

变频器

RX2系列用

## DriveProgramming

---

用户手册

3G3RX2系列

CX-Drive

## 预告

- (1) 严禁擅自对本手册的部分或全部内容进行影印、复制或转载。
- (2) 因产品改良的关系，本手册记载的产品规格等有时可能会不经预告而变更。恕不事先通知。
- (3) 关于本手册的内容，我们做了严谨的编辑，如果发现在错误之处，烦请联系本公司分部或营业所。此时，请一并告知卷末记载的手册编号。

## 商标

- Sysmac为欧姆龙株式会社在日本和其它国家用于欧姆龙工厂自动化产品的商标或注册商标。
- Microsoft、Windows、Excel是美国Microsoft Corporation在美国及其他国家的注册商标或商标。
- EtherCAT® 是德国Beckhoff Automation GmbH的注册商标和专利技术，由该公司授权使用。
- ODVA、CIP、CompoNet、DeviceNet、EtherNet/IP是ODVA的商标。

除此之外，本文中记载的系统名称、产品名称为各公司的商标或注册商标。

# 前言

非常感谢您购买变频器 / 伺服支持工具 CX-Drive 和 3G3RX2 系列变频器。本用户手册对变频器用 DriveProgramming 的规格和操作方法进行说明。

使用时，请一并阅览高性能通用变频器 3G3RX2 系列用户手册 (SBCE-CN5-437) 及 CX-Drive 操作手册 (SBCE-CN5-375)。

## 阅读对象

本用户手册提供给下列阅读对象：

具有电工专业知识的人员 ( 合格的电气工程师或具有同等知识的人员 ) ；

- 引进 FA 设备的人员；
- 设计 FA 系统的人员；
- 安装或连接 FA 设备的人员；
- FA 现场管理人员。

## 注意

本用户手册介绍了使用 DriveProgramming 所需的信息。

使用前请仔细阅读本用户手册，并充分理解说明内容。

阅读后，请将其放于便于取用之处妥善保管，以便随时查阅。

务必确保将本用户手册交付至终端用户手中。

# 章节构成

本用户手册根据用户的用途由下列内容构成。

第 ### 章	概要
第 1 章 概要	本章对 DriveProgramming 的概要及系统构成进行介绍。
第 2 章 规格	本章对 DriveProgramming 的规格进行介绍。
第 3 章 DriveProgramming 的使用步骤	本章对 DriveProgramming 的使用步骤、相关参数和程序结构进行介绍。
第 4 章 DriveProgramming 编辑器	本章对 DriveProgramming 编辑器的启动、保存、读取及各部分的详情进行介绍。
第 5 章 DriveProgramming 的用户变量	本章对 DriveProgramming 的用户变量进行介绍。
第 6 章 DriveProgramming 指令	本章对 DriveProgramming 的指令进行介绍。
第 7 章 DriveProgramming 使用参数的注意事项	本章对 DriveProgramming 使用参数的注意事项进行介绍。
第 8 章 异常及其处理	本章对发生异常时的程序动作、DriveProgramming 功能固有的异常和原因及其处理方法进行介绍。



# 手册的阅读方法

## 页面构成

本用户手册各页面的构成和符号如下所述。

章标题 →

小节标题 →

项目标题 →

8 異常とその処理

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 機能固有の異常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**

インバータにトリップが発生した場合、トリップ内容と“on trip goto 命令”の有無により DriveProgramming の動作が異なります。動作の状態を、以下の表に示します。

エラー内容	ユーザートリップ	DriveProgramming 関連トリップ	その他のトリップ
on trip goto の有無	E50 ~ E59	E43 ~ E45	
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	指定レベルに飛び、動作継続	プログラム停止	指定レベルに飛び、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 端子 OFF ON (A017=01 側) するか、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、『高性能汎用インバータ 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原因)	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)

操作手順 →

表示操作手順。

备注、补充、参考 →

备注、补充、参考等各个项目使用图标表示。

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指令をドラッグ&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せん。

4 - 17
DriveProgramming ユーザーズマニュアル (SBCE-369)

小节标题 →

表示当前页的小节标题。

章编号 →

表示当前页的章编号。

项目标题 →

表示当前页的项目标题。

(注) 此页面是用于说明的样本。可能与实际内容有差异。

DriveProgramming 用户手册 (SBCE-CN5-440)

3

## 图标

本用户手册中使用的图标，意思如下。



### 安全要点

---

指出为了安全使用产品而必须实施或回避的注意事项。



### 使用注意事项

---

表示为了避免产品无法动作、误动作，或者对产品性能、功能产生不良影响而应当采取或避免的事项。



### 参考

---

项目内容请根据需要阅读。  
对应当了解的信息及使用时可作为参考的相关内容进行说明。



表示详细信息及相关信息的记载位置。

# 目录构成

---

		1
1	概要	2
2	规格	3
3	DriveProgramming 的使用步骤	4
4	DriveProgramming 编辑器	5
5	DriveProgramming 的用户变量	6
6	DriveProgramming 指令	7
7	DriveProgramming 使用参数的注意事项	8
8	异常及其处理	

# 目次

---

前言 .....	1
章节构成 .....	2
手册的阅读方法 .....	3
目录构成 .....	5
目次 .....	6
承诺事项 .....	9
安全注意事项 .....	10
安全要点 .....	13
法规与标准 .....	14
相关手册 .....	15
手册修订履历 .....	16

## 第 1 章 概要

---

1-1 DriveProgramming 的概要 .....	1-2
1-2 准备工作和系统构成 .....	1-4

## 第 2 章 规格

---

2-1 规格 .....	2-2
--------------	-----

## 第 3 章 DriveProgramming 的使用步骤

---

3-1 使用步骤 .....	3-2
3-2 DriveProgramming 的相关参数 .....	3-5
3-3 程序的结构 .....	3-12
3-3-1 任务的功能 .....	3-12
3-3-2 子程序的功能 .....	3-12
3-3-3 任务的处理 .....	3-12
3-3-4 DriveProgramming 功能的启动 / 停止和任务动作 .....	3-14
3-3-5 DriveProgramming 功能的重新启动 .....	3-16
3-3-6 发生跳闸时的任务动作 .....	3-17

## 第 4 章 DriveProgramming 编辑器

4-1	DriveProgramming 编辑器的启动.....	4-2
4-2	各部分的说明.....	4-6
4-2-1	DriveProgramming 编辑器.....	4-6
4-2-2	工具栏.....	4-6
4-2-3	DriveProgramming 区域.....	4-9
4-2-4	指令框窗口.....	4-13
4-2-5	用户参数窗口.....	4-14
4-2-6	属性窗口.....	4-15
4-2-7	输出窗口的错误列表标签.....	4-16
4-3	任务的添加、删除和重命名.....	4-17
4-4	子程序的插入、删除和调用.....	4-18
4-5	流程图程序的创建.....	4-19
4-6	文本程序的创建.....	4-20
4-7	已传送 (上传) 程序的编辑.....	4-21
4-8	程序的保存.....	4-22
4-9	程序的传送和核对.....	4-24
4-10	程序的执行 (DriveProgramming 功能选择).....	4-25
4-11	其它便捷功能.....	4-27

## 第 5 章 DriveProgramming 的用户变量

5-1	用户用的变量和用户参数.....	5-2
5-2	输入输出端子变量.....	5-5
5-3	定时器变量.....	5-10
5-4	变频器设定变量.....	5-12
5-5	变频器监控变量.....	5-14
5-6	输入变量.....	5-17
5-7	输出变量.....	5-20

## 第 6 章 DriveProgramming 指令

6-1	指令的种类.....	6-2
6-2	指令的格式.....	6-3
6-3	指令一览表.....	6-4
6-4	程序控制指令.....	6-10
6-5	算术运算指令和逻辑运算指令.....	6-23
6-6	输入输出控制指令.....	6-36
6-7	定时器控制指令.....	6-48
6-8	参数控制指令.....	6-54
6-9	变频器控制指令.....	6-58

## 第 7 章 DriveProgramming 使用参数的注意事项

---

7-1 受设定顺序影响的变频器参数一览表 .....	7-2
7-2 受额定电流 (%) 影响的参数一览表 .....	7-5
7-3 受 PID 有效 / 无效影响的参数一览表 .....	7-6

## 第 8 章 异常及其处理

---

8-1 故障诊断 .....	8-2
8-1-1 发生异常时的 DriveProgramming 动作 .....	8-2
8-1-2 异常复位时的 DriveProgramming 动作 .....	8-3
8-1-3 报警代码一览 .....	8-4

# 承诺事项

---

## ● 保修内容

### (1) 保修期

本软件的保修期为产品购买后或交付到指定场所后的 1 年之内。

“产品样本等”中另有记载的情况除外。

### (2) 保修范围

- 获得本软件使用授权的用户在上述保修期内发现计算机程序存在问题(与手册明显不一致)而退回本公司时,本公司将更换成记录了无问题的计算机程序的媒体。{或根据本公司选择,通过本公司主页下载无问题(与相应手册明显不一致)的计算机程序。}此外,发现本公司造成的计算机程序记录媒体不良而退回本公司时,本公司将免费更换成记录在合格媒体上的计算机程序。
- 本公司判断无法消除计算机程序的问题(与手册明显不一致)时,将退还用户支付的本软件购买费用。

## ● 责任免除

- 上一条规定的更换或退还购买费用是对本软件的全部保修责任,因软件问题对用户造成的直接、间接损失及波及效果造成的损失等,欧姆龙概不负责。
- 对于因本公司以外的第三方变更、改造本软件而产生的问题,以及因本软件的安装及其它使用导致本公司产品、安装的软件及所有计算机设备、计算机程序、网络、数据库受到 DDoS 攻击(分散型 DoS 攻击)、计算机病毒及其他技术上有害程序侵害和非法访问而造成的损失,本公司概不负责。
- 对于本公司以外的第三方根据本软件所开发的软件及其产生的后果,本公司概不负责。

## ● 本软件的用途

请勿将本软件用于本手册未记述的用途。

## ● 规格的变化

本软件的规格及附件可能会因为改善或其他原因而发生变更。

## ● 适用范围




以上内容只适用于在中国购买和使用的情况。在国外购买及使用的情况请向本公司营业人员咨询。

# 安全注意事项

## 安全信息的标识及其含义

为了安全使用 DriveProgramming，本用户手册使用下列标识及图标说明注意事项。这里所记载的注意事项均为与安全有重大相关的内容。请务必遵守。标识和含义如下所述。

## 警告标识的含义

 <b>危险</b>	操作不当可能导致操作人员重伤甚至死亡。此外还有可能引发重大财产损失。
 <b>警告</b>	操作不当时可能导致操作人员轻度、中度受伤，严重时可能致重伤或死亡。此外还有可能引发重大财产损失。
 <b>注意</b>	操作不当时，可能导致操作人员轻度、中度受伤，或者蒙受财产损失。

## 图标说明

	⊘ 图标表示禁止 ( 不得做 )。 在 ⊘ 中或旁边用图示或文字表示具体内容。 左图表示“禁止拆卸”。
	△ 图标表示危险、注意。 在 △ 中或旁边用图示或文字表示具体内容。 左图表示“谨防触电”。
	△ 图标表示危险、注意。 在 △ 中或旁边用图示或文字表示具体内容。 左图表示“不特定的一般危险”。
	△ 图标表示注意 ( 包含警告 )。 在 △ 中或用文字表示具体内容。 左图表示“注意高温”。
	● 图标表示强制事项 ( 必须遵守的事项 )。 在 ● 中或旁边用图示或文字表示具体内容。 左图表示“一般强制事项”。
	● 图标表示强制事项 ( 必须遵守的事项 )。 在 ● 中或用文字表示具体内容。 左图表示“务必接地”。












**警告**

<p>请确认输入电源 OFF 后正确接线。 否则可能会因触电导致重伤。</p>	
<p>请由电气工程专业人员进行接线作业。 否则可能会因触电导致重伤。</p>	
<p>变更接线、变更滑动开关 (SW1 ~ SW6)、拆装操作器及选装件类、更换冷却风扇时，请务必先切断变频器的输入电源。 否则可能会因触电导致重伤。</p>	
<p>接地端子请务必接地。 (200V 级：D 型接地、400V 级：C 型接地) 否则可能会因触电、起火导致重伤。</p>	
<p>在通电状态下以及切断电源后 15 分钟以内 *1*2，请勿拆下端子台盖板。 否则可能会因触电导致重伤。</p>	
<p>请勿用湿手操作操作器、开关类。 否则可能会因触电导致重伤。</p>	
<p>紧急断开输入功能动作的状态下，也不意味着主电源已切断。 请在关闭变频器的输入电源后再确认产品。 否则可能会因触电导致重伤。</p>	
<p>通电时及切断电源后不久，变频器散热片、制动电阻器、电机等可能仍处于高温，因此请勿触摸。 否则可能会导致烫伤。</p>	
<p>请在使用本软件的计算机上安装杀毒软件，降低计算机感染病毒的风险。同时请确保杀毒软件始终为最新版本。</p>	
<p>为了避免源于 OS 漏洞的安全风险，请确保使用本软件的计算机 OS 始终为最新版本。</p>	
<p>请及时升级本软件，确保其始终为最新版本，以起到追加功能、优化操作性、强化安全性的目的。</p>	
<p>对于通过本软件设定的用户名及密码，请进行妥善设置及管理，防止被他人非法利用。</p>	
<p>请在控制系统及设备的网络中导入防火墙（断开未使用的通信端口、限制通信主机），将其与 IT 网络隔开，并在防火墙内连接本软件的控制系統。</p>	
<p>如需通过本软件远程访问控制系统或设备，请使用虚拟专用网络（VPN）。</p>	

\*1. 10 分钟：3G3RX2-A2004 ~ A2220 及 3G3RX2-A4007 ~ A4220

\*2. 15 分钟：3G3RX2-A2300 ~ A2550 及 3G3RX2-A4300 ~ A4550、  
-B4750、-B4900、-B411K、-B413K

## ⚠ 注意

<p>不要直接将电阻与端子 (PD/+1、P/+、N/ - ) 连接。 否则可能导致轻度起火、发热、设备损坏。</p>	
<p>请安装确保安全的停止装置。 否则可能导致轻度损伤。 ※ 保持制动器并非为确保安全的停止装置。</p>	
<p>制动电阻器及再生制动单元产生的热量可能会引起中度烫伤。 请务必使用指定的制动电阻器及再生制动单元，使用制动电阻器时，请安装监视电阻器温度的热敏继电器。 此外，请组入在制动电阻器及再生制动单元异常过热时切断变频器电源的时序。</p>	
<p>产品内部有高压部分，短路会导致产品损坏以及其他物品损坏。在设置及布线时，可以通过设置外盖等，防止切割粉屑及导线碎屑等金属进入产品内部。</p>	
<p>负载的布线短路可能引起物品损坏。 作为安全对策，请在变频器电源侧设置与变频器功率相当的布线用断路器 (MCCB) 等。</p>	
<p>请勿进行分解、修理或改装。 否则可能导致受伤。</p>	
<p>调试、调整、维护、更换时误设定参数，可能会导致意外动作。</p>	
<p>输出端子功能的输出过程中 DriveProgramming 停止时，会保持输出状态，因此请采取停止周边设备等安全措施。</p>	

# 安全要点

---

## 关于运行、调整

- DriveProgramming 使用时刻功能进行控制时，可能会因 LCD 操作器的电池消耗而导致意外动作。电池消耗请使用 [E042]RTC 错误进行检测，并采取停止变频器及程序等措施。拆下 LCD 操作器或断线时， DriveProgramming 根据时刻功能会进入待机状态。
- 使用 DriveProgramming 时，请在确认程序数据已正常下载后再开始动作。
- 本产品可实现从低速到高速的设定，请在充分确认所使用的电机设备的容许范围后再运行。

## 法规与标准

---

出口 ( 或向非居住者提供 ) 本产品中符合外汇及外国贸易管理法规定的出口许可、批准对象货物 ( 或技术 ) 要求的产品时，须依照该法获得出口许可、批准 ( 或劳务交易许可 )。

# 相关手册

---


操作该产品时，需掌握连接设备的信息。  
相关产品的信息请参阅下列手册。

名称	手册编号
CX-Drive 操作手册	SBCE-CN5-375
高性能型通用变频器 RX2 系列用户手册	SBCE-CN5-437



## 参考

---

关于变频器的动作，请参阅本体的  《高性能型通用变频器 RX2 用户手册 (SBCE-CN5-437)》。

---

# 手册修订履历

手册修订记号使用字母附加在封面和封底的 Man.No. 的末尾。

示例

Man.No. **SBCE-CN5-440B**

↑ 修订记号

修订记号	修订日	修订内容和修订页
A	2019 年 3 月	初版
B	2023 年 4 月	增加安全对策的说明

# 1

## 概要

本章对 DriveProgramming 的概要及系统构成进行介绍。

---

1-1 DriveProgramming 的概要 .....	1-2
1-2 准备工作和系统构成 .....	1-4

# 1-1 DriveProgramming 的概要

DriveProgramming 是变频器内置的简易时序功能。

创建时序程序及确认状态时，使用变频器 / 伺服支持工具 CX-Drive。

将创建的程序传送 ( 下载 ) 至 3G3RX2 系列变频器后，变频器单体即可执行简单的时序控制。

## DriveProgramming 的特点

DriveProgramming 具有下列特点。

- DriveProgramming 可使用流程图方式和文本语言方式进行编程。
- 程序可划分成任务单位 ( 最多 5 个 ) 进行创建。
- 可并列处理 5 个任务。
- 可通过设定输入端子，从外部执行用户程序。
- 可将输入端子及输出端子分配至程序进行使用。
- 频率设定及加减速时间等需现场调整的程序常数 ( 变量 ) 在指定了用户参数 (UE-10 ~ UE-73、UF-02 ~ UF-32) 时，无需连接计算机即可使用 LCD 操作器进行变更。
- 用户程序保存在变频器本体内部的 EEPROM 中，因此可在变频器接通电源后立即启动程序。
- LCD 操作器备有时刻功能。因此，可创建使用时刻功能的程序。使用时刻功能时，需要另售的电池 (CR2032、3V)。



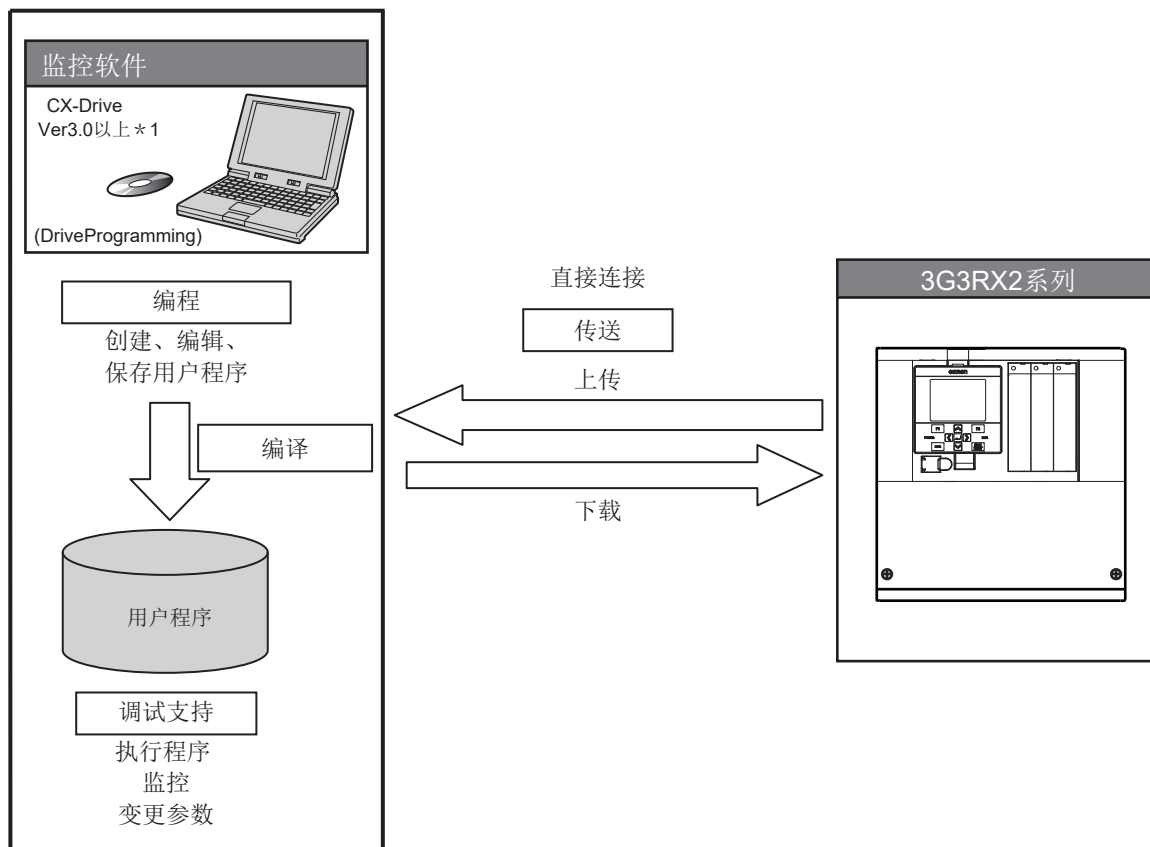
### 安全要点

DriveProgramming 使用时刻功能进行控制时，可能会因 LCD 操作器的电池消耗而导致意外动作。电池消耗请使用 [E042]RTC 错误进行检测，并采取停止变频器及程序等措施。拆下 LCD 操作器或断线时，DriveProgramming 根据时刻功能会进入待机状态。

CX-Drive 的 DriveProgramming 编辑器可执行的主要功能如下所述。

功能	说明
编程	支持创建、编辑、保存、读取、打印用户程序。
编译	检查用户程序并生成中间代码。
传送	将用户程序下载至变频器。 从变频器上传用户程序。
调试支持	执行、停止程序。 可确认变频器状态监控等。



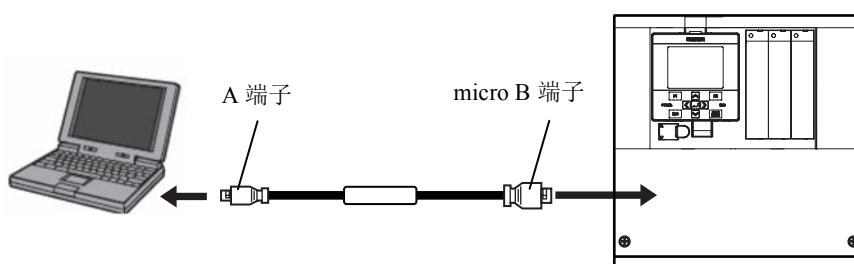


\*1. Ver3.0 以上：3G3RX2 对应版

## 1-2 准备工作和系统构成

使用 CX-Drive 的 DriveProgramming 功能创建程序，并使用 3G3RX2 系列变频器执行该程序时，需进行以下准备。

- 3G3RX2 系列变频器本体
- Windows 个人电脑  
(支持的操作系统请参阅《CX-One 用户手册》(SBCA-CN5-346)。)
- CX-Drive 需使用 Ver.3.0 以上版本。  
(CX-Drive 随附于 FA 整合工具包 CX-One。)
- 请准备以下电缆作为 PC-变频器连接电缆。  
需使用市售的 USB 电缆 (micro-B)。



### 参考

关于 CX-Drive 的安装及版本升级方法，请参阅  《CX-Drive 的操作手册》(SBCE-CN5-375)。

# 2

## 规格

本章对 DriveProgramming 的规格进行介绍。

---

2-1 规格 .....	2-2
--------------	-----

## 2-1 规格

DriveProgramming 的相关规格如下所述。

项目	规格		
程序规格	编程语言	流程图方式和文本语言方式	
	输入设备	Windows 个人电脑 (支持的操作系统请参阅 □ 《CX-One 用户手册》(SBCA-CN5-346)。)	
	程序容量	1 个任务 最多 1,024 步 5 个任务最多共 7,680 字节	
	编程支持功能	变频器 / 伺服支持工具 CX-Drive 支持的功能 <ul style="list-style-type: none"> <li>• 编辑和显示</li> <li>• 编译程序 (检查程序语法)</li> <li>• 下载、上传和全部清除程序</li> </ul>	
	执行形式	<ul style="list-style-type: none"> <li>• 解释器方式</li> <li>• 执行周期: 从每步 1ms 或 2ms 中选择 (参数 [UE-01])</li> <li>• 支持子程序调用 (最多可嵌套 8 层)</li> <li>• 多任务: 最多 5 个任务</li> </ul>	
输入输出功能	外部输入	DriveProgramming 启动	EzSQ 功能选择 使用 (UE-02) 选择 通过输入端子 PRG 进行启动 / 停止或常时启动
		输入端子	X(00) ~ X(10) / 最多 11 点
		频率指令输入 (模拟输入)	XA(0): 0 ~ 10V / 0 ~ 20mA (Ai1 端子) XA(1): 0 ~ 10V / 0 ~ 20mA (Ai2 端子) XA(2): -10 ~ 10V (Ai3 端子)
	外部输出	输出端子 (含继电器接点)	Y(00) ~ Y(06) / 最多 7 点
		监视器输出 (模拟输出)	YA(0): FM 端子 YA(1): Ao1 端子 YA(2): Ao2 端子
指令	程序控制指令	<ul style="list-style-type: none"> <li>• 循环 (for)</li> <li>• 无条件分支 (goto)</li> <li>• 时间控制 (wait)</li> <li>• 条件分支 (if then, ifs then, select case, until, while)</li> <li>• 子程序 (call, sub)</li> <li>• 其他 (entry, end, inc, dec, cont)</li> </ul>	
	运算指令	<ul style="list-style-type: none"> <li>• 四则运算 (+, -, *, /)</li> <li>• 求余 (mod)、代入 (=)</li> <li>• 绝对值 (abs)</li> <li>• 逻辑运算 (or, and, xor, not)</li> </ul>	
	输入输出控制	<ul style="list-style-type: none"> <li>• 输入输出端子功能 (位输入、字输入、位输出、字输出)</li> <li>• 变频器输入端子的读取</li> <li>• 变频器输出端子的读取、写入</li> </ul>	
	定时器控制	基于定时器的延迟动作及定时器计数器的控制	
	参数控制	指定参数 No. 的设定数据的变更	
	变频器控制	<ul style="list-style-type: none"> <li>• 正转运行和反转运行的执行和停止</li> <li>• 通过 DriveProgramming 产生跳闸 (E050 ~ E059 / 10 点)</li> <li>• 频率指令、加减速时间的设定</li> </ul>	

项目	规格	
功能变量	用户参数变量	U(00) ~ U(63) / 64 点
	内部用户变量	UL(00) ~ UL(15) / 16 点
	频率指令变量	SET-Freq
	加速时间变量	ACCEL
	减速时间变量	DECEL
	变频器监控变量	可将变频器的监控功能用作变量。 FM, Iout, Dir, PID-FB, F-CNV, Tmon, Vout, Power, RUN-Time, ON-Time, PlsCnt, POS, STATUS, DCV, ERR CNT, ERR(1) ~ ERR(10), RETRY(1) ~ RETRY(10)
	输入变量	可使用变频器的输入端子选择 (CA-01 ~ CA-11) 的可选功能。 FW, RV, CF1, CF2, CF3, CF4, SF1, SF2, SF3, SF4, SF5, SF6, SF7, ADD, SCHG, STA, STP, F/R, AHD, FUP, FDN, UDC, F-OP, SET, RS, JG, DB, 2CH, FRS, EXT, USP, CS, SFT, BOK, OLR, KHC, OKHC, PID, PIDC, PID2, PIDC2, PID3, PIDC3, PID4, PIDC4, SVC1, SVC2, SVC3, SVC4, PRO, PIO1, PIO2, SLEP, WAKE, TL, TRQ1, TRQ2, PPI, CAS, SON, FOC, ATR, TBS, ORT, LAC, PCLR, STAT, PUP, PDN, CP1, CP2, CP3, CP4, ORL, ORG, FOT, ROT, SPD, PSET, MI1, MI2, MI3, MI4, MI5, MI6, MI7, MI8, MI9, MI10, MI11, PCC, ECOM, PRG, HLD, REN, DISP, PLA, PLB, EMF, COK, PLZ, TCH
	输出变量	可使用变频器的输出端子选择 (CC-01 ~ CC-07) 的可选功能。 RUN, FA1, FA2, FA3, FA4, FA5, IRDY, FWR, RVR, FREF, REF, SETM, OPO, AL, MJA, OTQ, IP, UV, TRQ, IPS, RNT, ONT, THM, THC, WAC, WAF, FR, OHF, LOC, LOC2, OL, OL2, BRK, BER, CON, ZS, DSE, PDD, POK, PCMP, OD, FBV, OD2, FBV2, NDc, Ai1Dc, Ai2Dc, Ai3Dc, WCAi1, WCAi2, WCAi3, LOG1, LOG2, LOG3, LOG4, LOG5, LOG6, LOG7, MO1, MO2, MO3, MO4, MO5, MO6, MO7, EMFC, EMBP, LBK, OVS, AC0, AC1, AC2, AC3, OD3, FBV3, OD4, FBV4, SSE
	输入端子变量	X(00) ~ X(10) / 11 点
	输出端子变量	Y(00) ~ Y(06) / 7 点
	内部用户接点	UB(0) ~ UB(15) / 16 点
	定时器输出接点	TD(0) ~ TD(15) / 16 点
	定时器计数器变量	TC (0) ~ TC(15) / 16 点
	模拟输入端子变量	XA(0): 0 ~ 10V / 0 ~ 20mA (Ai1 端子) XA(1): 0 ~ 10V / 0 ~ 20mA (Ai2 端子) XA(2): -10 ~ 10V (Ai3 端子)
	模拟输出端子变量	YA(0) ~ YA(2) (分配至 FM、Ao1、Ao2 端子)



# 3

## DriveProgramming 的使用步骤

3

本章对 DriveProgramming 的使用步骤、相关参数和程序结构进行介绍。

---

3-1	使用步骤 .....	3-2
3-2	DriveProgramming 的相关参数 .....	3-5
3-3	程序的结构 .....	3-12
3-3-1	任务的功能 .....	3-12
3-3-2	子程序的功能 .....	3-12
3-3-3	任务的处理 .....	3-12
3-3-4	DriveProgramming 功能的启动 / 停止和任务动作 .....	3-14
3-3-5	DriveProgramming 功能的重新启动 .....	3-16
3-3-6	发生跳闸时的任务动作 .....	3-17

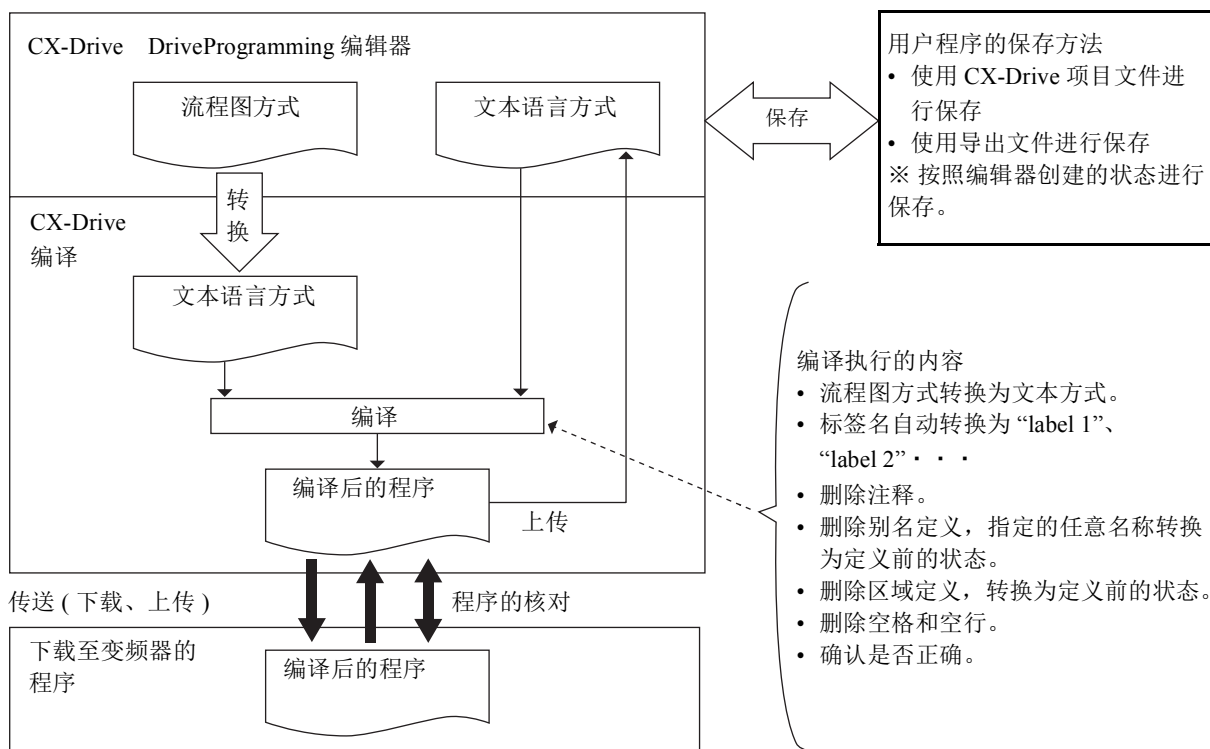
## 3-1 使用步骤

DriveProgramming 的程序创建至执行的流程如下所述。

项目	参照页
程序的创建	P. 3-3
程序的编译	P. 3-3
程序的下载	P. 3-4
DriveProgramming 功能的选择	P. 3-4
程序的启动	P. 3-4

DriveProgramming 的程序使用变频器 / 伺服支持工具 CX-Drive 的 DriveProgramming 编辑器进行创建。创建程序至传送 ( 下载 ) 到变频器的流程如下所述。

操作等详情请参阅 □ 「第 4 章 DriveProgramming 编辑器」。





## 程序的创建

用户程序可使用流程图方式和文本语言方式进行创建。也可根据任务及子程序，从这 2 种方式中选择编程方式。

使用 CX-Drive 中的 DriveProgramming 编辑器输入用户程序。

DriveProgramming 编辑器由 DriveProgramming 区域、指令框窗口、用户参数窗口、属性窗口、输出窗口构成。

- DriveProgramming 区域可使用流程图方式或文本语言方式创建程序。
- 指令框窗口按类别集中显示支持的指令块。
- 用户参数窗口可设定程序开始执行时的参数。
- 属性窗口可编辑流程图方式当前所选区块的属性。
- 输出窗口显示编译后的编译异常及警告。

## 程序的编译

DriveProgramming 区域创建的程序将在执行编译后创建最终的“编译后的程序”，并传送（下载）至变频器。

编译时，将确认程序的内容正确与否及程序语法、参数输入限制、最大步数等，存在不允许的输入时将中断编译，并显示错误信息。

编译时还会执行下述操作，并创建最终的“编译后的程序”。因此，将变频器中保存过的程序传送（上传）至 CX-Drive 时，将读取“编译后的程序”，动作相同但格式和内容有所变更。

- 流程图方式转换为文本语言方式。
- 程序内设定了任意名称的标签名自动转换为“label 1”、“label 2” . . .
- 删除程序中的注释。
- 删除别名定义，指定的任意名称转换为定义前的状态。
- 删除区域定义，转换为定义前的状态。
- 删除程序内的空格和空行。
- 确认是否正确。



### 使用注意事项

- 通过传送（下载）至变频器或转换程序时的编译，将删除设定的注释、别名定义、区域定义的内容。  
需保存这些内容时，请保存并管理传送（下载）至变频器前或转换程序前的程序。  
可通过保存 CX-Drive 的整个项目文件或保存程序的导出文件保存程序。
- 程序的核对会比较“编译后的程序”与变频器内部的程序，因此不会核对注释、别名定义、区域定义等。

## 程序的下载

将编译后的程序下载至变频器，并保存至变频器的 EEPROM。

即使不使用工具 (CX-Drive)，接通电源后也可启动 EEPROM 中保存的程序。

## DriveProgramming 功能的选择

使用 DriveProgramming 功能时，请将 EzSQ 功能选择 (UE-02) 设为有效 (01 或 02)。运行过程中，也可变更 EzSQ 功能选择 (UE-02)。

参数 No.	功能名称	数据	说明
UE-02	EzSQ 功能选择	00: 无效	将 DriveProgramming 功能设为无效。不执行程序。程序执行过程中将数据变更为“00: 无效”时，程序将停止。
		01: 有效 [PRG] 端子	输入“99: PRG”功能中设定的输入端子 <sup>*1</sup> 时，将启动 DriveProgramming 功能的程序。
		02: 始终有效	在变频器电源接通后，自动启动 DriveProgramming 的程序。程序停止过程中将数据变更为“02: 有效”时，程序启动。

\*1. 输入端子以 1 ~ 9、A、B 为对象。

## 程序的启动

EzSQ 功能选择 (UE-02) 设定为“01: [PRG] 端子”时，请在输入端子选择的任意一者中设定“99: PRG”。将设定了 PRG 端子的输入端子设为“ON”时，程序将启动。PRG 端子为“ON”时将继续执行，“OFF”时停止。

EzSQ 功能选择 (UE-02) 设定为“02”时，将在设定后立即执行程序。此外，在下次接通电源时将自动执行程序。

程序启动之后，完成一系列处理到达“end”指令时，必须重新启动程序才能执行。

若需重复程序，创建时请在程序内构建循环避免到达“end”指令。

下载的 DriveProgramming 功能的程序会保存至变频器本体的 EEPROM，因此在下载程序后，即使不使用支持工具也可启动程序。

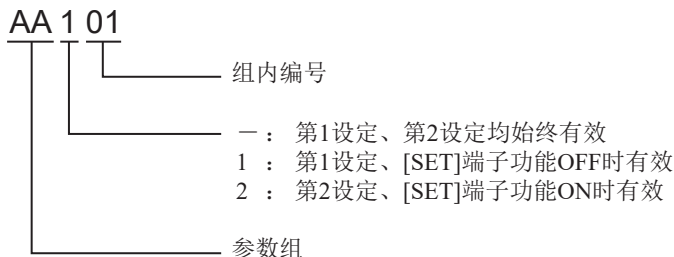
参数 No.	功能名称	数据	说明
CA-01 ~ CA-11	输入端子 1 ~ 9、A、B	99: PRG EzSQ 程序开始	EzSQ 功能选择 (UE-02) 设定为“01: [PRG] 端子”时，进行了该设定的输入端子将启动程序。

## 3-2 DriveProgramming 的相关参数

下面对 DriveProgramming 的相关变频器的参数进行说明。

参数编号如下所述。

有第 1 设定、第 2 设定时，本章使用第 1 设定进行说明。第 2 设定的设定值及动作与第 1 设定相同。



### 程序的初始化

对下载至变频器本体的 DriveProgramming 功能的程序进行初始化时，请在出厂初始化范围选择 (Ub-01) 中选择“04: 跳闸履历 + 参数 + Drive Programming”，使用初始化执行选择 (Ub-05) 执行初始化。

参数 No.	功能名称	数据	说明
Ub-01	出厂初始化选择	00: 无效 01: 跳闸履历 02: 参数初始化 03: 跳闸履历+参数 04: 跳闸履历+参数+ Drive Programming 05: 端子功能以外 06: 通信功能以外 07: 端子 & 通信功能以外 08: 仅 Drive Programming	包含 DriveProgramming 的程序在内进行初始化时，选择“04”范围。

## 运行指令选择的设定

通过 DriveProgramming 功能的程序控制运行指令时，使用 FW(正转)变量和 RV(反转)变量。使用 FW 变量或 RV 变量时，请务必在运行指令选择中设定“00: [FW]/[RV] 端子”。

参数 No.	功能名称	数据	说明
AA111/ AA211	运行指令选择	00: [FW]/[RV] 端子 01: 3 线 02: LCD 操作器的 RUN 键 03: (RS485) 04: 选装件 1 05: 选装件 2 06: 选装件 3	使用变量 FW/RV 执行运行 / 停止时，将运行指令设定为 00: [FW]/[RV] 端子。
CA-01 ~ CA-11	输入端子 1 ~ 9、A、 B	99: PRG EzSQ 程序开始	EzSQ 功能选择 (UE-02) 设定为“01: [PRG] 端子 (基于 PRG 端子的启动 / 停止)”时，进行了该设定的输入端子将启动程序。

## 变频器输入输出功能的设定

DriveProgramming 可将变频器的输入输出功能 (输入输出端子、模拟输入输出) 用作程序的输入输出功能。

使用输入输出功能时，需根据用途设定各输入输出功能。

下面对 DriveProgramming 功能的输入输出功能的设定方法进行说明。

通过以下设定，可通过 DriveProgramming 功能的功能变量自由控制输入输出功能。

作为 DriveProgramming 以外的输入输出功能进行设定时，请参阅 □ 《高功能型通用变频器 RX2 系列用户手册 (SBCE-CN5-437)》。

参数 No.	功能名称	数据	说明
AA101	第 1 主速指令选择	14: 程序功能	在变量 SET-Freq 中设定频率时设定 14: 程序功能。
CA-01 ~ CA-11	输入端子 1 ~ 9、A、 B	86 ~ 96: M11 ~ M111	PRG: EzSQ 功能 PRG 端子 *2
CC-01 ~ CC-07	输出端子 11 ~ 15 选 择	69 ~ 75: MO1 ~ MO7	MO1 ~ MO7: 通用输出 1 ~ 7
Cd-03	FM 端子输出选择	Cd-03	YA(0): 通用模拟输出
Cd-04	Ao1 端子输出选择	Cd-04	YA(1): 通用模拟输出
Cd-05	Ao2 端子输出选择	Cd-05	YA(2): 通用模拟输出
AC-01	加减速时间输入种类	04: Drive Programming	变量 ACCEL、DECEL 中设定加速 / 减速时间时，设定主速指令 [AA101]14: 程序功能。

\*1. 将主速指令选择 [AA101] 及运行指令选择 [AA111] 设定成上述以外的值，也可使用 LCD 操作器及模拟信号输入速度 / 运行指令。

\*2. 仅在将 EzSQ 功能选择设定为 PRG 端子启动 ([UE-02]=01) 时，请将 099[PRG] 分配至输入端子。



### 使用注意事项

---

- 输入端子 1 ~ 9、A、B 选择中选择 MI1 ~ MI11(通用输入 1 ~ 11) 时，输入端子 1 ~ 9、A、B 动作选择 (CA-21 ~ CA-31) 仍有效。
  - 输出端子 11 ~ 15 选择、继电器输出端子 16、继电器输出端子 AL 选择中选择 MO1 ~ MO7 (通用输出 1 ~ 7) 时，输出端子 11 ~ 15 接点选择 (CC-11 ~ CC-15)、继电器输出 16、继电器输出 AL 的接点选择 (CC-16、CC-17) 仍有效。
  - DriveProgramming 将模拟输入输出功能分配至 XA(0) ~ XA(2) 和 YA(0) ~ YA(2)。无论 AA101、Cd-03 ~ Cd-05 的设定如何，均可在程序中使用各功能变量监控模拟输入输出状态。
  - DriveProgramming 的程序中，输入端子功能选择 (CA-01 ~ CA-11) 未选择 MI1 ~ MI11(通用输入 1 ~ 11) 时，无法监控输入信号的状态。此外，输出端子功能选择 (CC-01 ~ CC-07) 未选择 MO1 ~ MO7(通用输出 1 ~ 7) 时，无法监控输出信号的状态。
-

## DriveProgramming 的监控功能

备有以下监控 DriveProgramming 状态的功能。

参数 No.	功能名称	数据	说明
db-01	程序下载监控	00: 无程序 01: 有程序	00: 未下载程序。 01: 已下载程序。
db-02	程序编号监控	0 ~ 9999	显示已传送的程序编号。
db-03 ~ db-07	程序计数器	1 ~ 1024	监控各任务 (Task1 ~ Task5) 执行中的行号。
db-08	用户监控 0	-2147483647 ~ 2147483647	监控程序内输出至 Umon(00) ~ Umon(04) 的数据。 db-08 ← Umon(00)
db-10	用户监控 1	-2147483647 ~ 2147483647	db-10 ← Umon(01)
db-12	用户监控 2	-2147483647 ~ 2147483647	db-12 ← Umon(02)
db-14	用户监控 3	-2147483647 ~ 2147483647	db-14 ← Umon(03)
db-16	用户监控 4	-2147483647 ~ 2147483647	db-16 ← Umon(04)
db-18	模拟输出监控 YA0	0 ~ 10000	监控程序内输出至 YA(00) ~ YA(02) 的数据。 YA0 db-18 ← YA(00)
db-19	模拟输出监控 YA1	0 ~ 10000	YA1 db-19 ← YA(01)
db-20	模拟输出监控 YA2	0 ~ 10000	YA2 db-20 ← YA(02)

## DriveProgramming 用户参数

作为 DriveProgramming 的用户参数，备有 64 个无符号的 1 字变量和 16 个带符号的 2 字变量。请用于程序初始数据、调整参数、运算结果的保存等。

参数 No.	功能名称	数据	说明
UE-10	EzSQ 用户参数 U(00)	0 ~ 65535	<ul style="list-style-type: none"> <li>与功能变量 U(00) ~ U(63) 对应的用户参数。</li> <li>可使用 LCD 操作器进行变更。变更后将保存至 EEPROM。</li> </ul>
UE-11	EzSQ 用户参数 U(01)	0 ~ 65535	
UE-12	EzSQ 用户参数 U(02)	0 ~ 65535	
UE-13	EzSQ 用户参数 U(03)	0 ~ 65535	
UE-14	EzSQ 用户参数 U(04)	0 ~ 65535	
UE-15	EzSQ 用户参数 U(05)	0 ~ 65535	
UE-16	EzSQ 用户参数 U(06)	0 ~ 65535	
UE-17	EzSQ 用户参数 U(07)	0 ~ 65535	
UE-18	EzSQ 用户参数 U(08)	0 ~ 65535	
UE-19	EzSQ 用户参数 U(09)	0 ~ 65535	
UE-20	EzSQ 用户参数 U(10)	0 ~ 65535	
UE-21	EzSQ 用户参数 U(11)	0 ~ 65535	
UE-22	EzSQ 用户参数 U(12)	0 ~ 65535	
UE-23	EzSQ 用户参数 U(13)	0 ~ 65535	
UE-24	EzSQ 用户参数 U(14)	0 ~ 65535	
UE-25	EzSQ 用户参数 U(15)	0 ~ 65535	
UE-26	EzSQ 用户参数 U(16)	0 ~ 65535	
UE-27	EzSQ 用户参数 U(17)	0 ~ 65535	
UE-28	EzSQ 用户参数 U(18)	0 ~ 65535	
UE-29	EzSQ 用户参数 U(19)	0 ~ 65535	
UE-30	EzSQ 用户参数 U(20)	0 ~ 65535	
UE-31	EzSQ 用户参数 U(21)	0 ~ 65535	
UE-32	EzSQ 用户参数 U(22)	0 ~ 65535	
UE-33	EzSQ 用户参数 U(23)	0 ~ 65535	
UE-34	EzSQ 用户参数 U(24)	0 ~ 65535	

参数 No.	功能名称	数据	说明
UE-35	EzSQ 用户参数 U(25)	0 ~ 65535	<ul style="list-style-type: none"> <li>与功能变量 U(00) ~ U(63) 对应的用户参数。</li> <li>可使用 LCD 操作器进行变更。变更后将保存至 EEPROM。</li> </ul>
UE-36	EzSQ 用户参数 U(26)	0 ~ 65535	
UE-37	EzSQ 用户参数 U(27)	0 ~ 65535	
UE-38	EzSQ 用户参数 U(28)	0 ~ 65535	
UE-39	EzSQ 用户参数 U(29)	0 ~ 65535	
UE-40	EzSQ 用户参数 U(30)	0 ~ 65535	
UE-41	EzSQ 用户参数 U(31)	0 ~ 65535	
UE-42	EzSQ 用户参数 U(32)	0 ~ 65535	
UE-43	EzSQ 用户参数 U(33)	0 ~ 65535	
UE-44	EzSQ 用户参数 U(34)	0 ~ 65535	
UE-45	EzSQ 用户参数 U(35)	0 ~ 65535	
UE-46	EzSQ 用户参数 U(36)	0 ~ 65535	
UE-47	EzSQ 用户参数 U(37)	0 ~ 65535	
UE-48	EzSQ 用户参数 U(38)	0 ~ 65535	
UE-49	EzSQ 用户参数 U(39)	0 ~ 65535	
UE-50	EzSQ 用户参数 U(40)	0 ~ 65535	
UE-51	EzSQ 用户参数 U(41)	0 ~ 65535	
UE-52	EzSQ 用户参数 U(42)	0 ~ 65535	
UE-53	EzSQ 用户参数 U(43)	0 ~ 65535	
UE-54	EzSQ 用户参数 U(44)	0 ~ 65535	
UE-55	EzSQ 用户参数 U(45)	0 ~ 65535	
UE-56	EzSQ 用户参数 U(46)	0 ~ 65535	
UE-57	EzSQ 用户参数 U(47)	0 ~ 65535	
UE-58	EzSQ 用户参数 U(48)	0 ~ 65535	
UE-59	EzSQ 用户参数 U(49)	0 ~ 65535	
UE-60	EzSQ 用户参数 U(50)	0 ~ 65535	
UE-61	EzSQ 用户参数 U(51)	0 ~ 65535	
UE-62	EzSQ 用户参数 U(52)	0 ~ 65535	



参数 No.	功能名称	数据	说明	
UE-63	EzSQ 用户参数 U(53)	0 ~ 65535	<ul style="list-style-type: none"> <li>与功能变量 U(00) ~ U(63) 对应的用户参数。</li> <li>可使用 LCD 操作器进行变更。变更后将保存至 EEPROM。</li> </ul>	
UE-64	EzSQ 用户参数 U(54)	0 ~ 65535		
UE-65	EzSQ 用户参数 U(55)	0 ~ 65535		
UE-66	EzSQ 用户参数 U(56)	0 ~ 65535		
UE-67	EzSQ 用户参数 U(57)	0 ~ 65535		
UE-68	EzSQ 用户参数 U(58)	0 ~ 65535		
UE-69	EzSQ 用户参数 U(59)	0 ~ 65535		
UE-70	EzSQ 用户参数 U(60)	0 ~ 65535		
UE-71	EzSQ 用户参数 U(61)	0 ~ 65535		
UE-72	EzSQ 用户参数 U(62)	0 ~ 65535		
UE-73	EzSQ 用户参数 U(63)	0 ~ 65535		
UF-02	EzSQ 用户参数	-2147483647 ~ 2147483647		<ul style="list-style-type: none"> <li>与功能变量 UL(00) ~ UL(15) 对应的用户参数。</li> <li>可使用 LCD 操作器进行变更。变更后将保存至 EEPROM。</li> </ul>
UF-03	UL(00)			
UF-04	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-05	UL(01)			
UF-06	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-07	UL(02)			
UF-08	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-09	UL(03)			
UF-10	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-11	UL(04)			
UF-12	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-13	UL(05)			
UF-14	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-15	UL(06)			
UF-16	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-17	UL(07)			
UF-18	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-19	UL(08)			
UF-20	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-21	UL(09)			
UF-22	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-23	UL(10)			
UF-24	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-25	UL(11)			
UF-26	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-27	UL(12)			
UF-28	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-29	UL(13)			
UF-30	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-31	UL(14)			
UF-32	EzSQ 用户参数	-2147483647 ~ 2147483647		
UF-33	UL(15)			

## 3-3 程序的结构

3G3RX2 系列的 DriveProgramming 最多可创建 5 个任务。

创建的各任务进行并列处理。

通过分配 1 个应用并将处理分配至多个任务，可调整各自的执行条件及执行顺序等。并且，各任务为并列处理，也可缩短处理时间。

### 3-3-1 任务的功能

任务是指 DriveProgramming 执行的程序单位。

3G3RX2 系列最多可创建 5 个任务，因此 5 个任务合起来为整个程序。

- 各任务同时启动。
- 按照 1ms 或 2ms 的处理时间 ( 通过 UE-01 选择 )，从任务 1 至任务 5 依次执行 1 个指令 (1 行“编译后的程序”)。
- 各任务的用户参数等功能变量均为通用。任务间的信息传送请使用用户参数及内部用户接点等。
- 在 1ms 或 2ms 处理时间的最后，会将各任务的运算结果反映至变频器动作及外部输出，并获取变频器状态及外部输入的状态。
- 执行“end”指令时，该任务结束，进入下一启动等待状态。

### 3-3-2 子程序的功能

子程序是指仅在调用时执行的单独程序处理。

使用子程序时，可将 1 个任务中多次执行的处理集中至 1 个程序，或可集中成在其它程序中可再次使用的程序。

DriveProgramming 功能请对各任务分别创建子程序。各任务之间无法通用。

子程序内可调用另一个子程序 ( 嵌套 )。

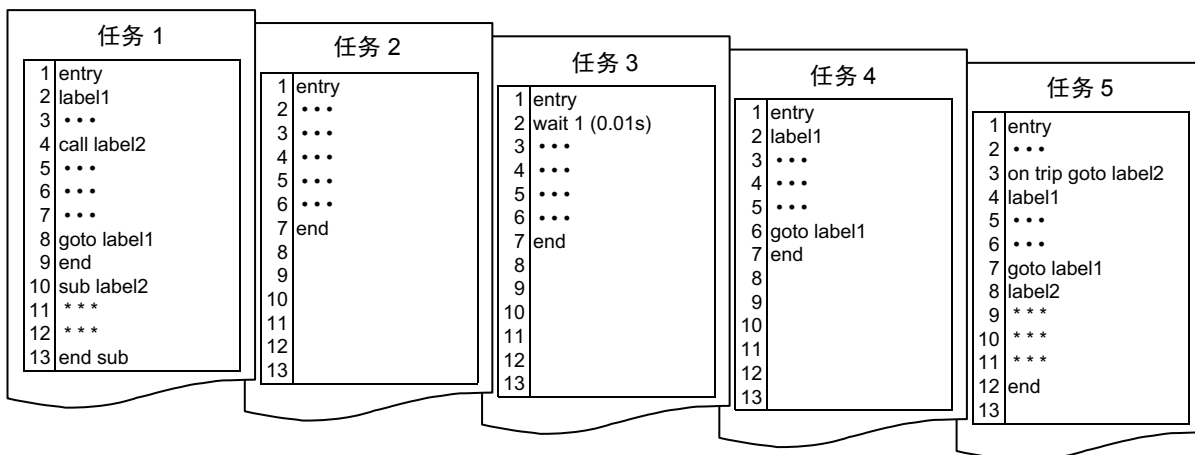
但子程序的嵌套最多为 8 层。

### 3-3-3 任务的处理

最多可同时启动 5 个任务，在 1ms 或 2ms 的处理时间 ( 通过 UE-01 选择 ) 内执行各任务的 1 个指令 (1 行“编译后的程序”)。各程序处理的流程如下图“编译后的程序”所述。

- 各任务同时启动。
- 按照 1ms 或 2ms 的处理时间 ( 通过 UE-01 选择 )，从任务 1 至任务 5 依次执行 1 个指令 (1 行“编译后的程序”)。
- 各任务的用户参数等功能变量均为通用。
- 在 1ms 或 2ms 处理时间 ( 通过 UE-01 选择 ) 的最后，会将各任务的运算结果反映至变频器动作及外部输出，并获取变频器状态及外部输入的状态。
- 如任务 1 所示，使用“call”指令调用子程序时，将从下一个处理起执行子程序的行。
- 如任务 2 所示，执行“end”指令时，该任务结束，进入下一启动等待状态。
- 如任务 3 所示，执行“wait”指令时，将重复执行直至条件成立。
- 如任务 4 所示，使用“goto”指令创建循环时，将一直重复该任务处理。
- 如任务 5 所示，执行 1 次“on trip goto”指令时，该内容将保存至变频器内部。发生跳闸时，将立即分支至“goto”位置。

● 编译后的程序 (例)



● 程序处理的流程

处理时间 [ms]	任务 1		任务 2		任务 3		任务 4		任务 5	
	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 【发生跳闸】
10	10	sub label2	5	...	2	wait 1(0.01s)	5	...	5	label2
12	11	***	6	...	2	wait 1(0.01s)	6	goto label1	7	***
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	...		

发了变频器跳闸，因此执行 Label2。

程序计数器 (db-03)

### 3-3-4 DriveProgramming 功能的启动 / 停止和任务动作

DriveProgramming 功能的程序启动 / 停止方法通过 EzSQ 功能 (UE-02) 进行设定。

#### ● 启动

- EzSQ 功能选择 (UE-02) 为“01: PRG 端子”时:  
输入端子中设定的 PRG 端子 ON
- EzSQ 功能选择 (UE-02) 为“02: 常时”时:  
变频器接通电源

#### ● 停止

- EzSQ 功能选择 (UE-02) 为“01: PRG 端子”时:  
输入端子中设定的 PRG 端子 OFF
- EzSQ 功能选择 (UE-02) 为“02: 常时”时:  
变频器电源切断

此外, 无论 UE-02 的设定如何, 检出程序异常的 E043 ~ E045 跳闸时和所有任务执行了“end”指令时, DriveProgramming 功能的程序将停止。



#### 安全要点

DriveProgramming 功能的程序停止时, 由 DriveProgramming 控制的输出端子将保持程序停止前的状态。

因此, 请采用通过 DriveProgramming 启动信号和报警 (跳闸) 信号检出变频器 DriveProgramming 功能的程序停止, 安全停止变频器周边设备的接线。



#### 使用注意事项

DriveProgramming 功能的程序停止时各功能变量的状态如下所述, 请识别各状态采取以下措施。

- 输出端子 (实际的变频器输出端子) 将保持程序停止前的状态。  
但未设定 DriveProgramming 功能的通用输出 MO1 ~ MO6 的输出将作为通常的变频器输出端子进行动作。
- 在频率指令、加减速时间、模拟输出中设定了 DriveProgramming 功能时, 各设定值将保持程序停止前的状态。
- 用户参数变量、内部用户变量、内部用户接点将保持程序停止前的状态。
- 输入端子及模拟输入等实际的变频器输入端子状态会一直更新状态, 而不保持输入状态。
- 输出变量 (RUN、FA1、AL 等功能位) 及变频器监控变量会根据变频器的状态一直更新状态, 而不保持状态。
- 只有 DriveProgramming 的输入变量 (FW、RV、CF1 等功能位) 和定时器变量会在程序停止的同时清除, 所有状态清零。

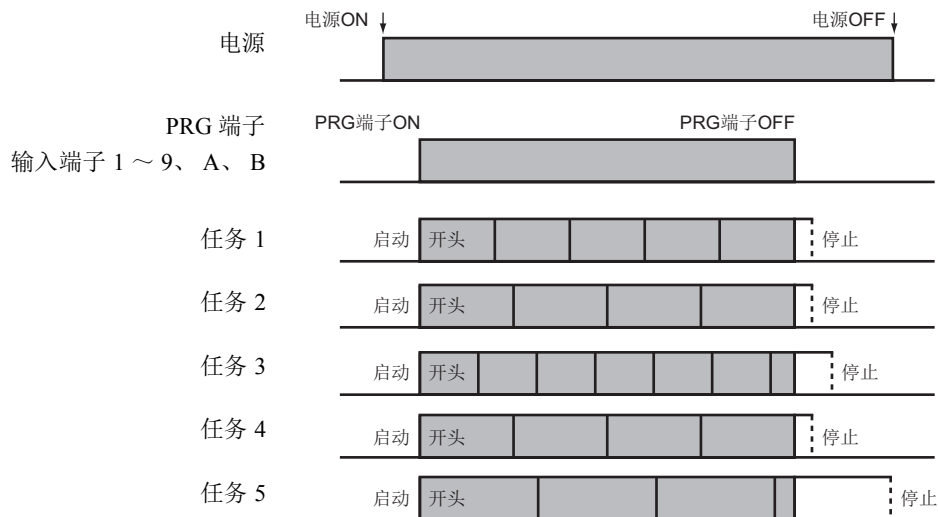
使用 CX-Drive 启动 / 停止 DriveProgramming 功能时, CX-Drive 将变更变频器的参数 EzSQ 功能选择 (UE-02)。

使用 CX-Drive 启动 / 停止 DriveProgramming 功能后, 请采取以下措施。

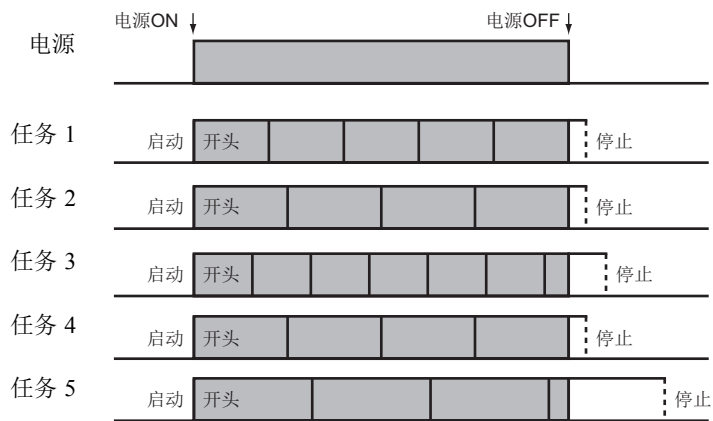
- 请在重新接通变频器的电源后, 将 UE-02 恢复成 EEPROM 中保存的值。
- 在重新接通变频器的电源前, 请勿执行以下 EEPROM 保存操作。
  - 从 CX-Drive 传送 (下载) 部分参数。
  - 通过 Modbus 通信或通信发出“enter”指令。

启动 DriveProgramming 功能的程序时，5 个任务将同时启动。  
 在任务内使用“goto”指令等构建了循环时，该任务启动后，将一直重复该循环。  
 停止 DriveProgramming 功能的程序时，5 个任务将同时停止。  
 在电机运行时停止了程序时，将根据停止方式选择 (AA115) 的设定停止电机 ( 减速停止或自由运行停止 )。  
 各设定的任务动作如下所述。

● EzSQ 功能选择 (UE-02) 设定为“01：PRG 端子”时



● EzSQ 功能选择 (UE-02) 设定为“02：常时”时



### 3-3-5 DriveProgramming 功能的重新启动

已停止的程序可通过以下操作重新启动。重新启动时，各任务将从程序开头起同时启动。

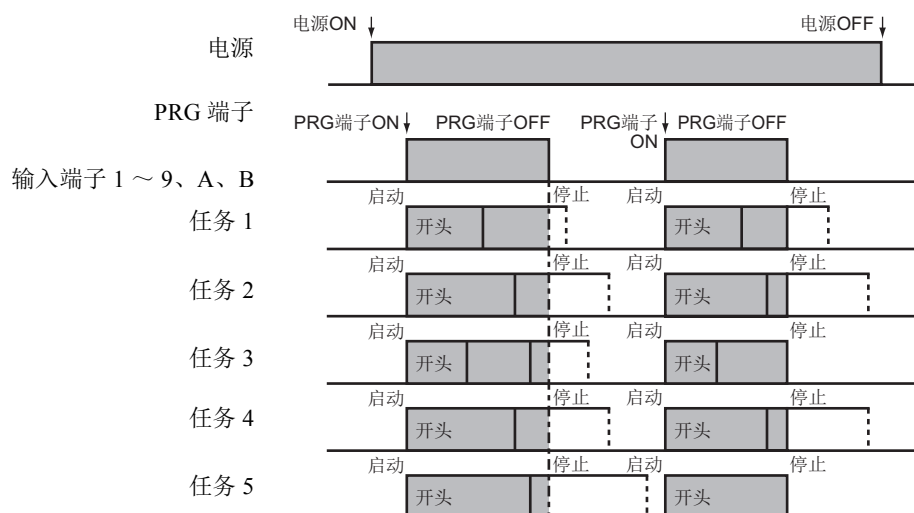
- EzSQ 功能选择 (UE-02) 为“01: PRG 端子”时：PRG 端子为 ON 状态时，从控制电路端子进行复位输入或对 PRG 端子进行重新输入。
- EzSQ 功能选择 (UE-02) 为“02: 常时”时：从控制电路端子进行复位输入或重新接通变频器电源。



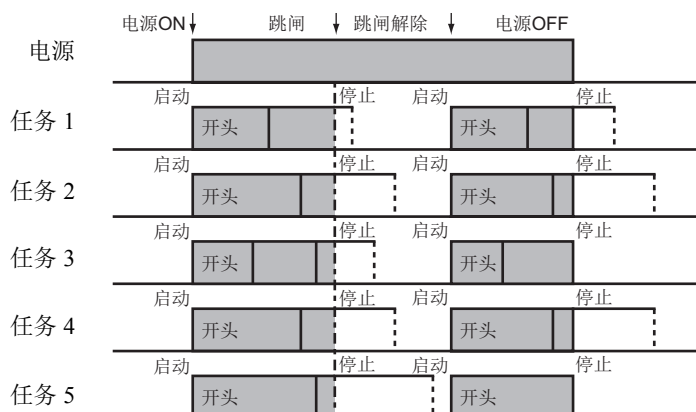
#### 使用注意事项

- 无法使用 LCD 操作器的 STOP/RESET 键重新启动 DriveProgramming 功能。在输入端子 1 ~ 9、A、B 中设定“28: RS(复位)”，并输入对应的端子。
- 复位选择 (CA-72) 设定为“00: ON 时复位”或“01: OFF 时复位”时，在未跳闸的状态下输入复位输入，也会重新启动 DriveProgramming 功能。

#### ● EzSQ 功能选择 (UE-02) 设定为“01: PRG 端子”时的重新启动



#### ● EzSQ 功能选择 (UE-02) 设定为“02: 常时”时的跳闸重新启动 (无“on trip goto”指令)



### 3-3-6 发生跳闸时的任务动作

DriveProgramming 动作的状态下，即使变频器检出跳闸 DriveProgramming 一般仍会继续动作。但检出 DriveProgramming 相关的跳闸 E043 ~ E045 时，将停止动作。

此外，使用“on trip goto”指令时，在发生跳闸后可能会分支成其它处理。

on trip goto 的有无	错误内容		
	用户跳闸 E050 ~ E059	DriveProgramming 相关跳 闸 E043 ~ E045	其它跳闸
无	继续动作	程序停止	继续动作
有	执行“on trip goto”指令 后，将切换至指定标签行 继续动作	程序停止	执行“on trip goto”指令 后，将切换至指定标签行 继续动作

跳闸 E043 ~ E045 的详细内容请参阅 □ 「第 8 章 异常及其处理」。





# 4

## DriveProgramming 编辑器

本章对 DriveProgramming 编辑器的启动、保存、读取及各部分的详情进行介绍。

4

4-1	DriveProgramming 编辑器的启动 .....	4-2
4-2	各部分的说明 .....	4-6
4-2-1	DriveProgramming 编辑器 .....	4-6
4-2-2	工具栏 .....	4-6
4-2-3	DriveProgramming 区域 .....	4-9
4-2-4	指令框窗口 .....	4-13
4-2-5	用户参数窗口 .....	4-14
4-2-6	属性窗口 .....	4-15
4-2-7	输出窗口的错误列表标签 .....	4-16
4-3	任务的添加、删除和重命名 .....	4-17
4-4	子程序的插入、删除和调用 .....	4-18
4-5	流程图程序的创建 .....	4-19
4-6	文本程序的创建 .....	4-20
4-7	已传送 (上传) 程序的编辑 .....	4-21
4-8	程序的保存 .....	4-22
4-9	程序的传送和核对 .....	4-24
4-10	程序的执行 (DriveProgramming 功能选择) .....	4-25
4-11	其它便捷功能 .....	4-27

## 4-1 DriveProgramming 编辑器的启动

DriveProgramming 使用变频器 / 伺服支持工具 CX-Drive 的 DriveProgramming 编辑器创建程序。下面对 CX-Drive 的 DriveProgramming 编辑器的构成和直至启动 DriveProgramming 编辑器的操作进行说明。

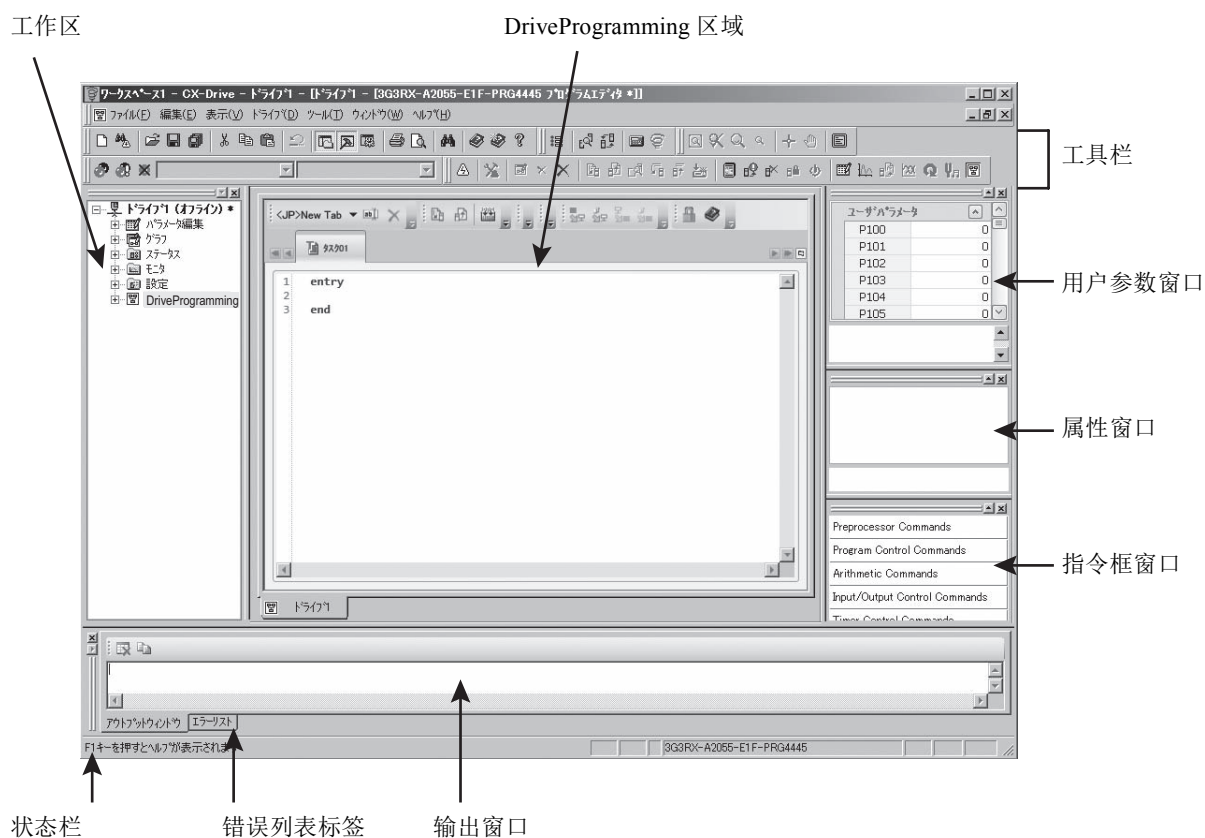


### 使用注意事项

首次启动 CX-Drive 的 DriveProgramming 编辑器时需使用密码。  
请向本公司销售负责人员进行确认。

### CX-Drive 和 DriveProgramming 编辑器的画面构成

DriveProgramming 编辑器作为变频器 / 伺服支持工具 CX-Drive 的一项功能而配备。启动 DriveProgramming 编辑器时，CX-Drive 的画面构成如下所示。



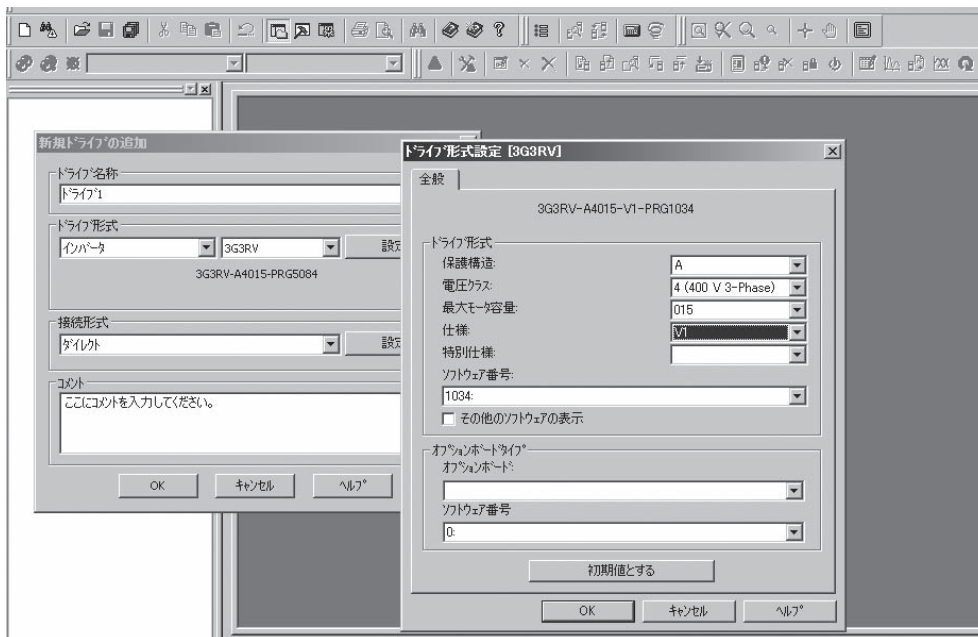
## CX-Drive 的启动

启动 CX-Drive 时, 请从 Windows 的 [开始] 中选择 [所有程序], 然后在 [OMRON] - [CX-One] - [CX-Drive] 文件夹中点击 CX-Drive 图标。

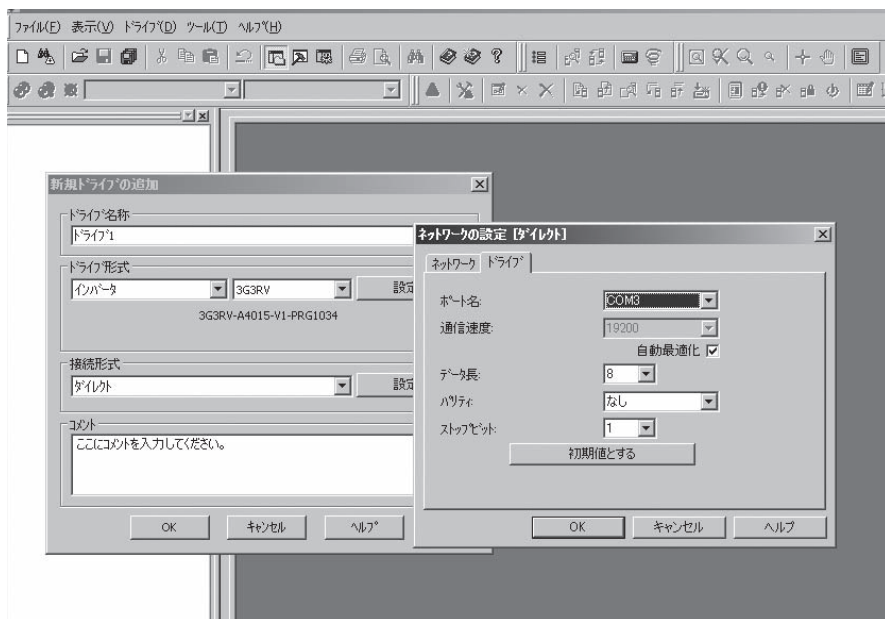
CX-Drive 启动后, 请从 CX-Drive 的菜单中选择 [文件] 并点击 [新建], 创建新的 CX-Drive 文件。显示 [新建驱动器] 窗口。

请从 [驱动器类型] 的变频器系列中选择 3G3RX2 系列, 然后按下右侧的 [设定] 按钮。

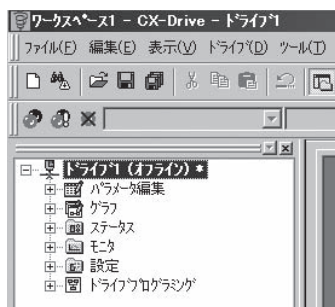
请在驱动器类型设定窗口中设定变频器的 [保护结构] / [电压等级] / [最大电机容量]。设定后请按下 OK 按钮, 关闭驱动器类型设定窗口。



然后在新建驱动器窗口中设定 CX-Drive 与变频器的连接方式。  
 在 [连接方式] 中选择直接后, 请按下右侧的 [设定] 按钮。  
 请在 [驱动器] 标签中, 设定 CX-Drive 使用的计算机的端口名。




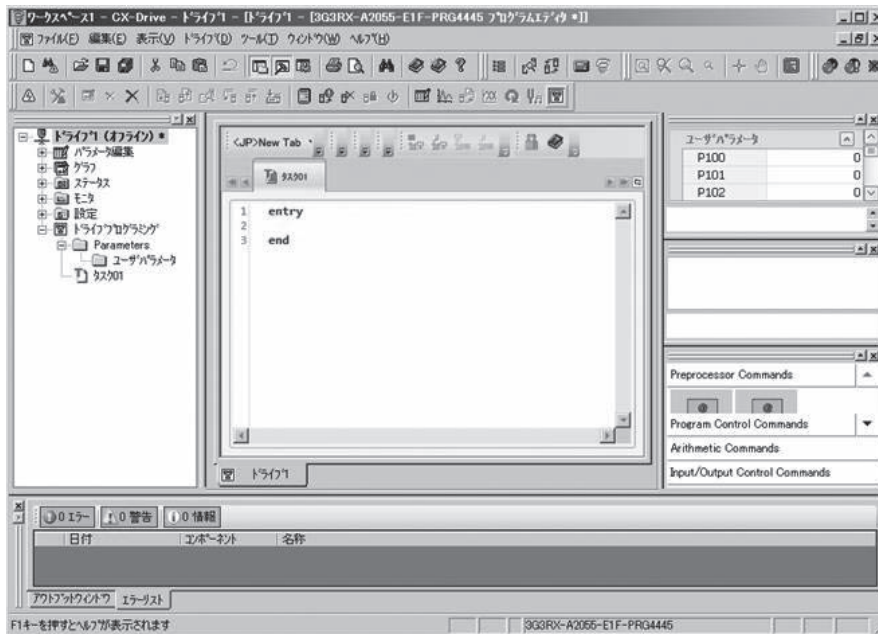
设定后按下 [OK] 关闭所有窗口后, 工作区中将登录新的项目。



## DriveProgramming 编辑器的启动

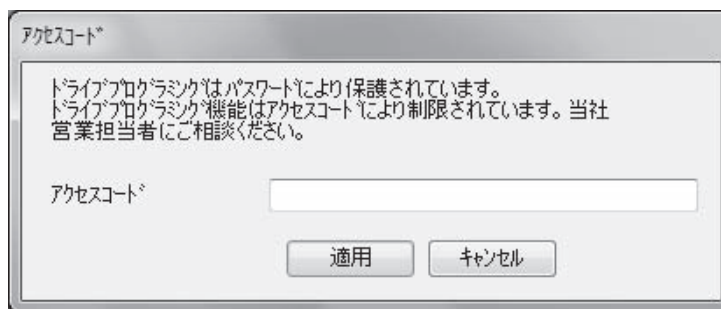
显示 DriveProgramming 编辑器有以下 3 种方法。

- 双击工作区中的 [DriveProgramming]。
- 在 CX-Drive 工具栏中点击  按钮。
- 从菜单中选择 [驱动器]，然后选择 [程序] - [程序编辑器]。



### 使用注意事项

启动 CX-Drive 后，首次启动 DriveProgramming 编辑器时需使用密码。  
 请向本公司销售负责人员确认密码，然后在以下画面中设定密码。

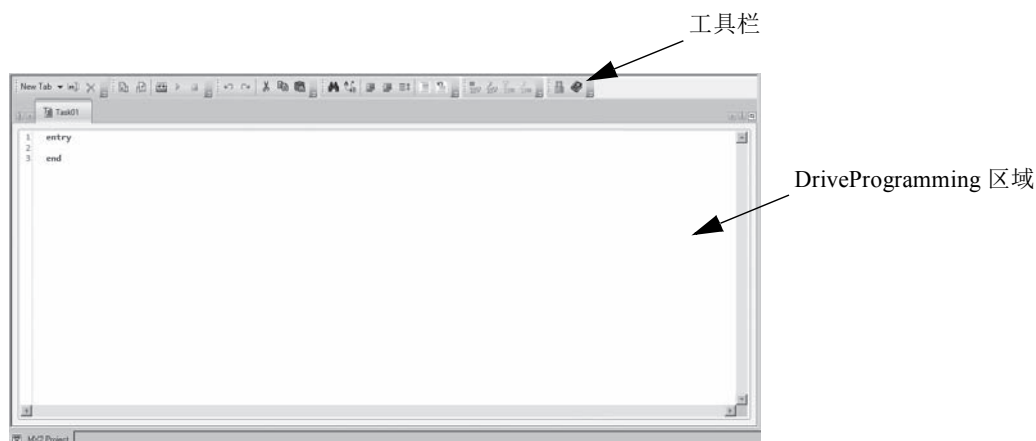


## 4-2 各部分的说明

下面对 CX-Drive 的 DriveProgramming 功能相关的编辑器和窗口的功能进行逐一说明。

### 4-2-1 DriveProgramming 编辑器

DriveProgramming 编辑器为 DriveProgramming 功能的主窗口。



该窗口由包含通用指令的工具栏，以及将程序显示为文本或流程图的 DriveProgramming 区域构成。

### 4-2-2 工具栏

DriveProgramming 编辑器备有以下指令。


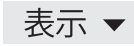
#### ● 通用指令

指令	图标	说明
新建任务 (流程图)		在程序中新建流程图任务。任务是指 DriveProgramming 执行的程序单位。
新建任务 (文本)		在程序中新建文本任务。任务是指 DriveProgramming 执行的程序单位。
新建子程序 (流程图)		新建流程图子程序。子程序是指仅在调用时执行的程序处理。
新建子程序 (文本)		新建文本子程序。子程序是指仅在调用时执行的程序处理。
重命名当前任务 / 子程序		重命名当前任务或子程序。
删除当前标签		删除当前任务或子程序。
撤销		撤销上一步操作。
恢复		恢复“撤销”的操作。
传送至驱动器		编译程序后无异常时，传送 (下载) 至驱动器 (变频器)。

指令	图标	说明
从驱动器传送		将程序从驱动器 (变频器) 传送 (上传) 至 DriveProgramming 编辑器。
编译		编译 DriveProgramming 区域的程序。 程序内发生的编译异常及警告显示在输出窗口的错误列表标签中。
启动		启动变频器内部的程序。CX-Drive 会确认变频器内部与 DriveProgramming 区域的程序, 仅在相同时启动程序。不同时间程序不会启动。 此外, 该程序的启动仅在连接了 CX-Drive 时有效。 变频器单独启动时, 请务必设定 EzSQ 功能选择 (UE-02)。
停止		停止变频器内部的程序。该动作的执行与 CX-Drive 的 DriveProgramming 区域的程序无关。
程序编号的设定		对程序设定编号。(可设定范围为 0 ~ 9999。)程序编号在将程序传送 (下载) 至驱动器 (变频器) 后生效, 可使用程序编号监控 (DriveProgramming) (db-02) 进行确认。
密码		可设定、变更或删除程序的密码。
帮助		显示 CX-Drive 的帮助。

#### ● 流程图程序的指令

指令	图标	说明
放大		放大。
缩小		缩小。
缩放复位		将缩放恢复为初始值。
选择		使用鼠标光标进行拖动, 可选择 1 个以上的程序块 *1。
全部滚动		通过拖动, 可在保持相同比例的状态下将视图范围移向任意方向。
左对齐		将当前所选区块 *1 左对齐。
左右居中		将当前所选区块 *1 左右居中。
右对齐		将当前所选区块 *1 右对齐。
上对齐		将当前所选区块 *1 上对齐。
上下居中		将当前所选区块 *1 左右居中。
下对齐		将当前所选区块 *1 下对齐。
方向		选择连接区块 *1 的方向。
自动配置		按当前所选的方向自动配置流程图中的区块 *1。

指令	图标	说明
显示接点		切换区块* <sup>1</sup> 接点的显示或隐藏。
显示		可从下列选项中选择程序的显示样式。 <ul style="list-style-type: none"> <li>• 仅文本</li> <li>• 仅图标</li> <li>• 图标和文本</li> <li>• 名称、图标、自变量</li> </ul>

\*1. 将流程图中的 1 个显示单位称作“区块”。区块由 1 个指令或在 1 个指令上添加标签后构成。

## ● 文本程序的指令





指令	图标	说明
查找		以 DriveProgramming 编辑器中的文本为对象进行查找。
替换		以 DriveProgramming 编辑器的文本为对象进行替换。
缩进		增加所选文本的缩进。
取消缩进		减小所选文本的缩进。
文本的格式化		自动格式化所选文本。
注释文本		将所选文本行转换为注释。
取消文本的注释		取消所选文本行的注释。

## ● 程序转换

可将流程图程序转换为文本程序或进行反向转换。

转换时会进行编译，因此发生编译异常时将无法转换。

此外，通过转换时的编译，动作仍相同，但会删除注释、别名定义、区域定义等，其格式和内容会有所变更。

指令	图标	说明
流程图转换		将当前的文本任务及子程序转换为流程图。
整体流程图转换		将整个程序转换为流程图。
文本转换		将当前流程图的任务及子程序转换为文本。
整体文本转换		将整个程序转换为文本。



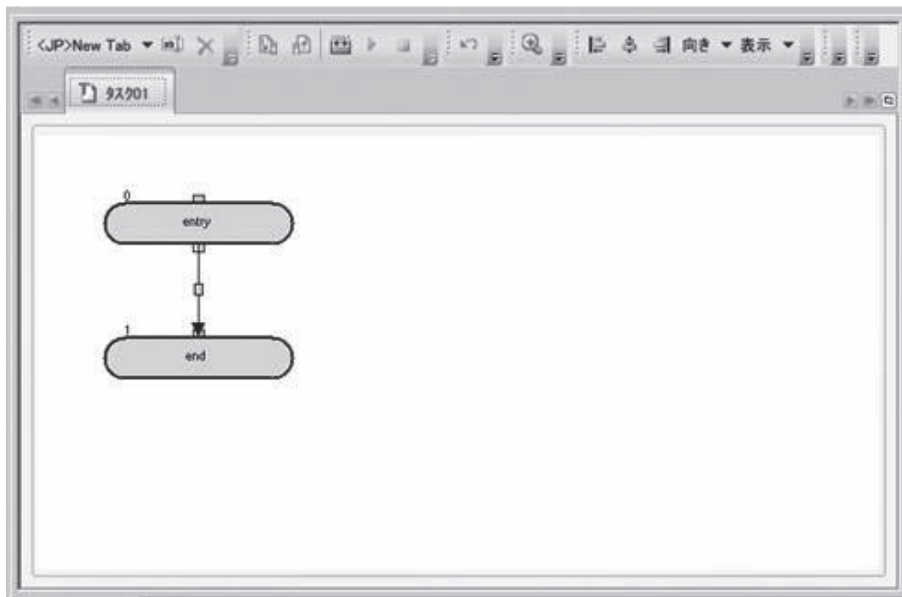
## ● 快捷键

DriveProgramming 区域可使用以下键盘快捷键。

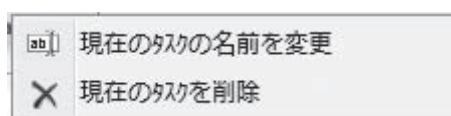
快捷键	内容
Ctrl + X	剪切
Ctrl + C	复制
Ctrl + V	粘贴
Ctrl + Z	撤销
Ctrl + Y	恢复
Ctrl + A	全选
Ctrl + F	查找和替换
Ctrl + 空格	指令一览表
Tab	选择下一个 (仅限流程图)
箭头键	移动所选区块
Home	滚动至开头 (仅限流程图)
End	滚动至末尾 (仅限流程图)
Page Up	将光标移动至行首
Page Down	将光标移动至行末
+	放大
-	缩小

### 4-2-3 DriveProgramming 区域

DriveProgramming 区域显示程序的设计状态。



DriveProgramming 区域的标签中含有多个页面。各标签为流程图或文本的任务及子程序。右击各标签的标签标题，可删除或重命名任务及子程序。



## 流程图程序

流程图程序将 1 个显示单位称作“区块”。

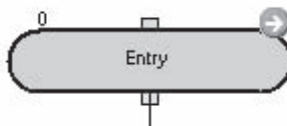
将多个区块配置在区域中，设定区块之间的相互关系并创建程序。

流程图程序中，程序在正常编译后，将使用绿底白色箭头图标突出显示各流程图任务的起点。

此外，程序正常编译时，将在各区块的左上角显示区块 No.。

区块 No. 是指对整个程序的各区块从 0 开始的连续编号。

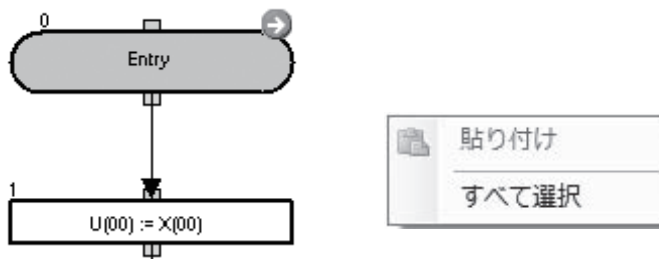
与文本程序转换时的行号不同。



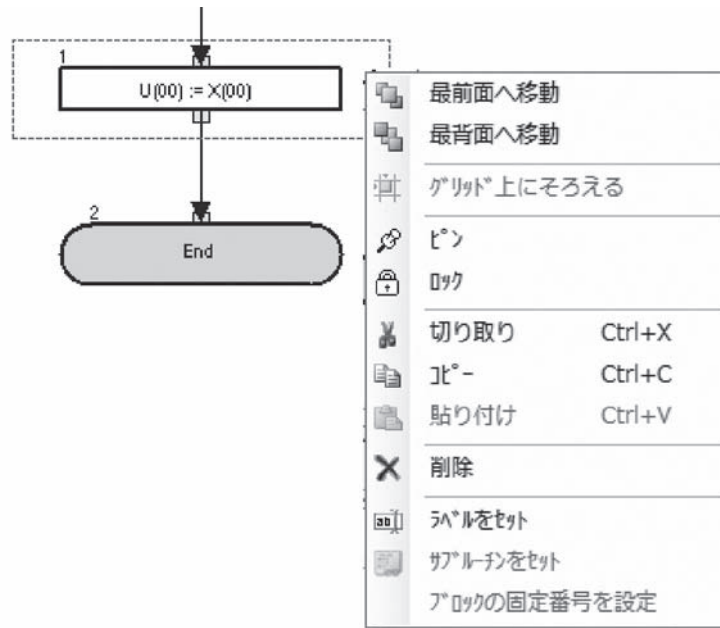
输出窗口会显示程序是否已正常编译。编译时发生异常的程序中，可根据红色感叹号图标识别流程图程序中存在错误的区块。将鼠标移动至异常图标上会显示编译异常，也可通过输出窗口的错误列表进行确认。



在流程图程序内未选择任何区块而右击时，可粘贴上一次复制的区块或选择所有区块。



右击流程图程序内的区块，将显示含多个选项的弹出菜单。



下面对流程图程序可使用的菜单指令进行说明。

指令	说明
[向最前面移动]	将所选区块配置至其它区块的前面。
[向最后面移动]	将所选区块配置至其它区块的后面。
[固定]	将所选区块固定至图表内的当前位置。此时，将无法通过拖动操作进行移动。
[锁定]	与 [固定] 的动作相同，并将区块的属性设定为只读。
[剪切]	删除所选区块，并保存至剪贴板中以供稍后粘贴。
[复制]	将所选区块保存至剪贴板中，以供稍后粘贴。
[粘贴]	粘贴之前复制至剪贴板的内容。也可在其它应用软件中作为图像数据进行粘贴。
[设置标签]	对所选区块设定标签名。
[设置子程序]	对所选区块设定子程序名。仅在可设定子程序的区块中显示为有效。

## 文本程序

文本程序使用文本语言创建程序。

文本程序无法正常编译时，将在输出窗口中显示程序的异常。此外，发生异常的行会显示红色下划线。

```
Dummy UL01 := A038
```

右击所选文本，将显示弹出菜单。



下面对文本程序可使用的菜单指令进行说明。

指令	说明
[查找]	在程序中查找所选文本。
[替换]	在程序中替换所选文本。
[剪切]	删除所选文本，并保存至剪贴板中以供稍后粘贴。
[复制]	将所选文本保存至剪贴板中，以供稍后粘贴。
[粘贴]	粘贴之前复制至剪贴板的内容。也可在其它应用软件中作为图像数据进行粘贴。
[跳至子程序]	跳至所选文本子程序。
[跳至标签]	跳至所选文本标签。
[撤销]	撤销最后一次变更。
[恢复]	恢复最后一次撤销的变更。
[帮助]	显示 CX-Drive 的帮助。

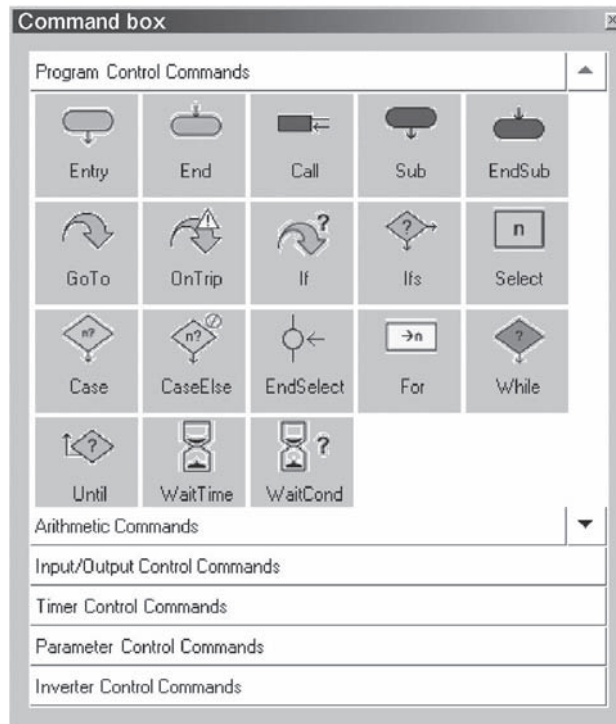
### 4-2-4 指令框窗口

使用指令框窗口时，可通过拖放在 DriveProgramming 区域中添加区块。流程图的区块按照指令类别集中显示。

指令框在启动 DriveProgramming 编辑器时显示。在 [视图] 菜单的 [DriveProgramming] 中点击 [指令框]，可切换显示 / 隐藏。

CX-Drive 的初始设定下，指令框窗口显示在 CX-Drive 的右侧。

双击窗口的标题栏 (粗边框)，可切换成独立的窗口显示。



此外，用鼠标右击，可选择显示样式。样式有 [大图标]、[小图标]、[列表] 3 种可选。无论使用哪种样式，将鼠标光标移动至区块上时，均会显示简短的帮助文本。

需显示类别下属的区块时，点击该类别的标题。

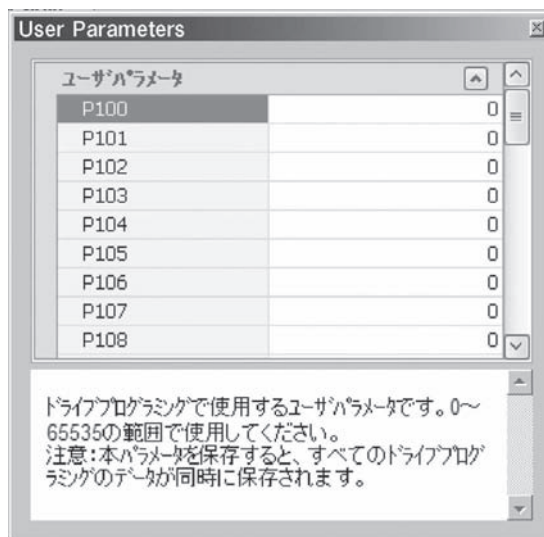
### 4-2-5 用户参数窗口

用户参数窗口可编辑作为程序变量进行动作的 DriveProgramming 用户参数。该参数分类显示。

用户参数在启动 DriveProgramming 编辑器时显示。在 [视图] 菜单的 [DriveProgramming] 中点击 [用户参数]，可切换显示 / 隐藏。

CX-Drive 的初始设定下，用户参数窗口显示在 CX-Drive 的右侧。

双击窗口的标题栏 (粗边框)，可切换成独立的窗口显示。



需变更参数值时，将光标移动至该行并点击参数 No. 右侧的编辑框，输入新的值。输入值超出有效范围时，将显示警告。

窗口下方会显示参数相关的说明。

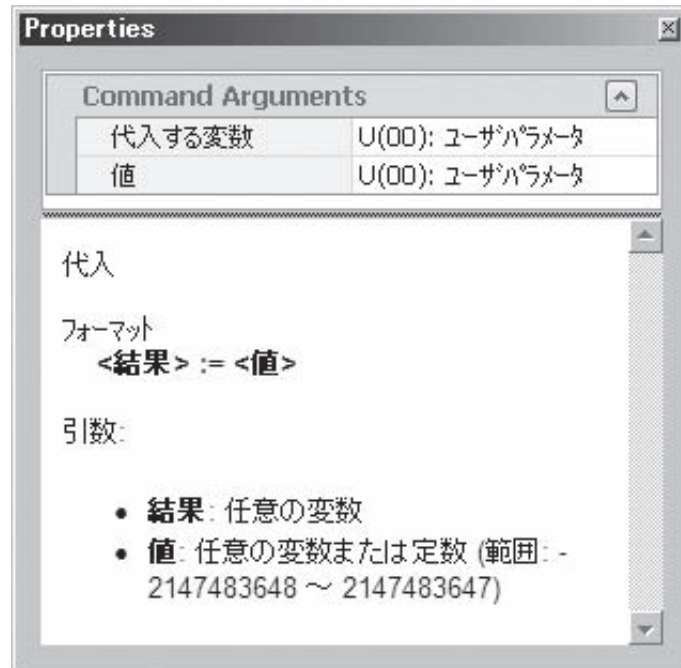
## 4-2-6 属性窗口

属性窗口可编辑流程图程序当前所选区块的属性。



属性在启动 DriveProgramming 编辑器时显示。在 [视图] 菜单的 [DriveProgramming] 中点击 [属性]，可切换显示 / 隐藏。

CX-Drive 的初始设定下，属性窗口显示在 CX-Drive 的右侧。

双击窗口的标题栏 (粗边框)，可切换成独立的窗口显示。



需变更区块的变量时，选择流程图程序的区块，将光标置于属性窗口内需输入处。

- 区块的变量有选项时，点击当前值，右侧将显示  图标。 点击或双击当前设定值时，将显示下拉菜单，展开可选择的选项。
- 区块的变量无选项时，可通过点击当前值并输入新的值，变更数值。输入值超出有效范围时，将显示警告。
- 区块的变量可设定选项以及进行任意设定时，可点击当前设定值直接设定数值。此外，双击当前设定值，将通过下拉菜单显示可选择的选项。

### 4-2-7 输出窗口的错误列表标签

点击输出窗口中的错误列表标签，将显示 DriveProgramming 的错误列表。

编译了 DriveProgramming 编辑器创建的程序时，将显示发生的编译异常及警告。

显示编译异常时，表示程序未正确编译。

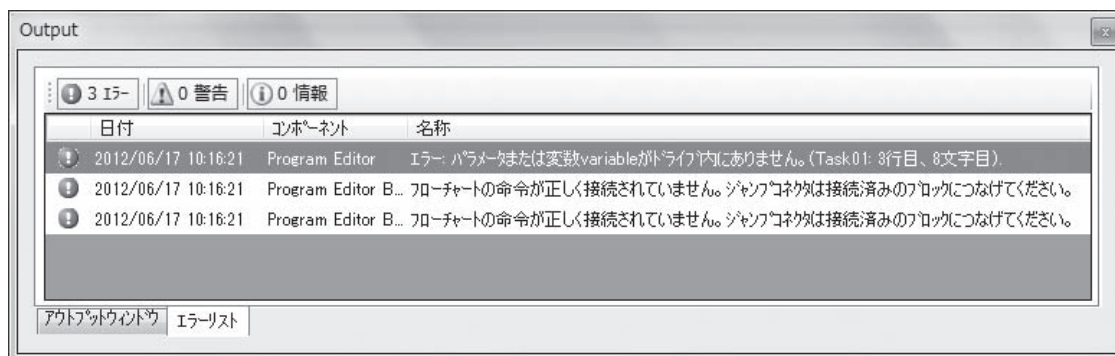
因此，无法传送至变频器，或执行流程图程序和文本程序的转换。

编译正确完成但程序中有应修正的内容时，将显示警告内容。

错误列表会在每次编译完成时自动更新，解决的异常及警告将得到清除。

CX-Drive 的初始设定下，输出窗口显示在 CX-Drive 的下侧。

双击窗口的标题栏（粗边框），可切换成独立的窗口显示。



3 15- [错误] 按钮会切换列表中异常的显示 / 隐藏。

0 警告 [警告] 按钮会切换列表中警告的显示 / 隐藏。

0 情報 [信息] 按钮会切换列表中信息的显示 / 隐藏。

列表信息显示如下。

信息	说明
日期	发生异常的日期和时间
组件	显示发生异常的区块。
名称	显示异常或警告信息的名称及内容。



## 4-3 任务的添加、删除和重命名

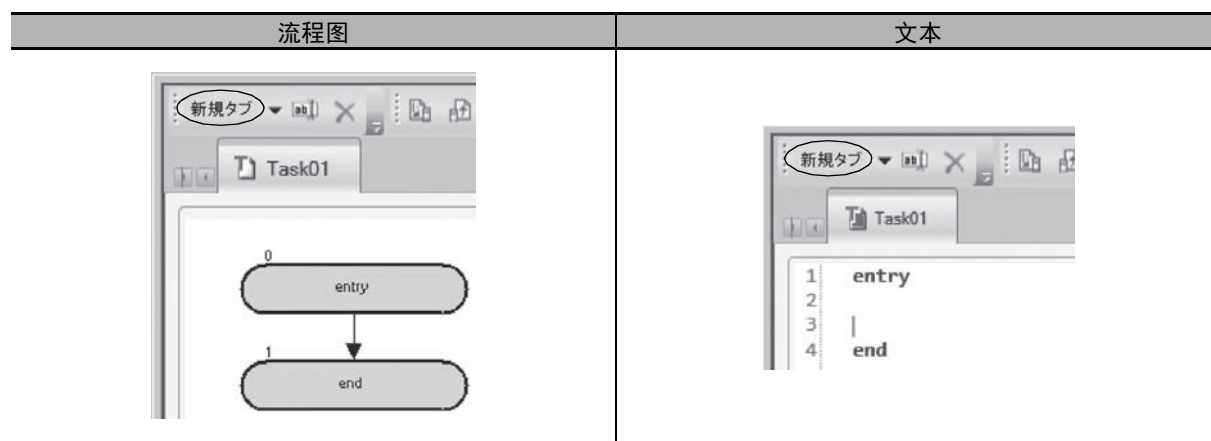
CX-Drive 的初始设定下，启动 DriveProgramming 编辑器时会显示空任务。

添加新任务时，请在 DriveProgramming 编辑器的工具栏中选择 [新建标签]，然后选择 [新建任务 (流程图)] 或 [新建任务 (文本)]。

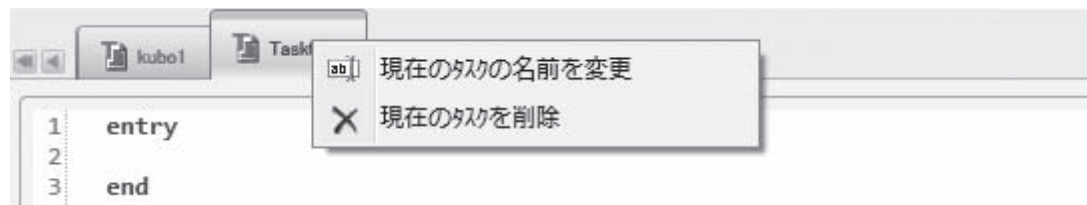
选择后，DriveProgramming 编辑器中将显示新任务。

显示的任务越靠左优先级越高，会在 1ms 或 2ms 的处理时间 (通过 UE-01 选择) 内优先处理。

此外，所有任务均需以“entry”指令开始，以“end”指令结束。



右击 DriveProgramming 编辑器的任务标签部分，可删除或重命名所选任务。



## 4-4 子程序的插入、删除和调用

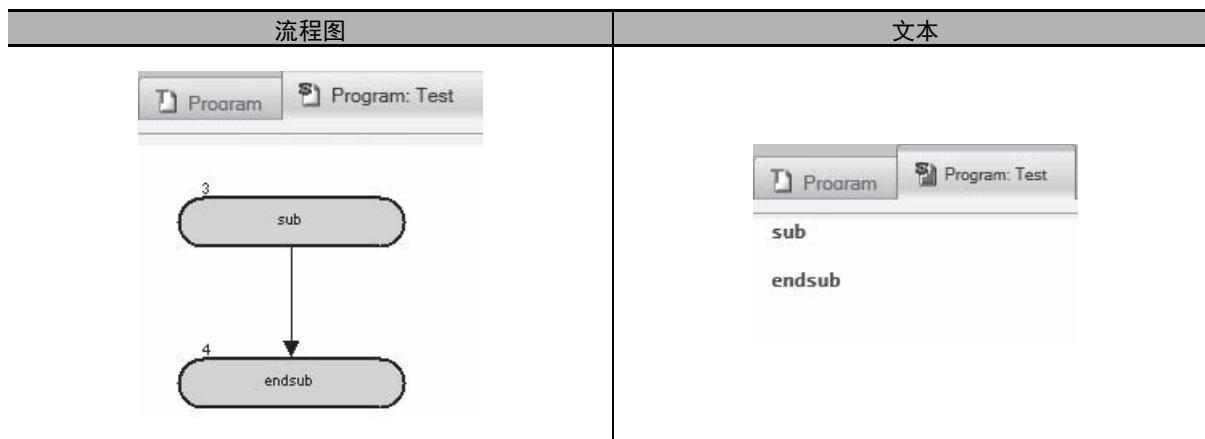
插入子程序时，请在选择需插入子程序的任务标签后，在 DriveProgramming 编辑器的工具栏中选择 [新建标签]，然后选择 [新建子程序 (流程图)] 或 [新建子程序 (文本)]。

选择后，DriveProgramming 编辑器中将显示新的子程序。

与任务相同，右击子程序的标签，可进行删除或重命名。

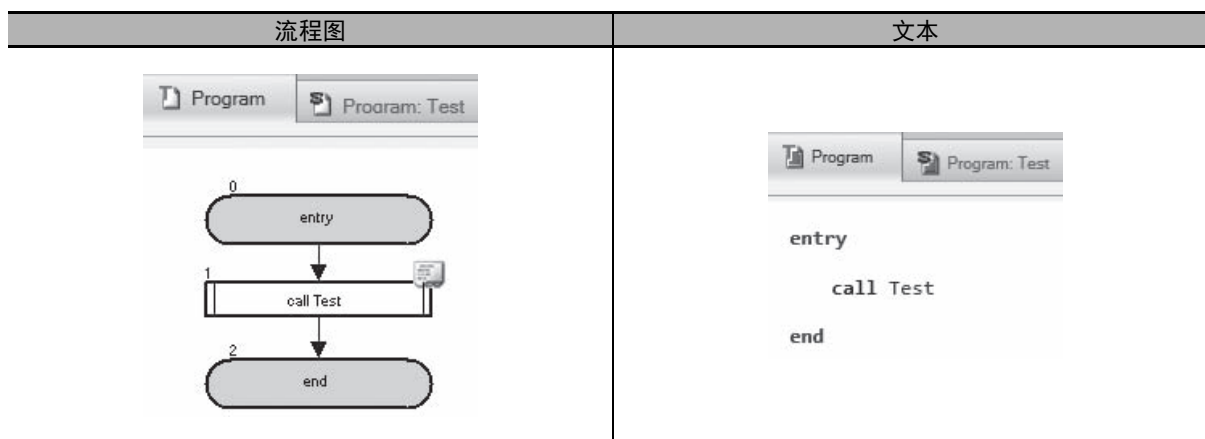
子程序的名称前显示所属任务名，按“任务名：子程序名”的格式显示。

所有子程序均需以“sub”指令开始，以“endsub”指令结束。



需执行子程序时，使用“call”指令指定子程序名。

各任务可调用的子程序仅限各任务下属的子程序。各任务之间无法通用。多个任务执行同个子程序处理时，请在各任务中创建相同的子程序。



## 4-5 流程图程序的创建

创建 DriveProgramming 的程序时，可以任务及子程序为单位选择流程图程序或文本程序。  
流程图程序请按照以下步骤进行创建。

- 1** 请打开 DriveProgramming 编辑器。  
将自动显示 DriveProgramming 的辅助窗口 ( 指令框、用户参数、属性及错误列表标签 )。
- 2** 请使用工具栏，从 [ 新建标签 ] 中选择 [ 新建任务 ( 流程图 ) ] 或 [ 新建子程序 ( 流程图 ) ]。
- 3** 请从指令框窗口中选择指令，并拖放至 DriveProgramming 编辑器中。  
编译后，编辑器中配置的各区块的左上方会显示区块 No.。  
区块 No. 是指对整个程序的各区块从 0 开始的连续编号。  
与文本程序转换时的行数不同。
- 4** 请点击区块编辑属性。  
在属性窗口中编辑与区块相关的变量。
- 5** 请按照程序顺序依次连接各区块。  
可从显示区块的橙色接点拖动至其它区块的绿色接点进行连接。
- 6** 请在用户参数窗口中编辑用户参数。
- 7** 请执行程序的编译、传送至变频器及保存程序等。  
编译程序，并确认程序中无编译异常。  
编译正常结束时，可传送至变频器。  
保存程序时，保存整个项目。此外，可使用程序的导出功能单独保存程序。

## 4-6 文本程序的创建

创建 DriveProgramming 的程序时，可以任务及子程序为单位选择流程图程序或文本程序。  
文本程序请按照以下步骤进行创建。

- 1** 请打开 DriveProgramming 编辑器。  
将自动显示 DriveProgramming 的辅助窗口 ( 指令框、用户参数、属性及错误列表标签 )。
- 2** 请使用工具栏，从 [新建标签] 中选择 [新建任务 ( 文本 )] 或 [新建子程序 ( 文本 )]。
- 3** 文本程序的编辑方法有 3 种。
  - 手动输入
  - 调用文本指令一览表 (Ctrl + 空格)
  - 从指令框窗口中拖放指令
- 4** 设定各指令的变量。  
从指令框窗口拖放的指令和从文本指令一览表中 (Ctrl + 空格) 选择的指令会用绿色背景显示所需变量。  
设定各变量完成指令。  
手动输入指令时，请参阅各输入显示的支持显示完成指令。各指令的详细内容请参阅 □ 「第 6 章 DriveProgramming 指令」。  
编译时会删除不需要的行及空格，因此创建程序时无需在意行和空格。
- 5** 请执行程序的编译、传送至变频器及保存程序等。  
编译程序，并确认程序中是否有编译异常。  
编译正常结束时，可传送至变频器。  
保存程序时，保存整个项目。此外，可使用程序的导出功能单独保存程序。

## 4-7 已传送 ( 上传 ) 程序的编辑

可从含有程序的变频器传送 ( 上传 ) 程序后进行编辑。

请按照以下步骤编辑程序。

- 1** 请打开 DriveProgramming 编辑器。  
将自动显示 DriveProgramming 的辅助窗口 ( 指令框、用户参数及属性 )。
- 2** 请在线连接 CX-Drive。请从菜单中选择 [ 驱动器 ] 后选择 [ 在线连接 ]，或者点击 CX-Drive 工具栏中的 [ 在线连接 ] 图标。
- 3** 点击 DriveProgramming 编辑器工具栏中的 [ 从驱动器传送 ] 图标。  
会从驱动器 ( 变频器 ) 传送程序，并自动显示在 DriveProgramming 编辑器的 DriveProgramming 区域。
- 4** 编辑传送 ( 上传 ) 的程序。  
变频器内部已下载“编译后的程序”。  
因此，传送 ( 上传 ) 的程序会显示为文本程序。  
需显示成流程图程序时，请点击 DriveProgramming 编辑器工具栏中的 [ 将整个程序转换为流程图 ]，转换为流程图程序。
- 5** 请执行程序的编译、传送至变频器及保存程序等。  
编译程序，并确认程序中无编译异常。  
编译正常结束时，可传送至变频器。  
保存程序时，保存整个项目。此外，可使用程序的导出功能单独保存程序。

存在 DriveProgramming 功能的程序时，也可使用 CX-Drive 工具栏的 [ 传送 [ 计算机 → 驱动器 ] ] / [ 传送 [ 驱动器 → 计算机 ] ] 图标，与变频器之间传送程序。此时，请在询问是否传送参数、程序或两者同时传送的对话框中选择程序。

## 4-8 程序的保存

使用 DriveProgramming 功能创建的程序可使用下述 2 种方法保存。  
请根据用途选择合适的方法进行保存。

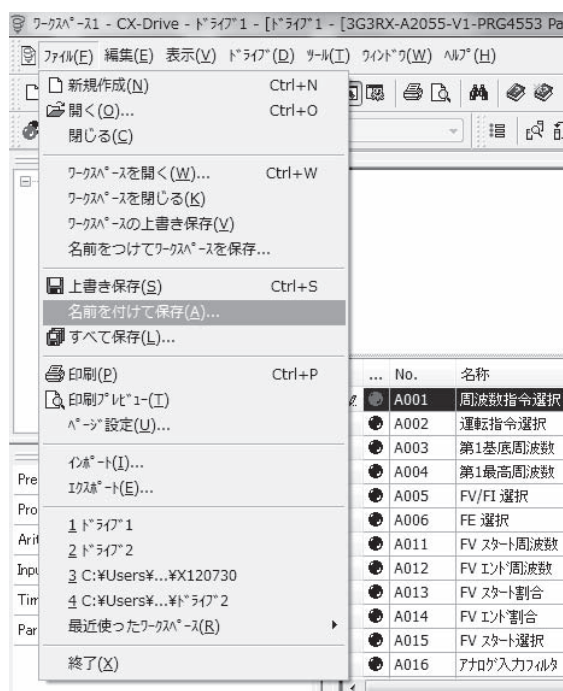
### ● CX-Drive 整个项目的保存

保存 CX-Drive 创建的整个项目时，可保存包括 DriveProgramming 程序的所有驱动器程序。

请点击菜单栏的 [文件] 选择 [另存为]，并输入文件名。

打开已保存的项目时，将自动加载该项目包含的 DriveProgramming。

程序在双击工作区的 DriveProgramming 启动 DriveProgramming 编辑器时显示。

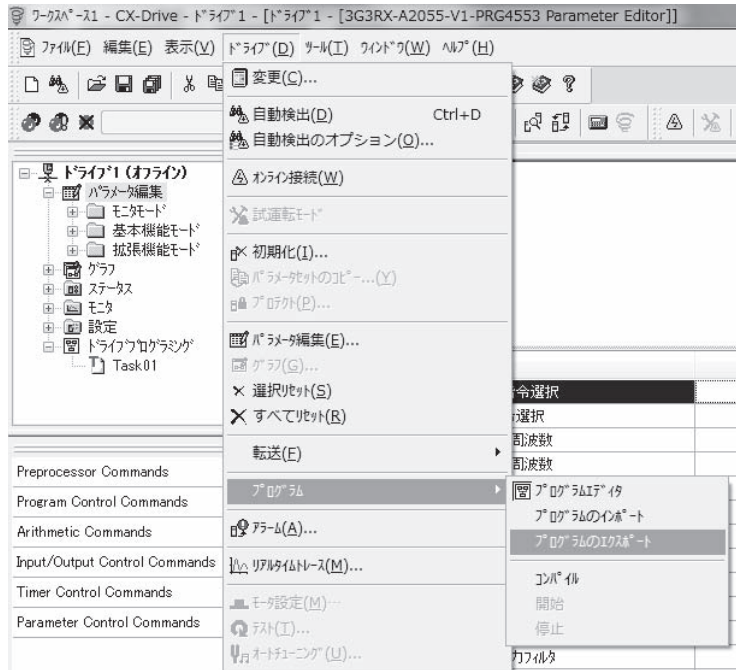


### ● DiveProgramming 的导出保存

可单独保存 DriveProgramming 功能的程序。

请点击菜单栏的 [驱动器]，从 [程序] 中选择 [导出程序] 并输入文件名。区分于其它驱动器信息，单独保存程序。

需将导出的程序文件导入 CX-Drive 时，请点击菜单栏的 [驱动器]，从 [程序] 中选择 [导入程序]，选择需导入的文件。



## 4-9 程序的传送和核对

---

可在 CX-Drive 的 DriveProgramming 功能与变频器本体之间传送和核对程序。也可同时传送和核对参数。

### 传送 ( 从计算机至变频器 )

---

编译创建的程序 ( 程序校验 )，无异常时传送 ( 下载 ) 至变频器。

可从 [ 驱动器 ] 菜单中选择 [ 传送 ] - [ 传送 [ 计算机 → 驱动器 ] ] 进行传送 ( 下载 )。

### 传送 ( 从变频器至计算机 )

---

将变频器中的程序及参数传送 ( 上传 ) 至 DriveProgramming。

可从 [ 驱动器 ] 菜单中选择 [ 传送 ] - [ 传送 [ 驱动器 → 计算机 ] ] 进行传送 ( 上传 )。

### 核对 ( 计算机与变频器 )

---

将 DriveProgramming 的程序及参数与变频器本体的数据进行核对。

可从 [ 驱动器 ] 菜单中选择 [ 传送 ] - [ 核对 [ 驱动器 - 计算机 ] ] 进行核对。

### 部分传送 ( 计算机至变频器 )

---

可从 [ 驱动器 ] 菜单中选择 [ 传送 ] - [ 部分传送 [ 计算机 → 驱动器 ] ] 进行部分传送 ( 下载 )。

### 部分传送 ( 从变频器至计算机 )

---

可从 [ 驱动器 ] 菜单中选择 [ 传送 ] - [ 部分传送 [ 驱动器 → 计算机 ] ] 进行部分传送 ( 上传 )。



## 4-10 程序的执行 (DriveProgramming 功能选择)

将程序传送 (下载) 至变频器后, 执行程序有以下 2 种方法。

### 通过 CX-Drive 执行程序

通过 CX-Drive 执行程序便于调试创建的程序。

可点击 DriveProgramming 编辑器工具栏中的  图标启动程序, 点击  图标停止程序。通过 CX-Drive 执行程序仅在下列情况下有效。

- 连接了 CX-Drive 时有效。  
启动程序后即使断开 CX-Drive 的连接也不会停止, 因此请务必在可通过关闭电源等方式立即停止变频器的状态下使用。
- 通过编译后的核对确认 CX-Drive 的程序与变频器的程序一致时, 启动有效。
- 程序的停止与 CX-Drive 的程序无关。

### 实际应用中的程序执行

通过 EzSQ 功能选择 (UE-02) 设定变频器的参数后执行程序。

程序完成一系列处理到达“end”指令时, 必须重新启动程序才能执行。

若需重复程序, 创建时请在程序内构建循环避免到达“end”指令。

在电机运行时停止了 DriveProgramming 功能时, 将根据停止方式选择 (AA115) 的设定停止电机 (减速停止或自由运行停止)。

参数 No.	功能名称	数据	说明
UE-02	EzSQ 功能选择	00: 无效	将 DriveProgramming 功能设为无效。不执行程序。程序执行过程中将数据变更为“00: 无效”时, 程序将停止。
		01: [PRG] 端子	输入“99: PRG”功能中设定的输入端子*1时, 将启动 DriveProgramming 的程序。
		02: 始终	在变频器电源接通后, 自动启动 DriveProgramming 的程序。程序停止过程中将数据变更为“02: 始终”时, 程序启动。
CA-01 ~ CA-11	输入端子 1 ~ 9、A、B	99: PRG	EzSQ 功能选择 (UE-02) 设定为“01: [PRG] 端子”时, 进行了该设定的输入端子将启动程序。

\*1. 输入端子以 1 ~ 9、A、B 为对象。



### 安全要点

DriveProgramming 功能的程序停止时，由 DriveProgramming 控制的输出端子功能将保持程序停止前的状态。

因此，请采用通过 DriveProgramming 启动信号和报警（跳闸）信号检出变频器 DriveProgramming 功能的程序停止，安全停止变频器周边设备的接线。



### 使用注意事项

DriveProgramming 功能的程序停止时各功能变量的状态如下所述，请识别各状态采取以下措施。

- 输出端子（实际的变频器输出端子）将保持程序停止前的状态。  
但未设定 DriveProgramming 功能的通用输出 MO1 ~ MO7 的输出将作为通常的变频器输出端子进行动作。
- 在频率指令、加减速时间、模拟输出中设定了 DriveProgramming 功能时，各设定值将保持程序停止前的状态。
- 用户参数变量、内部用户变量、内部用户接点将保持程序停止前的状态。
- 输入端子及模拟输入等实际的变频器输入端子状态会一直更新状态，而不保持输入状态。
- 输出变量（RUN、FA1、AL 等功能位）及变频器监控变量会根据变频器的状态一直更新状态，而不保持状态。
- 只有 DriveProgramming 的输入变量（FW、RV、CF1 等功能位）和定时器变量会在程序停止的同时清除，所有状态清零。

通过 CX-Drive 启动 / 停止 DriveProgramming 功能时，CX-Drive 启动和停止时分别将变频器的参数 EzSQ 功能选择（UE-02）的值暂时（仅 RAM 上的数据）变更为“02”、“00”。

使用 CX-Drive 启动 / 停止 DriveProgramming 功能后，请采取以下措施。

- 请在重新接通变频器的电源后，将 UE-02 恢复成 EEPROM 中保存的值。
- 在重新接通变频器的电源前，请勿执行以下 EEPROM 保存操作。
  - 从 CX-Drive 传送（下载）部分参数。
  - 通过 Modbus 通信或通信发出“enter”指令。

设定为 UE-02 = 02 的情况下，通过 CX-Drive 重新启动程序时，请停止接通电源时启动的程序后再执行。

在接通电源时启动的程序启动的状态下，无法通过 CX-Drive 重新启动程序。

变频器运行的状态下，点击 CX-Drive 的程序启动按钮，最终将显示以下信息。

“驱动器与 CX-Drive 的初始化数据不同。”



在显示该信息时点击“是”，则无论变频器的运行状态如何，均会强制启动程序。

请确认装置状态，确保安全后再启动。

## 4-11 其它便捷功能

### 从流程图转换为文本

从流程图转换为文本有 2 种方法。

指令	图标	说明
文本转换		将当前流程图的任务及子程序转换为文本。
整体文本转换		将整个程序转换为文本。





#### 使用注意事项

- 将流程图程序转换为文本程序时，会编译程序。因此，编译发生异常时将无法转换。
- 编译后，流程图程序各区块的左上方会显示区块 No.。  
区块 No. 是指对整个程序的各区块从 0 开始的连续编号。  
与文本程序转换时的行数不同。

### 从文本转换为流程图

从文本转换为流程图有 2 种方法。

指令	图标	说明
流程图转换		将当前的文本任务及子程序转换为流程图。
整体流程图转换		将整个程序转换为流程图。



#### 使用注意事项

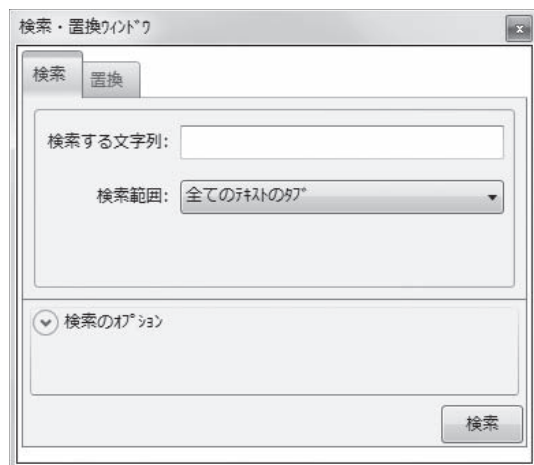
- 将文本程序转换为流程图程序时，会编译程序。因此，编译发生异常时将无法转换。
- 编译后，流程图程序各区块的左上方会显示区块 No.。  
区块 No. 是指对整个程序的各区块从 0 开始的连续编号。  
与文本程序转换时的行数不同。
- 转换时的编译会删除文本程序创建的注释、别名定义、区域定义等，动作相同，但其格式和内容会有所变更。

## 查找和替换功能

查找和替换功能仅限文本程序使用。通过使用查找和替换功能，可查找或替换文本程序内的任意字符串。

需使用查找功能时，点击  图标或按下快捷键 Ctrl + F，选择 [查找] 标签。

需使用替换功能时，点击  图标或按下快捷键 Ctrl + F，选择 [替换] 标签。



查找功能



替换功能

## 注释的添加 ( 文本程序 )

可在文本程序的任务及子程序中添加注释。流程图程序无法添加注释。需在文本行中添加注释时，在“”后输入注释。注释显示为绿色。

### ● 示例

```
#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
```



### 使用注意事项

- 将程序传送 ( 下载 ) 至变频器或为转换程序而进行编译时，文本程序中创建的注释会被删除。需保存创建的注释时，请保存并管理编译前的程序。可通过保存 CX-Drive 的整个项目文件或保存程序的导出文件保存程序。
- 程序的核对会比较“编译后的程序”与变频器内部的程序，因此不会核对注释、别名定义、区域定义等。

## 别名定义 ( 文本程序 )

可在文本程序任务中的“entry”指令之前定义别名。无法在子程序或流程图程序中定义别名。

别名定义是指为参数、变量、指令和数值常数指定任意名称。

使用别名定义时，可像处理参数、变量、指令和数值那样，在程序中处理指定的名称。有助于提高程序的可视性。

- **本地别名：**可在当前任务及子程序内使用该别名定义。无法在程序内的其它任务及子程序中使用。任务中本地别名定义的格式如下所示。

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

### ● 示例

```
#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
```

- **全局别名：**可在程序内的所有任务及子程序中使用该别名定义。在最多 5 个任务中，各任务均可设定。全局别名定义的格式如下所示。

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

### ● 示例

```
#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
```



### 使用注意事项

- 将程序传送 ( 下载 ) 至变频器或为转换程序而进行编译时，文本程序中创建的别名定义会被删除，指定的任意名称将转换为正常名称或数值。需保存创建的别名定义时，请保存并管理编译前的程序。可通过保存 CX-Drive 的整个项目文件或保存程序的导出文件保存程序。
- 程序的核对会比较“编译后的程序”与变频器内部的程序，因此不会核对注释、别名定义、区域定义等。
- 别名定义中无法使用与已使用的变量及指令相同的名称。否则将显示编译异常。

## 区域定义 ( 文本程序 )

可在文本程序的任务及子程序中定义区域。流程图程序无法定义区域。

区域定义是指通过指定区域将程序分组。

使用区域定义时，可将程序任意分割后显示，分割后的各区域可折叠，从而更易于查看程序。

### ● 示例

```
#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

#region Stop...
```



### 使用注意事项

- 将程序传送 ( 下载 ) 至变频器或为转换程序而进行编译时，文本程序中创建的区域定义会被删除。  
需保存创建的区域定义时，请保存并管理编译前的程序。可通过保存 CX-Drive 的整个项目文件或保存程序的导出文件保存程序。
- 程序的核对会比较“编译后的程序”与变频器内部的程序，因此不会核对注释、别名定义、区域定义等。

# 5

## DriveProgramming 的用户变量

本章对 DriveProgramming 的用户变量进行介绍。

---

5-1 用户用的变量和用户参数 .....	5-2
5-2 输入输出端子变量 .....	5-5
5-3 定时器变量 .....	5-10
5-4 变频器设定变量 .....	5-12
5-5 变频器监控变量 .....	5-14
5-6 输入变量 .....	5-17
5-7 输出变量 .....	5-20

## 5-1 用户用的变量和用户参数

备有创建程序用的用户参数变量、内部用户变量、内部用户接点。

请用于程序用户界面、运算的开始数据、运算过程中的数据保存及暂存、数据保存等。

### 用户参数变量 U(00) ~ U(63)

DriveProgramming 功能的用户参数变量 U(00) ~ U(63) 为无符号单字变量。

变频器本体的 DriveProgramming 用户参数 U00 ~ U63(UE-10 ~ UE-73) 分别对应 DriveProgramming 功能的用户参数变量 U(00) ~ U(63)。

请用于以下用途。

用途	说明
用户界面	用作调整运算结果用的参数。可根据应用状态，通过变频器的参数 (UE-10 ~ UE-73) 进行调整。
运算的开始数据	用户参数变量 U(00) ~ U(63) 可用作运算用的变量。EEPROM 中保存的 UE-10 ~ UE-73 数据可用作运算的开始数据。
运算过程中的数据保存及暂存	可用作保存、暂存运算过程中数据的变量。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
U(00) ~ U(63)	用户参数变量 对应 UE-10 ~ UE-73 的变量	0 ~ 65535	0	-	无符号 单字	R/W

- 接通电源时，参数 UE-10 ~ UE-73 的 EEPROM 中保存的数据会设定至用户参数变量 U(00) ~ U(63) 中。
- 程序启动后的用户参数变量 U(00) ~ U(63) 的状态可通过监控参数 UE-10 ~ UE-73 进行确认。但 LCD 操作器中显示的数据为显示瞬间的数据。需更新数据时，请在显示参数 No. 后再次显示数据。
- 使用 LCD 操作器变更参数 UE-10 ~ UE-73 并按下 Enter 键后，变更后的数据将保存至各参数的 EEPROM 中，并反映至当前的用户参数变量 U(00) ~ U(63) 中。再次接通电源时，EEPROM 中保存的数据将设定至 U(00) ~ U(63)。
- 用户参数变量 U(00) ~ U(63) 的数据在程序中变更时，参数 UE-10 ~ UE-73 也不会保存至 EEPROM。



#### 使用注意事项

DriveProgramming 功能的程序停止时，用户参数变量将保持程序停止前的状态。程序再次启动时，将从保持的数据起开始。



## 内部用户变量 UL(00) ~ UL(15)

DriveProgramming 功能的内部用户变量 UL(00) ~ UL(15) 为带符号双字变量。请用于四则运算结果等运算过程中的数据保存及暂存。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
UL(00) ~ UL(15)	内部用户变量 对应 UF-02 ~ UF-32 的变 量	-2147483647 ~ 2147483647	0	-	带符号 双字	R/W

- 接通电源时，参数 UF-02 ~ UF-32 的 EEPROM 中保存的数据会设定至用户参数变量 UL(00) ~ UL(15) 中。
- 用户参数变量 UL(00) ~ UL(15) 的数据在程序中变更时，参数 UF-02 ~ UF-32 也不会保存至 EEPROM。
- 需将内部用户变量 UL(00) ~ UL(15) 的高位字数据移动至单字数据的用户参数变量 U(00) ~ U(63) 时，使用以下运算。  
 代入正高位字数据：U(00)=UL(00)/65536  
 代入负高位字数据：U(00)=UL(00)/65535  
 执行“U(00)=UL(00)”时，低位字数据将移动至 U(00)。



### 使用注意事项

DriveProgramming 功能的程序停止时，内部用户变量将保持程序停止前的状态。程序再次启动时，将从保持的数据起开始。

## 内部用户接点 UB(0) ~ UB(15)

DriveProgramming 功能的内部用户接点 UB(0) ~ UB(15) 为位访问的变量。请用于位运算的数据保存及暂存。

使用 UBw 变量时，也可用作以内部用户接点 UB(0) ~ UB(15) 为低位字节的字访问数据。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
UB(0) ~ UB(15)	内部用户接点 (位访问)	0: OFF 1: ON	0	-	位	R/W

- 内部用户接点 UB(0) ~ UB(15) 可用作位单位的变量。
- 接通电源时，UB(0) ~ UB(15) 会清零。  
无法使用 EEPROM 等保存数据。
- 需设定开始数据时，请创建将开始数据设定为 UB(0) ~ UB(15) 的程序。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
UBw	内部用户接点 (字访问)	0 ~ 255	0	-	无符号 单字	R/W

- 内部用户接点 (字访问) UBw 是指可将内部用户接点 UB(0) ~ UB(15) 用作字单位变量的功能。
- 内部用户接点 UB(0) ~ UB(15) 配置在低位字节。高位字节的数据读取为零，写入高位字节时将忽略该设定。

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
功能	UB (15)	UB (14)	UB (13)	UB (12)	UB (11)	UB (10)	UB (9)	UB (8)	UB (7)	UB (6)	UB (5)	UB (4)	UB (3)	UB (2)	UB (1)	UB (0)



### 使用注意事项

---

DriveProgramming 功能的程序停止时，内部用户接点将保持程序停止前的状态。程序再次启动时，将从保持的数据起开始。

---

## 5-2 输入输出端子变量

下面对 DriveProgramming 功能使用变频器输入端子、输出端子、继电器输出、模拟输入端子、模拟输出端子用的变量进行说明。请用作变频器的周边设备与 DriveProgramming 功能的接口。

### 输入端子变量 X(00) ~ X(10)

在变频器的输入端子中设定通用输入 MI1 ~ MI11 时，可用作 DriveProgramming 功能的输入端子变量 X(00) ~ X(10)。

使用 Xw 变量时，也可将输入端子变量 X(00) ~ X(10) 用作字访问数据。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
X(00) ~ (10)	输入端子变量 (位访问)	0: OFF 1: ON	0	-	位	R

- 以位为单位将输入端子 1 ~ 9、A、B 端子的状态读取至变量。该变量为只读。
- 在输入端子 1 ~ 9、A、B 选择 (CA-01 ~ CA-11) 中设定“86 ~ 96: 通用输入 (MI1 ~ MI11)”时，会将输入端子 1 ~ 9、A、B 的状态作为 DriveProgramming 的输入端子变量 X(00) ~ X(10) 进行获取。输入端子变量的编号 X(00) ~ X(10) 根据设定的通用输入 MI1 ~ 11 的编号顺序进行设定，而不是根据 1 ~ 9、A、B 的端子编号进行设定。

功能变量	输入端子 CA-01 ~ CA-11
X(00)	86: MI1
X(01)	87: MI2
X(02)	88: MI3
X(03)	89: MI4
X(04)	90: MI5
X(05)	91: MI6
X(06)	92: MI7
X(07)	93: MI8
X(08)	94: MI9
X(09)	95: MI10
X(10)	96: MI11

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Xw	输入端子变量 (字访问)	0 ~ 65535	0	-	无符号 单字	R

- 输入端子变量 (字访问)Xw 是指可将输入端子变量 X(00) ~ X(10) 用作字单位变量的功能。该变量为只读。
- 输入端子变量 X(00) ~ X(10) 配置至位 0 ~ 10, 位 11 以后及不使用的输入端子变量读取为零。

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
功能	X (15)	X (14)	X (13)	X (12)	X (11)	X (10)	X (9)	X (8)	X (7)	X (6)	X (5)	X (4)	X (3)	X (2)	X (1)	X (0)



### 使用注意事项

- DriveProgramming 功能的程序停止时, 输入端子变量不会保持状态, 而是始终更新输入端子的状态。
- 输入端子 1 ~ 9、A, B 选择 MI1 ~ 11(通用输入 1 ~ 11) 时, 智能输入端子选择 CA-21 ~ CA-31 仍可选择 NO(a 接点)、NC(b 接点)。

## 输出端子变量 Y(00) ~ Y(06)

在变频器的输出端子中设定通用输出 MO1 ~ MO7 时, 可用作 DriveProgramming 功能的输出端子 Y(00) ~ Y(06)。

使用 Yw 变量时, 也可将输出端子变量 Y(00) ~ Y(06) 用作字访问数据。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Y(00) ~ Y(06)	输出端子变量 (位访问)	0: OFF 1: ON	0	-	位	R/W

- 可以位为单位将输出端子 11 ~ 15 端子、继电器输出 (16, AL) 的状态作为变量进行控制。
- 在输出端子 11 ~ 15 选择 (CC-01 ~ CC-05) 及继电器输出 (16, AL) 选择 (CC-06、CC-07) 中设定 “69 ~ 75: MO1 ~ MO7(通用输出)” 时, 可将输出端子 11 ~ 15 及继电器输出 (16、AL) 作为 DriveProgramming 功能的输出端子变量 Y(00) ~ Y(06) 进行控制。输出端子变量的编号 Y(00) ~ Y(06) 根据设定的通用输出 MO1 ~ MO7 的编号顺序进行设定, 而不是根据 11 ~ 15、16、AL 的端子编号进行设定。

功能变量	输出端子 CC-01 ~ CC-07
Y(00)	69: MO1
Y(01)	70: MO2
Y(02)	71: MO3
Y(03)	72: MO4
Y(04)	73: MO5
Y(05)	74: MO6
Y(06)	75: MO7

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Yw	输出端子变量 (字访问)	0 ~ 65535	0	-	无符号 单字	R/W

- 输出端子变量 (字访问) Yw 是指可将输出端子变量 Y(00) ~ Y(06) 用作字单位变量的功能。
- 输出端子变量 Y(00) ~ Y(06) 配置至位 0 ~ 6。位 7 以后及不使用的输出端子变量的数据读取为零，写入了数据时将忽略该设定。

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
功能	Y (15)	Y (14)	Y (13)	Y (12)	Y (11)	Y (10)	Y (9)	Y (8)	Y (7)	Y (6)	Y (5)	Y (4)	Y (3)	Y (2)	Y (1)	Y (0)



### 安全要点

DriveProgramming 功能的程序停止时，由 DriveProgramming 控制的输出端子功能将保持程序停止前的状态。

因此，请采用通过 DriveProgramming 启动信号和报警 (跳闸) 信号检出变频器

DriveProgramming 功能的程序停止，安全停止变频器周边设备的接线。



### 使用注意事项

- DriveProgramming 功能的程序停止时，输出端子变量将保持程序停止前的状态。程序再次启动时，将从保持的数据起开始。  
但未设定 DriveProgramming 功能的通用输出 MO1 ~ MO7 的输出与程序无关，作为变频器的输出端子进行控制。
- 在多个任务中使用同个端子时，最后执行的任务的输出状态为有效。考虑控制的稳定性，不建议使用多个任务进行控制。
- 在输出端子 11 ~ 15 及继电器输出 16、AL 选择中选择 MO1 ~ MO7(通用输出 1 ~ 7) 时，智能输出端子选择 CC-11 ~ CC-15、继电器输出 (16、AL) 接点选择 (CC-06、CC-07) 仍可选择 NO(a 接点)、NC(b 接点)。

## 模拟输入端子变量 XA(0) ~ XA(2)

可将变频器的频率指令输入 ( 模拟电压输入 )Ai1 端子、频率指令输入 ( 模拟电流输入 )Ai2 端子用作 DriveProgramming 功能的模拟输入端子变量 XA(0)、XA(1)。

此外, 也可将频率指令输入辅助 ( 模拟电压输入 )Ai3 端子用作 DriveProgramming 功能的模拟输入端子变量 XA(2)。

无论参数的设定如何, 始终可监控模拟输入的状态。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
XA(0)	0 ~ 10V / 0 ~ 20mA (Ai1 端子)	0 ~ 10,000	0	0.01%	无符号 单字	R
XA(1)	0 ~ 10V / 0 ~ 20mA (Ai2 端子)					
XA(2)	-10 ~ 10V (Ai3 端子)	-10,000 ~ 10,000			带符号 单字	

- 模拟输入端子变量 XA(0)、XA(1) 为无符号单字变量, XA(2) 为带符号单字变量。该变量为只读。
- 将模拟输入的最大输入 10V 及 20mA 作为 100%, 以 0.01% 为单位进行显示。
- 使用下述变频器本体的多个参数, 模拟输入端子可设定变频器功能。请使用各参数选择功能。用作仅限 DriveProgramming 功能的模拟输入时, 请勿使用以下参数设定模拟输入端子。  
AA101, AH-51, AH-70, AA101, AA102, bA110, Ad-01
- 模拟输入的调整请使用变频器本体的参数 (Ai1: Cb-03 ~ Cb-07、Ai2: Cb-13 ~ Cb-17、Ai3: Cb-23 ~ Cb-26) 进行操作。



### 使用注意事项

DriveProgramming 功能的程序停止时, 模拟输入端子变量不会保持状态, 而是始终更新输入端子的状态。

## 模拟输出端子变量 YA(0) ~ YA(2)

可将变频器的数字输出 (PWM 输出)FM 端子、模拟输出 (电压输出)Ao1 端子用作 DriveProgramming 功能的模拟输出端子变量 YA(0) ~ YA(1)。

3G3RX2 系列也可将模拟输出 (电流输出)Ao2 端子用作 DriveProgramming 功能的模拟输出端子变量 YA(2)。

无论参数的设定如何, 始终可监控模拟输出的状态。

通过 DriveProgramming 功能控制模拟输出时, 请使用变频器本体的参数 FM 端子输出选择 (Cd-03)、Ao1 端子输出选择 (Cd-04)、Ao2 端子输出选择 (Cd-05) 设定 DriveProgramming。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
YA(0)	FM 端子	0 ~ 10,000	0	0.01%	无符号 单字	R/W
YA(1)	Ao1 端子					
YA(2)	Ao2 端子					

- 模拟输出端子变量 YA(0) ~ YA(2) 为无符号单字变量。
- 将模拟输出的最大输出负载率 100% 及 10V、20mA 作为 100%，以 0.01% 为单位进行设定。
- 使用变频器本体的参数 FM 端子输出选择 (Cd-03)、Ao1 端子输出选择 (Cd-04)、Ao2 端子输出选择 (Cd-05) 设定 DriveProgramming 时，可通过 DriveProgramming 功能控制模拟输出端子。参数中未设定 DriveProgramming 时，也可监控模拟输出端子的状态。

功能变量	参数设定
YA(0)	在 FM 端子输出选择 (Cd-03) 中设定 0 ~ 65535 (d, F- 代码的寄存器编号)
YA(1)	在 Ao1 端子输出选择 (Cd-04) 中设定 0 ~ 65535 (d, F- 代码的寄存器编号)
YA(2)	在 Ao2 端子输出选择 (Cd-05) 中设定 0 ~ 65535 (d, F- 代码的寄存器编号)

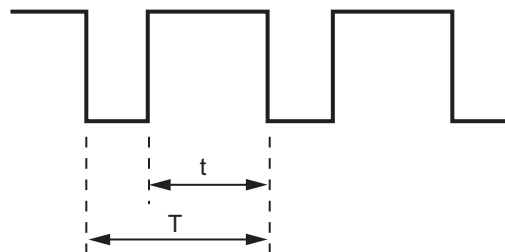
- 模拟输出的调整请使用变频器本体的参数 (Cd-14、Cd-24、Cd-34、Cd-23、Cd-33) 进行操作。



### 使用注意事项

- DriveProgramming 功能的程序停止时，模拟输出端子变量将保持程序停止前的状态。程序再次启动时，将从保持的数据起开始。但未设定 DriveProgramming 的输出与程序无关，作为变频器的模拟输出进行控制。
- 数字输出 (PWM 输出) FM 端子的输出使用 PWM 信号进行输出。  
以 0.00 ~ 100.00% 的可变值输出 6.4ms 周期的脉冲宽度 (负载率 t/T)。

PWM 输出



周期 T: 恒定 (6.4ms)

负载率 t/T: 变化

## 5-3 定时器变量

下面对 DriveProgramming 功能的定时器控制指令使用的定时器变量进行说明。

### 定时器变量 TD(0) ~ TD(15)

DriveProgramming 功能的定时器控制指令使用的定时器输出接点。

使用定时器控制指令的“delay on/off”指令、“timer set”指令指定定时器输出接点时，与指定定时器输出接点对应的定时器计数器变量将开始计数。达到指定计数值时，输出定时器输出接点。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
TD(0) ~ TD(15)	定时器输出接点 (位访问)	0: OFF 1: ON	0	—	位	R

- 定时器输出接点 TD(0) ~ TD(15) 是以位为单位的变量。该变量为只读。
- 程序启动时，定时器输出接点 TD(0) ~ TD(15) 会清零。
- 定时器输出接点 TD(0) ~ TD(15) 在使用“delay on/off”指令及“timer set”指令指定时开始动作。对应的定时器计数器变量清零的同时，TD(0) ~ TD(15) 设定为“0(OFF)”。定时器计数器变量达到指定计数值时设定为“1(ON)”，并保持该状态。“timer off”指令执行时，TD(0) ~ TD(15) 的状态会清零。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
TDw	定时器输出接点 (字访问)	0 ~ 255	0	—	无符号 单字	R

- 定时器输出接点 (字访问) TDw 是指可将定时器输出接点 TD(0) ~ TD(15) 用作字单位变量的功能。该变量为只读。
- 定时器输出接点 TD(0) ~ TD(15) 配置在低位字节。高位字节的数据读取为零。

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
功能	TD (15)	TD (14)	TD (13)	TD (12)	TD (11)	TD (10)	TD (9)	TD (8)	TD (7)	TD (6)	TD (5)	TD (4)	TD (3)	TD (2)	TD (1)	TD (0)



#### 使用注意事项

DriveProgramming 功能的程序停止时，定时器计数器变量和定时器输出接点会清零，而不会保持状态。



## 定时器计数器 TC(0) ~ TC(15)

DriveProgramming 功能的定时器控制指令使用的定时器计数器变量。可参照定时器的计数值。  
不通过定时器控制指令使用时，作为自由运行计数器动作。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
TC(0) ~ TC(15)	定时器计数器	0 ~ 2147483646	0	10ms	无符号 双字	R

- 定时器计数器 TC(0) ~ TC(15) 为无符号双字变量。
- 不通过定时器控制指令使用时，在程序开始的同时，作为以 10ms 为周期进行计数递增的自由运行定时器计数器动作。
- 定时器开始指令“timer set”、延迟动作指令“delayon/off”执行时，作为指定定时器接点输出用的定时器计数器动作。执行指令时将清零，然后在计数递增至指定时间后停止。
- 定时器停止指令“timer off”执行时会清零。



### 使用注意事项

DriveProgramming 功能的程序停止时，定时器计数器变量和定时器输出接点会清零，而不会保持状态。

## 5-4 变频器设定变量

下面对设定变频器频率指令及加减速时间的变量进行说明。  
请用于通过 DriveProgramming 功能的程序控制变频器。

### 频率指令变量 SET-Freq

通过 DriveProgramming 功能直接控制频率指令时，请在变频器本体的 AA101 主速指令选择中设定“14：程序功能”。频率指令变量 SET-Freq 将生效。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
SET-Freq	频率指令变量	0 ~ 59,000	0	0.01Hz	无符号 单字	R/W

- 频率指令变量 SET-Freq 为无符号单字变量。
- 仅在主速指令选择中设定“14：程序功能”时有效。  
在第 1 / 第 2 主速指令选择 (AA101/AA201) 中设定。  
在输入端子 1 ~ 9、A、B 中分配“24：SET 第 2 控制”进行切换。
- 接通电源时，频率指令变量 SET-Freq 会清零。  
无法使用 EEPROM 等保存数据。
- 需设定开始数据时，请创建将开始数据设定为 SET-Freq 的程序。
- 变频器实际可输出的频率为第 1 最低频率 (Hb130) ~ 最高频率。  
设定了超出范围的数据时，动作如下。
  - 设定小于第 1 最低频率 (Hb130)  
控制方式选择 (AA121) 设定为“09：0Hz 区无传感器矢量控制”、“10：带传感器矢量控制 (IM)”  
时，第 1 最低频率 (Hb130) 无效，输出指定频率。
  - 设定超出最高频率  
通过第 1 / 第 2 最高频率 (Hb105/Hd105/Hb205/Hd205) 中设定的频率限制频率指令。
- 频率指令变量 SET-Freq 设定的频率指令也可使用变频器本体的主速指令 ( 监控 + 设定 )(FA-01) 进行监控。



#### 使用注意事项

DriveProgramming 功能的程序停止时，频率指令变量将保持程序停止前的状态。程序再次启动时，将从保持的数据起开始。

## 加速、减速时间变量 ACCEL、DECEL

通过 DriveProgramming 功能直接控制频率指令的加减速时间时，请在变频器本体的加减速时间输入种类 (AC-01) 中设定“04: DriveProgramming”。

加速时间变量 ACCEL 和减速时间变量 DECEL 将生效。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
ACCEL	加速时间变量	0 ~ 360,000	参数设定	0.01 秒	无符号 双字	R/W
DECEL	减速时间变量					

- 加速时间变量 ACCEL 和减速时间变量 DECEL 为无符号双字变量。
- 仅在加减速时间输入种类 (AC-01) 中设定“04: DriveProgramming”时，加速时间变量 ACCEL 和减速时间变量 DECEL 才会生效。
- 仅在电源接通后的程序首次启动时，变频器本体的加减速时间的参数设定将设定为 DriveProgramming 功能的加速时间变量 ACCEL 和减速时间变量 DECEL。

根据第 1 / 第 2 控制的选择，将设定第 1 / 第 2 加速时间 1(AC120/AC220)、第 1 / 第 2 减速时间 1(AC122/AC222) 的加减速时间。

- 加速时间变量 ACCEL 和减速时间变量 DECEL 中设定的数据不会保存至 EEPROM。
- 加减速时间的内部处理以 40ms 为周期。即使在程序中变更加速时间变量 ACCEL 和减速时间变量 DECEL，反映仍需最多 40ms 的时间。
- 程序中的加速时间变量 ACCEL 和减速时间变量 DECEL 设定了超出数据范围的数据或 3G3RX2 系列设定为“0”时，该设定数据不会反映，并按照之前设定的数据进行动作。



### 使用注意事项

DriveProgramming 功能的程序停止时，加速时间变量和减速时间变量将保持程序停止前的状态。程序再次启动时，将从保持的数据起开始。

## 5-5 变频器监控变量

可将变频器内部的监控功能和状态监控用作 DriveProgramming 功能的变量。

关于各监控功能的详情，请参阅 □《高功能型通用变频器 RX2 系列用户手册 (SBCE-CN5-437)》。

但数据单位可能会与变频器本体不同，因此请务必使用以下单位。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
FM	输出频率监控 (dA-01)	0 ~ 59000	—	0.01Hz	无符号 单字	R

可监控变频器的输出频率。监控内容相当于输出频率监控 (dA-01)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Iout	输出电流监控 (dA-02)	0 ~ 65535	—	0.1%	无符号 单字	R

可监控变频器的输出电流。监控内容相当于输出电流监控 (dA-02)。以变频器的额定电流为 100%，以 0.1% 为单位进行显示。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Dir	运行方向监控 (dA-03)	00: 停止中 01: 0Hz 输出中 02: 正转中 03: 反转中	—	—	无符号 单字	R

可监控变频器的运转方向。监控内容相当于运行方向监控 (dA-03)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
PID-FB	PID 1 反馈数据 1 监控 (db-30)	-10000 ~ 10000	—	0.01%	带符号 双字	R

可监控 PID 反馈值。监控内容相当于 PID1 反馈值监控 (db-30)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
F-CNV	输出频率转换监控 (dA-06)	0 ~ 5900000	—	0.01	无符号 双字	R

可监控输出频率 (换算后)。监控内容相当于输出频率监控 (换算后)(dA-06)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Tmon	输出转矩监控 (dA-17)	-10000 ~ 10000	—	%	带符号 单字	R

可监控输出转矩。监控内容相当于输出转矩监控 (dA-17)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Vout	输出电压监控 (dA-18)	0 ~ 8000	—	0.1V	无符号 单字	R

可监控输出电压。监控内容相当于输出电压监控 (dA-18)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Power	输入功率监控 (dA-30)	0 ~ 60000 ( ~ 132kW) 0 ~ 20000 (160kW ~ )	—	0.1kW	无符号 单字	R

可监控输入功率。监控内容相当于输入功率监控 (dA-30)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
Run-Time	累计运行时间监控 (dC-22)	0 ~ 1000000	—	时间	无符号 双字	R

可监控累计运行时间。监控内容相当于累计运行时间监控 (dC-22)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
ON-Time	累计电源 ON 时间 (dC-24)	0 ~ 1000000	—	时间	无符号 双字	R

可监控电源 ON 时间。监控内容相当于累计电源 ON 时间 (dC-24)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
PlsCnt	脉冲计数器监控 (dA-28)	0 ~ 2147483647	—	—	带符号 双字	R

DriveProgramming 功能的用户监控 UMon(0) ~ UMon(4) 为带符号双字变量。

DriveProgramming 功能的程序中，在用户监控 UMon(0) ~ UMon(4) 中设定数据时，可在变频器本体的用户监控 (db-08、db-10、db-12、db-14、db-16) 中显示。请在外部显示程序的运算状态时使用。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
POS	当前位置监控 (dA-20)	AA121=10 且 AA123=03 时 -2147483648 ~ 2147483647 上述以外 -536870912 ~ 536870911	—	—	带符号 双字	R

可监控当前位置。监控内容相当于当前位置监控 (dA-20)。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
ERR-CNT	跳闸次数监控 (dE-01)	0 ~ 65535	—	次	无符号 单字	R

可监控变频器的异常累计次数。监控内容相当于异常次数监控。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
ERR (1) ~ (10)	跳闸监控 1 原因 ~ 跳闸监控 10 原因	0 ~ 255	—	—	无符号 单字	R

可监控变频器最近 10 次的跳闸内容。监控内容相当于异常监控 1 ~ 10。该变量为只读。跳闸内容为 LCD 操作器跳闸履历中显示的跳闸原因 (E\*\*\*) 的数值部分 (\*\*\*)。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
RETRY (1) ~ (10)	重试监控 1 原因 ~ 重试监控 10 原因	0 ~ 255	—	—	无符号 单字	R

可监控变频器最近 10 次的重试原因。监控内容相当于重试原因监控 1 ~ 10。该变量为只读。内容为 LCD 操作器重试履历中显示的重试原因 (r\*\*\*) 的数值部分 (\*\*\*)。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
DCV	直流电压监控 (dA-40)	0 ~ 10000	—	—	无符号 单字	R

可监控变频器的内部直流电压。监控内容相当于直流电压监控 (dA-40)。该变量为只读。

功能变量	说明	数据范围	初始数据	单位	数据大小	R/W
STATUS	变频器状态监控	0: 初始化 1: 接地检测 2: 停止 3: 运行待机 4: 运行准备 5: 运行 6: 停止待机 7: 重试待机 8: 重试	—	—	无符号 单字	R

可用于监控变频器的状态信息。

## 5-6 输入变量

可通过 DriveProgramming 功能的程序执行输入端子中可设定的功能。下述变量分别对应输入端子中可设定的功能。

各变量中设定“1: ON”时, 各功能将在输入端子 ON 的同时动作。

各变量中设定“0: OFF”时, 将停止该动作。即使不在输入端子选择 CA-01 ~ CA-11 中设定各功能, 仍可在程序中执行。下表的参考中记述了变频器本体的各功能设定数据。关于各功能的详情, 请参阅

□ 《高功能型通用变频器 RX2 系列用户手册 (SBCE-CN5-437)》进行确认。

### ● 示例

在 FW(正转) 变量中设定“1”时, 将执行正转运行指令。

FW: =1 变频器开始正转运行。

FW: =0 变频器停止正转运行, 并开始减速。



### 使用注意事项

- DriveProgramming 功能的程序停止时, 输入变量会清零, 而不会保持状态。
- FW(正转) 变量及 RV(反转) 变量仅在变频器本体的运行指令选择(AA111) 设定为“00: FW/RV 端子”时有效。使用其它设定时不会动作。
- 变频器接通电源后, 即使在 FW(正转) 变量及 RV(反转) 变量中设定“1”, 该设定仍会被忽略, 不会执行正转运行及反转运行。请设定为“0”后再设定为“1”。为了避免该动作, 创建程序时请仅在电源接通后使用“wait”指令等留出 1 秒的等待时间。
- 输入端子选择中设定的变频器本体的输入端子功能与输入变量为逻辑 OR 关系。

功能变量	说明	R/W	参考
no	无分配	R/W	CA-01 ~ CA-11 = 0
FW	正转	R/W	CA-01 ~ CA-11 = 1
RV	反转	R/W	CA-01 ~ CA-11 = 2
CF1	多段速 1	R/W	CA-01 ~ CA-11 = 3
CF2	多段速 2	R/W	CA-01 ~ CA-11 = 4
CF3	多段速 3	R/W	CA-01 ~ CA-11 = 5
CF4	多段速 4	R/W	CA-01 ~ CA-11 = 6
SF1 ~ SF7	多段速位 1 ~ 7	R/W	CA-01 ~ CA-11 = 7 ~ 13
ADD	频率加法	R/W	CA-01 ~ CA-11 = 14
SCHG	指令切换	R/W	CA-01 ~ CA-11 = 15
STA	3 线启动	R/W	CA-01 ~ CA-11 = 16
STP	3 线停止	R/W	CA-01 ~ CA-11 = 17
F/R	3 线正反	R/W	CA-01 ~ CA-11 = 18
AHD	模拟量指令保持	R/W	CA-01 ~ CA-11 = 19
FUP	远程操作增速	R/W	CA-01 ~ CA-11 = 20
FDN	远程操作减速	R/W	CA-01 ~ CA-11 = 21
UDC	远程操作数据清除	R/W	CA-01 ~ CA-11 = 22
F-OP	强制指令切换	R/W	CA-01 ~ CA-11 = 23
SET	第 2 控制	R/W	CA-01 ~ CA-11 = 24
RS	复位	R/W	CA-01 ~ CA-11 = 28
JG	点动	R/W	CA-01 ~ CA-11 = 29
DB	外部直流制动	R/W	CA-01 ~ CA-11 = 30
2CH	2 段加减速	R/W	CA-01 ~ CA-11 = 31

功能变量	说明	R/W	参考
FRS	自由运行停止	R/W	CA-01 ~ CA-11 = 32
EXT	外部异常	R/W	CA-01 ~ CA-11 = 33
USP	复电重启防止	R/W	CA-01 ~ CA-11 = 34
CS	商用切换	R/W	CA-01 ~ CA-11 = 35
SFT	软件锁	R/W	CA-01 ~ CA-11 = 36
BOK	制动确认	R/W	CA-01 ~ CA-11 = 37
OLR	过载限制切换	R/W	CA-01 ~ CA-11 = 38
KHC	累计输入功率清除	R/W	CA-01 ~ CA-11 = 39
OKHC	累计输出功率清除	R/W	CA-01 ~ CA-11 = 40
PID	PID1 无效	R/W	CA-01 ~ CA-11 = 41
PIDC	PID1 积分复位	R/W	CA-01 ~ CA-11 = 42
PID2	PID2 无效	R/W	CA-01 ~ CA-11 = 43
PIDC2	PID2 积分复位	R/W	CA-01 ~ CA-11 = 44
PID3	PID3 无效	R/W	CA-01 ~ CA-11 = 45
PIDC3	PID3 积分复位	R/W	CA-01 ~ CA-11 = 46
PID4	PID4 无效	R/W	CA-01 ~ CA-11 = 47
PIDC4	PID4 积分复位	R/W	CA-01 ~ CA-11 = 48
SVC1 ~ SVC4	PID1 多段目标值 1 ~ 4	R/W	CA-01 ~ CA-11 = 51 ~ 54
PRO	PID 增益切换	R/W	CA-01 ~ CA-11 = 55
PIO1	PID 输出切换	R/W	CA-01 ~ CA-11 = 56
PIO2	PID 输出切换 2	R/W	CA-01 ~ CA-11 = 57
SLEP	SLEEP 条件成立	R/W	CA-01 ~ CA-11 = 58
WAKE	WAKE 条件成立	R/W	CA-01 ~ CA-11 = 59
TL	转矩限制有效	R/W	CA-01 ~ CA-11 = 60
TRQ1	转矩限制切换 1	R/W	CA-01 ~ CA-11 = 61
TRQ2	转矩限制切换 2	R/W	CA-01 ~ CA-11 = 62
PPI	PPI 控制切换	R/W	CA-01 ~ CA-11 = 63
CAS	控制增益切换	R/W	CA-01 ~ CA-11 = 64
SON	伺服 ON	R/W	CA-01 ~ CA-11 = 65
FOC	预备励磁	R/W	CA-01 ~ CA-11 = 66
ATR	转矩控制有效	R/W	CA-01 ~ CA-11 = 67
TBS	转矩偏置有效	R/W	CA-01 ~ CA-11 = 68
ORT	定位	R/W	CA-01 ~ CA-11 = 69
LAC	LAD 取消	R/W	CA-01 ~ CA-11 = 71
PCLR	位置偏差清除	R/W	CA-01 ~ CA-11 = 72
STAT	脉冲串位置 指令输入许可	R/W	CA-01 ~ CA-11 = 73
PUP	位置偏置加法	R/W	CA-01 ~ CA-11 = 74
PDN	位置偏置减法	R/W	CA-01 ~ CA-11 = 75
CP1 ~ CP4	位置指令选择 1 ~ 4	R/W	CA-01 ~ CA-11 = 76 ~ 79
ORL	原点限制信号	R/W	CA-01 ~ CA-11 = 80
ORG	原点复位启动信号	R/W	CA-01 ~ CA-11 = 81
FOT	正转驱动停止	R/W	CA-01 ~ CA-11 = 82
ROT	反转驱动停止	R/W	CA-01 ~ CA-11 = 83
SPD	速度位置切换	R/W	CA-01 ~ CA-11 = 84
PSET	位置数据预置	R/W	CA-01 ~ CA-11 = 85
MI1 ~ MI11	通用输入 1 ~ 11	R/W	CA-01 ~ CA-11 = 86 ~ 96
PCC	脉冲计数器清除	R/W	CA-01 ~ CA-11 = 97
ECOM	EzCOM 启动	R/W	CA-01 ~ CA-11 = 98



功能变量	说明	R/W	参考
PRG	EzSQ 程序开始	R/W	CA-01 ~ CA-11 = 99
HLD	加减速停止	R/W	CA-01 ~ CA-11 = 100
REN	运行许可信号	R/W	CA-01 ~ CA-11 = 101
DISP	显示固定	R/W	CA-01 ~ CA-11 = 102
PLA	脉冲串输入 A	R/W	CA-01 ~ CA-11 = 103
PLB	脉冲串输入 B	R/W	CA-01 ~ CA-11 = 104
EMF	紧急时强制运行	R/W	CA-01 ~ CA-11 = 105
COK	接触器检测信号	R/W	CA-01 ~ CA-11 = 107
PLZ	脉冲串输入 Z	R/W	CA-01 ~ CA-11 = 109
TCH	示教信号	R/W	CA-01 ~ CA-11 = 110

## 5-7 输出变量

可在 DriveProgramming 功能的程序中使用输出端子中可设定的功能。下述变量分别对应输出端子中可设定的功能。

在各变量中设定“1: ON”及“0: OFF”时，各输出端子功能与将输出端子设定为 ON 及 OFF 时相同。即使不在输出端子选择 (CC-01 ~ CC-05) 及继电器输出 (16C、AL) 选择 (CC-06、CC-07) 中设定各功能，仍可在程序中进行监控。下表的参考中记述了变频器本体的各功能设定数据。关于各功能的详情，请参阅 □《高功能型通用变频器 RX2 系列用户手册 (SBCE-CN5-437)》。



### 使用注意事项

DriveProgramming 功能的程序停止时，输出变量不会保持状态，而是始终更新各功能的状态。

功能变量	说明	R/W	参考
-	无分配	R	CC-01 ~ CC-07 = 0
RUN	运行中	R	CC-01 ~ CC-07 = 1
FA1	恒速到达时	R	CC-01 ~ CC-07 = 2
FA2	设定频率以上	R	CC-01 ~ CC-07 = 3
FA3	设定频率一致	R	CC-01 ~ CC-07 = 4
FA4	设定频率以上 2	R	CC-01 ~ CC-07 = 5
FA5	设定频率一致 2	R	CC-01 ~ CC-07 = 6
IRDY	运行准备完成	R	CC-01 ~ CC-07 = 7
FWR	正转运行中	R	CC-01 ~ CC-07 = 8
RVR	反转运行中	R	CC-01 ~ CC-07 = 9
FREF	频率指令面板	R	CC-01 ~ CC-07 = 10
REF	运行指令面板	R	CC-01 ~ CC-07 = 11
SETM	第 2 控制选择中	R	CC-01 ~ CC-07 = 12
OPO	选项输出	R	CC-01 ~ CC-07 = 16
AL	报警信号	R	CC-01 ~ CC-07 = 17
MJA	重故障信号	R	CC-01 ~ CC-07 = 18
OTQ	过转矩	R	CC-01 ~ CC-07 = 19
IP	瞬时停电中	R	CC-01 ~ CC-07 = 20
UV	欠电压中	R	CC-01 ~ CC-07 = 21
TRQ	转矩限制中	R	CC-01 ~ CC-07 = 22
IPS	停电减速中	R	CC-01 ~ CC-07 = 23
RNT	RUN 超时	R	CC-01 ~ CC-07 = 24
ONT	电源 ON 超时	R	CC-01 ~ CC-07 = 25
THM	电子热敏保护警告 (电机)	R	CC-01 ~ CC-07 = 26
THC	电子热敏保护警告 (变频器)	R	CC-01 ~ CC-07 = 27
WAC	电容器寿命预警	R	CC-01 ~ CC-07 = 29
WAF	风扇寿命预警	R	CC-01 ~ CC-07 = 30
FR	运行指令信号	R	CC-01 ~ CC-07 = 31
OHF	散热片过热预警	R	CC-01 ~ CC-07 = 32
LOC	低电流信号	R	CC-01 ~ CC-07 = 33
LOC2	低电流信号 2	R	CC-01 ~ CC-07 = 34
OL	过载预警	R	CC-01 ~ CC-07 = 35
OL2	过载预警 2	R	CC-01 ~ CC-07 = 36
BRK	制动释放	R	CC-01 ~ CC-07 = 37
BER	制动异常	R	CC-01 ~ CC-07 = 38

功能变量	说明	R/W	参考
CON	接触器控制	R	CC-01 ~ CC-07 = 39
ZS	0Hz 检测信号	R	CC-01 ~ CC-07 = 40
DSE	速度偏差过大	R	CC-01 ~ CC-07 = 41
PDD	位置偏差过大	R	CC-01 ~ CC-07 = 42
POK	定位完成	R	CC-01 ~ CC-07 = 43
PCMP	脉冲计数 比较匹配输出	R	CC-01 ~ CC-07 = 44
OD	PID 偏差过大	R	CC-01 ~ CC-07 = 45
FBV	PID 反馈比较	R	CC-01 ~ CC-07 = 46
OD2	PID2 偏差过大	R	CC-01 ~ CC-07 = 47
FBV2	PID2 反馈比较	R	CC-01 ~ CC-07 = 48
NDc	通信断线	R	CC-01 ~ CC-07 = 49
Ai1Dc ~ Ai3Dc	模拟断线 Ai1 ~ 3	R	CC-01 ~ CC-07 = 50 ~ 52
WCAi1 ~ WCAi3	窗口比较器 Ai1 ~ 3	R	CC-01 ~ CC-07 = 56 ~ 58
LOG1 ~ LOG7	逻辑运算结果 1 ~ 7	R	CC-01 ~ CC-07 = 62 ~ 68
MO1 ~ MO7	通用输出 1 ~ 7	R	CC-01 ~ CC-07 = 69 ~ 75
EMFC	强制运行中信号	R	CC-01 ~ CC-07 = 76
EMBP	旁路模式中信号	R	CC-01 ~ CC-07 = 77
LBK	LCD 操作器电池耗尽	R	CC-01 ~ CC-07 = 80
OVS	受电过电压	R	CC-01 ~ CC-07 = 81
AC0 ~ AC3	报警代码位 0 ~ 3	R	CC-01 ~ CC-07 = 84 ~ 87
OD3	PID3 偏差过大	R	CC-01 ~ CC-07 = 89
FBV3	PID3 反馈比较	R	CC-01 ~ CC-07 = 90
OD4	PID4 偏差过大	R	CC-01 ~ CC-07 = 91
FBV4	PID4 反馈比较	R	CC-01 ~ CC-07 = 92
SSE	PID 软启动异常	R	CC-01 ~ CC-07 = 93



# 6

## DriveProgramming 指令

本章对 DriveProgramming 的指令进行介绍。

---

6-1 指令的种类 .....	6-2
6-2 指令的格式 .....	6-3
6-3 指令一览表 .....	6-4
6-4 程序控制指令 .....	6-10
6-5 算术运算指令和逻辑运算指令 .....	6-23
6-6 输入输出控制指令 .....	6-36
6-7 定时器控制指令 .....	6-48
6-8 参数控制指令 .....	6-54
6-9 变频器控制指令 .....	6-58

## 6-1 指令的种类

---

各指令的种类如下。

- 程序控制指令
- 算术运算指令和逻辑运算指令
- 输入输出控制指令
- 定时器控制指令
- 参数控制指令
- 变频器控制指令

## 6-2 指令的格式

各指令由指令及其自变量 (0 ~ 最多 5 个) 构成。

示例如下。

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
entry						表示程序的开头

无自变量时为空白。

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
wait	<待机时间>					按<待机时间> (×0.01)s 进行待机。

Wait 指令有 1 个自变量，输入待机时间 (变量或常数)。

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
while	<条件>					符合<条件时>执行< 指令组>。
	<指令组>					符合<条件>时的指令 组。
wend						While 循环后结束。

while 指令时，对自变量设定条件。(详情请参阅 □「条件 (P.6-6)」)。

在 while ~ wend 之间，记述符合<条件>时执行的指令组。

## 6-3 指令一览表

### 程序控制指令

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
entry						表示任务的开始。
end						表示任务的结束。
call	<子程序名>					分支至<子程序名>。
sub	<子程序名>					表示子程序的开始。
end sub						表示子程序的结束。
goto	<标签名>					无条件分支至 <标签名>。
ontrip goto	<标签名>					发生跳闸时，分支至 <标签名>。
if	<条件>			goto	<标签名>	符合条件时分支至 <标签名>。
if	<条件>			then		开始 if 结构。 符合<条件>时，将执行此指令后至 else 指令的<指令组 1>，然后跳至 endif 的下一步。 不符合<条件>时，将执行 else 指令后面至 endif 指令的 <指令组 2>， 然后跳至 endif 指令的下一步。
	<指令组 1>					符合<条件>时的指令组。
else						开始不符合<条件>时的指令。
	<指令组 2>					不符合<条件>时的指令组。
endif						结束 if 结构。



指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
select	<条件变量>					<条件变量>为<条件值>时，将执行 case 以下的指令。
case	<条件值 1 >					开始条件值的执行指令。
	<指令组 1 >					<条件值 1 >的指令组。
	[case 条件值 n] [<指令组 n >]					可使用 case 设定多个条件。
caseelse						<条件值>以外的指令开始
	<指令组>					<条件值>以外的指令组。
end select						结束 select case 语句。
for	<变量>	<开始值>	<结束值>	<增量值>		从<开始值>至<结束值>按<增分值>进行循环。
	<指令组>					重复指令组。
next						for 循环后结束。
while	<条件>					符合<条件时>执行<指令组>。
	<指令组>					符合<条件>时的指令组。
wend						While 循环后结束。
until	<条件>					执行<指令组>直至符合<条件>为止。
	<指令组>					不符合<条件>时执行的指令组。
loop						until 循环后结束。
wait	<待机时间>					按<待机时间>(×0.01)s 进行待机。
wait	<条件>					待机至符合<条件>为止。



### 参考

关于<条件>的格式，请参阅 「条件 (P.6-6)」。

## 条件

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
条件	<变量 1 / 常数>	=	<变量 2 / 常数>			<变量 1 / 常数>等于 <变量 2 / 常数>时为 TRUE
	<变量 1 / 常数>	<	<变量 2 / 常数>			<变量 1 / 常数>小于 <变量 2 / 常数>时为 TRUE
	<变量 1 / 常数>	< =	<变量 2 / 常数>			<变量 1 / 常数>为< 变量 2 / 常数>以下时 为 TRUE
	<变量 1 / 常数>	>	<变量 2 / 常数>			<变量 1 / 常数>大于 <变量 2 / 常数>时为 TRUE
	<变量 1 / 常数>	> =	<变量 2 / 常数>			<变量 1 / 常数>为< 变量 2 / 常数>以上时 为 TRUE
	<变量 1 / 常数>	<>	<变量 2 / 常数>			<变量 1 / 常数>不等 于<变量 2 / 常数>时 为 TRUE

## 算术运算指令和逻辑运算指令

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
算术运算	<变量 1>	:=	<变量 2 / 常数>			将<变量 2 / 常数>代 入<变量 1>。
	<变量 1>	:=	<变量 2 / 常数>	+	<变量 3 / 常数>	将<变量 2 / 常数>加 上<变量 3 / 常数>后 代入<变量 1>。
	<变量 1>	:=	<变量 2 / 常数>	-	<变量 3 / 常数>	将<变量 2 / 常数>减 去<变量 3 / 常数>后 代入<变量 1>。
	<变量 1>	:=	<变量 2 / 常数>	*	<变量 3 / 常数>	将<变量 2 / 常数>乘 以<变量 3 / 常数>后 代入<变量 1>。
	<变量 1>	:=	<变量 2 / 常数>	/	<变量 3 / 常数>	将<变量 2 / 常数>除 以<变量 3 / 常数>后 代入<变量 1>。
除后余数	<变量 1>	:=	<变量 2 / 常数>	mod	<变量 3 / 常数>	将<变量 2 / 常数>除 以<变量 3 / 常数>后 的余数代入<变量 1>。
绝对值	<变量 1>	:= abs	<变量 2 / 常数>			将<变量 2 / 常数>的 绝对值代入<变量 1>。

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
逻辑运算	<变量 1>	:=	<变量 2 / 常数>	and	<变量 3 / 常数>	对 <变量 2 / 常数> 和 <变量 3 / 常数> 进行逻辑与运算, 然后将结果代入 <变量 1>。
	<变量 1>	:=	<变量 2 / 常数>	or	<变量 3 / 常数>	对 <变量 2 / 常数> 和 <变量 3 / 常数> 进行逻辑或运算, 然后将结果代入 <变量 1>。
	<变量 1>	:=	<变量 2 / 常数>	xor	<变量 3 / 常数>	对 <变量 2 / 常数> 和 <变量 3 / 常数> 进行逻辑异或运算, 然后将结果代入 <变量 1>。
	<变量 1>	:= not	<变量 2 / 常数>			反转 <变量 2 / 常数> 的位, 然后将结果代入 <变量 1>。

## 增量和减量指令

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
增量	inc	<变量>				将 <变量> 的内容加 1。
减量	Dec	<变量>				将 <变量> 的内容减 1。

## 输入输出控制指令

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
输入端子变量代入	<变量>	:=	X(i)			将输入端子变量的信息代入 <变量>。 0=off 1=on
	<变量>	:=	Xw			以字为单位将输入端子变量的信息代入 <变量>。
输出端子变量输出	Y(**)	:=	<变量 / 常数>			以位为单位输出至输出端子变量。 0=off 1=on
	Yw	:=	<变量 / 常数>			以字为单位输出至输出端子变量。
输入变量输出	<输入变量>	:=	<变量 / 常数>			输出至输入变量。 0=off 1=on
输出变量代入	<变量>	:=	<输出变量>			将输出变量的状态代入 <变量>。 0=off 1=on
内部用户接点控制	<变量>	:=	UB(i)			将内部用户接点信息代入 <变量>。 0=off 1=on
	<变量>	:=	UBw			以字为单位将内部用户接点信息代入 <变量>。
	UB(**)	:=	<变量 / 常数>			以位为单位输出至内部用户接点。 0=off 1=on
	UBw	:=	<变量 / 常数>			以字为单位输出至内部用户接点。

## 定时器控制指令

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
delay	on / off	<变量 1 >	TD(i)	<变量 2 / 常数>		在指定定时器中设定 <变量 2 / 常数>后启动。 在<变量 2 / 常数>时间后， 将<变量 1 >设为 ON 或 OFF。 同时将 TD(i) 设为 ON。
timer set	TD(*)	<变量 / 常数>				在指定定时器中设定 <变量 / 常数>后启动。 在<变量 / 常数>时间后将 TD(i) 设为 ON。
timer off	TD(*)					停止指定定时器。

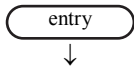
## 参数控制指令

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
ChgParam	<参数 >	:=	<变量 / 常数 >			将<参数>的内容改写为 <变量 / 常数>。
MonParam	<变量 >	:=	<参数 >			将<参数>的内容代入 <变量 >。

## 变频器控制指令

指令	自变量 1	自变量 2	自变量 3	自变量 4	自变量 5	内容
FW	:=	1				使变频器正转。
RV	:=	1				使变频器反转。
stop						减速停止变频器的运行。 ※ 变频器跳闸时起复位作用。
trip	<变量 / 常数>					发出与<变量 / 常数>对应的用户跳闸。
SET-Freq	:=	<变量 / 常数>				在变频器的频率指令变量中设定<变量 / 常数> (×0.01Hz)。
ACCEL	:=	<变量 / 常数>				在变频器的加速时间变量中设定<变量 / 常数> (×10ms)。
DECEL	:=	<变量 / 常数>				在变频器的减速时间变量中设定<变量 / 常数> (×10ms)。

## 6-4 程序控制指令


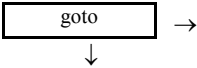
Entry		
指令	说明	自变量
	表示任务的开始。	---
格式		
流程图方式	文本语言方式	
	entry	
(注) 各任务的开头必须使用。		
End		
指令	说明	自变量
	表示任务的结束。	---
格式		
流程图方式	文本语言方式	
	end	
(注) 各任务的末尾必须使用。		
Call		
指令	说明	自变量
	分支至<子程序名>。	<b>子程序名:</b> 子程序名通过用户定义的名称或别名进行识别。
格式		
流程图方式	文本语言方式	
	call <子程序名>	
(注) 1. 子程序执行结束时, 执行 call 指令的下一个指令行。 2. 使用流程图方式在 call 指令中设定子程序名时, 请右击指令, 从菜单中选择 [ 设置子程序 ]。然后选择选项进行设定。		

Sub		
指令	说明	自变量
	表示子程序的开始。	---
格式		
流程图方式	文本语言方式	
	sub	
(注) 各子程序的开头必须使用。		
End Sub		
指令	说明	自变量
	表示子程序的结束。	---
格式		
流程图方式	文本语言方式	
	endsub	
(注) 各子程序的末尾必须使用。		

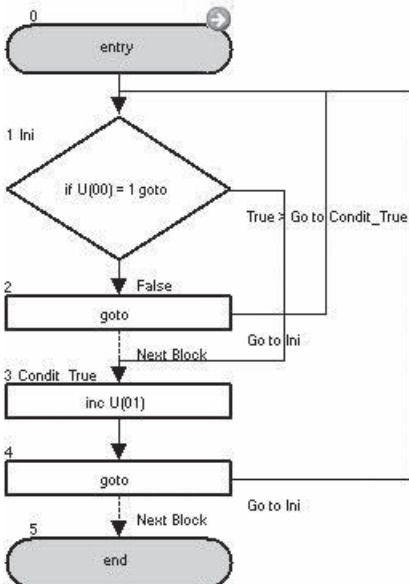
● 示例

流程图	文本
<b>Main</b>	<b>Main</b>
<pre> graph TD     0([entry]) --&gt; 1[SET-Freq := 6000]     1 --&gt; 2[UB(0) := 1]     2 --&gt; 3{if UB(0) = 1 then}     3 -- True --&gt; 4[call RunFW]     3 -- False --&gt; 5[else]     5 --&gt; 6[call RunRV]     6 --&gt; 7[endif]     7 --&gt; 8[stop]     8 --&gt; 9[wait RUN = 0]     9 --&gt; 10[UB(0) := not UB(0)]     10 --&gt; 11[goto]     11 --&gt; 12([end])     4 --&gt; 3     11 -- Go to Loop_ --&gt; 3     </pre>	<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>
<b>Main: RunFW</b>	<b>Main: RunFW</b>
<pre> graph TD     13([sub]) --&gt; 14[FW := 1]     14 --&gt; 15[wait X(01) = 1]     15 --&gt; 16([endsub])     </pre>	<pre> sub   FW := 1   wait X(01) = 1 endsub     </pre>
<b>Main: RunRV</b>	<b>Main: RunRV</b>
<pre> graph TD     17([sub]) --&gt; 18[RV := 1]     18 --&gt; 19[wait X(02) = 1]     19 --&gt; 20([endsub])     </pre>	<pre> sub   RV := 1   wait X(02) = 1 endsub     </pre>
区块 No.	动作
1, 2	将变频器的输出频率设为 60.00Hz，并将内部用户接点 (UB(0)) 设为 ON。
3 ~ 7	UB(0) 为 ON 时执行子程序 (RunFW)，UB(0) 为 OFF 时执行子程序 (RunRV)。
8 ~ 11	子程序结束时停止变频器，等待 RUN OFF，然后反转 UB(0)，返回 3。
13 ~ 16	使变频器正转，使用 wait 指令进行待机直至 X(01) 变为 ON。(子程序: RunFW)
17 ~ 20	使变频器反转，使用 wait 指令进行待机直至 X(02) 变为 ON。(子程序: RunRV)



Go To		
指令	说明	自变量
	无条件分支至<标签名>。	<b>标签名:</b> 用于识别任务内特定功能块而使用的名称
格式		
流程图方式	文本语言方式	
	goto <标签名>	
<p>(注) 1. 该指令需连接下一个执行的指令。这是为了明确程序的流程。</p> <p>2. 流程图方式中标签名的命名方法请右击指令后从菜单中选择 [设置标签]。请设定任意名称。</p>		

### ● 示例

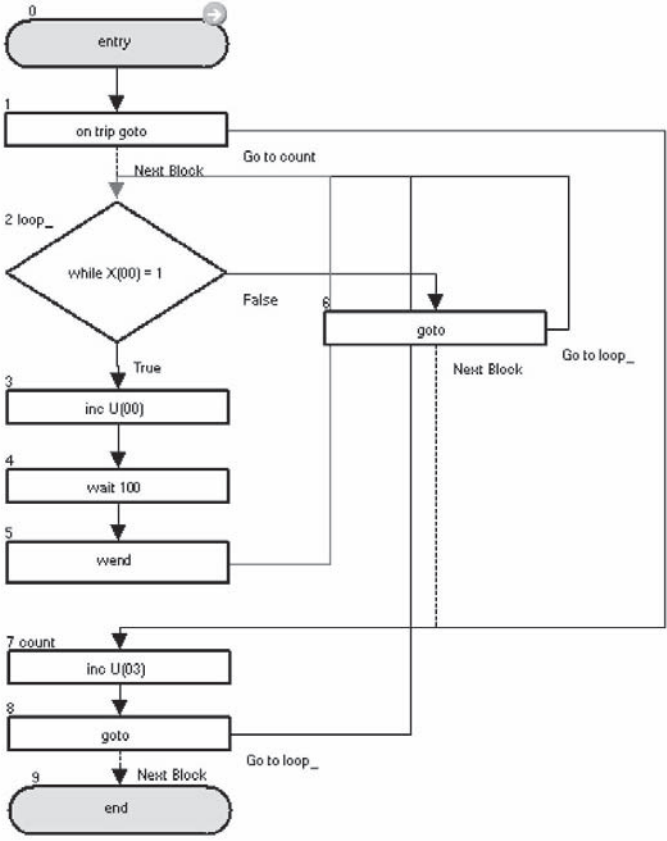
流程图	文本
 <pre> graph TD     0([0 entry]) --&gt; 1{1 Ini if U(00) = 1 goto}     1 -- True --&gt; 3[3 Condit True inc U(01)]     1 -- False --&gt; 2[2 goto]     2 --&gt; 4[4 goto]     3 --&gt; 4     4 --&gt; 5([5 end])   </pre>	<pre> entry :Ini   if U(00) = 1 goto Condit_True goto Ini :Condit_True   inc U(01) goto Ini end   </pre>

区块 No.	动作
1	U(00) 为 1 时分支至 3: Condit_True, U(00) 为 0 时跳至下一步的 2。
2	无条件分支至 1: Ini。
3	将 U(01)+1。
4	无条件分支至 1: Ini。


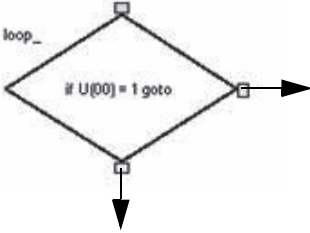
On Trip		
指令	说明	自变量
	变频器发生跳闸时，分支至<标签名>。	<b>标签名:</b> 用于识别任务内特定功能块而使用的名称
格式		
流程图方式	文本语言方式	
	on trip goto <标签名>	

- (注) 1. on trip goto 指令是指在变频器发生跳闸时进行分支的功能。  
 2. 执行 1 次 on trip goto 时，该内容将保存至变频器内部，并确认每个扫描跳闸状态。发生跳闸后在下一步处理时，将分支至 goto 位置。未发生跳闸时不会执行任何操作。  
 3. 即使执行 on trip goto 指令，该任务的 end 指令执行时 on trip goto 指令仍将解除，即使发生跳闸也不会分支。  
 4. on trip goto 指令的分支执行 1 次时，任何情况下都不会执行第 2 次分支。on trip goto 指令的分支执行 1 次后，请务必重新启动程序。  
 5. 流程图方式中标签名的命名方法请右击指令后从菜单中选择 [设置标签]。请设定任意名称。

● 示例

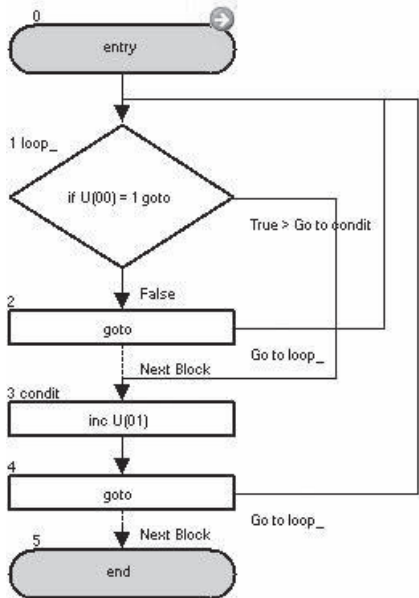
流程图	文本
	<pre> entry   on trip goto count :loop_   while X(00) = 1     inc U(00)     wait 100   wend   goto loop_ :count   inc U(03)   goto loop_ end                     </pre>

区块 No.	动作
1	执行 1 次 on trip goto 指令，设定成发生跳闸时分支至 7: count 的状态。
2 ~ 6	X(00) 为 1 时，将 U(00)+1，并使用 wait 指令待机 1.00 秒，然后返回 2: loop_。X(00) 为 0 时无条件分支至 2: loop_。
7 ~ 8	将 U(03)+1，并无条件分支至 2: loop_。





If		
指令	说明	自变量
	符合<条件>时, 分支至<标签名>。	<b>条件:</b> 根据“<左边值><比较运算符><右边值>”的格式比较 2 个变量或常数 • 左边值: 任意常数或变量 (-128 ~ 127 的范围) • 比较运算符: =, <, >, <=, >=, <> • 右边值: 任意常数或变量 (-128 ~ 127 的范围) <b>标签名:</b> 用于识别任务内特定功能块而使用的名称
格式		
流程图方式	文本语言方式	
	if <条件> goto <标签名>	

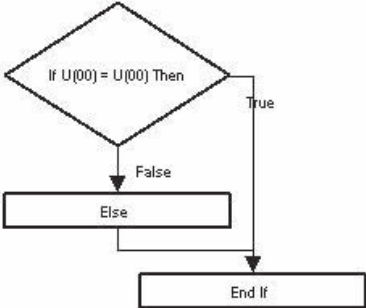
(注) 流程图方式中标签名的命名方法请右击指令后从菜单中选择 [设置标签]。请设定任意名称。

### ● 示例

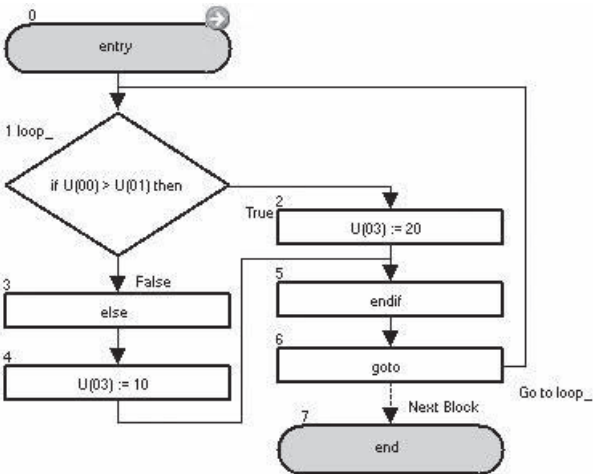
流程图	文本
	<pre> entry :loop_   if U(00) = 1 goto condit   goto loop_ :condit   inc U(01)   goto loop_ end </pre>

区块 No.	动作
1	U(00) 为 1 时分支至 3: condit, U(00) 为 1 以外时跳至下一步的 2。
2	无条件分支至 1: loop_。
3 ~ 4	将 U(01)+1, 并无条件分支至 1: loop_。



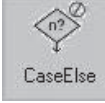

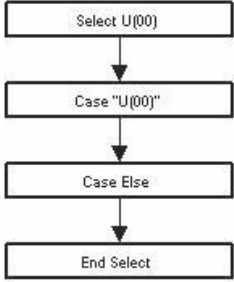
Ifs/Else/End If		
指令	说明	自变量
 Ifs  Ifs ...  Else  EndIf	符合<条件>时，将执行此指令后至 else 指令的<指令组 1>，然后跳至 endif 指令的下一步。 不符合<条件>时，将执行 else 指令后面至 endif 指令的<指令组 2>，然后跳至 endif 指令的下一步。	<b>条件:</b> 根据“<左边值><比较运算符><右边值>”的格式比较 2 个变量或常数 • 左边值: 任意常数或变量(-128 ~ 127 的范围) • 比较运算符: =, <, >, <=, >=, <> • 右边值: 任意常数或变量(-128 ~ 127 的范围) <b>指令组 1:</b> else 指令前的 1 个以上指令。可包含嵌套指令 (最多 8 层)。 <b>指令组 2:</b> endif 指令前的 1 个以上指令。可包含嵌套指令 (最多 8 层)。

格式	
流程图方式	文本语言方式
	if <条件> then <指令组 1> else <指令组 2> endif

● 示例

流程图	文本
	<pre> entry :loop_   if U(00) &gt; U(01) then     U(03) := 20   endif   goto loop_ endif end                     </pre>

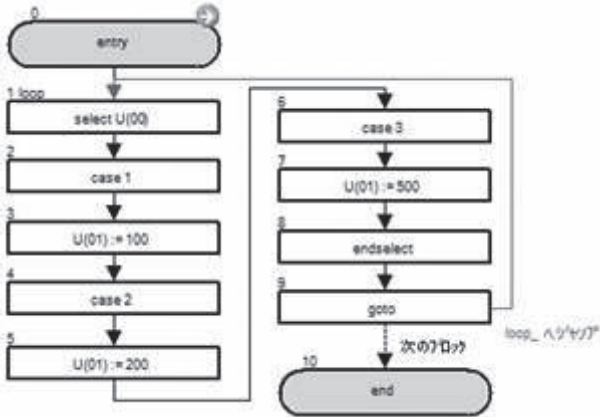
区块 No.	动作
1 ~ 5	U(00) > U(01) 时，将 20 代入后面的 U(03)，并分支至 endif 指令。U(00) > U(01) 以外时，将 10 代入 else 指令后面的 U(03)，并跳至 endif 指令。
6	无条件分支至 1: loop_。

Select/Case/CaseElse/EndSelect		
指令	说明	自变量
	<p>可根据变量值执行多个程序段。特定 case 指令的 &lt;条件变量&gt; 与 &lt;条件值 n&gt; 一致时，执行 &lt;指令组 n&gt;。</p> <p>&lt;条件变量&gt; 与 case 指令的所有条件值均不一致时，执行 &lt;与条件不一致时的指令组&gt; (caseelse)。</p> <p>该指令适用于需根据 1 个 &lt;条件变量&gt; 进行多项选择的情况。</p> <p>该指令与子程序的 call 指令组合使用，适用于整理程序。</p>	<p><b>条件变量:</b> 使用的条件变量</p> <p><b>条件值 n:</b> 任意常数或变量 (-128 ~ 127 的范围)</p> <p><b>指令组 n:</b> 下一个 case 或 endselect 指令前的 1 个以上指令。可包含嵌套指令 (最多 8 层)。</p>
		
		
		
格式		
流程图方式	文本语言方式	
	<pre>select &lt;条件变量&gt; case &lt;条件值 1 &gt; &lt;指令组 1 &gt; ... case &lt;条件值 n &gt; &lt;指令组 n &gt; ... case else &lt;与条件不一致时的指令组&gt; endselect</pre>	




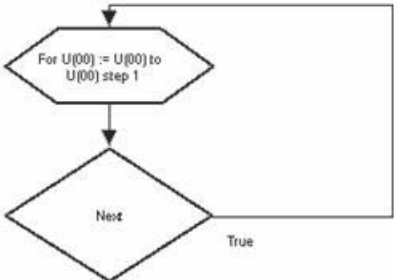
(注) select 计作 1 个嵌套。

子程序中使用 select ~ endselect 时计作 2 个嵌套。

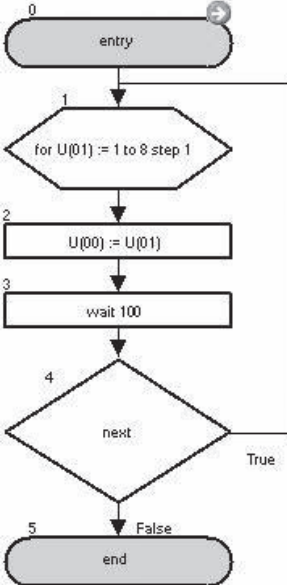
### ● 示例

流程图	文本
	<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>




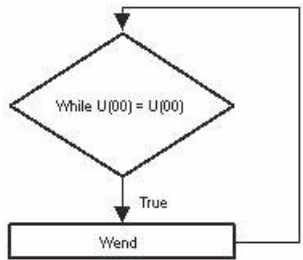
区块 No.	动作
1 ~ 8	根据 U(00) 的值，U(00) 为 1 时将 100 代入 U(01)，为 2 时将 200 代入 U(01)，为 3 时将 500 代入 U(01)，并分支至 endselect 指令。
9	无条件分支至 1: loop_。

For/Next		
指令	说明	自变量
 For ...	重复执行<指令组>，使<变量>从<开始值>起按<增量值>递增，直至达到<结束值>。执行 next 指令时、将判定<结束值>，并在<变量>上加上<增量值>。	<b>变量：</b> 任意变量 <b>开始值：</b> 初始值。在首次循环中分配至变量的值。 常数 (-128 ~ 127 的范围) <b>结束值：</b> 结束循环的值 常数 (-128 ~ 127 的范围) <b>增量值：</b> 在每个循环中，变量都会加上此值。 常数 (-128 ~ 127 的范围) <b>指令组：</b> next 指令前的 1 个以上指令。可包含嵌套指令 (最多 8 层)。
 For		
 Next		
格式		
流程图方式	文本语言方式	
	for <变量> := <开始值> to <结束值> step <增量值> <指令组> next	

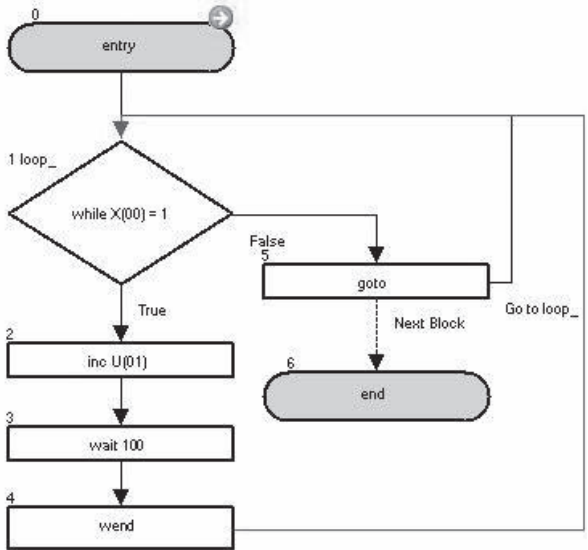
● 示例

流程图	文本
	<pre> entry   for U(01) := 1 to 8 step 1     U(00) := U(01)     wait 100   next end                     </pre>




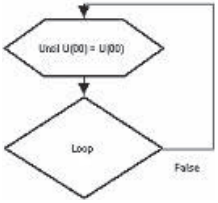
区块 No.	动作
1 ~ 4	将 1 代入 U(01) 后，将 U(01) 的值代入 U(00)，然后使用 wait 指令待机 1.00 秒，U(01) 为 8 以下时将 U(01)+1，并执行 for 指令的下一个指令。U(01) 为 8 以上时跳至 next 指令的下一步。

While/Wend		
指令	说明	自变量
 While ...   While   Wend	符合条件时，执行<指令组>。	<b>条件:</b> 根据“<左边值><比较运算符><右边值>”的格式比较 2 个变量或常数 • 左边值: 任意常数或变量 (-128 ~ 127 的范围) • 比较运算符: =, <, >, <=, >=, <> • 右边值: 任意常数或变量 (-128 ~ 127 的范围) <b>指令组:</b> wend 指令前的 1 个以上指令。可包含嵌套指令 (最多 8 层)。
格式		
流程图方式	文本语言方式	
 <pre> graph TD     Start(( )) --&gt; Cond{While U(00) = U(00)}     Cond -- True --&gt; Body[Wend]     Body --&gt; Cond           </pre>	<pre> while &lt;条件&gt; &lt;指令组&gt; wend           </pre>	

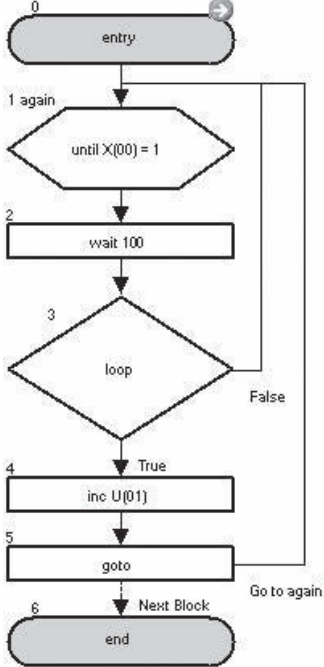
### ● 示例

流程图	文本
 <pre> graph TD     Entry([entry]) --&gt; Cond{while X(00) = 1}     Cond -- True --&gt; Inc[inc U(01)]     Inc --&gt; Wait[wait 100]     Wait --&gt; Wend[wend]     Cond -- False --&gt; Goto[goto]     Goto --&gt; End([end])     Goto -- Go to loop_ --&gt; Cond           </pre>	<pre> entry :loop_   while X(00) = 1     inc U(01)     wait 100   wend   goto loop_ end           </pre>

区块 No.	动作
1 ~ 4	X(00) 为 1 时，将 U(01)+1，并使用 wait 指令待机 1.00 秒，然后分支至 while 指令。 X(00) 为 1 以外时执行 wend 指令的下一个指令。
5	无条件分支至 1: loop_。



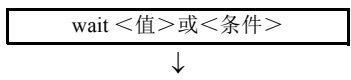
Until/Loop		
指令	说明	自变量
 Until ...	执行<指令组>直至符合<条件>。	<b>条件:</b> 根据“<左边值><比较运算符><右边值>”的格式比较 2 个变量或常数 • 左边值: 任意常数或变量 (-128 ~ 127 的范围) • 比较运算符: =, <, >, <=, >=, <> • 右边值: 任意常数或变量 (-128 ~ 127 的范围) <b>指令组:</b> loop 指令前的 1 个以上指令。可包含嵌套指令 (最多 8 层)。
 Until		
 Loop		
格式		
流程图方式	文本语言方式	
	until <条件> <指令组> loop	

● 示例

流程图	文本
	<pre> entry :again   until X(00) = 1     wait 100   loop     inc U(01)     goto again end                     </pre>

区块 No.	动作
1 ~ 3	X(00) 为 1 以外时, 使用 wait 指令待机 1.00 秒, 然后分支至 until 指令。 X(00) 为 1 时, 执行至 loop 指令后, 跳至下一个指令。
4 ~ 5	将 U(00)+1, 并无条件分支至 1: again。

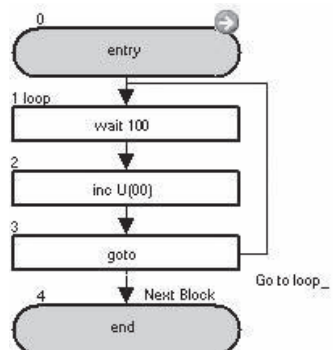


Wait		
指令	说明	自变量
	让程序待机直至达到指定秒数或符合条件。	<b>值:</b> 任意变量或常数 (以 10ms 指定的时间) 等待时间值 (0 ~ 32767×10ms) <b>条件:</b> 根据 “<左边值><比较运算符><右边值>” 的格式比较 2 个变量或常数 • 左边值: 任意变量或常数 (0 ~ 127 的范围) • 比较运算符: =, <, >, <=, >=, <> . • 右边值: 任意变量或常数 (0 ~ 127 的范围)
		
格式		
流程图方式	文本语言方式	
	wait <值>或<条件>	
<p>(注) 1. 使用 wait 指令时, 在待机至到达设定的等待时间或符合设定条件前, 不会跳至下一个指令。因此, 在 wait 指令的待机期间, 不会执行该任务处理。对于必须按固定周期进行监视的处理, 创建程序时请划分成其它任务, 确保其按固定周期进行处理, 而不使用 wait 指令。</p> <p>2. 请勿在 wait 指令使用的常数中设定负值。否则无法判断是否到达等待时间或是否符合条件, 程序将因 wait 指令而停止, 不会动作。</p> <p>3. wait 指令并非测量时间用的正确方式。需测量准确时间时, 请使用内部定时器或 LCD 操作器的时刻功能。</p>		

### 使用注意事项

- 需使 LCD 操作器显示时刻时, 需设定时刻。
- 使用时刻功能时, 需使用另售的电池 (CR2032, 3V)。

- 使用<值>设定等待时间的示例: 将根据该指令按所设<值>的时间进行待机。


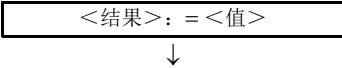
流程图	文本
 <pre> graph TD     0([entry]) --&gt; 1[wait 100]     1 --&gt; 2[inc U(00)]     2 --&gt; 3[goto]     3 -- Go to loop_ --&gt; 1     3 -- Next Block --&gt; 4([end])           </pre>	<pre> entry :loop_   wait 100   inc U(00)   goto loop_ end           </pre>

区块 No.	动作
1	使用 wait 指令待机 1.00 秒。
2 ~ 3	将 U(00)+1, 并无条件分支至 1: loop_, 再待机 1 秒。

- 使用<条件>设定待机解除条件的示例：待机至符合条件为止。

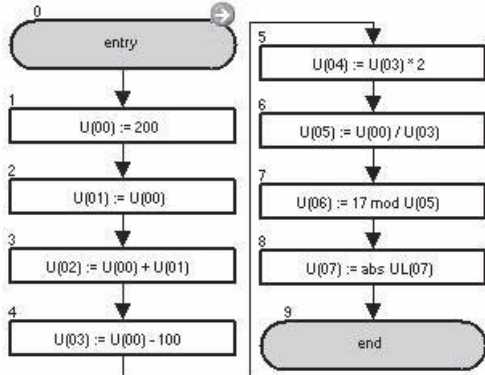
流程图		文本
<pre> graph TD     0([0 entry]) --&gt; 1[1 loop]     1 --&gt; 2[2 inc U(00)]     2 --&gt; 3[3 goto]     3 -- Go to loop_ --&gt; 1     3 -- Next Block --&gt; 4([4 end]) </pre>		<pre> entry :loop_   wait X(00) = 1   inc U(00)   goto loop_ end </pre>
区块 No.	动作	
1	待机至 X(00) 变为 1 为止。	
2 ~ 3	将 U(00)+1，并无条件分支至 1: loop_。	

## 6-5 算术运算指令和逻辑运算指令


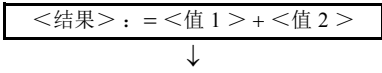
=( 代入 )		
指令	说明	自变量
	将<值>代入<结果>。	<b>结果:</b> 任意变量 <b>值:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)
格式		
流程图方式	文本语言方式	
	<结果> := <值>	

(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生上溢或下溢。

### ● 示例

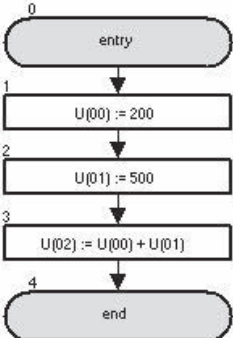
流程图	文本
	<pre> entry   U(00) := 200   U(01) := U(00)   U(02) := U(00) + U(01)   U(03) := U(00) - 100   U(04) := U(03) * 2   U(05) := U(00) / U(03)   U(06) := 17 mod U(05)   U(07) := abs UL(07) end </pre>


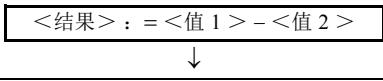
区块 No.	动作
1	将 200 代入 U(00)。
2	将 U(00) 的值代入 U(01)。
3	将 U(00) 与 U(01) 之和代入 U(02)。
4	将 U(00) 与 100 之差代入 U(03)。
5	将 U(03) 与 2 的乘积代入 U(04)。
6	将 U(00) 与 U(03) 的商代入 U(05)。
7	将 U(05) 除以 17 后的余数代入 U(06)。
8	将 UL(07) 的绝对值代入 U(07)。

+ (加)		
指令	说明	自变量
	将<值 1>与<值 2>相加。	<b>结果:</b> 任意变量 <b>值 1:</b> 任意常数或变量 (-128 ~ 127 的范围) <b>值 2:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)
格式		
流程图方式	文本语言方式	
	<结果> := <值 1> + <值 2>	

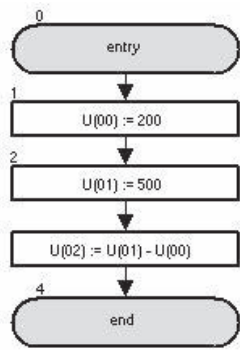
(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生上溢或下溢。

### ● 示例



流程图	文本
 <pre> graph TD     0([entry]) --&gt; 1[U(00) := 200]     1 --&gt; 2[U(01) := 500]     2 --&gt; 3[U(02) := U(00) + U(01)]     3 --&gt; 4([end]) </pre>	<pre> entry  U(00) := 200 U(01) := 500 U(02) := U(00) + U(01)  end </pre>
区块 No.	动作
1	将 200 代入 U(00)。
2	将 500 代入 U(01)。
3	将 U(00) 与 U(01) 之和代入 U(02)。

-( 减 )		
指令	说明	自变量
	将<值 1 >减去<值 2 >。	<b>结果:</b> 任意变量 <b>值 1:</b> 任意常数或变量 (-128 ~ 127 的范围) <b>值 2:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)
格式		
流程图方式	文本语言方式	
	<结果> := <值 1 > - <值 2 >	
(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生上溢或下溢。		

### ● 示例

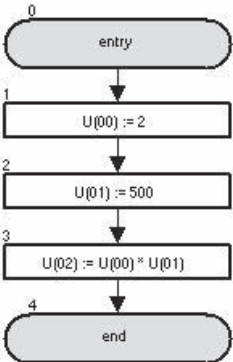
流程图	文本
 <pre> graph TD     0([0 entry]) --&gt; 1[U(00) := 200]     1 --&gt; 2[U(01) := 500]     2 --&gt; 3[U(02) := U(01) - U(00)]     3 --&gt; 4([4 end]) </pre>	<pre> entry   U(00) := 200   U(01) := 500   U(02) := U(01) - U(00) end </pre>

区块 No.	动作
1	将 200 代入 U(00)。
2	将 500 代入 U(01)。
3	将 U(01) 与 U(00) 之差代入 U(02)。


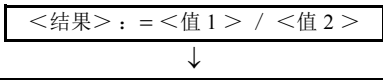
* (乘)		
指令	说明	自变量
	将<值 1 >乘以<值 2 >。	<b>结果:</b> 任意变量 <b>值 1:</b> 任意常数或变量 (-128 ~ 127 的范围) <b>值 2:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)
格式		
流程图方式	文本语言方式	
	<结果> := <值 1> * <值 2>	

(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生上溢或下溢。

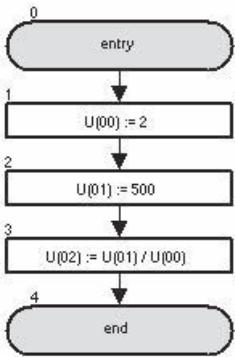
### ● 示例

流程图	文本
 <pre> graph TD     0([entry]) --&gt; 1[U(00) := 2]     1 --&gt; 2[U(01) := 500]     2 --&gt; 3[U(02) := U(00) * U(01)]     3 --&gt; 4([end]) </pre>	<pre> entry   U(00) := 2   U(01) := 500   U(02) := U(00) * U(01) end </pre>


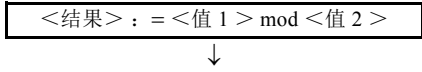
区块 No.	动作
1	将 2 代入 U(00)。
2	将 500 代入 U(01)。
3	将 U(00) 与 U(01) 的乘积代入 U(02)。

/ (除)		
指令	说明	自变量
	将<值 1 >除以<值 2 >。	<b>结果:</b> 任意变量 <b>值 1:</b> 任意常数或变量 (-128 ~ 127 的范围) <b>值 2:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)
格式		
流程图方式	文本语言方式	
	<结果> := <值 1 > / <值 2 >	
(注) DriveProgramming 在发生上溢、下溢或除零时会报错。请在应用中采取必要措施, 以免发生这种情况。		

### ● 示例

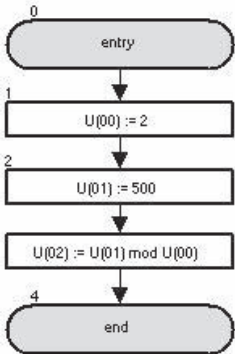
流程图	文本
 <pre> graph TD     0([entry]) --&gt; 1[U(00) := 2]     1 --&gt; 2[U(01) := 500]     2 --&gt; 3[U(02) := U(01) / U(00)]     3 --&gt; 4([end]) </pre>	<pre> entry   U(00) := 2   U(01) := 500   U(02) := U(01) / U(00) end </pre>

区块 No.	动作
1	将 2 代入 U(00)。
2	将 500 代入 U(01)。
3	将 U(01) 与 U(00) 的商代入 U(02)。

Mod( 求余数 )		
指令	说明	自变量
	<值 1 > 除以 <值 2 > 后所得的余数	<b>结果:</b> 任意变量 <b>值 1:</b> 任意常数或变量 (-128 ~ 127 的范围) <b>值 2:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)
格式		
流程图方式	文本语言方式	
	<结果> := <值 1> mod <值 2>	


(注) DriveProgramming 在发生上溢、下溢或除零时会报错。请在应用中采取必要措施, 以免发生这种情况。

### ● 示例

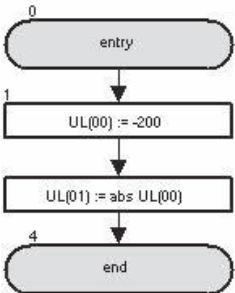
流程图	文本
	<pre> entry   U(00) := 2   U(01) := 500   U(02) := U(01) mod U(00) end </pre>


区块 No.	动作
1	将 2 代入 U(00)。
2	将 500 代入 U(01)。
3	将 U(01) 除以 U(00) 后所得的余数代入 U(02)。



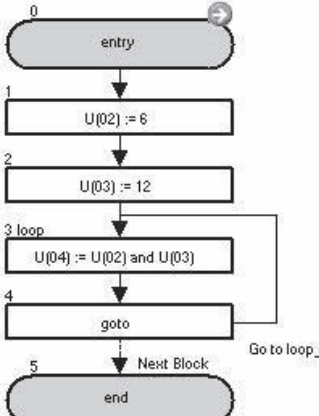
Abs(绝对值)		
指令	说明	自变量
	取<值>的绝对值	<b>结果:</b> 任意变量 <b>值:</b> 任意变量或常数 (-2,147,483,647 ~ 2,147,483,647 的范围)
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 5px; display: inline-block;">             &lt;结果&gt; := abs &lt;值&gt;           </div> ↓	<结果> := abs <值>	
(注) 1. DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生这种情况。 2. 变量或常数为最大负数 -2,147,483,648 时, 将无法转换为绝对值。仍识别 -2,147,483,648 为该负值。		

### ● 示例


流程图	文本
 <pre> graph TD     0([0 entry]) --&gt; 1[1 UL(00) := -200]     1 --&gt; 2[UL(01) := abs UL(00)]     2 --&gt; 4([4 end])           </pre>	<pre> entry   UL(00) := -200   UL(01) := abs UL(00) end           </pre>
区块 No.	动作
1	将 -200 代入 UL(00)。
2	将 UL(00) 的绝对值代入 UL(01)。

And( 逻辑与 )																	
指令	说明	自变量															
	按二进制对<值 1 >和<值 2 >执行逻辑与运算 and( 逻辑与 )	<b>结果:</b> 任意变量 <b>值 1:</b> 任意变量或常数 (-128 ~ 127 的范围) <b>值 2:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)															
	<table border="1"> <thead> <tr> <th>值 1</th> <th>值 2</th> <th>结果</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	值 1	值 2	结果	0	0	0	0	1	0	1	0	0	1	1	1	
	值 1	值 2	结果														
	0	0	0														
	0	1	0														
1	0	0															
1	1	1															
格式																	
流程图方式	文本语言方式																
<div style="border: 1px solid black; padding: 2px; display: inline-block;">           &lt;结果&gt; := &lt;值 1 &gt; and &lt;值 2 &gt;         </div> ↓	<结果> := <值 1 > and <值 2 >																
(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生这种情况。																	

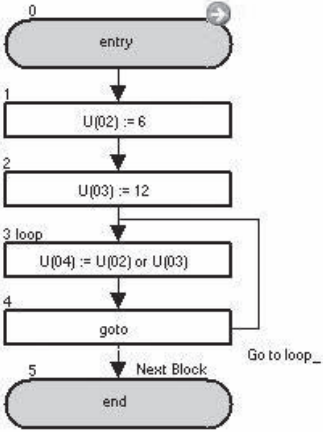
### ● 示例

流程图	文本
	<pre> entry   U(02) := 6   U(03) := 12 :loop_   U(04) := U(02) and U(03)   goto loop_ end           </pre>


区块 No.	动作
1	将 6 代入 U(02)。
2	将 12 代入 U(03)。
3	将 U(02) 的 6(二进制: 00000110) 与 U(03) 的 12(二进制: 00001100) 的逻辑与运算结果 4(二进制: 00000100) 代入 U(04)。
4	无条件分支至 3: loop_。

Or( 逻辑或 )																			
指令	说明	自变量																	
	按二进制对<值 1 >和<值 2 >执行逻辑或运算 or( 逻辑或 )	<b>结果:</b> 任意变量 <b>值 1:</b> 任意变量或常数 (-128 ~ 127 的范围) <b>值 2:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)																	
		<table border="1"> <thead> <tr> <th>值 1</th> <th>值 2</th> <th>结果</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>1</td> </tr> </tbody> </table>	值 1	值 2	结果	0	0	0	0	1	1	1	0	1	1	1	1		
	值 1	值 2	结果																
	0	0	0																
	0	1	1																
1	0	1																	
1	1	1																	
格式																			
流程图方式		文本语言方式																	
<div style="border: 1px solid black; padding: 2px; display: inline-block;">           &lt;结果&gt; := &lt;值 1 &gt; or &lt;值 2 &gt;         </div> ↓		<结果> := <值 1 > or <值 2 >																	
(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生这种情况。																			

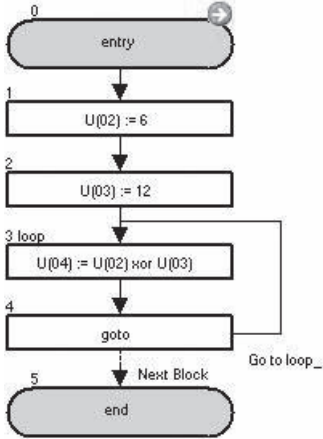
### ● 示例

流程图	文本
	<pre> entry   U(02) := 6   U(03) := 12 :loop_   U(04) := U(02) or U(03)   goto loop_ end           </pre>


区块 No.	动作
1	将 6 代入 U(02)。
2	将 12 代入 U(03)。
3	将 U(02) 的 6(二进制: 00000110)与 U(03) 的 12(二进制: 00001100)的逻辑或运算结果 14(二进制: 00001110)代入 U(04)。
4	无条件分支至 3: loop_。

XOr( 逻辑异或 )																
指令	说明	自变量														
	按二进制对<值 1 >和<值 2 >执行逻辑异或运算 xor( 逻辑异或 )	<b>结果:</b> 任意变量 <b>值 1:</b> 任意变量或常数 (-128 ~ 127 的范围) <b>值 2:</b> 任意变量或常数 (-2,147,483,648 ~ 2,147,483,647 的范围)														
	<table border="1"> <thead> <tr> <th>值 1</th> <th>值 2</th> <th>结果</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>	值 1	值 2	结果	0	0	0	0	1	1	1	0	1	1	1	0
值 1	值 2	结果														
0	0	0														
0	1	1														
1	0	1														
1	1	0														
格式																
流程图方式	文本语言方式															
<div style="border: 1px solid black; padding: 2px; display: inline-block;">           &lt;结果&gt; := &lt;值 1 &gt; xor &lt;值 2 &gt;         </div> ↓	<结果> := <值 1 > xor <值 2 >															
(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生这种情况。																

### ● 示例

流程图	文本
	<pre> entry   U(02) := 6   U(03) := 12 :loop_   U(04) := U(02) xor U(03)   goto loop_ end           </pre>

区块 No.	动作
1	将 6 代入 U(02)。
2	将 12 代入 U(03)。
3	将 U(02) 的 6(二进制: 00000110) 与 U(03) 的 12(二进制: 00001100) 的逻辑异或运算结果 10(二进制: 00001010) 代入 U(04)。
4	无条件分支至 3: loop_。

Not( 二进制的非运算 )									
指令	说明	自变量							
	按二进制对<值 1 >执行非运算 ( 位反转 ) not( 非 )	<b>结果:</b> 除位数据大小的变量以外的任意变量 *1 <b>值:</b> 除位数据大小的变量以外的任意变量或常数 *1 (-2,147,483,648 ~ 2,147,483,647 的范围 )							
	<table border="1"> <thead> <tr> <th>值 1</th> <th>结果</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>		值 1	结果	0	1	1	0	
	值 1		结果						
0	1								
1	0								
格式									
流程图方式		文本语言方式							
<div style="border: 1px solid black; padding: 5px; display: inline-block;">           &lt;结果&gt; := not &lt;值&gt;         </div> ↓		<结果> := not <值>							

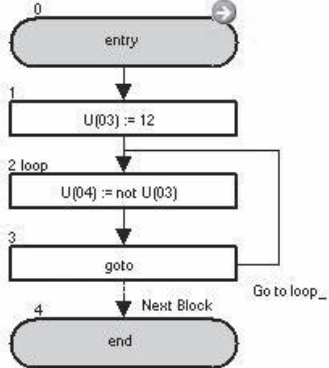
\*1. 使用 UB(1) = not UB(0) 之类的指令时, 无法获得正确结果。

反转位数据大小的变量时, 请如下例所示使用 xor 指令。


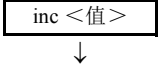
- 例 1: UB(1) = UB(0) xor 1
- 例 2: UB(2) = X(00) xor 1

(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生这种情况。

### ● 示例

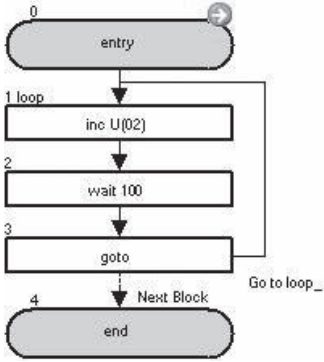
流程图	文本
	<pre> entry   U(03) := 12 :loop_   U(04) := not U(03)   goto loop_ end </pre>

区块 No.	动作
1	将 12 代入 U(03)。
2	将 U(03) 的 12( 二进制: 0000110) 的非运算 ( 位反转 ) 结果 65523( 二进制: 111110011) 代入 U(04)。
3	无条件分支至 3: loop_。

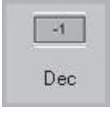
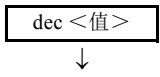
Inc(加 1)		
指令	说明	自变量
	将<值>加 1。	值: 任意变量
格式		
流程图方式	文本语言方式	
	inc <值>	

(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生这种情况。

### ● 示例

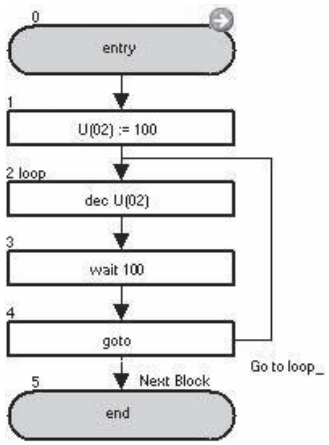
流程图	文本
	<pre>entry :loop_   inc U(02)   wait 100   goto loop_ end</pre>

区块 No.	动作
1	将 U(02) 的值 +1。
2	使用 wait 指令待机 1.00 秒。
3	无条件分支至 1: loop_。

Dec( 减 1)		
指令	说明	自变量
	将<值>减 1。	值: 任意变量
格式		
流程图方式	文本语言方式	
	dec <值>	

(注) DriveProgramming 在发生上溢或下溢时会报错。请在应用中采取必要措施, 以免发生这种情况。


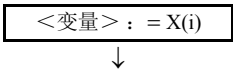
### ● 示例

流程图	文本
 <pre> graph TD     0([0 entry]) --&gt; 1[U(02) := 100]     1 --&gt; 2[2 loop]     2 --&gt; 3[dec U(02)]     3 --&gt; 4[wait 100]     4 --&gt; 5[goto]     5 -- Go to loop_ --&gt; 2     5 -- Next Block --&gt; 6([6 end]) </pre>	<pre> entry   U(02) := 100 :loop_   dec U(02)   wait 100   goto loop_ end </pre>

区块 No.	动作
1	将 100 代入 U(02)。
2	将 U(02) 的值 -1。
3	使用 wait 指令待机 1.00 秒。
4	无条件分支至 2: loop_。

## 6-6 输入输出控制指令

使用这些输入输出控制指令进行输入输出控制。也可使用 =( 代入 ) 指令，但输入输出控制指令的自变量数据较小，可更高效地使用程序容量。

var = X(i)		
指令	说明	自变量
	将输入端子变量状态的 1 位代入 <变量>。	<b>变量:</b> 任意变量 ( 变量值为 0 或 1 ) <b>i:</b> 输入端子变量的编号 (00 ~ 07 的范围)
格式		
流程图方式	文本语言方式	
	<变量> := X(i)	

(注) 输入端子变量是识别变频器本体输入端子状态的变量。需进行以下设定。输入端子变量的编号顺序取决于所设通用输入的编号顺序。

在输入端子 1 ~ 9、A、B 选择 (CA-01 ~ CA-11) 中设定通用输入 MI1 ~ MI11(86 ~ 96)。

X(00) = MI1( 功能编号: 86)

X(01) = MI2( 功能编号: 87)

X(02) = MI3( 功能编号: 88)

X(03) = MI4( 功能编号: 89)

X(04) = MI5( 功能编号: 90)

X(05) = MI6( 功能编号: 91)


X(06) = MI7( 功能编号: 92)

X(07) = MI8( 功能编号: 93)

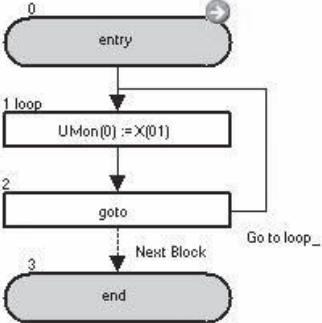
X(08) = MI9( 功能编号: 94)

X(09) = MI10( 功能编号: 95)

X(10) = MI11( 功能编号: 96)

(注) 详情请参阅  「5-2 输入输出端子变量 (P.5-5)」。


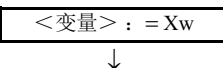
### ● 示例

流程图	文本
	<pre>entry :loop_   UMon(0) := X(01) goto loop_ end</pre>

在上述示例中，使用参数 UMon(0) [db-08] 监控输入端子 X(01) 的状态。

区块 No.	动作
1	将 X(01) 代入 UMon(0)。
2	无条件分支至 1: loop_。



var = Xw		
指令	说明	自变量
	以字为单位将输入端子变量的状态代入 <变量>。	变量: 任意变量
格式		
流程图方式	文本语言方式	
	<变量> := Xw	

(注) 输入端子变量是识别变频器本体输入端子状态的变量。需进行以下设定。输入端子变量的编号顺序取决于所设通用输入的编号顺序。

在输入端子 1 ~ 9、A、B 选择 (CA-01 ~ CA-11) 中设定通用输入 MI1 ~ MI11(86 ~ 96)。

以字为单位代入时, 高位字节及未设定的输入端子变量将读取为零。

<分配示例>

Xw = 1(位 0) ← X(00)=MI1 (功能编号: 86)

Xw = 2(位 1) ← X(01)=MI2 (功能编号: 87)

Xw = 4(位 2) ← X(02)=MI3 (功能编号: 88)

Xw = 8(位 3) ← X(03)=MI4 (功能编号: 89)

Xw = 16(位 4) ← X(04)=MI5 (功能编号: 90)

Xw = 32(位 5) ← X(05)=MI6 (功能编号: 91)

Xw = 64(位 6) ← X(06)=MI7 (功能编号: 92)

Xw = 128(位 7) ← X(07)=MI8 (功能编号: 93)

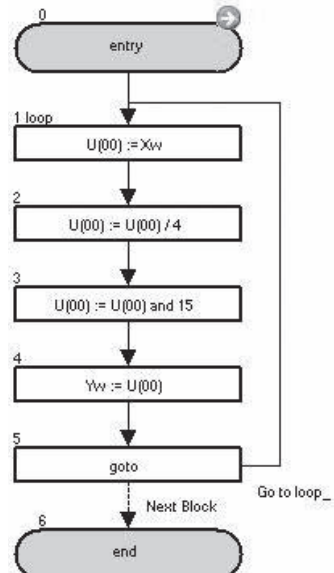
Xw = 256(位 8) ← X(08)=MI9 (功能编号: 94)

Xw = 512(位 9) ← X(09)=MI10 (功能编号: 95)

Xw = 1024(位 10) ← X(10)=MI11(功能编号: 96)


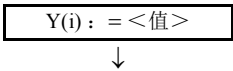
详情请参阅 □ 「5-2 输入输出端子变量 (P.5-5)」。

## ● 示例

流程图	文本
	<pre> entry :loop_   U(00) := Xw   U(00) := U(00) / 4   U(00) := U(00) and 15   Yw := U(00)   goto loop_ end </pre>

在上述示例中, 获取输入端子 X(02) ~ X(05) 的状态后, 输出至输出端子 Y(00) ~ Y(03)。

区块 No.	动作
1	将 Xw(输入端子的值) 代入 U(00)。
2	将 U(00) 的值除以 4(右移 2 位) 后, 将 X(02) 分配至位 0。
3	对 U(00) 和 15(二进制: 00001111) 执行逻辑与运算, 然后将高于 X(06) 的位设为零。
4	将 U(00) 代入 Yw。
5	无条件分支至 1: loop_。

Y(i) = 值		
指令	说明	自变量
	以位为单位输出至输出端子变量。	<b>i:</b> 输出端子变量的编号 (0 ~ 6 的范围) <b>值:</b> 任意变量或常数
格式		
流程图方式	文本语言方式	
	Y(i) := <值>	

(注) 输出端子变量是控制变频器本体输出端子的变量。需进行以下设定。输出端子变量的编号顺序取决于所设通用输出的编号顺序。

在输出端子 11 ~ 15 选择 (CC-01 ~ CC-05)、继电器输出 (16、AL) 选择 (CC-06、CC-07) 中设定通用输出 MO1 ~ MO7(69 ~ 75)。

<分配示例>

Y(00) = MO1 (通用输出 1: 69)

Y(01) = MO2 (通用输出 2: 70)

Y(02) = MO3 (通用输出 3: 71)

Y(03) = MO4 (通用输出 4: 72)

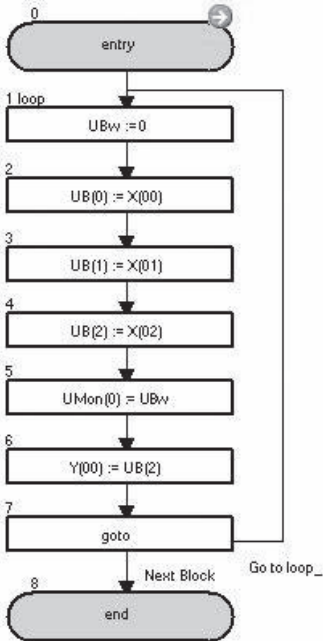
Y(04) = MO5 (通用输出 5: 73)

Y(05) = MO6 (通用输出 6: 74)

Y(06) = MO7 (通用输出 7: 75)


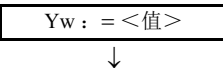
详情请参阅 □ 「5-2 输入输出端子变量 (P.5-5)」。

● 示例

流程图	文本
	<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>

在上述示例中，使用参数 UMon(0) [db-08] 监控 X(00) ~ X(02) 的状态，并将 X(02) 输出至 Y(00)。

区块 No.	动作
1	将 0 代入 UBw。
2 ~ 4	将 X(00) ~ X(02) 代入 UB(0) ~ UB(2)。
5	将 UBw 代入 UMon(0)。
6	将 UB(2) 代入 Y(00)。
7	无条件分支至 1: loop_。

Yw = 值		
指令	说明	自变量
	以字为单位输出至输出端子变量。 将各位分别反映至各输出。	值: 任意变量或常数
格式		
流程图方式	文本语言方式	
	Yw := <值>	

(注) 输出端子变量是控制变频器本体输出端子的变量。需进行以下设定。输出端子变量的编号顺序取决于所设通用输出的编号顺序。

在输出端子 11 ~ 15 选择 (CC-01 ~ CC-05)、继电器输出 (16、AL) 选择 (CC-06、CC-07) 中设定通用输出 MO1 ~ MO7(69 ~ 75)。

以字为单位代入时, 高位字节及未设定的输出端子变量将读取为零, 设定时则忽略该设定。

<分配示例>

Yw=1(位 0) → Y(00) = MO1 (通用输出 1: 69)

Yw=2(位 1) → Y(01) = MO2 (通用输出 2: 70)

Yw=4(位 2) → Y(02) = MO3 (通用输出 3: 71)

Yw=8(位 3) → Y(03) = MO4 (通用输出 4: 72)

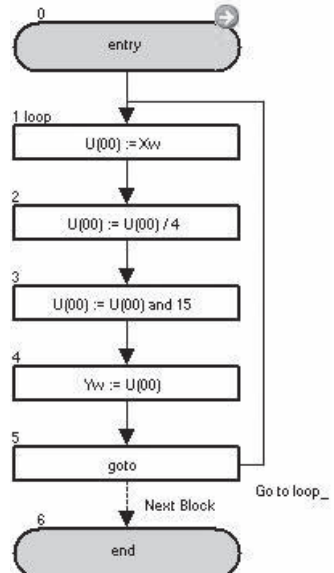
Yw=16(位 4) → Y(04) = MO5 (通用输出 5: 73)

Yw=32(位 5) → Y(05) = MO6 (通用输出 6: 74)

Yw=64(位 6) → Y(06) = MO7 (通用输出 7: 75)


详情请参阅 □ 「5-2 输入输出端子变量 (P.5-5)」。

## ● 示例

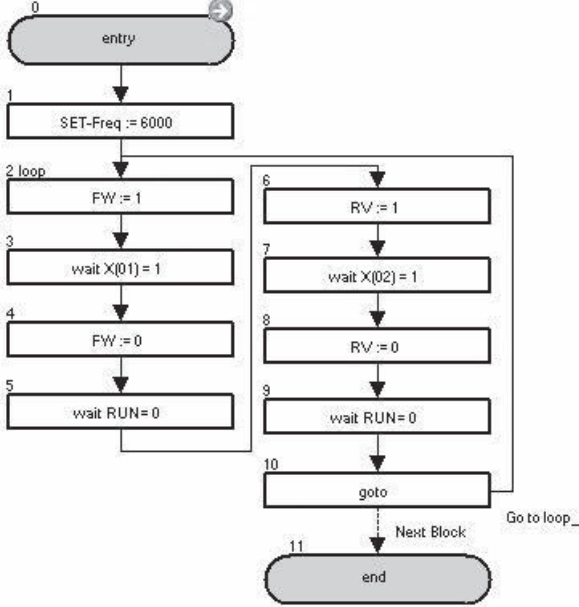
流程图	文本
	<pre> entry :loop_   U(00) := Xw   U(00) := U(00) / 4   U(00) := U(00) and 15   Yw := U(00)   goto loop_ end </pre>

在上述示例中, 获取输入端子 X(02) ~ X(05) 的状态后, 输出至输出端子 Y(00) ~ Y(03)。

区块 No.	动作
1	将 Xw(输入端子的值) 代入 U(00)。
2	将 U(00) 的值除以 4(右移 2 位) 后, 将 X(02) 分配至位 0。
3	对 U(00) 和 15(二进制: 00001111) 执行逻辑与运算, 然后将高于 X(06) 的位设为零。
4	将 U(00) 代入 Yw。
5	无条件分支至 1: loop_。



func = 值		
指令	说明	自变量
	将<值>代入输入变量。	<b>功能:</b> 输入变量的任意功能位 (各位的详情请参阅 □ 「5-6 输入变量 (P.5-17)」。)。 <b>值:</b> 任意变量或常数
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 2px; display: inline-block;">&lt;功能&gt; := &lt;值&gt;</div> ↓	<功能> := <值>	

● 示例

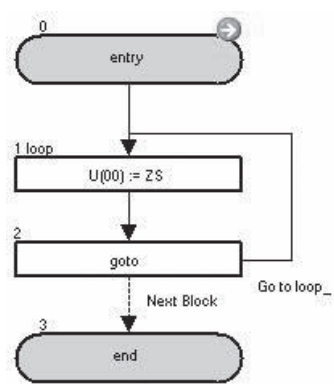
流程图	文本
	<pre> entry   SET-Freq := 6000 :loop_   FW := 1   wait X(01) = 1   FW := 0   wait RUN = 0   RV := 1   wait X(02) = 1   RV := 0   wait RUN = 0   goto loop_ end                     </pre>

在上述示例中，连续重复以 60.00Hz 正转直至 X(01) 变为 ON，然后反转直至 X(02) 变为 ON 的动作。

区块 No.	动作
1	将输出频率设为 60.00Hz。
2	使变频器正转。
3 ~ 5	待机直至 X(01) 变为 ON，然后停止变频器，使用 wait 指令进行待机直至停止。
6	使变频器反转。
7 ~ 9	待机直至 X(02) 变为 ON，然后停止变频器，使用 wait 指令进行待机直至停止。
10	无条件分支至 2: loop_。


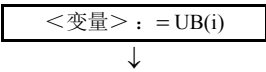
var = func		
指令	说明	自变量
	将输出变量的状态代入<变量>。	<b>变量:</b> 任意变量 <b>功能:</b> 输出变量的任意位
格式		
流程图方式		文本语言方式
<div style="border: 1px solid black; padding: 2px; display: inline-block;">&lt;变量&gt; := &lt;功能&gt;</div> ↓		<变量> := <功能>
详情请参阅  「5-7 输出变量 (P.5-20)」。		

### ● 示例

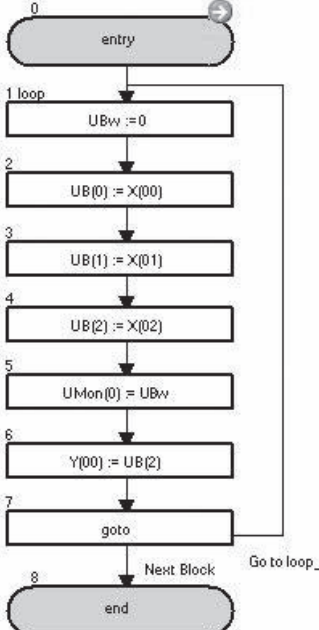
流程图	文本
 <pre> graph TD     0([0 entry]) --&gt; 1[1 loop U(00) := ZS]     1 --&gt; 2[2 goto]     2 -- Go to loop_ --&gt; 1     2 -.- Next Block -.-&gt; 3([3 end])           </pre>	<pre> entry :loop_   U(00) := ZS   goto loop_ end           </pre>

在上述示例中，ZS(0Hz 检测信号) 为 ON 时将 1 代入 U(00)，为 OFF 时将 0 代入 U(00)。

区块 No.	动作
1	将 ZS 的状态代入 U(00)。
2	无条件分支至 1: loop_。

var = UB(i)		
指令	说明	自变量
	将内部用户接点值的 1 位代入 <变量>。	<b>变量:</b> 任意变量 (变量值为 0 或 1) <b>i:</b> 内部用户接点的编号 (0 ~ 7 的范围)
格式		
流程图方式	文本语言方式	
	<变量> := UB(i)	

## ● 示例

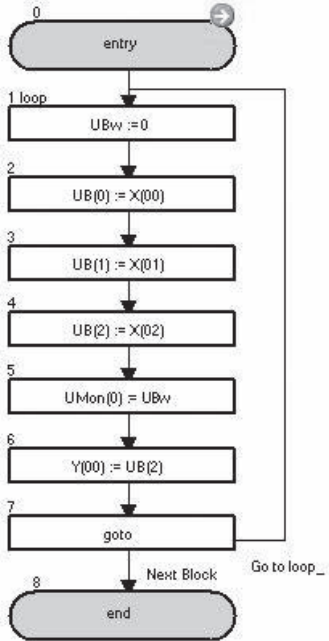
流程图	文本
	<pre> entry :loop_   ubw := 0   UB(0) := X(00)   UB(1) := X(01)   UB(2) := X(02)   UMon(0) := ubw   Y(00) := UB(2)   goto loop_ end </pre>

在上述示例中，使用参数 UMon(0) [db-08] 监控 X(00) ~ X(02) 的状态，并将 X(02) 输出至 Y(00)。

区块 No.	动作
1	将 0 代入 UBw。
2 ~ 4	将 X(00) ~ X(02) 代入 UB(0) ~ UB(2)。
5	将 UBw 代入 UMon(0)。
6	将 UB(2) 代入 Y(00)。
7	无条件分支至 1: loop_。

var = UBw		
指令	说明	自变量
	以字为单位将内部用户接点的值代入 <变量>。	<b>变量:</b> 任意变量
格式		
流程图方式		文本语言方式
<div style="border: 1px solid black; padding: 5px; display: inline-block;">&lt;变量&gt; := UBw</div> ↓		<变量> := UBw
UBw = 1(位 0) ← UB(0) UBw = 2(位 1) ← UB(1) UBw = 4(位 2) ← UB(2) UBw = 8(位 3) ← UB(3) UBw = 16(位 4) ← UB(4) UBw = 32(位 5) ← UB(5) UBw = 64(位 6) ← UB(6) UBw = 128(位 7) ← UB(7) UBw = 256(位 8) ← UB(8) UBw = 512(位 9) ← UB(9) UBw = 1024(位 10) ← UB(10) UBw = 2048(位 11) ← UB(11) UBw = 4096(位 12) ← UB(12) UBw = 8192(位 13) ← UB(13) UBw = 16384(位 14) ← UB(14) UBw = 32768(位 15) ← UB(15)		


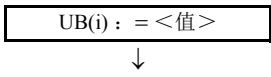
### ● 示例

流程图	文本
	<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>

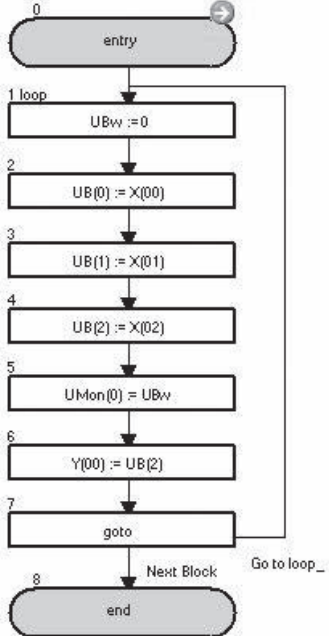
在上述示例中，使用参数 UMon(0) [db-08] 监控 X(00) ~ X(02) 的状态，并将 X(02) 输出至 Y(00)。

区块 No.	动作
1	将 0 代入 UBw。
2 ~ 4	将 X(00) ~ X(02) 代入 UB(0) ~ UB(2)。
5	将 UBw 代入 UMon(0)。
6	将 UB(2) 代入 Y(00)。
7	无条件分支至 1: loop_。




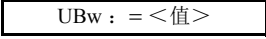
Ub(i) = 值		
指令	说明	自变量
 UBi=value	将内部用户接点代入 <值>。	<b>i:</b> 内部用户接点的编号 (0 ~ 7 的范围) <b>值:</b> 任意变量或常数
格式		
流程图方式	文本语言方式	
 ↓	UB(i) := <值>	

### ● 示例

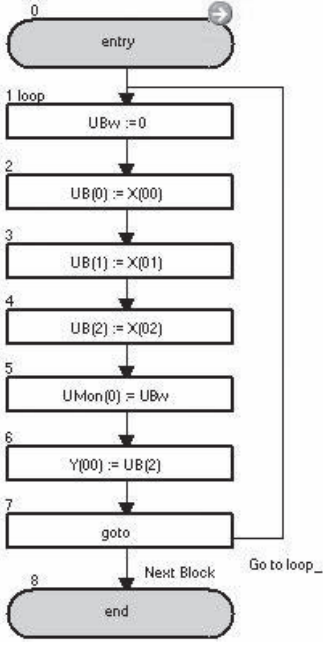
流程图	文本
	<pre> entry :loop_   ubw := 0   UB(0) := X(00)   UB(1) := X(01)   UB(2) := X(02)   UMon(0) := ubw   Y(00) := UB(2)   goto loop_ end </pre>

在上述示例中，使用参数 UMon(0) [db-08] 监控 X(00) ~ X(02) 的状态，并将 X(02) 输出至 Y(00)。

区块 No.	动作
1	将 0 代入 UBw。
2 ~ 4	将 X(00) ~ X(02) 代入 UB(0) ~ UB(2)。
5	将 UBw 代入 UMon(0)。
6	将 UB(2) 代入 Y(00)。
7	无条件分支至 1: loop_。

Ubw = 值		
指令	说明	自变量
	以字为单位将<值>代入内部用户接口。	值: 任意变量或常数
格式		
流程图方式	文本语言方式	
 ↓	UBw := <值>	
UBw = 1(位 0) → UB(0) UBw = 2(位 1) → UB(1) UBw = 4(位 2) → UB(2) UBw = 8(位 3) → UB(3) UBw = 16(位 4) → UB(4) UBw = 32(位 5) → UB(5) UBw = 64(位 6) → UB(6) UBw = 128(位 7) → UB(7) UBw = 256(位 8) → UB(8) UBw = 512(位 9) → UB(9) UBw = 1024(位 10) → UB(10) UBw = 2048(位 11) → UB(11) UBw = 4096(位 12) → UB(12) UBw = 8192(位 13) → UB(13) UBw = 16384(位 14) → UB(14) UBw = 32768(位 15) → UB(15)		


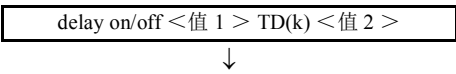
### ● 示例

流程图	文本
	<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>

在上述示例中，使用参数 UMon(0) [db-08] 监控 X(00) ~ X(02) 的状态，并将 X(02) 输出至 Y(00)。

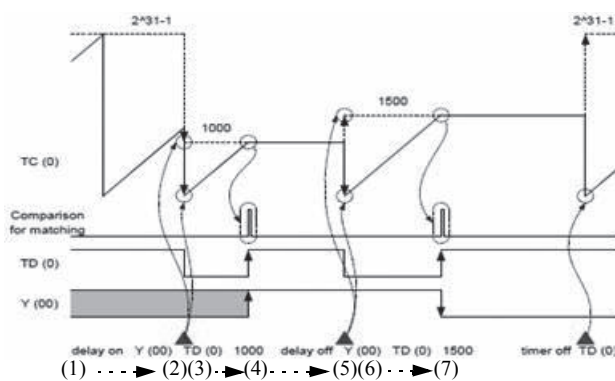
区块 No.	动作
1	将 0 代入 UBw。
2 ~ 4	将 X(00) ~ X(02) 代入 UB(0) ~ UB(2)。
5	将 UBw 代入 UMon(0)。
6	将 UB(2) 代入 Y(00)。
7	无条件分支至 1: loop_。

## 6-7 定时器控制指令

Delay on/off		
指令	说明	自变量
	启动定时器 (k)。定时器计数器 TC(k) 从 0 开始计数，每 10ms 递增直至达到 <值 2>，达到 <值 2> 时定时器接点 TD(K) 切换为 ON，执行 <值 1> 中指定的处理 (on 或 off)。	<b>on/off:</b> 延迟时间后的动作设定 (on/off) <b>值 1:</b> 任意接点变量或变量 <b>值 2:</b> 任意变量或常数 (以 ×10ms 指定的时间) <b>TD(k):</b> : 所用定时器的定时器输出接点 (k 为 0 ~ 7 的范围)
格式		
流程图方式	文本语言方式	
	delay on/off <值 1> TD(k) <值 2>	

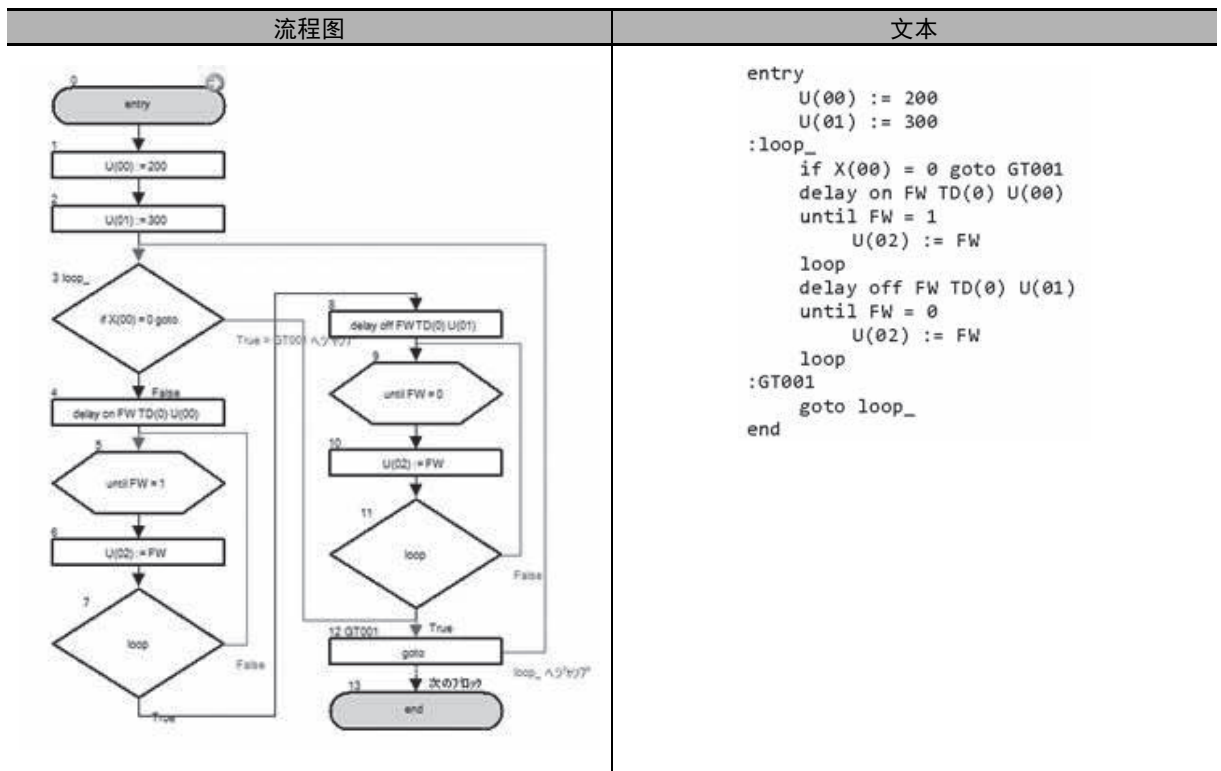
- (注) 1. 执行 delay on/off 指令时，将启动定时器 (k)。TD(k) 为 ON 的状态下定时器 (k) 启动时，TD(k) 将变为 OFF，但 <值 1> 变量保持原值不变。  
定时器 (k) 启动后，程序处理将进入下一个指令。
2. 执行 delay on/off 指令时，执行时的 <值 1>、TD(k)、<值 2> 的数据将保存至内部。执行后即使变更 <值 1>、<值 2> 中设定的变量值，delay on/off 指令动作仍不变。
3. 执行 delay on/off 指令后，在该处理完成前再次启动同一定时器 (k) 时，之前执行的处理将被取消，将按照再次设定的执行内容启动定时器 (k)。因此，创建程序时，请确保启动定时器 (k) 后在处理完成前不再次执行。
4. 启动的定时器 (k) 的定时器数据可通过定时器计数器变量 TC(k) 进行监控。定时器处理完成与否通过定时器输出接点 TD(k) 进行确认 (完成时 ON)。

### ● 时序图




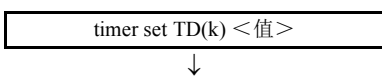
- (1) 定时器处于自由运行状态
- (2) delay on 指令启动
- (3) 延迟动作中
- (4) 时间到
- (5) delay off 指令启动
- (6) 延迟动作中
- (7) 时间到

## ● 示例

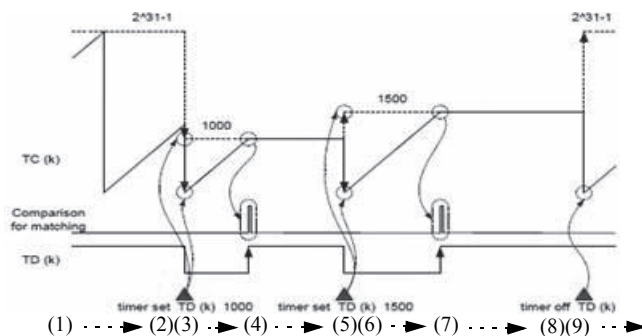


在上述示例中，非 X(00)=0 时，使用 delay on 指令开始正转运行，使用 delay off 指令停止运行。

区块 No.	动作
1	将 200 代入 U(00)。
2	将 300 代入 U(01)。
3	X(00) 为 0 时，分支至 8: GT001。X(00) 不为 0 时，进入下一步。
4	使用 delay on 指令按设定值 U(00) 启动定时器 (0)。
5 ~ 7	在 FW 变为 1 之前，将 FW 的状态代入 U(02)。FW 变为 1 时，进入下一步。
8	使用 delay off 指令按设定值 U(01) 启动定时器 (0)。
9 ~ 11	在 FW 变为 0 之前，将 FW 的状态代入 U(02)。FW 变为 0 时，进入下一步。
12	无条件分支至 3: loop_。

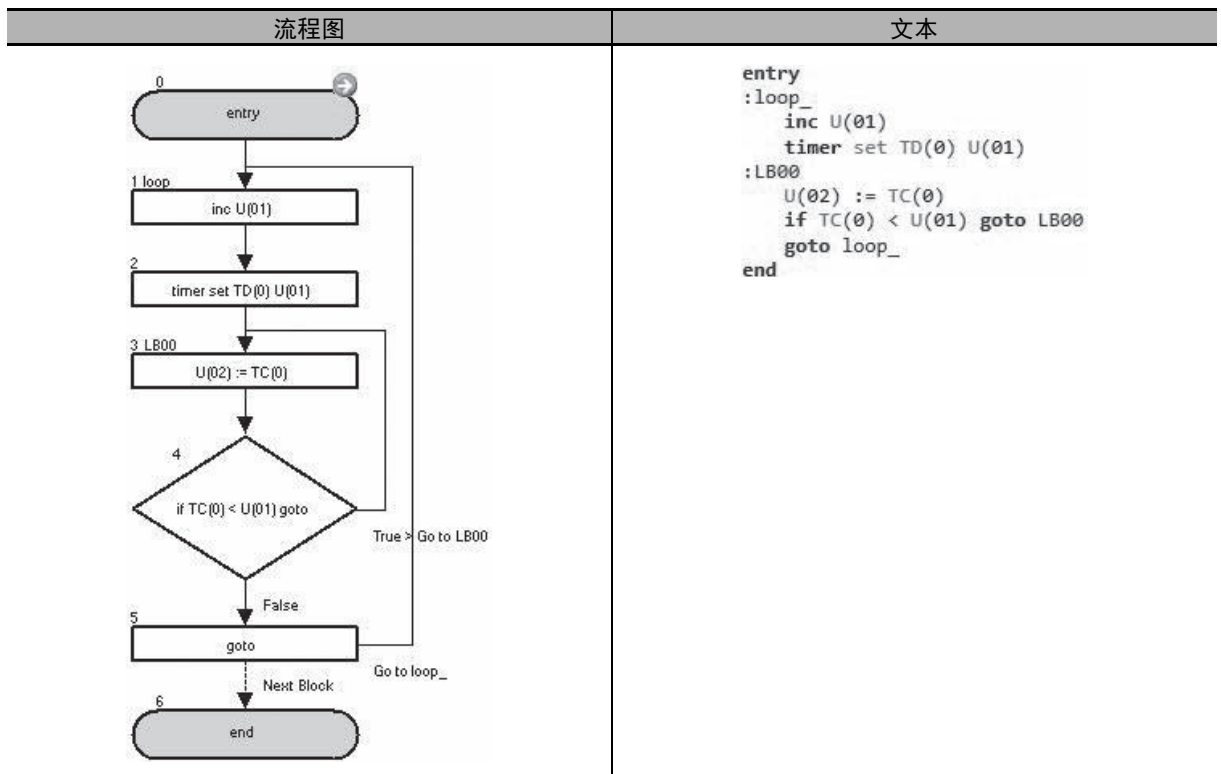
Timer Set		
指令	说明	自变量
	启动定时器 (k)。定时器计数器 TC(k) 从 0 开始计数，每 10ms 递增直至达到<值>，达到<值>时将定时器接点 TD(k) 设为 ON。	<b>TD(k):</b> : 所用定时器的定时器输出接点 (k 为 0 ~ 7 的范围) <b>值:</b> 任意变量或常数 (以 ×10ms 指定的时间)
格式		
流程图方式	文本语言方式	
	timer set TD(k) <值>	
<p>(注) 1. 执行 timer set 指令时，将启动定时器 (k)。TD(k) 为 ON 的状态下定时器 (k) 启动时，TD(k) 将变为 OFF。定时器 (k) 启动后，程序处理将进入下一个指令。</p> <p>2. 执行 timer set 指令时，执行时的 TD(k)、&lt;值&gt;的数据将保存至内部。执行后即使变更&lt;值&gt;中设定的变量值，timer set 指令动作仍不变。</p> <p>3. 执行 timer set 指令后，在该处理完成前再次启动同一定时器 (k) 时，之前执行的处理将被取消，将按照再次设定的执行内容启动定时器 (k)。因此，创建程序时，请确保启动定时器 (k) 后在处理完成前不再次执行。</p> <p>4. 启动的定时器 (k) 的定时器数据可通过定时器计数器变量 TC(k) 进行监控。定时器处理完成与否通过定时器输出接点 TD(k) 进行确认 (完成时 ON)。</p>		

### ● 时序图




- (1) 定时器处于自由运行状态
- (2) timer set 指令启动
- (3) 延迟动作中
- (4) 时间到
- (5) timer set 指令启动
- (6) 延迟动作中
- (7) 时间到
- (8) timer off 指令启动
- (9) 定时器处于自由运行状态

## ● 示例

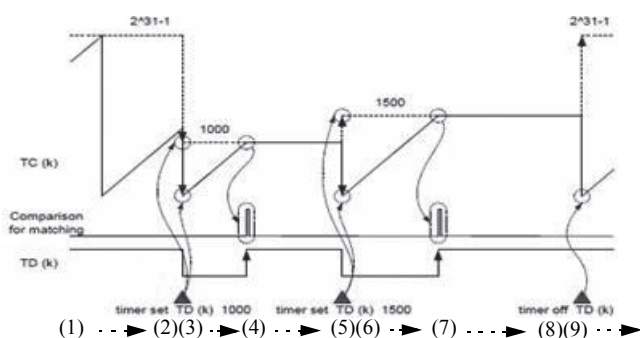


在上述示例中，定时器每次执行完成时 TD(0) 的设定值均会加 1，每次执行各循环所需的时间均会延长。定时器的当前值反映在 U(02) 中。

区块 No.	动作
1	将 U(01)+1。
2	使用 timer set 指令按设定值 U(01) 启动定时器 (0)。
3	将定时器 (0) 的当前值 TC(0) 代入 U(02)。
4	定时器 (0) 的当前值 TC(0) 小于设定值 U(01) 时，分支至 3: LB00。 其它情况下，进入下一步。
5	无条件分支至 1: loop_。

Timer Off		
指令	说明	自变量
	将定时器计数器 TC(k) 复位为 0，在自由运行模式下启动。	<b>TD(k):</b> : 所用定时器的定时器输出接点 (k 为 0 ~ 7 的范围)
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 5px; display: inline-block;">timer off TD(k)</div> ↓	timer off TD(k)	

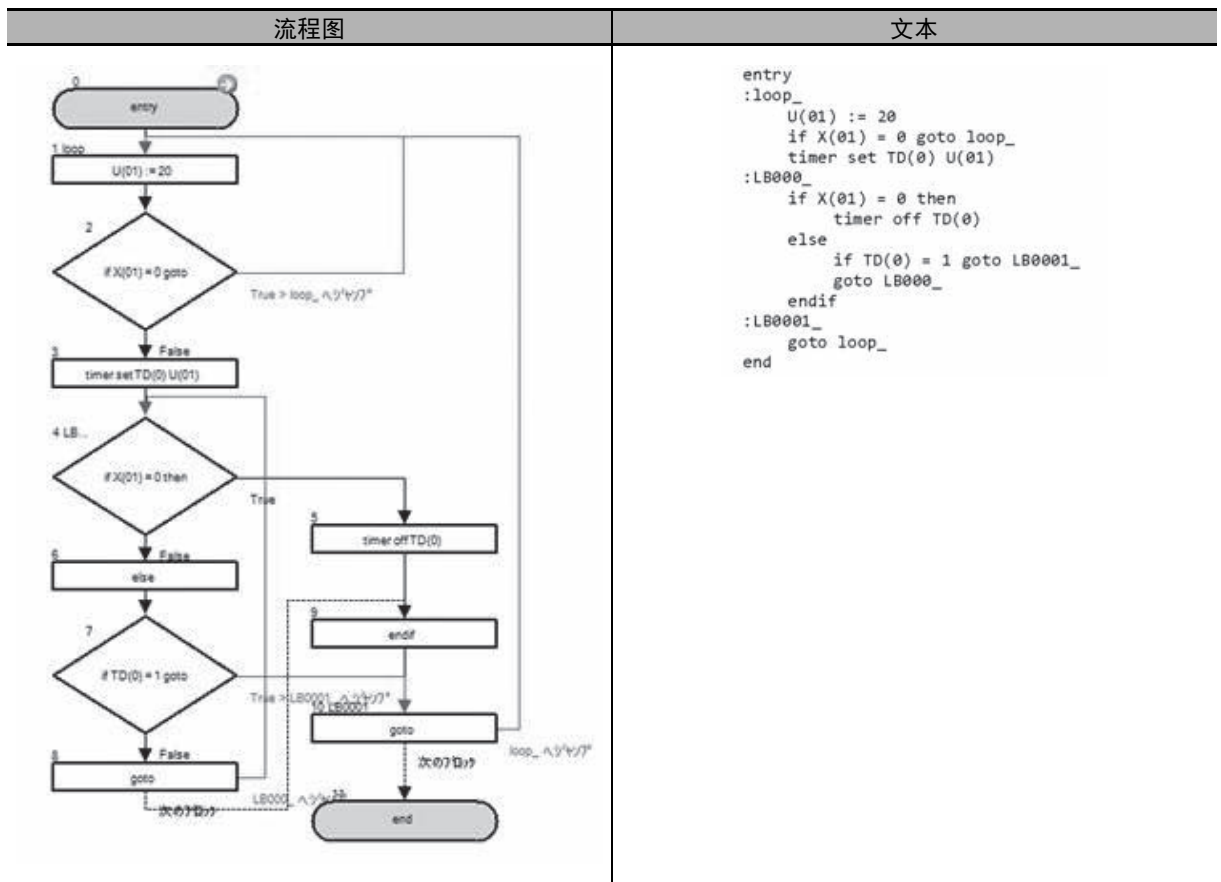
● 时序图



- (1) 定时器处于自由运行状态
- (2) timer set 指令启动
- (3) 延迟动作中
- (4) 时间到
- (5) timer set 指令启动
- (6) 延迟动作中
- (7) 时间到
- (8) timer off 指令启动
- (9) 定时器处于自由运行状态




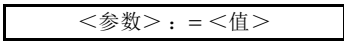
## ● 示例



在上述示例中，X(01)不为0时，将重复定时器(0)的启动/完成。X(01)=0时，停止定时器并监视X(01)。

区块 No.	动作
1	将 20 代入 U(01)。
2	X(01) 为 0 时，分支至 1: loop_。X(01) 不为 0 时，进入下一步。
3	使用 timer set 指令按设定值 U(01) 启动定时器 (0)。
4 ~ 9	监视 X(01)，X(01) 不为 0 且 TD(0)=1 时分支至 10: LB0001_。 X(01) 为 0 时，停止定时器 (0) 并进入下一步。
10	无条件分支至 1: loop_。

## 6-8 参数控制指令

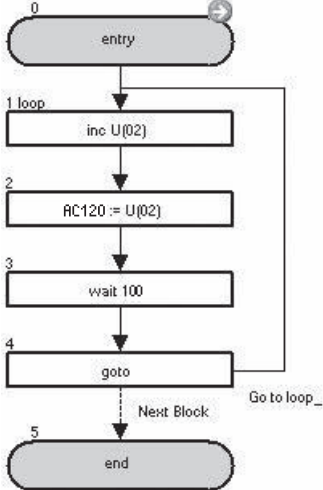
ChgParam		
指令	说明	自变量
	将<参数>指定的变频器的参数数据变更为<值>。 可变更任意变频器参数。	<b>参数:</b> 参数代码 (Fxxx、Axxx、bXXX、Cxxx、Hxxx、Pxxx、 oxxx、Uxxx) <b>值:</b> 任意变量或常数
格式		
流程图方式	文本语言方式	
	<参数> := <值>	

(注) 1. 与使用变频器本体的 LCD 操作器进行设定一样, 各参数存在以下限制事项。发生限制事项中所述的状态时, 变频器将检出执行指令错误 (E045), DriveProgramming 功能的程序将停止。关于参数的详细限制事项, 请参阅 □「第 7 章 DriveProgramming 使用参数的注意事项」以及 □《高性能型通用变频器 RX2 系列用户手册 (SBCE-CN5-437)》。

- 设定的数据超出了设定参数的设定范围。
- 设定的参数与相关参数之间发生了匹配异常。
- 设定了所选模式下无法使用的功能。
- 在运行期间变更了运行期间不可变更的参数。
- 在软件锁选择 (UA-16) 禁止变更参数数据的状态下进行了设定。


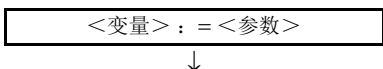
2. 即使使用 ChgParam 指令变更了参数的设定数据, 该参数在 EEPROM 中的数据也不会变更。

### ● 示例

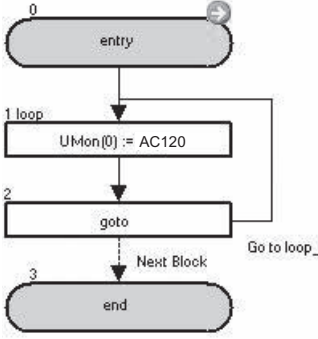
流程图	文本
 <pre> graph TD     0([entry]) --&gt; 1[inc U(02)]     1 --&gt; 2[AC120 := U(02)]     2 --&gt; 3[wait 100]     3 --&gt; 4[goto]     4 -- Go to loop_ --&gt; 1     4 -.- Next Block -.-&gt; 5([end])   </pre>	<pre> entry :loop_   inc U(02)   AC120 := U(02)   wait 100   goto loop_ end   </pre>

在上述示例中, AC120(第 1 加速时间 1) 的值每秒加 1。

区块 No.	动作
1	将 U(02)+1。
2	将 U(02) 代入参数 AC120。
3	使用 wait 指令待机 1.00 秒。
4	无条件分支至 1: loop_。


MonParam		
指令	说明	自变量
	将<参数>指定的变频器参数的内容代入<变量>。	<b>参数:</b> 参数代码 (Fxxx、Axxx、bxxx、Cxxx、dxxx、Hxxx、Pxxx、oxxx、Uxxx) <b>变量:</b> 任意变量
格式		
流程图方式	文本语言方式	
	<变量> := <参数>	

### ● 示例

流程图	文本
 <pre> graph TD     0([0 entry]) --&gt; 1[1 loop UMon(0) := AC120]     1 --&gt; 2[2 goto]     2 -- Go to loop_ --&gt; 1     2 -- Next Block --&gt; 3([3 end]) </pre>	<pre> entry :loop_   UMon(0) := AC120   goto loop_ end </pre>

在上述示例中，将参数 AC120(第 1 加速时间 1) 的值代入参数 UMon(0) [db-08] (用户参数监控) 进行监控。

区块 No.	动作
1	将参数 FA-01 代入 UMon(0)。
2	无条件分支至 1: loop_。

RtcSet		
指令	说明	自变量
	在变量中设定 LCD 数字操作器发来的 6 字节时刻数据。该数据对应年、月、日、星期、时、分。 十六进制的变量值对应年、月、日、星期、时、分(十进制)。 <b>rtcset on:</b> 连续更新 6 字节数据。 <b>rtcset off:</b> 仅更新 1 次 6 字节数据。	<b>on/off:</b> 连续 / 1 次 <b>用户变量:</b> 任意用户参数变量或内部用户变量 (U(xx) 或 UL(xx))
格式		
流程图方式		文本语言方式
<div style="border: 1px solid black; padding: 5px; display: inline-block;">rtcset on/off &lt;用户变量&gt;</div> ↓		rtcset on/off <用户变量>

- (注) 1. 该功能使用 LCD 操作器配备的时刻功能。在未连接 LCD 操作器的情况下执行 rtcset on/off 指令时, 该处理不会完成, 整个程序会使用该指令进入待机状态。因此, 建议在创建程序时, 仅在程序启动后启动 1 次 rtcset on 指令, 确保始终对指定变量输出时刻数据。
2. 使用时刻功能时, 需在 LCD 操作器中放入另售的电池 (CR2032, 3V)。需使用 rtcset on/off 指令时, 请正确设定时刻。此外, 时刻功能存在时刻误差, 因此请定期调整时刻。
3. 执行 rtcset on/off 指令时, 将以字节为单位输出 BCD 数据。  
 星期数据从星期日至星期六依次用 00 至 06 表示。但用户参数变量 U(k) 及内部用户变量无法识别 BCD 数据。创建程序时, 请注意 DriveProgramming 编辑器及 LCD 操作器会将 BCD 数据识别为十六进制数据, 并将其转换为十进制数据进行显示。
4. 执行 rtcset on/off U(k) 时, 按照以下顺序从高位字节起开始设定。  
 请确保含有 U(k) 的连续 3 个用户参数变量。

用户参数变量	高位字节	低位字节
U(k)	年 BCD 数据	月 BCD 数据
U(k+1)	日 BCD 数据	星期 BCD 数据
U(k+2)	时 BCD 数据	分 BCD 数据

5. 执行 rtcset on/off UL(k) 时, 按照以下顺序从高位字节起开始设定。  
 请确保含有 UL(k) 的连续 2 个内部用户变量。

用户参数变量	高位字		低位字	
	高位字节	低位字节	高位字节	低位字节
UL(k)	年 BCD 数据	月 BCD 数据	日 BCD 数据	星期 BCD 数据
UL(k+1)	时 BCD 数据	分 BCD 数据	0(填充)	0(填充)

6. 请在切断变频器本体电源的状态下装拆 LCD 操作器。否则会导致故障。  
 与 LCD 操作器的连接发生断线异常时, 在断线后的至少 1 分 45 秒内会保持断线时的时刻。因此, 在断线后的至少 1 分 45 秒内仍可执行 rtcset on/off 指令。断线时间超出 1 分 45 秒时, 全零数据将输出至指定变量。
7. LCD 操作器的电池电量不足时, 将无法保持时刻功能的数据。变频器接通电源期间, 该数据由控制电路电源保持, 但再次接通电源时, 时刻功能会恢复为初始数据 (2000 年 1 月 1 日 SUN00: 00)。



### 安全要点

DriveProgramming 使用时刻功能进行控制时，可能会因 LCD 操作器的电池消耗而导致意外动作。

电池消耗请使用 [E042]RTC 错误进行检测，并采取停止变频器及程序等措施。

拆下 LCD 操作器或断线时， DriveProgramming 根据时刻功能会进入待机状态。



### 使用注意事项

- 需使 LCD 操作器显示时刻时，需设定时刻。
- 使用时刻功能时，需使用另售的电池 (CR2032, 3V)。

### ● 示例



流程图	文本
<pre> graph TD     0([0 entry]) --&gt; 1[1 rtcset off U(00)]     1 --&gt; 2([2 end])           </pre>	<pre> entry   rtcset off U(00) end           </pre>

在 2012 年 10 月 18 日 (星期四) 下午 2 时 29 分执行了该程序示例时， DriveProgramming 编辑器及 LCD 操作器中 U(00)、U(01) 及 U(02) 显示如下。

用户参数变量	时刻功能 BCD 数据 (实际十六进制数据)	显示 (将十六进制 BCD 数据转 换为十进制数据)	时刻功能 BCD 数据的含义
U(00)	1210	4624	“12”表示 2012 年，“10”表示 10 月
U(01)	1804	6148	“18”表示 18 日，“04”表示星期四
U(02)	1429	5161	“14”表示下午 2 时，“29”表示 29 分

## 6-9 变频器控制指令

Run FW		
指令	说明	自变量
	变频器开始正转运行。 该指令为事先设定成 FW := 1 的 func= 值指令的快捷键。	---
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 2px; display: inline-block;">FW := 1</div> ↓	FW := 1	
Run RV		
指令	说明	自变量
	变频器开始反转运行。 该指令为事先设定成 RV := 1 的 func= 值指令的快捷键。	---
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 2px; display: inline-block;">RV := 1</div> ↓	RV := 1	
Stop FW		
指令	说明	自变量
	使变频器从正转运行变为减速停止。 该指令为事先设定成 FW := 0 的 func= 值指令的快捷键。	---
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 2px; display: inline-block;">FW := 0</div> ↓	FW := 0	
Stop RV		
指令	说明	自变量
	使变频器从反转运行变为减速停止。 该指令为事先设定成 RV := 0 的 func= 值指令的快捷键。	---
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 2px; display: inline-block;">RV := 0</div> ↓	RV := 0	

停止 (func = 值)			
指令	说明	自变量	
 func=value	<p>使用 func = 值的指令，使正在运行的变频器减速停止。</p> <p>请参阅  「6-6 入出力制御命令 func = 值 (P.6-40)」。</p> <p>将 &lt;值&gt; 代入所选 &lt;功能&gt; 变量。</p>	<p><b>功能：</b> 请选择以下任意变量。</p> <p>FW：Run FW 时正转运行</p> <p>RV：Run RV 时反转运行</p> <p><b>值：</b> 请设定“0”(停止)以减速停止。</p>	
格式			
流程图方式		文本语言方式	
(Forward operation) <div style="border: 1px solid black; padding: 2px; display: inline-block;">FW := 0</div> 或 <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">RV := 0</div>	(Forward operation) FW := 0	(Reverse Operation) RV := 0	
↓	↓		




### 使用注意事项

- DriveProgramming 功能的程序停止时，FW(正转)变量及 RV(反转)变量会清零，而不会保持状态。
- FW(正转)变量及 RV(反转)变量仅在变频器本体的运行指令选择(AA111)设定为“01: [FW][RV] 端子”时有效。使用其它设定时不会动作。
- 变频器接通电源后，即使在 FW(正转)变量及 RV(反转)变量中设定“1”，该设定仍会被忽略，不会执行正转运行及反转运行。请设定为“0”后再设定为“1”。为了避免该动作，创建程序时请仅在电源接通后使用 wait 指令等留出 1 秒的等待时间。

Stop		
指令	说明	自变量
	使运行中的变频器减速停止。 此外，检出变频器跳闸时起复位作用。	---
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 2px; display: inline-block;">stop</div> ↓	stop	

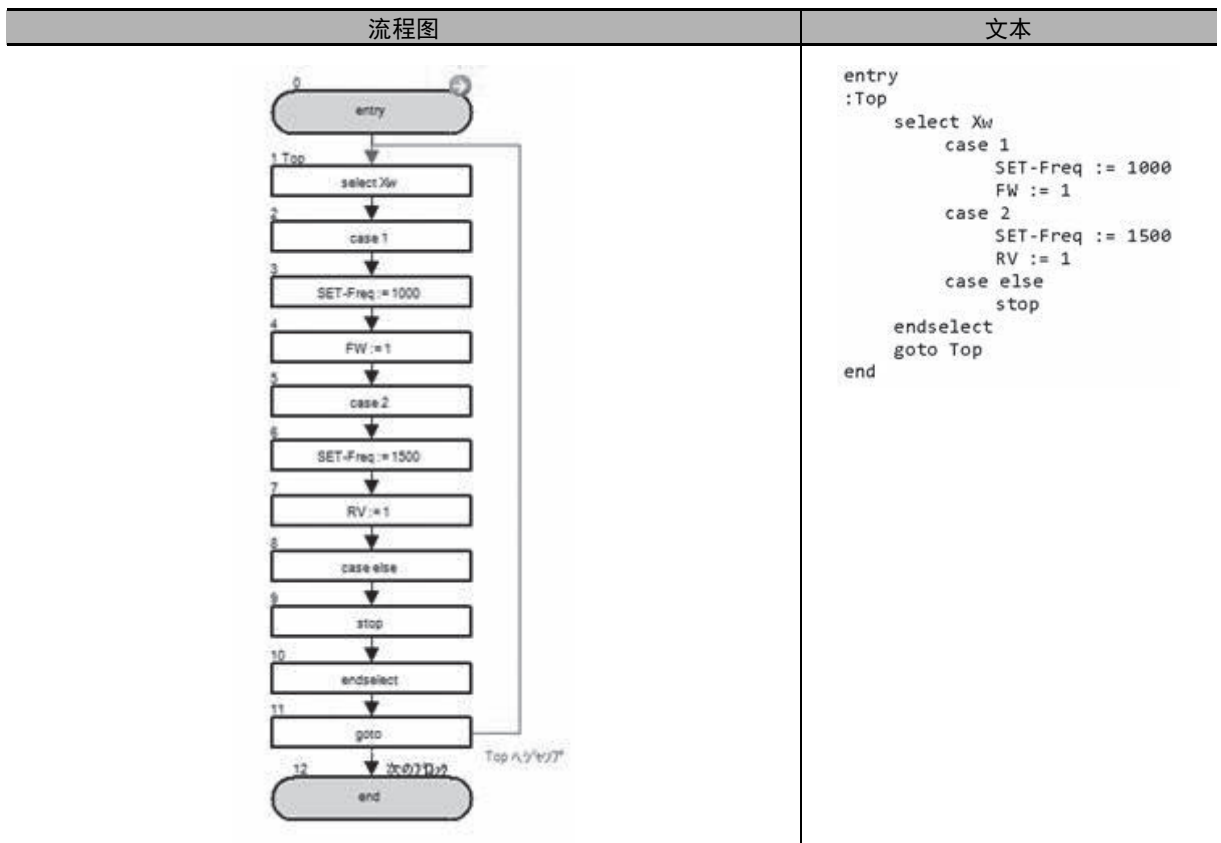
- (注) 1. stop 指令的功能与 LCD 操作器的 STOP/RESET 键相同。  
使运行中的变频器减速停止。此外，检出变频器跳闸时起复位作用。
2. 为防止 stop 指令起复位作用，请在变频器本体的 STOP 键选择 (AA-13) 中设定“00: 无效”。在该设定下，即便使变频器减速停止，stop 指令也不会起复位作用。但 LCD 操作器的 STOP/RESET 键也将无效。
3. 使用 stop 指令时会起复位作用，或者 LCD 操作器的 STOP/RESET 键会失效，因此建议使用上述的“停止 (func= 值)”进行停止。

Set Freq		
指令	说明	自变量
	设定变频器的频率。 该指令会在 =( 代入 ) 指令的左边指定频率指令变量 SET-Freq。 单位: 0.01Hz	值: 任意变量或常数 (0 ~ 40,000 的范围)
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 2px; display: inline-block;">SET-Freq : = &lt;值&gt;</div> ↓	SET-Freq : = <值>	

- (注) 1. 变频器实际可输出的频率为第 1 最低频率 (Hb130) ~ 最高频率。设定了超出范围的数据时，动作如下。
- 设定小于第 1 最低频率 (Hb130)  
以 0.00Hz 动作。  
控制方式选择 (AA121) 设定为“09: 0Hz 区无传感器矢量控制”、“10: 带传感器矢量控制”时，第 1 最低频率 (Hb130) 无效，输出指定频率。
  - 设定超出最高频率  
通过第 1 / 第 2 最高频率 (Hb105/Hd105/Hb205/Hd205) 中设定的频率限制频率指令。
2. DriveProgramming 功能的程序停止时，频率指令变量将保持程序停止前的状态。程序再次启动时，将从保持的数据起开始。




## ● 示例

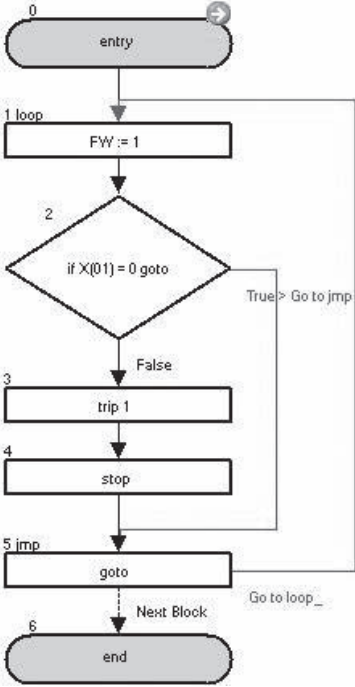


在上述示例中，通用输入接点 Xw 为 1 时，电机会以 10Hz 沿正转方向运转。通用输入接点 Xw 为 2 时，则会以 15Hz 沿反方向运转。为其它值时，电机会停止。

区块 No.	动作
1	在 select 指令后，根据 Xw 的值分支至以下 case 指令。 Xw=1 时分支至 case 1，Xw=2 时分支至 case 2，其它情况下分支至 case else。
2 ~ 4	Xw=1 时执行。将频率指令变量设为 10.00Hz，使变频器正转后，执行 endselect 指令的下一步。
5 ~ 7	Xw=2 时执行。将频率指令变量设为 15.00Hz，使变频器反转后，执行 endselect 指令的下一步。
8 ~ 10	Xw 为 1、2 以外时执行。停止变频器，并执行 endselect 指令的下一步。
11	无条件分支至 1: Top。



Trip		
指令	说明	自变量
	使变频器跳闸。 变频器检出跳闸时将异常停止。	<b>值:</b> 任意变量或常数 (0 ~ 9 的范围) 0 ~ 9 对应 E050 ~ E059 的报警代码。
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 5px; display: inline-block;">trip &lt;值&gt;</div> ↓	trip <值>	
<p>(注) 1. 即使在 1 个任务中使变频器跳闸, DriveProgramming 功能的其它任务仍将继续执行, 不会停止。</p> <p>2. 为任意变量或常数 (最大 127), 也可设定 10 以上的值, 但设定 10 以上时即使跳闸, 所有报警代码仍将显示 E059。</p>		

### ● 示例

流程图	文本
	<pre> entry :loop_   FW := 1   if X(01) = 0 goto jmp   trip 1   stop :jmp   goto loop_ end </pre>

在上述示例中, 输入端子变量 X(01) 设为 ON 时, 变频器会发生用户跳闸。

区块 No.	动作
1	使变频器正转。
2	X(01) 为 0 时, 分支至 5: jmp。其它情况下, 进入下一步。
3	变频器发生用户跳闸 1。
4	停止变频器。
5	无条件分支至 1: loop_。

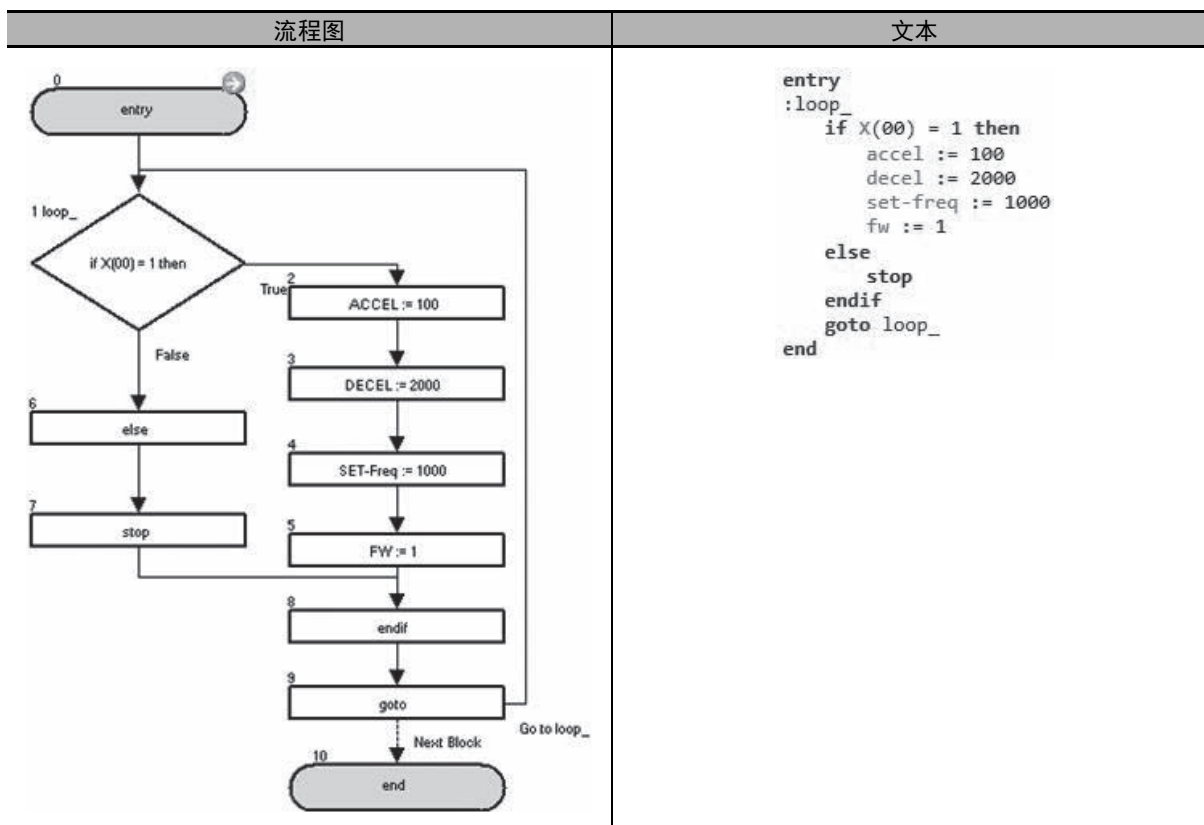
Accel		
指令	说明	自变量
	设定变频器的加速时间。 该指令会在 =( 代入 ) 指令的左边指定加速时间变量 ACCEL。 单位: 10ms	值: 任意变量或常数
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 5px; display: inline-block;">ACCEL := &lt;值&gt;</div> ↓	ACCEL := <值>	
Decel		
指令	说明	自变量
	设定变频器的减速时间。 该指令会在 =( 代入 ) 指令的左边指定减速时间变量 DECEL。 单位: 10ms	值: 任意变量或常数
格式		
流程图方式	文本语言方式	
<div style="border: 1px solid black; padding: 5px; display: inline-block;">DECEL := &lt;值&gt;</div> ↓	DECEL := <值>	



#### 使用注意事项

- 仅在加减速时间输入种类 (AC-01) 中设定“04: Drive Programming”时, 加速时间变量 ACCEL 和减速时间变量 DECEL 才会生效。
- DriveProgramming 功能的程序停止时, 加速时间变量和减速时间变量将保持程序停止前的状态。程序再次启动时, 将从保持的数据起开始。

## ● 示例



在上述示例中，输入端子变量 X(00) 设为 ON 时，将加速时间设为 1.00 秒，将减速时间设为 20.00 秒。

区块 No.	动作
1 ~ 6	X(00) 为 1 时，将加速时间设为 1.00 秒，将减速时间设为 20.00 秒，将输出频率设为 10.00Hz，并使变频器正转。 X(00) 为 1 以外时，跳至 else 指令的下一步。
7 ~ 8	停止变频器。
9	无条件分支至 1: loop_。

# 7

## DriveProgramming 使用参数的注意事项

本章对 DriveProgramming 使用参数的注意事项进行介绍。

---

7-1 受设定顺序影响的变频器参数一览表 .....	7-2
7-2 受额定电流 (%) 影响的参数一览表 .....	7-5
7-3 受 PID 有效 / 无效影响的参数一览表 .....	7-6

## 7-1 受设定顺序影响的变频器参数一览表

参数的设定范围可能会受到其它参数的设定数据的限制。  
代表示例如下所述。

参数 No.	说明	数据
Hb104	第 1 IM 基底频率	10.00 ~ 最高频率 [Hb105](Hz)
Hb105	第 1 IM 最高频率	基底频率 [Hb104] ~ 590.00 (Hz)
Hd104	第 1 SM(PMM) 基底频率	10.00 ~ 最高频率 [Hd105](Hz)
Hd105	第 1 SM(PMM) 最高频率	基底频 [Hd104] ~ 590.00 (Hz)
Hb204	第 2 IM 基底频率	10.00 ~ 最高频率 [Hb205](Hz)
Hb205	第 2 IM 最高频率	基底频率 [Hb204] ~ 590.00 (Hz)
Hd204	第 2 SM(PMM) 基底频率	10.00 ~ 最高频率 [Hd205](Hz)
Hd205	第 2 SM(PMM) 最高频率	基底频率 [Hd204] ~ 590.00 (Hz)
AA104	第 1 辅助速度设定	0.00/ 最低频率 [Hb130][Hd130] ~ 最高频率 [Hb105][Hd105](Hz)
AA204	第 2 辅助速度设定	0.00/ 最低频率 [Hb230][Hd230] ~ 最高频率 [Hb205][Hd205](Hz)
Ab110	第 1 多段速 0 速	0.00/ 最低频率 [Hb130][Hd130] ~ 最高频率 [Hb105][Hd105](Hz)
Ab210	第 2 多段速 0 速	0.00/ 最低频率 [Hb230][Hd230] ~ 最高频率 [Hb205][Hd205](Hz)
Ab-11 ~ Ab-25	多段速 1 速 ~ 15 速	0.00/ 最低频率 [Hb130][Hd130] ~ 最高频率 [Hb105][Hd105](Hz)
bA-60	制动电阻动作电路 (BRD) 使用率	0.0 ~ 10.0×( 制动电阻动作电路 (BRD) 电阻值 [bA-63]/ 最小电阻值 ) <sup>2</sup> (%)
bA102	第 1 频率上限限位	第 1 频率上限限位 [bA102] ≤ 第 1 IM 最高频率 [Hb105] 第 1 频率上限限位 [bA102] ≤ 第 1 SM(PMM) 最高频率 [Hd105]
bA103	第 1 频率下限限位	第 1 频率下限限位 [bA103] ≤ 第 1 IM 最高频率 [Hb105] 第 1 频率下限限位 [bA103] ≤ 第 1 SM(PMM) 最高频率 [Hd105]
bA202	第 2 频率上限限位	第 2 频率上限限位 [bA202] ≤ 第 2 IM 最高频率 [Hb205] 第 2 频率上限限位 [bA202] ≤ 第 2 SM(PMM) 最高频率 [Hd205]
bA203	第 2 频率下限限位	第 2 频率下限限位 [bA203] ≤ 第 2 IM 最高频率 [Hb205] 第 2 频率下限限位 [bA203] ≤ 第 2 SM(PMM) 最高频率 [Hd205]
bC120	第 1 自由电子热敏保护频率 1	0.00 ~ 第 1 自由电子热敏保护频率 2 [bC122](Hz)
bC122	第 1 自由电子热敏保护频率 2	第 1 自由电子热敏保护频率 1[bC120] ~ 第 1 自由电子热敏保护频率 3[bC124](Hz)
bC124	第 1 自由电子热敏保护频率 3	第 1 自由电子热敏保护频率 2[bC122] ~ 590.00(Hz)
bC220	第 2 自由电子热敏保护频率 1	0.00 ~ 第 2 自由电子热敏保护频率 2 [bC222](Hz)
bC222	第 2 自由电子热敏保护频率 2	第 2 自由电子热敏保护频率 1[bC220] ~ 第 2 自由电子热敏保护频率 3[bC224](Hz)

参数 No.	说明	数据
bC224	第 2 自由电子热敏保护频率 3	第 2 自由电子热敏保护频率 2[bC222] ~ 590.00(Hz)
bA-31	瞬停不间断功能开始电压	200Vclass: 0 ~ 410.0(V) 400Vclass: 0 ~ 820.0(V) 瞬停不间断功能开始电压 [bA-31] ≤ 瞬停不间断目标电平 [bA-32]
bA-32	瞬停不间断目标电平	200Vclass: 0 ~ 410.0(V) 400Vclass: 0 ~ 820.0(V) 瞬停不间断功能开始电压 [bA-31] ≤ 瞬停不间断目标电平 [bA-32]
Cb-05	[Ai1] 端子 起始比例	0.0 ~ [Ai1] 端子 终止比例 [Cb-06](%)
Cb-06	[Ai1] 端子 终止比例	[Ai1] 端子 起始比例 [Cb-05] ~ 100.0(%)
Cb-15	[Ai2] 端子 起始比例	0.0 ~ [Ai2] 端子 终止比例 [Cb-16](%)
Cb-16	[Ai2] 端子 终止比例	[Ai2] 端子 起始比例 [Cb-15] ~ 100.0(%)
Cb-25	[Ai3] 端子起始比例	-100.0 ~ [Ai3] 端子终止比例 [Cb-26](%)
Cb-26	[Ai3] 端子终止比例	[Ai3] 端子起始比例 [Cb-25] ~ 100.0(%)
CE-40	窗口比较器 [Ai1] 上限电平	设定上限电平。 设定范围: 0 ~ 100 设定下限值: 窗口比较器 [Ai1] 下限电平 [CE-41]+ 窗口比较器 [Ai1] 滞后宽度 [CE-42] ×2
CE-41	窗口比较器 [Ai1] 下限电平	设定下限电平。 设定范围: 0 ~ 100 设定上限值: 窗口比较器 [Ai1] 上限电平 [CE-40]- 窗口比较器 [Ai1] 滞后宽度 [CE-42]×2
CE-42	窗口比较器 [Ai1] 滞后宽度	针对上下限电平, 设定滞后宽度。 设定范围: 0 ~ 10 设定上限值: (窗口比较器 [Ai1] 上限电平 [CE-40]- 窗口比较器 [Ai1] 下限电平 [CE-41])/2
CE-43	窗口比较器 [Ai2] 上限电平	设定上限电平。 设定范围: 0 ~ 100 设定下限值: 窗口比较器 [Ai2] 下限电平 [CE-44]+ 窗口比较器 [Ai2] 滞后宽度 [CE-45] ×2
CE-44	窗口比较器 [Ai2] 下限电平	设定下限电平。 设定范围: 0 ~ 100 设定上限值: 窗口比较器 [Ai2] 上限电平 [CE-43]- 窗口比较器 [Ai2] 滞后宽度 [CE-45]×2
CE-45	窗口比较器 [Ai2] 滞后宽度	针对上下限电平, 设定滞后宽度。 设定范围: 0 ~ 10 设定上限值: (窗口比较器 [Ai2] 上限电平 [CE-43]- 窗口比较器 [Ai2] 下限电平 [CE-44])/2
CE-46	窗口比较器 [Ai3] 上限电平	设定上限电平。 设定范围: 0 ~ 100 设定下限值: 窗口比较器 [Ai3] 下限电平 [CE-48]+ 窗口比较器 [Ai3] 滞后宽度 [CE-47] ×2
CE-47	窗口比较器 [Ai3] 下限电平	设定下限电平。 设定范围: 0 ~ 100 设定上限值: 窗口比较器 [Ai3] 上限电平 [CE-46]- 窗口比较器 [Ai3] 滞后宽度 [CE-48]×2
CE-48	窗口比较器 [Ai3] 滞后宽度	针对上下限电平, 设定滞后宽度。 设定范围: 0 ~ 10 设定上限值: (窗口比较器 [Ai3] 上限电平 [CE-46]- 窗口比较器 [Ai3] 下限电平 [CE-47])/2
Hb150	第 1 自由 V/f 频率 1	0.00 ~ 第 1 自由 V/f 频率 2[Hb152](Hz)

参数 No.	说明	数据
Hb152	第 1 自由 V/f 频率 2	第 1 自由 V/f 频率 1[Hb150] ~ 第 1 自由 V/f 频率 3[Hb154](Hz)
Hb154	第 1 自由 V/f 频率 3	第 1 自由 V/f 频率 2[Hb152] ~ 第 1 自由 V/f 频率 4[Hb156](Hz)
Hb156	第 1 自由 V/f 频率 4	第 1 自由 V/f 频率 3[Hb154] ~ 第 1 自由 V/f 频率 5[Hb158](Hz)
Hb158	第 1 自由 V/f 频率 5	第 1 自由 V/f 频率 4[Hb156] ~ 第 1 自由 V/f 频率 6[Hb160](Hz)
Hb160	第 1 自由 V/f 频率 6	第 1 自由 V/f 频率 5[Hb158] ~ 第 1 自由 V/f 频率 7[Hb162](Hz)
Hb162	第 1 自由 V/f 频率 7	第 1 自由 V/f 频率 6[Hb160] ~ 第 1 IM 基底频率 [Hb104](Hz)
Hb250	第 2 自由 V/f 频率 1	0.00 ~ 第 2 自由 V/f 频率 2[Hb252](Hz) 0
Hb252	第 2 自由 V/f 频率 2	第 2 自由 V/f 频率 1[Hb250] ~ 第 2 自由 V/f 频率 3[Hb254](Hz)
Hb254	第 2 自由 V/f 频率 3	第 2 自由 V/f 频率 2[Hb252] ~ 第 2 自由 V/f 频率 4[Hb256](Hz)
Hb256	第 2 自由 V/f 频率 4	第 2 自由 V/f 频率 3[Hb254] ~ 第 2 自由 V/f 频率 5[Hb258](Hz)
Hb258	第 2 自由 V/f 频率 5	第 2 自由 V/f 频率 4[Hb256] ~ 第 2 自由 V/f 频率 6[Hb260](Hz)
Hb260	第 2 自由 V/f 频率 6	第 2 自由 V/f 频率 5[Hb258] ~ 第 2 自由 V/f 频率 7[Hb262](Hz)
Hb262	第 2 自由 V/f 频率 7	第 2 自由 V/f 频率 6[Hb260] ~ 第 2 IM 基底频率 [Hb204](Hz)



## 7-2 受额定电流 (%) 影响的参数一览表

对于设定电流的参数，其初始数据及设定范围会受到变频器额定电流的影响。

通过 DriveProgramming 功能的程序进行设定时，请以占变频器额定电流的比率 (0.01% 为单位) 进行设定。

数据中含额定电流的参数代表示例如下所述。

参数 No.	说明	数据
AF136	第 1 制动释放电流 (正转)	0.00 ~ 200.00(%) (变频器额定电流比)
AF143	第 1 制动释放电流 (反转)	0.00 ~ 200.00(%) (变频器额定电流比)
bA121	第 1 过电流抑制电平	0.00 ~ 200.00(%) (变频器额定电流比)
bA123	第 1 过载限制 1 电平	20.00 ~ 200.00(%) (变频器额定电流比)
bA127	第 1 过载限制 2 电平	20.00 ~ 200.00(%) (变频器额定电流比)
bb-43	频率捕获重启电平	0.00 ~ 200.00(%) (变频器额定电流比)
bb-46	频率牵引重启时的过电流限制电平	0.00 ~ 200.00(%) (变频器额定电流比)
bb160	第 1 过电流检测电平	20.00 ~ 220.00(%) (ND 额定电流比)
bC110	第 1 电子热敏保护电平	0.00 ~ 300.00(%) (变频器额定电流比)
bC121	第 1 自由电子热敏保护电流 1	0.00 ~ 300.00(%) (变频器额定电流比)
bC123	第 1 自由电子热敏保护电流 2	0.00 ~ 300.00(%) (变频器额定电流比)
bC125	第 1 自由电子热敏保护电流 3	0.00 ~ 300.00(%) (变频器额定电流比)
CE102	第 1 低电流检测电平 1	0.00 ~ 200.00(%) (变频器额定电流比)
CE103	第 1 低电流检测电平 2	0.00 ~ 200.00(%) (变频器额定电流比)
CE106	第 1 过载预警电平 1	0.00 ~ 200.00(%) (变频器额定电流比)
CE107	第 1 过载预警电平 2	0.00 ~ 200.00(%) (变频器额定电流比)
PA-23	输出电流监控任意设定值	0.00 ~ 300.00(%) (变频器额定电流比)
AF236	第 2 制动释放电流 (正转)	0.00 ~ 200.00(%) (变频器额定电流比)
AF243	第 2 制动释放电流 (反转)	0.00 ~ 200.00(%) (变频器额定电流比)
bA221	第 2 过电流抑制电平	0.00 ~ 200.00(%) (变频器额定电流比)
bA223	第 2 过载限制 1 电平	0.00 ~ 200.00(%) (变频器额定电流比)
bA227	第 2 过载限制 2 电平	20.00 ~ 200.00(%) (变频器额定电流比)
bb260	第 2 过电流检测电平	20.00 ~ 220.00(%) (ND 额定电流比)
bC210	第 2 电子热敏保护电平	0.00 ~ 300.00(%) (变频器额定电流比)
bC221	第 2 自由电子热敏保护电流 1	0.00 ~ 300.00(%) (变频器额定电流比)
bC223	第 2 自由电子热敏保护电流 2	0.00 ~ 300.00(%) (变频器额定电流比)
bC225	第 2 自由电子热敏保护电流 3	0.00 ~ 300.00(%) (变频器额定电流比)
CE202	第 2 低电流检测电平 1	0.00 ~ 200.00(%) (变频器额定电流比)
CE203	第 2 低电流检测电平 2	0.00 ~ 200.00(%) (变频器额定电流比)
CE206	第 2 过载预警电平 1	0.00 ~ 200.00(%) (变频器额定电流比)
CE207	第 2 过载预警电平 2	0.00 ~ 200.00(%) (变频器额定电流比)

## 7-3 受 PID 有效 / 无效影响的参数一览

设定值会根据 AH-04、05 的设定进行比例缩放。

参数 No.	说明	数据
db-30	PID1 反馈数据 1 监控	A) 转换值 (标尺转换单位) = ([AH-05] - [AH-04]) / 100.00 * [输入 (%)] + [AH-04]
db-32	PID1 反馈数据 2 监控	
db-34	PID1 反馈数据 3 监控	B) 内部换算值 (%) = ([输入 (※)] - [AH-04]) / ([AH-05] - [AH-04]) * 100.00 ※ 相当于 mon param 读取值 (标尺与 AH-04、05 相同)
db-42	PID1 目标值监控	
db-44	PID1 反馈数据监控	
FA-30	PID1 目标值 1	
FA-32	PID1 目标值 2	
FA-34	PID1 目标值 3	
AH-10	PID1 目标值 1 设定值	
AH-12	PID1 多段目标值 1	
AH-14	PID1 多段目标值 2	
AH-16	PID1 多段目标值 3	
AH-18	PID1 多段目标值 4	
AH-20	PID1 多段目标值 5	
AH-22	PID1 多段目标值 6	
AH-24	PID1 多段目标值 7	
AH-26	PID1 多段目标值 8	
AH-28	PID1 多段目标值 9	
AH-30	PID1 多段目标值 10	
AH-32	PID1 多段目标值 11	
AH-34	PID1 多段目标值 12	
AH-36	PID1 多段目标值 13	
AH-38	PID1 多段目标值 14	
AH-40	PID1 多段目标值 15	
AH-44	PID1 目标值 2 设定值	
AH-48	PID1 目标值 3 设定值	

设定值会根据 AJ-04、05 的设定进行比例缩放。

参数 No.	说明	数据
db-36	PID2 反馈数据监控	A) 转换值 (标尺转换单位) = ([AJ-05] - [AJ-04]) / 100.00 * [输入 (%)] + [AJ-04]
FA-36	PID2 目标值	
AJ-10	PID2 目标值设定值	B) 内部换算值 (%) = ([输入 (※)] - [AJ-04]) / ([AJ-05] - [AJ-04]) * 100.00 ※ 相当于 mon param 读取值 (标尺与 AJ-04、05 相同)
db-55	PID2 输出监控	
db-56	PID2 偏差监控	

设定值会根据 AJ-24、25 的设定进行比例缩放。

参数 No.	说明	数据
db-38	PID3 反馈数据监控	A) 转换值 (标尺转换单位)
FA-38	PID3 目标值	$= ([AJ-25] - [AJ-24]) / 100.00 * [输入 (%)] + [AJ-24]$
AJ-30	PID3 目标值设定值	B) 内部换算值 (%)
db-57	PID3 输出监控	$= ([输入 (*)] - [AJ-24]) / ([AJ-25] - [AJ-24]) * 100.00$
db-58	PID3 偏差监控	※ 相当于 mon param 读取值 (标尺与 AJ-24、25 相同)

设定值会根据 AJ-44、45 的设定进行比例缩放。

参数 No.	说明	数据
db-40	PID4 反馈数据监控	A) 转换值 (标尺转换单位)
FA-40	PID4 目标值	$= ([AJ-45] - [AJ-44]) / 100.00 * [输入 (%)] + [AJ-44]$
AJ-50	PID4 目标值设定值	B) 内部换算值 (%)
db-59	PID4 输出监控	$= ([输入 (*)] - [AJ-44]) / ([AJ-45] - [AJ-44]) * 100.00$
db-60	PID4 偏差监控	※ 相当于 mon param 读取值 (标尺与 AJ-44、45 相同)



# 8

## 异常及其处理

本章对发生异常时的程序动作、 DriveProgramming 功能固有的异常和原因及其处理方法进行介绍。

---

8-1 故障诊断 .....	8-2
8-1-1 发生异常时的 DriveProgramming 动作 .....	8-2
8-1-2 异常复位时的 DriveProgramming 动作 .....	8-3
8-1-3 报警代码一览 .....	8-4

## 8-1 故障诊断

下面对发生异常时的 DriveProgramming 动作、DriveProgramming 功能固有的错误代码及其处理方法进行说明。

### 8-1-1 发生异常时的 DriveProgramming 动作

DriveProgramming 动作的状态下，即使变频器检出跳闸 DriveProgramming 一般仍会继续动作。但检出 DriveProgramming 相关的跳闸 E043 ~ E045 时，将停止动作。此外，使用“on trip goto”指令时，在发生跳闸后可能会分支成其它处理。

on trip goto 的有无	错误内容		
	用户跳闸 E050 ~ E059	DriveProgramming 相关 E043 ~ E045	其它跳闸
无	继续动作	程序停止	继续动作
有	执行“on trip goto”指令 后，将切换至指定标签行 继续动作	程序停止	执行“on trip goto”指令 后，将切换至指定标签行 继续动作



#### 安全要点

DriveProgramming 功能的程序停止时，由 DriveProgramming 控制的输出端子将保持程序停止前的状态。

因此，请采用通过 DriveProgramming 启动信号和报警（跳闸）信号检出变频器 DriveProgramming 功能的程序停止，安全停止变频器周边设备的接线。

### 8-1-2 异常复位时的 DriveProgramming 动作

异常复位时的 DriveProgramming 动作因其复位输入方法和复位选择 (CA-72) 的设定而异。其动作状态如下表所述。

需重新启动 DriveProgramming 功能时，请在输入端子选择 (CA-01 ~ CA-11) 中设定“28: RS(复位)”，并输入对应的输入端子。

对象动作	程序状态	跳闸状态	控制端子的复位	
			CA-72 = 00、01	CA-72 = 02、03
Drive Programming 的动作	动作中	正常时	重新启动	继续动作
		跳闸中	重新启动	重新启动
	停止中	正常时	重新启动	继续停止 *1
		跳闸中	重新启动	重新启动
变频器本体的复位动作	正常时	复位	不动作	
	跳闸中	复位	复位	

\*1. 需重新启动程序时，请在将 EzSQ 功能选择 (UE-02) 设为“00(无效)”后，设定“02(始终有效)”。  
此外，UE-02 = 01 时，也可通过重新接通 PRG 端子进行重新启动。

## 8-1-3 报警代码一览

下面对 DriveProgramming 功能固有的报警代码和原因及其处理方法进行说明。

关于其它异常，请参阅 □ 《高功能型通用变频器 RX2 系列用户手册 (SBCE-CN5-437)》中的“12-2 错误编号和处理方法”。

报警代码	报警 (变频器发生跳闸的原因)	可能的原因	检查方法	处理方法
E043	EzSQ 不当指令错误	变频器中未下载程序，但 PRG 端子变为 ON。	可通过上传程序，检查变频器中是否存在程序。	请重新创建程序或将程序下载至变频器。
E044	EzSQ 嵌套次数错误	嵌套的子程序超过 8 层。	上传程序后检查嵌套层数。	请修改程序，确保嵌套层数为 8 层以下。
		“for next”指令的嵌套层数超过 8 层。		
		“if”指令的嵌套层数超出 8 层。		
E045	EzSQ 执行指令错误	“goto”指令的跳转目标为结束“for”或其它循环的“next”指令。	检查“goto”指令是否跳转至结束循环的指令。	请修改“goto”指令的跳转目标。
		四则运算指令导致以下异常。 上溢、下溢或除零。	检查程序，确认发生上溢、下溢或除零的指令。	请修改程序，确保四则运算指令不会导致上溢、下溢或除零。
		“ChgParam”、“MonParam”指令导致以下异常。 <ul style="list-style-type: none"> <li>• 变更为不存在的参数。</li> <li>• 写入值超出设定范围</li> <li>• 在变频器运行期间变更了运行期间不可变更的参数值</li> <li>• 变更了由软件锁限制更新的参数值 (软件锁有效时)</li> </ul>	<ul style="list-style-type: none"> <li>• 确认参数及参数的写入值。</li> <li>• 在变频器运行期间发生异常时，检查可否在变频器运行期间变更参数。</li> <li>• 检查软件锁选择 (UA-16) 的设定。</li> </ul>	<ul style="list-style-type: none"> <li>• 请修改参数或参数的写入值，确保其在设定范围内。</li> <li>• 请修改成不在运行期间变更运行期间不可变更的参数值的状态。</li> <li>• 请将软件锁选择 (UA-16) 变更为“00”，将软件锁设为无效。</li> </ul>



## 承诺事项

承蒙对欧姆龙株式会社(以下简称“本公司”)产品的一贯厚爱和支持,藉此机会再次深表谢意。

如果未特别约定,无论贵司从何处购买的产品,都将适用本承诺事项中记载的事项。

请在充分了解这些注意事项基础上订购。

### 1. 定义

本承诺事项中的术语定义如下。

- (1)“本公司产品”:是指“本公司”的FA系统机器、通用控制器、传感器、电子/结构部件。
- (2)“产品目录等”:是指与“本公司产品”有关的欧姆龙综合产品目录、FA系统设备综合产品目录、安全组件综合产品目录、电子/机构部件综合产品目录以及其他产品目录、规格书、使用说明书、操作指南等,包括以电子数据方式提供的资料。
- (3)“使用条件等”:是指在“产品目录等”资料中记载的“本公司产品”的使用条件、额定值、性能、运行环境、操作使用方法、使用时的注意事项、禁止事项以及其他事项。
- (4)“客户用途”:是指客户使用“本公司产品”的方法,包括将“本公司产品”组装或运用到客户生产的部件、电子电路板、机器、设备或系统等产品中。
- (5)“适用性等”:是指在“客户用途”中“本公司产品”的(a)适用性、(b)动作、(c)不侵害第三方知识产权、(d)法规法令的遵守以及(e)满足各种规格标准。

### 2. 关于记载事项的的注意事项

对“产品目录等”中的记载内容,请理解如下要点。

- (1)额定值及性能值是在单项试验中分别在各条件下获得的值,并不构成对各额定值及性能值的综合条件下获得值的承诺。
- (2)提供的参考数据仅作为参考,并非可在该范围内一直正常运行的保证。
- (3)应用示例仅作参考,不构成对“适用性等”的保证。
- (4)如果因技术改进等原因,“本公司”可能会停止“本公司产品”的生产或变更“本公司产品”的规格。

### 3. 使用时的注意事项

选用及使用本公司产品时请理解如下要点。

- (1)除了额定值、性能指标外,使用时还必须遵守“使用条件等”。
- (2)客户应事先确认“适用性等”,进而再判断是否选用“本公司产品”。“本公司”对“适用性等”不做任何保证。
- (3)对于“本公司产品”在客户的整个系统中的设计用途,客户应负责事先确认是否已进行了适当配电、安装等事项。
- (4)使用“本公司产品”时,客户必须采取如下措施:(i)相对额定值及性能指标,必须在留有余量的前提下使用“本公司产品”,并采用冗余设计等安全设计(ii)所采用的安全设计必须确保即使“本公司产品”发生故障时也可将“客户用途”中的危险降到最小程度、(iii)构建随时提示使用者危险的完整安全体系、(iv)针对“本公司产品”及“客户用途”定期实施各项维护保养。
- (5)因DDoS攻击(分布式DoS攻击)、计算机病毒以及其他技术性有害程序、非法侵入,即使导致“本公司产品”、所安装软件、或者所有的计算机器材、计算机程序、网络、数据库受到感染,对于由此而引起的直接或间接损失、损害以及其他费用,“本公司”将不承担任何责任。  
对于(i)杀毒保护、(ii)数据输入输出、(iii)丢失数据的恢复、(iv)防止“本公司产品”或者所安装软件感染计算机病毒、(v)防止对“本公司产品”的非法侵入,请客户自行负责采取充分措施。
- (6)“本公司产品”是作为应用于一般工业产品的通用产品而设计生产的。如果客户将“本公司产品”用于以下所列用途,则本公司对产品不作任何保证。但“本公司”已表明可用于特殊用途,或已与客户有特殊约定时,另行处理。
  - (a)必须具备很高安全性的用途(例:核能控制设备、燃烧设备、航空/宇宙设备、铁路设备、升降设备、娱乐设备、医疗设备、安全装置、其他可能危及生命及人身安全的用途)
  - (b)必须具备很高可靠性的用途(例:燃气、自来水、电力等供应系统、24小时连续运行系统、结算系统、以及其他处理权利、财产的用途等)
  - (c)具有苛刻条件或严酷环境的用途(例:安装在室外的设备、会受到化学污染的设备、会受到电磁波影响的设备、会受到振动或冲击的设备等)
  - (d)“产品目录等”资料中未记载的条件或环境下的用途
- (7)除了不适用于上述3.(6)(a)至(d)中记载的用途外,“本产品目录等资料中记载的产品”也不适用于汽车(含二轮车,以下同)。请勿配置到汽车上使用。关于汽车配置用产品,请咨询本公司销售人员。

### 4. 保修条件

“本公司产品”的保修条件如下。

- (1)保修期限 自购买之日起1年。(但是,“产品目录等”资料中有明确说明时除外。)
- (2)保修内容 对于发生故障的“本公司产品”,由“本公司”判断并可选择以下其中之一方式进行保修。
  - (a)在本公司的维修保养服务点对发生故障的“本公司产品”进行免费修理(但是对于电子、结构部件不提供修理服务。)
  - (b)对发生故障的“本公司产品”免费提供同等数量的替代品
- (3)当故障因以下任何一种情形引起时,不属于保修的范围。
  - (a)将“本公司产品”用于原本设计用途以外的用途
  - (b)超过“使用条件等”范围的使用
  - (c)违反本注意事项“3.使用时的注意事项”的使用
  - (d)非因“本公司”进行的改装、修理导致故障时
  - (e)非因“本公司”出品的软件导致故障时
  - (f)“本公司”生产时的科学、技术水平无法预见的原因
  - (g)除上述情形外的其它原因,如“本公司”或“本公司产品”以外的原因(包括天灾等不可抗力)

### 5. 责任限制

本承诺事项中记载的保修是关于“本公司产品”的全部保证。对于因“本公司产品”而发生的其他损害,“本公司”及“本公司产品”的经销商不负任何责任。

### 6. 出口管理

客户若将“本公司产品”或技术资料出口或向境外提供时,请遵守中国及各国关于安全保障进出口管理方面的法律、法规。否则,“本公司”有权不予提供“本公司产品”或技术资料。

IC320GC-zh

202304

注:规格如有变更,恕不另行通知。请以最新产品说明书为准。

欧姆龙自动化(中国)有限公司

<http://www.fa.omron.com.cn> 咨询热线:400-820-4535