

购买欧姆龙产品的客户须知

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. 关于记载事项的注意事項

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. 使用时的注意事項

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.使用时的注意事項”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

Cat. No. I580-CN5-02

欧姆龙自动化(中国)有限公司 版权所有 2023

变频器

MX2/RX系列

DriveProgramming

用户手册

3G3MX2-ZV1系列

3G3RX-V1系列

CX-Drive



I580-CN5-02

© OMRON, 2014

版权所有。未经OMRON公司事先书面允许，不得将本出版物的任何部分以任何形式或任何方式（机械、电子、照相、录制或其它方式）进行复制、存入检索系统或传送。

使用本手册所包含的信息不负专利责任。由于OMRON公司始终致力于改进其高质量产品，所以本手册所包含的信息可随时改变而不另行通知。虽然在编制本手册时收录了每种可能的注意事项，但对于仍然可能出现的错误或遗漏，OMRON公司不承担任何责任。同样，由于使用本手册所包含的信息而造成的损害也不承担任何责任。

引言

感谢您购买变频器 / 伺服支持工具 CX-Drive 和 3G3MX2-ZV1/3G3RX-V1 系列变频器。

本手册对变频器用 DriveProgramming 的规格和操作方法进行说明。

使用本产品时，请参阅“MX2-V1 系列多功能小型变频器用户手册”（样本编号：I585）或“3G3RX-V1 系列高性能通用型变频器用户手册”（样本编号：I578）及“CX-Drive 操作手册”（样本编号：W453）。

读者对象

本手册主要供下列人员使用，这些人员必须具备电气系统相关知识（电气工程师或同等水平者）。

- 负责引进 FA 系统的人员；
- 负责 FA 系统设计的人员；
- 负责 FA 系统安装和连接的人员；
- 负责 FA 系统及设备管理的人员。

使用须知

本手册介绍了使用驱动编程（DriveProgramming）所需的信息。

使用前请仔细阅读本手册，并充分理解说明内容。

阅读后，请将其放于便于取及之处妥善保管，以便随时查阅。

务必确保将本手册交付至终端用户手中。

手册构成

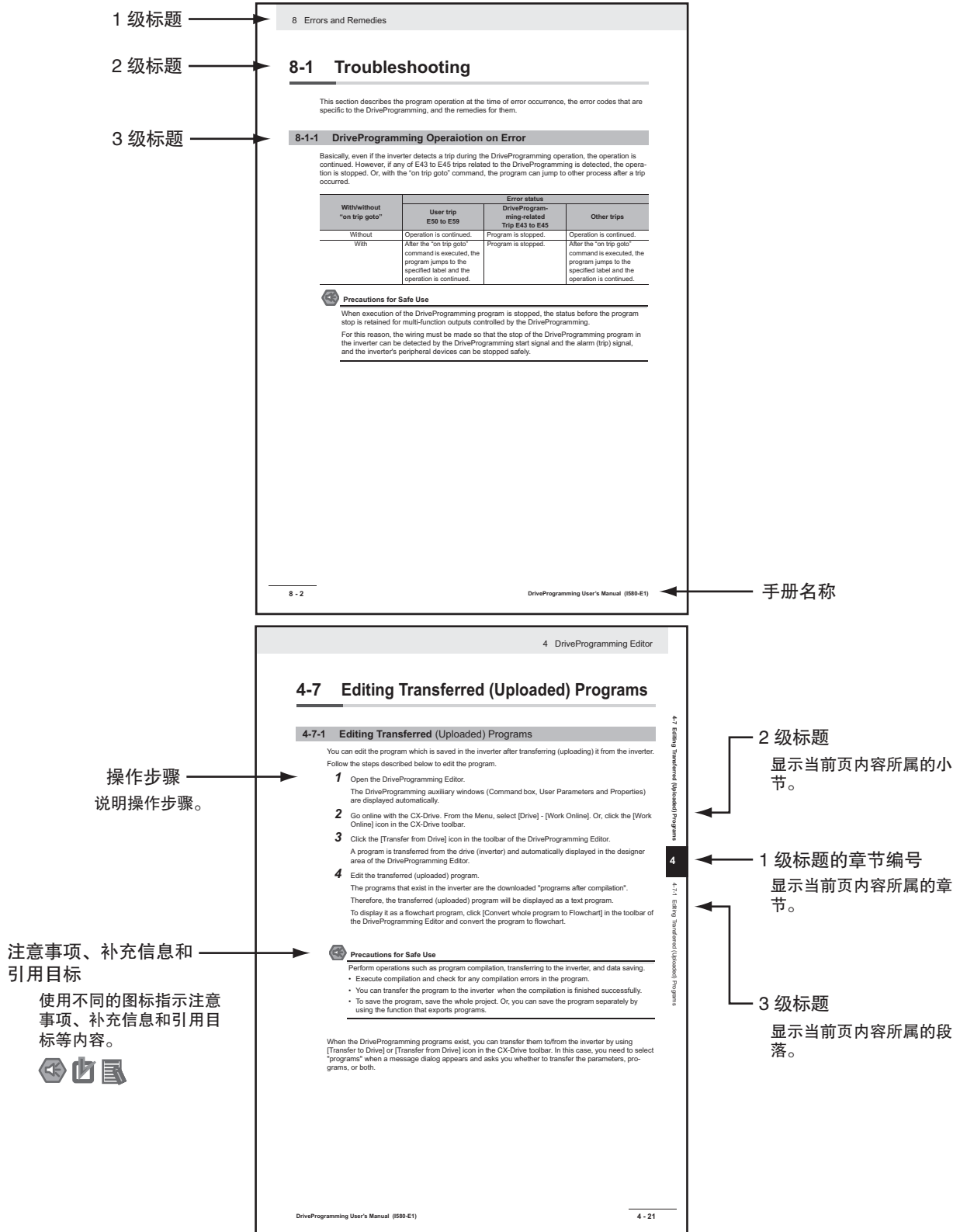
为便于用户查阅，本手册由下述章节构成。

| 第 ### 章 | 概述 |
|-------------------------------------|--|
| 第 1 章节 概述 | 本章节对 DriveProgramming 进行了概述，并介绍了 DriveProgramming 的系统配置。 |
| 第 2 章节 规格 | 本章节阐述 DriveProgramming 的规格。 |
| 第 3 章节 DriveProgramming 的操作步骤 | 本章节介绍 DriveProgramming 的操作步骤以及相关的参数和程序结构。 |
| 第 4 章节 DriveProgramming Editor | 本章节介绍如何启动 DriveProgramming Editor、保存和加载数据以及编辑器各部分详细信息。 |
| 第 5 章节 DriveProgramming 用户变量 | 本章节介绍 DriveProgramming 的用户变量。 |
| 第 6 章节 DriveProgramming 命令 | 本章节介绍适用于 DriveProgramming 的命令。 |
| 第 7 章节 使用 DriveProgramming 参数时的注意事项 | 本章节介绍为 DriveProgramming 使用参数时的注意事项。 |
| 第 8 章节 错误与纠正措施 | 本章节介绍出错时的程序操作、与 DriveProgramming 有关的错误以及错误原因与纠正措施。 |

手册结构

页面结构

本手册采用下列页面结构和图标。



注 上述页面仅为说明示例，并非是本手册中的实际内容。

特殊信息

本手册中的特殊信息可分为以下几类：



安全使用注意事项

表示允许和禁止事项，以确保安全使用本产品。



正确使用注意事项

表示允许和禁止事项，以确保正常运行和性能。



附加信息

表示需要阅读的附加信息，用于加深理解或简化操作。

本手册中的章节

| | | |
|---|------------------------------|---|
| 1 | 概述 | 1 |
| 2 | 规格 | 2 |
| 3 | DriveProgramming 的操作步骤 | 3 |
| 4 | DriveProgramming Editor | 4 |
| 5 | DriveProgramming 用户变量 | 5 |
| 6 | DriveProgramming 命令 | 6 |
| 7 | 使用 DriveProgramming 参数时的注意事项 | 7 |
| 8 | 错误与纠正措施 | 8 |

目录

| | |
|----------------|----|
| 引言 | 1 |
| 手册构成 | 2 |
| 手册结构 | 3 |
| 本手册中的章节 | 5 |
| 目录 | 6 |
| 阅读和理解本手册 | 9 |
| 安全注意事项 | 12 |
| 安全使用注意事项 | 15 |
| 规定和标准 | 16 |
| 相关手册 | 17 |
| 修订记录 | 18 |

第 1 章 概述

| | |
|-------------------------------|-----|
| 1-1 DriveProgramming 概述 | 1-2 |
| 1-2 准备工作和系统配置 | 1-4 |

第 2 章 规格

| | |
|--------------|-----|
| 2-1 规格 | 2-2 |
|--------------|-----|

第 3 章 DriveProgramming 的操作步骤

| | |
|---|------|
| 3-1 操作步骤 | 3-2 |
| 3-2 与 DriveProgramming 有关的参数 | 3-5 |
| 3-3 程序结构 | 3-13 |
| 3-3-1 任务 | 3-13 |
| 3-3-2 子程序 | 3-13 |
| 3-3-3 任务处理 | 3-13 |
| 3-3-4 DriveProgramming 启动 / 停止和任务操作 | 3-15 |
| 3-3-5 DriveProgramming 重新启动 | 3-17 |
| 3-3-6 跳闸时的任务操作 | 3-18 |

第 4 章 DriveProgramming Editor

| | |
|--|-----|
| 4-1 启动 DriveProgramming Editor | 4-2 |
| 4-2 DriveProgramming Editor 的各部分 | 4-6 |
| 4-2-1 DriveProgramming Editor | 4-6 |
| 4-2-2 工具栏 | 4-6 |

| | | |
|-------|------------------------------|------|
| 4-2-3 | DriveProgramming 区域 | 4-9 |
| 4-2-4 | 工具箱窗口 | 4-13 |
| 4-2-5 | 块参数窗口 | 4-14 |
| 4-2-6 | 属性窗口 | 4-15 |
| 4-2-7 | 输出窗口中的错误列表选项卡 | 4-16 |
| 4-3 | 添加、删除和重命名任务 | 4-17 |
| 4-4 | 插入、删除和调用子程序 | 4-18 |
| 4-5 | 创建流程图程序 | 4-19 |
| 4-6 | 创建文本程序 | 4-20 |
| 4-7 | 编辑传送 (上传) 的程序 | 4-21 |
| 4-8 | 保存程序 | 4-22 |
| 4-9 | 传送和验证程序 | 4-24 |
| 4-10 | 执行程序 (DriveProgramming 功能选择) | 4-25 |
| 4-11 | 其他有用功能 | 4-27 |

第 5 章 DriveProgramming 用户变量

| | | |
|-----|-------------|------|
| 5-1 | 用户变量和用户参数 | 5-2 |
| 5-2 | 输入 / 输出端子变量 | 5-4 |
| 5-3 | 定时器变量 | 5-9 |
| 5-4 | 变频器设置变量 | 5-10 |
| 5-5 | 变频器监控变量 | 5-12 |
| 5-6 | 多功能输入变量 | 5-14 |
| 5-7 | 多功能输出变量 | 5-16 |

第 6 章 DriveProgramming 命令

| | | |
|-----|-------------|------|
| 6-1 | 命令类别 | 6-2 |
| 6-2 | 命令格式 | 6-3 |
| 6-3 | 命令列表 | 6-4 |
| 6-4 | 程序控制命令 | 6-10 |
| 6-5 | 四则运算和逻辑运算命令 | 6-23 |
| 6-6 | I/O 控制单元 | 6-36 |
| 6-7 | 定时器控制命令 | 6-46 |
| 6-8 | 参数控制命令 | 6-52 |
| 6-9 | 变频器控制命令 | 6-58 |

第 7 章 使用 DriveProgramming 参数时的注意事项

| | | |
|-----|--------------------|-----|
| 7-1 | 受设置顺序影响的变频器参数 | 7-2 |
| 7-2 | 受额定电流 [%] 影响的参数 | 7-4 |
| 7-3 | 受 PID 启用 / 禁用影响的参数 | 7-5 |

第 8 章 错误与纠正措施

| | |
|--|-----|
| 8-1 故障诊断 | 8-2 |
| 8-1-1 出错时的 DriveProgramming 操作 | 8-2 |
| 8-1-2 复位错误时的 DriveProgramming 操作 | 8-3 |
| 8-1-3 报警代码一览表 | 8-4 |

阅读和理解本手册

保证及有限责任声明

保证声明

OMRON 的排他性保证是指产品自售出起一年 (或其它指定期间) 内在材料和工艺上无缺陷。OMRON 未以明示或暗示的方式表述或保证产品的非侵权性、适销性或特定用途的适用性。任何买方或用户均同意自主决定这些产品是否适当满足其预定用途。OMRON 对于所有其它明示或暗示的保证概不负责。

有限责任

OMRON 对于任何与产品相关的特殊、间接或直接损坏、利润损失或商业损失概不负责，不论此类索赔是基于合同、保证、疏忽还是严格责任。
不论何种情况，OMRON 对于任何诉讼行为的赔偿责任均不得超过相关产品的价格。
除非 OMRON 经分析后确认产品的使用、存放、安装和维护得当且未遭污染、滥用、误用或者不当改造或修理，否则在任何情况下，OMRON 对于与产品相关的保证、修理或其它主张不承担任何责任。

应用注意事项

适用性声明

OMRON 对于客户在其应用中的产品组合或产品使用的标准、规范或条例方面的合规性不承担任何责任。根据客户的要求，OMRON 将提供相应的第三方认证来明确适用于产品的额定值和使用限制。此信息本身不足以充分确定产品与终端产品、机器、系统及其它应用或用途组合的适用性。

以下为必须特别注意的应用示例，但下述内容并非包括产品所有可能的用途，也并不暗示所列用途均适用于该类产

- 品：
- 户外用途、涉及潜在化学污染或电气干扰的用途或本手册中未说明的状况或用途。
 - 核能控制系统、燃烧系统、铁道系统、航空系统、医疗设备、娱乐器械、车辆、安全设备及个别行业或政府规范监管的设备。
 - 存在人身或财产安全隐患的系统、机器及设备。请务必了解并遵守产品适用的所有禁用条款。

如果产品整体设计不足以应对此类风险，且未在整个设备或系统内针对特定用途妥善调校并安装 OMRON 产品，则不得将产品用存在严重人身或财产隐患的场合。

可编程产品

使用可编程产品时，OMRON 不对用户的程序或其引起的后果承担任何责任。

免责声明

规格变更

基于产品改进和其它原因，产品规格及附件可能会随时变更。公司通常在公布规格、性能或重大结构变更后更改型号。但对某些产品规格进行变更时并不另行通知。在不确定规格时，我们会根据客户的要求为其应用场合指定特殊的型号或设立关键的规格。请随时垂询 OMRON 代理商以确认所购产品的实际规格。

尺寸与重量

尺寸和重量仅为公称值，即使已说明公差，也不得用于制造用途。

性能参数

本手册所提供的性能参数仅供用户确定适用性时使用，并不予以保证。这些数据仅表示在 OMRON 测试条件下的结果，用户必须将其与实际应用条件相联系。实际性能遵守 OMRON 保证声明和有限责任条款的规定。

错误与疏漏



本手册所述信息经仔细审核，力求准确无误；但对于笔误、排版或校对错误或疏漏，我方概不负责。

安全注意事项







安全使用的符号和含义

本手册使用下列注意事项和警告标识提示安全使用 DriveProgramming 所需关注的信息。此处所示信息与安全息息相关，请务必严格遵守。

警告标识的含义

| | |
|---|--|
|  警告 | <p>表示潜在的危險情况，如不加以避免，可能会造成死亡或严重伤害。此外，还可能会造成严重的财产损失。</p> |
|  注意 | <p>表示潜在危險情况，若不避免，可能会造成轻度或中度人身伤害或财产损失。</p> |







本手册中的警示符号

| | |
|---|---|
|  | <p>⊘ 表示禁止事项（一定不能做的事项）。 采用图示或在⊘标识内或旁边附带文字说明表示特定内容。 当其出现在左侧时，表示“禁止拆卸”。</p> |
|  | <p>△表示危險和注意事项。 采用图示或在△标识内或旁边附带文字说明表示特定内容。 当其出现在左侧时，表示“谨防触电”。</p> |
|  | <p>△表示危險和注意事项。 采用图示或在△标识内或旁边附带文字说明表示特定内容。 当其出现在左侧时，表示“非特定情形下的常规危險事项”。</p> |
|  | <p>△符号表示注意事项（包含警告事项）。 采用图示或在△标识内或旁边附带文字说明表示特定内容。 当其出现在左侧时，表示“表面温度过高，存在灼伤的危險”。</p> |
|  | <p>●表示强制性事项（必须要做的事项）。 采用图示或在●标识内或旁边附带文字说明表示特定内容。 当其出现在左侧时，表示“一般强制性事项”。</p> |
|  | <p>●表示强制性事项（必须要做的事项）。 采用图示或在●标识内或旁边附带文字说明表示特定内容。 当其出现在左侧时，表示“需要接地”。</p> |


警告

| | |
|---|--|
|  | 请关闭电源并进行正确接线。 否则可能会因触电而导致严重的人身伤害。 |
|  | 必须由具备相应资质的人员执行接线作业。 否则可能会因触电而导致严重的人身伤害。 |
|  | 在输入电源接通的情况下，请勿更改接线和滑动开关、运转或拆卸操作器及选装设备以及更换散热片。 否则可能会因触电而导致严重的人身伤害。 |
|  | 请务必对装置进行接地。否则可能会因触电或起火而导致严重的人身伤害。 (200V 级别：D 类接地，400V 级别：C 类接地) |
|  | 请勿在接通电源和断电后 10 分钟内拆下端子盖。 否则可能会因触电而导致严重的人身伤害。 |
|  | 请勿用湿手操作操作器或开关。 否则可能会因触电而导致严重的人身伤害。 |
|  | 必须在切断电源后对变频器进行检查。 否则可能会因触电而导致严重的人身伤害。 即使在激活紧急关闭功能时也无需关闭主电源。 |
|  | 请勿触摸变频器散热片、制动电阻和电机，因为其在通电及切断电源后的一段时间内温度较高，否则可能会导致灼伤。 |
|  | 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。 |
|  | 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。 |
|  | 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。 |
|  | 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。 |
|  | 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统。 |
|  | 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）。 |

⚠ 注意

| | |
|--|---|
|  | 请勿将电阻直接连接至端子（+1、P/+2, N/-）上，否则可能会导致小范围起火、发热或设备受损。 |
|  | 请安装一台止动设备，以确保安全。否则可能会导致轻微受伤。 (报闸并非是为确保安全而设计的止动设备。) |
|  | 请务必使用指定的制动电阻 / 再生制动单元。使用制动电阻时，请安装一个热动继电器，以监控电阻的温度。否则，可能会因制动电阻 / 再生制动单元发热而导致中度烧伤。请配置一个控制序列，以便在检测到制动电阻 / 再生制动单元异常过热时切断变频器的电源。 |
|  | 变频器内有大量高压部件，若发生短路，则可能会导致变频器或其它部件受损。请在开口部位安放防护盖或采取其它防护措施，以确保在执行安装和接线作业时无切屑或废旧导线等金属物进入内部。 |
|  | 请采取一定的安全防护措施，如在电源侧安装一个与变频器容量相匹配的塑壳断路器（MCCB）。 否则，可能会因负载短路而导致财产损失。 |
|  | 请勿拆解、修理或改造本产品，否则可能会导致受伤。 |

安全使用注意事项

操作和调整

- 若参数设定不当，则在启动、调整、维护或更换作业时可能会产生意外动作。因此，请务必确保参数设定正确无误后再执行操作。
- 若 DriveProgramming 因多功能输出而停止，则会保持输出状态。请采取适当的安全措施，如关停外围设备等。
- 若 DriveProgramming 中采用时钟指令，则可能会因 LCD 数字操作器蓄电池电量不足而导致意外动作。请采取适当的措施，如通过将时钟数据恢复为初始设定并使变频器 / 程序停止运行以检测蓄电池电量不足等。
当 LCD 数字操作器拆下或断开时，DriveProgramming 会在始终指令的作用下处于等待状态。
- 使用 DriveProgramming 时，请务必在开始操作前确认程序数据已成功下载。

规定和标准

本产品属于日本《外汇和外贸管理法》规定的需要出具强制出口证明或出口许可证的商品（或技术）类别，根据该法规定，若要出口（或向非定居外国人提供）本产品的任何部分，均需提供出口证明或出口许可证（或获得劳务交易批准）。

商标

Windows、Windows XP、Windows Vista 和 Windows 7 为微软公司在美国和其它国家的注册商标。

相关手册

有关操作本产品所需连接设备的相关信息，请参阅下述相关产品手册信息。

| 名称 | 样本编号 |
|--------------------------|------|
| MX2-V1 系列多功能小型变频器用户手册 | I585 |
| 3G3RX-V1 系列高功能通用型变频器用户手册 | I578 |
| CX-Drive 操作手册 | W453 |



附加信息

有关变频器操作信息，请参阅“MX2-V1 系列多功能小型变频器用户手册”（样本编号：I585）或“3G3RX-V1 系列高功能通用型变频器用户手册”（样本编号：I578）。

修订记录

手册封面和封底上样本编号的后缀部分即为修订号。

例

Cat.No. I580-CN5-02

↑ 修订号

| 修订号 | 修订日期 | 修订内容 |
|-----|------------|-----------|
| 01 | 2013 年 3 月 | 初版 |
| 02 | 2023 年 4 月 | 增加安全对策的说明 |

1

概述

本章节对 DriveProgramming 进行了概述，并介绍了 DriveProgramming 的系统配置。

| | |
|-------------------------------|-----|
| 1-1 DriveProgramming 概述 | 1-2 |
| 1-2 准备工作和系统配置 | 1-4 |

1-1 DriveProgramming 概述

DriveProgramming 是变频器内置的简单顺序控制功能。

若要创建顺序控制程序并检查它们的状态，可使用变频器 / 伺服支持工具 CX-Drive。

将创建的程序传送（下载）到 3G3MX2-ZV1/3G3RX-V1 系列变频器后，这款独立型变频器就能执行简单的顺序控制。

DriveProgramming 的特点

DriveProgramming 具有如下特点。

- DriveProgramming 同时支持流程图和文本语言编程。
- 您创建的程序最多可划分成五个任务。
- 这五个任务可并行处理。
- 也可以设置多功能输入端子，然后在外部执行用户程序。
- 您可以通过为多功能 I/O 端子分配参数的方式使用它们。
- 在现场调整时，使用数字操作器可以更改所需的频率、加速 / 减速时间及其他参数（变量）的设置，而且更改时无需连接计算机，只需指定用户参数（P100 到 P131）即可。
- 由于用户程序存储在变频器内部 EEPROM 中，您可以在变频器接通电源后立即启动程序。
- 可选的 LCD 数字操作器（型号：3G3AX-OP05）内置了时钟功能。购买并连接 LCD 数字操作器后，您可以使用操作器的时钟功能创建程序。

MX2

但 3G3MX2-ZV1 系列变频器不可与 LCD 数字操作器（型号：3G3AX-OP05）连接，因此也就无法使用其时钟功能。



安全使用注意事项

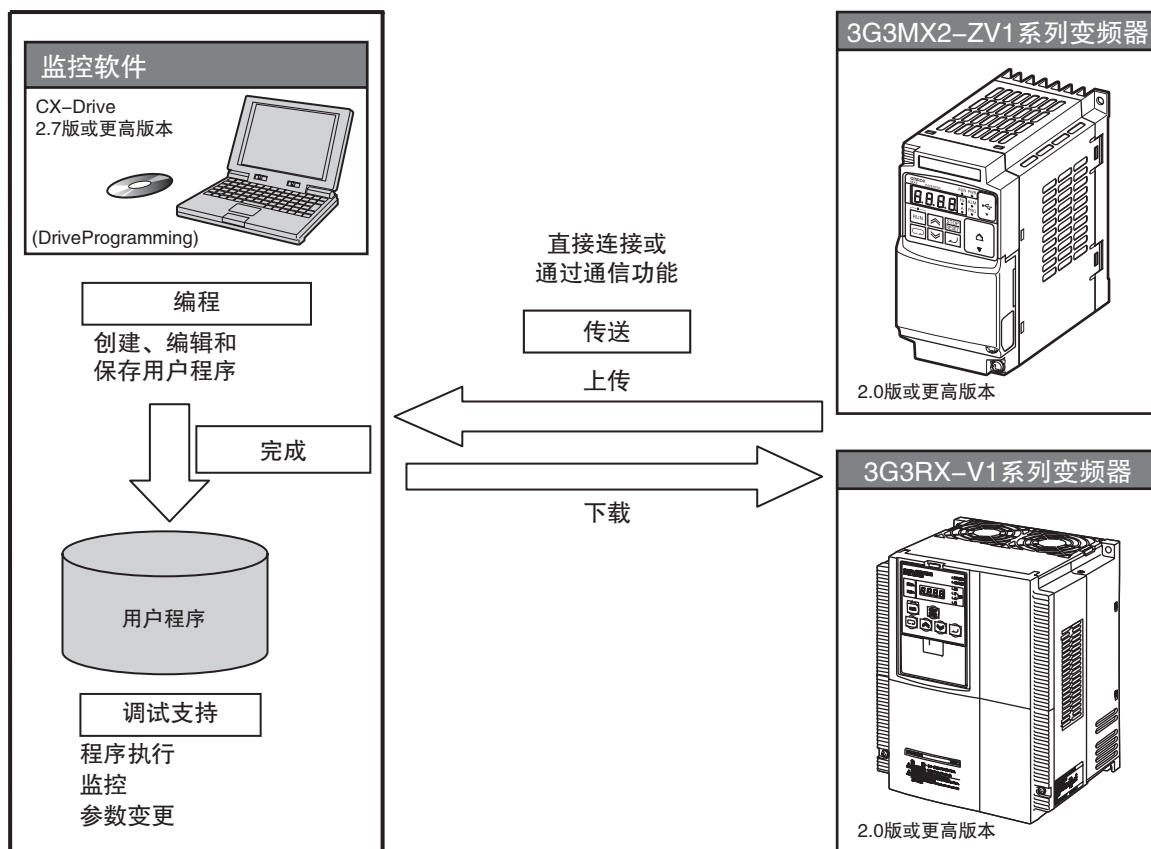
在 DriveProgramming 中使用时钟命令时，如果 LCD 数字操作器的电池电量不足，则会导致意外动作。

请采取检查电池电量等措施，比如说检查时钟数据是否恢复初始设定并已停止变频器或程序，籍此判断电池电量是否不足。

拆下或断开 LCD 数字操作器时，DriveProgramming 会在时钟命令的作用下处于等待状态。

下表显示了 CX-Drive 中可用的 DriveProgramming Editor 的主要功能。

| 功能 | 描述 |
|------|---------------------------------|
| 编程 | 支持创建、编辑、保存、读取和打印用户程序。 |
| 编译 | 检查用户程序并生成中间代码。 |
| 传送 | 将用户程序下载到变频器。 从变频器上传用户程序。 |
| 调试支持 | 开始并停止执行程序。 用户可检查变频器状态监视器等内容。 |



1-2 准备工作和系统配置

为了使用 CX-Drive 中的 DriveProgramming 创建用户程序并在 3G3MX2-ZV1/3G3RX-V1 系列变频器中予以执行，您必须准备好以下事项。

- 3G3MX2-ZV1/3G3RX-V1 系列变频器 (2.0 版或更高版本)
- 个人计算机 (PC)(Windows 系统)

| 32 位 PC | 64 位 PC |
|--|---------------------------|
| Windows XP SP3、Windows Vista 和 Windows 7 | Windows Vista 和 Windows 7 |

- CX-Drive 需要使用以下版本。

MX2: 3G3MX2-ZV1 系列变频器要使用 2.8 版或更高版本

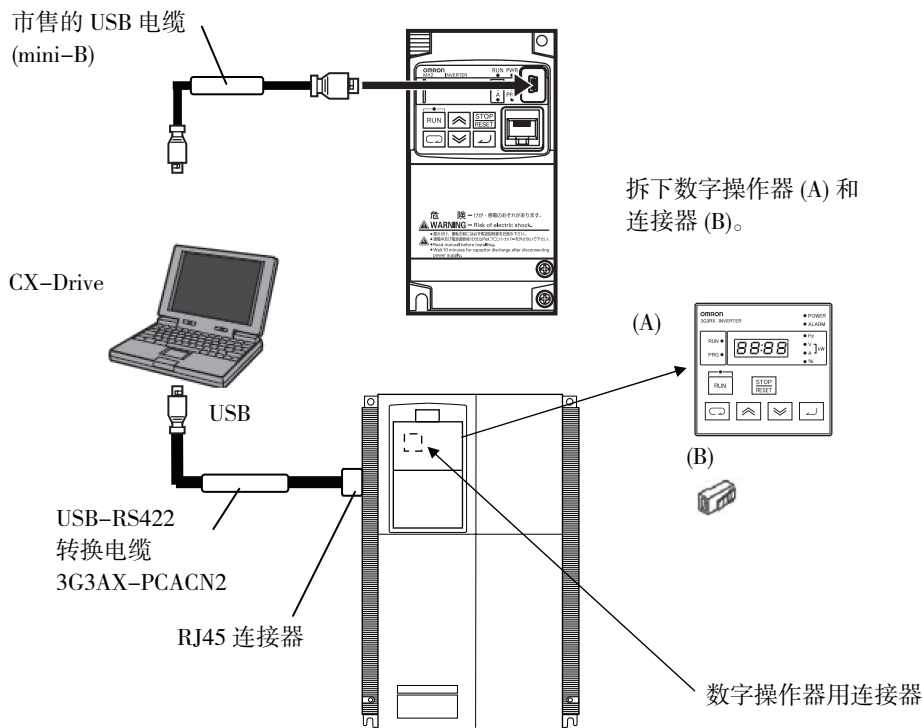
RX: 3G3RX-V1 系列变频器要使用 2.7 版或更高版本

(CX-Drive 包含在 FA 集成工具包 CX-One 中。)

- 准备以下 PC- 变频器连接电缆。

MX2: 市售的 USB 电缆 (mini-B)

RX: USB 到 RJ-45 转换电缆 (型号: 3G3AX-PCACN2)



正确使用注意事项

- 只有 3G3MX2-ZV1/3G3RX-V1 系列变频器具备 DriveProgramming 功能。

请务必检查变频器铭牌上刻印的型号 (3G3MX2-ZV1/3G3RX-****-V1) 或版本 (2.0 版)，因为传统型号 3G3MX2/3G3RX 系列变频器不具有该功能。

- DriveProgramming 功能包含在以下版本或更高版本的 CX-Drive 中。

如果您的 CX-Drive 版本较低，您需要升级到相应版本。

MX2: 2.8 版或更高版本 (2.7 版或更低版本无法执行操作。)

RX: 2.7 版或更高版本 (2.6 版或更低版本无法执行操作。)



附加信息

有关如何安装 CX-Drive 的信息，请参阅《CX-Drive 操作手册》(W453)。

EtherCAT/CompoNet/DeviceNet 通信

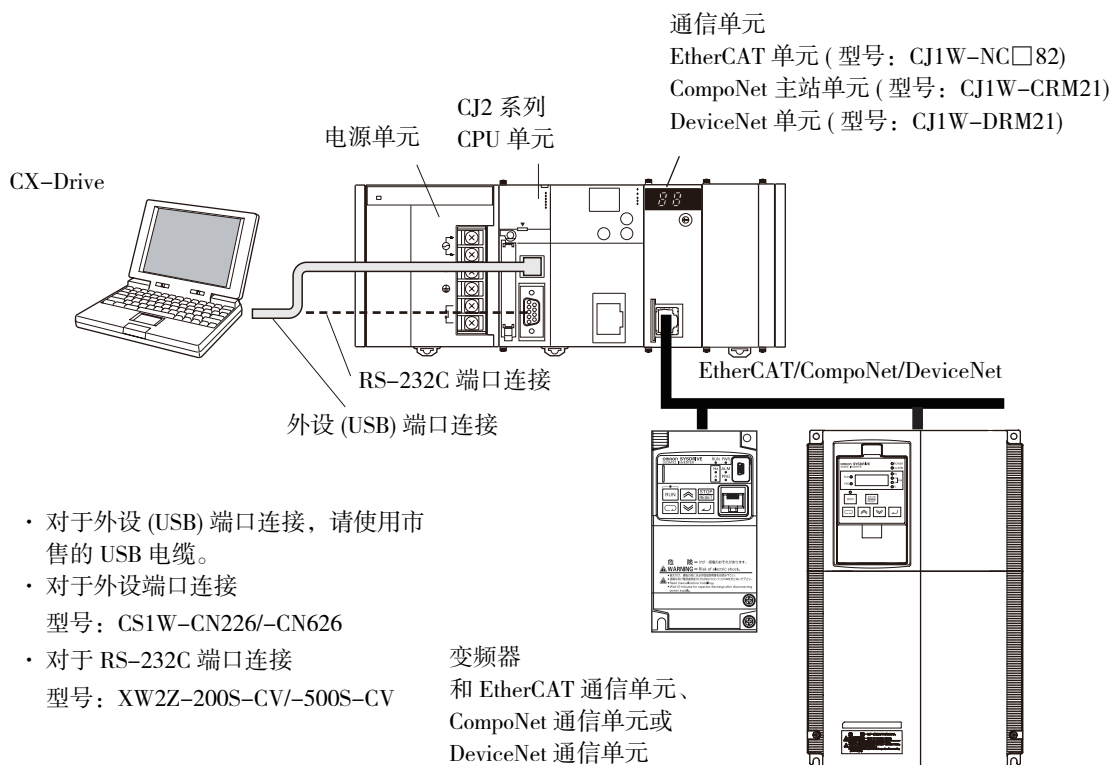
借助可编程控制器上安装的 EtherCAT 主站位置控制单元 (型号: CJ1W-NC□82)、CompoNet 主站单元 (型号: CJ1W-CRM21/CS1W-CRM21) 或 DeviceNet 单元 (型号: CJ1W-DRM21/CS1W-DRM21-V1), CX-Drive 可通过通信功能连接到变频器。

若要借助可编程控制器在变频器和 CX-Drive 之间建立通信连接, 您必须事先在 CX-Drive 中进行以下设置。

- 选择驱动器项目的属性, 然后单击 [Drive Type](驱动器类型) 下的 [Settings](设置), 将选件板类型设置为变频器通信单元。
- 选择驱动器项目属性, 然后选择 [Connection type](连接类型) 下的连接网络。

接着, 单击 [Settings](设置), 在 [Network](网络) 下进行网络配置, 以及在 [Gateway PLC](网关 PLC) 下设置连接 CX-Drive 和 PLC 的连接方法。

● 配置



- 对于外设 (USB) 端口连接, 请使用市售的 USB 电缆。
- 对于外设端口连接
型号: CS1W-CN226/-CN626
- 对于 RS-232C 端口连接
型号: XW2Z-200S-CV/-500S-CV

在您的 Windows 个人计算机上安装 CX-Drive。将计算机连接到可编程控制器, 然后通过 EtherCAT、CompoNet 或 DeviceNet 通信与变频器建立起在线连接。

建立好在线连接后, 您可以使用 DriveProgramming Editor 创建用户程序并将它们传送 (下载) 到变频器。

2

规格

本章节阐述 DriveProgramming 的规格。

| | |
|--------------|-----|
| 2-1 规格 | 2-2 |
|--------------|-----|

2-1 规格

下表列出了与 DriveProgramming 相关的规格。

| 项目 | 规格 | | |
|-----------|--------|--|---|
| 程序规格 | 编程语言 | 流程图和文本语言方法 | |
| | 输入设备 | Windows 个人计算机 (操作系统: Windows XP-SP3、Windows Vista 或 Windows 7) | |
| | 程序容量 | 最多 1,024 步: 6 KB (总共五个任务, 最多 1,024 步) | |
| | 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"> · 编辑和显示程序 · 编译程序 (检查程序语法) · 下载、上传和全部清除程序 | |
| | 执行格式 | <ul style="list-style-type: none"> · 由解释器执行 · 执行周期: 2ms/步 (可通过并-行处理 5 个任务来执行 5 个命令) · 支持子程序调用 (最多可嵌套 8 层) | |
| 输入 / 输出功能 | 外部输入 | DriveProgramming 启动 | 在 DriveProgramming 功能选择 (A017) 中选择 <ul style="list-style-type: none"> · 通过多功能输入 PRG 端子 (A017 = 01) 进行启动 / 停止 MX2: 分配给多功能输入 S1 到 S7 选择 (C001 到 C007)。 RX: 分配给多功能输入 S1 到 S8 选择 (C001 到 C008) <ul style="list-style-type: none"> · 接通 / 关闭电源时启动 / 停止 (A017=02) |
| | | 多功能输入 | MX2: X(00) 到 X(07)/8 个点 <ul style="list-style-type: none"> · 多功能输入 S1 到 S7 端子 · X(07) 用于脉冲串输入 RP 端子 (仅在 P003 = 02 时启用) RX: X(00) 到 X(07)/8 个点 <ul style="list-style-type: none"> · 多功能输入 S1 到 S8 端子 |
| | | 频率参考输入 (多功能模拟输入) | XA(0): 0 到 10V(FV 端子) |
| | | | XA(1): 4 到 20mA(FI 端子) |
| | 外部输出 | 多功能输出 / 多功能继电器输出 | MX2: Y(00) 到 Y(02)/3 个点 <ul style="list-style-type: none"> · 多功能输出 P1 和 P2 端子 · 多功能继电器输出端子 (MA、MB) RX: Y(00) 到 Y(05)/6 个点 <ul style="list-style-type: none"> · 多功能输出 P1 到 P5 端子 · 多功能继电器输出端子 (MA、MB) |
| | | | 显示器输出 (多功能数字输出) |
| | | 显示器输出 (多功能模拟输出) | YA(1): 0 到 10V(AM 端子) YA(2): MX2: 无适用的功能 RX: 4 到 20mA(AMI 端子) |

| 项目 | 规格 | |
|------|---|---|
| 命令 | 程序控制命令 <ul style="list-style-type: none"> · 循环 (“for”) · 无条件转移 (“goto”) · 时间控制 (“wait”) · 条件转移 (“if then”、“ifs then”、“select case”、“until”、“while”) · 子程序 (“call”、“sub”) · 其他 (“entry”、“end”、“inc”、“dec”) | |
| | 算术命令 <ul style="list-style-type: none"> · 四则运算 (+、-、* 和 /) · 余数 (“mod”) 和分配 (=) · 绝对值 (“abs”) · 逻辑运算 (“or”、“and”、“xor”、“not”) | |
| | I/O 控制 <ul style="list-style-type: none"> · 多功能 I/O(位输入、字输入、位输出和字输出) · 读取变频器输入端子 · 读取 / 写入变频器输出端子 | |
| | 定时器控制 | 通过定时器和定时器计数器控制进行接点控制 |
| | 参数控制 | · 更改指定参数编号的设置数据 · 将参数设置保存到 EEPROM 仅限 RX: 当前时间读数 (一次 / 持续) |
| | 变频器控制 | · 执行并停止正向 / 反向操作 · 通过 DriveProgramming 生成一个跳闸 (E50 到 E59/10 个点) · 频率参考和加速 / 减速时间设置 |
| 功能变量 | 用户参数变量 | U(00) 到 U(31)/32 个点 |
| | 内部用户变量 | UL(00) 到 UL(07)/8 个点 |
| | 频率参考变量 | SET-Freq |
| | 加速时间变量 | ACCEL |
| | 减速时间变量 | DECEL |
| | 变频器监控器变量 | 变频器的监控器功能 (d001 到 d102) 可用作变量。 MX2 和 RX: FM、Iout、Dir、PID-FB、F-CNV、Tmon、Vout、Power、RUN-Time、ON-Time、POS、STATUS、DCV、ERR-CNT、ERR(1) 到 ERR(6) 以及 UMon(0) 到 UMon(2) |
| | 多功能输入变量 | 可以使用变频器多功能输入 S1 选择 (C001) 的功能选项。 MX2 和 RX: FW、RV、CF1、CF2、CF3、CF4、JG、DB、SET、TCH、FRS、EXT、USP、CS、SFT、AT、RS、STA、STP、F-R、PID、PIDC、UP、DWN、UDC、OPE、SF1、SF2、SF3、SF4、SF5、SF6、SF7、OLR、TL、TRQ1、TRQ2、BOK、LAC、PCLR、ADD、F-TM、ATR、KHC、AHD、CP1、CP2、CP3、ORL、ORG 和 SPD 仅限 MX2: RS485、HLD、ROK 和 DISP 仅限 RX: SET3、CAS、PPI、ORT、STAT、SON、FOC、FOT、ROT、PCNT 和 PCC |

| 项目 | 规格 |
|----------|---|
| 功能变量 | 可以使用变频器多功能输出 P1 选择 (C021) 的功能选项。 MX2 和 RX: RUN、FA1、FA2、OL、OD、AL、FA3、OTQ、UV、TRQ、RNT、ONT、 THM、BRK、BER、ZS、DSE、POK、FA4、FA5、OL2、ODc、OIDc、FBV、 NDc、LOG1、LOG2、LOG3、WAC、WAF、FR、OHF、LOC、IRDY、FWR、 RVR、MJA、WCO 和 WCOI 仅限 MX2: FREF、REF、SETM 和 EDM 仅限 RX: IP、O2Dc、LOG4、LOG5、LOG6 和 WCO2 |
| 输入端子变量 | MX2: X(00) 到 X(07)/8 个点 注: X(07) 用于脉冲串输入 RP 选择 (仅在 P003 = 02 时启用) RX: X(00) 到 X(07)/8 个点 |
| 输出端子变量 | MX2: Y(00) 到 Y(02)/3 个点 RX: Y(00) 到 Y(05)/6 个点 |
| 内部用户接点 | UB(0) 到 UB(7)/8 个点 |
| 定时器输出接点 | TD(0) 到 TD(7)/8 个点 |
| 定时器计数器变量 | TC(0) 到 TC(7)/8 个点 |
| 模拟输入端子变量 | MX2: XA(0) 和 XA(1) RX: XA(0) 到 XA(2) |
| 模拟输出端子变量 | MX2: YA(0) 和 YA(1) RX: YA(0) 到 YA(2) |

3

DriveProgramming 的操作步骤

本章节介绍 DriveProgramming 的操作步骤以及相关的参数和程序结构。

| | |
|---|------|
| 3-1 操作步骤 | 3-2 |
| 3-2 与 DriveProgramming 有关的参数 | 3-5 |
| 3-3 程序结构 | 3-13 |
| 3-3-1 任务 | 3-13 |
| 3-3-2 子程序 | 3-13 |
| 3-3-3 任务处理 | 3-13 |
| 3-3-4 DriveProgramming 启动 / 停止和任务操作 | 3-15 |
| 3-3-5 DriveProgramming 重新启动 | 3-17 |
| 3-3-6 跳闸时的任务操作 | 3-18 |

3-1 操作步骤

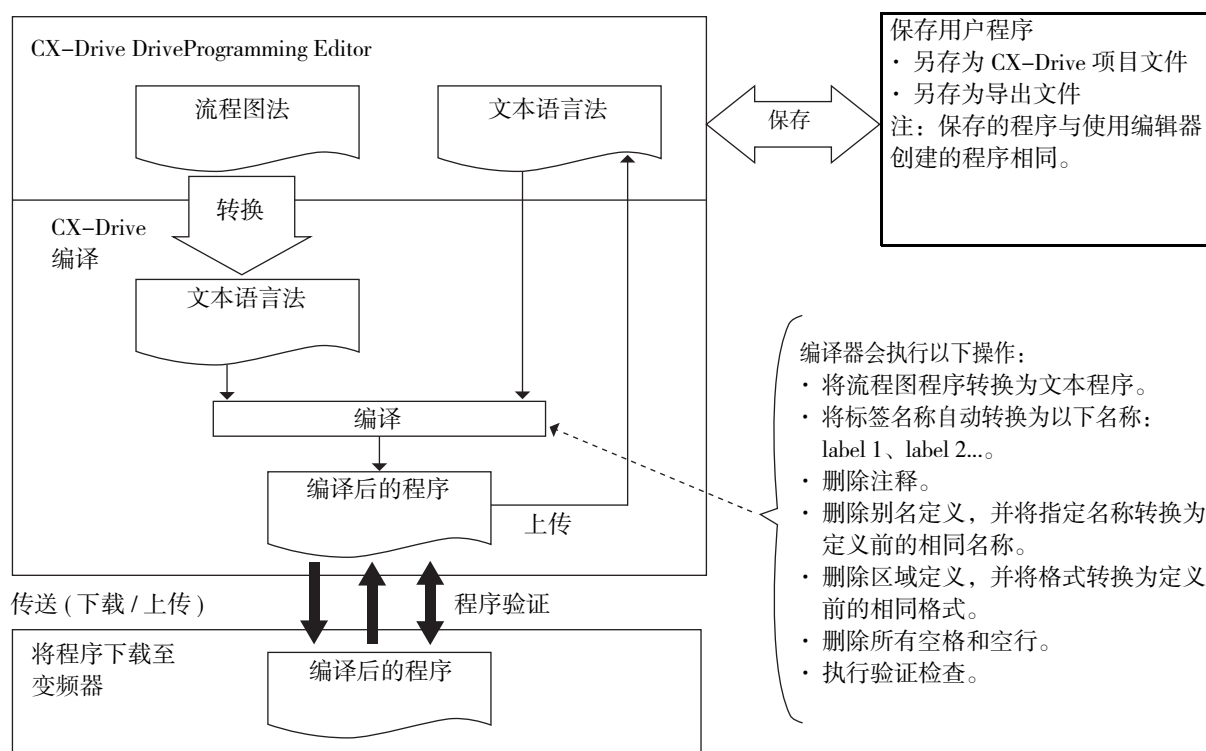
下图显示了使用 DriveProgramming 时从编程到执行程序的操作流程。

| 项目 | 参考 |
|------------------------|--------|
| 编程 | P. 3-3 |
| 编译程序 | P. 3-3 |
| 下载程序 | P. 3-4 |
| 选择 Driveprogramming 功能 | P. 3-4 |
| 启动程序 | P. 3-4 |

使用变频器 / 伺服支持工具 CX-Drive 中的 DriveProgramming Editor 创建 DriveProgramming 程序。

下图显示了从编程到传送 (下载) 至变频器的操作流程。

有关操作和其他信息的详细内容, 请参阅 “第 4 章节 DriveProgramming Editor”。



编程

可使用流程图或文本语言创建用户程序。也可在创建任务或子程序时选择其中一种方法。

使用 CX-Drive 中的 DriveProgramming Editor 输入用户程序。

DriveProgramming Editor 由 DriveProgramming 区域、工具箱窗口、块参数窗口、属性窗口和输出窗口构成。

- 在 DriveProgramming 区域，可以使用流程图或文本语言创建程序。
- 工具箱窗口按类别显示命令块。
- 在块参数窗口中，可设置开始执行程序时使用的参数。
- 在属性窗口中，可编辑在流程图中所选块的属性。
- 输出窗口会在编译结束后显示编译错误和警告。

编译程序

在 DriveProgramming 区域中创建的程序被编译并转换为最终的“编译后程序”。接着，该程序被传送（下载）至变频器。

编译器会检查程序验证、程序语法、参数输入限制和最大步数等项目。如果被禁止输入的内容，编译会停止，并显示错误消息。

编译器还会执行下述操作，并创建最终的“编译后程序”。因此，如果将曾在变频器中保存过的程序传送（上传）到 CX-Drive，在读取该程序时会显示“编译后程序”。尽管操作与编译前相同，但格式和内容都存在一些区别。

- 将流程图程序转换为文本程序。
- 将程序中指定的标签名称自动转换为 label 1、label 2...
- 删除程序中输入的注释。
- 删除别名定义，并将指定名称转换为定义前的相同名称。
- 删除区域定义，并将格式转换为定义前的相同格式。
- 删除程序中的所有空格和空行。
- 执行验证检查。



正确使用注意事项

- 为转换程序或将程序传送（下载）至变频器，执行编译时指定的注释、别名定义和区域定义都会被删除。

若要保存这些内容，请先保存程序，然后再执行程序转换或将程序传送（下载）至变频器。

可通过保存 CX-Drive 中的整个项目或导出程序文件的方式保存程序。

- 程序验证是指比较“编译后程序”与变频器内部的程序。因此，注释、别名定义、区域定义等内容都不会进行验证。

下载程序

将编译好的程序下载至变频器，并将其保存至变频器中的 EEPROM。
无需使用工具 (CX-Drive)，接通电源后便可启动 EEPROM 中保存的程序。

选择 Driveprogramming 功能

将 DriveProgramming 功能选择 (A017) 设为 “启用 (01 或 02)”，以便使用 DriveProgramming 功能。即便在执行操作期间，也可以更改 DriveProgramming 功能选择 (A017)。

| 参数编号 | 功能名称 | 数据 | 描述 |
|------|-----------------------|-----------------------------------|---|
| A017 | DriveProgramming 功能选择 | 00: 禁用 (默认设置) | 禁用 DriveProgramming 功能。 将不执行程序。 如果在程序执行期间将设置更改为 00(禁用)，程序会停止。 |
| | | 01: 启用 (通过多功能输入 PRG 端子启动 / 停止) | 在被设置为 82(PRG) 的多功能输入端子 *1 变为 ON 时启动 DriveProgramming 程序。 |
| | | 02: 启用 (接通 / 关闭电源时启动 / 停止) | 接通变频器电源后自动启动 DriveProgramming 程序。 如果在程序停止期间将设置更改为 02(启用)，程序将会启动。 |

*1 MX2 的多功能输入端子为 S1 到 S7，RX 的多功能输入端子为 S1 到 S8。

启动程序

DriveProgramming 功能选择 (A017) 设为 01(启用: 通过多功能输入 PRG 端子启动 / 停止) 时，请将多功能输入 S1 到 S7/S8 选择之一设为 82(PRG)。此时程序会在设为 PRG 的多功能输入端子变为 ON 时启动。PRG 端子变为 ON 期间，程序会一直执行，直到该端子变为 OFF 时才停止。

DriveProgramming 功能选择 (A017) 设为 02 时，程序会在设置完成后立即启动。程序还会在下次接通电源时自动启动。

除非重新启动程序，否则在程序启动并完成一系列进程后，便会执行 “end” 命令，结束运行。

若要重复运行程序，请创建一个循环程序，让程序在执行 “end” 命令之前执行循环程序。

下载的 DriveProgramming 程序保存在变频器的 EEPROM 中。因此，无需使用支持工具便可在下载后启动程序。

| 参数编号 | 功能名称 | 数据 | 描述 |
|-------------|-------------------------------|----------------------------------|---|
| C001 到 C007 | MX2 和 RX: 多功能输入 S1 到 S7 选择 | 82: PRG (DriveProgramming 启动) | DriveProgramming 功能选择 (A017) 设为 01(启用: 通过多功能输入 PRG 端子启动 / 停止) 时，程序会通过具备此设置的多功能输入端子启动。 |
| C008 | 仅限 RX: 多功能输入 S8 选择 | | |

3-2 与 DriveProgramming 有关的参数

本章节介绍与 DriveProgramming 有关的变频器参数。

选择 Driveprogramming 功能

将 DriveProgramming 功能选择 (A017) 设为启用 (01 或 02), 以便使用 DriveProgramming 功能。即便在执行操作期间, 也可以更改 DriveProgramming 功能选择 (A017)。

| 参数编号 | 功能名称 | 数据 | 描述 |
|-------------|-------------------------------|--------------------------------|---|
| A017 | DriveProgramming 功能选择 | 00: 禁用 (默认设置) | 禁用 DriveProgramming 功能。 将不执行程序。 如果在程序执行期间将设置更改为 00(禁用), 程序会停止。 |
| | | 01: 启用 (通过多功能输入 PRG 端子启动 / 停止) | 在被设置为 82(PRG) 的多功能输入端子 ^{*1} 变为 ON 时启动 DriveProgramming 程序。 |
| | | 02: 启用 (接通 / 关闭电源时启动 / 停止) | 接通变频器电源后自动启动 DriveProgramming 程序。 如果在程序停止期间将设置更改为 02(启用), 程序将会启动。 |
| C001 到 C007 | MX2 和 RX: 多功能输入 S1 到 S7 选择 | 82: PRG (DriveProgramming 启动) | DriveProgramming 功能选择 (A017) 设为 01 (启用: 通过多功能输入 PRG 端子启动 / 停止) 时, 程序会通过具备此设置的多功能输入端子启动。 |
| C008 | 仅限 RX: 多功能输入 S8 选择 | | |

*1 MX2 的多功能输入端子为 S1 到 S7, RX 的多功能输入端子为 S1 到 S8。

初始化程序

若要初始化下载至变频器的 DriveProgramming 程序, 请在“初始化选择 (b084)”中选择 04(清除故障监视器 + 初始化数据 + 清除 DriveProgramming), 并通过“初始化执行 (b180)”命令执行初始化。

| 参数编号 | 功能名称 | 数据 | 描述 |
|------|-------|---|-----------------------------------|
| b084 | 初始化选择 | 04: 清除故障监视器 + 初始化数据 + 清除 DriveProgramming | 同时选择 04 来初始化 DriveProgramming 程序。 |
| b180 | 初始化执行 | 01: 执行初始化 | 对所选数据执行初始化。 |

RUN 命令选择设置

使用 FW(正向)和 RV(反向)变量来控制 DriveProgramming 程序中的 RUN 命令。使用 FW 或 RV 变量时,请务必将 RUN 命令选择(A002/A202)设为 01(控制电路端子台)。

不论多功能输入选择中的设置为 00(FW:正向)(仅限 MX2)还是 01(RV:反向),您都能通过程序进行控制。以下项目之间的关系为 OR:

MX2: 程序的 RUN 命令和设为 00(FW:正向)的多功能输入端子。

RX: 程序的 RUN 命令和正向 RUN 命令端子 FW 与设为 01(RV:反向)的多功能输入端子。

MX2:

| 参数编号 | 功能名称 | 数据 | 描述 |
|----------------|------------------|--|---|
| A002/ A202 | 第一个/第二个 RUN 命令选择 | 01: 控制电路端子台 (DriveProgramming) 02: 数字操作器 03: Modbus 通信 04: 选项 1 05: 选项 2 | 选择 01(控制电路端子台)可使用 FW 或 RV 变量控制程序中的 RUN 命令。如果选择其他设置,则会禁用 FW 和 RV 变量。 |
| C001 到 C007 | 多功能输入 S1 到 S7 选择 | 00: FW(正向) 01: RV(反向) | 程序中的 RUN 命令与通过控制电路端子台发出的 RUN 命令之间的关系为 OR。 |

RX:

- 正向 RUN 命令: 正向 RUN 命令端子 FW(程序中的 RUN 命令与通过控制电路端子台发出的 RUN 命令之间的关系为 OR。)

| 参数编号 | 功能名称 | 数据 | 描述 |
|----------------|------------------|--|---|
| A002 | RUN 命令选择 | 01: 控制电路端子台 (DriveProgramming) 02: 数字操作器 03: Modbus 通信 04: 选项 1 05: 选项 2 | 选择 01(控制电路端子台)可使用 FW 或 RV 变量控制程序中的 RUN 命令。如果选择其他设置,则会禁用 FW 和 RV 变量。 |
| C001 到 C008 | 多功能输入 S1 到 S8 选择 | 01: RV(反向) | 程序中的 RUN 命令与通过控制电路端子台发出的 RUN 命令之间的关系为 OR。 |

设置变频器 I/O 功能

在 DriveProgramming 中，您可以将变频器的 I/O 功能 (多功能 I/O 和模拟 I/O) 用作程序的 I/O 功能。若要使用 I/O 功能，必须根据使用目的设置各个 I/O 功能。

本章节介绍如何为 DriveProgramming 设置 I/O 功能。

通过以下设置，可借助 DriveProgramming 的功能变量控制 I/O 功能。

将这些功能用作 DriveProgramming I/O 功能以外的用途时，请参阅“MX2-V1 系列多功能小型变频器用户手册” (样本编号：I585) 或“3G3RX-V1 系列高功能通用型变频器用户手册” (样本编号：I578)。

MX2:

| 参数编号 | 功能名称 | 数据 | 描述 |
|----------------|-----------------------|---------------------------------------|--|
| A001/ A201 | 第一个 / 第二个频率基准选择 | 07: DriveProgramming | 使用此设置时，可通过 DriveProgramming 的功能变量指定变频器频率基准。 如果选择 01(控制电路端子台) 以外的选项，模拟输入 (FV、FI) 会与频率基准断开，然后可将它们用作 DriveProgramming 的模拟输入 XA(00) 和 XA(01)。 |
| C001 到 C007 | 多功能输入 S1 到 S7 选择 | 56 到 62: MI1 到 MI7 (通用输入 1 到 7) | 设置 56 到 62(MI1 到 MI7)，将通用输入端子用于 DriveProgramming。 这些参数设置对应着功能变量 X(00) 到 X(06)。 |
| P003 | 脉冲串输入 RP 选择 | 02: MI8 (DriveProgramming 的通用输入 8) | 设置 02: MI8(DriveProgramming 的通用输入 8)，将通用输入 RP 端子用于 DriveProgramming。 RP 端子对应着功能变量 X(07)。 |
| C021 和 C022 | 多功能输出 P1 和 P2 选择 | 44 到 46: MO1 到 MO3 (通用输出 1 到 3) | 设置 44 到 46(MO1 到 MO3)，将通用输出端子用于 DriveProgramming。 您可通过将对应的功能变量 Y(00) 到 Y(02) 变更为 ON/OFF 来控制多功能输出端子。 |
| C026 | 多功能继电器输出 (MA、MB) 功能选择 | | |
| C027 | MP 选择 | 12: DriveProgramming (YA(0)) | 设置 12 (DriveProgramming)，将通用脉冲输出 YA(0) 端子用于 DriveProgramming。 |
| C028 | AM 选择 | 13: DriveProgramming (YA(1)) | 设置 13 (DriveProgramming)，将通用模拟输出 (电压)YA(1) 端子用于 DriveProgramming。 |
| P031 | 加速 / 减速时间输入类型 | 03: DriveProgramming | 设置此参数可通过 DriveProgramming 控制加速 / 减速时间。 |

**正确使用注意事项**

- 在多功能输入 S1 到 S7 选择被设置为 MI1 到 MI7(通用输入 1 到 7) 的情况下, 仍可将多功能输入 S1 到 S7 操作选择 (C011 到 C017) 设置为 NO(常开接点) 或 NC(常闭接点)。
- 在多功能输出 P1 和 P2 选择或多功能继电器输出 (MA、MB) 功能选择被设置为 MO1 到 MO3(通用输出 1 到 3) 的情况下, 仍可将多功能输出 P1 和 P2 操作选择 (C031、C032) 或多功能继电器输出 (MA、MB) 操作选择 (C036) 设置为 NO(常开接点) 或 NC(常闭接点)。
- 如果打开 3G3MX2-ZV1 系列变频器的安全功能选择开关来启用安全功能, 多功能输入 S3 选择 (C003) 和多功能输入 S4 选择 (C004) 会分别自动设为 77(GS1 输入) 以及 78(GS2 输入)。在此情况下, 多功能输入 S3/S4 操作选择 (C013/C014) 会固定为 NC(常闭接点), 而且无法变更。
若要为多功能输入 S3/S4 选择分配通用输入, 请关闭安全功能选择开关。
- 在 DriveProgramming 中, 模拟 I/O 功能会分配给 XA(0)、XA(1)、YA(0) 和 YA(1)。不论 A001、A201、C027 和 C028 的设置如何, 您都能使用这些功能变量监控程序中的模拟 I/O 状态。
- 在 DriveProgramming 程序中, 您无法监控通用 I/O 未设为 C001 到 C007、C021、C022 或 C026 的多功能 I/O 端子的状态。

RX:

| 参数编号 | 功能名称 | 数据 | 描述 |
|-------------|-----------------------|---------------------------------|--|
| A001 | 频率基准选择 | 07: DriveProgramming | <ul style="list-style-type: none"> · 使用此设置时, 可通过 DriveProgramming 的功能变量指定变频器频率基准。 · 如果选择 01(控制电路端子台) 以外的选项, 模拟输入 (FV、FI、FE) 会与频率基准断开, 然后可将它们用作 DriveProgramming 的模拟输入 XA(00) 到 XA(02)。 |
| C001 到 C008 | 多功能输入 S1 到 S8 选择 | 56 到 63: MI1 到 MI8 (通用输入 1 到 8) | <ul style="list-style-type: none"> · 设置 56 到 63(MI1 到 MI8), 将通用输入端子用于 DriveProgramming。 · 这些参数设置对应着功能变量 X(00) 到 X(07)。 |
| C021 到 C025 | 多功能输出 P1 到 P5 选择 | 44 到 49: MO1 到 MO6 (通用输出 1 到 6) | <ul style="list-style-type: none"> · 设置 44 到 49(MO1 到 MO6), 将通用输出端子用于 DriveProgramming。 · 您可通过将对应的功能变量 Y(00) 到 Y(05) 变更为 ON/OFF 来控制多功能输出端子。 |
| C026 | 多功能继电器输出 (MA、MB) 功能选择 | | |
| C027 | MP 选择 | 12: DriveProgramming (YA(0)) | 设置 12 (DriveProgramming), 将通用脉冲输出 YA(0) 端子用于 DriveProgramming。 |
| C028 | AM 选择 | 13: DriveProgramming (YA(1)) | 设置 13 (DriveProgramming), 将通用模拟输出 (电压)YA(1) 端子用于 DriveProgramming。 |
| C029 | AMI 选择 | 14: DriveProgramming (YA(2)) | 设置 14 (DriveProgramming), 将通用脉冲模拟输出 (电流)YA(2) 端子用于 DriveProgramming。 |
| P031 | 加速 / 减速时间输入类型 | 03: DriveProgramming | 设置此参数可通过 DriveProgramming 控制加速 / 减速时间。 |



正确使用注意事项

- 在多功能输入 S1 到 S8 选择被设置为 MI1 到 MI8(通用输入 1 到 8)的情况下, 仍可将多功能输入 S1 到 S8 操作选择 (C011 到 C018) 设置为 NO(常开接点) 或 NC(常闭接点)。
- 在多功能输出 P1 到 P5 选择或多功能继电器输出 (MA、MB) 功能选择被设置为 MO1 到 MO6(通用输出 1 到 6)的情况下, 仍可将多功能输出 P1 到 P5 操作选择 (C031 到 C035) 或为多功能继电器输出 (MA、MB) 操作选择 (C036) 设置为 NO(常开接点) 或 NC(常闭接点)。
- 如果启用了 3G3RX-V1 系列变频器的紧急关闭功能 (SW1 = ON), 多功能输入 S1 选择 (C001) 和多功能输入 S3 选择 (C003) 会分别自动设为 18(RS) 和 64(EMR)。在此情况下, 多功能输入 S3/S4 操作选择 (C011/C013) 会分别固定为 NO(常开接点) 和 NC(常闭接点), 而且无法变更。
若要为多功能输入 S1/S3 选择分配通用输入, 请关闭紧急关闭功能。
- 在 DriveProgramming 中, 模拟 I/O 功能会分配给 XA(0) 到 XA(2) 以及 YA(0) 到 YA(2)。不论 A001、C027 到 C029 的设置如何, 您都能使用这些变量监控程序中的模拟 I/O 状态。
- 在 DriveProgramming 程序中, 您无法监控通用 I/O 未设为- C001 到 C008 和 C021 到 C026 的多功能 I/O 端子的状态。

DriveProgramming 的监控功能

以下功能用来监控 DriveProgramming 的状态。



正确使用注意事项

- 如果数据超出四位，数据的前四位会显示在数字操作器上，如下文所示。
数字结尾的点表示小数点位置。您可以据此算出数字位数。
显示数据 0 到 9,999 时：0. 到 9999。
显示数据 10,000 到 65,535 时：1000 到 6553
显示数据 -1,230,000 到 1,230,000 时：-123 和 1230
- 由于数字操作器仅显示前四位，因此无法检查或设置数据的其他位数。
请使用 CX-Drive 检查或设置超过四位的数据。

| 参数编号 | 功能名称 | 数据 | 描述 |
|------|-------------------------------|-----------------------------------|---|
| d023 | 程序计数器 (DriveProgramming) | 0 到 1,024 | 执行 DriveProgramming 期间显示程序行数。 显示的数字为任务 1 的行数 (DriveProgramming Editor 中最左侧选项卡)。 · 显示 “编译后程序” 的行数。 · 同时显示正在执行的子程序的行数。 注：程序停止执行后会显示 “0000”。 |
| d024 | 程序编号监视器 (DriveProgramming) | 1 | 始终显示 “1”。 |
| d025 | 用户监视器 0 (DriveProgramming) | -2,147,483,647 到 2,147,483,647 | 此监视器功能对应于功能变量 UMon(0)，可显示程序中的任意数据。 |
| d026 | 用户监视器 1 (DriveProgramming) | -2,147,483,647 到 2,147,483,647 | 此监视器功能对应于功能变量 UMon(1)，可显示程序中的任意数据。 |
| d027 | 用户监视器 2 (DriveProgramming) | -2,147,483,647 到 2,147,483,647 | 此监视器功能对应于功能变量 UMon(2)，可显示程序中的任意数据。 |

DriveProgramming 的用户参数

最多提供 32 个参数作为 DriveProgramming 的用户参数。
这些参数的用途各不相同，比如程序初始数据设置、调整参数以及保存计算结果。



正确使用注意事项

- 如果数据超出四位，数据的前四位会显示在数字操作器上，如下文所示。
数字结尾的点表示小数点位置。您可以据此算出数字位数。
显示数据 0 到 9,999 时：0. 到 9999.
显示数据 10,000 到 65,535 时：1000 到 6553
显示数据 -1,230,000 到 1,230,000 时：-123 和 1230
- 由于数字操作器仅显示前四位，因此无法检查或设置数据的其他位数。
请使用 CX-Drive 检查或设置超过四位的数据。

| 参数编号 | 功能名称 | 数据 | 描述 |
|------|------------------------------|------------|---|
| P100 | DriveProgramming 用户参数 U00 | 0 到 65,535 | <ul style="list-style-type: none"> 这些用户参数对应着功能变量 U(00) 到 U(28)。 您可以使用数字操作器更改数据。更改后的数据会保存到 EEPROM 中。 在 DriveProgramming 中执行写入到 EEPROM 命令时，设置的数据会保存到 EEPROM 中。 |
| P101 | DriveProgramming 用户参数 U01 | | |
| P102 | DriveProgramming 用户参数 U02 | | |
| P103 | DriveProgramming 用户参数 U03 | | |
| P104 | DriveProgramming 用户参数 U04 | | |
| P105 | DriveProgramming 用户参数 U05 | | |
| P106 | DriveProgramming 用户参数 U06 | | |
| P107 | DriveProgramming 用户参数 U07 | | |
| P108 | DriveProgramming 用户参数 U08 | | |
| P109 | DriveProgramming 用户参数 U09 | | |
| P110 | DriveProgramming 用户参数 U10 | | |
| P111 | DriveProgramming 用户参数 U11 | | |
| P112 | DriveProgramming 用户参数 U12 | | |
| P113 | DriveProgramming 用户参数 U13 | | |

| 参数编号 | 功能名称 | 数据 | 描述 |
|------|------------------------------|------------|--|
| P114 | DriveProgramming 用户参数 U14 | 0 到 65,535 | <ul style="list-style-type: none"> • 这些用户参数对应着功能变量 U(00) 到 U(28)。 • 您可以使用数字操作器更改数据。更改后的数据会保存到 EEPROM 中。 • 在 DriveProgramming 中执行写入到 EEPROM 命令时，设置的数据会保存到 EEPROM 中。 |
| P115 | DriveProgramming 用户参数 U15 | | |
| P116 | DriveProgramming 用户参数 U16 | | |
| P117 | DriveProgramming 用户参数 U17 | | |
| P118 | DriveProgramming 用户参数 U18 | | |
| P119 | DriveProgramming 用户参数 U19 | | |
| P120 | DriveProgramming 用户参数 U20 | | |
| P121 | DriveProgramming 用户参数 U21 | | |
| P122 | DriveProgramming 用户参数 U22 | | |
| P123 | DriveProgramming 用户参数 U23 | | |
| P124 | DriveProgramming 用户参数 U24 | | |
| P125 | DriveProgramming 用户参数 U25 | | |
| P126 | DriveProgramming 用户参数 U26 | | |
| P127 | DriveProgramming 用户参数 U27 | | |
| P128 | DriveProgramming 用户参数 U28 | | |
| P129 | DriveProgramming 用户参数 U29 | 0 到 65,535 | <ul style="list-style-type: none"> • 这些用户参数对应着功能变量 U(29) 到 U(31)。 • 您可以使用数字操作器更改数据。更改后的数据会保存到 EEPROM 中。 • 在 DriveProgramming 中执行写入到 EEPROM 命令时，设置的数据会保存到 EEPROM 中。 • 变频器断电时，断电时设置的数据会保存到 EEPROM 中。 |
| P130 | DriveProgramming 用户参数 U30 | | |
| P131 | DriveProgramming 用户参数 U31 | | |

3-3 程序结构

在 3G3MX2-ZV1/3G3RX-V1 系列变频器的 DriveProgramming 中，您最多可以创建五个任务。
创建的任务会并行处理。

通过将一个应用程序分为多个进程，并将这些进程分配给多个任务，您可以为每个进程调整执行条件、
执行顺序等。此外，并行任务处理还可缩短处理时间。

3-3-1 任务

任务是指在 DriveProgramming 中执行的一小段程序。

对于 3G3MX2-ZV1/3G3RX-V1 系列变频器来说，您创建的程序最多可以包括五个任务。

- 所有任务都会同时启动。
- 按照任务 1 到任务 5 的顺序，每个任务中的一个命令（一行“编译后程序”）会在 2ms 处理时间内执行。
- 如用户参数等所有功能变量都在各任务之间共享。若要在各任务之间传送信息，可使用用户参数、内部用户接点等方式。
- 在 2ms 处理时间结束后，每个任务的操作结果会反映至变频器操作、外部输出等。与此同时，变频器和外部输入端子等的状态会变为读取。
- 执行“end”命令时，任务即告完成，并等待下次启动。

3-3-2 子程序

子程序是一种单独的程序处理，它仅在调用时执行。

子程序十分有用，可用于将程序组织成各个部分，从而可在同一个任务中执行多次，或在其他程序中重复使用。

在 DriveProgramming 中，您可以为每个任务插入子程序。不过，子程序不能在任务之间共享。

您可以在一个子程序内调用另一个子程序（嵌套）。

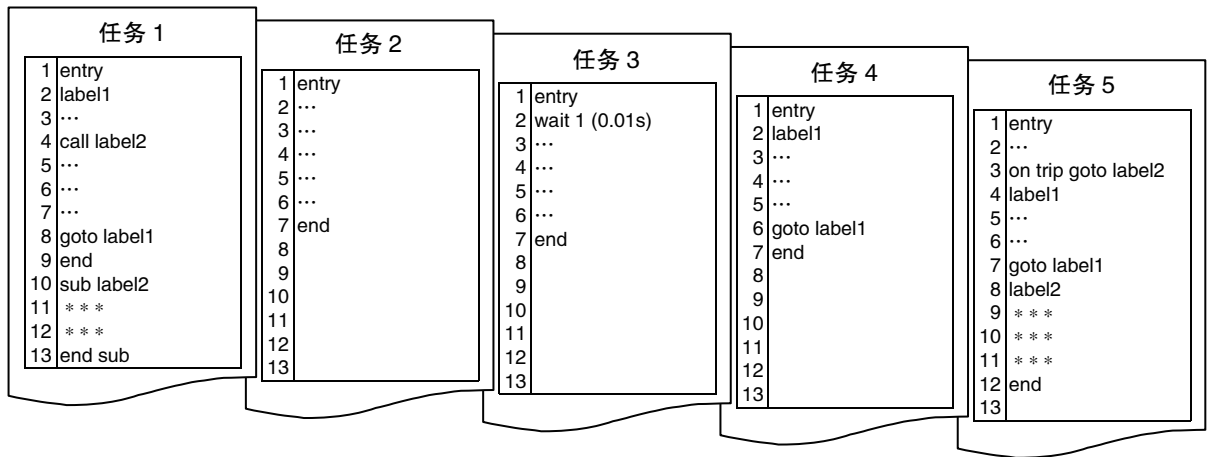
不过，子程序最多只能嵌套 8 层。

3-3-3 任务处理

最多可以同时启动五个任务，而且每个任务的一个命令（“编译后程序”的一行）会在 2ms 处理时间内执行。下面的“编译后程序（示例）”插图显示了程序处理流程。

- 所有任务都会同时启动。
- 按照任务 1 到任务 5 的顺序，每个任务中的一个命令（一行“编译后程序”）会在 2ms 处理时间内执行。
- 如用户参数等所有功能变量都在各任务之间共享。
- 在 2ms 处理时间结束后，每个任务的操作结果会反映至变频器操作、外部输出等。与此同时，变频器和外部输入端子等的状态会变为读取。
- 如任务 1 所示，如果通过调用命令调用了子程序，子程序语句会在下一步处理时执行。
- 如任务 2 所示，如果执行了结束命令，任务会完成，并等待下次启动。
- 如任务 3 所示，如果执行了等待命令，任务会反复执行，直至符合特定条件。
- 如任务 4 所示，如果使用 goto 命令创建了循环，任务进程会一直重复下去。
- 如任务 5 所示，如果执行了 on trip goto 命令，处理数据会存储在变频器中。
程序会在跳闸后立即跳转到 goto 目标。

● 编译后程序 (示例)



● 程序处理流程

| 处理时间 [ms] | 任务 1 | | 任务 2 | | 任务 3 | | 任务 4 | | 任务 5 | |
|-----------|------|-------------|------|-------|------|----------------|------|-------------|------|------------------------|
| | 行号 | | 行号 | | 行号 | | 行号 | | 行号 | |
| 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 [Trip occurred] |
| 10 | 10 | sub label2 | 5 | --- | 2 | wait 1 (0.01s) | 5 | --- | 5 | label2 |
| 12 | 11 | *** | 6 | --- | 2 | wait 1 (0.01s) | 6 | goto label1 | 6 | *** |
| 14 | 12 | *** | 7 | end | 3 | --- | 2 | label1 | 7 | *** |
| 16 | 13 | end sub | | | 4 | --- | 3 | --- | 2 | *** |
| 18 | 5 | --- | | | 5 | --- | 4 | --- | 3 | end |
| 20 | 6 | --- | | | 6 | --- | 5 | --- | 8 | |
| 22 | 7 | --- | | | 7 | end | 6 | goto label1 | 9 | |
| 24 | 8 | goto label1 | | | | | 2 | label1 | 10 | |
| 26 | 2 | label1 | | | | | 3 | --- | 11 | |
| 28 | 3 | --- | | | | | 4 | --- | 12 | |
| 30 | 4 | call label2 | | | | | 5 | --- | | |
| 32 | 10 | sub label2 | | | | | 6 | goto label1 | | |
| 34 | 11 | *** | | | | | 2 | label1 | | |
| 36 | 12 | *** | | | | | 3 | --- | | |
| 38 | 13 | end sub | | | | | 4 | --- | | |

由于发生了变频器跳闸，因此执行 Label2。

程序计数器 (d023) 显示

3-3-4 DriveProgramming 启动 / 停止和任务操作

您可以在 DriveProgramming 功能选择 (A017) 中设置 DriveProgramming 程序的启动 / 停止方法。

● 启动

- 将 DriveProgramming 功能选择 (A017) 设为 01(启用: 通过多功能输入 PRG 端子启动 / 停止) 时: 程序会在设为 PRG 的多功能输入端子变为 ON 时启动。
- 将 DriveProgramming 功能选择 (A017) 设为 02(启用: 接通 / 关闭电源时启动 / 停止) 时: 程序会在变频器接通电源时启动。

● 停止

- 将 DriveProgramming 功能选择 (A017) 设为 01(启用: 通过多功能输入 PRG 端子启动 / 停止) 时: 程序会在设为 PRG 的多功能输入端子变为 OFF 时停止。
- 将 DriveProgramming 功能选择 (A017) 设为 02(启用: 接通 / 关闭电源时启动 / 停止) 时: 程序会在变频器关闭电源时停止。

此外, 不论 A017 设置如何, 如果因为程序错误而检测到 E43 到 E45 跳闸, 或者在所有任务中执行了结束命令时, DriveProgramming 程序都会停止。



安全使用注意事项

DriveProgramming 程序停止后, DriveProgramming 控制的多功能输出会保留程序停止之前的状态。因此需对系统进行相关配置, 确保变频器中 DriveProgramming 程序停止操作可被 DriveProgramming 启动信号和报警 (跳闸) 信号检测到, 而且变频器周边设备可安全停止。



正确使用注意事项

以下是 DriveProgramming 程序停止时功能变量的状态。请根据每种状态采取必要的措施。

- 对于多功能输出 (变频器实际输出端子), 它们会保留程序停止之前的状态。但是, 未设为 MO1 到 MO6 的输出端子 (DriveProgramming 的通用输出) 会作为变频器的正常多功能输出发挥作用。
- 为频率基准、加速 / 减速时间或模拟输出选择了 DriveProgramming 功能时, 这些功能在程序停止之前的设置值也会保留。
- 程序停止之前的用户参数变量、内部用户变量和内部用户接点数据会保留下来。
- 变频器实际输入端子的状态 (比如多功能和模拟输入端子) 不会保留下来, 而是会发生更新。
- 多功能输出变量 (RUN、FA1 和 AL 等功能位) 和变频器监视器变量数据不会保留下来, 而是始终根据变频器状态进行更新。
- 只有 DriveProgramming 的多功能输入变量 (FW、RV 和 CF1 等功能位) 以及定时器变量会在程序停止时清除, 而且所有数据都归零。

通过 CX-Drive 启动 / 停止 DriveProgramming 功能时, 变频器参数 DriveProgramming 功能选择 (A017) 的值只会在 RAM 数据中暂时更改为 02(启动) 或 00(停止)。

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

- 重新接通变频器电源, 然后将 A017 的值恢复为 EEPROM 中保存的值。
- 在接通变频器电源之前, 请勿执行以下 EEPROM 保存操作。
 - 从 CX-Drive 传送 (下载) 一部分参数。
 - 通过 Modbus 通信或通信选件发出 “enter” 命令。

DriveProgramming 程序启动时，五个任务会同时启动。

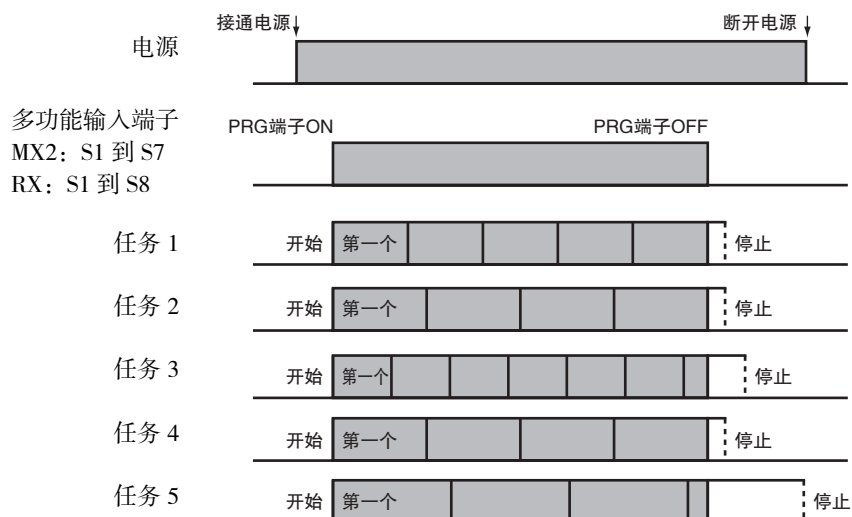
如果使用 “goto” 等命令在任务中创建了一个循环构造，任务会在启动后重复该循环。

DriveProgramming 程序停止时，五个任务会同时停止。

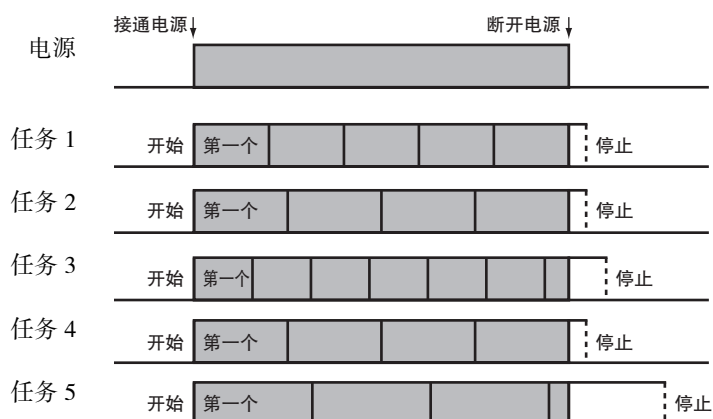
如果程序在电机运行期间停止，电机会根据停止选择 (b091)(减速停止或自由停止) 的设置停止。

不同设置对应的任务操作如下所示。

● DriveProgramming 功能选择 (A017) 设为 01(启用) 时：



● DriveProgramming 功能选择 (A017) 设为 02(启用) 时：



3-3-5 DriveProgramming 重新启动

您可以执行以下操作，重新启动已停止的程序。程序重新启动后，所有任务都会从头开始同时启动。

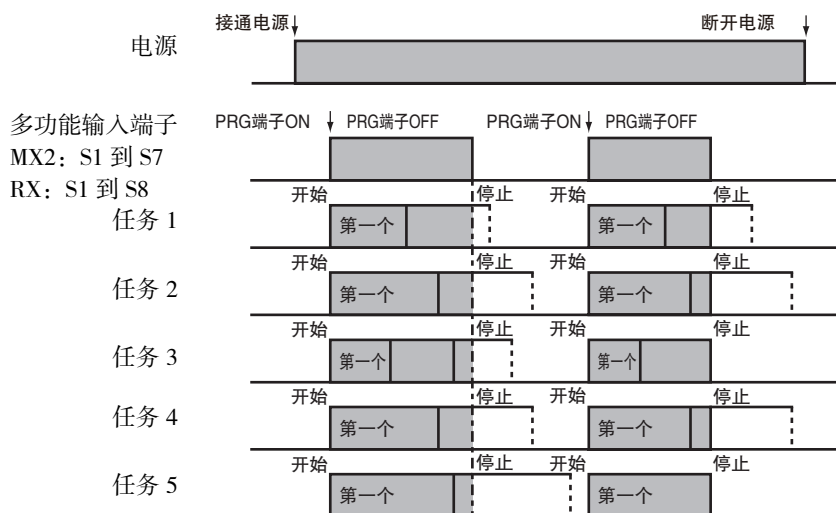
- DriveProgramming 功能选择 (A017) 设为 01(启用: 通过多功能输入 PRG 端子启动 / 停止) 时: 在 PRG 端子为 ON 期间通过控制电路端子重置输入, 或重新将 PRG 端子变为 ON。
- DriveProgramming 功能选择 (A017) 设为 02(启用: 接通 / 关闭电源时启动 / 停止) 时: 通过控制电路端子发出重置输入, 或重新接通变频器电源。



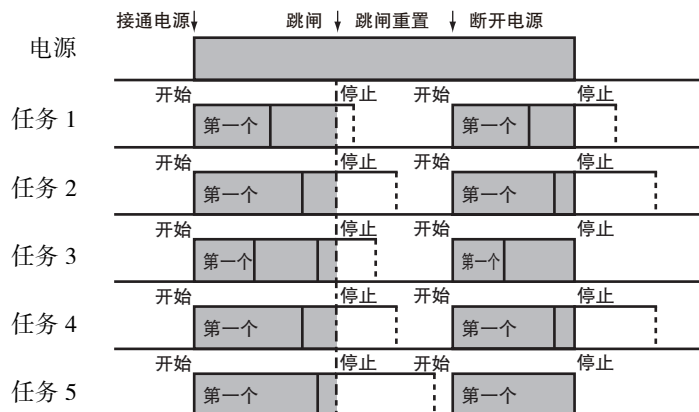
正确使用注意事项

- 按数字操作器或 LCD 数字操作器 (选件) 的停止 / 复位键无法重新启动 DriveProgramming。请将多功能输入端子 (MX2: S1 到 S7/RX: S1 到 S8) 之一设为 18(RS: 重置), 然后将该端子变为 ON。
- 若要通过重置输入重新启动 DriveProgramming, 请将重置选择 (C102) 设为 03(仅限跳闸重置) 以外的值。设置 03(仅限跳闸重置) 时, 您无法重新启动 DriveProgramming。
- 将重置选择 (C102) 设为 00(接通电源时跳闸重置) 或 01(关闭电源时跳闸重置) 时, 即便没有跳闸也能通过重置输入重新启动 DriveProgramming。

● DriveProgramming 功能选择 (A017) 设为 01(启用) 时重新启动:



● DriveProgramming 功能选择 (A017) 设为 02(启用) 时在跳闸后重新启动 (无需发出“on trip goto”命令):



3-3-6 跳闸时的任务操作

基本上，即便变频器在 DriveProgramming 操作期间检测到跳闸，操作也会继续执行。但是，如果检查到与 DriveProgramming 有关的 E43 到 E45 跳闸，操作会停止。

或者，如果遇到 “on trip goto” 命令，程序会在发生跳闸后跳至其他进程。

| 有 / 无 on trip goto | 出错状态 | | |
|-----------------------|---------------------------------------|--|---------------------------------------|
| | 用户跳闸 E50 到 E59 | 与 DriveProgramming 有关的 E43 到 E45 跳闸 | 其他跳闸 |
| 无 | 操作继续。 | 程序停止。 | 操作继续。 |
| 有 | 执行 “on trip goto” 命令后，程序会跳至指定标签并继续操作。 | 程序停止。 | 执行 “on trip goto” 命令后，程序会跳至指定标签并继续操作。 |

有关 E43 到 E59 跳闸的详细信息，请参阅第 8 章节 *错误与纠正措施*。

4

DriveProgramming Editor

本章节介绍如何启动 DriveProgramming Editor、保存和加载数据以及编辑器各部分详细信息。

4

| | |
|---|------|
| 4-1 启动 DriveProgramming Editor | 4-2 |
| 4-2 DriveProgramming Editor 的各部分 | 4-6 |
| 4-2-1 DriveProgramming Editor | 4-6 |
| 4-2-2 工具栏 | 4-6 |
| 4-2-3 DriveProgramming 区域 | 4-9 |
| 4-2-4 工具箱窗口 | 4-13 |
| 4-2-5 块参数窗口 | 4-14 |
| 4-2-6 属性窗口 | 4-15 |
| 4-2-7 输出窗口中的错误列表选项卡 | 4-16 |
| 4-3 添加、删除和重命名任务 | 4-17 |
| 4-4 插入、删除和调用子程序 | 4-18 |
| 4-5 创建流程图程序 | 4-19 |
| 4-6 创建文本程序 | 4-20 |
| 4-7 编辑传送 (上传) 的程序 | 4-21 |
| 4-8 保存程序 | 4-22 |
| 4-9 传送和验证程序 | 4-24 |
| 4-10 执行程序 (DriveProgramming 功能选择) | 4-25 |
| 4-11 其他有用功能 | 4-27 |

4-1 启动 DriveProgramming Editor

若要创建DriveProgramming程序，需使用变频器/伺服电机支持工具CX-Drive中的DriveProgramming Editor。本章节介绍如何配置 CX-Drive 中的 DriveProgramming Editor，以及启动 DriveProgramming Editor 之前的操作。

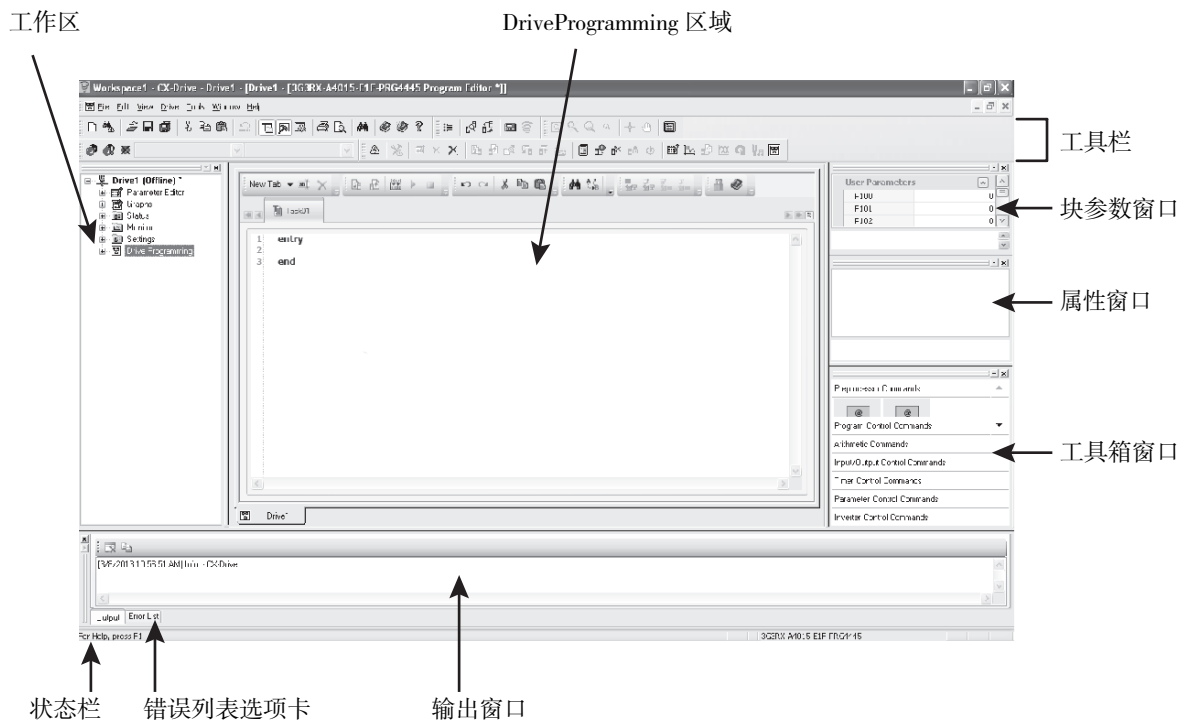


正确使用注意事项

- DriveProgramming 功能包含在以下版本或更高版本的 CX-Drive 中。
如果您的 CX-Drive 版本较低，需要升级到相应版本。
MX2: 2.8 版或更高版本 (2.7 版或更低版本无法执行操作。)
RX: 2.7 版或更高版本 (2.6 版或更低版本无法执行操作。)
- 首次启动 CX-Drive 中的 DriveProgramming Editor 时需要使用密码。
请联系当地欧姆龙代理商。

CX-Drive 和 DriveProgramming Editor 画面布局

DriveProgramming Editor是变频器/伺服电机支持工具CX-Drive所含的多项功能之一。启动DriveProgramming Editor时，CX-Drive 的画面布局会出现如下变化。



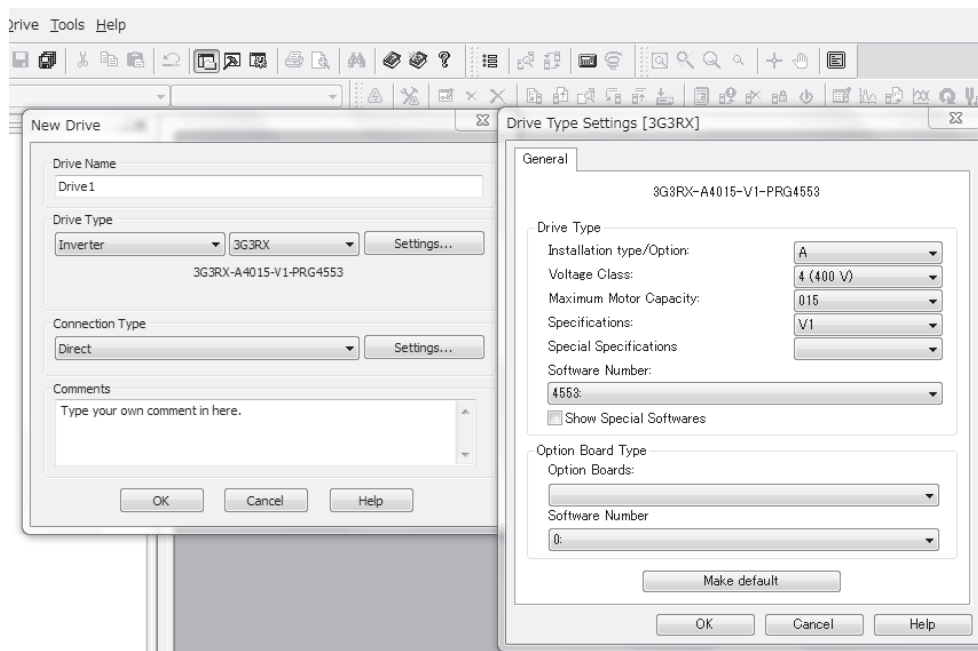
启动 CX-Drive

使用以下方法启动CX-Drive：从Windows [开始]菜单选择[所有程序] - [OMRON] - [CX-One] - [CX-Drive]。单击 CX-Drive 文件夹中的 CX-Drive 图标。

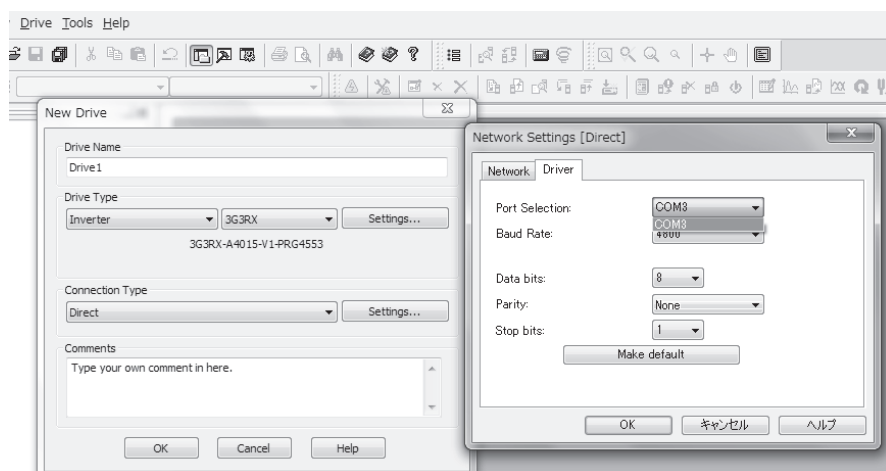
启动 CX-Drive 后，从 CX-Drive 菜单中选择 [File] (文件)，然后单击 [New] (新建) 创建一个新的 CX-Drive 文件。此时会出现 [New Drive] (新建驱动器) 对话框。

从 [Drive Type] (驱动器类型) 下的下拉列表中选择 3G3MX2 或 3G3RX 系列。单击右侧的 [Settings] (设置) 按钮。在 [Drive Type Settings] (驱动器类型设置) 对话框中，设置 [Installation Type/Option] (安装类型/选项)、[Voltage Class] (电压等级) 和 [Maximum Motor Capacity] (最大电机容量)。

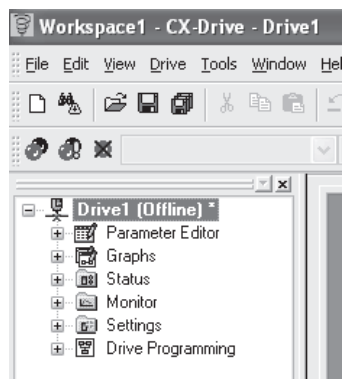
请务必从 [Specifications] (规格) 下拉列表中选择 [V1]，以便选择 3G3MX2-ZV1/3G3RX-V1 系列变频器。设置这些项目后，单击 [OK] (确定) 按钮关闭 [Drive Type Settings] (驱动器类型设置) 对话框。



在 [New Drive](新建驱动器) 对话框中, 为 CX-Drive 和变频器设置连接类型。
在 [Connection Type](连接类型) 下方, 选择 [Direct](直接连接), 然后单击右侧的 [Settings](设置) 按钮。
在 [Driver](驱动器) 选项卡页面上, 将 [Port Selection](端口选择) 设为安装 CX-Drive 的计算机的端口名称。




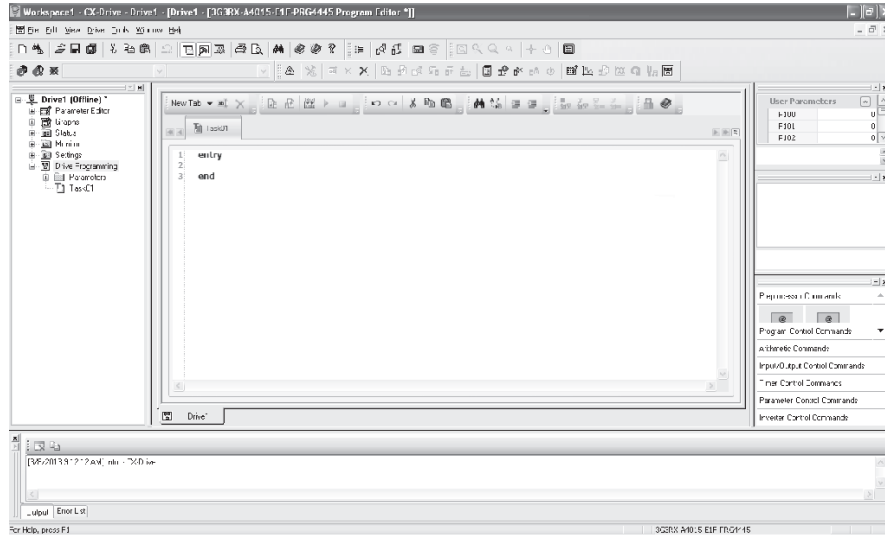
设置这些项目后, 单击 [OK](确定) 按钮关闭所有对话框。此时便会将新的项目注册到工作区中。



启动 DriveProgramming Editor

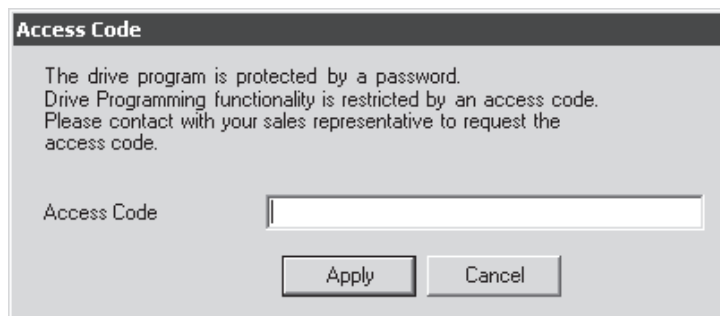
显示 DriveProgramming Editor 的方法有三种：

- 双击工作区中的 [DriveProgramming]。
- 单击 CX-Drive 工具栏上的  按钮。
- 从 [Drive](驱动器) 菜单中选择 [Program](程序) – [Program Editor](程序编辑器)。



正确使用注意事项

启动 CX-Drive 后首次启动 DriveProgramming Editor 时需要使用密码。
请联系销售代表索取密码，然后将其输入以下对话框。

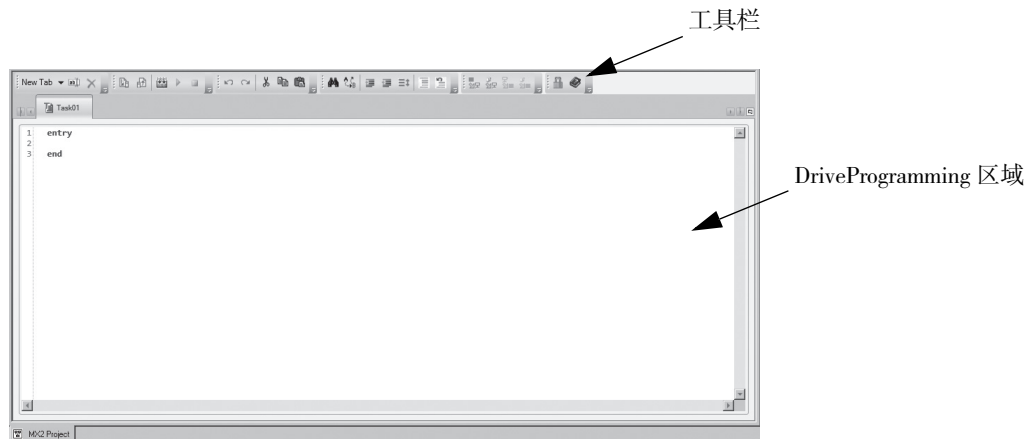


4-2 DriveProgramming Editor 的各部分

本章节介绍 CX-Drive 中与 DriveProgramming 有关的编辑器和窗口的各项功能的详细信息。

4-2-1 DriveProgramming Editor

DriveProgramming Editor 是 DriveProgramming 功能的主窗口。



此窗口包括含有常用命令的工具栏，以及将程序显示为文本或流程图的 DriveProgramming 区域。

4-2-2 工具栏

DriveProgramming Editor 提供以下命令：


● 常用命令

| 命令 | 图标 | 说明 |
|-------------|----|--|
| 新建任务 (流程图) | | 为程序创建新的流程图任务。任务是指在 DriveProgramming 中执行的一小段程序。 |
| 新建任务 (文本) | | 为程序创建新的文本任务。任务是指在 DriveProgramming 中执行的一小段程序。 |
| 新建子程序 (流程图) | | 创建新的流程图子程序。子程序是程序的一部分，仅在调用时执行。 |
| 新建子程序 (文本) | | 创建新的文本子程序。子程序是程序的一部分，仅在调用时执行。 |
| 重命名当前任务 | | 重命名当前任务或子程序。 |
| 删除当前任务 | | 删除当前任务或子程序。 |
| 撤消 | | 撤消最近一次更改。 |
| 恢复 | | 恢复撤消的操作。 |
| 传送到驱动器 | | 编译程序后，如果没有错误，再将程序传送 (下载) 至驱动器 (变频器)。 |

| 命令 | 图标 | 说明 |
|--------|----|---|
| 从驱动器传送 | | 将程序从驱动器 (变频器) 传送 (上传) 到 DriveProgramming Editor。 |
| 启动 | | 启动变频器中的程序。CX-Drive 会首先将变频器中的程序与 DriveProgramming 区域中的程序加以比较。仅当两者相同时，程序才会启动。如果两者不同，程序不会启动。这种程序启动方法仅在 CX-Drive 连接时可用。若要使用独立型变频器启动程序，请务必设置 DriveProgramming 功能选择 (A017)。 |
| 停止 | | 停止变频器中的程序。执行此操作时不会受 CX-Drive 中 DriveProgramming 区域上程序的影响。 |
| 编译 | | 编译 DriveProgramming 区域中的程序。程序中出现的编译错误和警告会在输出窗口错误列表选项卡页面上显示。 |
| 设置密码 | | 允许设置、更改或删除程序密码。 |
| 帮助 | | 显示 CX-Drive 帮助。 |

● 流程图程序的命令

| 命令 | 图标 | 说明 |
|--------|----|------------------------------|
| 放大 | | 调高缩放水平。 |
| 缩小 | | 调低缩放水平。 |
| 缩放重置 | | 将缩放水平恢复为初始值。 |
| 选择模式 | | 通过用鼠标光标拖动的方式，选择程序中一个或多个块 *1。 |
| 平移模式 | | 通过拖动的方式移动任何方向的视场，同时保持缩放比例。 |
| 水平左对齐 | | 在水平方向朝左对齐当前所选的块 *1。 |
| 水平居中对齐 | | 在水平方向居中对齐当前所选的块 *1。 |
| 水平朝右对齐 | | 在水平方向朝右对齐当前所选的块 *1。 |
| 垂直顶部对齐 | | 在垂直方向顶部对齐当前所选的块 *1。 |
| 垂直居中对齐 | | 在垂直方向居中对齐当前所选的块 *1。 |
| 垂直底部对齐 | | 在垂直方向底部对齐当前所选的块 *1。 |
| 方位 | | 选择连接块的方位 *1。 |
| 自动排列 | | 按当前所选的方位自动排列流程图中的块 *1。 |
| 显示接点 | | 显示 / 隐藏块接点 *1。 |

| 命令 | 图标 | 说明 |
|----|---|--|
| 显示 |  | 允许从下列选项中选择程序的显示样式。 <ul style="list-style-type: none"> · 仅限文本 · 仅限图标 · 文本和图标 · 名称、图标和自变量 |

*1. 流程图上的显示单位称为“块”。块由命令或命令与附加到命令上的标签构成。

● 文本程序的命令

| 命令 | 图标 | 说明 |
|----------|---|----------------------------------|
| 查找 |  | 查找 DriveProgramming Editor 中的文本。 |
| 替换 |  | 替换 DriveProgramming Editor 中的文本。 |
| 增加缩进 |  | 增加所选文本的缩进。 |
| 减小缩进 |  | 减小所选文本的缩进。 |
| 格式化所选文本 |  | 自动格式化所选文本。 |
| 注释所选文本 |  | 将所选文本行转换为注释。 |
| 取消注释所选文本 |  | 取消注释所选文本行。 |

● 程序转换

可将流程图程序转换为文本程序，反之亦可。

转换时，程序会编译一次。请注意，如果出现编译错误，转换不会完成。

此外，转换期间执行编译时，注释、别名定义、区域定义等内容都会被删除。但是，程序只会发生部分更改，该操作不会改变程序的格式和内容。

| 命令 | 图标 | 说明 |
|-------------|---|----------------------|
| 将文本转换为流程图 |  | 将当前文本任务 / 子程序转换为流程图。 |
| 将整个程序转换为流程图 |  | 将整个程序转换为流程图。 |
| 将流程图转换为文本 |  | 将当前流程图任务 / 子程序转换为文本。 |
| 将整个程序转换为文本 |  | 将整个程序转换为文本。 |

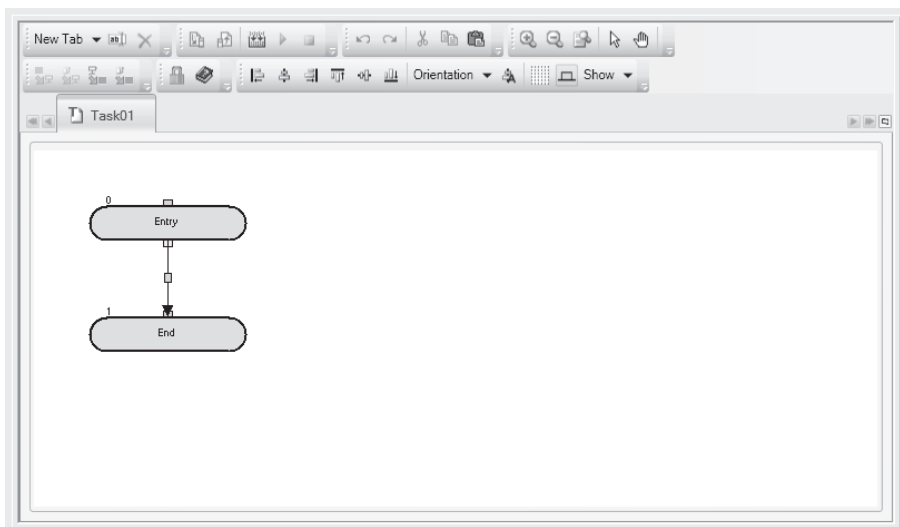
● 快捷键

在 DriveProgramming 区域中，可以看到以下快捷键。

| 快捷键 | 说明 |
|-----------|---------------|
| Ctrl + X | 剪切 |
| Ctrl + C | 复制 |
| Ctrl + V | 粘贴 |
| Ctrl + Z | 撤消 |
| Ctrl + Y | 恢复 |
| Ctrl + A | 全选 |
| Ctrl + F | 查找并替换 |
| Ctrl + 空格 | 指令一览表 |
| Tab | 选择下一个 (仅限流程图) |
| 箭头键 | 移动所选的块 |
| Home | 滚动到顶端 (仅限流程图) |
| End | 滚动到底端 (仅限流程图) |
| Page Up | 将光标移动到行首 |
| Page Down | 将光标移动到行末 |
| + | 放大 |
| - | 缩小 |

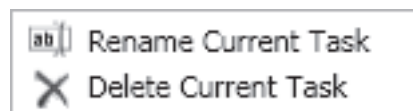
4-2-3 DriveProgramming 区域

DriveProgramming 区域会显示程序的当前设计。



此区域可能会含有多个不同页面，并组织到选项卡中。每个选项卡都是流程图或文本中的一个任务或子程序。

右键单击选项卡标题时，可删除或重命名任务或子程序。



流程图程序

使用流程图程序法时，每个显示单位被称为“块”。

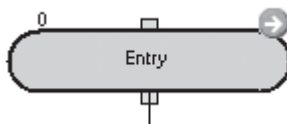
创建程序时，可在区域中放入多个块，然后在各个块之间设置交互。

在流程图程序中，如果程序编译成功，一个带白色箭头绿色圆圈的图标会突出显示每个流程图任务的起点。

此外，程序编译成功后，每个块的左上方会显示块编号。

块编号为从“0”开始的连续编号。整个程序中的每个块都会分配一个块编号。

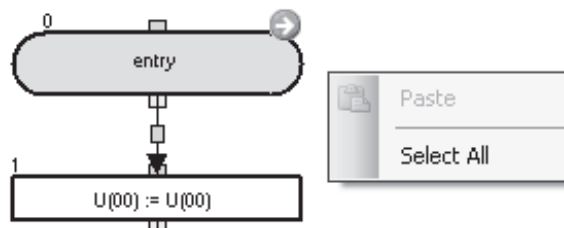
该编号与程序转换为文本后的行数并不一致。



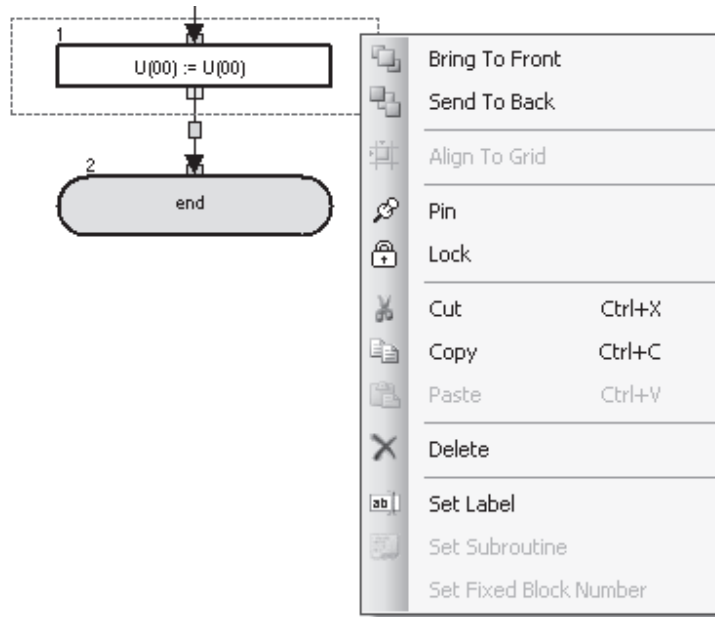
输出窗口会指示程序是否编译成功。如果编译时出错，带惊叹号的红色图标会标识流程图程序中出错的块。将鼠标放在错误图标上会显示编译错误，也可以在输出窗口中的错误列表中查看错误。



如果在流程图中的空白区域单击右键，则会显示一个弹出菜单。使用该菜单可以粘贴上次复制的块，或选择全部块。



如果在流程图块上单击右键，则会显示含有更多选项的弹出菜单。



下表显示了流程图程序中可用的菜单命令。

| 命令 | 说明 |
|-------------------------------|---|
| [Bring To Front](置于前端) | 在图中将所选块放在其他块的前端。 |
| [Send To Back] (发送至后端) | 在图中将所选块放在其他块的后端。 |
| [Pin](固定) | 将所选块固定在图中的当前位置。此时使用拖放操作无法移动块。 |
| [Lock](锁定) | 作用与 [Pin](固定) 类似，除此之外，此操作还会将块属性设为只读。 |
| [Cut](剪切) | 删除所选块，并将其保存到剪贴板中，以备稍后粘贴。 |
| [Copy](复制) | 将所选块保存到剪贴板中，以备稍后粘贴。 |
| [Paste](粘贴) | 将剪贴板中之前复制的内容放在设计区域。请注意，您也可以将它们作为图像粘贴到其他应用程序中。 |
| [Set Label](设置标签) | 为所选的块设置标签名称。 |
| [Set Subroutine] (设置子程序) | 为所选的块设置子程序名称。此菜单命令仅适用于可设置子程序的块。 |

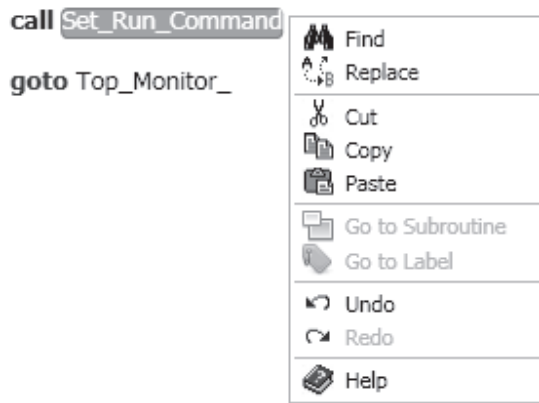
文本程序

使用文本程序法时，需要使用文本语言创建程序。

文本程序如果编译不成功，相关程序错误会显示在输出窗口中。出错的语句所在的行会呈红色突出显示。

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

右键单击所选的文本会显示弹出菜单。



下表显示了文本程序中可用的菜单命令。

| 命令 | 说明 |
|-------------------------------|---|
| [Find](查找) | 查找程序代码中的选定文本。 |
| [Replace](替换) | 替换程序代码中的选定文本。 |
| [Cut](剪切) | 删除所选文本，并将其保存到剪贴板中，以备稍后粘贴。 |
| [Copy](复制) | 将所选文本保存到剪贴板中，以备稍后粘贴。 |
| [Paste](粘贴) | 将剪贴板中之前复制的内容放在设计区域。请注意，您也可以将它们作为图像粘贴到其他应用程序中。 |
| [Go to Subroutine] (转至子程序) | 跳至选定的文本子程序。 |
| [Go to Label](转至标签) | 跳至选定的文本标签。 |
| [Undo](撤消) | 撤消最近一次更改。 |
| [Redo](恢复) | 恢复撤消的操作。 |
| [Help](帮助) | 显示 CX-Drive 帮助。 |

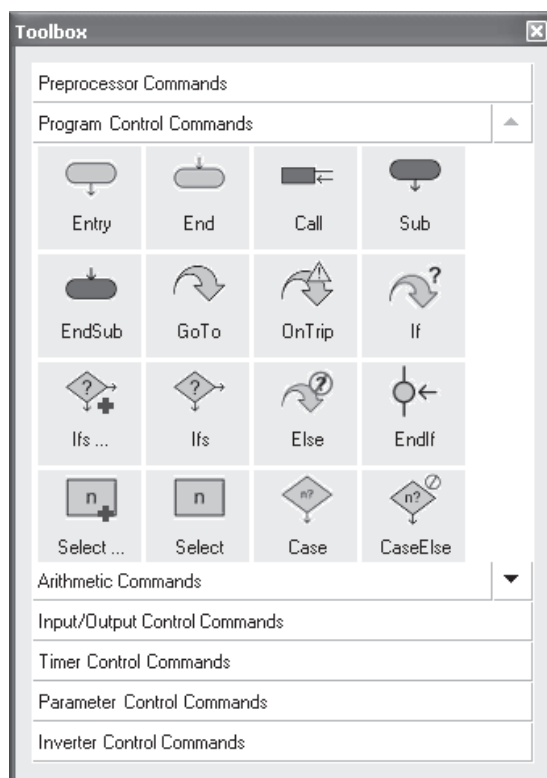
4-2-4 工具箱窗口

借助工具箱窗口，您可以通过拖放操作，将块添加到 DriveProgramming 区域。它会显示特定命令支持的块，并按类别加以组织。

工具箱会在启动 DriveProgramming Editor 时显示。此外，您还可以在 [View](视图) 菜单中单击 [DriveProgramming] - [Toolbox](工具箱) 菜单来显示或隐藏工具箱。

默认情况下，工具箱窗口显示在 CX-Drive 右侧。

您可以双击标题栏 (窗口上较宽的框架)，将该窗口分离开来。

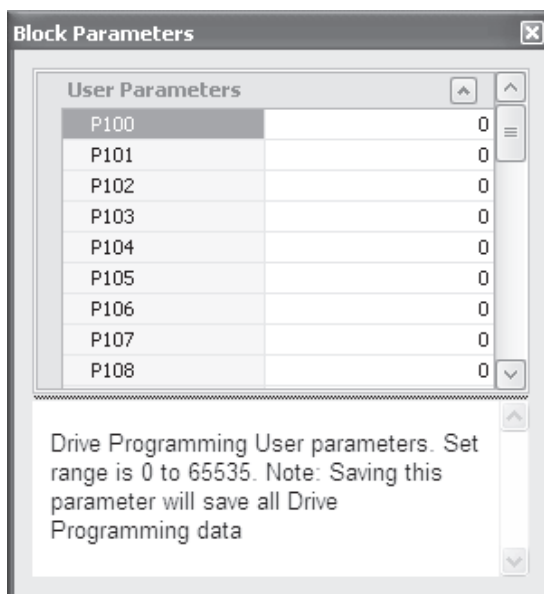


此外，您也可以用鼠标右键单击工具箱，选择它的显示样式。可用的样式有三种，分别是大图标、小图标和列表。无论使用哪种样式，只要将鼠标光标放在块上，都会显示简短的帮助文本。

单击类别标题可显示属于该类别的块。

4-2-5 块参数窗口

使用块参数窗口可以编辑用作程序变量的 DriveProgramming 用户参数。显示的参数会按类别加以组织。块参数会在启动 DriveProgramming Editor 时显示。此外，您还可以在 [View](视图) 菜单中单击 [DriveProgramming] – [Block Parameters](块参数) 菜单来显示或隐藏块参数。默认情况下，块参数窗口显示在 CX-Drive 右侧。您可以双击标题栏 (窗口上较宽的框架)，将该窗口分离开来。



若要更改参数值，请将光标放在参数所在的行，然后单击参数编号右侧的编辑框，接着输入新的值。如果输入的值超出有效范围，则会显示警告。窗口下半部分会显示用户参数的帮助文本。

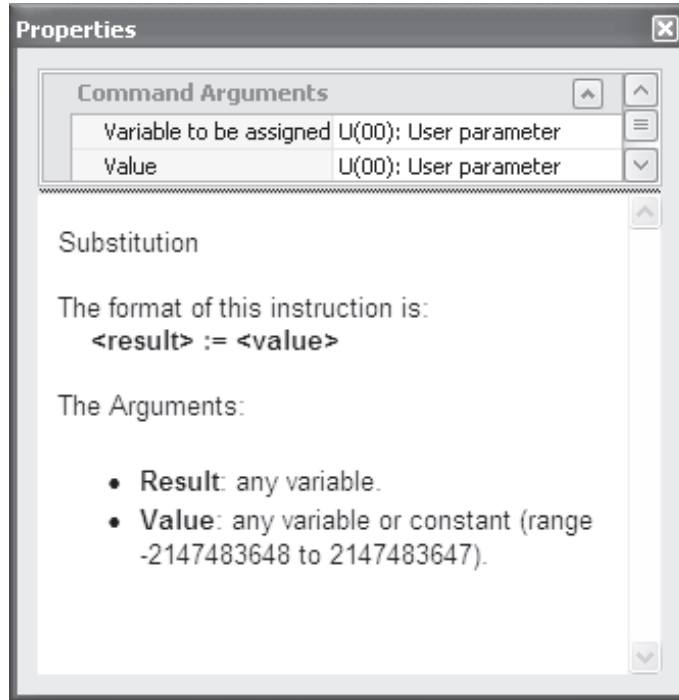
4-2-6 属性窗口

使用属性窗口可以编辑流程图程序中当前所选的块的属性。



属性会在启动 DriveProgramming Editor 时显示。此外，您也可以在 [View](视图) 菜单中单击 [DriveProgramming] - [Properties](属性) 菜单来显示或隐藏工具箱。

默认情况下，属性窗口显示在 CX-Drive 右侧。

您可以双击标题栏 (窗口上较宽的框架)，将该窗口分离开来。

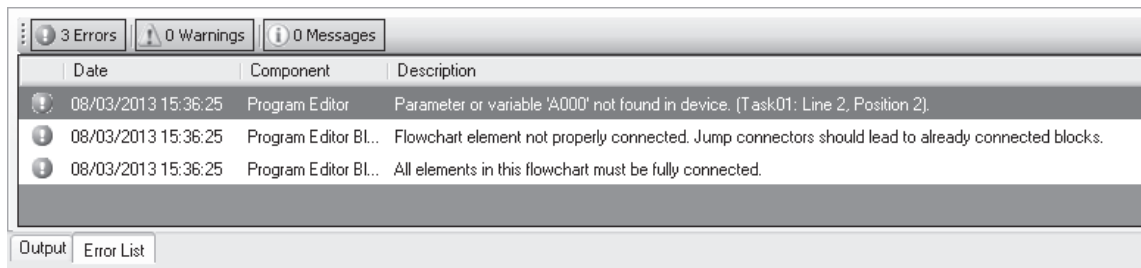


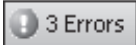


若要更改一个块命令自变量，请在流程图中选择块，然后将光标放在属性窗口中要编辑的部分。

- 如果块自变量含有选项，请单击当前值显示最右侧的  图标。单击  或双击当前值会在下拉菜单中显示可用选项。
- 如果块自变量不含选项，请单击当前值，然后输入新的值来进行更改。如果输入的值超出有效范围，则会显示警告。
- 如果块自变量同时含有选项和自定义值，可单击当前值直接设置相关值。此外，双击当前值会在下拉菜单中显示可用选项。

4-2-7 输出窗口中的错误列表选项卡

单击输出窗口中的错误列表选项卡时，会显示与 DriveProgramming 有关的错误列表。错误列表会显示当前使用 DriveProgramming Editor 创建的程序在编译后出现的编译错误和警告。如果显示任何编译错误，则表明程序未得到正确编译。此时，系统不会将程序传送至变频器，也不会将流程图和文本之间进行转换。即便编译成功，也可能会显示一些警告，显示程序中需要予以纠正的内容。每次完成编译后，错误列表都会自动更新，清除已解决的错误和警告。默认情况下，输出窗口显示在 CX-Drive 底部。您可以双击标题栏（窗口上较宽的框架），将该窗口分离开来。



-  [Errors](错误) 按钮会切换列表中显示的错误。
-  [Warnings](警告) 按钮会切换列表中显示的警告。
-  [Messages](消息) 按钮会切换列表中显示的消息。

列表中的消息含有以下信息：

| 信息 | 说明 |
|----|------------------|
| 日期 | 显示出现错误的日期和时间。 |
| 组件 | 标识出现错误的块。 |
| 名称 | 显示错误或警告消息的名称或描述。 |

4-3 添加、删除和重命名任务

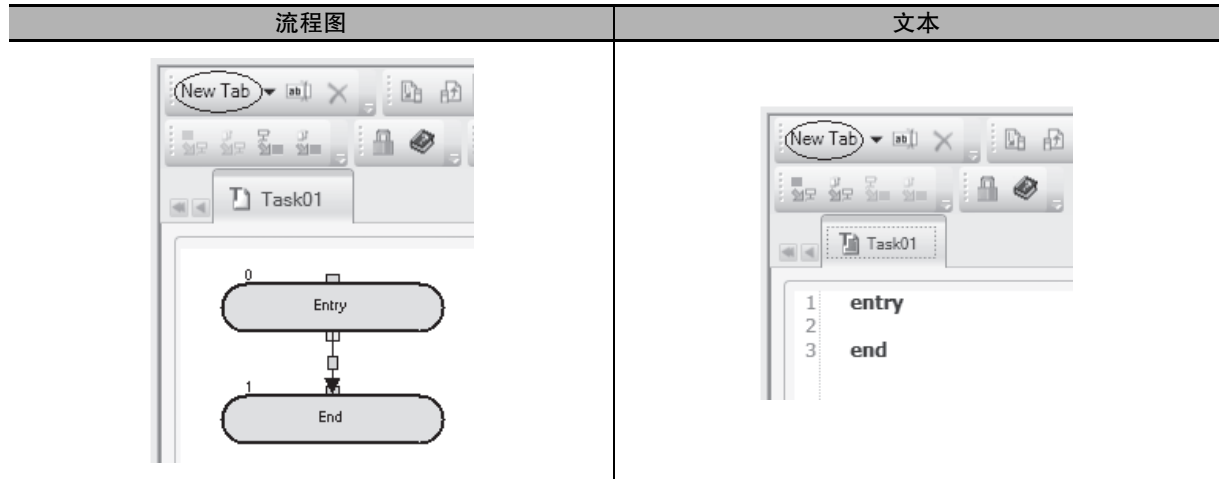
默认情况下，从 CX-Drive 启动时，DriveProgramming Editor 会显示一个空任务。

若要添加新任务，请在 DriveProgramming Editor 的工具栏中选择 [New Tab](新建选项卡)，然后选择 [New Task (flowchart)](新建任务 (流程图)) 或 [New Task (text)](新建任务 (文本))。

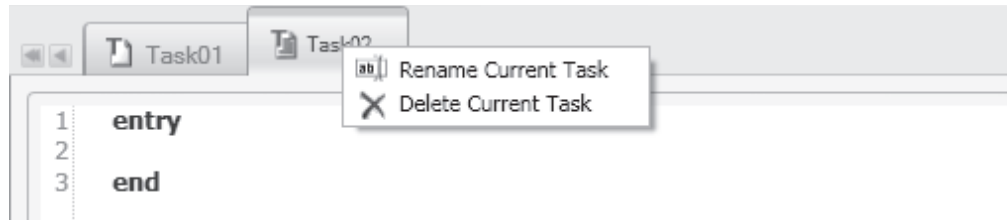
此时 DriveProgramming Editor 上会显示一个新任务。

在显示的任务中，更靠近左侧的任务在 2ms 处理周期中具有更高的优先级。

所有任务都必须从 “entry” 命令开始，以 “end” 命令结束。



在 DriveProgramming Editor 中右键单击选项卡时，可删除或重命名选定的任务。



4-4 插入、删除和调用子程序

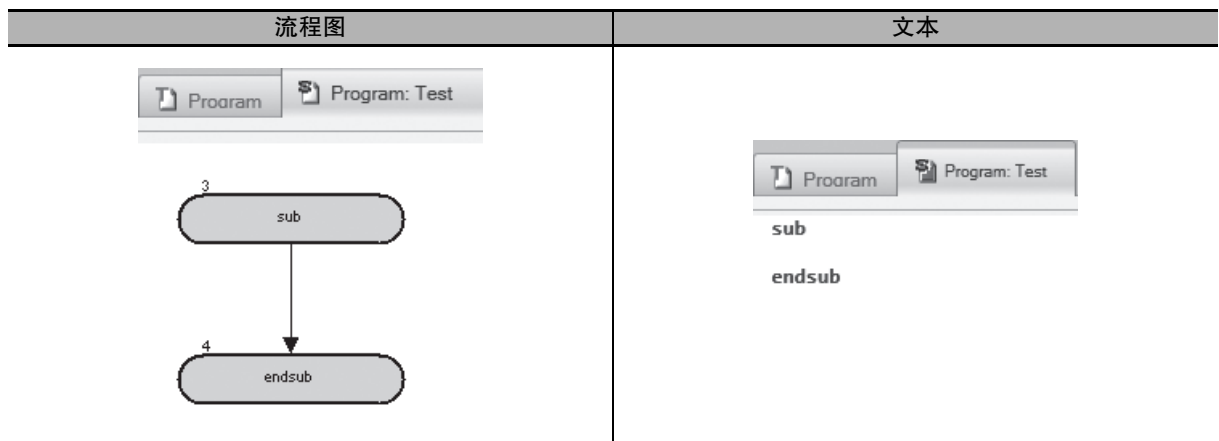
若要插入子程序，请选择要插入子程序的任务所在的选项卡，然后在 DriveProgramming Editor 的工具栏中选择 [New Tab](新建选项卡)- [New Subroutine (flowchart)](新建子程序 (流程图)) 或 [New Subroutine (text)](新建子程序 (文本))。

此时 DriveProgramming Editor 上会显示一个新的子程序。

与任务类似，可通过右键单击选项卡的方式删除或重命名子程序。

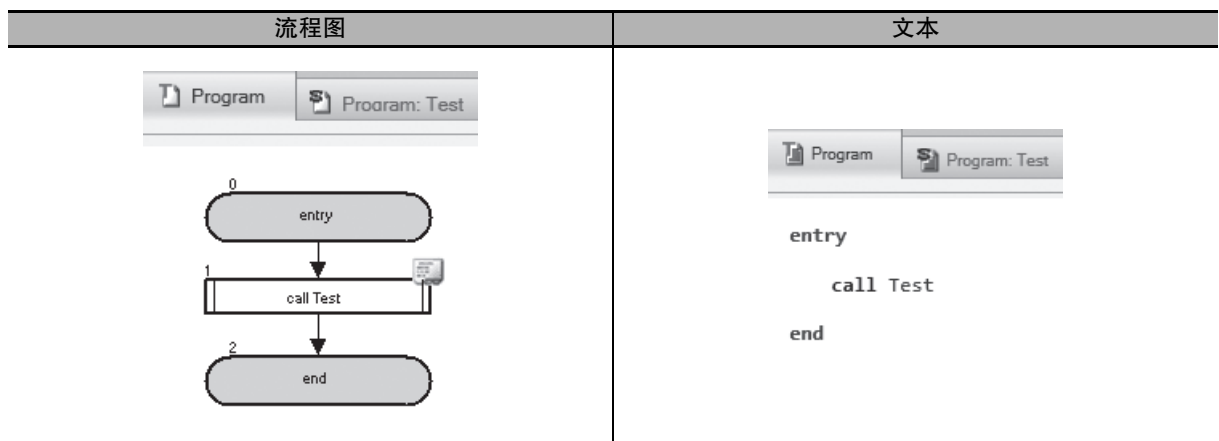
子程序名称会按“任务名称：子程序名称”的格式显示在子程序所属的任务名称右侧。

所有子程序都必须从“sub”命令开始，以“endsub”命令结束。



若要执行子程序，请在“call”命令中指定子程序名称。

调用时只能调用属于任务的子程序。各任务之间无法共享同一个子程序。若要将子程序用于多个任务，请在每个任务中插入同一个子程序。



4-5 创建流程图程序

创建 DriveProgramming 程序时，可为每个任务或子程序选择流程图或文本。
请按照以下步骤创建一个流程图程序。

- 1** 打开 DriveProgramming Editor。
DriveProgramming 辅助窗口 (工具箱、块参数、属性和错误列表选项卡) 会自动显示。
- 2** 从工具栏中的 [New Tab](新建选项卡) 选择 [New Task (flowchart)](新建任务 (流程图)) 或 [New Subroutine (flowchart)](新建子程序 (流程图))。
- 3** 从工具箱窗口中选择命令，然后通过拖动的方式将它们移动到 DriveProgramming Editor。
编译完成后，编辑器上每个块的左上方会显示块编号。
块编号为从“0”开始的连续编号。整个程序中的每个块都会分配一个块编号。
块编号与程序转换为文本后的行数并不一致。
- 4** 单击块可编辑其属性。
在属性窗口中，编辑与块关联的自变量。
- 5** 根据程序顺序将块连接起来。
连接块时，可将某个块的橙色接点拖动到另一个块的绿色接点。
- 6** 在块参数窗口中编辑用户参数。
- 7** 执行编译程序、将程序传送至变频器以及保存数据等操作。
执行编译，并检查程序中是否出现编译错误。
编译成功完成后，可将程序传送至变频器。
若要保存程序，请保存整个项目。或者，可使用导出程序功能单独保存程序。

4-6 创建文本程序

创建 DriveProgramming 程序时，可为每个任务或子程序选择流程图或文本。
请按照以下步骤创建一个文本程序。

- 1** 打开 DriveProgramming Editor。
DriveProgramming 辅助窗口 (工具箱、块参数、属性和错误列表选项卡) 会自动显示。
- 2** 从工具栏中的 [New Tab](新建选项卡) 选择 [New Task (text)](新建任务 (文本)) 或 [New Subroutine (text)](新建子程序 (文本))。
- 3** 编辑文本程序代码的方式有三种：
 - 手动键入
 - 调用文本命令列表 (Ctrl + 空格)
 - 从工具箱窗口拖动命令
- 4** 设置每个命令的自变量。
对于从工具箱窗口拖动而来的命令以及从文本命令列表 (Ctrl + 空格) 选择的命令，它们所需的自变量会显示为绿色背景。
设置每个自变量，完成命令。
每次通过手动键入的方式编辑命令时，弹出式支持都会显示，帮助您完成命令。有关每个命令的详细信息，请参阅第 6 章节 *DriveProgramming 命令*。
创建程序时，您无需担心不必要的行和空格，因为编译程序时会删除这些内容。
- 5** 执行编译程序、将程序传送至变频器以及保存数据等操作。
执行编译，并检查程序中是否出现编译错误。
编译成功完成后，可将程序传送至变频器。
若要保存程序，请保存整个项目。或者，可使用导出程序功能单独保存程序。

4-7 编辑传送 (上传) 的程序

您可以编辑从变频器传送 (上传) 的程序。
请按照以下步骤编辑程序。

- 1** 打开 DriveProgramming Editor。
DriveProgramming 辅助窗口 (工具箱、块参数和属性) 会自动显示。
- 2** 与 CX-Drive 建立在线连接。从菜单中选择 [Drive](驱动器)- [Work Online](在线工作)。或者, 单击 CX-Drive 工具栏中的 [Work Online](在线工作) 图标。
- 3** 单击 DriveProgramming Editor 工具栏中的 [Transfer from Drive](从驱动器传送) 图标。
此时会从驱动器(变频器)传送程序, 并自动显示在 DriveProgramming Editor 的 DriveProgramming 区域。
- 4** 编辑传送 (上传) 的程序。
“编译后程序”已下载至变频器。
因此, 传送 (上传) 的程序会显示为文本程序。
若要将其显示为流程图程序, 请单击 DriveProgramming Editor 工具栏中的 [Convert whole program to Flowchart](将整个程序转换为流程图), 从而将程序转换为流程图。
- 5** 执行编译程序、将程序传送至变频器以及保存数据等操作。
执行编译, 并检查程序中是否出现编译错误。
编译成功完成后, 可将程序传送至变频器。
若要保存程序, 请保存整个项目。或者, 可使用导出程序功能单独保存程序。

存在 DriveProgramming 程序时, 您可以使用 CX-Drive 工具栏中的 [Transfer to Drive](传送至驱动器) 或 [Transfer from Drive](从驱动器传送) 图标将它们传送至变频器或从变频器传送出来。此时会显示一个消息对话框, 询问您是否传送参数、程序或同时传送这两者, 此处需要选择 “程序”。

4-8 保存程序

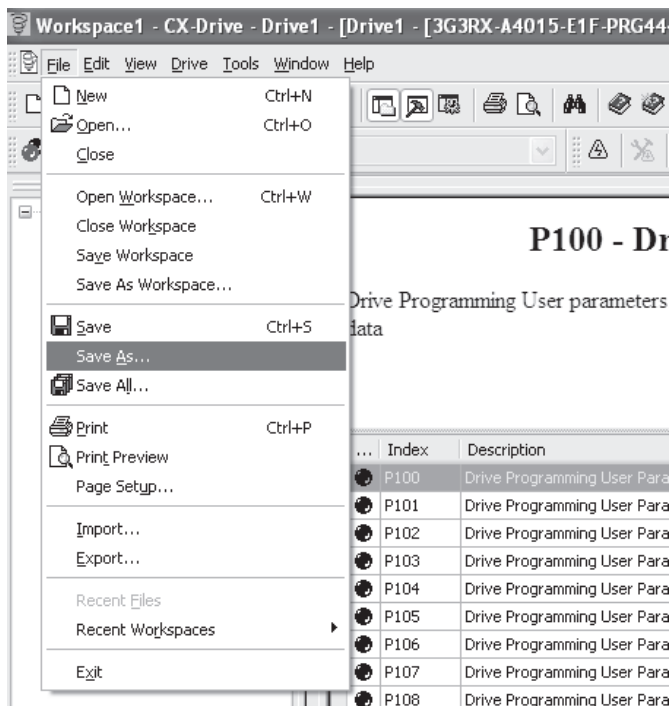
如下文所述，可使用两种方法保存 DriveProgramming 功能创建的程序。
请根据用途选择合适的方法。

● 保存整个 CX-Drive 项目

保存使用 CX-Drive 创建的整个项目时，可保存包括 DriveProgramming 程序在内的所有驱动器数据。
单击菜单栏中的 [File](文件)，然后选择 [Save As](另存为)，接着输入文件名。

打开保存的项目时，该项目中所含的 DriveProgramming 程序会自动加载。

您可以在工作区中双击 DriveProgramming 显示程序，然后启动 DriveProgramming Editor。

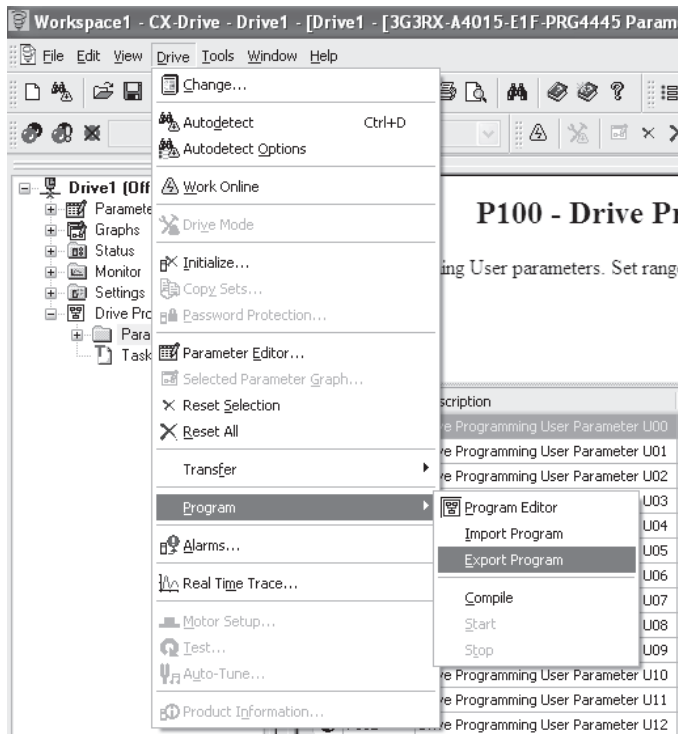


● 通过导出的方式保存 DriveProgramming 程序

您可以单独保存 DriveProgramming 程序。

单击菜单栏中的 [Drive](驱动器), 然后选择 [Program](程序)–[Export Program](导出程序), 接着输入文件名。CX-Drive 会将程序与其他驱动器信息分开, 然后仅保存程序。

若要将已导出的文件导入到 CX-Drive, 请单击菜单栏中的 [Drive](驱动器), 然后选择 [Program](程序)–[Import Program](导入程序), 接着选择文件名。



4-9 传送和验证程序

您可以在变频器和 CX-Drive 的 DriveProgramming 功能之间传送和验证程序。同时，也可以执行参数传送和验证。

传送 (从 PC 至变频器)

编译创建的程序 (检查程序)，如果未发现错误，再将其传送 (下载) 至变频器。
从 [Drive](驱动器) 菜单中，选择 [Transfer](传送)- [To Drive](至驱动器) 来传送 (下载) 数据。

传送 (从变频器到 PC)

将变频器中的程序和参数传送 (上传) 到 DriveProgramming。
从 [Drive](驱动器) 菜单中，选择 [Transfer](传送)- [From Drive](从驱动器) 来传送 (上传) 数据。

验证 (在 PC 和变频器之间)

将 DriveProgramming 程序和参数与变频器中的数据加以比较。
从 [Drive](驱动器) 菜单中，选择 [Transfer](传送)- [Compare with Drive](与驱动器比较) 来执行验证。

部分传送 (从 PC 至变频器)

从 [Drive](驱动器) 菜单中，选择 [Transfer](传送)- [Selection To Drive](将所选内容传送至驱动器) 来传送 (下载) 所选数据。

部分传送 (从变频器到 PC)



从 [Drive](驱动器) 菜单中，选择 [Transfer](传送)- [Selection From Drive](从驱动器传送所选内容) 来传送 (上传) 所选数据。

4-10 执行程序 (DriveProgramming 功能选择)

如下文所述，将程序传送 (下载) 至变频器后，可通过两种方式执行程序。

通过 CX-Drive 执行程序

通过 CX-Drive 执行时便于调试创建的程序。

使用 DriveProgramming Editor 工具栏中的  图标启动程序。若要停止程序，请使用  图标。请注意，仅在符合以下条件时，才能通过 CX-Drive 执行程序。

- CX-Drive 已连接至变频器。

程序启动后，即便 CX-Drive 断开它也不会停止。因此，执行程序时，请确保可通过关闭电源等方式立即停止变频器。

- 通过编译后验证确定 CX-Drive 中的程序与变频器中的程序一致。
- 执行程序停止时不受 CX-Drive 中程序的影响。

在应用程序中执行程序

您可以通过设置变频器参数 DriveProgramming 功能选择 (A017) 来执行程序。

除非重新启动程序，否则在程序启动并完成一系列进程后，便会执行 “end” 命令，结束运行。

若要重复运行程序，请创建一个循环程序，让程序在执行 “end” 命令之前执行循环程序。

如果 DriveProgramming 功能在电机运行期间停止，电机会根据停止选择 (b091)(减速停止或自由停止) 的设置停止。

| 参数编号 | 功能名称 | 数据 | 说明 |
|-------------|----------------------------|--------------------------------|---|
| A017 | DriveProgramming 功能选择 | 00: 禁用 (默认设置) | 禁用 DriveProgramming 功能。将不执行程序。如果在程序执行期间将设置更改为 00(禁用)，程序会停止。 |
| | | 01: 启用 (通过多功能输入 PRG 端子启动 / 停止) | 在设置为 82(PRG) 的多功能输入端子 *1 变为 ON 时启动 DriveProgramming 程序。 |
| | | 02: 启用 (接通 / 关闭电源时启动 / 停止) | 接通变频器电源后自动启动 DriveProgramming 程序。如果在程序停止期间将设置更改为 02(启用)，程序将会启动。 |
| C001 到 C007 | MX2 和 RX: 多功能输入 S1 到 S7 选择 | 82: PRG (DriveProgramming 启动) | DriveProgramming 功能选择 (A017) 设为 01(启用: 通过多功能输入 PRG 端子启动 / 停止) 时，程序会通过具备此设置的多功能输入端子启动。 |
| C008 | 仅限 RX: 多功能输入 S8 选择 | | |

*1. MX2 的多功能输入端子为 S1 到 S7，RX 的多功能输入端子为 S1 到 S8。



安全使用注意事项

DriveProgramming 程序停止执行后，DriveProgramming 控制的多功能输出会保留程序停止之前的状态。

因此需对系统进行相关配置，确保变频器中 DriveProgramming 程序停止操作可被 DriveProgramming 启动信号和报警 (跳闸) 信号检测到，而且变频器周边设备可安全停止。



正确使用注意事项

以下是 DriveProgramming 程序停止时功能变量的状态。请根据每种状态采取必要的措施。

- 对于多功能输出 (变频器实际输出端子), 它们会保留程序停止之前的状态。
但是, 未设为 MO1 到 MO6 的输出端子 (DriveProgramming 的通用输出) 会作为变频器的正常多功能输出发挥作用。
- 为频率基准、加速 / 减速时间或模拟输出选择了 DriveProgramming 功能时, 这些功能在程序停止之前的设置值也会保留。
- 程序停止之前的用户参数变量、内部用户变量和内部用户接点数据会保留下来。
- 变频器实际输入端子的状态 (比如多功能和模拟输入端子) 不会保留下来, 而是会发生更新。
- 多功能输出变量 (RUN、FA1 和 AL 等功能位) 和变频器监视器变量数据不会保留下来, 而是始终根据变频器状态进行更新。
- 只有 DriveProgramming 的多功能输入变量 (FW、RV 和 CF1 等功能位) 以及定时器变量会在程序停止时清除, 而且所有数据都归零。

通过 CX-Drive 启动 / 停止 DriveProgramming 功能时, 变频器参数 DriveProgramming 功能选择 (A017) 的值只会在 RAM 数据中暂时更改为 02(启动) 或 00(停止)。

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

- 重新接通变频器电源, 然后将 A017 的值恢复为 EEPROM 中保存的值。
- 在接通变频器电源之前, 请勿执行以下 EEPROM 保存操作。
 - 从 CX-Drive 传送 (下载) 一部分参数。
 - 通过 Modbus 通信或通信选件发出 “enter” 命令。

A017 设为 02 时, 请先停止在接通电源时启动的正在运行程序, 然后再通过 CX-Drive 重新启动程序。

如果在接通电源时启动的程序仍在运行, CX-Drive 便无法重新启动程序。

变频器执行操作期间, 如果在 CX-Drive 中单击按钮启动程序, 则最后一步会显示以下消息。

“The program is running, so it cannot be transferred to the drive.” (程序正在运行, 因此无法将其传送到驱动器。)



如果此时单击 [OK](确定), 程序会不受变频器操作状态的影响强行启动。

在启动程序之前, 请检查设备状态, 确保安全。

4-11 其他有用功能

将流程图转换为文本

将流程图程序转换为文本程序的方法有两种。

| 命令 | 图标 | 说明 |
|------------|---|--------------------|
| 将流程图转换为文本 |  | 将当前流程图任务或子程序转换为文本。 |
| 将整个程序转换为文本 |  | 将整个程序转换为文本。 |





正确使用注意事项

- 将流程图程序转换为文本程序时，程序会编译一次。如果出现编译错误，转换操作会失败。
- 编译完成后，流程图程序中每个块的左上方会显示块编号。
块编号为从“0”开始的连续编号。整个程序中的每个块都会分配一个块编号。
块编号与程序转换为文本后的行数并不一致。

将文本转换为流程图

将文本程序转换为流程图程序的方法有两种。

| 命令 | 图标 | 说明 |
|-------------|---|--------------------|
| 将文本转换为流程图 |  | 将当前文本任务或子程序转换为流程图。 |
| 将整个程序转换为流程图 |  | 将整个程序转换为流程图。 |





正确使用注意事项

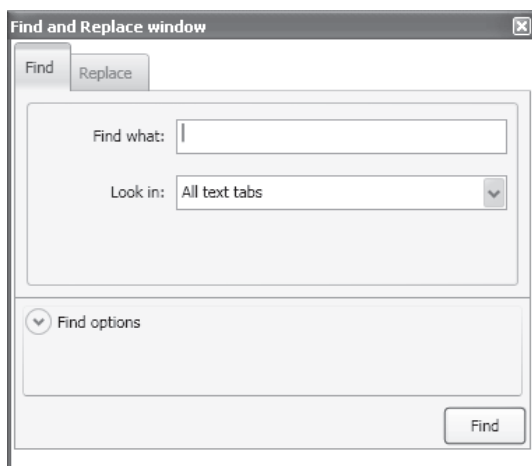
- 将文本程序转换为流程图程序时，程序会编译一次。如果出现编译错误，转换操作会失败。
- 编译完成后，流程图程序中每个块的左上方会显示块编号。
块编号为从“0”开始的连续编号。整个程序中的每个块都会分配一个块编号。
块编号与程序转换为文本后的行数并不一致。
- 转换期间执行编译时，文本程序中的注释、别名定义、区域定义等内容都会被删除。但是，程序只会发生部分更改，该操作不会改变程序的格式和内容。

查找并替换功能

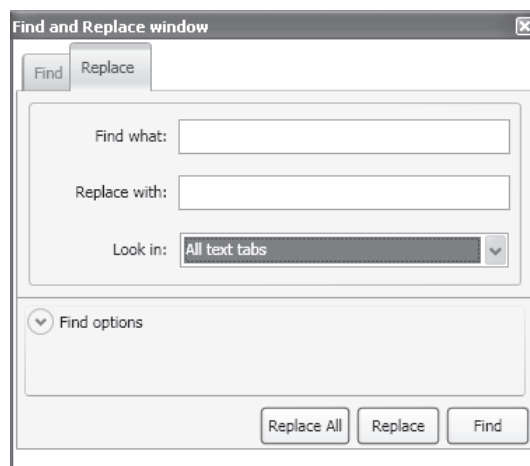
查找并替换功能只能在文本程序中使用。使用查找并替换功能时，可查找或替换文本程序中的任何字符串。

若要使用查找功能，请单击  图标，或按快捷键 Ctrl + F 并选择 [Find](查找) 选项卡。

若要使用替换功能，请单击  图标，或按快捷键 Ctrl + F 并选择 [Replace](替换) 选项卡。



查找



替换

添加注释 (文本程序)

您可以为文本任务或子程序添加注释。您无法为流程图程序添加注释。若要在文本行中添加注释，请先输入单引号 '，然后键入注释内容。注释呈绿色显示。

● 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
```



正确使用注意事项

- 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。若要保存创建的注释，请先保存程序，然后再执行程序编译。可通过保存 CX-Drive 中的整个项目或导出程序文件的方式保存程序。
- 在验证过程中，“编译后程序”会与变频器内部程序加以对比。因此，注释、别名定义、区域定义等内容都不会进行验证。

别名定义 (文本程序)

您可以在文本程序任务中的“entry”命令之前定义别名。但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、命令和数值常数指定名称。

使用别名定义时，可像处理参数、变量、命令和数值那样，在程序中处理指定的名称。这对加强程序的可读性十分有用。

- **本地别名：**您可以为当前所选的任务或子程序使用此定义。但无法将其用于程序中的其他任务或子程序。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：**您可以将此定义用于程序中的所有任务和子程序。可在最多全部五个任务中设置别名定义。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
```



正确使用注意事项

- 编译程序以便将其传送 (下载) 至变频器或进行程序转换时，文本程序中创建的别名定义会被删除。指定的名称会转换为正常名称和数值。若要保存创建的别名定义，请先保存程序，然后再执行程序编译。可通过保存 CX-Drive 中的整个项目或导出程序文件的方式保存程序。
- 在验证过程中，“编译后程序”会与变频器内部程序加以对比。因此，注释、别名定义、区域定义等内容都不会进行验证。
- 使用别名定义时，无法使用已被使用的变量或命令，否则将显示编译错误。

区域定义 (文本程序)

您可以在文本程序的任务或子程序中定义区域。但无法在流程图程序中定义区域。

区域定义是指通过指定区域将程序分为多组行。

使用区域定义时，可将程序分为多个部分，并在显示时将每个部分折叠起来。这有助于提高程序的可读性。

● 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...

```



正确使用注意事项

- 编译程序以便将其传送 (下载) 至变频器或进行程序转换时，文本程序中创建的区域定义会被删除。
若要保存创建的区域定义，请先保存程序，然后再执行程序编译。可通过保存 CX-Drive 中的整个项目或导出程序文件的方式保存程序。
- 在验证过程中，“编译后程序”会与变频器内部程序加以对比。因此，注释、别名定义、区域定义等内容都不会进行验证。

5

DriveProgramming 用户变量

本章节介绍 DriveProgramming 的用户变量。

| | |
|-----------------------|------|
| 5-1 用户变量和用户参数 | 5-2 |
| 5-2 输入 / 输出端子变量 | 5-4 |
| 5-3 定时器变量 | 5-9 |
| 5-4 变频器设置变量 | 5-10 |
| 5-5 变频器监控变量 | 5-12 |
| 5-6 多功能输入变量 | 5-14 |
| 5-7 多功能输出变量 | 5-16 |

5-1 用户变量和用户参数

以下变量可在创建参数时使用：用户参数变量、内部用户变量和内部用户接点。
请将这些变量用于程序用户界面、计算使用的初始数据、计算期间的数据保存、数据保存等等。

用户参数变量 U(00) 到 U(31)

DriveProgramming 用户参数变量 U(00) 到 U(31) 属于无符号单字变量。
变频器的 DriveProgramming 用户参数 U00 到 U31(P100 到 P131) 分别对应着 DriveProgramming 用户参数变量 U(00) 到 U(31)。请在以下应用中使用这些变量。

| 应用 | 说明 |
|-----------|--|
| 用户界面 | 使用变量作为参数来调整计算结果。您可以根据应用状态使用变频器参数 (P100 到 P131) 来调整它们。 |
| 计算使用的初始数据 | 您可以使用用户参数变量 U(00) 到 U(31) 作为计算变量。EEPROM 中保存的参数 P100 到 P131 的数据可用作计算使用的初始数据。 |
| 计算期间保存数据 | 计算期间可使用变量暂时保存数据。 |
| 数据保存 | 通过 “eepwrt” 命令，可将计算结果保存到 EEPROM 中，以便在关闭电源后仍能保存该结果。 使用参数 U(29) 到 U(31) 时，计算结果会在断电时自动存储在 EEPROM 中。电源重新恢复时，保存的数据可用作初始数据。 |

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|---------------|-----------------------------|------------|------|----|-----------|----|
| U(00) 到 U(31) | 用户参数变量 (对应着 P100 到 P131) | 0 到 65,535 | 0 | - | 无符号 单字 | 读写 |

- 接通电源时，EEPROM 中保存的参数 P100 到 P131 的数据会自动设置给用户参数变量 U(00) 到 U(31)。
- 通过监控参数 P100 到 P131，可在程序开始执行后检查用户参数变量 U(00) 到 U(31) 的数据。请注意，数字操作器上显示的数据为显示时的数据。若要更新数据，请先显示参数编号，然后再显示数据。
- 使用数字操作器更改参数 P100 到 P131 并按 Enter 键时，更改的值会保存到 EEPROM 中，并反映在当前用户参数变量 U(00) 到 U(31) 中。重新接通电源时，EEPROM 中保存的数据会自动设置给变量 U(00) 到 U(31)。
- 即使用户参数变量 U(00) 到 U(31) 的数据在程序中发生更改，更改后的数据也不会作为参数 P100 到 P131 的数据保存在 EEPROM 中。
- 若要将设置给程序中用户参数变量 U(00) 到 U(31) 的数据保存到 EEPROM，请先后执行 “eepwrt” 和 “ChgParam” 命令，然后指定要保存的数据。随后数据会保存到 EEPROM 中，并分配给参数 P100 到 P131。
- 只有用户参数变量 U(29) 到 U(31) 的数据会在断电时自动保存。这些数据会保存在 EEPROM 中，对应着参数 P129 到 P131。断电时用户参数变量 U(00) 到 U(28) 的数据不会保存。



正确使用注意事项

DriveProgramming 程序停止时，程序停止前的用户参数变量数据会保留下来。重新执行程序时，进程会开始使用此前保留的数据。

内部用户变量 UL(00) 到 UL(07)

DriveProgramming 的内部用户变量 UL(00) 到 UL(07) 为带符号双字变量。执行算术运算等计算期间，可使用这些变量保存数据。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|-----------------|--------|--------------------------------|------|----|-------|----|
| UL(00) 到 UL(07) | 内部用户变量 | -2,147,483,648 到 2,147,483,647 | 0 | - | 带符号双字 | 读写 |

- 接通电源时，内部用户变量 UL(00) 到 UL(07) 会清零。它们的数据无法通过任何方式（比如保存到 EEPROM 中）保存下来。
- 若要设置初始数据，请创建程序，并在程序中将初始数据设置给 UL(00) 到 UL(07)。
- 若要将内部用户变量 UL(00) 到 UL(07) 中的高位字数据移动到单字用户参数变量 U(00) 到 U(31)，请执行以下操作。
 - 分配正高位字数据： $U(00) = UL(00)/65536$
 - 分配负高位字数据： $U(00) = UL(00)/65535$
 执行 $U(00) = UL(00)$ 会将低位字数据移动到 U(00)。



正确使用注意事项

DriveProgramming 程序停止时，程序停止前的内部用户变量会保留下来。重新执行程序时，进程会开始使用此前保留的数据。

内部用户接点 UB(0) 到 UB(7)

DriveProgramming 的内部用户接点 UB(0) 到 UB(7) 属于按位访问变量。使用这些变量可以在未操作期间保存数据。

此外，也可以使用变量 UBw 作为按字访问数据，而该数据的低位字节含有内部用户接点 UB(0) 到 UB(7)。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|---------------|---------------|-----------------|------|----|------|----|
| UB(0) 到 UB(7) | 内部用户接点 (按位访问) | 0: OFF 1: ON | 0 | - | 位 | 读写 |

- 您可以将内部用户接点 UB(0) 到 UB(7) 用作位大小变量。
- 接通电源时，UB(0) 到 UB(7) 会清零。它们的数据无法通过任何方式（比如保存到 EEPROM 中）保存下来。
- 若要设置初始数据，请创建程序，并在程序中将初始数据设置给 UB(0) 到 UB(7)。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|---------|------|----|-------|----|
| UBw | 内部用户接点 (按字访问) | 0 到 255 | 0 | - | 无符号单字 | 读写 |

- 内部用户接点 (按字访问) UBw 功能将内部用户接点 UB(0) 到 UB(7) 用作字大小变量。
- 内部用户接点 UB(0) 到 UB(7) 会设置在低位字节。高位字节数据均为零。如果将数据写入高位字节，这些数据会被忽略。

| 位 | 8 到 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| 功能 | 不使用 | UB(7) | UB(6) | UB(5) | UB(4) | UB(3) | UB(2) | UB(1) | UB(0) |



正确使用注意事项

DriveProgramming 程序停止时，程序停止前的内部用户接点会保留下来。重新执行程序时，进程会开始使用此前保留的数据。

5-2 输入 / 输出端子变量

本章节介绍将以下变频器端子用于 DriveProgramming 功能时提供的变量: 多功能输入端子、多功能输出端子、多功能继电器输出、模拟输入端子和模拟输出端子。请将这些变量用作变频器周边设备与 DriveProgramming 功能之间的接口。

输入端子变量 X(00) 到 X(07)

您可以将变频器的多功能输入端子用作 DriveProgramming 功能的输入端子变量 X(00) 到 X(07), 方法是将这些多功能输入端子设为通用输入 MI1 到 MI8。

此外, 也可以将输入端子变量 X(00) 到 X(07) 用作按字访问数据, 也就是将其设置给变量 Xw 的低位字节。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|---------------|------------------|-----------------|------|----|------|----|
| X(00) 到 X(07) | 输入端子变量 (按位访问) | 0: OFF 1: ON | 0 | - | 位 | R |

MX2:

- 多功能输入端子 S1 到 S7 和脉冲串输入 RP 端子的状态会捕获下来, 设置为位大小变量。此变量为只读变量。
- 将多功能输入 S1 到 S7 选择 (C001 到 C007) 设为 56 到 62(MI1 到 MI7: 通用输入), 并将脉冲串输入 RP 选择 (P003) 设为 02(MI8: DriveProgramming 的通用输入 8) 时, 输入端子 S1 到 S7 以及 RP 端子的状态会捕获下来, 设置为 DriveProgramming 的输入端子变量 X(00) 到 X(07)。

输入端子变量 X(00) 到 X(07) 会根据设置的通用输入 MI1 到 MI7 的数字顺序而不是端子编号 S1 到 S7 获得相应编号。

RX:

- 多功能输入端子 S1 到 S8 的状态会捕获下来, 设置为位大小变量。此变量为只读变量。
- 多功能输入 S1 到 S8 选择 (C001 到 C008) 设为 56 到 63(MI1 到 MI8: 通用输入) 时, S1 到 S8 的输入端子状态会捕获下来, 设置为 DriveProgramming 输入端子变量 X(00) 到 X(07)。

输入端子变量 X(00) 到 X(07) 会根据设置的通用输入 MI1 到 MI8 的数字顺序而不是端子编号 S1 到 S8 获得相应编号。

| 功能变量 | 设置多功能输入选择和 脉冲串输入 RP 选择 (MX2: C001 到 C007 和 P003/RX: C001 到 C008) |
|-------|--|
| X(00) | 56: MI1 |
| X(01) | 57: MI2 |
| X(02) | 58: MI3 |
| X(03) | 59: MI4 |
| X(04) | 60: MI5 |
| X(05) | 61: MI6 |
| X(06) | 62: MI7 |
| X(07) | MX2 P003 = 02/RX 63: MI8 |

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|------------------|---------|------|----|-----------|----|
| Xw | 输入端子变量 (按字访问) | 0 到 255 | 0 | - | 无符号 单字 | R |

- 输入端子变量(按字访问)Xw功能会将输入端子变量X(00)到X(07)用作字大小变量。此变量为只读变量。
- 输入端子变量 X(00) 到 X(07) 会设定给低位字节。对于高位字节和未使用的输入端子变量，它们的读数均为零。

| 位 | 8 到 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| 功能 | 不使用 | X(07) | X(06) | X(05) | X(04) | X(03) | X(02) | X(01) | X(00) |



正确使用注意事项

- DriveProgramming 程序停止时，输入端子变量的状态不会保留下来，但它们会根据实际输入端子状态更新。
- 在多功能输入端子 (MX2: S1 到 S7/RX: S1 到 S8) 被设置为 MI1 到 MI8(通用输入 1 到 8) 的情况下，仍可将多功能输入 S1 到 S7 操作选择 (MX2: C011 到 C017/RX: C011 到 C018) 设置为 NO (常开接点) 或 NC(常闭接点)。

输出端子变量 MX2: Y(00) 到 Y(02)/RX: Y(00) 到 Y(05)

您可以将变频器多功能输出端子用作 DriveProgramming 功能的输出端子变量 MX2: Y(00) 到 Y(02)/RX: Y(00) 到 Y(05)，方法是将多功能输出端子设为通用输出 MX2: MO1 到 MO3/RX: MO1 到 MO6。

此外，也可以将输出端子变量 Y(00) 到 Y(05) 用作按字访问数据，也就是将其设置给变量 Yw 的低位字节。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|---|------------------|-----------------|------|----|------|----|
| MX2: Y(00) 到 Y(02) RX: Y(00) 到 Y(05) | 输出端子变量 (按位访问) | 0: OFF 1: ON | 0 | - | 位 | R |

MX2:

- 您可以控制多功能输出端子 P1 和 P2 以及多功能继电器输出端子 (MA 和 MB) 的状态，将它们用作位大小变量。
- 通过将多功能输出 P1 和 P2 选择 (C021 和 C022) 或多功能继电器输出 (MA、MB) 功能选择 (C026) 设为 44 到 46(MO1 到 MO3: 通用输出)，您可以将输出端子 P1 和 P2 或继电器输出端子 (MA 和 MB) 用作 DriveProgramming 的输出端子变量 Y(00) 到 Y(02)。输出端子变量 Y(00) 到 Y(02) 会根据设置的通用输出 MO1 到 MO3 的数字顺序而不是端子编号 P1、P2、MA 或 MB 获得相应编号。
- 3G3MX2-ZV1 系列变频器的最大通用输出数量为三个 (MO1 到 MO3)。

RX:

- 您可以控制多功能输出端子 P1 到 P5 以及多功能继电器输出端子 (MA 和 MB) 的状态，将它们用作位大小变量。
- 通过将多功能输出 P1 和 P5 选择 (C021 到 C025) 或多功能继电器输出 (MA、MB) 功能选择 (C026) 设为 44 到 49(MO1 到 MO6: 通用输出)，您可以将输出端子 P1 到 P5 或继电器输出端子 (MA 和 MB) 用作 DriveProgramming 的输出端子变量 Y(00) 到 Y(05)。输出端子变量 Y(00) 到 Y(05) 会根据设置的通用输出 MO1 到 MO6 的数字顺序而不是端子编号 P1 到 P5、MA 或 MB 获得相应编号。

| 功能变量 | 多功能输出选择设置 (MX2: C021、C022 和 C026/RX: C021 到 C026) |
|-------|--|
| Y(00) | MX2 和 RX 44: MO1 |
| Y(01) | MX2 和 RX 45: MO2 |
| Y(02) | MX2 和 RX 46: MO3 |
| Y(03) | 仅限 RX 47: MO4 |
| Y(04) | 仅限 RX 48: MO5 |
| Y(05) | 仅限 RX 49: MO6 |

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|------------------|---------|------|----|-----------|----|
| Yw | 输出端子变量 (按字访问) | 0 到 255 | 0 | - | 无符号 单字 | R |

- 输出端子变量(按字访问)Yw功能将输出端子变量MX2: Y(00)到Y(02)/RX: Y(00)到Y(05)用作字大小变量。
- 输出端子变量 Y(00) 到 Y(05) 会设定给低位字节。对于高位字节和未使用的输出端子变量，它们的读数均为零。如果将数据写入高位字节，这些数据会被忽略。

| 型号 | 位 | 8 到 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|----|--------|-----|-----|-------|-------|-------|-------|-------|-------|
| MX2 | 功能 | 不使用 | 不使用 | 不使用 | 不使用 | 不使用 | 不使用 | Y(02) | Y(01) | Y(00) |
| RX | 功能 | 不使用 | 不使用 | 不使用 | Y(05) | Y(04) | Y(03) | Y(02) | Y(01) | Y(00) |



安全使用注意事项

DriveProgramming 程序停止后，DriveProgramming 控制的多功能输出会保留程序停止之前的状态。因此需对系统进行相关配置，确保变频器中 DriveProgramming 程序停止操作可被 DriveProgramming 启动信号和报警(跳闸)信号检测到，而且变频器周边设备可安全停止。



正确使用注意事项

- DriveProgramming 程序停止时，程序停止前的输出端子变量数据会保留下来。重新执行程序时，进程会开始使用此前保留的数据。
但是，如果输出未设为 DriveProgramming 使用的 MO1 到 MO6(通用输出)，那么它们将作为变频器多功能输出加以控制，而不受程序的影响。
- 如果多个任务使用同一个多功能端子，则最后执行的任务的输出状态有效。出于安全控制考虑，我们建议您避免使用多个任务进行控制。
- 在多功能输出选择(MX2: P1 和 P2/RX: P1 到 P5)或多功能继电器输出(MA、MB)功能选择被设置为 MO1 到 MO6(通用输出 1 到 6)的情况下，仍可将多功能输出操作选择(MX2: C031 和 C032/RX: C031 到 C035)或为多功能继电器输出(MA、MB)操作选择(C036)设置为 NO(常开接点)或 NC(常闭接点)。

模拟输入端子变量 XA(0) 到 XA(2)

您可以将变频器的频率基准输入 (模拟电压输入)FV 端子和频率基准输入 (模拟电流输入)FI 端子, 用作 DriveProgramming 功能的模拟输入端子变量 XA(0) 和 XA(1)。

对于 3G3RX-V1 系列变频器, 还可将频率基准辅助输入 (模拟电压输入)FE 端子用作 DriveProgramming 功能的模拟输入端子变量 XA(2)。

不论参数设置如何, 您都能持续监控模拟输入的状态。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|-------|--|------------------|------|-------|-----------|----|
| XA(0) | MX2 和 RX: 模拟输入端子变量 (FV 和 FC 端子之间: 0 到 10V 输入) | 0 到 10,000 | 0 | 0.01% | 无符号 单字 | R |
| XA(1) | MX2 和 RX: 模拟输入端子变量 (FI 和 FC 端子之间: 4 到 20mA 输入) | | | | | |
| XA(2) | 仅限 RX: 模拟输入端子变量 (FE 和 FC 端子之间: -10 到 10V 输入) | -10,000 到 10,000 | | | 带符号 单字 | |

• 模拟输入端子变量 XA(0) 和 XA(1) 属于无符号单字变量, 而 XA(2) 属于带符号单字变量。此变量为只读变量。

• 这些变量会以最大模拟输入 10V 或 20mA 的 0.01% 为增量进行显示。

• 通过设置如下变频器参数, 您可以将变频器功能分配给模拟输入端子。请在每个参数中选择一项功能。

如果只将模拟输入端子用于 DriveProgramming, 请勿在以下参数中选择模拟输入端子。

MX2: A001、A201、A005、A076、A079、A141、A142、b040 和 P033

RX: A001、A005、A006、A076、A079、A141、A142、b040 和 P033

• 若要调整模拟输入, 请使用以下变频器参数: (FV: A011 到 A016, FI: A101 到 A105, 以及 FE: A111 到 A114)。



正确使用注意事项

DriveProgramming 程序停止时, 模拟输入端子变量的状态不会保留下来, 但它们会根据实际输入端子状态更新。

模拟输出端子变量 YA(0) 到 YA(2)

您可以将变频器的多功能数字输出 (PWM 输出)MP 端子和多功能模拟输出 (电压输出)AM 端子, 用作 DriveProgramming 功能的输出端子变量 YA(0) 和 YA(1)。

对于 3G3RX-V1 系列变频器, 还可将多功能模拟输出 (电流输出)AMI 端子用作 DriveProgramming 功能的模拟输出端子变量 YA(2)。

不论参数设置如何, 您都能持续监控模拟输出的状态。

若要通过 DriveProgramming 功能控制模拟输出, 请为变频器参数 MP 选择 (C027)、AM 选择 (C028) 以及 AMI 选择 (C029) 选择 DriveProgramming。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|-------|---|------------|------|-------|-----------|----|
| YA(0) | MX2 和 RX: 模拟输出端子变量 (MP 端子: 6.4ms 周期, PWM 输出) | 0 到 10,000 | 0 | 0.01% | 无符号 单字 | 读写 |
| YA(1) | MX2 和 RX: 模拟输出端子变量 (AM 端子: 0 到 10V 输出) | | | | | |
| YA(2) | 仅限 RX: 模拟输出端子变量 (AMI 端子: 4 到 20mA 输出) | | | | | |

- 模拟输出端子变量 YA(0) 到 YA(2) 属于无符号单字变量。
- 请以最大输出负载 10V 或 20mA 的 0.01% 为增量设置这些变量。
- 为变频器参数 MP 选择 (C027)、AM 选择 (C028) 和 AMI 选择 (C029) 的设置选择 DriveProgramming, 以便通过 DriveProgramming 功能控制模拟输出端子。即便没有为参数选择 DriveProgramming, 您也可以监控模拟输出端子的状态。

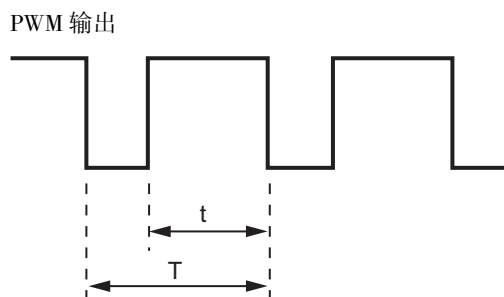
| 功能变量 | 参数设定 |
|--------|---|
| YA(00) | 将 MP 选择 (C027) 设置为 12(DriveProgramming: YA(0)) |
| YA(01) | 将 AM 选择 (C028) 设置为 13(DriveProgramming: YA(1)) |
| YA(02) | 将 AMI 选择 (C029) 设置为 14(DriveProgramming: YA(2)) |

- 使用变频器参数 (C105 到 C107、C109 和 C110) 调节模拟输出。



正确使用注意事项

- DriveProgramming 程序停止时, 程序停止前的模拟输出端子变量数据会保留下来。重新执行程序时, 进程会开始使用此前保留的数据。但是, 如果输出未设为 DriveProgramming, 那么它们将作为变频器模拟输出加以控制, 而不受程序的影响。
- 多功能数字输出 (PWM 输出)MP 端子会提供 PWM 信号输出。
该端子会在 6.4ms 周期内, 输出 0.00 到 100.00%(变量) 之间的值作为脉冲宽度 (占空比 t/T)。



周期 T: 恒定 (6.4ms)

占空比 t/T : 可变

5-3 定时器变量

本章节介绍为 DriveProgramming 定时器控制命令提供的定时器变量。

定时器变量 TC(0) 到 TC(7) 以及 TD(0) 到 TD(7)

DriveProgramming 的定时器控制命令中还会使用定时器计数器变量和定时器输出接点。

在定时器控制命令继电器 ON/OFF 或定时器设置中指定了定时器输出接点时，对应于指定接点的定时器计数器变量会开始计数。计数器达到指定值时，定时器输出接点会变为 ON。

如果定时器控制命令中未使用变量，它们将用作自由运行计数器。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|---------------|----------|-------------------|------|------|-----------|----|
| TC(0) 到 TC(7) | 定时器计数器变量 | 0 到 2,147,483,647 | 0 | 10ms | 无符号 双字 | R |

- 定时器计数器变量 TC(0) 到 TC(7) 属于无符号双字变量。此变量为只读变量。
- 定时器计数器变量 TC(0) 到 TC(7) 分别对应着定时器输出接点 TD(0) 到 TD(7)。
- 定时器计数器变量 TC(0) 到 TC(7) 用作 31 位自由运行计数器。它们在程序启动时从零开始计数，然后每 10ms 计数一次。
- 在继电器 ON/OFF 命令或定时器设置命令中指定了定时器输出接点时，TC(0) 到 TC(7) 中的对应定时器计数器变量会发挥定时器控制计数器的作用。

执行命令时，变量 TC(0) 到 TC(7) 会清零，且每 10ms 计数一次。当它们达到指定值时，计数便会停止。

- 执行定时器停止命令时，对应的定时器计数器变量会清零，然后用作 31 位自由运行计数器，且每 10ms 计数一次。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|---------------|-------------------|-----------------|------|----|------|----|
| TD(0) 到 TD(7) | 定时器输出接点 (按位访问) | 0: OFF 1: ON | 0 | - | 位 | R |

- 定时器输出接点 TD(0) 到 TD(7) 属于位大小变量。此变量为只读变量。
- 程序启动时，定时器输出接点 TD(0) 到 TD(7) 的值会清零。
- 在继电器 ON/OFF 命令或定时器设置命令中指定定时器输出接点 TD(0) 到 TD(7) 后，它们会开始操作。TD(0)到TD(7)会在它们对应的定时器计数器变量清零时设为 0(OFF)。定时器计数器变量达到指定值时，接点会设为“1”(ON)，且状态会保留下来。如果执行了定时器停止命令，TD(0)到TD(7)的状态会清零。
- 定时器计数器变量 TC(0) 到 TC(7) 会用作自由运行计数器，对应的定时器输出接点 TD(0) 到 TD(7) 的状态也不会更改。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|-------------------|---------|------|---------|-----------|----|
| TDw | 定时器输出接点 (按字访问) | 0 到 255 | 0 | 0.01 Hz | 无符号 单字 | R |

- 定时器输出接点 (按字访问)TDw 功能将定时器输出接点 TD(0) 到 TD(7) 用作字大小变量。此变量为只读变量。
- 定时器输出接点 TD(0) 到 TD(7) 会设置给低位字节。高位字节数据的读数则为零。

| 位 | 8 到 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| 功能 | 不使用 | TD(7) | TD(6) | TD(5) | TD(4) | TD(3) | TD(2) | TD(1) | TD(0) |



正确使用注意事项

如果 DriveProgramming 程序停止，定时器计数器变量和定时器输出接点的数据不会保留下来，而是清零。

5-4 变频器设置变量

本章节介绍为设置变频器频率基准和加速 / 减速时间提供的变量。
使用这些变量可以通过 DriveProgramming 程序控制变频器。

频率基准变量 SET-Freq

通过 DriveProgramming 功能直接控制频率基准时，请将变频器参数频率基准选择设为 07(DriveProgramming)，以便启用频率基准变量 SET-Freq。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|----------|--------|------------|------|---------|-----------|----|
| SET-Freq | 频率基准变量 | 0 到 40,000 | 0 | 0.01 Hz | 无符号 单字 | 读写 |

- 频率基准变量 SET-Freq 属于无符号单字变量。
- 仅当将频率基准选择设为 07(DriveProgramming) 时才能启用该变量。
 - MX2: 在第一个 / 第二个频率基准选择 (A001/A201) 中设置。
将多功能输入 S1 到 S7 选择 (C001 到 C007) 之一分配给 08(设置: 第二个控制), 以便切换控制。
 - RX: 在频率基准选择 (A001) 中设置。无法在第一个 / 第二个 / 第三个控制之间切换。
- 接通电源时，频率基准变量 SET-Freq 会清零。
它们的数据无法通过任何方式 (比如保存到 EEPROM 中) 保存下来。
- 若要设置初始数据，请创建程序，并在程序中将初始数据设置给 SET-Freq。
- 变频器可以实际输出的频率范围介于启动频率 (b082) 和最大频率之间。
如果设置的数据超出此范围，变频器操作如下。
 - 低于启动频率 (b082)
 - MX2 和 RX: 按频率 0.00Hz 操作。
 - 仅限 RX: 第一个控制方法 (A044) 设为 04(OSLV: 0Hz 无传感器矢量控制) 或 05(V2: 传感器矢量控制) 时，启动频率 (b082) 会禁用，而且会输出指定频率。
 - 大于最大频率
 - MX2: 将频率基准限制为第一 / 第二最大频率 (A004/A204) 中设置的值。
 - RX: 将频率基准限制为第一 / 第二 / 第三最大频率 (A004/A204/A304) 中设置的值。
- 您可以使用变频器参数“输出频率设置/监控(F001)”来监控频率基准变量 SET-Freq 中设置的频率基准。



正确使用注意事项

DriveProgramming 程序停止时，程序停止前的频率基准变量数据会保留下来。重新执行程序时，进程会开始使用此前保留的数据。

加速 / 减速时间变量 ACCEL 和 DECEL

通过 DriveProgramming 功能直接控制频率基准的加速 / 减速时间时，请将变频器参数“加速 / 减速时间输入类型 (P031)”设置为 03(DriveProgramming)，以便启用加速时间变量 ACCEL 和减速时间变量 DECEL。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|-------|--------|------------------|------|--------|-----------|----|
| ACCEL | 加速时间变量 | MX2: 0 到 360,000 | 参数设定 | 0.01 秒 | 无符号 双字 | 读写 |
| DECEL | 减速时间变量 | RX: 1 到 360,000 | | | | |

- 加速时间变量 ACCEL 和减速时间变量 DECEL 属于无符号双字变量。
- 仅当将加速 / 减速时间输入类型 (P031) 设为 03(DriveProgramming) 时，加速时间变量 ACCEL 和减速时间变量 DECEL 才会启用。
- 仅当在接通电源后首次启动程序时，变频器加速/减速时间参数中设置的值才会设置给 DriveProgramming 的加速时间变量 ACCEL 和减速时间变量 DECEL。

MX2: 第一个 / 第二个加速时间 1(F002/F202) 和第一个 / 第二个减速时间 1(F003/F203) 的加速 / 减速时间会根据第一个 / 第二个控制选择进行设置。

RX: 第一个 / 第二个 / 第三个加速时间 1(F002/F202/F302) 和第一个 / 第二个 / 第三个减速时间 1(F003/F203/F303) 的加速 / 减速时间会根据第一个 / 第二个 / 第三个控制选择进行设置。

- 加速时间变量 ACCEL 和减速时间变量 DECEL 中设置的数据不会保存到 EEPROM 中。
如果希望保存该数据，请使用“eepwrt”命令，将其保存到用户参数变量或 EEPROM 中，以供变频器加速 / 减速时间参数使用。
- 加速 / 减速时间的内部处理在 40ms 周期内执行。即便在程序中更改了加速时间变量 ACCEL 或减速时间变量 DECEL 的值，最多也要花 40ms 的时间才能让更改反映在操作中。
- 在以下情况下，程序中设置的数据不会反映，而以前设置的数据会用于操作：为加速时间变量 ACCEL 或减速时间变量 DECEL 设置的数据超出范围，或为 3G3RX-V1 系列变频器设置了 0。



正确使用注意事项

DriveProgramming 程序停止时，程序停止前的加速时间变量 ACCEL 和减速时间变量 DECEL 会保留下来。重新执行程序时，进程会开始使用此前保留的数据。

5-5 变频器监控变量

您可以将变频器的内部监控功能和状态监控功能用作 DriveProgramming 功能的变量。

有关每个监控功能的详细信息，请参阅“MX2-V1 系列多功能小型变频器用户手册”（样本编号：I585）或“3G3RX-V1 系列高性能通用型变频器用户手册”（样本编号：I578）。

请注意，变频器使用的数据单位可能与 DriveProgramming 所用数据单位不同。请务必使用以下单位。

● MX2 和 RX

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|------------|------|---------|-----------|----|
| FM | 输出频率监控 [d001] | 0 到 40,000 | - | 0.01 Hz | 无符号 单字 | R |

使用此功能可监控变频器输出频率。监控的数据等同于输出频率监控 (d001) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|-----------|------|------|-----------|----|
| Iout | 输出电流监控 (d002) | 0 到 9,999 | - | 0.1% | 无符号 单字 | R |

使用此功能可监控变频器输出电流。监控的数据等同于输出电流监控 (d002) 的数据。该数据会以变频器额定电流的 0.1% 为增量显示。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|-------------------------|------|----|-----------|----|
| Dir | 运行方向监控 (d003) | 0: 停止 1: 正向 2: 反向 | - | - | 无符号 单字 | R |

使用此功能可监控变频器运转方向。监控的数据等同于运行方向监控 (d003) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|--------|------------------|---------------|------|-------|-----------|----|
| PID-FB | PID 反馈值监控 (d004) | 0 到 1,000,000 | - | 0.01% | 无符号 双字 | R |

使用此功能可监控 PID 反馈值。监控的数据等同于 PID 反馈值监控 (d004) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|-------|------------------------|---------------|------|------|-----------|----|
| F-CNV | 输出频率监控 (转换后) (d007) | 0 到 4,000,000 | - | 0.01 | 无符号 双字 | R |

使用此功能可监控转换后的输出频率。监控的数据等同于输出频率监控 (转换后) (d007) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|------------|------|----|-----------|----|
| Tmon | 输出转矩监控 (d012) | -200 到 200 | - | % | 无符号 单字 | R |

使用此功能可监控输出转矩。监控的数据等同于输出转矩监控 (d012) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|-----------|------|-------|-----------|----|
| Vout | 输出电压监控 (d013) | 0 到 6,000 | - | 0.1 V | 无符号 单字 | R |

使用此功能可监控输出电压。监控的数据等同于输出电压监控 (d013) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|-----------|------|-------|-----------|----|
| 电能 | 输入功率监控 (d014) | 0 到 9,999 | - | 0.1kW | 无符号 单字 | R |

使用此功能可监控输入功率。监控的数据等同于输入功率监控 (d014) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|----------|----------------|-------------|------|----|-----------|----|
| RUN-Time | 运行总时间监控 (d016) | 0 到 999,000 | - | 时间 | 无符号 双字 | R |

使用此功能可监控运行总时间。监控的数据等同于运行总时间监控 (d016) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|---------|----------------|-------------|------|----|-----------|----|
| ON-Time | 开机总时间监控 (d017) | 0 到 999,000 | - | 时间 | 无符号 双字 | R |

使用此功能可监控开机总时间。监控的数据等同于开机总时间监控 (d017) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|----------------------|-----------------------------|-----------------------------------|------|----|-----------|----|
| UMon(0) 到 UMon(2) | 用户监控 0 到 2 (d025 到 d027) | -2,147,483,647 到 2,147,483,647 | 0 | - | 带符号 双字 | 读写 |

DriveProgramming 的用户监控变量 UMon(0) 到 UMon(2) 属于带符号双字变量。如果在 DriveProgramming 程序中为用户监控变量 UMon(0) 到 UMon(2) 设置了任何数据, 这些数据会通过变频器用户监控参数 (d025、d026 和 d027) 显示。使用此功能可在外部显示程序计算状态。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|--|------|----|-----------|----|
| POS | 当前位置监控 (d030) | -268,435,455 到 268,435,455 (-1,073,741,823 到 1,073,741,823)* ¹ | - | 1 | 带符号 双字 | R |

使用此功能可监控当前位置。监控的数据等同于当前位置监控 (d030) 的数据。

*1. () 中显示的范围仅适用于 RX。

将 V2 控制模式选择 (P012) 设为 03(HAPR: 高分辨率绝对位置控制模式) 时会使用此显示范围。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|---------|--------------|------------|------|----|-----------|----|
| ERR-CNT | 故障计数器 (d080) | 0 到 65,535 | - | 次 | 无符号 单字 | R |

使用此功能可监控变频器的故障总数。监控的数据等同于故障计数器数据 (d080) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|--------------------|---------------------------|---------|------|----|-----------|----|
| ERR(1) 到 ERR(6) | 故障监控 1 到 6(d081 到 086) | 0 到 127 | - | - | 无符号 单字 | R |

使用此功能可监控变频器最近六次跳闸的数据。监控的数据等同于故障监控 1 到 6(d081 到 086) 的数据。此变量为只读变量。跳闸数据以代码形式显示, 与变频器 Modbus 通信中使用的变频器故障监控因素列表中使用的代码相同。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|------|---------------|------------|------|-------|-----------|----|
| DCV | 直流电压监控 (d102) | 0 到 10,000 | - | 0.1 V | 无符号 单字 | R |

使用此功能可监控变频器内部直流电压。监控的数据等同于直流电压监控 (d102) 的数据。此变量为只读变量。

| 功能变量 | 说明 | 数据范围 | 默认数据 | 单位 | 数据大小 | 读写 |
|--------|---------|------|------|----|-----------|----|
| STATUS | 变频器状态监控 | - | - | - | 无符号 单字 | R |

使用此功能可监控变频器状态。

每个位对应着以下状态。此变量为只读变量。

| 位 | 9 到 15 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|--------|-----|-----|-----|-----|-----|-----|-----|------|------|
| 功能 | 不使用 | 不使用 | 不使用 | 不使用 | 不使用 | 不使用 | 不使用 | 不使用 | 跳闸期间 | 运行期间 |

5-6 多功能输入变量

您可以通过 DriveProgramming 程序执行分配给多功能输入端子的功能。以下变量对应着可分配给多功能输入端子的功能。

如果任意变量设为 1(ON)，对应的功能会启用，就像多功能输入端子变为 ON 一样。

如果任意变量设为 0(OFF)，对应的功能会禁用。您可以通过程序执行这些功能，即便未将功能分配给多功能输入选择 (MX2: C001 到 C007/RX: C001 到 C008)。下表中的“参考”列显示了变频器的功能设置数据。有关每个功能的详细信息，请参阅“MX2-V1 系列多功能小型变频器用户手册”(样本编号: I585) 或“3G3RX-V1 系列高功能通用型变频器用户手册”(样本编号: I578)。

● 示例

如果将 FW(正向) 设为 1，则会执行正向运转命令。

FW: = 1 变频器开始正向运转。

FW: = 0 变频器停止正向运转并开始减速。



正确使用注意事项

- 如果 DriveProgramming 程序停止，多功能输入变量的数据不会保留下来，而是清零。
- 仅当变频器运行命令选择 (A002) 设为 01(控制电路端子台) 时，变量 FW(正向) 和 RV(反向) 才会启用。使用其他设置选项时，变频器不会运转。
- 即便在接通电源后立即将变量 FW(正向) 或 RV(反向) 设为 1，这些设置也会被忽略，而且变频器也不会执行正向或反向运转。先设为 0，然后再设为 1。为避免此操作，请创建一个程序，并使用等待命令在其中指定接通电源后等待 1 秒时间。
- 只有变量 AT(模拟输入切换) 需要设置多功能输入选择 (MX2: C001 到 C007/RX: C001 到 C008)。若要使用变量 AT(模拟输入切换)，请将多功能输入选择参数之一设为 16(AT: 模拟输入切换)。如果未设置，预计的操作不会执行，即便在程序中将变量 AT(模拟输入切换) 设为 1 也是如此。
- 多功能输入选择中设置的变频器多功能输入功能与多功能输入变量之间的关系为逻辑 OR。

● MX2 和 RX

| 功能变量 | 说明 | 读写 | 参考 |
|-------------------------|-----------------|----|------------------------------------|
| FW | 正向 | 读写 | MX2: C001 到 C007 = 00 RX: FW 端子 |
| RV | 反向 | 读写 | C001 到 C008 = 01 |
| CF1 到 CF4 | 多步速度设置二进制 1 到 4 | 读写 | C001 到 C008 = 02 到 05 |
| JG | 点动 | 读写 | C001 到 C008 = 06 |
| DB | 外部直流制动 | 读写 | C001 到 C008 = 07 |
| SET | 第二个控制 | 读写 | C001 到 C008 = 08 |
| TCH (2CH)* ¹ | 2-步加速 / 减速 | 读写 | C001 到 C008 = 09 |
| FRS | 自由运转停止 | 读写 | C001 到 C008 = 11 |
| EXT | 外部跳闸 | 读写 | C001 到 C008 = 12 |
| USP | 恢复供电时防止重新启动功能 | 读写 | C001 到 C008 = 13 |
| CS | 工业级切换 | 读写 | C001 到 C008 = 14 |
| SFT | 软件锁 | 读写 | C001 到 C008 = 15 |
| AT | (模拟输入切换) | 读写 | C001 到 C008 = 16 |
| RS | 复位 | 读写 | C001 到 C008 = 18 |
| STA | 3 线式启动 | 读写 | C001 到 C008 = 20 |
| STP | 3 线式停止 | 读写 | C001 到 C008 = 21 |
| F-R (F/R)* ¹ | 3 线式正向 / 反向 | 读写 | C001 到 C008 = 22 |
| PID | PID 禁用 | 读写 | C001 到 C008 = 23 |

| 功能变量 | 说明 | 读写 | 参考 |
|-----------|---------------|----|-----------------------|
| PIDC | PID 集成复位 | 读写 | C001 到 C008 = 24 |
| UP | 远程操作加速 | 读写 | C001 到 C008 = 27 |
| DWN | 远程操作减速 | 读写 | C001 到 C008 = 28 |
| UDC | 远程操作数据清除 | 读写 | C001 到 C008 = 29 |
| OPE | 强制操作员功能 | 读写 | C001 到 C008 = 31 |
| SF1 到 SF7 | 多步速度设置位 1 到 7 | 读写 | C001 到 C008 = 32 到 38 |
| OLR | 过载限制切换 | 读写 | C001 到 C008 = 39 |
| TL | 转矩限制启用 | 读写 | C001 到 C008 = 40 |
| TRQ1 和 2 | 转矩限制切换 1 和 2 | 读写 | C001 到 C008 = 41 和 42 |
| BOK | 制动确认 | 读写 | C001 到 C008 = 44 |
| LAC | LAD 取消 | 读写 | C001 到 C008 = 46 |
| PCLR | 当前位置清除 | 读写 | C001 到 C008 = 47 |
| ADD | 设置频率 A145 增加 | 读写 | C001 到 C008 = 50 |
| F-TM | 强制端子台 | 读写 | C001 到 C008 = 51 |
| ATR | 转矩基准输入允许 | 读写 | C001 到 C008 = 52 |
| KHC | 集成电源清除 | 读写 | C001 到 C008 = 53 |
| AHD | 模拟命令保持 | 读写 | C001 到 C008 = 65 |
| CP1 到 3 | 位置命令选择 1 到 3 | 读写 | C001 到 C008 = 66 到 68 |
| ORL | 原点搜索限制信号 | 读写 | C001 到 C008 = 69 |
| ORG | 原点搜索启动信号 | 读写 | C001 到 C008 = 70 |
| SPD | 速度 / 位置切换 | 读写 | C001 到 C008 = 73 |

*1. 多功能输入的功能选项缩写名称不同于功能变量名称。多功能输入的功能选项缩写名称在 () 中显示。

● 仅限 MX2

| 功能变量 | 说明 | 读写 | 参考 |
|---------------------------|-----------|----|------------------|
| RS485 (485)* ¹ | 启动联合变频器通信 | 读写 | C001 到 C007 = 81 |
| HLD | 保持输出频率 | 读写 | C001 到 C007 = 83 |
| ROK | 允许运行命令 | 读写 | C001 到 C007 = 84 |
| DISP | 显示固定 | 读写 | C001 到 C007 = 86 |

*1. 多功能输入的功能选项缩写名称不同于功能变量名称。多功能输入的功能选项缩写名称在 () 中显示。

● 仅限 RX

| 功能变量 | 说明 | 读写 | 参考 |
|------|-------------|----|------------------|
| SET3 | 第三控制 | 读写 | C001 到 C008 = 17 |
| CAS | 控制增益切换 | 读写 | C001 到 C008 = 26 |
| PPI | P/PI 切换 | 读写 | C001 到 C008 = 43 |
| ORT | 方位 | 读写 | C001 到 C008 = 45 |
| STAT | 脉冲串位置命令输入允许 | 读写 | C001 到 C008 = 48 |
| SON | Servo ON | 读写 | C001 到 C008 = 54 |
| FOC | 初步励磁 | 读写 | C001 到 C008 = 55 |
| FOT | 正向驱动停止 | 读写 | C001 到 C008 = 71 |
| ROT | 反向驱动停止 | 读写 | C001 到 C008 = 72 |
| PCNT | 脉冲计数器 | 读写 | C001 到 C008 = 74 |
| PCC | 脉冲计数器清除 | 读写 | C001 到 C008 = 75 |

5-7 多功能输出变量

您可以通过使用 DriveProgramming 程序执行可分配给多功能输出端子的功能。以下变量对应着可分配给多功能输出端子的功能。

将每个变量设为 1(ON) 或 0(OFF) 时，操作与多功能输出功能将多功能输出端子变为 ON/OFF 时的操作相同。即便未在多功能输出选择 (MX2: C021 和 C022/RX: C021 到 C025) 或多功能继电器输出 (MA、MB) 功能选择 (C026) 参数中设置功能，您也可以在程序上监控状态。下表中的“参考”列显示了变频器的功能设置数据。有关每个功能的详细信息，请参阅“MX2-V1 系列多功能小型变频器用户手册” (样本编号: I585) 或“3G3RX-V1 系列高功能通用型变频器用户手册” (样本编号: I578)。



正确使用注意事项

DriveProgramming 程序停止时，多功能输出变量的状态不会保留下来，但它们会根据实际功能状态更新。

● MX2 和 RX

| 功能变量 | 说明 | 读写 | 参考 |
|-------|--------------|----|------------------|
| RUN | 运行期间的信号 | R | C021 到 C026 = 00 |
| FA1 | 恒速到达信号 | R | C021 到 C026 = 01 |
| FA2 | 设置频率超出信号 | R | C021 到 C026 = 02 |
| OL | 过载警告 | R | C021 到 C026 = 03 |
| OD | 超出 PID 偏差 | R | C021 到 C026 = 04 |
| AL | 警报信号 | R | C021 到 C026 = 05 |
| FA3 | 只有设置频率信号 | R | C021 到 C026 = 06 |
| OTQ | 转矩过大 / 过低信号 | R | C021 到 C026 = 07 |
| UV | 电压不足期间的信号 | R | C021 到 C026 = 09 |
| TRQ | 转矩限制 | R | C021 到 C026 = 10 |
| RNT | 运行时间超出 | R | C021 到 C026 = 11 |
| ONT | 通电时间超出 | R | C021 到 C026 = 12 |
| THM | 电子元件过热警告 | R | C021 到 C026 = 13 |
| BRK | 制动释放 | R | C021 到 C026 = 19 |
| BER | 制动错误 | R | C021 到 C026 = 20 |
| ZS | 0Hz 检测信号 | R | C021 到 C026 = 21 |
| DSE | 超出速度偏差 | R | C021 到 C026 = 22 |
| POK | 位置就绪 | R | C021 到 C026 = 23 |
| FA4 | 设置频率超出信号 2 | R | C021 到 C026 = 24 |
| FA5 | 只有设置频率信号 2 | R | C021 到 C026 = 25 |
| OL2 | 过载警告 2 | R | C021 到 C026 = 26 |
| ODc | 模拟 FV 断开检测 | R | C021 到 C026 = 27 |
| OIDc | 模拟 FI 断开检测 | R | C021 到 C026 = 28 |
| FBV | PID 反馈比较信号 | R | C021 到 C026 = 31 |
| NDc | 通信断开检测 | R | C021 到 C026 = 32 |
| LOG1 | 逻辑运算输出 1 | R | C021 到 C026 = 33 |
| LOG 2 | 逻辑运算输出 2 | R | C021 到 C026 = 34 |
| LOG 3 | 逻辑运算输出 3 | R | C021 到 C026 = 35 |
| WAC | 电容器使用寿命警告信号 | R | C021 到 C026 = 39 |
| WAF | 冷却风扇使用寿命警告信号 | R | C021 到 C026 = 40 |
| FR | 启动接点信号 | R | C021 到 C026 = 41 |
| OHF | 散热片过热警告 | R | C021 到 C026 = 42 |

| 功能变量 | 说明 | 读写 | 参考 |
|------|----------|----|------------------|
| LOC | 低电流信号 | R | C021 到 C026 = 43 |
| IRDY | 操作就绪 | R | C021 到 C026 = 50 |
| FWR | 正向运转信号 | R | C021 到 C026 = 51 |
| RVR | 反向运转信号 | R | C021 到 C026 = 52 |
| MJA | 严重故障信号 | R | C021 到 C026 = 53 |
| WCO | 窗口比较器 FV | R | C021 到 C026 = 54 |
| WCOI | 窗口比较器 FI | R | C021 到 C026 = 55 |

● 仅限 MX2

| 功能变量 | 说明 | 读写 | 参考 |
|------|---------|----|------------------|
| FREF | 频率命令源 | R | C001 到 C007 = 58 |
| REF | 运行命令源 | R | C001 到 C007 = 59 |
| SETM | 电机 2 选择 | R | C001 到 C007 = 60 |
| EDM | 安全设备监控 | R | C001 到 C007 = 62 |

● 仅限 RX

| 功能变量 | 说明 | 读写 | 参考 |
|-------|------------|----|------------------|
| IP | 瞬时断电期间的信号 | R | C021 到 C026 = 08 |
| O2Dc | 模拟 FE 断开检测 | R | C021 到 C026 = 29 |
| LOG 4 | 逻辑运算输出 4 | R | C021 到 C026 = 36 |
| LOG 5 | 逻辑运算输出 5 | R | C021 到 C026 = 37 |
| LOG 6 | 逻辑运算输出 6 | R | C021 到 C026 = 38 |
| WCO2 | 窗口比较器 FE | R | C021 到 C026 = 56 |

6

DriveProgramming 命令

本章节介绍适用于 DriveProgramming 的命令。

| | |
|-----------------------|------|
| 6-1 命令类别 | 6-2 |
| 6-2 命令格式 | 6-3 |
| 6-3 命令列表 | 6-4 |
| 6-4 程序控制命令 | 6-10 |
| 6-5 四则运算和逻辑运算命令 | 6-23 |
| 6-6 I/O 控制单元 | 6-36 |
| 6-7 定时器控制命令 | 6-46 |
| 6-8 参数控制命令 | 6-52 |
| 6-9 变频器控制命令 | 6-58 |

6-1 命令类别

命令分为以下几类。

- 程序控制命令
- 四则运算和逻辑运算命令
- I/O 控制命令
- 定时器控制命令
- 参数控制命令
- 变频器控制命令

6-2 命令格式

每种命令都由命令及其自变量 (0 到 5 个) 构成。
现举例如下。

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|-------|-------|-------|-------|-------|-------|---------|
| entry | | | | | | 表示程序开始。 |

如果没有自变量，则单元格为空。

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|------|----------|-------|-------|-------|-------|-----------------------------------|
| wait | < 等待时间 > | | | | | < 等待时间 > 等待 (等待时间 0.01) 秒。 |

“wait”命令有一个自变量。在此自变量中，请输入等待时间 (变量或常数)。

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|-------|---------|-------|-------|-------|-------|---------------------------|
| while | < 条件 > | | | | | 符合 < 条件 > 时执行 < 命令集 >。 |
| | < 命令集 > | | | | | 符合 < 条件 > 时执行的 命令集。 |
| wend | | | | | | 转至 “while” (循环)。 |

对于 “while” 命令，请在自变量中设置一个条件。(详情请参考第 6-6 页上的条件。)

在 “while” 和 “wend” 之间，写入符合 < 条件 > 时执行的命令集。

6-3 命令列表

程序控制命令

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|--------------|---------|-------|-------|-------|-------|--|
| entry | | | | | | 表示任务开始。 |
| end | | | | | | 表示任务结束。 |
| call | <子程序> | | | | | 跳转到<子程序>。 |
| sub | <子程序> | | | | | 表示子程序开始。 |
| end sub | | | | | | 表示子程序结束。 |
| goto | <标签> | | | | | 无条件跳至<标签>。 |
| on trip goto | <标签> | | | | | 跳闸时跳至<标签>。 |
| if | <条件> | | | goto | <标签> | 符合条件时跳至<标签>。 |
| if | <条件> | | | then | | “if”结构开始。 如果符合<条件>, 它会执行此命令后、“else”命令前的<命令集 1>, 然后跳至“endif”后的下一步。 如果不符合<条件>, 它会执行“else”命令后、“endif”命令前的<命令集 2>, 然后跳至“endif”后的下一步。 |
| | <命令集 1> | | | | | 符合<条件>时执行的命令集。 |
| else | | | | | | 开始不符合<条件>时的命令。 |
| | <命令集 2> | | | | | 不符合<条件>时执行的命令集。 |
| endif | | | | | | “if”结构结束。 |

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|------------|---------------------------|---------|---------|---------|-------|--|
| select | < 条件变量 > | | | | | < 条件变量 > 等于 < 条件值 > 时执行 “case” 命令后的命令。 |
| case | < 条件值 1 > | | | | | 开始分配给条件值的命令。 |
| | < 命令集 1 > | | | | | 分配给 < 条件值 1 > 的命令集 |
| | [case 条件值 n] < 命令集 n > | | | | | 您可以使用 “case” 设置一个或多个条件。 |
| case else | | | | | | 开始分配给条件值以外值的命令。 |
| | < 命令集 > | | | | | 分配给 < 条件值 > 以外值的命令集 |
| end select | | | | | | 结束 “case” 语法。 |
| for | < 变量 > | < 开始值 > | < 结束值 > | < 增量值 > | | 开始一个按 < 增量值 > 从 < 开始值 > 递增至 < 结束值 > 的循环。 |
| | < 命令集 > | | | | | 反复执行的命令集 |
| next | | | | | | 跳至 “for” (循环)。 |
| while | < 条件 > | | | | | 符合 < 条件 > 时执行 < 命令集 >。 |
| | < 命令集 > | | | | | 符合 < 条件 > 时执行的命令集。 |
| wend | | | | | | 跳至 “while” (循环)。 |
| until | < 条件 > | | | | | 执行 < 命令集 > 直至符合 < 条件 >。 |
| | < 命令集 > | | | | | 不符合 < 条件 > 时执行的命令集。 |
| loop | | | | | | 跳至 “until” (循环)。 |
| wait | < 等待时间 > | | | | | < 等待时间 > 等待 (等待时间 0.01) 秒。 |
| wait | < 条件 > | | | | | 等待下去, 直至满足 < 条件 >。 |



附加信息

对于 < 条件 > 格式, 请参阅第 6-6 页上的条件。

条件

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|-----------|-----------------|-------|-----------------|-------|-------|---|
| Condition | < 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 > 等于或 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 > 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 > 等于或 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 > 不等于 < 变量 2/ 常数 > 时为 TRUE。 |

四则运算和逻辑运算命令

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|------------------------------|----------|--------|-----------------|-------|-----------------|--|
| Four arithmetic operation | < 变量 1 > | := | < 变量 2/ 常数 > | | | 将 < 变量 2/ 常数 > 分 配给 < 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 >，然 后将结果分配给 < 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 >，然 后将结果分配给 < 变量 1 >。 |
| | < 变量 1 > | := | < 变量 2/ 常数 > | * | < 变量 3/ 常数 > | 将 < 变量 2/ 常数 > 乘 上 < 变量 3/ 常数 >，然 后将结果分配给 < 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 >，然 后将结果分配给 < 变量 1 >。 |
| Remainder for division | < 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 >，然 后将余数分配给 < 变量 1 >。 |
| Absolute value | < 变量 1 > | := abs | < 变量 2/ 常数 > | | | 将 < 变量 2/ 常数 > 的绝 对值分配给 < 变量 1 >。 |

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|-------------------|----------|--------|---------------|-------|---------------|--|
| Logical operation | < 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 > 执行逻辑与运算, 然后将结果分配给 < 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 > 执行逻辑或运算, 然后将结果分配给 < 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 > 执行逻辑异或运算, 然后将结果分配给 < 变量 1 >。 |
| | < 变量 1 > | := not | < 变量 2 / 常数 > | | | 反转 < 变量 2 / 常数 > 的位, 然后将结果分配给 < 变量 1 >。 |

增量和减量命令

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|-----------|-------|--------|-------|-------|-------|--------------|
| Increment | inc | < 变量 > | | | | 将 < 变量 > 加一。 |
| Decrement | Dec | < 变量 > | | | | 将 < 变量 > 减一。 |

I/O 控制命令

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|---|-------------|-------|-------------|-------|-------|--|
| Input terminal variable assignment | < 变量 > | := | X(i) | | | 将输入端子变量信息分配给 < 变量 >。 0 = off, 1 = on |
| | < 变量 > | := | Xw | | | 以字为单位将输入端子变量信息分配给 < 变量 >。 |
| Output terminal variable output | Y(**) | := | < 变量 / 常数 > | | | 以位为单位将数据输出到输出端子变量。 0 = off, 1 = on |
| | Yw | := | < 变量 / 常数 > | | | 以字为单位将数据输出到输出端子变量。 |
| Multi-function input variable output | < 多功能输入变量 > | := | < 变量 / 常数 > | | | 将数据输出到多功能输入变量。 0 = off, 1 = on |
| Multi-function output variable assignment | < 变量 > | := | < 多功能输出变量 > | | | 将多功能输出变量状态分配给 < 变量 >。 0 = off, 1 = on |

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|-------------------------------|--------|-------|-------------|-------|-------|---|
| Internal user contact control | < 变量 > | := | UB(i) | | | 将内部用户接点信息分配给 < 变量 >。 0 = off, 1 = on |
| | < 变量 > | := | UBw | | | 以字为单位将内部用户接点信息分配给 < 变量 >。 |
| | UB(**) | := | < 变量 / 常数 > | | | 以位为单位将数据输出到内部用户接点。 0 = off, 1 = on |
| | UBw | := | < 变量 / 常数 > | | | 以字为单位将数据输出到内部用户接点。 |

定时器控制命令

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|-----------|--------|-------------|-------|---------------|-------|---|
| delay | on/off | < 变量 1 > | TD(i) | < 变量 2 / 常数 > | | 为指定定时器设置 < 变量 2 / 常数 >, 并启动定时器。 在 < 变量 2 / 常数 > 时间后, 将 < 变量 1 > 更改为 ON 或 OFF。 同时, 它会将 TD(i) 更改为 ON。 |
| timer set | TD(*) | < 变量 / 常数 > | | | | 为指定定时器设置 < 变量 / 常数 >, 并启动定时器。 在 < 变量 / 常数 > 时间后, 将 TD(i) 更改为 ON。 |
| timer off | TD(*) | | | | | 停止指定定时器。 |

参数控制命令

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|----------------------|--------|--------|-------------|-------|-------|---|
| ChgParam | < 参数 > | := | < 变量 / 常数 > | | | 将 < 参数 > 的内容替换为 < 变量 / 常数 >。 |
| MonParam | < 变量 > | := | < 参数 > | | | 将 < 参数 > 的内容分配给 < 变量 >。 |
| eepwrt | | | | | | 执行以下参数写入命令时 (仅限一次), 请求写入到变频器 EEPROM。 |
| ChgParam | < 参数 > | := | < 变量 / 常数 > | | | 将 < 参数 > 内容替换为 < 变量 / 常数 >, 同时将其保存到 EEPROM。 |
| rtcset ^{*1} | on/off | < 变量 > | | | | 将 LCD 数字操作器的时钟信息分配给 < 变量 >。 |


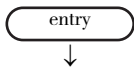

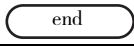

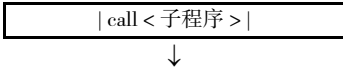
*1. 只有 3G3RX-V1 系列变频器可使用 “rtcset” 命令。

3G3MX2-ZV1 系列变频器无法使用 “rtcset” 命令, 原因是它们并未与 LCD 数字操作器 (型号: 3G3AX-OP05) 连接。

变频器控制命令

| 命令 | 自变量 1 | 自变量 2 | 自变量 3 | 自变量 4 | 自变量 5 | 说明 |
|----------|-------------|-------------|-------|-------|-------|--|
| FW | := | 1 | | | | 按正向操作变频器。 |
| RV | := | 1 | | | | 按反向操作变频器。 |
| stop | | | | | | 将变频器减速至停止。 注: 变频器处于跳闸状态时, 此命令起着复位的作用。 |
| trip | < 变量 / 常数 > | | | | | 根据 < 变量 / 常数 > 发出用户跳闸。 |
| SET-Freq | := | < 变量 / 常数 > | | | | 将 < 变量 / 常数 > (0.01Hz) 分配给变频器频率基准变量。 |
| ACCEL | := | < 变量 / 常数 > | | | | 将 < 变量常数 > (10ms) 分配给变频器加速时间变量。 |
| DECEL | := | < 变量 / 常数 > | | | | 将 < 变量 / 常数 > (10ms) 分配给变频器减速时间变量。 |


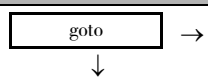
6-4 程序控制命令

| Entry | | |
|---|--------------|----------------------|
| 命令 | 说明 | 自变量 |
|  | 表示任务开始。 | --- |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | entry | |
| 注 必须将此命令放在每个任务的开头。 | | |
| End | | |
| 命令 | 说明 | 自变量 |
|  | 表示任务结束。 | --- |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | end | |
| 注 必须将此命令放在每个任务的结尾。 | | |
| Call | | |
| 命令 | 说明 | 自变量 |
|  | 跳转到 < 子程序 >。 | 子程序：子程序由您定义的名称或别名标识。 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | call < 子程序 > | |
| 注 1. 执行完子程序后会执行“call”命令后的下一个命令行。 | | |
| 2. 若要为流程图中的“call”命令设置子程序名称，请右键单击该命令，然后从菜单中选择 [Set Subroutine] (设置子程序)，接着从显示的选项中选择。 | | |

| Sub | | |
|---|----------|-----|
| 命令 | 说明 | 自变量 |
|  | 表示子程序开始。 | --- |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | sub | |
| 注 必须将此命令放在每个子程序的开头。 | | |
| End sub | | |
| 命令 | 说明 | 自变量 |
|  | 表示子程序结束。 | --- |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | endsub | |
| 注 必须将此命令放在每个子程序的结尾。 | | |

● 示例

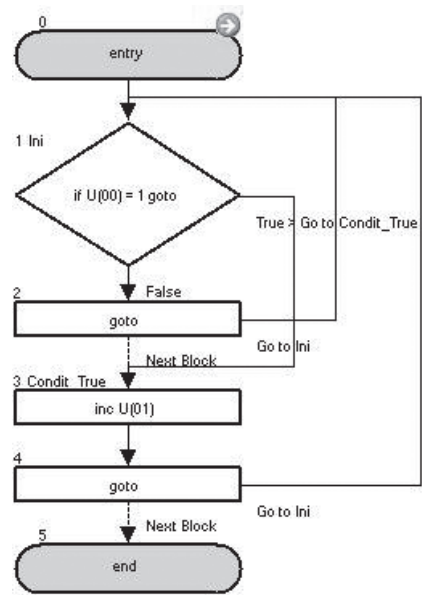
| 流程图 | 文本 |
|--|--|
| Main | Main |
| <pre> graph TD 0([entry]) --> 1[SET-Freq := 6000] 1 --> 2[UB(0) := 1] 2 --> 3{if UB(0) = 1 then} 3 -- True --> 4[call RunFW] 3 -- False --> 5[else] 5 --> 6[call RunRV] 6 --> 7[endif] 7 --> 8[stop] 8 --> 9[wait RUN = 0] 9 --> 10[UB(0) := not UB(0)] 10 --> 11[goto] 11 -- Go to Loop_ --> 3 11 -- Next Block --> 12([end]) </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> |
| Main: RunFW | Main: RunFW |
| <pre> graph TD 13([sub]) --> 14[FW := 1] 14 --> 15[wait X(01) = 1] 15 --> 16([endsub]) </pre> | Main: RunRV |
| Main: RunRV | Main: RunRV |
| <pre> graph TD 17([sub]) --> 18[RV := 1] 18 --> 19[wait X(02) = 1] 19 --> 20([endsub]) </pre> | |
| 块编号 | 操作 |
| 1 和 2 | 将变频器输出频率设为 60.00Hz，并将内部用户接点 UB(0) 更改为 ON。 |
| 3 到 7 | 如果 UB(0) 为 ON，它会执行子程序 RunFW。如果 UB(0) 为 OFF，它会执行子程序 RunRV。 |
| 8 到 11 | 执行完子程序后，它会停止变频器，等待 RUN 更改为 OFF，反转 UB(0)，然后返回块 3。 |
| 13 到 16 | 按正向操作变频器，然后在“wait”命令的作用下等待，直至 X(01) 更改为 ON。(子程序: RunFW) |
| 17 到 20 | 按反向操作变频器，然后在“wait”命令的作用下等待，直至 X(02) 更改为 ON。(子程序: RunRV) |

| Go To | | |
|---|---------------|---------------------|
| 命令 | 说明 | 自变量 |
|  | 无条件跳至 < 标签 >。 | 标签：用来标识任务中特定功能块的名称。 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | goto < 标签 > | |

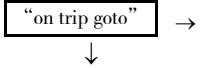
注 1. 此命令必须连接至接下来要执行的命令。这是声明程序流程所必需的。

2. 若要在流程图中设置标签名称, 请右键单击命令, 然后从菜单中选择 [Set Label](设置标签), 接着您可以指定任意名称。

● 示例

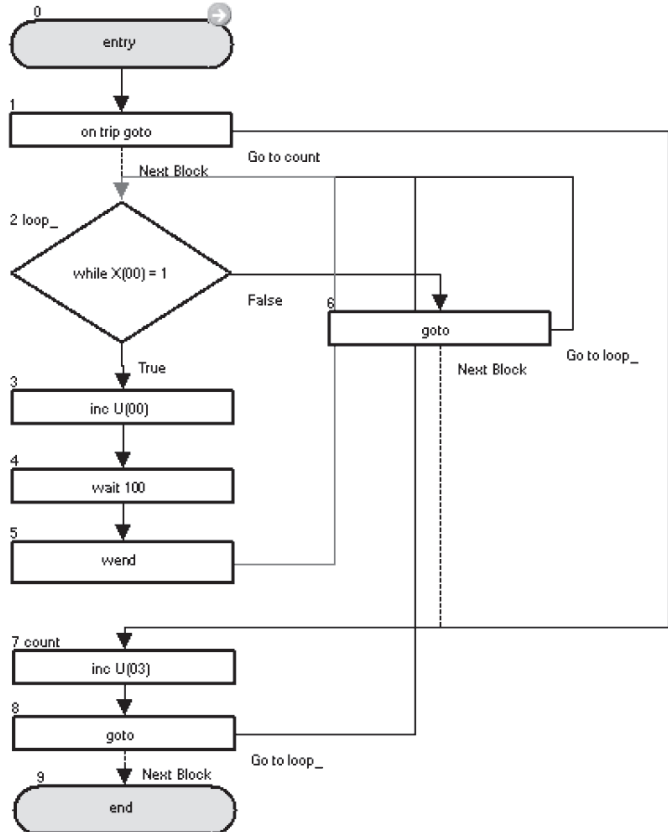
| 流程图 | 文本 |
|--|--|
|  | <pre> entry :Ini if U(00) = 1 goto Condit_True goto Ini :Condit_True inc U(01) goto Ini end </pre> |

| 块编号 | 操作 |
|-----|--|
| 1 | 如果 U(00) 为 1, 它会跳至块 3: Condit_True。如果 U(00) 为 0, 它会跳至下一个块 2。 |
| 2 | 无条件跳至块 1: Ini。 |
| 3 | 将 U(01) 加一。 |
| 4 | 无条件跳至块 1: Ini。 |


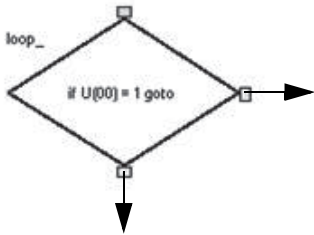
| On Trip | | |
|---|---------------------|---------------------|
| 命令 | 说明 | 自变量 |
|  | 变频器中跳闸时跳至 < 标签 >。 | 标签：用来标识任务中特定功能块的名称。 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | on trip goto < 标签 > | |

- 注 1. “on trip goto” 命令会在变频器中出现跳闸时处理程序分支。
 2. 执行 “on trip goto” 命令时，执行日志会保存在变频器中，而且程序会检查每个扫描的跳闸状态。程序在跳闸后达到下一步时，它会跳至 “goto” 命令的目标。未出现跳闸则不会执行任何操作。
 3. 如果执行了任务中的 “end” 命令，则会取消 “on trip goto” 命令，此时即便出现跳闸也不会处理程序分支。
 4. 程序因执行 “on trip goto” 命令出现分支时，它在任何情况下都不会再次出现分支。程序因 “on trip goto” 命令出现分支后，请务必重新启动程序。
 5. 若要在流程图中设置标签名称，请右键单击命令，然后从菜单中选择 [Set Label](设置标签)，接着您可以指定任意名称。

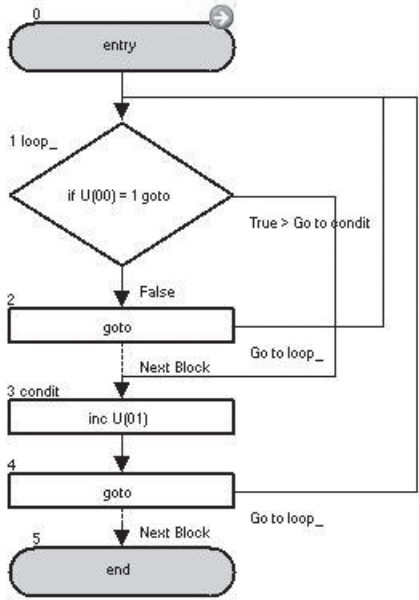
● 示例

| 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> |





| 块编号 | 操作 |
|-------|---|
| 1 | 执行 “on trip goto” 命令一次，确保程序在出现跳闸时跳至块 7: count。 |
| 2 到 6 | 如果 X(00) 为 1，它会将 U(00) 加一，并在 “wait” 命令的作用下等待 1.00 秒，然后返回块 2: loop_。如果 X(00) 为 0，则无条件跳至块 2: loop_。 |
| 7 到 8 | 将 U(03) 加一，并无条件跳至块 2: loop_。 |

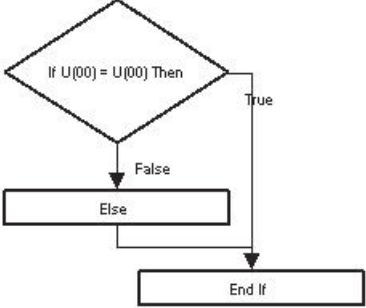
| If | | |
|---|-----------------------|---|
| 命令 | 说明 | 自变量 |
|  | 符合 < 条件 > 时跳至 < 标签 >。 | 条件: 根据 < 左侧数值 > < 比较运算符 > < 右侧数值 > 的格式比较两个变量或常数。 • 左侧数值: 任何变量或常数 (范围 -128 到 127) • 比较运算符: =、<、>、<=、>= 或 <> • 右侧数值: 任何变量或常数 (范围 -128 到 127) 标签: 用来标识任务中特定功能块的名称。 |
| 格式 | | |
| 流程图法 | 文本语法 | |
|  | if < 条件 > goto < 标签 > | |
| 注 若要在流程图中设置标签名称, 请右键单击命令, 然后从菜单中选择 [Set Label](设置标签), 接着您可以指定任意名称。 | | |

● 示例

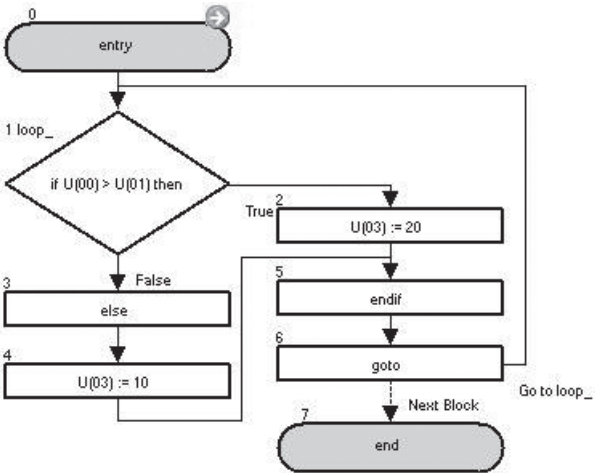
| 流程图 | 文本 |
|---|--|
|  | <pre> entry :loop_ if U(00) = 1 goto condit goto loop_ :condit inc U(01) goto loop_ end </pre> |

| 块编号 | 操作 |
|-------|--|
| 1 | 如果 U(00) 为 1, 它会跳至块 3: condit。如果 U(00) 不为 1, 它会跳至下一个块 2。 |
| 2 | 无条件跳至块 1: loop_。 |
| 3 到 4 | 将 U(01) 加一, 并无条件跳至块 1: loop_。 |





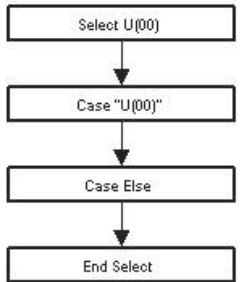
| Ifs/Else/End If | | |
|---|--|---|
| 命令 | 说明 | 自变量 |
|  | 如果符合 <条件>, 它会执行此命令后、“else”命令前的 <命令集 1>, 然后跳至 “endif” 后的下一步。 如果不符合 <条件>, 它会执行 “else” 命令后、“endif” 命令前的 <命令集 2>, 然后跳至 “endif” 后的下一步。 | 条件: 根据 <左侧数值> <比较运算符> <右侧数值> 的格式比较两个变量或常数。 • 左侧数值: 任何变量或常数 (范围 -128 到 127) • 比较运算符: =、<、>、<=、>= 或 <> • 右侧数值: 任何变量或常数 (范围 -128 到 127) 命令集 1: “else” 命令前的一个或多个命令。它可以包含嵌套命令 (最多八层)。 命令集 2: “endif” 命令前的一个或多个命令。它可以包含嵌套命令 (最多八层)。 |
|  | | |
|  | | |
|  | | |

| 格式 | |
|---|--|
| 流程图法 | 文本语法法 |
|  <pre> graph TD A{If U(00) = U(00) Then} -- True --> C[End If] A -- False --> B[Else] B --> C </pre> | <pre> if <条件> then <命令集 1> else <命令集 2> endif </pre> |

● 示例

| 流程图 | 文本 |
|---|--|
|  <pre> graph TD 0([entry]) --> 1{if U(00) > U(01) then} 1 -- True --> 2[U(03) := 20] 1 -- False --> 3[else] 3 --> 4[U(03) := 10] 2 --> 5[endif] 4 --> 5 5 --> 6[goto] 6 -- Go to loop_ --> 1 6 -- Next Block --> 7([end]) </pre> | <pre> entry :loop_ if U(00) > U(01) then U(03) := 20 else U(03) := 10 endif goto loop_ end </pre> |

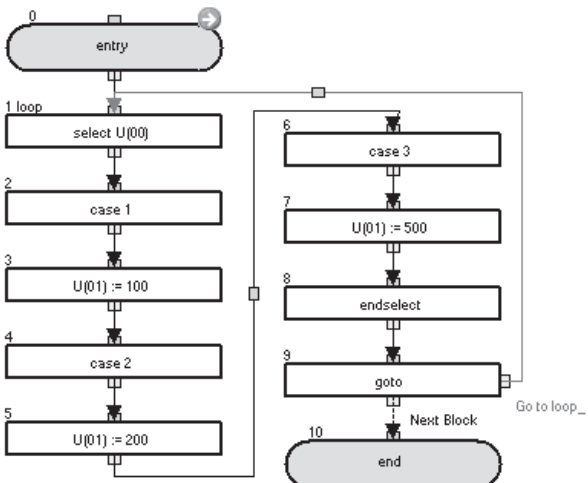
| 块编号 | 操作 |
|-------|---|
| 1 到 5 | 如果 U(00) 大于 U(01), 它会在 “if” 后将 20 分配给 U(03), 然后跳至 “endif” 命令。如果 U(00) 不大于 U(01), 它会在 “else” 命令后将 10 分配给 U(03), 然后跳至 “endif” 命令。 |
| 6 | 无条件跳至块 1: loop_。 |

| Select/Case/Case Else/EndSelect | | |
|---|--|---|
| 命令 | 说明 | 自变量 |
|  Select | <p>允许根据变量值执行多个程序段。在 < 条件变量 > 与任何 “case” 命令的 < 条件值 n> 匹配时执行 < 命令集 n>。</p> <p>在 < 条件变量 > 与 “case” 命令的任何条件值都不匹配时执行 < 其他命令集 >(case else)。</p> <p>如果需要根据 < 条件变量 > 在多个选项之间进行选择，此命令十分有用。</p> <p>与调用子程序的 “call” 命令配合使用时，此命令也可用于组织程序。</p> | <p>条件变量：使用的条件变量</p> <p>条件值 n：任意变量或常数 (范围 -128 到 127)</p> <p>命令集 n：下一个 “case” 或 “endselect” 命令前的一个或多个命令。它可以包含嵌套命令 (最多八层)。</p> |
|  Case | | |
|  CaseElse | | |
|  EndSelect | | |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | <pre>select < 条件变量 > case < 条件值 1> < 命令集 1> case < 条件值 n> < 命令集 n> case else < 其他命令集 > endselect</pre> | |




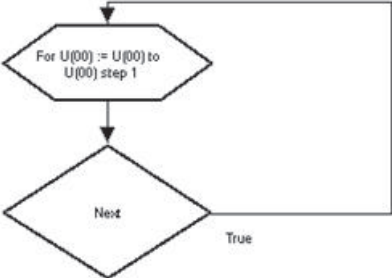
注 “select” 计为嵌套层 1。

子程序中使用的一组 “select” 和 “endselect” 计为嵌套层 2。

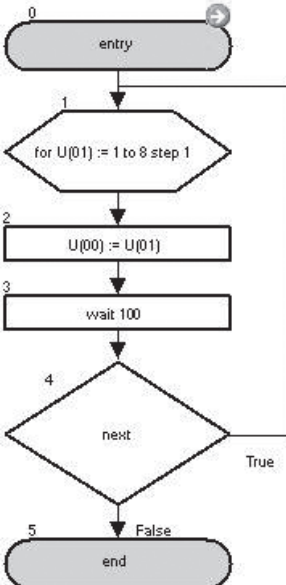
● 示例

| 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> |




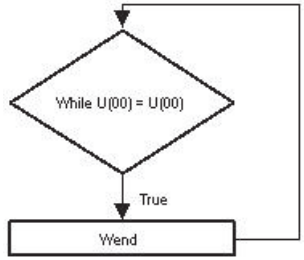
| 块编号 | 操作 |
|-------|---|
| 1 到 8 | 如果 U(00) 为 1，将 100 分配给 U(01)；如果 U(00) 为 2，将 200 分配给 U(01)。在其他情况下，将 500 分配给 U(01)，而且程序跳至 “endselect” 命令。 |
| 9 | 无条件跳至块 1: loop_。 |

| For/Next | | |
|--|---|---|
| 命令 | 说明 | 自变量 |
|  For ...  For  Next | 反复执行 <命令集>，直至有 <起始值> 的 <变量> 达到 <结束值>。每次执行时，<增量值> 都会加到 <变量> 上。执行下一个命令时，程序会判断 <结束值>，并将 <增量值> 加到 <变量> 上。 | 变量： 任意变量 起始值： 初始值。这是在第一个循环中分配给变量的值。 结束值： 这是退出循环的值。 增量值： 在每个循环中，变量都会加上此值。 命令集： “next”命令前的一个或多个命令。它可以包含嵌套命令（最多八层）。 |
| 格式 | | |
| 流程图法 | 文本语法 | |
|  | for <变量> := <起始值> to <结束值> step <增量值> <命令集> next | |

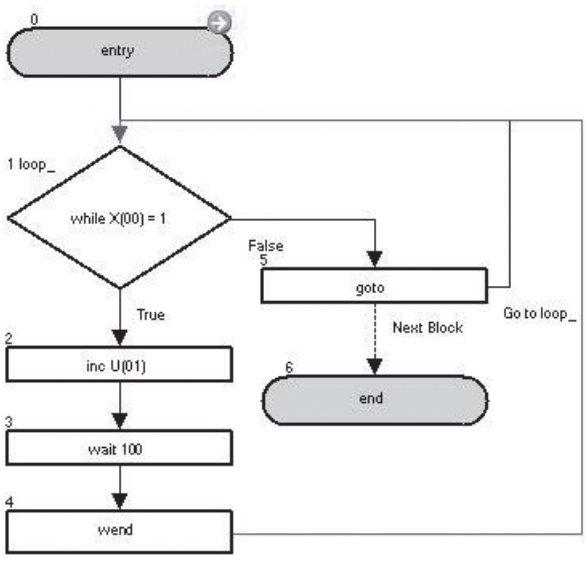
● 示例




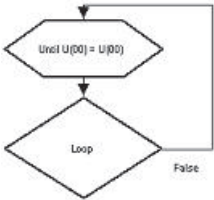
| 流程图 | 文本 |
|---|--|
|  | <pre> entry for U(01) := 1 to 8 step 1 U(00) := U(01) wait 100 next end </pre> |

| 块编号 | 操作 |
|-------|---|
| 1 到 4 | 将 1 分配给 U(01)，并将 U(01) 的值分配给 U(00)。在“wait”命令的作用下等待 1.00 秒。如果 U(01) 小于 8，则将 U(01) 加一，然后执行“for”后的下一个命令。如果 U(01) 大于或等于 8，则转至“next”命令后的下一步。 |

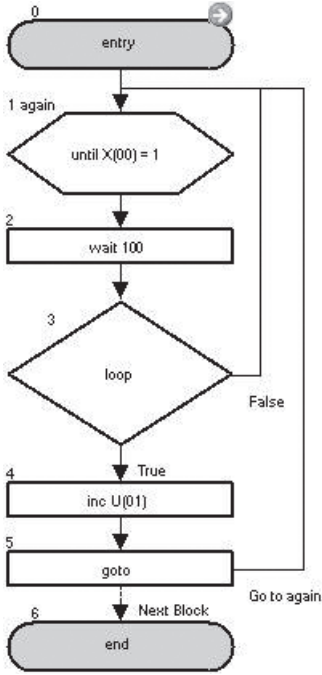
| While/Wend | | |
|---|------------------|--|
| 命令 | 说明 | 自变量 |
|  While ...  While  Wend | 符合条件时执行 < 命令集 >。 | 条件: 根据 < 左侧数值 > < 比较运算符 > < 右侧数值 > 的格式比较两个变量或常数。 • 左侧数值: 任何变量或常数 (范围 -128 到 127) • 比较运算符: =、<、>、<=、>= 或 <> • 右侧数值: 任何变量或常数 (范围 -128 到 127) 命令集: “wend” 命令前的一个或多个命令。它可以包含嵌套命令 (最多八层)。 |
| 格式 | | |
| 流程图法 | | 文本语言法 |
|  | | <pre>while < 条件 > < 命令集 > wend</pre> |

● 示例



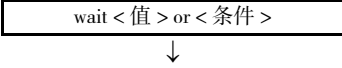
| 流程图 | | 文本 |
|---|--|--|
|  | | <pre>entry :loop_ while X(00) = 1 inc U(01) wait 100 wend goto loop_ end</pre> |
| 块编号 | 操作 | |
| 1 到 4 | 如果 X(00) 为 1, 它会将 U(01) 加一, 并在 “wait” 命令的作用下等待 1.00 秒, 然后跳至 “while” 命令。如果 X(00) 不为 1, 它会执行 “wend” 命令后的下一个命令。 | |
| 5 | 无条件跳至块 1: loop_。 | |

| Until/Loop | | |
|---|---------------------------------|--|
| 命令 | 说明 | 自变量 |
|  Until ...  Until  Loop | 执行 < 命令集 > 直至符合 < 条件 >。 | 条件: 根据 < 左侧数值 > < 比较运算符 > < 右侧数值 > 的格式比较两个变量或常数。 • 左侧数值: 任何变量或常数 (范围 -128 到 127) • 比较运算符: =、<、>、<=、>= 或 <> • 右侧数值: 任何变量或常数 (范围 -128 到 127) 命令集: “loop” 命令前的一个或多个命令。它可以包含嵌套命令 (最多八层)。 |
| 格式 | | |
| 流程图法 | 文本语法 | |
|  | until < 条件 > < 命令集 > loop | |

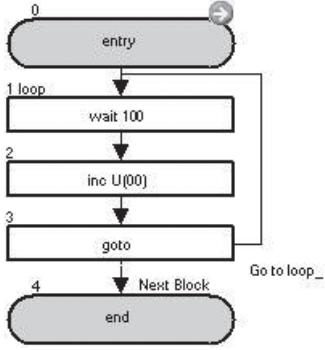
● 示例

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

| 块编号 | 操作 |
|-------|--|
| 1 到 3 | 如果 X(00) 不为 1, 在 “wait” 命令的作用下等待 1.00 秒, 然后跳至 “until” 命令。如果 X(00) 为 1, 它会执行 “loop” 命令, 并转至下一个命令。 |
| 4 到 5 | 将 U(00) 加一, 并无条件跳至块 1: again。 |

| Wait | | |
|--|----------------------|--|
| 命令 | 说明 | 自变量 |
|  WaitTime  WaitCond | 让程序等待指定的秒数或直至符合某个条件。 | 值: 任何变量或常数 (指定的时间 10 ms) 等待时间值 (0 到 32767 10 ms) 条件: 根据 <左侧数值><比较运算符><右侧数值> 的格式比较两个变量或常数。 • 左侧数值: 任何变量或常数 (范围 0 到 127) • 比较运算符: =、<、>、<=、>= 或 <> • 右侧数值: 任何变量或常数 (范围 0 到 127) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | wait < 值 > or < 条件 > | |
| <p>注 1. 使用“wait”命令时, 程序不会转至下一个命令, 而是等待设置的等待时间, 或等待符合某个条件。因此, 在“wait”命令的等待时间内, 任务处理不会执行。如果在此期间需要执行定期监控, 请将其分开, 并分配给另一个任务, 从而确保它会定期执行, 而不使用“wait”命令。</p> <p>2. 请勿为“wait”命令中使用的常数设置一个负值。 否则, 程序将无法检查等待时间是否过去或是否符合条件, 而且是始终停在“wait”命令处, 而不执行任何操作。</p> <p>3. “wait”命令并非精确的时间计量方法。若要精确计量时间, 请使用内部定时器或 LCD 数字操作器的时钟功能。</p> | | |

- 使用 < 值 > 设置了等待时间的程序示例: 程序会在“wait”命令处等待 < 值 > 中设置的时间。


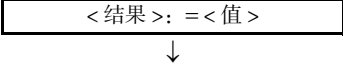
| 流程图 | 文本 |
|---|---|
|  | <pre> entry :loop_ wait 100 inc U(00) goto loop_ end </pre> |

| 块编号 | 操作 |
|-------|--|
| 1 | 在“wait”命令的作用下等待 1.00 秒。 |
| 2 到 3 | 将 U(00) 加一, 并无条件跳至 1: loop_, 再等待 1 秒钟。 |

- 在 < 条件 > 中设置了退出等待状态的条件的程序示例：程序会等待下去，直至符合条件。

| 流程图 | | 文本 |
|---|------------------------------|---|
| <pre> graph TD 0([0 entry]) --> 1[1 loop wait X(00) = 1] 1 --> 2[2 inc U(00)] 2 --> 3[3 goto] 3 -- Go to loop_ --> 1 3 -- Next Block --> 4([4 end]) </pre> | | <pre> entry :loop_ wait X(00) = 1 inc U(00) goto loop_ end </pre> |
| 块编号 | 操作 | |
| 1 | 等待下去，直至 X(00) 变为 1。 | |
| 2 到 3 | 将 U(00) 加一，并无条件跳至块 1: loop_。 | |

6-5 四则运算和逻辑运算命令


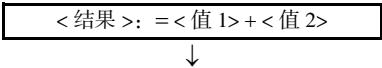
| =(替换) | | |
|---|---------------------|---|
| 命令 | 说明 | 自变量 |
|  | 将 < 值 > 分配给 < 结果 >。 | 结果: 任意变量 值: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | <结果> := <值> | |

注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。

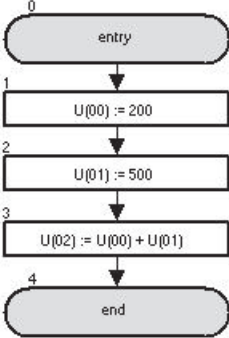
● 示例


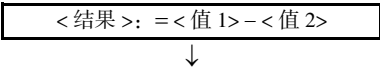
| 流程图 | 文本 |
|--|--|
| <pre> graph TD 0([entry]) --> 1[U(00) := 200] 1 --> 2[U(01) := U(00)] 2 --> 3[U(02) := U(00) + U(01)] 3 --> 4[U(03) := U(00) - 100] 4 --> 5[U(04) := U(03) * 2] 5 --> 6[U(05) := U(00) / U(03)] 6 --> 7[U(06) := 17 mod U(05)] 7 --> 8[U(07) := abs UL(07)] 8 --> 9([end]) </pre> | <pre> 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> |

| 块编号 | 操作 |
|-----|-------------------------------|
| 1 | 将 200 分配给 U(00)。 |
| 2 | 将 U(00) 的值分配给 U(01)。 |
| 3 | 将 U(00) 和 U(01) 之和分配给 U(02)。 |
| 4 | 将 U(00) 和 100 之差分配给 U(03)。 |
| 5 | 将 U(03) 和 2 的乘积分配给 U(04)。 |
| 6 | 将 U(00) 和 U(03) 之商分配给 U(05)。 |
| 7 | 将 U(05) 除以 17 之后的余数分配给 U(06)。 |
| 8 | 将 UL(07) 的绝对值分配给 U(07)。 |

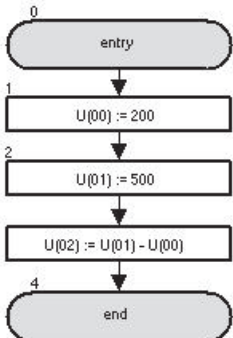
| +(加) | | |
|---|-----------------------|--|
| 命令 | 说明 | 自变量 |
|  | 将 <值 1> 与 <值 2> 相加。 | 结果: 任意变量 值 1: 任意变量或常数 (范围 -128 到 127) 值 2: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | <结果> := <值 1> + <值 2> | |
| 注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。 | | |

● 示例


| 流程图 | 文本 |
|--|---|
|  <pre> graph TD 0([entry]) --> 1[U(00) := 200] 1 --> 2[U(01) := 500] 2 --> 3[U(02) := U(00) + U(01)] 3 --> 4([end]) </pre> | <pre> entry U(00) := 200 U(01) := 500 U(02) := U(00) + U(01) end </pre> |
| 块编号 | 操作 |
| 1 | 将 200 分配给 U(00)。 |
| 2 | 将 500 分配给 U(01)。 |
| 3 | 将 U(00) 和 U(01) 之和分配给 U(02)。 |

| -(减) | | |
|---|-----------------------|---|
| 命令 | 说明 | 自变量 |
|  | 将 <值 1> 减去 <值 2>。 | 结果: 任意变量 值 1: 任意变量或常数 (范围 -128 到 127) 值 2: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | <结果> := <值 1> - <值 2> | |
| 注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。 | | |

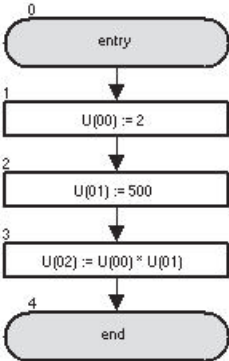
● 示例

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


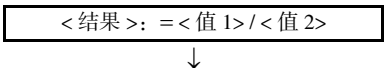
| 块编号 | 操作 |
|-----|------------------------------|
| 1 | 将 200 分配给 U(00)。 |
| 2 | 将 500 分配给 U(01)。 |
| 3 | 将 U(01) 和 U(00) 之差分配给 U(02)。 |

| *(乘) | | |
|---|-----------------------|--|
| 命令 | 说明 | 自变量 |
|  | 将 <值 1> 乘以 <值 2>。 | 结果: 任意变量 值 1: 任意变量或常数 (范围 -128 到 127) 值 2: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <结果> := <值 1> * <值 2> </div> ↓ | <结果> := <值 1> * <值 2> | |
| 注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。 | | |

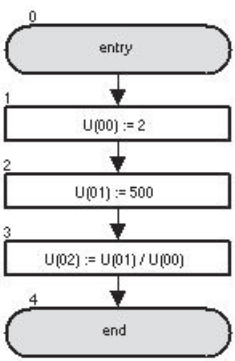
● 示例

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

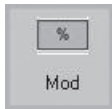
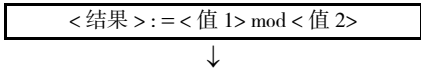
| 块编号 | 操作 |
|-----|-------------------------------|
| 1 | 将 2 分配给 U(00)。 |
| 2 | 将 500 分配给 U(01)。 |
| 3 | 将 U(00) 和 U(01) 的乘积分配给 U(02)。 |

| /(除) | | |
|---|-----------------------|---|
| 命令 | 说明 | 自变量 |
|  | 将 < 值 1 > 除以 < 值 2 >。 | 结果: 任意变量 值 1: 任意变量或常数 (范围 -128 到 127) 值 2: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | <结果>: = <值 1> / <值 2> | |
| 注 如果出现上溢、下溢或除零, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。 | | |

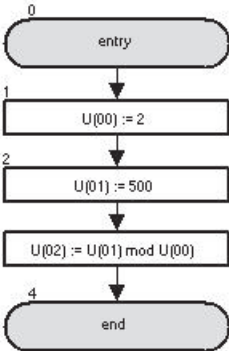
● 示例

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


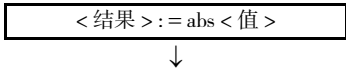
| 块编号 | 操作 |
|-----|------------------------------|
| 1 | 将 2 分配给 U(00)。 |
| 2 | 将 500 分配给 U(01)。 |
| 3 | 将 U(01) 和 U(00) 之商分配给 U(02)。 |

| Mod(求余数) | | |
|---|-------------------------------|--|
| 命令 | 说明 | 自变量 |
|  | < 值 1> 除以 < 值 2> 后所得的余数。 | 结果: 任意变量 值 1: 任意变量或常数 (范围 -128 到 127) 值 2: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | < 结果 > := < 值 1 > mod < 值 2 > | |
| 注 如果出现上溢、下溢或除零，DriveProgramming 会检测为错误。请在应用中采取必要措施，防止出现这种情况。 | | |

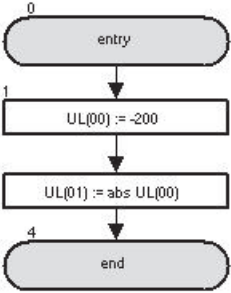
● 示例


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

| 块编号 | 操作 |
|-----|-----------------------------------|
| 1 | 将 2 分配给 U(00)。 |
| 2 | 将 500 分配给 U(01)。 |
| 3 | 将 U(01) 除以 U(00) 后所得的余数分配给 U(02)。 |

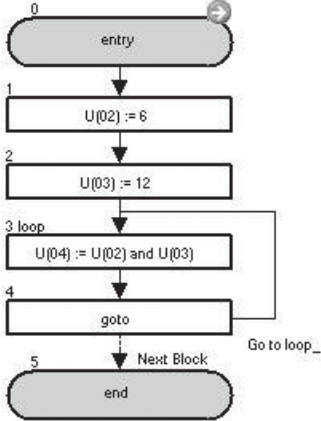
| Abs(绝对值) | | |
|---|-----------------|---|
| 命令 | 说明 | 自变量 |
|  | 查找 < 值 > 的绝对值。 | 结果: 任意变量 值: 任意变量或常数 (范围 -2,147,483,647 到 2,147,483,647) |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | <结果> := abs <值> | |
| 注 1. 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。 2. 如果变量或常数为最大负数 -2,147,483,648, 则无法将其转换为绝对值。对值 -2,147,483,648 求绝对值时结果仍为此负数。 | | |

● 示例


| 流程图 | 文本 |
|--|--|
|  | <pre> entry UL(00) := -200 UL(01) := abs UL(00) end </pre> |
| 块编号 | 操作 |
| 1 | 将 -200 分配给 UL(00)。 |
| 2 | 将 UL(00) 的绝对值分配给 UL(01)。 |

| And(逻辑与) | | | | | | | | | | | | | | | | | |
|---|--|---|-----|----|---|---|---|---|---|---|---|---|---|---|---|---|--|
| 命令 | 说明 | 自变量 | | | | | | | | | | | | | | | |
|  | 按二进制格式对 < 值 1> 和 < 值 2> 执行逻辑与运算。 and(逻辑与) | 结果: 任意变量 值 1: 任意变量或常数 (范围 -128 到 127) 值 2: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) | | | | | | | | | | | | | | | |
| | <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 | | | | | | | | | | | | | | | | | |
| 流程图法 | 文本语法 | | | | | | | | | | | | | | | | |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;"> < 结果 > := < 值 1 > and < 值 2 > </div> ↓ | < 结果 > := < 值 1 > and < 值 2 > | | | | | | | | | | | | | | | | |
| 注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。 | | | | | | | | | | | | | | | | | |

● 示例

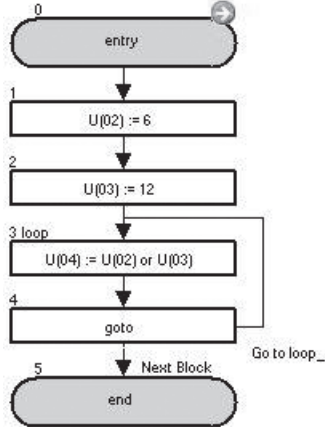
| 流程图 | 文本 |
|--|--|
|  | <pre> entry U(02) := 6 U(03) := 12 :loop_ U(04) := U(02) and U(03) goto loop_ end </pre> |

| 块编号 | 操作 |
|-----|---|
| 1 | 将 6 分配给 U(02)。 |
| 2 | 将 12 分配给 U(03)。 |
| 3 | 将 6(U(02), 二进制格式: 00000110) 和 12(U(03), 二进制格式: 00001100) 的逻辑与运算结果 4(二进制格式: 00000100) 分配给 U(04)。 |
| 4 | 无条件跳至块 3: loop_。 |


| Or(逻辑或) | | | | | | | | | | | | | | | | | |
|--|--|--|-----|----|---|---|---|---|---|---|---|---|---|---|---|---|--|
| 命令 | 说明 | 自变量 | | | | | | | | | | | | | | | |
|  | 按二进制格式对 < 值 1> 和 < 值 2> 执行逻辑或运算。 or(逻辑或) | 结果: 任意变量 值 1: 任意变量或常数 (范围 -128 到 127) 值 2: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) | | | | | | | | | | | | | | | |
| | <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 | | | | | | | | | | | | | | | | | |
| 流程图法 | 文本语言法 | | | | | | | | | | | | | | | | |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;"> < 结果 > := < 值 1 > or < 值 2 > </div> ↓ | < 结果 > := < 值 1 > or < 值 2 > | | | | | | | | | | | | | | | | |

注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。

● 示例

| 流程图 | 文本 |
|---|---|
|  <pre> graph TD 0([entry]) --> 1[U(02) := 6] 1 --> 2[U(03) := 12] 2 --> 3[U(04) := U(02) or U(03)] 3 --> 4[goto] 4 -- Go to loop_ --> 3 4 -- Next Block --> 5([end]) </pre> | <pre> entry U(02) := 6 U(03) := 12 :loop_ U(04) := U(02) or U(03) goto loop_ end </pre> |


| 块编号 | 操作 |
|-----|--|
| 1 | 将 6 分配给 U(02)。 |
| 2 | 将 12 分配给 U(03)。 |
| 3 | 将 6(U(02), 二进制格式: 00000110) 和 12(U(03), 二进制格式: 00001100) 的逻辑或运算结果 14(二进制格式: 00001110) 分配给 U(04)。 |
| 4 | 无条件跳至块 3: loop_。 |

| XOr(逻辑异或) | | | | | | | | | | | | | | | | |
|---|--|---|-----|----|---|---|---|---|---|---|---|---|---|---|---|---|
| 命令 | 说明 | 自变量 | | | | | | | | | | | | | | |
|  | 按二进制格式对 < 值 1> 和 < 值 2> 执行逻辑异或运算。 XOr(逻辑异或) | 结果: 任意变量 值 1: 任意变量或常数 (范围 -128 到 127) 值 2: 任意变量或常数 (范围 -2,147,483,648 到 2,147,483,647) | | | | | | | | | | | | | | |
| | <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 | | | | | | | | | | | | | | | | |
| 流程图法 | 文本语法 | | | | | | | | | | | | | | | |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;"> < 结果 > := < 值 1 > xor < 值 2 > </div> ↓ | < 结果 > := < 值 1 > xor < 值 2 > | | | | | | | | | | | | | | | |
| 注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。 | | | | | | | | | | | | | | | | |

● 示例

| 流程图 | 文本 |
|---|--|
| <pre> graph TD 0([entry]) --> 1[U(02) := 6] 1 --> 2[U(03) := 12] 2 --> 3[U(04) := U(02) xor U(03)] 3 --> 4[goto] 4 -- Go to loop_ --> 3 4 -- Next Block --> 5([end]) </pre> | <pre> entry U(02) := 6 U(03) := 12 :loop_ U(04) := U(02) xor U(03) goto loop_ end </pre> |

| 块编号 | 操作 |
|-----|--|
| 1 | 将 6 分配给 U(02)。 |
| 2 | 将 12 分配给 U(03)。 |
| 3 | 将 6(U(02), 二进制格式: 00000110) 和 12(U(03), 二进制格式: 00001100) 的逻辑异或运算结果 10 (二进制格式: 00001010) 分配给 U(04)。 |
| 4 | 无条件跳至块 3: loop_。 |

| Not(二进制格式的非运算) | | | | | | | | |
|---|---|---|-----|----|---|---|---|---|
| 命令 | 说明 | 自变量 | | | | | | |
|  | 按二进制格式对 < 值 1> 执行非运算(位反转) not(非) | 结果: 任意变量, 但带有位数据大小的变量除外*1 值: 任意变量, 但带有位数据大小的变量和常数除外*1 (范围 -2,147,483,648 到 2,147,483,647) | | | | | | |
| | <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 | | | | | | | | |
| 流程图法 | | 文本语言法 | | | | | | |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;"> < 结果 > := not < 值 > </div> ↓ | | < 结果 > := not < 值 > | | | | | | |

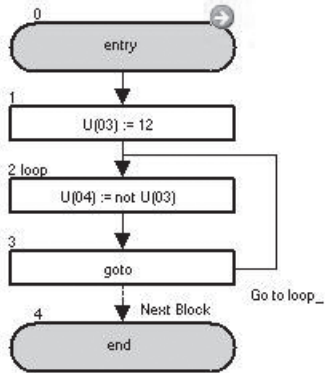
*1. 使用 UB(1) = not UB(0) 之类命令时, 无法获得正确的结果。

请使用 xor 命令反转带有位数据大小的变量, 具体示例如下所示。


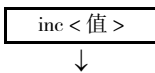
- 示例 1: UB(1) = UB(0) xor 1
- 示例 2: UB(2) = X(00) xor 1

注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。

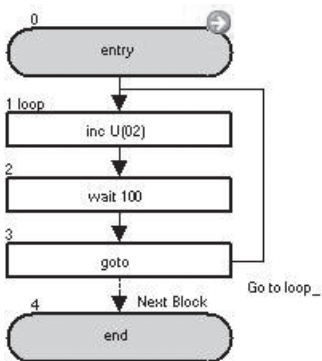
● 示例

| 流程图 | 文本 |
|--|---|
|  <pre> graph TD 0([0 entry]) --> 1[U(03) := 12] 1 --> 2[U(04) := not U(03)] 2 --> 3[goto] 3 -- Go to loop_ --> 2 3 -- Next Block --> 4([4 end]) </pre> | <pre> entry U(03) := 12 :loop_ U(04) := not U(03) goto loop_ end </pre> |


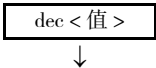
| 块编号 | 操作 |
|-----|--|
| 1 | 将 12 分配给 U(03)。 |
| 2 | 将 12(U(03), 二进制: 00000110) 的非运算结果 65523(二进制: 111110011) 分配给 U(04)。 |
| 3 | 无条件跳至块 3: loop_。 |

| Inc(加一) | | |
|---|-------------|---------|
| 命令 | 说明 | 自变量 |
|  | 将 < 值 > 加一。 | 值: 任意变量 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | inc < 值 > | |
| 注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。 | | |

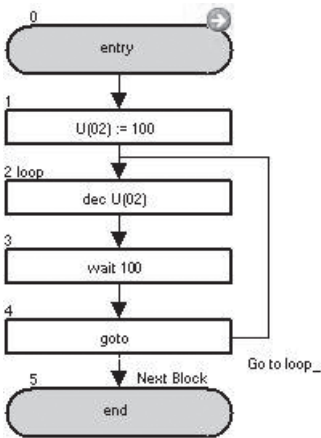
● 示例

| 流程图 | 文本 |
|---|---|
|  <pre> graph TD 0([0 entry]) --> 1[1 loop: inc U(02)] 1 --> 2[2 wait 100] 2 --> 3[3 goto] 3 -- Go to loop_ --> 1 3 -- Next Block --> 4([4 end]) </pre> | <pre> entry :loop_ inc U(02) wait 100 goto loop_ end </pre> |

| 块编号 | 操作 |
|-----|---------------------------|
| 1 | 将 U(02) 加一。 |
| 2 | 在 “wait” 命令的作用下等待 1.00 秒。 |
| 3 | 无条件跳至块 1: loop_。 |

| Dec(减一) | | |
|---|-------------|---------|
| 命令 | 说明 | 自变量 |
|  | 将 < 值 > 减一。 | 值: 任意变量 |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | dec < 值 > | |
| <p>注 如果出现上溢或下溢, DriveProgramming 会检测为错误。请在应用中采取必要措施, 防止出现这种情况。</p> | | |


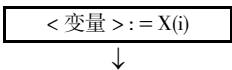
● 示例

| 流程图 | 文本 |
|--|--|
|  | <pre> entry U(02) := 100 :loop_ dec U(02) wait 100 goto loop_ end </pre> |

| 块编号 | 操作 |
|-----|-------------------------|
| 1 | 将 100 分配给 U(02)。 |
| 2 | 将 U(02) 减一。 |
| 3 | 在“wait”命令的作用下等待 1.00 秒。 |
| 4 | 无条件跳至块 2: loop_。 |

6-6 I/O 控制单元

使用这些 I/O 控制命令控制输入和输出。尽管可以使用 =(分配) 命令控制 I/O，但 I/O 控制命令可有效利用程序容量，因为它们的自变量所需的数据更小。

| var = X(i) | | |
|---|--------------------------|---|
| 命令 | 说明 | 自变量 |
|  | 将输入端子变量状态的某个位分配给 < 变量 >。 | 变量: 任意变量 (变量值为 0 或 1) i: 输入端子变量 (范围 00 到 07) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | < 变量 > := X(i) | |

注 输入端子变量是指检测变频器输入端子状态的变量。使用时需要执行以下设置。输入端子变量的数字顺序与设置的通用输入编号的数字顺序一致。

MX2: 将多功能输入 S1 到 S7 选择 (C001 到 C007) 设置为 56 到 62(MI1 到 MI7: 通用输入)。将脉冲串输入 RP 选择 (P003) 设置为 02(MI8: DriveProgramming 的通用输入 8)。

RX: 将多功能输入 S1 到 S8 选择 (C001 到 C008) 设置为 56 到 63(MI1 到 MI8: 通用输入)。

< 分配示例 >

X(00) = MI1(功能编号 56)

X(01) = MI2(功能编号 57)

X(02) = MI3(功能编号 58)

X(03) = MI4(功能编号 59)

X(04) = MI5(功能编号 60)

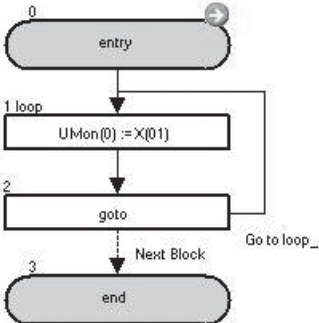
X(05) = MI6(功能编号 61)

X(06) = MI7(功能编号 62)

X(07) = MI8(MX2: P003 = 02/RX: 功能编号 63)


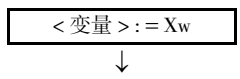
注 详情请参考 5-2 第 5-4 页上的输入/输出端子变量。

● 示例

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

在上述示例中，输入端子 X(01) 的状态由参数 UMon(0)(d025) 监控。

| 块编号 | 操作 |
|-----|----------------------|
| 1 | 将 X(01) 分配给 UMon(0)。 |
| 2 | 无条件跳至块 1: loop_。 |

| var = Xw | | |
|---|-------------------------|----------|
| 命令 | 说明 | 自变量 |
|  | 以字为单位将输入端子变量状态分配给 <变量>。 | 变量: 任意变量 |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | <变量> := Xw | |

注 输入端子变量是指检测变频器输入端子状态的变量。使用时需要执行以下设置。输入端子变量的数字顺序与设置的通用输入编号的数字顺序一致。

MX2: 将多功能输入 S1 到 S7 选择 (C001 到 C007) 设置为 56 到 62(MI1 到 MI7: 通用输入)。将脉冲串输入 RP 选择 (P003) 设置为 02(MI8: DriveProgramming 的通用输入 8)。

RX: 将多功能输入 S1 到 S8 选择 (C001 到 C008) 设置为 56 到 63(MI1 到 MI8: 通用输入)。

以字为单位分配数据时, 高位字节数据和未设置的输入端子变量的读数为零。

< 分配示例 >

X(00) = MI1(功能编号 56) 到 Xw = 1(位 0)

X(01) = MI2(功能编号 57) 到 Xw = 2(位 1)

X(02) = MI3(功能编号 58) 到 Xw = 4(位 2)

X(03) = MI4(功能编号 59) 到 Xw = 8(位 3)

X(04) = MI5(功能编号 60) 到 Xw = 16(位 4)

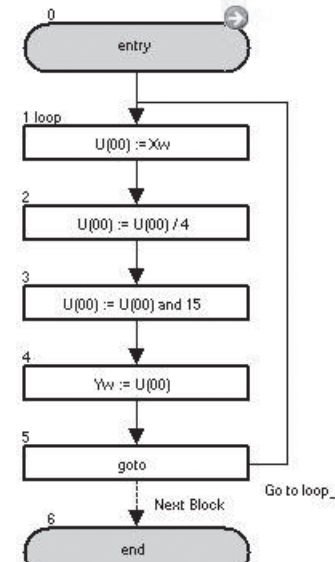
X(05) = MI6(功能编号 61) 到 Xw = 32(位 5)

X(06) = MI7(功能编号 62) 到 Xw = 64(位 6)

X(07) = MI8(MX2: P003 = 02/RX: 功能编号 63) 到 Xw = 128(位 7)


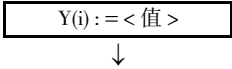
详情请参考 5-2 第 5-4 页上的输入/输出端子变量。

● 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, 输入端子 X(02) 到 X(05) 的状态会捕获下来并输出到输出端子 Y(00) 到 Y(03)。

| 块编号 | 操作 |
|-----|---|
| 1 | 将 Xw 的值 (输入端子值) 分配给 U(00)。 |
| 2 | 将 U(00) 的值除以 4 (朝右偏移两位), 从而将 X(02) 分配给位 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) 的位更改为零。 |
| 4 | 将 U(00) 分配给 Yw。 |
| 5 | 无条件跳至块 1: loop_。 |

| Y(i) = 值 | | |
|---|--------------------|---------------------------------------|
| 命令 | 说明 | 自变量 |
|  | 以位为单位将数据输出到输出端子变量。 | i: 输出端子变量 (范围 0 到 6) 值: 任意变量或常数 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | Y(i) := <值 > | |

注 输出端子变量是指控制变频器输出端子状态的变量。使用时需要执行以下设置。输出端子变量的数字顺序与设置的通用输出编号的数字顺序一致。

MX2: 将多功能输出 P1 和 P2 选择 (C021 和 C022) 以及多功能继电器输出 (MA、MB) 功能选择 (C026) 设置为 44 到 46(MO1 到 MO3: 通用输出)。

RX: 将多功能输出 P1 到 P5 选择 (C021 到 C025) 以及多功能继电器输出 (MA、MB) 功能选择 (C026) 设置为 44 到 49(MO1 到 MO6: 通用输出)。

< 分配示例 >

Y(00) = MO1(多功能编号 44)

Y(01) = MO2(多功能编号 45)

Y(02) = MO3(多功能编号 46)

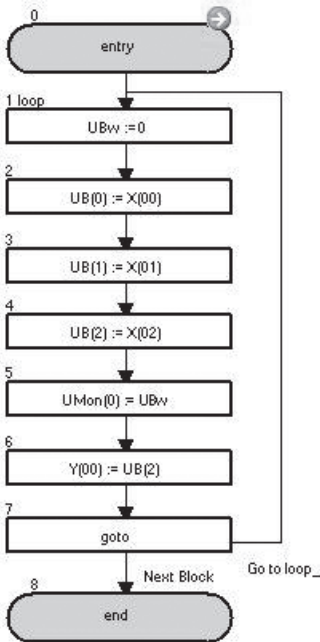
Y(03) = MO4(仅限 RX: 多功能编号 47)

Y(04) = MO5(仅限 RX: 多功能编号 48)

Y(05) = MO6(仅限 RX: 多功能编号 49)


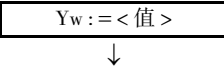
详情请参考 5-2 第 5-4 页上的输入 / 输出端子变量。

● 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，X(00) 到 X(02) 的状态使用参数 UMon(0)(d025) 监控，而且 X(02) 会输出到 Y(00)。

| 块编号 | 操作 |
|-------|--|
| 1 | 将 0 分配给 UBw。 |
| 2 到 4 | 将变量 X(00) 到 X(02) 分配给变量 UB(0) 到 UB(2)。 |
| 5 | 将 UBw 分配给 UMon(0)。 |
| 6 | 将 UB(2) 分配给 Y(00)。 |
| 7 | 无条件跳至块 1: loop_。 |

| Yw = 值 | | |
|---|-------------------------------------|------------|
| 命令 | 说明 | 自变量 |
|  | 以字为单位将数据输出到输出端子变量。 在对应的输出中反映每个位。 | 值: 任意变量或常数 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | Yw := <值 > | |

注 输出端子变量是指控制变频器输出端子状态的变量。使用时需要执行以下设置。输出端子变量的数字顺序与设置的通用输出编号的数字顺序一致。

MX2: 将多功能输出 P1 和 P2 选择 (C021 和 C022) 以及多功能继电器输出 (MA、MB) 功能选择 (C026) 设置为 44 到 46(MO1 到 MO3: 通用输出)。

RX: 将多功能输出 P1 到 P5 选择 (C021 到 C025) 以及多功能继电器输出 (MA、MB) 功能选择 (C026) 设置为 44 到 49(MO1 到 MO6: 通用输出)。

以字为单位分配数据时, 高位字节数据和未设置的输入端子变量的读数为零。

即便在这些位置进行设置, 设置也会被忽略。

< 分配示例 >

Yw = 1(位 0) 到 Y(00) = MO1(多功能编号 44)

Yw = 2(位 1) 到 Y(01) = MO2(多功能编号 45)

Yw = 4(位 2) 到 Y(02) = MO3(多功能编号 46)

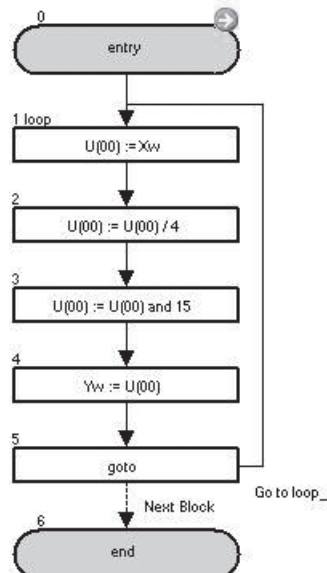
Yw = 8(位 3) 到 Y(03) = MO4(仅限 RX: 多功能编号 47)

Yw = 16(位 4) 到 Y(04) = MO5(仅限 RX: 多功能编号 48)

Yw = 32(位 5) 到 Y(05) = MO6(仅限 RX: 多功能编号 49)


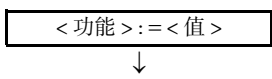
详情请参考 5-2 第 5-4 页上的输入/输出端子变量。

● 示例

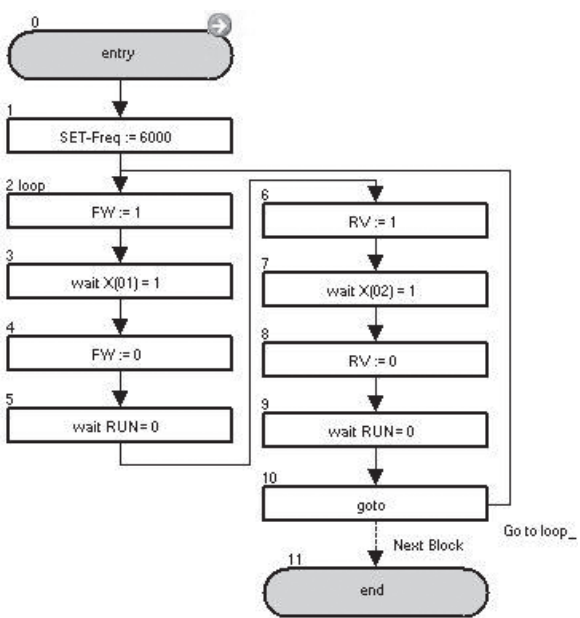
| 流程图 | 文本 |
|---|---|
|  | <pre> entry :loop_ U(00) := Xw U(00) := U(00) / 4 U(00) := U(00) and 15 Yw := U(00) goto loop_ end </pre> |

在上述示例中, 输入端子 X(02) 到 X(05) 的状态会捕获下来并输出到输出端子 Y(00) 到 Y(03)。

| 块编号 | 操作 |
|-----|---|
| 1 | 将 Xw 的值 (输入端子值) 分配给 U(00)。 |
| 2 | 将 U(00) 的值除以 4 (朝右偏移两位), 从而将 X(02) 分配给位 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) 的位更改为零。 |
| 4 | 将 U(00) 分配给 Yw。 |
| 5 | 无条件跳至块 1: loop_。 |


| func = 值 | | |
|---|---------------------|--|
| 命令 | 说明 | 自变量 |
|  | 将 < 值 > 分配给多功能输入变量。 | 功能： 多功能输入变量的任意功能位 (请参阅 5-6 第 5-14 页上的 多功能输入变量了解每个位的详情。) 值： 任意变量或常数 |
| 格式 | | |
| 流程图法 | 文本语法 | |
|  | < 功能 > := < 值 > | |

● 示例

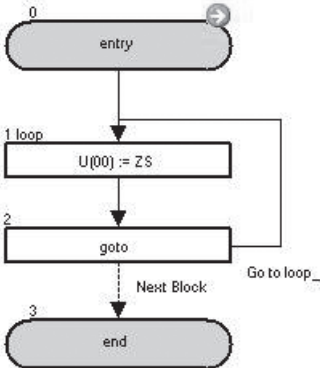
| 流程图 | 文本 |
|--|--|
|  | <pre> entry SET-Freq := 6000 :loop_ FW := 1 wait X(01) = 1 FW := 0 wait RUN = 0 RV := 1 wait X(02) = 1 RV := 0 goto loop_ end </pre> |

在上述示例中，以下操作会重复执行：以 60.00Hz 执行正向操作，直至 X(01) 变为 ON。接着，执行反向操作，直至 X(02) 变为 ON。

| 块编号 | 操作 |
|-------|---|
| 1 | 将输出频率设为 60.00Hz。 |
| 2 | 按正向操作变频器。 |
| 3 到 5 | 等待下去，直至 X(01) 变为 ON，接着停止变频器，并在“wait”命令的作用下等待，直至变频器完全停止。 |
| 6 | 按反向操作变频器。 |
| 7 到 9 | 等待下去，直至 X(02) 变为 ON，接着停止变频器，并在“wait”命令的作用下等待，直至变频器完全停止。 |
| 10 | 无条件跳至块 2: loop_。 |


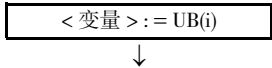
| var = func | | |
|--|-----------------------|---|
| 命令 | 说明 | 自变量 |
|  | 将多功能输出变量状态分配给 < 变量 >。 | 变量: 任意变量 功能: 多功能输出变量的任意位 |
| 格式 | | |
| 流程图法 | 文本语法 | |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;"> < 变量 > := < 功能 > </div> ↓ | < 变量 > := < 功能 > | |
| 详情请参考 5-7 第 5-16 页上的 多功能输出变量。 | | |

● 示例

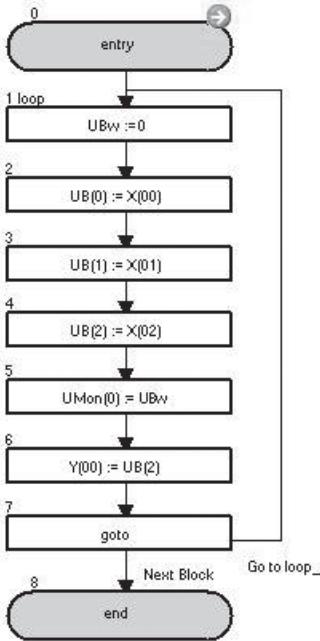
| 流程图 | 文本 |
|--|--|
|  | <pre> entry :loop_ U(00) := ZS goto loop_ end </pre> |

在上述示例中, ZS(0Hz检测信号)为ON时会把1分配给U(00), ZS(0Hz检测信号)为OFF时会把0分配给U(00)。

| 块编号 | 操作 |
|-----|-------------------|
| 1 | 将 ZS 状态分配给 U(00)。 |
| 2 | 无条件跳至块 1: loop_。 |


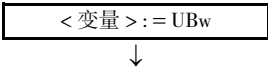
| var = UB(i) | | |
|---|------------------------|--|
| 命令 | 说明 | 自变量 |
|  | 将内部用户接点值中的某个位分配给 <变量>。 | 变量：任意变量 (变量值为 0 或 1。) i：内部用户接点编号 (范围 0 到 7) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | <变量> := UB(i) | |

● 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，X(00) 到 X(02) 的状态使用参数 UMon(0)(d025) 监控，而且 X(02) 会输出到 Y(00)。

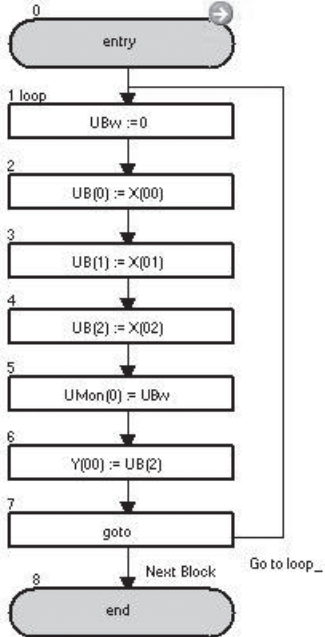
| 块编号 | 操作 |
|-------|--|
| 1 | 将 0 分配给 UBw。 |
| 2 到 4 | 将变量 X(00) 到 X(02) 分配给变量 UB(0) 到 UB(2)。 |
| 5 | 将 UBw 分配给 UMon(0)。 |
| 6 | 将 UB(2) 分配给 Y(00)。 |
| 7 | 无条件跳至块 1: loop_。 |

| var = UBw | | |
|---|--------------------------|---------|
| 命令 | 说明 | 自变量 |
|  | 以字为单位将内部用户接点值分配给 < 变量 >。 | 变量：任意变量 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | < 变量 > := UBw | |

注 以字为单位分配数据时，未使用的高位字节的读数将为零。如果执行任何设置，这些设置会被忽略。


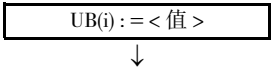
UB(0) 到 UBw = 1(位 0)
 UB(1) 到 UBw = 2(位 1)
 UB(2) 到 UBw = 4(位 2)
 UB(3) 到 UBw = 8(位 3)
 UB(4) 到 UBw = 16(位 4)
 UB(5) 到 UBw = 32(位 5)
 UB(6) 到 UBw = 64(位 6)
 UB(7) 到 UBw = 128(位 7)

● 示例

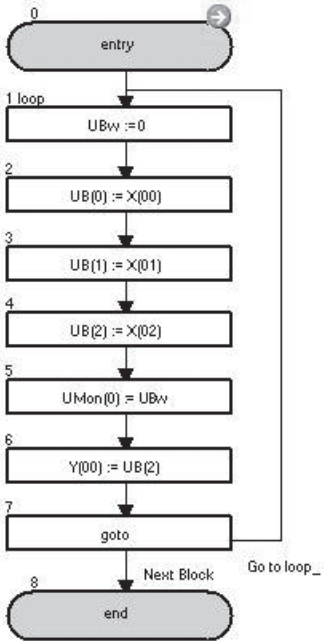
| 流程图 | 文本 |
|--|--|
|  <pre> graph TD 0([entry]) --> 1[UBw := 0] 1 --> 2[UB(0) := X(00)] 2 --> 3[UB(1) := X(01)] 3 --> 4[UB(2) := X(02)] 4 --> 5[UMon(0) = UBw] 5 --> 6[Y(00) := UB(2)] 6 --> 7[goto] 7 -- Go to loop_ --> 1 7 -- Next Block --> 8([end]) </pre> | <pre> entry :loop_ ubw := 0 UB(0) := X(00) UB(1) := X(01) UB(2) := X(02) UMon(0) := ubw Y(00) := UB(2) goto loop_ end </pre> |

在上述示例中，X(00) 到 X(02) 的状态使用参数 UMon(0)(d025) 监控，而且 X(02) 会输出到 Y(00)。

| 块编号 | 操作 |
|-------|--|
| 1 | 将 0 分配给 UBw。 |
| 2 到 4 | 将变量 X(00) 到 X(02) 分配给变量 UB(0) 到 UB(2)。 |
| 5 | 将 UBw 分配给 UMon(0)。 |
| 6 | 将 UB(2) 分配给 Y(00)。 |
| 7 | 无条件跳至块 1: loop_。 |


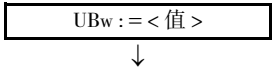
| UB(i) = 值 | | |
|---|--------------------|---|
| 命令 | 说明 | 自变量 |
|  | 将 < 值 > 分配给内部用户接点。 | i: 内部用户接点编号 (范围 0 到 7) 值: 任意变量或常数 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | UB(i) := < 值 > | |

● 示例

| 流程图 | 文本 |
|--|--|
|  <pre> graph TD 0([entry]) --> 1[UBw := 0] 1 --> 2[UB(0) := X(00)] 2 --> 3[UB(1) := X(01)] 3 --> 4[UB(2) := X(02)] 4 --> 5[UMon(0) := UBw] 5 --> 6[Y(00) := UB(2)] 6 --> 7[goto] 7 -- Go to loop_ --> 1 7 -- Next Block --> 8([end]) </pre> | <pre> entry :loop_ ubw := 0 UB(0) := X(00) UB(1) := X(01) UB(2) := X(02) UMon(0) := ubw Y(00) := UB(2) goto loop_ end </pre> |

在上述示例中，X(00) 到 X(02) 的状态使用参数 UMon(0)(d025) 监控，而且 X(02) 会输出到 Y(00)。

| 块编号 | 操作 |
|-------|--|
| 1 | 将 0 分配给 UBw。 |
| 2 到 4 | 将变量 X(00) 到 X(02) 分配给变量 UB(0) 到 UB(2)。 |
| 5 | 将 UBw 分配给 UMon(0)。 |
| 6 | 将 UB(2) 分配给 Y(00)。 |
| 7 | 无条件跳至块 1: loop_。 |

| UBw = 值 | | |
|---|-------------------------|------------|
| 命令 | 说明 | 自变量 |
|  | 以字为单位将 < 值 > 分配给内部用户接口。 | 值: 任意变量或常数 |
| 格式 | | |
| 流程图法 | 文本语法 | |
|  | UBw := < 值 > | |

注 以字为单位分配数据时, 未使用的高位字节的读数将为零。如果执行任何设置, 这些设置会被忽略。

UBw = 1(位 0) 到 UB(0)

UBw = 2(位 1) 到 UB(1)

UBw = 4(位 2) 到 UB(2)

UBw = 8(位 3) 到 UB(3)

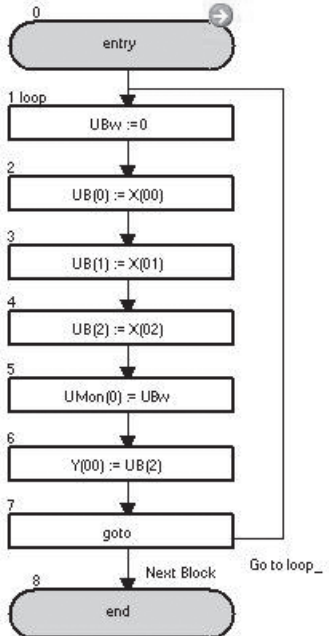
UBw = 16(位 4) 到 UB(4)

UBw = 32(位 5) 到 UB(5)

UBw = 64(位 6) 到 UB(6)

UBw = 128(位 7) 到 UB(7)

● 示例

| 流程图 | 文本 |
|---|--|
|  <pre> graph TD 0([0 entry]) --> 1[1 loop UBw := 0] 1 --> 2[2 UB(0) := X(00)] 2 --> 3[3 UB(1) := X(01)] 3 --> 4[4 UB(2) := X(02)] 4 --> 5[5 UMon(0) = UBw] 5 --> 6[6 Y(00) := UB(2)] 6 --> 7[7 goto] 7 --> 8([8 end]) 7 -- "Go to loop_" --> 1 </pre> | <pre> entry :loop_ ubw := 0 UB(0) := X(00) UB(1) := X(01) UB(2) := X(02) UMon(0) := ubw Y(00) := UB(2) goto loop_ end </pre> |

在上述示例中, X(00) 到 X(02) 的状态使用参数 UMon(0)(d025) 监控, 而且 X(02) 会输出到 Y(00)。

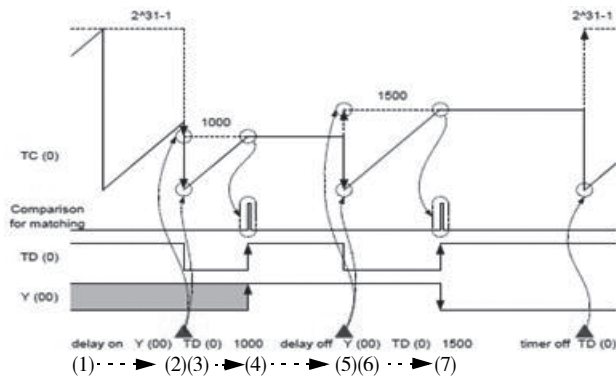
| 块编号 | 操作 |
|-------|--|
| 1 | 将 0 分配给 UBw。 |
| 2 到 4 | 将变量 X(00) 到 X(02) 分配给变量 UB(0) 到 UB(2)。 |
| 5 | 将 UBw 分配给 UMon(0)。 |
| 6 | 将 UB(2) 分配给 Y(00)。 |
| 7 | 无条件跳至块 1: loop_。 |

6-7 定时器控制命令

| Delay on/off | | |
|---|---|--|
| 命令 | 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，而且每 10ms 加一，直至达到 <值 2>。达到 <值 2> 时，定时器接点 TD(k) 会变为 ON，而且会执行为 <值 1> 指定的处理 (ON 或 OFF)。 | on/off: 延迟后的操作设置 (on/off) 值 1: 任意接点变量或变量 值 2: 任意变量或常数 (指定时间 10ms) TD(k): 使用的定时器的定时器输出接点 (k 的范围为 0 到 7) |
| 格式 | | |
| 流程图法 | 文本语法 | |
|  | delay on/off <值 1> TD(k) <值 2> | |

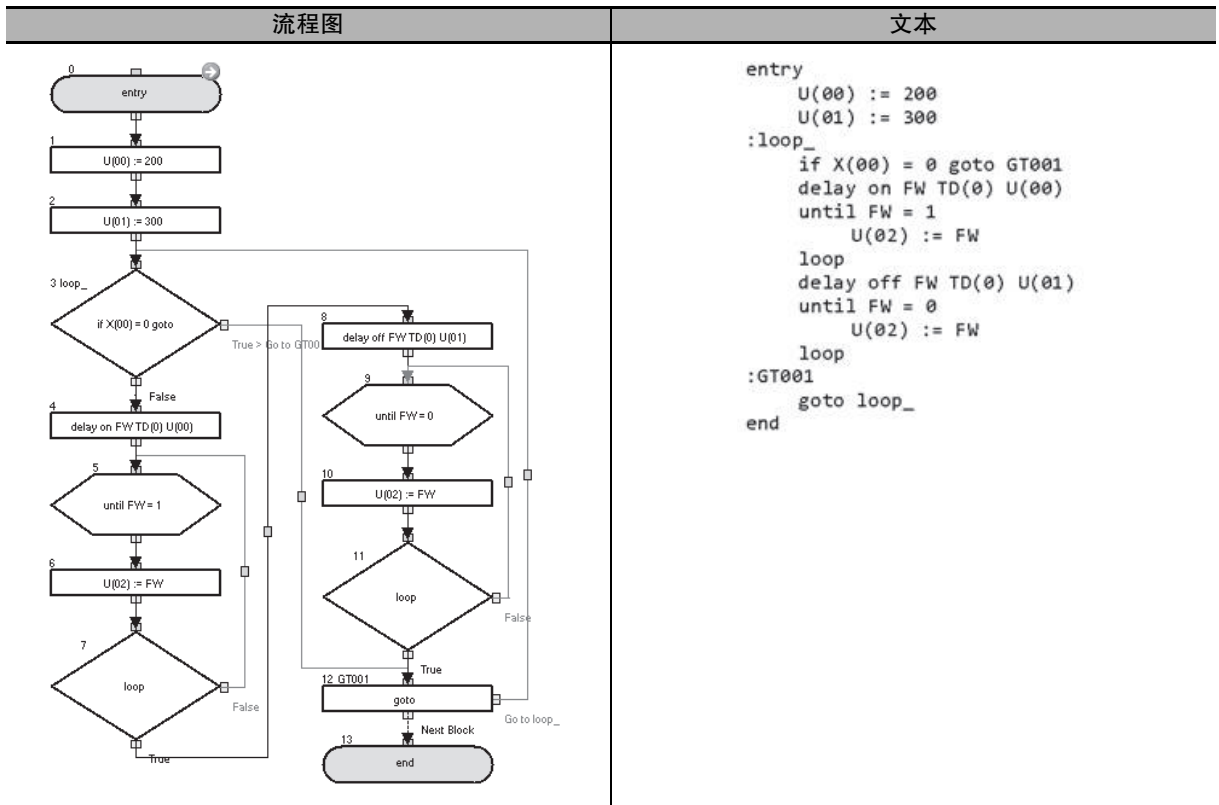
- 注 1. 定时器 (k) 会在程序执行 “delay on/off” 命令时启动。如果 TD(k) 为 ON，它会在定时器 (k) 启动时变为 OFF。但是，<值 1> 的变量不会改为原始值以外的值。
定时器 (k) 启动后，程序会转至下一个命令。
2. 执行 “delay on/off” 命令时，<值 1>、TD(k) 和 <值 2> 的数据会在内部保存。即便设为 <值 1> 和 <值 2> 的变量值在执行 “delay on/off” 命令后发生变更，命令操作也不会更改。
 3. 执行 “delay on/off” 命令后，如果同一个定时器 (k) 在其处理完成前重新启动，正在执行的处理会被取消，而定时器 (k) 会使用新的设置重新启动。因此，创建程序时，请确保定时器 (k) 在启动后不会重新启动，直至处理已完成。
 4. 您可以使用定时器计数器变量 TC(k) 监控已启动定时器 (k) 的数据。使用定时器输出接点 TD(k) 检查定时器处理的完成情况 (在完成时变为 ON)。

● 时序图





- (1) 定时器处于自由运行状态。
- (2) 启动 “delay on” 命令。
- (3) 延迟操作正在执行。
- (4) 时间已过。
- (5) 启动 “delay off” 命令。
- (6) 延迟操作正在执行。
- (7) 时间已过。

● 示例

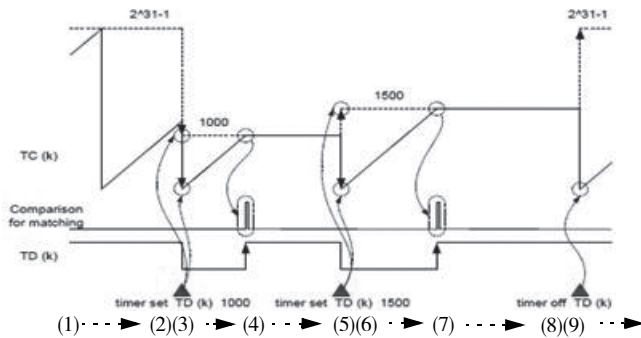


在上述示例中，当 X(00) 不是 0 时，正向操作会在 “delay on” 命令的作用下启动，并在 “delay off” 命令的作用下停止。

| 块编号 | 操作 |
|--------|---|
| 1 | 将 200 分配给 U(00)。 |
| 2 | 将 300 分配给 U(01)。 |
| 3 | 如果 X(00) 为 0 则跳至块 8: GOTO1。如果 X(00) 不为 0 则跳至下一步。 |
| 4 | 在 “delay on” 命令的作用下在 U(00) 设置值期限内启动定时器 (0)。 |
| 5 到 7 | 将 FW 状态分配给 U(02)，直至 FW 更改为 1。在 FW 状态更改为 1 时转至下一步。 |
| 8 | 在 “delay off” 命令的作用下在 U(01) 设置值期限内启动定时器 (0)。 |
| 9 到 11 | 将 FW 状态分配给 U(02)，直至 FW 更改为 0。在 FW 状态更改为 0 时转至下一步。 |
| 12 | 无条件跳至块 3: loop_。 |

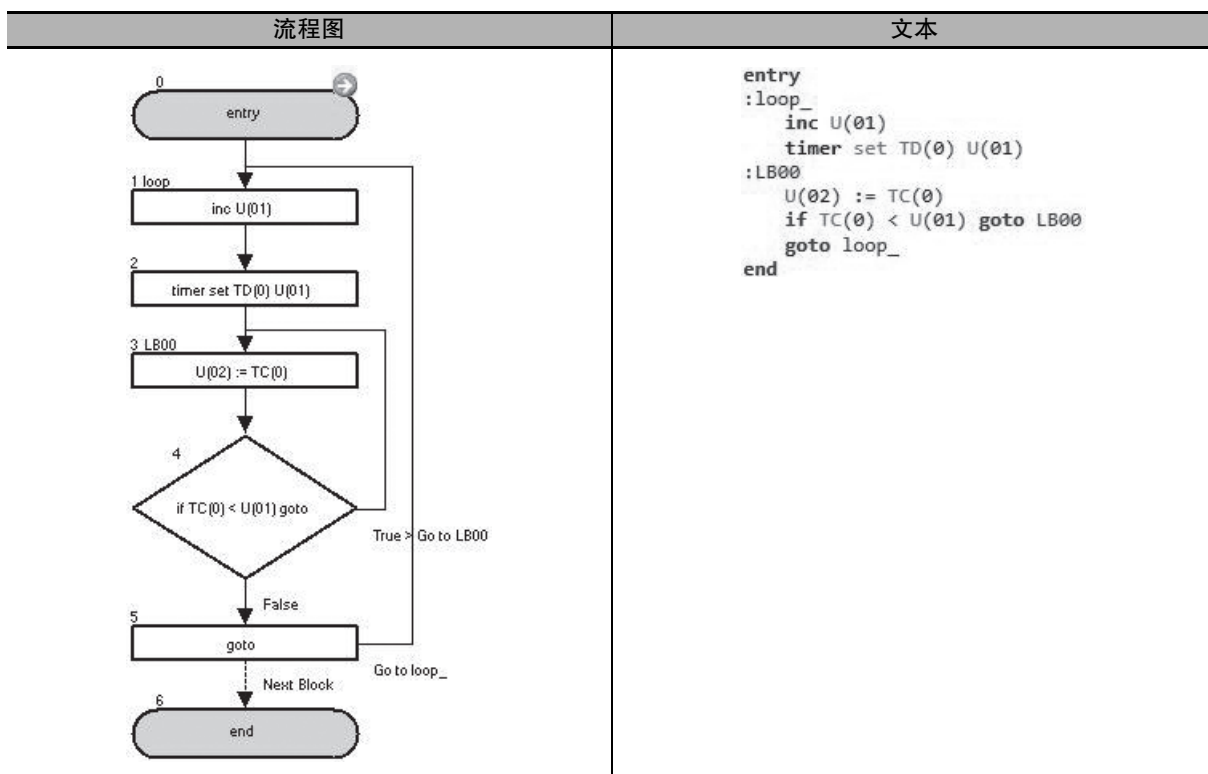
| Timer set | | |
|--|--|---|
| 命令 | 说明 | 自变量 |
|  | 启动定时器 (k)。定时器计数器 TC(k) 从 0 开始计数，而且每 10ms 加一，直至达到 < 值 >。达到 < 值 > 时，定时器接点 TD(k) 会更改为 ON。 | TD(k): 使用的定时器的定时器输出接点 (k 的范围为 0 到 7) 值: 任意变量或常数 (指定时间 10ms) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | timer set TD(k) < 值 > | |
| 注 1. 定时器 (k) 会在程序执行 “timer set” 命令时启动。如果 TD(k) 为 ON，它会在定时器 (k) 启动时变为 OFF。定时器 (k) 启动后，程序会转至下一个命令。 2. 执行 “timer set” 命令时，TD(k) 数据和 < 值 > 会在内部保存。即便设为 < 值 > 的变量值在执行 “timer set” 命令后发生变更，命令操作也不会更改。 3. 执行 “timer set” 命令后，如果同一个定时器 (k) 在其处理完成前重新启动，正在执行的处理会被取消，而定时器 (k) 会使用新的设置重新启动。因此，创建程序时，请确保定时器 (k) 在启动后不会重新启动，直至处理已完成。 4. 您可以使用定时器计数器变量 TC(k) 监控已启动定时器 (k) 的数据。使用定时器输出接点 TD(k) 检查定时器处理的完成情况 (在完成时变为 ON)。 | | |

● 时序图



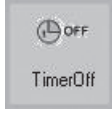
- (1) 定时器处于自由运行状态。
- (2) 启动 “timer set” 命令。
- (3) 延迟操作正在执行。
- (4) 时间已过。
- (5) 启动 “timer set” 命令。
- (6) 延迟操作正在执行。
- (7) 时间已过。
- (8) 启动 “timer set” 命令。
- (9) 定时器处于自由运行状态。

● 示例

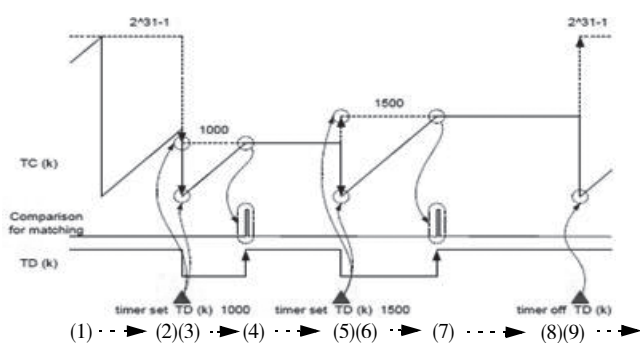


在上述示例中，每次执行完定时器后，TD(0) 的设置值都会加一，而且每次执行时每次循环所需的时间会越来越长。定时器的当前值会反映在 U(02) 中。

| 块编号 | 操作 |
|-----|--|
| 1 | 将 U(01) 加一。 |
| 2 | 在“timer set”命令的作用下在 U(01) 设置值期限内启动定时器 (0)。 |
| 3 | 将定时器 (0) 的当前值 TC(0) 分配给 U(02)。 |
| 4 | 如果定时器 (0) 的当前值 TC(0) 小于设置值 U(01)，则跳至块 3: LB00。 在其他情况下，程序会转至下一步。 |
| 5 | 无条件跳至块 1: loop_。 |

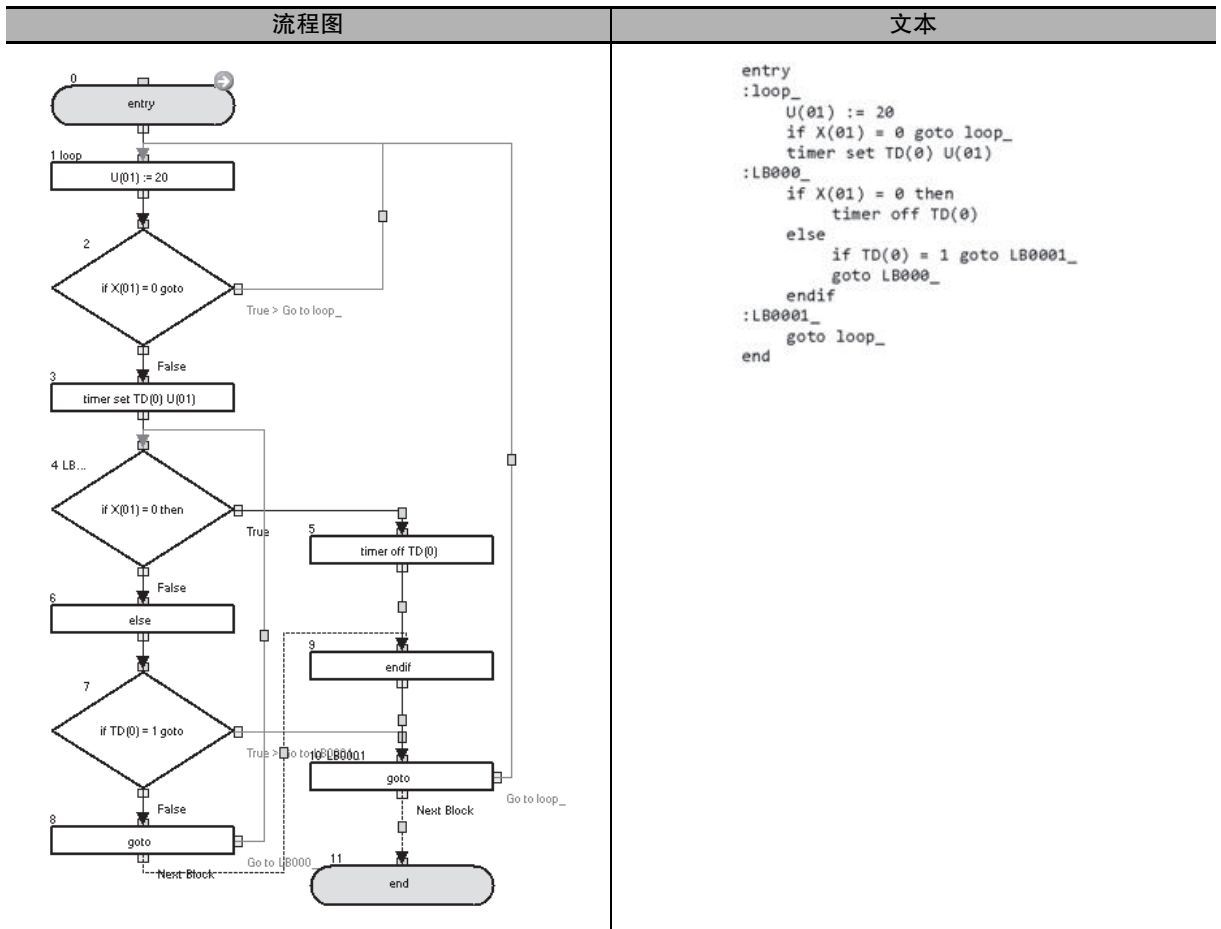
| Timer Off | | |
|---|--------------------------------------|--------------------------------------|
| 命令 | 说明 | 自变量 |
|  | 将定时器计数器 TC(k) 重置为 0，然后在自由运行模式下启动定时器。 | TD(k): 使用的定时器的定时器输出接点 (k 的范围为 0 到 7) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;">timer off TD(k)</div> ↓ | timer off TD(k) | |

● 时序图



- (1) 定时器处于自由运行状态。
- (2) 启动“timer set”命令。
- (3) 延迟操作正在执行。
- (4) 时间已过。
- (5) 启动“timer set”命令。
- (6) 延迟操作正在执行。
- (7) 时间已过。
- (8) 启动“timer off”命令。
- (9) 定时器处于自由运行状态。


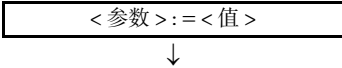
● 示例



在上述示例中，当 X(01) 不为 0 时，定时器 (0) 的启动 / 结束操作会反复执行。如果 X(01) 更改为 0，定时器会停止，而且会监控 X(01)。

| 块编号 | 操作 |
|-------|---|
| 1 | 将 20 分配给 U(01)。 |
| 2 | 如果 X(01) 为 0 则跳至块 1: loop_。如果 X(01) 不为 0 则跳至下一步。 |
| 3 | 在“timer set”命令的作用下在 U(01) 设置值期限内启动定时器 (0)。 |
| 4 到 9 | 监控 X(01)，并在 TD(0) 为 1 且 X(01) 不为 0 时跳至块 10: LB0001_。 当 X(01) 更改为 0 时，定时器 (0) 会停止，而且程序会转至下一步。 |
| 10 | 无条件跳至块 1: loop_。 |

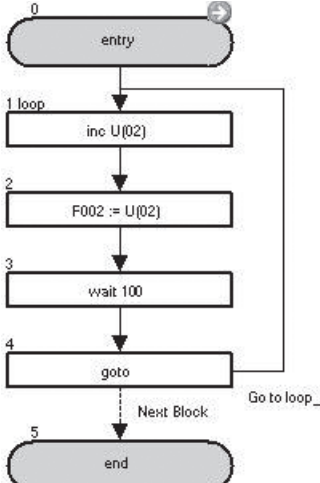
6-8 参数控制命令

| ChgParam | | |
|---|---|---|
| 命令 | 说明 | 自变量 |
|  | 将 < 参数 > 中指定的变频器参数数据更改为 < 值 >。 您可以更改任意变频器参数。 | 参数： 参数代码 (Fxxx、Axxx、bXXX、Cxxx、Hxxx 或 Pxxx) 值： 任意变量或常数 |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | < 参数 > := < 值 > | |

注 1. 与使用变频器数字操作器的设置类似，每个参数都存在以下限制。如果出现限制中所列的任何事件，变频器都会检测到命令错误 1(E45)，而 DriveProgramming 程序也会停止。有关每个参数存在的限制的详细信息，请参阅第 7 章节 *使用 DriveProgramming 参数时的注意事项* 以及“MX2-V1 系列多功能小型变频器用户手册”（样本编号：1585）或“3G3RX-V1 系列高性能通用型变频器用户手册”（样本编号：1578）。


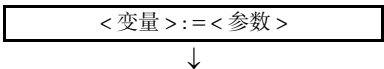
- 将超出设置范围的数据设置给参数。
 - 要设置的参数与相关参数之间出现匹配错误。
 - 设置了所选模式无法使用的功能。
 - 操作期间更改了在操作期间不能更改的参数。
 - 在软件锁选择 (b031) 禁止参数数据更改时设置了参数。
2. 即便使用“ChgParam”命令更改设置参数数据，对应参数数据在 EEPROM 中也不会更改。若要同时更改程序和 EEPROM 中的数据，请在执行“eepwrt”命令后立即执行“ChgParam”命令。

● 示例

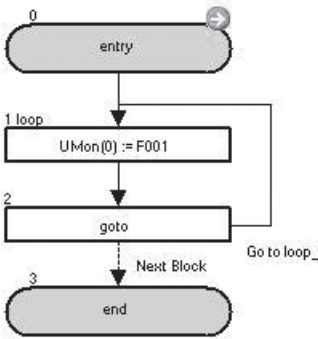
| 流程图 | 文本 |
|--|---|
|  <pre> graph TD 0([entry]) --> 1[inc U(02)] 1 --> 2[F002 := U(02)] 2 --> 3[wait 100] 3 --> 4[goto] 4 -- Go to loop_ --> 1 4 -- Next Block --> 5([end]) </pre> | <pre> entry :loop_ inc U(02) F002 := U(02) wait 100 goto loop_ end </pre> |

在上述示例中，F002(第一加速时间 1) 的值每秒加一。

| 块编号 | 操作 |
|-----|-------------------------|
| 1 | 将 U(02) 加一。 |
| 2 | 将 U(02) 分配给参数 F002。 |
| 3 | 在“wait”命令的作用下等待 1.00 秒。 |
| 4 | 无条件跳至块 1: loop_。 |


| MonParam | | |
|---|---------------------------------|--|
| 命令 | 说明 | 自变量 |
|  | 将 < 参数 > 中指定的变频器参数内容分配给 < 变量 >。 | 参数: 参数代码 (Fxxx、Axxx、bXXX、Cxxx、dxxx、Hxxx 或 Pxxx) 变量: 任意变量 |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | < 变量 > := < 参数 > | |

● 示例

| 流程图 | 文本 |
|--|--|
|  <pre> graph TD 0([0 entry]) --> 1[1 loop UMon(0) := F001] 1 --> 2[2 goto] 2 -- "Go to loop_" --> 1 2 -.-> 3([3 end]) </pre> | <pre> entry :loop_ UMon(0) := F001 goto loop_ end </pre> |

在上述示例中，参数F001(输出频率设置/监控)的值会分配给参数UMon(0)(d025)(用户监控)，并受到监控。

| 块编号 | 操作 |
|-----|-----------------------|
| 1 | 将参数 F001 分配给 UMon(0)。 |
| 2 | 无条件跳至块 1: loop_。 |

| EepWrt | | |
|---|---|-----|
| 命令 | 说明 | 自变量 |
|  | 执行此命令后立即启用允许执行“ChgParam”命令，从而将数据写入到 EEPROM。 | --- |
| 格式 | | |
| 流程图法 | 文本语言法 | |
| <pre> graph TD A[eepwrt] --> B[] </pre> | eepwrt | |

注 1. 执行“eepwrt”命令后，立即使用“ChgParam”命令设置参数数据。

此时参数数据在程序中会发生更改，同时也会保存到 EEPROM 中。重新接通电源时，可使用 EEPROM 中保存的数据作为初始数据。如果在“eepwrt”命令后未立即执行“ChgParam”命令，数据不会保存在 EEPROM 中。此外，如果在“eepwrt”命令后立即执行两个“ChgParam”命令，后一个“ChgParam”命令无法将数据保存到 EEPROM 中。

- 每个“eepwrt”命令最多要求 10ms 处理时间。执行“eepwrt”命令的任务会等待“eepwrt”命令完成，然后再转至下一个命令。
- 避免在多个任务中同时执行“eepwrt”命令，也不要连续执行该命令。每个“eepwrt”命令最多要求 10ms 处理时间。如果连续执行“eepwrt”命令，第二个命令会在第一个命令完成后执行。在此期间，连续执行“eepwrt”命令的任务不会转至下一个命令，而是进入等待状态。如果连续执行多个“eepwrt”命令，其他任务处理可能会延误，也可能因为处理时间不足而检测到错误。
- 如果在接通电源后一秒内执行“eepwrt”命令，该处理将无法正确执行。
创建程序时，请使用“wait”等命令，以便在接通电源后等待 1 秒。
- 变频器的内置 EEPROM 对写入次数有限制（约 100,000 次）。

因此，如果频繁使用“eepwrt”命令，变频器的使用寿命可能会变短。请考虑在程序中采取以下措施。



- 创建程序时，确保关闭电源前的最新数据会写入到 EEPROM 中。
- 创建程序时，确保仅在参数等数据发生更改时再写入到 EEPROM。
- 电源关闭时，请使用用户参数变量 U(29) 到 U(31) 保存数据。

● 示例：(仅将 F002 保存到 EEPROM)

| 流程图 | 文本 |
|---|---|
| <pre> graph TD 0([0 entry]) --> 1[1 loop eepwrt] 1 --> 2[2 F002 := U(02)] 2 --> 3[3 F003 := U(03)] 3 --> 4[4 goto] 4 -- Go to loop_ --> 1 4 -- Next Block --> 5([5 end]) </pre> | <pre> entry :loop_ eepwrt F002 := U(02) F003 := U(03) goto loop_ end </pre> |

在上述示例中，参数 F002 会更改为 U(02) 的值并保存在 EEPROM 中。但是，F003 会更改为 U(03) 的值，而且不会保存到 EEPROM 中。

| 块编号 | 操作 |
|-----|--|
| 1 | 通过设置将数据保存到 EEPROM 中，以备执行下面的 ChgParam 命令。 |
| 2 | 将 U(02) 分配给参数 F002。数据也会保存到 EEPROM 中，因为“eepwrt”命令恰好在此命令前一步执行。 |
| 3 | 将 U(03) 分配给参数 F003。此值不会保存到 EEPROM 中。 |
| 4 | 无条件跳至块 1: loop_。 |

| 仅限 RX: RtcSet | | |
|---|---|---|
| 命令 | 说明 | 自变量 |
|  | 将 LCD 数字操作器发来的 6 字节时钟数据设置给变量。此数据对应着年、月、日、星期几、小时和分钟。 十六进制变量值对应着年、月、日、星期几、小时和分钟 (十进制)。 rtcset on: 持续更新该 6 字节数据。 rtcset off: 仅更新该 6 字节数据一次。 | On/off: 持续更新 / 更新一次 用户变量: 任意用户参数变量或内部用户变量 U(xx) 或 UL(xx) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | rtcset on/off <用户变量 > | |

- 注 1. 此功能使用 LCD 数字操作器 (可选) 的时钟功能。如果在 LCD 数字操作器未连接时执行 “rtcset on/off” 命令, 则该命令处理不会完成, 而且整个程序会进入等待状态, 等待相关的命令。因此, 我们建议您在创建程序时, 仅在程序启动后再启动 “rtcset on” 命令, 确保时钟数据始终都能输出到指定变量。
2. LCD 数字操作器在发货时会将时钟时间设为默认数据 (2009/1/1/THU 00:00)。请先准确设置时钟时间, 然后再使用 “rtcset on/off” 命令。此外, 请务必记住, 使用此命令时, 时钟功能的时间存在一定误差 (每月误差 -1.5 到 1.5 分钟)。
3. 执行 “rtcset on/off” 命令时, BCD 数据以字节为单位输出。
一周中的每一天 (星期日到星期六) 分别表示为 00 到 06。但是, 您无法使用用户参数变量 U(k) 或内部用户变量检测 BCD 数据。创建程序时, 请注意 BCD 数据会检测为十六进制数据, 并在 DriveProgramming Editor 和数字操作器中转换为十进制数据。
4. 执行 “rtcset on/off U(k)” 时, 数据会按照以下顺序从高位字节开始设置。
必须确保连续三个用户参数变量包括 U(k)。

| 用户参数变量 | 高位字节 | 低位字节 |
|--------|-----------|------------|
| U(k) | 年份 BCD 数据 | 月份 BCD 数据 |
| U(k+1) | 日期 BCD 数据 | 星期几 BCD 数据 |
| U(k+2) | 小时 BCD 数据 | 分钟 BCD 数据 |

5. 执行 “rtcset on/off UL(k)” 时, 数据会按照以下顺序从高位字节开始设置。
必须确保连续两个内部用户变量包括 UL(k)。

| 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) | 年份 BCD 数据 | 月份 BCD 数据 | 日期 BCD 数据 | 星期几 BCD 数据 |
| UL(k+1) | 小时 BCD 数据 | 分钟 BCD 数据 | 0 (填充) | 0 (填充) |

6. 必须在变频器关闭电源的情况下安装并拆下数字操作器, 否则可能会导致错误。
如果错误导致与 LCD 数字操作器断开, 时钟数据会保留至少 1 分 45 秒。因此, 断开后仍可在 1 分 45 秒内执行 “rtcset on/off” 命令。而当时间超出 1 分 45 秒后, 时钟功能处理将无法完成, 整个程序也会进入等待命令状态。与 LCD 数字操作器恢复连接时, 程序会转至下一步。为防止整个程序进入等待状态, 创建程序时必须确保 “” 命令仅在程序启动且指定变量始终输出到时钟数据时启动。请勿通过执行 “rtcset on/off” 命令的方式在程序中检查时间。若要检查时间, 请使用指定变量。连接断开时间超出 1 分 45 秒时, 全零数据会输出到指定变量。
7. LCD 数字操作器电池电量不足时, 操作器将无法继续维持时钟数据。变频器接通电源期间, 该数据由控制电路电源维护。但是, 再次接通电源时, 时钟数据会恢复为初始数据 (2009/1/1 THU 00:00)。



安全使用注意事项

在 DriveProgramming 中使用时钟命令时, 如果 LCD 数字操作器的电池电量不足, 则会导致意外动作。

请采取检查电池电量等措施, 比如说检查时钟数据是否恢复初始设定并已停止变频器或程序, 籍此判断电池电量是否不足。

拆下或断开 LCD 数字操作器时, DriveProgramming 会在时钟命令的作用下处于等待状态。


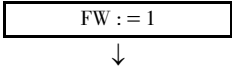

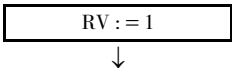

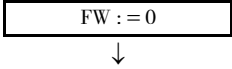
● 示例

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

如果在 2012 年 10 月 18 日 (星期四) 下午 2:29 执行此程序, 则 U(00)、U(01) 和 U(02) 在 DriveProgramming Editor 和数字操作器上的显示内容如下:

| 用户参数变量 | 时钟功能 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) | 时钟功能 BCD 数据的含义 |
|--------|---------------------------|----------------------------------|----------------------------|
| U(00) | 1210 | 4624 | “12”表示 2012 年, “10”表示十月 |
| U(01) | 1804 | 6148 | “18”表示 18 日, “04”代表星期四 |
| U(02) | 1429 | 5161 | “14”代表下午 2 点, “29”代表 29 分钟 |

6-9 变频器控制命令

| Run FW | | |
|---|---|---|
| 命令 | 说明 | 自变量 |
|  | 使得变频器开始正向运转。 此命令属于“func = value”命令，其中左侧会设为 FW(正向)变量。 | --- |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | FW := 1 | |
| Run RV | | |
| 命令 | 说明 | 自变量 |
|  | 使得变频器开始反向运转。 此命令属于“func = value”命令，其中左侧会设为 RV(反向)变量。 | --- |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | RV := 1 | |
| 停止方法 (func = value) | | |
| 命令 | 说明 | 自变量 |
|  | 使用 I/O 控制命令“func = value”使得正在运行的变频器减速并停止。 <value> 用来设置 <function>。 | function: 选择以下选项之一。 · 使用 Run FW 正向运转时: FW 变量 · 使用 Run RV 反向运转时: RV 变量 value: 设为 0(停止)可停止变频器。 |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | FW := 0 | |

注 1. 使用 Run FW 或 Run RV 命令时，可将变量 FW 或 RV 从 1(运转)更改为 0(停止)。

2. 您也可以不使用 Run FW 或 Run RV 命令，而是使用 Stop Method (func = value)，将变量 FW 或 RV 设置为 1(运行)来启动变频器。



正确使用注意事项

- 如果 DriveProgramming 程序停止，变量 FW(正向)和 RV(反向)的状态不会保留下来，而是清零。
- 仅当变频器的第一运行命令选择 (A002) 设为 01(控制电路端子台)时，变量 FW(正向)和 RV(反向)才会启用。使用其他设置选项时，变频器不会运转。
- 即便在接通电源后立即将变量 FW(正向)或 RV(反向)设为 1，这些设置也会被忽略，而且变频器也不会执行正向或反向运转。先设为 0，然后再设为 1。为避免此操作，请创建一个程序，并使用等待命令在其中指定接通电源后等待 1 秒时间。

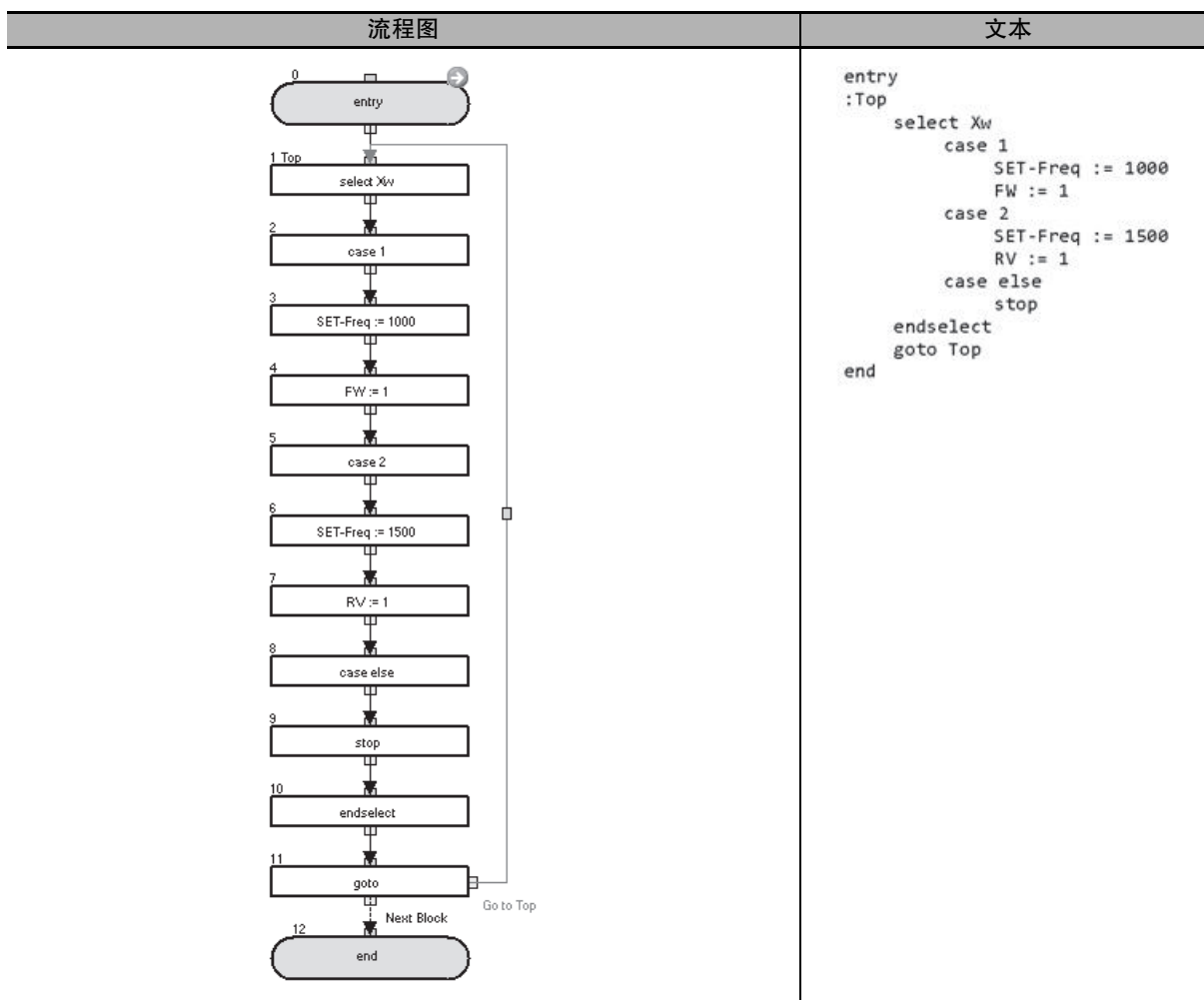
| Stop | | |
|---|--|-----|
| 命令 | 说明 | 自变量 |
|  | 使得正在运行的变频器减速并停止。 变频器检测到跳闸时，此命令会起着复位的作用。 | --- |
| 格式 | | |
| 流程图法 | 文本语法 | |
|  | stop | |

- 注 1. “stop”命令的作用与数字操作器的 STOP/RESET 按键相同。
此命令会使得正在运行的变频器减速并停止。变频器检测到跳闸时，此命令会起着复位的作用。
2. 为防止“stop”命令起着复位的作用，请将变频器的停止键选择 (b087) 设为 01(禁用)。使用此设置时，即便变频器在“stop”命令的作用下减速并停止，它也不会执行复位。但是，需要注意的是，数字操作器的 STOP/RESET 键也会被禁用。
3. 由于“stop”命令可能会起着复位的作用，并导致数字操作器的 STOP/RESET 键被禁用，我们建议您使用前述 Stop Method (func = value) 命令来停止变频器。

| Set Freq | | |
|---|--|------------------------------|
| 命令 | 说明 | 自变量 |
|  | 设置变频器频率。 此命令属于“a = (Assign)”命令，其中左侧会设为频率基准变量 SET-Freq。 单位：0.01Hz | 值：任意变量或常数 (范围 0 到 40,000) |
| 格式 | | |
| 流程图法 | 文本语法 | |
|  | SET-Freq := <value> | |

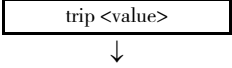
- 注 1. 变频器可以实际输出的频率范围介于启动频率 (b082) 和最大频率之间。
如果设置的数据超出此范围，变频器操作如下。
- 低于启动频率 (b082)
 - MX2 和 RX: 按频率 0.00Hz 操作。
 - 仅限 RX: 第一个控制方法 (A044) 设为 04(OSLV: 0Hz 无传感器矢量控制) 或 05(V2: 传感器矢量控制) 时，启动频率 (b082) 会禁用，而且会输出指定频率。
 - 大于最大频率
 - MX2: 将频率基准限制为第一 / 第二最大频率 (A004/A204) 中设置的值。
 - RX: 将频率基准限制为第一 / 第二 / 第三最大频率 (A004/A204/A304) 中设置的值。
2. DriveProgramming 程序停止时，程序停止前的频率基准变量数据会保留下来。重新执行程序时，进程会开始使用此前保留的数据。

● 示例



在上述示例中，如果通用输入接点 Xw 为 1，电机会以 10Hz 频率沿正向运转。如果通用输入接点 Xw 为 2，电机会以 15Hz 频率沿反向运转。使用其他值时，电机会停止。

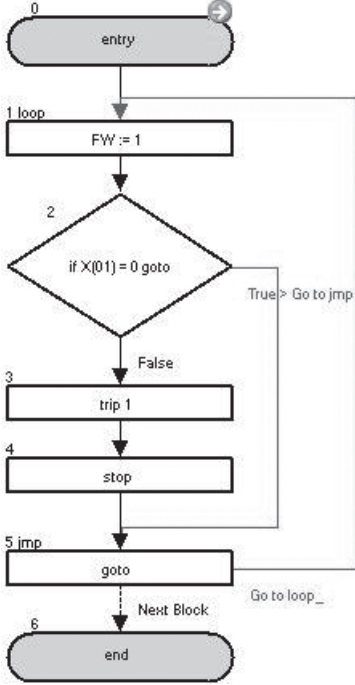
| 块编号 | 操作 |
|--------|--|
| 1 | 在“select”命令后，程序会根据 Xw 的值分叉到以下“case”命令。 Xw = 1 时分叉到“case 1”，Xw = 2 时分叉到“case 2”，以及在 Xw = 1 和 2 以外的值 a 时分叉到“case else”。 |
| 2 到 4 | Xw = 1 时执行。将频率基准变量设为 10.00Hz，沿正向操作变频器，以及在“endsselect”命令后执行下一步。 |
| 5 到 7 | Xw = 2 时执行。将频率基准变量设为 15.00Hz，沿反向操作变频器，以及在“endsselect”命令后执行下一步。 |
| 8 到 10 | Xw = 1 和 2 以外的值时执行。在“endsselect”命令后停止变频器并执行后续步骤。 |
| 11 | 无条件跳至块 1: Top。 |

| Trip | | |
|---|----------------------------|--|
| 命令 | 说明 | 自变量 |
|  | 生成变频器跳闸。 变频器会在检测到跳闸时停止。 | 值: 任意变量或常数 (范围 0 到 9) 0 到 9 对应着报警代码 E50 到 E59。 |
| 格式 | | |
| 流程图法 | 文本语法法 | |
|  | trip <value> | |

注 1. 即便出现变频器跳闸, DriveProgramming 程序也不会停止, 而是继续执行处理。

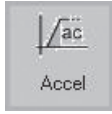
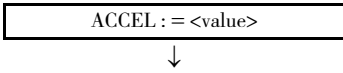
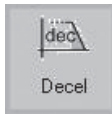
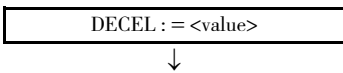
2. 不过, 您可以使用任意变量或常数 (最大 127) 设置 10 或更大的值, 即便显示的错误代码始终为 E59。

● 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 时变频器中出现用户跳闸。

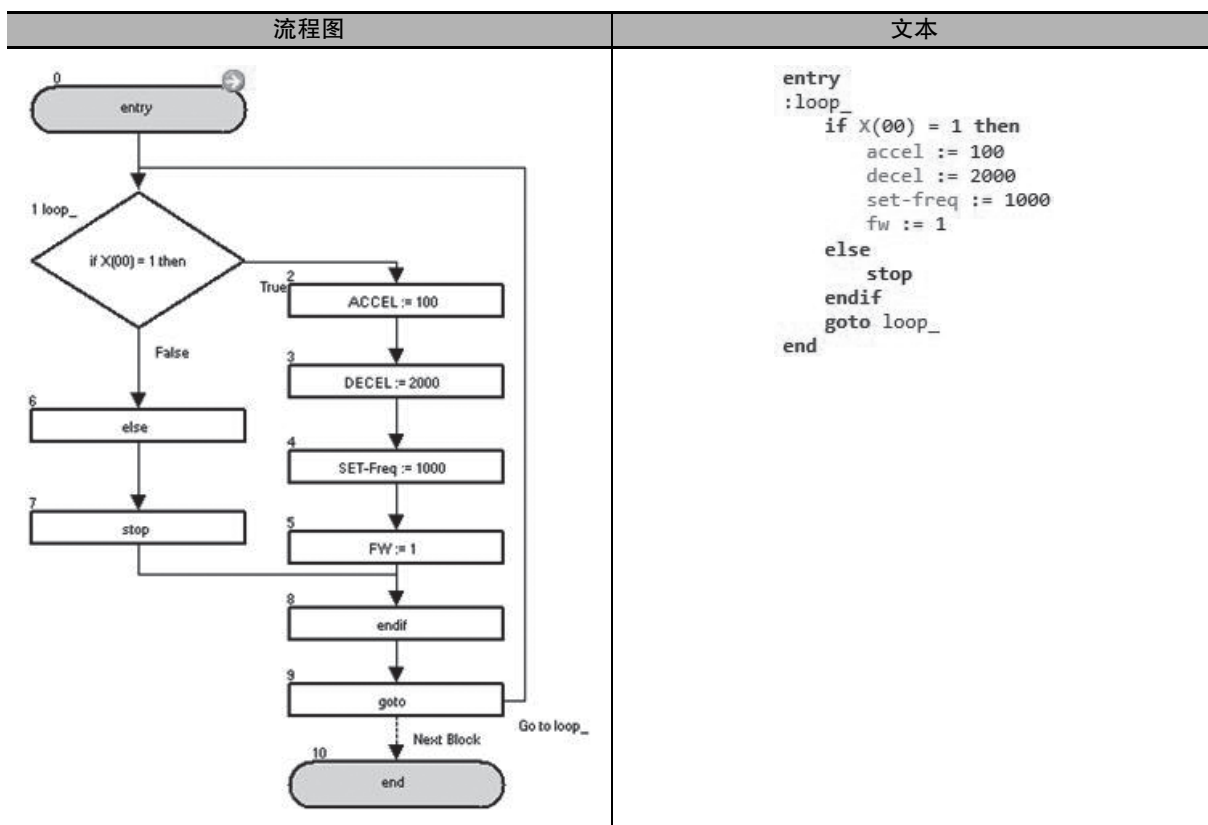
| 块编号 | 操作 |
|-----|---|
| 1 | 按正向操作变频器。 |
| 2 | 如果 X(01) 为 0, 则无条件跳至块 5: jmp。如果 X(01) 不为 0, 则程序转至下一步。 |
| 3 | 在变频器中生成用户跳闸 1。 |
| 4 | 停止变频器。 |
| 5 | 无条件跳至块 1: loop_。 |

| Accel | | |
|--|---|---|
| 命令 | 说明 | 自变量 |
|  | 设置变频器加速时间。 此命令属于“a = (Assign)”命令，其中左侧会设为加速时间变量 ACCEL。 单位：10ms | 值：任意变量或常数 (MX2：范围 0 到 360,000/RX：范围 1 到 360,000) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | ACCEL := <value> | |
| Decel | | |
| 命令 | 说明 | 自变量 |
|  | 设置变频器减速时间。 此命令属于“a = (Assign)”命令，其中左侧会设为减速时间变量 DECEL。 单位：10ms | 值：任意变量或常数 (MX2：范围 0 到 360,000/RX：范围 1 到 360,000) |
| 格式 | | |
| 流程图法 | 文本语言法 | |
|  | DECEL := <value> | |

**正确使用注意事项**

- 仅当将加速 / 减速时间输入类型 (P031) 设为 03(DriveProgramming) 时，加速时间变量 ACCEL 和减速时间变量 DECEL 才会启用。
- 在以下情况下，程序中设置的数据不会反映，而以前设置的数据会用于操作：为加速时间变量 ACCEL 或减速时间变量 DECEL 设置的数据超出范围，或为 3G3RX-V1 系列变频器设置了 0。
- DriveProgramming 程序停止时，程序停止前的加速时间变量 ACCEL 和减速时间变量 DECEL 会保留下来。重新执行程序时，进程会开始使用此前保留的数据。

● 示例



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

| 块编号 | 操作 |
|-------|---|
| 1 到 6 | 如果 X(00) 为 1，加速时间设为 1.00 秒，减速时间设为 20.00 秒，同时输出频率设为 10.00Hz。变频器会沿正向操作。 如果 X(00) 不为 1，则程序会在“else”命令后转至下一步。 |
| 7 到 8 | 停止变频器。 |
| 9 | 无条件跳至块 1: loop_。 |

7

使用 DriveProgramming 参数时的 注意事项

本章节介绍为 DriveProgramming 使用参数时的注意事项。

| | |
|------------------------------|-----|
| 7-1 受设置顺序影响的变频器参数 | 7-2 |
| 7-2 受额定电流 [%] 影响的参数 | 7-4 |
| 7-3 受 PID 启用 / 禁用影响的参数 | 7-5 |

7-1 受设置顺序影响的变频器参数

在某些情况下，参数设置范围会受到其他参数设置数据的限制。

以下是一些有代表性的示例。

有关详细信息，请参阅“MX2-V1 系列多功能小型变频器用户手册”（样本编号：I585）或“3G3RX-V1 系列高性能通用型变频器用户手册”（样本编号：I578）中的“10-2 警告功能”。

| 参数编号 | 说明 | 数据 |
|----------------|----------------------|---|
| A003 | 第一基准频率 | 30. 到第一上限频率 (A004) |
| A004 | 第一上限频率 | MX2: 第一基准频率 (A003) 到 400 RX: 30 到 400 |
| A203 | 第二基准频率 | 30. 到第二上限频率 (A204) |
| A204 | 第二上限频率 | MX2: 第二基准频率 (A203) 到 400 RX: 30 到 400 |
| A303 | 仅限 RX: 第三基准频率 | 30. 到第三上限频率 (A304) |
| A304 | 仅限 RX: 第三上限频率 | 30. 到 400 |
| A013 | FV 开始比率 | 0. 到 FV 结束比率 (A014) |
| A014 | FV 结束比率 | FV 开始比率 (A013) 到 100 |
| A103 | FI 开始比率 | 0. 到 FI 结束比率 (A104) |
| A104 | FI 结束比率 | FI 开始比率 (A103) 到 100 |
| A113 | 仅限 RX: FE 开始比率 | -100 到 FE 结束比率 (A114) |
| A114 | 仅限 RX: FE 结束比率 | FE 开始比率 (A113) 到 100 |
| A020 | 第一多步速度基准 0 | 0.00/ 启动频率到第一上限频率 (A004) |
| A220 | 第二多步速度基准 0 | 0.00/ 启动频率到第二上限频率 (A204) |
| A320 | 仅限 RX: 第三多步速度基准 0 | 0.00/ 启动频率到第三上限频率 (A304) |
| A021 到 A035 | 多步速度基准 1 到 15 | 0.00/ 启动频率到第一上限频率 (A004) |
| b015 | 自由电子热频率 1 | 0. 到自由电子热频率 2 |
| b017 | 自由电子热频率 2 | 自由电子热频率 1 到自由电子热频率 3 |
| b019 | 自由电子热频率 3 | 自由电子热频率 2 到 400 |
| b051 | 断电时的启动电压 | 0.0 到 999.9/1000 |
| b052 | 断电时的减速保持等级 | 0.0 到 999.9/1000 |
| b060 | 窗口比较器 FV 上限水平 | 设置上限水平。 设置范围: 0 到 100 下限: 下限水平 (b061)+ 滞后宽度 (b062)2 |
| b061 | 窗口比较器 FV 下限水平 | 设置下限水平。 设置范围: 0 到 100 上限: 上限水平 (b060)+ 滞后宽度 (b062)2 |
| b062 | 窗口比较器 FV 滞后宽度 | 为上限和下限水平设置滞后宽度。 设置范围: 0 到 10 上限: (上限水平 (b060)- 下限水平 (b061)) / 2 |
| b063 | 窗口比较器 FI 上限水平 | 设置上限水平。 设置范围: 0 到 100 下限: 下限水平 (b064)+ 滞后宽度 (b065)x2 |
| b064 | 窗口比较器 FI 下限水平 | 设置下限水平。 设置范围: 0 到 100 上限: 上限水平 (b063)- 滞后宽度 (b065)x2 |
| b065 | 窗口比较器 FI 滞后宽度 | 为上限和下限水平设置滞后宽度。 设置范围: 0 到 10 上限: (上限水平 (b063)- 下限水平 (b064)) / 2 |
| b066 | 仅限 RX: 窗口比较器 FE 上限水平 | 设置上限水平。 设置范围: -100 到 100 最小值: 下限水平 (b067)+ 滞后宽度 (b068)2 |

| 参数编号 | 说明 | 数据 |
|------|----------------------|---|
| b067 | 仅限 RX: 窗口比较器 FE 下限水平 | 设置下限水平。 设置范围: -100 到 100 最大值: 上限水平 (b066)- 滞后宽度 (b068)2 |
| b068 | 仅限 RX: 窗口比较器 FE 滞后宽度 | 为上限和下限水平设置滞后宽度。 设置范围: 0 到 10 最大值: (上限水平 (b066)- 下限水平 (b067)) /2 |
| b100 | 自由 V/f 频率 1 | 0 到自由 V/f 频率 2(b102) |
| b102 | 自由 V/f 频率 2 | 0 到自由 V/f 频率 3(b104) |
| b104 | 自由 V/f 频率 3 | 0 到自由 V/f 频率 4(b106) |
| b106 | 自由 V/f 频率 4 | 0 到自由 V/f 频率 5(b108) |
| b108 | 自由 V/f 频率 5 | 0 到自由 V/f 频率 6(b110) |
| b110 | 自由 V/f 频率 6 | 0 到自由 V/f 频率 7(b112) |
| b112 | 自由 V/f 频率 7 | 0 到 400 |

7-2 受额定电流 [%] 影响的参数

对于设置电流值的参数来说，它们的默认数据和设置范围会受变频器额定电流限制。

设置这些参数时，请在 DriveProgramming 程序中设置变频器额定电流的百分比 [%]。

以下是一些有代表性的示例。

有关详细信息，请参阅“MX2-V1 系列多功能小型变频器用户手册”（样本编号：I585）或“3G3RX-V1 系列高性能通用型变频器用户手册”（样本编号：I578）中的“8-5 Modbus 通信寄存器编号列表”。

| 参数编号 | 说明 | 数据 |
|------|-----------------|--|
| b012 | 第一电热保护等级 | 0.20 到 1.00 额定电流 |
| b212 | 第二电热保护等级 | 0.20 到 1.00 额定电流 |
| b312 | 仅限 RX：第三电热保护等级 | 0.20 到 1.00 额定电流 |
| b022 | 第一过载限制水平 | < 重载 > |
| b222 | 仅限 MX2：第二过载限制水平 | 0.20 到 2.00 额定电流（0.4 到 55 kW） |
| b025 | 第一过载限制水平 2 | 0.20 到 1.80 额定电流（75 到 132 kW） |
| b028 | 频率牵引重新启动等级 | < 轻载 > 0.20 到 1.50 额定电流 |
| b126 | 制动释放电流 | 0.20 到 2.00 额定电流（0.4 到 55 kW） 0.20 到 1.80 额定电流（75 到 132 kW） |
| C030 | 数字电流监控基准值 | < 重载 > 0.20 到 2.00 额定电流 < 轻载 > 0.20 到 1.50 额定电流 |
| C039 | 低电流检测等级 | < 重载 > 0.00 到 2.00 额定电流（0.4 到 55 kW） 0.00 到 1.80 额定电流（75 到 132 kW） < 轻载 > 0.00 到 1.50 额定电流 |
| C041 | 第一过载警告等级 | 0.0：功能未激活 < 重载 > 0.00 到 2.00 额定电流（0.4 到 55 kW） |
| C241 | 仅限 MX2：第二过载警告等级 | 0.00 到 2.00 额定电流（75 到 132 kW） < 轻载 > 0.00 到 1.50 额定电流 |
| C111 | 第一过载警告等级 2 | < 重载 > 0.00 到 2.00 额定电流（0.4 到 55 kW） 0.00 到 1.80 额定电流（75 到 132 kW） < 轻载 > 0.00 到 1.50 额定电流 |

7-3 受 PID 启用 / 禁用影响的参数

设置值会根据 A071/A075 的设置进行比例缩放。

| 参数编号 | 说明 | 数据 | |
|------|-------------------|--------------------|---------|
| | | A071=01 或 02 | A071=00 |
| A011 | FV 启动频率 | 转换后的值 = 设置值 (A075) | 设置值 |
| A012 | FV 结束频率 | 转换后的值 = 设置值 (A075) | 设置值 |
| A020 | 第一多步速度基准 0 | 转换后的值 = 设置值 (A075) | 设置值 |
| A021 | 多步速度基准 1 | 转换后的值 = 设置值 (A075) | 设置值 |
| A022 | 多步速度基准 2 | 转换后的值 = 设置值 (A075) | 设置值 |
| A023 | 多步速度基准 3 | 转换后的值 = 设置值 (A075) | 设置值 |
| A024 | 多步速度基准 4 | 转换后的值 = 设置值 (A075) | 设置值 |
| A025 | 多步速度基准 5 | 转换后的值 = 设置值 (A075) | 设置值 |
| A026 | 多步速度基准 6 | 转换后的值 = 设置值 (A075) | 设置值 |
| A027 | 多步速度基准 7 | 转换后的值 = 设置值 (A075) | 设置值 |
| A028 | 多步速度基准 8 | 转换后的值 = 设置值 (A075) | 设置值 |
| A029 | 多步速度基准 9 | 转换后的值 = 设置值 (A075) | 设置值 |
| A030 | 多步速度基准 10 | 转换后的值 = 设置值 (A075) | 设置值 |
| A031 | 多步速度基准 11 | 转换后的值 = 设置值 (A075) | 设置值 |
| A032 | 多步速度基准 12 | 转换后的值 = 设置值 (A075) | 设置值 |
| A033 | 多步速度基准 13 | 转换后的值 = 设置值 (A075) | 设置值 |
| A034 | 多步速度基准 14 | 转换后的值 = 设置值 (A075) | 设置值 |
| A035 | 多步速度基准 15 | 转换后的值 = 设置值 (A075) | 设置值 |
| A101 | FI 启动频率 | 转换后的值 = 设置值 (A075) | 设置值 |
| A102 | FI 结束频率 | 转换后的值 = 设置值 (A075) | 设置值 |
| A145 | 频率增加量设置 | 转换后的值 = 设置值 (A075) | 设置值 |
| A220 | 第二多步速度基准 0 | 转换后的值 = 设置值 (A075) | 设置值 |
| A320 | 仅限 RX: 第三多步速度基准 0 | 转换后的值 = 设置值 (A075) | 设置值 |
| F001 | 输出频率设置 / 监控 | 转换后的值 = 设置值 (A075) | 设置值 |
| d004 | PID 反馈值监控 | 转换后的值 = 反馈值 (A075) | 反馈值 |

8

错误与纠正措施

本章节介绍出错时的程序操作、与 DriveProgramming 有关的错误以及错误原因与纠正措施。

| | |
|--|-----|
| 8-1 故障诊断 | 8-2 |
| 8-1-1 出错时的 DriveProgramming 操作 | 8-2 |
| 8-1-2 复位错误时的 DriveProgramming 操作 | 8-3 |
| 8-1-3 报警代码一览表 | 8-4 |

8-1 故障诊断

本章节介绍出错时的程序操作、与 DriveProgramming 有关的错误代码以及相应的纠正措施。

8-1-1 出错时的 DriveProgramming 操作

基本上，即便变频器在 DriveProgramming 操作期间检测到跳闸，操作也会继续执行。但是，如果检测到与 DriveProgramming 有关的 E43 到 E45 跳闸，操作会停止。或者，如果遇到“on trip goto”命令，程序会在发生跳闸后跳至其他进程。

| 有 / 无 “on trip goto” | 出错状态 | | |
|-------------------------|-------------------------------------|--|-------------------------------------|
| | 用户跳闸 E50 到 E59 | 与 DriveProgramming 有关的 E43 到 E45 跳闸 | 其他跳闸 |
| 无 | 操作继续。 | 程序停止。 | 操作继续。 |
| 有 | 执行“on trip goto”命令后，程序会跳至指定标签并继续操作。 | 程序停止。 | 执行“on trip goto”命令后，程序会跳至指定标签并继续操作。 |



安全使用注意事项

DriveProgramming 程序停止后，DriveProgramming 控制的多功能输出会保留程序停止之前的状态。因此需对系统进行相关配置，确保变频器中 DriveProgramming 程序停止操作可被 DriveProgramming 启动信号和报警（跳闸）信号检测到，而且变频器周边设备可安全停止。

8-1-2 复位错误时的 DriveProgramming 操作

复位错误时的 DriveProgramming 操作会因复位输入方法和复位选择 (C102) 设置的不同而异。下表列出了各种情况下的操作。

● 复位输入方法

- 若要重新启动 DriveProgramming，请将多功能输入选择 (MX2: C001 到 C007/RX: C001 到 C008) 设为 18(RS(复位))，并将对应的输入端子置于 ON。
- 按数字操作器或 LCD 数字操作器 (选件) 的停止 / 复位键无法重新启动 DriveProgramming。

● 复位选择 (C102)

- 若要通过复位输入重新启动 DriveProgramming，必须将复位选择 (C102) 设为 03(仅限跳闸复位) 以外的值。设置 03(仅限跳闸重置) 时，无法重新启动 DriveProgramming。
- 将重置选择 (C102) 设为 00(接通电源时跳闸重置) 或 01(关闭电源时跳闸重置) 时，即便没有跳闸也能通过重置输入重新启动 DriveProgramming。
- 将复位选择 (C102) 设为 02(仅在跳闸期间启用) 时，DriveProgramming 只能在变频器检测到跳闸时重新启动。

如果变频器中未出现跳闸，它无法使用复位输入重新启动。

| 操作方式 | 程序状态 | 跳闸状态 | 使用控制端子复位 | | | 使用操作器的停止 / 复位键复位 | | |
|---------------------------|------|------|-----------------|---------|---------|------------------|---------|---------|
| | | | C102=00 或 01 | C102=02 | C102=03 | C102=00 或 01 | C102=02 | C102=03 |
| DriveProgramming 操作 *1 | 运行期间 | 正常状态 | 重新启动 | 保持运行 | 保持运行 | 保持运行 | 保持运行 | 保持运行 |
| | | 跳闸期间 | 重新启动 | 重新启动 | 保持运行 | 保持运行 | 保持运行 | 保持运行 |
| | 停止期间 | 正常状态 | 重新启动 | 保持停止 *2 | 保持停止 *2 | 保持停止 *2 | 保持停止 *2 | 保持停止 *2 |
| | | 跳闸期间 | 重新启动 | 重新启动 | 保持停止 *2 | 保持停止 *2 | 保持停止 *2 | 保持停止 *2 |
| 变频器复位操作 | 正常状态 | 复位 | 禁用 | 禁用 | 复位 | 禁用 | 禁用 | |
| | 跳闸期间 | 复位 | 复位 | 复位 *3 | 复位 | 复位 | 复位 *3 | |

*1 上表假设 DriveProgramming 功能选择 (A017) 设为 01 时 PRG 端子为 ON。如果 PRG 端子为 OFF，保持停止操作会适用于所有情况，因为程序并未运行。

*2 若要重新启动程序，请先将 DriveProgramming 功能选择设为 00(禁用)，然后将其更改为 02(启用(电源接通 / 关闭时启动 / 停止))。此外，如果 A017=01，可将 PRG 端子重新置于 ON 来重新启动。

*3 如果 C102=03，复位操作只会清除跳闸。当前值等内部计算数据不会初始化。

8-1-3 报警代码一览表

本章节介绍与 DriveProgramming 有关的报警代码以及报警原因和纠正措施。

对于其他错误，请参阅“MX2-V1 系列多功能小型变频器用户手册”（样本编号：I585）或“3G3RX-V1 系列高功能通用型变频器用户手册”（样本编号：I578）。

| 报警代码 | 报警 (变频器跳闸原因) | 可能的原因 | 检查内容 | 纠正措施 |
|------|-----------------|---|---|--|
| E43 | 无效的命令 | 虽然没有程序被下载至变频器，但端子 PRG 仍然置于 ON。 | 上传程序，检查该程序是否实际存在于变频器中。 | 重新创建程序并将其下载至变频器。 |
| E44 | 嵌套计数错误 | 子程序嵌套层数超过八层。 | 上传程序并检查嵌套层数。 | 纠正程序，确保嵌套层数不到八层。 |
| | | “for next”命令嵌套层数超过八层。 | | |
| | | “if”命令嵌套层数超过八层。 | | |
| E45 | 命令错误 1 | “goto”命令的跳转目标指向终止“for”或其他循环的下一个命令。 | 检查“goto”命令的跳转目标是否为终止循环的命令。 | 纠正“goto”命令的跳转目标 |
| | | 无法找到通过 U(i) 等引用的变量。 | 检查为 U(i) 指定的数值。 $00 \leq i \leq 31$ | 纠正为 U(i) 指定的变量值，或限制 U(i) 变量的数据范围。 |
| | | 四则运算导致以下错误： 上溢、下溢或除零 | 检查程序，找出导致上溢、下溢或除零的命令。 | 纠正错误，确保四则运算不会导致上溢、下溢或除零。 |
| | | “ChgParam”命令导致以下错误： · 更改为不存在的参数 · 写入值超出设置范围 · 更改了在变频器操作期间不能更改的参数值 · 更改了受软件锁选择（启用此设置时）保护的参数值 | · 检查参数或写入至参数的值。 · 在变频器操作期间出现错误时，检查能否在变频器操作期间更改参数。 · 检查软件锁选择 (b031) 的设置。 | · 纠正参数或写入至参数的值，确保它处于设置范围内。 · 禁用软件锁功能。 · 如果在操作期间可以更改参数，请将软件锁选择 (b031) 的设置更改为 10，以便允许在变频器操作期间更改参数。 |