書き込み値が設定範囲外 ・インバータの運転中に運転中変更不可のパラメータ値を変更 ・ソフトウェアロックによって更新が制限されているパラメータ値の変更 (ソフトウェアロックが有効である場合)	「goto」命令が、ループを終了させる命令にジャンプしていないかチェックします。 U (i) に指定している数値をチェックします。 $00 \leq i \leq 31$ プログラムをチェックし、オーバーフロー、アンダーフロー、またはゼロ除算が発生するような命令を確認します。 ・パラメータまたはパラメータに書き込まれる値を確認します。 ・インバータの運転中に異常が発生した場合は、パラメータがインバータ運転中に変更可能かどうかをチェックします。 ・ソフトウェアロック選択 (b031) の設定をチェックします。	「goto」命令のジャンプ先を修正してください。 U (i) の変数の値を修正するか、U (i) の変数のデータの範囲を制限してください。 四則演算命令によってオーバーフロー、アンダーフロー、またはゼロ除算が発生しないように、プログラムを修正してください。 ・パラメータまたはパラメータに書き込まれる値が設定範囲内になるように修正してください。 ・ソフトウェアロックを無効にしてください。 ・運転中変更可のパラメータの場合は、ソフトウェアロック選択 (b031) の設定を「10」に変更してインバータ運転中のパラメータの更新を有効に切り替えてください。

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

製品に関するお問い合わせ先



0120-919-066

携帯電話の場合、

055-982-5015 (有料) をご利用ください。

受付時間：9:00～17:00 (土・日・12/31～1/3を除く)

クイック

オムロン



オムロンFAクイックチャット

www.fa.omron.co.jp/contact/tech/chat/

技術相談員にチャットでお問い合わせいただけます。(I-Webメンバーズ限定)

受付時間：平日9:00～12:00 / 13:00～17:00 (土日祝日・年末年始・当社休業日を除く)

※受付時間、営業日は変更の可能性がございます。最新情報はリンク先をご確認ください。



その他のお問い合わせ：納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。



オムロン制御機器の最新情報をご覧ください。緊急時のご購入にもご利用ください。

www.fa.omron.co.jp

- 本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。
- 本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物 (又は技術) に該当するものを輸出 (又は非居住者に提供) する場合は同法に基づく輸出許可、承認 (又は役務取引許可) が必要です。

オムロン商品のご用命は