



中华人民共和国国家标准

GB/T 15969.5—2002
idt IEC 61131-5:2000

可编程序控制器 第5部分:通信

Programmable controllers—
Part 5: Communications

2002-11-25 发布

2003-05-01 实施

中华人民共和国
国家质量监督检验检疫总局 发布

GB/T 15969.5—2002

目 次

前言	VI
IEC 前言	VII
1 范围	1
2 引用标准	1
3 定义	2
4 符号和缩略语	3
5 模型	4
5.1 PLC 网络通信模型	4
5.2 PLC 功能模型	4
5.3 PLC 硬件模型	5
5.4 软件模型	5
6 PLC 通信服务	6
6.1 PLC 子系统及其状态	6
6.2 应用特指的功能	12
7 PLC 通信功能块	16
7.1 通信功能块概述	16
7.2 通信 FB 参数的语义	17
7.3 设备检验	21
7.4 轮询数据采集	24
7.5 编程数据采集	27
7.6 参数控制	36
7.7 互锁控制	38
7.8 编程报警报告	45
7.9 连接管理	49
7.10 通信功能块的应用举例	52
8 一致性和实现者特指的特性和参数	55
8.1 一致性	55
8.2 实现特指的性能和参数	55
附录 A(标准的附录) 映象到 ISO/IEC 9506-5	57
附录 B(标准的附录) 使用 GB/T 16720.2 的 PLC 行为特性	75
图 1 GB/T 15969.5 的范围	1
图 2 PLC 通信模型	4
图 3 PLC 的功能模型	4
图 4 PLC 的硬件模型	5
图 5 PLC 的软件模型	6
图 6 PLC 的电源	9

GB/T 15969.5—2002

图 7 状态信息的类型描述	11
图 8 互锁控制的时间线	13
图 9 REMOTE-VAR 功能	18
图 10 状态信号发送原理	19
图 11 ERROR 和 STATUS 输出的时序图	20
图 12 STATUS 功能块	21
图 13 USTATUS 功能块	22
图 14 STATUS 功能块的时序图	22
图 15 STATUS 功能块的状态图	23
图 16 USTATUS 功能块的状态图	24
图 17 READ 功能块	25
图 18 READ 功能块的时序图	26
图 19 READ 功能块的状态图	26
图 20 编程数据采集数据流	27
图 21 USEND 功能块	28
图 22 URCV 功能块	28
图 23 USEND 和 URCV 功能块的时序图	29
图 24 USEND 功能块的状态图	29
图 25 URCV 功能块的状态图	30
图 26 BSEND 功能块	32
图 27 BRCV 功能块	32
图 28 BSEND 和 BRCV 功能块的时序图	33
图 29 BSEND 功能块的状态图	33
图 30 BRCV 功能块的状态图	35
图 31 WRITE 功能块	37
图 32 WRITE 功能块的时序图	37
图 33 WRITE 功能块的状态图	38
图 34 SEND 功能块	39
图 35 RCV 功能块	40
图 36 SEND 和 RCV 功能块的时序图	41
图 37 SEND 功能块的状态图	42
图 38 RCV 功能块的状态图	43
图 39 NOTIFY 功能块	45
图 40 ALARM 功能块	46
图 41 ALARM 功能块的时序图	46
图 42 NOTIFY 功能块的状态图	47
图 43 ALARM 功能块的状态图	48
图 44 CONNECT 功能块	50
图 45 CONNECT 功能块的时序图	50
图 46 CONNECT 功能块的状态图	51

GB/T 15969.5—2002

图 47 功能块图语言的举例	55
表 1 表述状态的实体	6
表 2 PLC 摘要状态	7
表 3 I/O 子系统的状态	8
表 4 处理单元的状态	9
表 5 电源的状态	9
表 6 存储器的状态	10
表 7 通信子系统的状态	10
表 8 实现者特指的子系统	10
表 9 状态信息的表述	11
表 10 设备检验特性	12
表 11 数据采集特性	13
表 12 控制性能	14
表 13 报警报告的性能	14
表 14 可启动和可停止的单元	14
表 15 I/O 状态的含义	15
表 16 I/O 状态	15
表 17 执行和 I/O 控制性能	15
表 18 可装载单元	16
表 19 应用程序传送特性	16
表 20 连接管理性能	16
表 21 通信功能块概述	17
表 22 通信 FB 参数的语义	18
表 23 SCOPE 参数的值	19
表 24 STATUS 的输出值及其说明	20
表 25 STATUS 状态图的转换	23
表 26 STATUS 状态图的动作表	23
表 27 USTATUS 状态图的转换	24
表 28 USTATUS 状态图的动作表	24
表 29 READ 状态图的转换	26
表 30 READ 状态图的动作表	27
表 31 USEND 功能块状态图的转换	29
表 32 USEND 功能块状态图的动作表	30
表 33 URCV 状态图的转换	30
表 34 URCV 功能块状态图的动作表	31
表 35 BSEND 状态图的转换	34
表 36 BSEND 状态图的动作表	34
表 37 BRCV 状态图的转换	35
表 38 BRCV 状态图的动作表	36
表 39 WRITE 状态图的转换	38

GB/T 15969.5—2002

表 40	WRITE 状态图的动作表	38
表 41	SEND 状态图的转换	42
表 42	SEND 状态图的动作表	43
表 43	RCV 状态图的转换	44
表 44	RCV 状态图的动作表	44
表 45	NOTIFY 状态图的转换	47
表 46	NOTIFY 状态图的动作表	47
表 47	ALARM 状态图的转换	48
表 48	ALARM 状态图的动作表	49
表 49	CONNECT 状态图的转换	51
表 50	CONNECT 状态图的动作表	52
表 51	相符的表头和相关的表	55
表 52	实现特指的特性和参数	56
表 A1	类型描述的映像	59
表 A2	SCOPE 和 SC_ID 参数的映像	59
表 A3	直接表达式的大小前缀	60
表 A4	STATUS 状态图的转换映像	61
表 A5	STATUS 状态图的动作映像	61
表 A6	USTATUS 状态图的转换映像	61
表 A7	USTATUS 状态图的动作映像	62
表 A8	READ 状态图的转换映像	62
表 A9	READ 状态图的动作映像	62
表 A10	USEND 状态图的转换映像	63
表 A11	USEND 状态图的动作映像	63
表 A12	URCV 状态图的转换映像	64
表 A13	URCV 状态图的动作映像	64
表 A14	BSEND 状态图的转换映像	65
表 A15	BSEND 状态图的动作映像	65
表 A16	BRCV 状态图的转换映像	66
表 A17	BRCV 状态图的动作映像	66
表 A18	WRITE 状态图的转换映像	67
表 A19	WRITE 状态图的动作映像	67
表 A20	SEND 状态图的转换映像	67
表 A21	SEND 状态图的动作映像	68
表 A22	RCV 状态图的转换映像	68
表 A23	RCV 状态图的动作映像	69
表 A24	NOTIFY 状态图的转换映像	71
表 A25	NOTIFY 状态图的动作映像	71
表 A26	ALARM 状态图的转换映像	72
表 A27	ALARM 状态图的动作映像	72

GB/T 15969.5—2002

表 A28	CONNECT 状态图的转换映像	73
表 A29	CONNECT 状态图的动作映像	74
表 A30	实现特指的特性和参数	74
表 B1	CreateProgramInvocation 服务的缺省	75
表 B2	用于 I/O State 参数的 Program Invocation 服务的缺省	75
表 B3	实现者特指的特性和参数	75

GB/T 15969.5—2002

前 言

可编程序控制器自 1969 年问世以来,已在工业自动化的各个领域广泛使用,并成为工业自动化系统的重要支柱。由 IEC/TC65/SC65B/WG7 制定的可编程序控制器国际标准 IEC 61131-1~61131-4 于 1992~1994 年间公布,我国于 1995 年将这些国际标准等效转化为我国国家标准 GB/T 15969.1~15969.4—1993。

可编程序控制器技术的发展十分迅速,首先在国际上已由单一机型发展为整套系列(微型、小型、中型、大型、特大型),且通信、联网、运算、自适应控制等功能大大增强;其次,应用范围亦由逻辑控制扩展到运动控制、过程控制、批量控制、配方控制等;第三,控制范围亦由单机扩展到整个车间以至全厂范围和无线电远程控制,从而覆盖了一部分由 DCS、NC、ROBOT 系统的应用领域。总之,可编程序控制器已不断朝纵向和横向集成扩展。

正是在这种局面下,IEC 决定对原有的 IEC 61131 系列国际标准进行全面、深入的修改和增补,其中包括对 IEC 61131-1~61131-4 的修改和增补,并增加 IEC 61131-5《可编程序控制器 第 5 部分:通信》、IEC 61131-7《可编程序控制器 第 7 部分:可编程序控制器模糊控制编程》和 IEC 61131-8《可编程序控制器 第 8 部分:应用和实现编程语言的指南》。IEC 61131-5 已于 2000 年 9 月通过最终文件(FDIS)并公布,为了保持我国国家标准 GB/T 15969 可编程序控制器系列的连续性,现将等同采用 IEC 61131-5。本标准在技术内容和编排方式上与国际标准等同,其术语和编号均与 IEC 61131-5 相一致。

本标准的附录 A 和附录 B 是标准的附录。

本标准由国家机械工业局提出。

本标准由全国工业过程测量和控制标准化技术委员会 CSBTS/TC124 归口。

本标准由北京机械工业自动化研究所负责起草。

本标准主要起草人:郝玉成、李百煌、杨昌焜、王蔚庭。

GB/T 15969.5—2002

IEC 前言

1) IEC(国际电工技术委员会)是由所有国家电工技术委员会(IEC 国家委员会)组成的国际标准化组织。IEC 的目标是推动在电气和电子领域内所有关于标准化问题的国际协作。为此目的,IEC 出版国际标准。这些国际标准的准备工作委托技术委员会进行,任何对所涉及主题有兴趣的 IEC 国家委员会均可参与准备工作,与 IEC 有联系的国际的、政府或非政府的组织亦可参与准备工作。IEC 和国际标准化组织(ISO)根据这两个国际组织的协议所确定的条件进行紧密的协作。

2) IEC 对有关技术问题的正式决议或协议尽可能地表达有关主题的国际上的一致意见,因为每一个技术委员会都有来自感兴趣的国家委员会的代表。

3) 所产生的文件以推荐标准的形式供国际上使用,并以标准、技术规范、技术报告或指南形式发布。在此意义上它们为各国家委员会所接受。

4) 为了促进国际上的统一,IEC 国家委员会同意在最大可能的范围内采用 IEC 国际标准作为其国家标准或地区标准。IEC 标准和相应国家或地区标准之间的任何不一致之处应在国家或地区标准中明确地指出。

5) IEC 以无标志方法表示它的批准,IEC 对任何申明符合某个 IEC 标准的设备不承担责任。

6) 应注意到国际标准中的某些成分有可能是某个专利主题。IEC 并不负责识别任何或所有专利。国际标准 IEC 61131-5 已由 65B 分委员会进行准备;它属于 IEC TC65:工业过程测量和控制技术委员会。

本标准文本基于以下的文件:

FDIS	投票报告
65B/XX/FDIS	65B/XX/RVD

有关批准本标准投票的全部信息可在以上列出的投票报告中找到。此出版物系根据 ISO/IEC 指导规程第 3 部分起草。

本标准应结合 IEC 61131 的其他标准文件阅读。IEC 61131 在总标题“可编程序控制器”下包括以下部分:

第 1 部分:通用信息

第 2 部分:设备特性

第 3 部分:编程语言

第 4 部分:用户导则(已作为 IEC TR 61131-4 技术报告发布)

第 5 部分:通信

第 7 部分:模糊控制编程

第 8 部分:应用和实现编程语言的指南(作为 IEC TR 61131-8 技术报告发布)

附录 A 和附录 B 是本标准的组成部分。

本标准和其他 IEC 标准有冲突之处(除基本安全标准外),应考虑本标准的条款只指导可编程序控制器及其相连接的外围设备的领域。

IEC TC65 委员会已决定此出版物的内容将保留到 2006 年不作更改。到此日期,此出版物将被:

- 再度确认
- 撤消
- 更新版本替代或
- 修改

以后将发布此标准的双语版本。

中华人民共和国国家标准

可程序控制器
第 5 部分:通信GB/T 15969.5—2002
idt IEC 61131-5:2000Programmable controllers—
Part 5: Communications

1 范围

本标准规定了可程序控制器的通信范围。从可程序控制器(PLC)的角度,它规定了任何设备如何与作为服务器的 PLC 进行通信以及 PLC 如何与任何设备进行通信。它特别规定了当 PLC 为其他设备提供服务和 PLC 应用程序能从其他设备请求服务时 PLC 的行为特性。它并不倾向于规定应用 PLC 作为路由器或网关时,任何设备之间怎样进行通信。规定 PLC 作为通信客户机和服务器的行为特性独立于专指的通信子系统,但 PLC 的通信功能可能取决于所用通信子系统的功能。

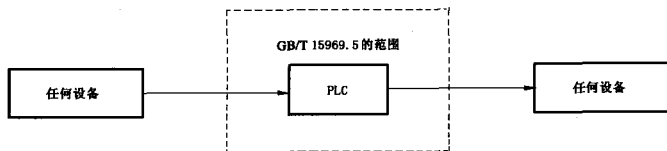


图 1 GB/T 15969.5 的范围

本标准的范围是 GB/T 15969.3 中图 1.2.2 所示“通信模型”的一个子集;这就是说,GB/T 15969.3 中的图 1.2.2b 和图 1.2.2c 包括在本标准的范围内。此外,本标准中所定义的方法适用于程序内或程序之间的通信。在附录中提供 PLC 行为特性到某些特殊通信子系统的映像。

2 引用标准

下列标准所包含的条文,通过在本标准中引用而构成为本标准的条文。本标准出版时,所示版本均为有效。所有标准都会被修订,使用本标准的各方应探讨使用下列标准最新版本的可能性。

- GB/T 15969.1—1995 可程序控制器 第 1 部分:通用信息(eqv IEC 61131-1:1993)
 GB/T 15969.2—1995 可程序控制器 第 2 部分:设备特性(eqv IEC 61131-2:1993)
 GB/T 15969.3—1995 可程序控制器 第 3 部分:编程语言(eqv IEC 61131-3:1993)
 GB/T 5271.1—2000 信息技术 词汇 第 1 部分:基本术语(eqv ISO/IEC 2382-1:1993)
 GB/T 16720.1—1996 工业自动化系统 制造报文规范 第 1 部分:服务定义(eqv ISO 9506-1:1990)
 GB/T 16720.2—1996 工业自动化系统 制造报文规范 第 2 部分:协议规范(eqv ISO/IEC 9505-2:1990)
 IEC 60050-351—1998 国际电工技术词汇 第 351 部分:自动控制

3 定义

本标准采用以下定义：

本标准基于 GB/T 15969.1～GB/T 15969.3 的概念并使用以下已在其他国际标准中定义的术语。

来自其他出版物的定义：

IEC 60050-351

control

控制

monitoring

监控

GB/T 15969—1995

access path

存取路径

application program

应用程序

application program archiving

应用程序归档

modifying the application program

修改应用程序

cold restart

冷启动

direct representation

直接表达式

input

输入

invocation

启用

main processing unit

主处理单元

output

输出

program

编程

programmable controller

可编程控制器

programmable controller system

可编程控制器系统

sub-element

子元素

testing the application program

测试应用程序

warm restart

热启动

GB/T 5271.1—1985

data

数据

GB/T 16979.1—1996

client

客户机

download

下载

event

事件

server

服务器

uninterruptible variable access

不可中断的变量存取

upload

上传

variable

变量

control

控制

本标准的定义：

3.1 报警 alarm

发出特定状态信号的事件。

3.2 数据采集 data acquisition

为过程监控和报告生成而进行的数据收集。

3.3 直接操作员接口 direct operator interface

客户机经由通信系统与操作员接口进行通信,而不需要应用程序的交互。

3.4 设备检验 device verification

允许其他设备确定 PLC 能否执行它在控制系统中执行的功能。

3.5 健康 health

PLC 或其子系统的健康状况仅由返回的三个可能值中的一个值来决定。这三个可能值根据健康值的下降程度排序:正常(GOOD),警告(WARNING)和异常(BAD)。

3.6 互锁控制 interlocked control

通过二个用户之间数据交换的同步进行控制。在不同的时间点,一个用户等待其他用户发出它所期望的某些数据。

3.7 本地 local

在 PLC 的内部;相对于远程。

3.8 参数控制 parametric control

由客户机写入到驻留在 PLC 中的控制变量来进行控制。

3.9 处理单元 processing unit

主处理单元的部分,它是 PLC 系统负责存储应用程序和数据以及应用程序的一部分。一台 PLC 系统有一个或一个以上的处理单元。

3.10 程序检验 program verification

测试 PLC 的应用程序,对它执行所设计的运行于过程环境的功能进行验证。

3.11 制造方法 recipe

用于生产一种产品的工艺规程、或用于这些工艺规程的数据、或两者的描述,而此产品的生产与以前的产品不同,它使用装配有控制器的过程或机器。

3.12 远程 remote

在 PLC 的外部;相对于本地。

3.13 状态 state

PLC 系统的状态由属性表指示,其中每个属性可以是 TRUE(真)或 FALSE(假)。零个、一个或多个属性可以同时为 TRUE。

3.14 非请求的 unsolicited

不需要显式请求的执行。

4 符号和缩略语

以下是频繁用于本标准中的某些缩略语。这些术语在本标准中被定义或引用。

CFB	Communication function block 通信功能块
FB	Function block 功能块
I/O	Input and output 输入和输出
IEC	International Electrotechnical Commission 国际电工技术委员会
ISO	International Organization for Standardization 国际标准化组织
MMS	Manufacturing Message Specification, 制造报文规范
OSI	Open Systems Interconnection 开放系统互连
PADT	Programming and debugging tool 编程和调试工具
PLC	Programmable controller 可编程序控制器
PU	Processing unit 处理单元

5 模型

本章规定用于本标准其余部分中的模型。

5.1 PLC 网络通信模型

PLC 对控制系统的其他部分提供某些特殊的应用功能。它也可能向其他的 PLC 请求某些功能。在本标准中定义的通信功能基于一个通信子系统,该子系统能将通信中的出错情况报告给 PLC 的信号处理功能(参阅 5.2)。

图 2 说明一个通信网络中的设备,展示了可能从 PLC2(图 2 中的可编程序控制器 2)请求 PLC 功能(客户机)的三个设备。二个标有粗线框的 PLC 属本标准范围内。

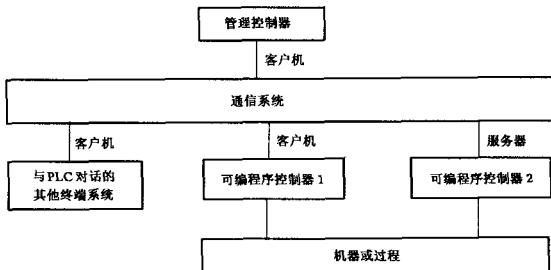


图 2 PLC 通信模型

注：从通信的观点来看,图 2 中的“管理控制器”以及“与 PLC 对话的其他终端系统”对 PLC 通信服务器来说其行为特性相同,即它们均向 PLC2 提出请求。

一个 PLC 能使用其客户机功能与任何设备进行通信,但是这些设备要具有与 PLC 相同的行为特性。

5.2 PLC 功能模型

一台 PLC 包括若干功能(参阅图 3)。对于本标准范围内的 PLC,至少具有一个通信功能。

图 3 取自 GB/T 15969.1 中的图 1,它说明了一个典型 PLC 中的一些子系统。

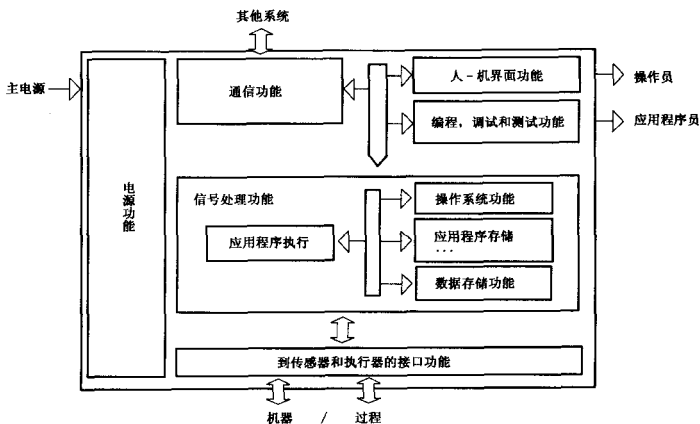


图 3 PLC 的功能模型

有一种功能,它是 PLC 系统的一部分,但它通常被看作 PLC 本身之外的部分,如人们熟悉的编程和调试工具(PADT)。PADT 在模型中通过通信功能与 PLC 进行交互。

到传感器和执行器的接口功能有到主处理单元(参阅 5.3 的硬件模型)的本地或远程 I/O。对每个应用程序而言,到传感器和执行器的接口功能有二个属性,它们定义 PLC 如何对机器/过程进行监视和控制。

输入属性有以下状态:

- 向应用程序提供的输入正在由传感器提供;
- 向应用程序提供的输入正保持为当前状态。

输出属性有以下状态:

- 执行器正由应用程序进行控制;
- 执行器正保持为当前状态。

5.3 PLC 硬件模型

图 4 给出了 PLC 的硬件模型和组成一台 PLC 的各种模块。一台 PLC 子系统包含有一个或多个模块。图 4 对应于 GB/T 15969.1 的图 B1 和 GB/T 15969.2 的图 1。

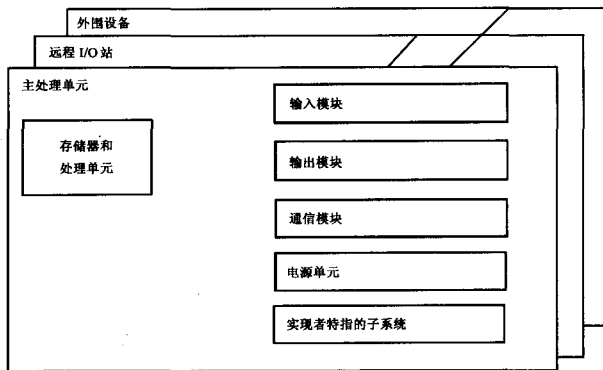


图 4 PLC 的硬件模型

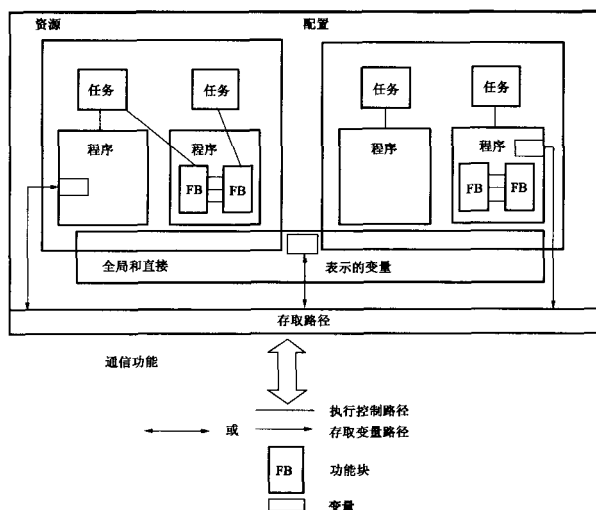
5.4 软件模型

图 5 为由 GB/T 15969.3 中图 1.2.1 所定义的 PLC 软件模型,它描述了 PLC 编程语言的基本高级语言元素及其相互关系。这些包括应用 GB/T 15969.3 定义的语言编程的元素,即程序和功能块;以及配置元素,即配置、资源、任务、全局变量和存取路径。此存取路径支持将 PLC 的程序安装到 PLC 系统内。

一个配置对应于 GB/T 15969.1 定义的 PLC 系统的语言元素。一个资源对应于一个“信号处理功能”和它的“人-机界面”以及如 GB/T 15969.1 定义的传感器/执行器接口功能(如任何功能)。一个配置包括一个或多个资源,每个资源包括在零个或多个任务控制下执行的一个或多个程序。一个程序可能包括零个或多个功能块或由 GB/T 15969.3 定义的其他语言元素。

配置和资源可由 GB/T 15969.1 定义的“操作员接口”“编程、测试和监控”或“操作系统”功能进行启动和停止。本标准中定义了通过通信服务启动或停止配置和资源的机制。由 GB/T 15969.1 定义的“通信功能”可装载或删除程序、资源、全局变量、存取路径(和其相应的存取特许权)以及配置。配置或资源的装载或删除等效于装载和删除它包含的所有元素。

存取路径及其相应的存取特许权,允许由通信服务存取 PLC 的变量。



注 1：此图只用于说明。图形表达是非标准的。

注 2：在只有单一资源的配置中，该资源不需要显性表示。

图 5 PLC 的软件模型

6 PLC 通信服务

本章阐述 PLC 状态信息的概念，并对 PLC 通过通信子系统提供给控制系统的服务作出规定（以下条文规定 PLC 应用程序如何应用通信子系统与其他设备进行交互）。

6.1 PLC 子系统及其状态

一台 PLC 提供的状态包括状态信息和故障指示。

在表 1 中列举的某些子系统能报告的状态。此外，还提供有关 PLC 一般信息的摘要状态。

表 1 表述状态的实体

序号	表述状态的实体
1	PLC(作为整体)
2	I/O 子系统(包括输入和输出模块以及其他智能模块设备)
3	处理单元
4	电源子系统
5	存储器子系统
6	通信子系统
7	实现者特指的子系统

注：这些状态意图提供有关控制器包括其硬件和固件子系统的信息，不考虑配置信息。它不倾向于提供有关的控制过程信息，也不提供 PLC 应用程序的信息。状态数据包括有关 PLC 及其子系统的 state 和 health 状态信息。

在本标准中使用二个与状态有关的概念:health 和 state。

PLC 或其子系统的“health”状态仅由返回的三种可能值中的一个值来决定。与每种状态值有关的语义规定如下,它们根据 health 状态的下降程度排序。

a) GOOD(正常)——如为 TRUE(真),PLC(或特定的子系统)没有检测到能禁止其运行预期功能的任何问题。

b) WARNING(警告)——如为 TRUE(真),PLC(或特定的子系统)没有检测到能禁止其运行预期功能的任何问题,但至少检测到某一个问题,它使得 PLC(或特定子系统)在某些功能上出现一些限制。这个限制可能是时间、性能等。对这些限制的进一步定义见以下的说明。

c) BAD(异常)——如为 TRUE(真),PLC(或特定的子系统)已检测到禁止 PLC(或特定的子系统)运行预期功能的至少一个问题。

PLC 系统的“state”状态由一个属性表指示,其中每个属性可能是 TRUE 或 FALSE。在同一时间内,可能有零个、一个或多个属性是 TRUE。每个属性的有关语义在本条文中叙述。

每个状态信息还可能实现者指定的属性。实现者特指的属性的某些举例是:

a) 附加的出错诊断(例如超过了 EEPROM 的写周期);

b) 附加的运行状态(例如启用自动标定);

c) 本地关键状态(例如要求自动再启动)。

实现不需要提供子系统的状态。所有存在于一个系统内的相似类型的子系统状况均需单独报告。应提供子系统的名称以允许区分相同类型的各个子系统。

6.1.1 PLC 摘要状态

PLC 提供如下摘要状态信息:

表 2 PLC 摘要状态

序号	项 目	描 述	
1	Health	GOOD	PLC 中的所有子系统都指示 GOOD 的 health 状态
2		WARNING	至少有一个子系统指示有一个 WARNING 的 health 状态,没有一个子系统指示有一个 BAD 的 health 状态
3		BAD	至少有一个子系统指示有一个 BAD 的 health 状态
4	Running	如为 TRUE,此属性指示已装载至少一部分用户应用程序并且该部分在 PLC 的控制下	
5	Local control	如为 TRUE,此属性指示本地越权控制是否被激活,如果是,则从网络控制 PLC 及其子系统的能力受到限制,例如这可能紧紧绑定于一个本地的键开关上的应用	
6	No outputs Disabled	如为 TRUE,此属性指示作为执行应用程序或其他方法的结果,PLC 可能改变其所有输出的物理状态。如不为 TRUE,则某些输出的物理状态不受 PLC 影响(有可能影响逻辑状态)。这典型地用于在 PLC 中的应用程序测试和修改	
7	No inputs Disabled	如为 TRUE,此属性指示作为执行应用程序或其他方法的结果,PLC 可能存取所有输入的物理状态。如不为 TRUE,则 PLC 不能存取某些输入的物理状态。这典型地用于 PLC 中应用程序的测试和修改。在这些场合,输入信号可通过仿真得到	
8	Forced	如为 TRUE,此属性指示至少有一个 PLC 的 I/O 点已被强制。当一个输入点被强制时,应用程序将接收由 PADT 所指定的值以替代机器或过程的实际值。当一个输出点被强制时,机器或过程将接收由 PADT 所指定的值以替代由执行应用程序而生成的值。当一个变量被强制时,则应用程序将使用由 PADT 所指定的值以替代由执行正常程序而生成的值	

表 2(完)

序号	项 目	描 述
9	User application present	如为 TRUE,此属性指示处理单元至少在处理一个用户应用程序
10	I/O subsystem	如为 TRUE,此属性指示“WARNING”或“BAD”是由一个子系统所引起
11	Processing unit Subsystem	如为 TRUE,此属性指示“WARNING”或“BAD”是由一个处理单元子系统所引起
12	Power supply subsystem	如为 TRUE,此属性指示“WARNING”或“BAD”是由一个电源子系统所引起
13	Memory Subsystem	如为 TRUE,此属性指示“WARNING”或“BAD”是由一个存储器子系统所引起
14	Communication Subsystem	如为 TRUE,此属性指示“WARNING”或“BAD”是由一个通信子系统所引起
15	Implementer Specified Subsystem	如为 TRUE,此属性指示“WARNING”或“BAD”是由一个实现者特指的子系统所引起

6.1.2 I/O 子系统

PLC 提供其 I/O 子系统的状态信息如下:

表 3 I/O 子系统的状态

序号	项 目	描 述	
1	Health	GOOD	指示在 I/O 子系统中没有检测到有错误
2		WARNING	指示在 I/O 子系统中已检测到有一个不重要故障,如在与远程 I/O 站进行通信时出现的可恢复错误
3		BAD	指示在 I/O 子系统中已检测到有一个重要故障,如在与远程 I/O 站进行通信时有通信丢失的重要故障
4	No outputs disabled	如为 TRUE,此属性指示作为执行应用程序或其他方法的结果,PLC 可能更改连接特定 I/O 子系统的所有输出的物理状态。如不为“真”,某些输出的物理状态不受影响(可能影响逻辑状态)。这典型地用于测试和修改在 PLC 中的应用程序	
5	No inputs disabled	如为 TRUE,此属性指示作为执行应用程序或其他方法的结果,PLC 可能存取连接特定 I/O 子系统的所有输入的物理状态。如不为“真”,则 PLC 不能存取某些输入的物理状态。这典型地用于测试和修改输入可模拟的应用程序	
6	I/O forced	如为 TRUE,此属性指示该系统至少有一个 I/O 点已被强制。当一个输入点被强制时,应用程序将接收由 PADT 所指定的值以替代机器或过程的实际值。当一个输出点被强制时,机器或过程将接收由 PADT 所指定的值以替代由执行应用程序而生成的值	
注:由实现者提供“重要故障”和“不重要故障”的定义			

6.1.3 处理单元

PLC 提供其处理单元的状态信息如下:

表 4 处理单元的状态

序号	项 目	描 述
1	Health	此属性识别处理单元的 health 状态,实现者规定 GOOD、WARNING 和 BAD 有效的条件
2		
3		
4	Running	如为 TRUE,此属性指示至少已装载用户的一部分应用程序且该部分在处理单元的控制下
5	Local control	如为 TRUE,此属性指示本地越权控制是否被激活,如果是,则从网络控制处理单元的能力受到限制,例如这可能紧紧绑定于一个本地的键开关上的应用
6	No outputs disabled	如为 TRUE,此属性指示作为执行应用程序或其他方法的结果,处理单元能更改所有由处理单元控制的输出的物理状态。如不为 TRUE,某些输出的物理状态不会被 PLC 影响(可能影响逻辑状态),这典型地用于测试和修改在处理单元中的应用程序
7	No inputs disabled	如为 TRUE,此属性指示作为执行应用程序或其他方法的结果,处理单元能存取来自处理单元的所有输入的物理状态。如不为 TRUE,则 PLC 不能存取某些输入的物理状态。这典型地用于测试和修改输入可模拟的应用程序
8	User application present	如为 TRUE,此属性指示处理单元至少存在有一个用户应用程序
9	Forced	如为 TRUE,此属性指示至少有一个连接的单元中的变量已被强制,如一个变量已被强制,则应用系统将采用由 PADT 规定的数值以替代由执行正常程序生成的数值。

6.1.4 电源子系统

PLC 能提供有关电源子系统的任何状态信息,图 6 是设想的 PLC 电源结构。PLC 系统的电源要求及其行为特性在 GB/T 15969.1 和 GB/T 15969.2 中描述。

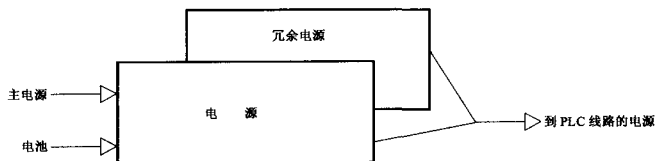


图 6 PLC 的电源

表 5 电源的状态

序号	项 目	描 述	
1	Health	GOOD	指示在电源中没有检测到供电电源不可保持运行的问题
2		WARNING	指示在电源中检测到有一个问题,它可能造成电源在一个限定的时间内不能运行
3		BAD	指示电源不能工作
4	In use	如为“TRUE,此属性指示电源子系统正在使用,也就是说它提供电源给 PLC	
5	Mains operating	如为 TRUE,此属性指示主电源在规定的电源范围内进行供电	
6	Mains low	如为 TRUE,此属性指示主电源没有在规定的电源范围内进行供电	
7	Battery operating	如为 TRUE,此属性指示电池在规定的电源范围内进行供电	
8	Battery low	如为 TRUE,此属性指示电池没有在规定的电源范围内进行供电	
9	Protection tripped	如为 TRUE,此属性指示在电源内的保护装置已断开到 PLC 的部分电源	

6.1.5 存储器子系统

PLC 提供其存储器子系统的状态信息如下:

表 6 存储器的状态

序 号	项 目	描 述	
1	Health	GOOD	在与此子系统有关的存储器中没有发现有错误
2		WARNING	至少检测到有一个可纠正的错误,没有检测到不能纠正的错误
3		BAD	至少检测到有一个不能纠正的错误
4	Protected ¹⁾	如为 TRUE,此属性指示在存储器子系统 中的存储器已被保护,不能更改存储器的内容。这通常表明,在此存储器中的应用程序不能被修改	
1) 此属性是逻辑状态模型而不是子系统的物理特性,如存储器中的某些部分被保护而某些部分没有被保护,则此存储器被认为是多子系统的存储器			

6.1.6 通信子系统

表 7 通信子系统的状态

序号	项 目	描 述	
1	Health	GOOD	指示没有故障,或发生可恢复的故障数量在允许范围内
2		WARNING	指示发生能恢复的故障数量超过允许范围
3		BAD	指示通信子系统不能和所有意图与之通信的设备进行通信
4	In use	如为 TRUE,此属性指示通信子系统当前正在运行。例如,在 MMS 通信接口的情况下,这意味着至少建立了一个应用连接。此外实现者应定义此属性的语义	
5	Local error	如为 TRUE,此属性指示通信子系统内部存在有某些故障,它禁止通信子系统的运行	
6	Remote error	如为 TRUE,此属性指示正在与之通信的设备中有某些故障,它禁止通信子系统的运行	
注 1: 正在报告其状态的通信子系统也许不能按照本节定义的方法报告本身的 BAD 状态。但是,在 PLC 内,能运行几个独立的通信子系统,他们均能提供状态信息			
注 2: 要求实现者特指的信息提供关于特殊接口的附加信息。ISO 网络接口还通过网络管理功能提供附加信息			

6.1.7 实现者特指的子系统

PLC 的其他子系统是实现者特指的子系统模块,这些子系统的举例如下:

- a) PID 控制器;
- b) 运动控制器;
- c) 其他辅助处理器等。

表 8 实现者特指的子系统

序 号	项 目	描 述	
1	Health	GOOD	指示在此子系统中没有检测到故障
2		WARNING	指示在此子系统中已检测到 一个不重要的故障
3		BAD	指示在此子系统中已检测到 一个重要故障
注：由实现者提供“不重要故障”和“重要故障”的定义			

6.1.8 状态信息的表达

状态信息使用变量表达,这些变量具有在 PLC 应用程序组态声明中预定义的存取路径。状态信息也可以表达为对远程通信伙伴直接表达式的变量。

表 9 状态信息的表达

序号	状态信息的表达
1	PLC 的摘要状态信息是带预定义存取路径 P-PLCSTATE 的变量
2	PLC 的摘要状态信息是带直接表达式 %S 的变量
3	PLC 的摘要状态信息和所有子系统的状态是带预定义存取路径 P-PLC STATUS 的变量
4	每个子系统的状态信息是带直接表达式 %SC<n> 的一个变量集
5	每个子系统的类型是带直接表达式 %SU<n> 的一个变量集
6	每个子系统的名称是带直接表达式 %SN<n> 的一个变量集
7	每个子系统的状态是带直接表达式 %SS<n> 的一个变量集
8	每个系统的实现者特指的状态是带直接表达式 %SI<n> 的一个变量集

如果 PLC 的摘要状态信息由一个变量表示,则该变量具有在组态声明中预定义的存取路径 P-PLCSTATE。此变量为 WORD 类型,并包含从最低有效位项目号 1 开始的 PLC 摘要状态信息。

如果 PLC 的摘要信息由带直接表达式的一个变量表示,则该直接表达式的形式为 %S 且为 WORD 类型,并包括从最低有效位项目号为 1 开始的 PLC 摘要状态信息。

如果全部状态信息以一个变量表示,则该变量具有在组态阶段中预定义的存取路径 P-PLCSTATUS,其结构类型如图 7:

```

ARRAY [0..P_NOS] OF
  STRUCT
    SUBSYSTEM; (SUMMARY, IO, PU, POWER, MEMORY, COMMUNIC
      ION, IMPLEMENTER);
    NAME; STRING[<Max_Name_Len>];
    STATE; ARRAY[0..15] OF BOOL;
    SPECIFIC; ARRAY[0..P_BIT] OF BOOL;
  END_STRUCT;

```

图 7 状态信息的类型描述

带号码 0 的数组元素包含 PLC 的摘要状态信息,每个有较高号码的元素包含一个子系统的状态。子元素 SUBSYSTEM 包括 PLC 或一个子系统的类型。子元素 NAME 包括 PLC 或一个子系统的名称。实现者应规定所支持的名称字符串的最大长度,例如,Max_Name_Len 的值。子元素 STATE 包括 PLC 或一个子系统状态信息,该状态信息是一个 BOOL 数组,其排列次序如同表 2~表 8 中的规定。实现者应指定数组 P-PLCSTATUS 的元素数量,即 P_NOS 的数值、支持的子系统类型、实现者特指的子系统中子元素 STATE 数值的语义、子元素 SPECIFIC(例如 P_BIT 的数值)的大小以及子元素 SPECIFIC 的语义。

当 <n> 为 0(表示 PLC 的摘要状态信息)和子系统 P_NOS 数值之间的一个数值时,每个子系统的状态信息为一个带直接表达式 %SC<n> 的变量。该变量与图 7 中描述类型的结构部分类型的变量有相同的内部表达式。

此外,可能存在有直接表达式 %SU<n>、%SN<n>、%SS<n> 和 %SI<n> 的变量集。<n> 表示为 0(表示 PLC 的摘要状态)和子系统数目 P_NOS 数值之间的一个数值。该变量与图 7 中描述类型的一个结构子元素类型的变量有相同的内部表达式。详细地说,也就是 %SU<n> 对应于子元素 SUBSYSTEM, %SN<n> 对应于子元素 NAME, %SS<n> 对应于子元素 STATE, 和 %SI<n> 对应于子元素 SPECIFIC。

6.2 应用特指的功能

本条文描述 PLC 利用通信子系统提供给控制系统的功能,如图 2 中所述。

PLC 通信功能	PLC 作为请求方	PLC 作为响应方	功能块可用
设备检验	是	是	是
数据采集	是	是	是
控制	是	是	是
用户应用程序之间的同步	是	是	是
报警报告	是	否	是
程序执行和 I/O 控制	否	是	否
应用程序传送	否	是	否
连接管理	是	是	是

以上每种情况在本条文中均单独予以说明。并不是在所有的 PLC 中都提供所有功能,功能块的定义见第 7 章。

有某些应用是以下定义的分类应用的组合,例如监控和数据采集。

以下元素,它通常由 PLC 提供,但不在本标准范围之内:

- a) 操作员接口;
- b) 编程、测试和修改应用程序以及程序检验。

PLC 具有使用操作员接口设备的能力,由操作员应用这些设备以监视或修改被控过程或者监视并修改被控过程。客户机亦能应用操作员接口与操作员进行通信。

直接操作员接口是指客户机不需要应用程序交互作用就能经由通信系统与操作员接口直接进行通信。

编程是基于一条指令接着一条指令或一个功能块接着一个功能块的建立 PLC 应用程序的过程。测试和修改是在一个已有的应用程序中寻找存在的错误和通过更改以纠正错误(调试)的过程。程序检验是测试 PLC 应用程序以验证它能在过程环境中执行设计所要求的功能。

6.2.1 设备检验

提供此功能是为了允许由其他设备确定 PLC 是否能在自动化系统中执行预期的功能。PLC 能提供自身及其子系统的状态。状态包括 Health 和 State 信息。一个设备可明确地向 PLC 请求其状态或 PLC 可使用通信接口所提供的服务启动一个未经请求的状态报告。关于 PLC 及其子系统的 health 和 state 信息的定义参阅 6.1。

表 10 设备检验特性

序号	设备检验
1	提供状态信息
2	启动非请求的状态报告

6.2.2 数据采集

包含在 PLC 中的数据表现为变量。这些数据可能有各种不同的来源并具有广泛的含义,客户机可以通过以下多种方法之一获得这些数据。

a) 轮询——在客户机确定的时间或条件下,由客户机读取一个或多个变量。通过 PLC 可以控制存取变量。在网络上只有被选择的变量是可存取的。

b) 编程——在由 PLC 应用程序确定的时间或条件下提供其数据给客户机。

c) 组态——客户机可以对 PLC 的通信接口进行组态,以启动到客户机的数据传输。

在 PLC 内对通信系统为可视的变量类型为:

- a) 具有直接表达式的变量。

b) 具有存取路径(参阅 GB/T 15969.3 中关于存取路径的定义)的其他变量。

如果直接表达式的变量对通信是可存取的,则这些变量将使用直接表达式作为标识符,PLC 服务器(即,持有这些变量的 PLC)可应用一个由实现者定义的算法对标识符作出解释。

注:当编制一个应用程序时,具有直接表达式的变量能够像“正常”变量那样使用。应用在变量申明中的 AT 结构,能将一个附加的符号名赋与有直接表达式的变量(参阅 GB/T 15969.3)。

一般而言,甚至在一个较小型的 PLC 中也会有几千的带直接表达式的变量。因此在 PLC 的对象词典中保持所有这种变量的名称和地址是不合理的。

PLC 系统有可能限制存取具有直接表达式的变量。实现者应指定由 PLC 支持的每一个数据类型存取不可中断的条件(例如大小,地址等)。

表 11 数据采集特性

序号	数 据 采 集
1	可存取的带直接表达式的变量
2	在配置层上的存取路径
3	在程序层上的存取路径
4	对存取带直接表达式变量进行限制的方法
5	不可中断地存取变量的条件

6.2.3 控制

PLC 支持两种控制方法:参数和互锁

参数控制是指 PLC 的运行通过写入值到驻留在 PLC 中的变量来控制。改变运行与否由应用程序或者由其他本地机制决定。

对变量的存取由持有此变量的 PLC 来控制。仅仅那些在存取路径说明中所选择的具有 READ_WRITE 限定词的变量才能在网络上存取并用于参数控制。

互锁控制是指客户机请求服务器执行一个应用操作并将操作的结果通知客户机。这种服务有二个方面,一个是客户机和服务器的同步化,另一个是它们之间的数据交换。

在互锁控制中,此数据交换发生在应用程序的同步点。这种服务用于对从一个应用程序到另一个应用程序的远程过程调用施加影响,图 8 中的时间线说明这一点。

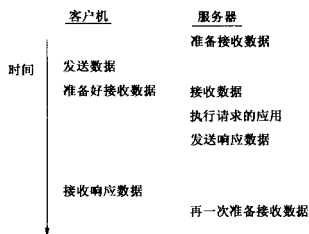


图 8 互锁控制的时间线

PLC 使用 SEND(客户机)和 RCV(服务器)功能块实现互锁控制。其他设备可以采取其他的方法仿效这些功能块的行为特性以存取此 PLC 的通信功能。

表 12 控制性能

序号	控 制
1	可存取带直接表达式的变量
2	在组态层上的存取路径
3	在编程层上的存取路径
4	对存取带直接表达式变量进行限制的方法
5	不可中断地存取变量的条件
6	互锁控制

6.2.4 用户应用之间的同步

用户应用之间可能需要同步服务。例如,一个用户应用在完成一个算法后可以启动另一个应用的执行,由互锁控制机制提供同步服务(参阅 6.2.3)。

6.2.5 报警报告

当一个预定义的条件发生时,PLC 能够向客户机发出报警报文。客户机将给 PLC 指示一个这些报警报文的确认。这和通常的数据采集不同,PLC 存储报警点的状态一直到由客户机确认为止。在客户机请求时,PLC 可生成非确认的报警报文摘要。

表 13 报警报告的性能

序号	报警报告
1	发送报文
2	接收确认
3	生成非确认报警报文的摘要

6.2.6 应用程序执行和 I/O 控制

由 Application Program Execution(应用程序执行)功能管理 PLC 应用程序的执行(参阅 5.2)。可以启动和停止 PLC 的应用程序。PLC 应用程序能由初始状态启动或从程序停止时的状态启动。

在 PLC 系统内的 PLC 应用程序包括一个配置和零个、一个或多个资源(参阅 GB/T 15969.3)。可以启动和停止配置和资源。当启动和停止配置时启动和停止资源,资源亦可独立于配置进行启动或停止。

对于与正在运行的应用程序有关的执行器(输出)的接口功能,可以被指向使用应用程序提供的值或者保持为一个已知的状态。在应用程序的状态改变时指定接口的状态。输出可以被指向设置为实现者所规定的状态、保持输出为当前的状态、或设置所有输出为零、或通过实现者指定的机制(如,表格,PLC 过程等)更改某些输出为用户指定的状态(接通或断开,对那些没有规定的则保持上一状态的输出)。

对于与正在运行的应用程序有关的传感器(输入)的接口功能,可以被指向提供来自传感器的实际值、或者继续使用以前提供的值。在应用程序状态改变时指定输入状态(参阅表 14)。

表 14 可启动和可停止的单元

序号	可启动和可停止的单元
1	配置
2	资源

与正在运行的资源有关的 I/O(输入或输出)将由程序控制或者保持为一个已知的状态,该状态是在资源被启动时决定的。资源的启动可以由初始状态(应用 START 的冷启动)启动或者从它们停止时的状态启动(应用 RESUME 的热启动)。输出所需要的状态可规定为停止过程的部分。

当配置或资源启动或停止时,I/O 的状态可设定为以下的值(参阅表 15、表 16、表 17):

表 15 I/O 状态的含义

I/O 状态的值	含 义	由... 设置
被控制的	执行器正在由应用程序或由正在启动的部分应用程序控制。正在将输入提供给应用程序或由传感器/执行器接口功能启动的部分应用程序	启动
保持输出	执行器不由正在运行的应用程序或正在启动的部分应用程序控制,它们保持为当前的状态。正在将传感器提供给应用程序或由传感器/执行器接口功能启动的部分应用程序	启动
保持当前状态	执行器不由正在运行的应用程序或由正在启动或停止的部分应用程序控制,它们保持为当前的状态。没有将传感器提供给正在运行的应用程序或正在启动的部分应用程序,他们保持为当前的状态	启动,停止
实现者指定的状态	执行器不由正在运行的应用程序或正在停止的应用程序部分控制,它们保持为由实现者指定的一个状态。应用程序没有在运行,因此没有规定传感器接口的状态	停止
零输出	执行器不由正在运行的应用程序或由正在停止的应部分用程序控制,它们保持为零状态。应用程序没有在运行,因此没有规定传感器接口的状态	停止
用户指定	执行器不由正在运行的应用程序或正在停止的部分应用程序控制,它们保持为由用户规定的状态。应用程序没有在运行,因此没有规定传感器接口的状态	停止

表 16 I/O 状态

序 号	I/O 状态
1	被控的
2	保持输出
3	保持当前状态
4	实现者指定的
5	零输出
6	用户规定
注: 通信子系统不应依赖作为硬件紧急停止切换时的替代,应该遵守通常的安全规则	

表 17 执行和 I/O 控制性能

序 号	执行和 I/O 状态
1	接收启动一个可启动和可停止单元的请求
2	接收停止一个可启动和可停止单元的请求
3	接收继续运行一个可启动和可停止单元的请求

6.2.7 应用程序传送

应用程序的传送是采用 PLC 的应用程序存储和数据存储功能来实现的(见 5.2)。应用程序传送允许客户上载可编程存储器的全部或一部分内容用于存档和检验,或者下载它们从而把 PLC 复原到已知的状态。这种功能还提供在修改可编程存储器的内容前把 PLC 置于安全状态的能力,以及当完成应用程序传送后以安全方式再启动 PLC 的能力。

程序传送的开始典型地是通过一个非 PLC 的设备来完成的。这些服务包括:

- a) 用于存档的上载;
- b) 用于检验的上载;
- c) 用于复原已知良好系统的下载;
- d) 下载离线开发的系统。

表 18 列出了能被上载或下载的可编程存储器的部分。

注: 装载单元包含 P_DDATE 变量。P_DDATE 变量的值是最后修改装载单元的时间。

表 18 可装载单元

项号	可装载单元
1	配置
2	资源
3	程序
4	全局变量
5	配置层的存取路径
6	程序层的存取路径
7	包含有 P_DDATE 变量的装载单元

实现者应指明其他的语言元素,如功能类型或功能块类型是否是可装载的,以及上载和下载这些语言元素的条件和限制。

本标准定义在整个 PLC、PLC 的子系统以及子系统的一部分实现上载和下载的一种方法。要下载整个 PLC 或 PLC 的一部分在下载前应明确地被停止。正在被下载的系统或系统的一部分直到下载完成后才能被其他设备使用。实现者需指明当一个客户机正在下载时,其他客户机能用 PLC 做什么。

表 19 应用程序传送特性

项号	应用程序传送
1	接收下载可装载单元的请求
2	接收上载可装载单元的请求

6.2.8 连接管理

连接管理提供安装、维护以及关闭与远程通信方之间的通信连接的手段。如果启动多重请求去操作一台设备,则这些请求可以通过相同的连接进行工作。并不是所有的通信子系统都需要建立连接,如点对点通信。

连接是由应用程序采用 CONNECT 功能块来显性地控制的,或在需要时由通信子系统来提供。

表 20 连接管理性能

项号	连接管理
1	安装连接
2	关闭连接
3	为多重请求使用一个连接

7 PLC 通信功能块

7.1 通信功能块概述

PLC 的通信功能及其相应的功能块描述如表 21:

表 21 通信功能块概述

项 号	子 条 文	通信功能块或功能名称
1	7.2 通信 FB 参数的语义(远程变量寻址)	REMOTE_VAR
2 3	7.3 设备检验	STATUS USTATUS
4	7.4 轮询数据采集	READ
5 6 7 8	7.5 编程数据采集	USEND URCV, BSEND, BRCV
9	7.6 参数控制	WRITE
10 11	7.7 互锁控制	SEND RCV
12 13	7.8 编程报警报告	NOTIFY, ALARM
14	7.9 连接管理	CONNECT

表 21 中给出的编号用于对这些通信功能块(CFB)的状态符合。

7.1.1 设备检验

STATUS 和 USTATUS 功能块使 PLC 能从其他设备获取状态信息。提供这些功能块的目的是允许 PLC 决定其他设备是否能够在自动系统中执行预期的功能。

7.1.2 数据采集

在其他设备内包含的数据可以表示为变量。该数据可来自多个资源并具有广泛的意义。PLC 可采用通信功能块两种方法中的一种得到该数据。

a) 轮询——在由 PLC 应用程序确定的某个时间或条件下,PLC 使用 READ 功能块获得一个或多个变量。变量的存取由正在读取的设备控制。

b) 编程——在由其他设备确定的某个时间或条件下提供此数据给 PLC。PLC 使用 URCV 功能块将此数据提供给 PLC 应用程序。PLC 使用 USEND 将未经请求的数据提供给其他设备。

其他设备支持的每一数据类型能被无中断地存取,其条件(大小,位置等)是由其他设备决定的。

7.1.3 控制

PLC 支持两种控制方法:参数控制和互锁控制。

参数控制是指其他设备的运行通过写入值到驻留在其中的变量来控制。变量的存取是由持有变量的 PLC 控制的。PLC 采用 WRITE 功能块执行来自 PLC 应用程序的动作。

互锁控制是指客户机请求服务器执行一个应用操作并且把该操作结果通知客户机。PLC 采用 SEND 和 RCV 功能块来分别担当客户机和服务器的角色。

7.1.4 报警报告

当一个预定条件发生时 PLC 具有向客户机发送报警报文的能力。客户机可指示这些报警的确认给 PLC。ALARM 和 NOTIFY 功能块可被 PLC 应用程序用做产生确认和非确认报警。

7.1.5 连接管理

PLC 应用程序用 CONNECT 功能块管理连接。

7.2 通信 FB 参数的语义

通信功能块采用共同的功能块输入和输出语义。这些功能块的输入和输出的含义描述如表 22。某些通信功能块有特殊的输入或输出参数,它们在描述通信功能块本身的地方进行描述。

表 22 通信 FB 参数的语义

参数名称	参数的数据类型	注释
EN_R	BOOL	启用接收数据
REQ/RESP	BOOL	在上升沿执行功能
ID	COMM_CHANNEL	通信通道的标识
R_ID	STRING	通道内的远程功能块的标识
SD i	ANY	要发送的用户数据
VAR_i	STRING 或 功能输出的数据类型 REMOTE_VAR	远程通信伙伴的变量的标识
DONE	BOOL	请求的功能被执行(良好且有效)
NDR	BOOL	接收到新的用户数据(良好且有效)
ERROR	BOOL	接收到新的非零状态
STATUS	INT	上次探测到的状态(错误或良好)
RD_i	ANY	上次接收到的用户数据

ID 输入引用由通信功能块实例所使用的通信通道,即它确定远程通信伙伴。ID 输入是 COMM_CHANNEL 数据类型,它是由实现者定义的。

注:由通信功能块实例的 ID 输入的给定值企图保持或引用对管理与远程通信伙伴进行通信的必需的信息。该信息可能依赖于实施和使用的通信子系统。

如果 PLC 通信功能块由相应的一对功能块实例提供,则 R_ID 输入用于标识远程通信伙伴的通信功能块的相应实例。

一个通信功能块实例应使用相同的通信通道,并且在它的整个生存期内与同一个相应的远程功能块实例进行通信。

要读取或写入的变量使用 READ 和 WRITE 功能块的 VAR_i 输入来标识。实际参数的典型情况为一个包含远程变量名的字符串。

此外 VAR_i 参数还可以有一个由实现者定义的名为 REMOTE_VAR 的数据类型。REMOTE_VAR 功能被定义用于为嵌套变量产生存取信息。(参阅图 9)

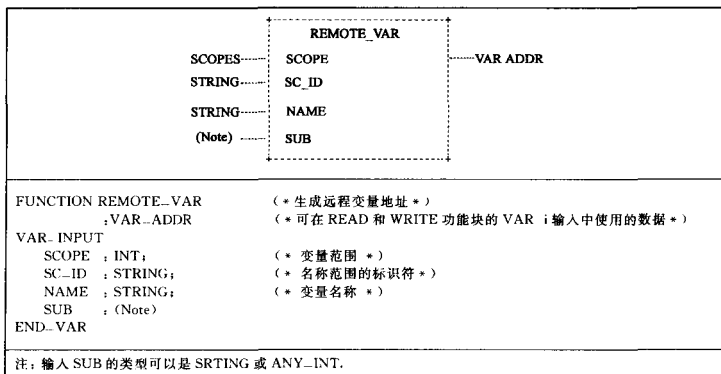


图 9 REMOTE_VAR 功能

SCOPE 是一个整数,它标识 GB/T 15969.3 编程语言的名称范围、通信系统、或支持作为远程变量范围的实现者。(参阅表 23)

表 23 SCOPE 参数的值

GB/T 15969.3 的名称范围	值
配置	0
资源	1
程序	2
功能块实例	3
保留用作将来的标准化	4...9
保留用作通信子系统的专用名称范围	10...99
实现者特指的	<0,>99

如果变量的 SCOPE 需要一个标识符,SC_ID 就被用作提供该标识符。NAME 包含有远程变量的名称。如果 SUB 是字符串类型,那么字符串的值就被解释为一子元素名字。如果 SUB 是 ANY_INT,那么该值就被解释为一索引。如果 SUB 是一空字符串,那么整个变量被编址。

功能输出的数据类型是由实现者定义的。功能的结果传送到 READ 和 WRITE 功能块的 VAR_i 输入或存储在一个有相同数据类型的变量中。REMOTE_VAR 功能的使用仅仅限制在产生 READ 和 WRITE 的功能块的 VAR_i 参数的有效数值。

可能有附加的功能来支持的通信子系统专用的、或支持产生 VAR_ADDR 类型输出的实现者特指的寻址模式。通信子系统特指的功能在本标准的附录 A 中规定。

除了依赖于功能块类型的可扩展的 SD_i、RD_i 和 VAR_i 参数外,所有的通信功能块参数都是强制性的。如果在一个功能块实例中有不止一个 SD_i 参数或如果 SD_i 参数由不同的功能块实例使用的话,则不要求 SD_i 或 RD_i 参数有相同的数据类型。实现者应指明 SD_i、RD_i 和 VAR_i 参数的数量,这些参数由一个通信功能块调用所支持。

此外,应描述是否对这些参数的使用有限制,例如,实际参数的数据类型或大小。

如果一个通信功能块要求数据类型是兼容的,并且如果相同的 IEC 61131-3 数据类型用在一个客户机 PLC 和服务器 PLC,则兼容性检查总为 TRUE。可以定义附加的通信子系统特殊兼容性规则。

输出被初始化为系统零。直到下一次调用该实例时,NDR、DONE 及 ERROR 才会变为 TRUE。这就是说,每一功能块隐含有如图 10 的结构,但没有显示在这些功能块的状态图中。

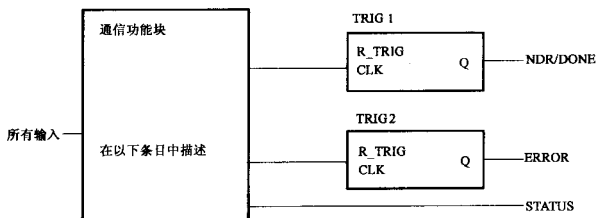


图 10 状态信号发送原理

如果 PLC 系统或通信功能块算法检测到一个通信功能块实例发生了通信错误,则功能块实例的 ERROR 及 STATUS 输出就会被置位。在该实例的两个调用期间内,ERROR 输出保持为 TRUE(见图 11)。

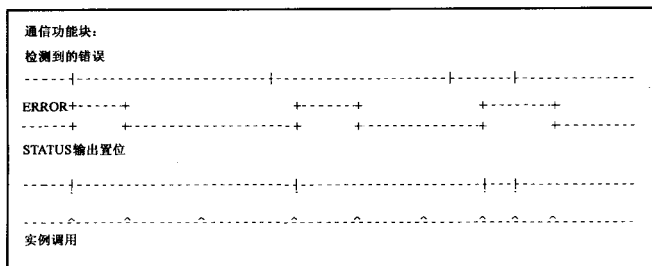


图 11 ERROR 和 STATUS 输出的时序图

仅在通信功能块实例调用的同步点, NDR、DONE、ERROR 和 STATUS 输出才被设定为新值, 即使在调用期间发生了错误, 也是这样。

表 24 定义各种功能块的 STATUS 输出值。并不是所有值都被所有功能块使用。

表 24 STATUS 的输出值及其说明

STATUS 值	解 释
0	无错误
1	低层错误,无通信可能
2	的来自远程通信伙伴的其他否定响应
3	在通信通道中不存在 R-ID
4	数据类型不匹配
5	接收到复位
6	接收器未启用
7	远程通信伙伴处于错误状态
8	对远程对象存取失败
9	接收器超出接收限度(新的用户数据)
10	存取本地对象被拒绝
11	请求的服务超出本地资源
12...20	保留用于将来的标准
-1	该功能的实例处于忙状态,在此时不能提供额外的服务
<-1 或 >20	小于-1 或大于 20 的值由执行者去定义

如果在一对多或一对全部的通信情况下,接收到的错误是来自一个或多个通信伙伴,则 STATUS 参数根据接收到的第一个错误来设置。

RD_i 参数包含接收到的数据。这些参数是输入/输出参数。

以下的子条文包含对通信功能块的描述。功能块的表达以图形和文本的方式给出,并且表示关联的输入和输出的类型及其意义。STATUS 输出具有所定义的合适的值。

正常运行由一个时序图来表示。

功能块运行是基于状态图来描述的。依赖于应用程序的许多转换被映象为功能块输入的条件。依赖于通信系统的转换独立于所使用的通信系统来进行描述。对某一通信系统的显性映象在附录 A 中描

述。功能块输出值对应于状态图的每一状态。

由通信系统或本地问题所引起的出错会异步发生在通信功能块的所有状态中。只有引起状态转换或要求动作的那些错误在状态图中才被清晰地描述。否则,将使用图 10 和图 11 中所示的 ERROR 和 STATUS 输出向应用程序发出出错信号。

通信功能块需要初始化。在功能块实例的第一次调用返回前,至少要保留包含在所有状态图内的初始化状态 INIT。在此状态下,应执行全部的动作以启用通信。如果没有显性地编程通道连接管理,则应建立连接到远程通信伙伴的通信通道。

7.3 设备检验

PLC 通信功能设备检验使用 STATUS 和 USTATUS 功能块。

STATUS 或 USATUS 功能块实例提供了 PLC 功能设备检验的实例。

PLC 可请求某一远程通信伙伴采用 STATUS 功能块将其状态发送返回给 PLC。

PLC 本身能够采用 USTATUS 功能块接收远程通信伙伴的状态信息。无论何时在 PHYS 和 LOG 输出中的远程通信伙伴的状态信息发生改变时,远程通信伙伴至少应通知 USATUS 实例。

FB 的输出 PHYS 包含远程设备的物理状态,FB 的输出 LOG 包含远程设备的逻辑通信状态。FB 的输出 LOCAL 可能包含有多达 128 位(bit)的附加状态信息。实现者应指定该附加信息的使用长度并定义其语义。

注: READ 功能块可以用于获得附加的状态信息。

ID 参数标识到远程通信伙伴的通信通道。

如果有错误发生,ERROR 输出产生一个周期的脉冲以指示错误,并且 STATUS 输出包含有此出错代码。(参阅图 12、图 13 和图 14)

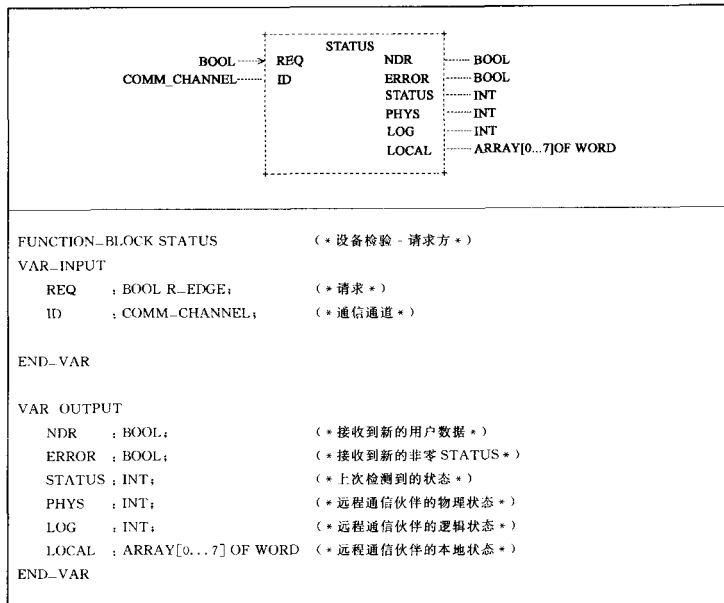


图 12 STATUS 功能块

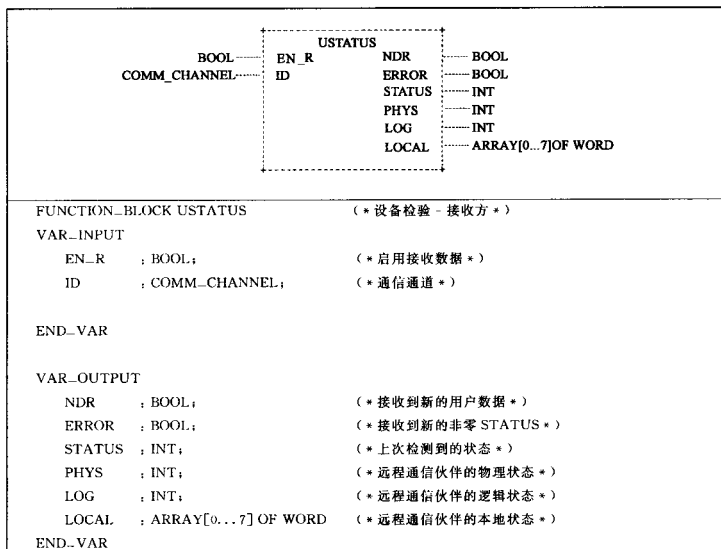


图 13 USTATUS 功能块

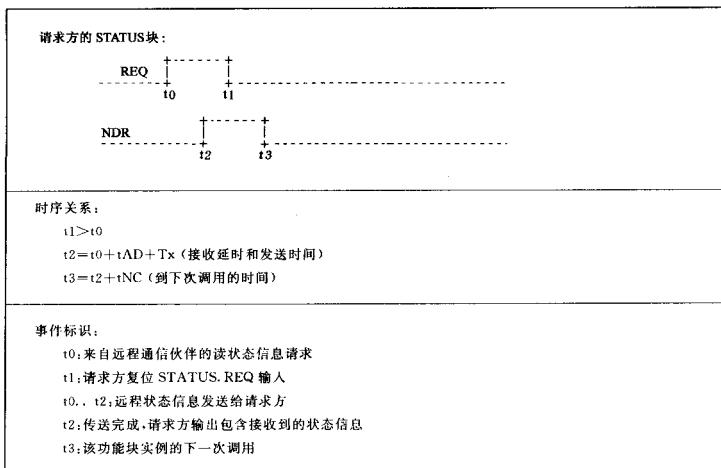


图 14 STATUS 功能块的时序图

图 15 所示的状态图描述了 STATUS 功能块的算法。表 25 和表 26 描述了该状态图的转换和在各个状态内执行的动作以及 STATUS 功能块输出的设置。

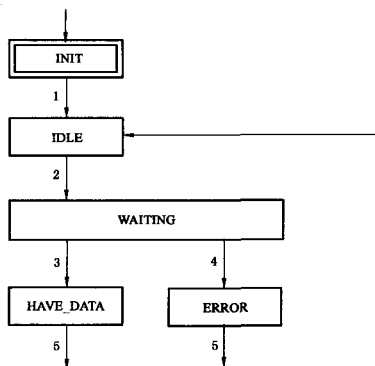


图 15 STATUS 功能块的状态图

表 25 STATUS 状态图的转换

转换	条 件
1	完成初始化
2	在 REQ 输入的上升沿
3	来自远程通信伙伴的肯定响应
4	来自远程通信伙伴的否定响应或检测到有通信故障
5	在该实例的下次调用之后

表 26 STATUS 状态图的动作表

状态	动作	FB 输出			
		NDR ^{c)}	ERROR ^{c)}	STATUS	PHYS, LOG, LOCAL
INIT ^{a)}	初始化输出	0	0	0	0
IDLE	无动作	0	0	—	—
WAITING	来自远程通信伙伴的状态信息请求	—	—	—1	—
HAVE..DATA	存放状态信息在实例中	1	0	0	新的状态信息
ERROR	指示出错	0	1	b)	—

— 指示 FB 输出“未变化”。

a) INIT 是冷启动状态。

b) 错误码放置在状态输出中。

c) 见图 10。

图 16 所示的状态图描述了 USTATUS 功能块的算法。表 27 和 28 描述该状态图的转换和在各个状态内执行的动作以及 USTATUS 功能块输出的设置。

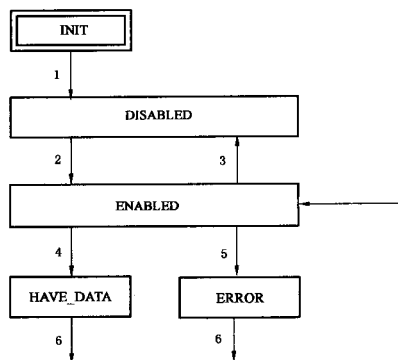


图 16 USTATUS 功能块的状态图

表 27 USTATUS 状态图的转换

转换	条件
1	完成初始化
2	EN_R=1
3	EN_R=0
4	从远程通信伙伴接收到状态信息
5	检测到通信问题
6	在该实例的下一次调用之后

表 28 USTATUS 状态图的动作表

状态	动作	FB 输出			
		NDR ^{c)}	ERROR ^{c)}	STATUS	PHYS, LOG, LOCAL
INIT ^{a)}	初始化输出	0	0	0	0
DISABLED	没有动作	0	0	---	---
ENABLED	没有动作	0	0	---	---
HAVE_DATA	存放状态信息	1	0	0	新的状态信息
ERROR	指示出错	0	1	b)	---

— 指示 FB 输出“未变化”。

a) INIT 是冷启动状态。

b) 错误码放置在状态输出中。

c) 见图 10。

7.4 轮询数据采集

PLC 通信功能块轮询数据采集使用 READ 功能块。

READ 功能块的一个实例提供 PLC 轮询数据采集功能的一个实例。

ID 参数标识到远程通信伙伴的通信通道。

READ 功能块的 VAR_i 输入包含了一个能被远程通信伙伴解释为变量标识符的字符串。远程通信伙伴将这些变量值发送返回给请求 READ 的实例。READ 功能块通过它的 RD_i 输出把接收到的变量值传送给它的应用程序。远程通信伙伴的每一个请求变量具有与在 READ 实例的 RD_i 输出中编程的变量有相兼容的数据类型。VAR_i 和 RD_i 参数是可扩展的。至少应存在 VAR₁ 和 RD₁。

如果远程通信伙伴是一台 PLC, 则具有存取路径的变量和有直接表达式的变量能通过 READ 功能块存取(参阅图 17 和图 18)。具有存取路径的变量可参见在 PLC 编程语言的 VAR_ACCESS 结构(见 GB/T 15969.3 的 2.7.1)。在 VAR_ACCESS 结构中指定的存取名用作 VAR_i 输入的变量标识符。如果用 READ 功能块对一个带有直接表达式的变量存取, 则 VAR_i 输入应包含此直接表达式, 例如字符串 %IW17。在一个 READ 功能块实例调用中, 具有存取路径的变量和有直接表达式的变量可以混合使用。

如果一个变量是通过在程序内说明的存取路径读取(参看 GB/T 15969.3 的 2.5.3), 则使用 REMOTE 功能。程序实例的名称应用在 SC_ID 输入中, 变量名称用在 NAME 输入中, 例如为了读取程序 DO7 中的变量 AB12, 则 REMOTE_VAR 功能以 REMOTE_VAR(2, "DO7", "AB12", "") 形式被调用。

如果读取一个结构变量子元素或一个数组元素, 则 REMOTE_VAR 功能的 SUB 输入用于识别 VAR_i 输入中的孩子元素或元素。

如果发生一个错误, ERROR 输出产生一个周期的脉冲以指示错误, 并且 STATUS 输出包含此出错代码。

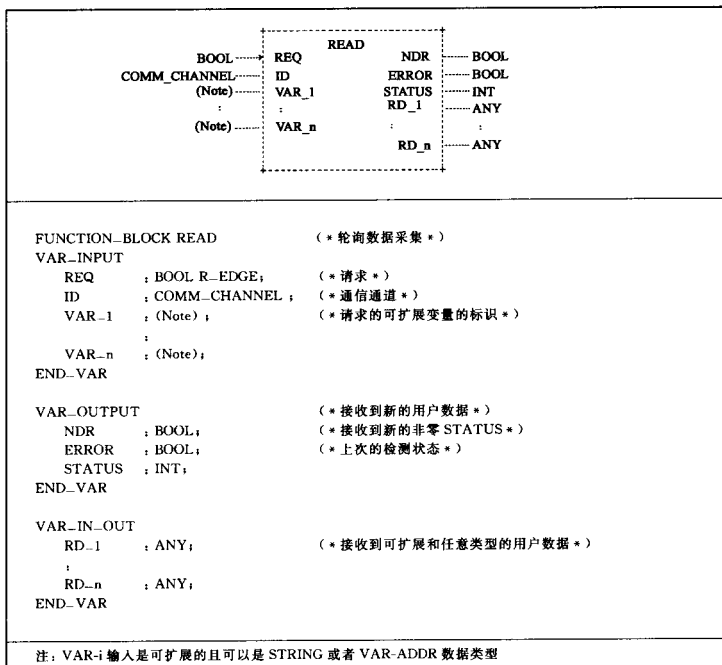


图 17 READ 功能块

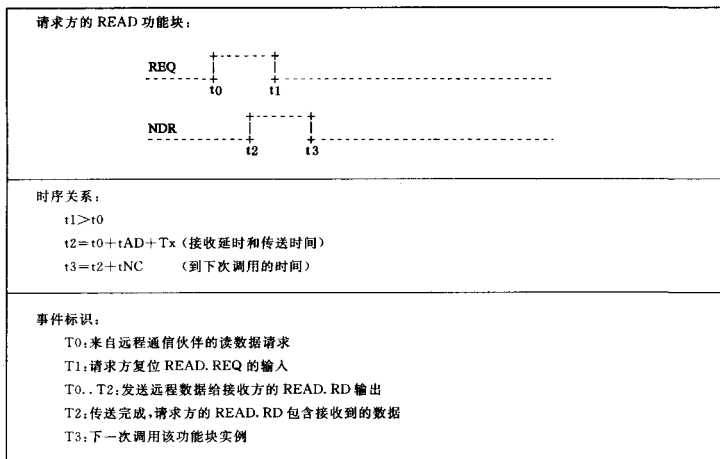


图 18 READ 功能块的时序图

图 19 所示的状态图描述了 READ 功能块的算法。表 29 和表 30 描述了状态图的转换和在各个状态内执行的动作以及 READ 功能块输出的设置。

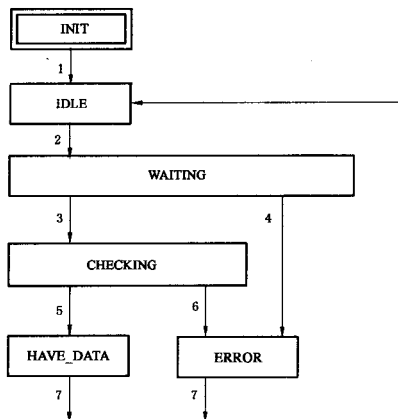


图 19 READ 功能块的状态图

表 29 READ 状态图的转换

转换	条 件
1	完成初始化
2	在 REQ 输入的上升沿
3	来自远程通信伙伴的肯定响应
4	来自远程通信伙伴的否定响应或检测到的其他通信问题

表 29(完)

转换	条 件
5	RD-i 数据类型和接收数据的数据类型相匹配
6	RD-i 数据类型和接收数据的数据类型不匹配
7	在该实例的下一个调用之后

表 30 READ 状态图的动作表

状态	动作	FB 输出			
		NDR ^{a)}	ERROR ^{a)}	STATUS	RD-1..RD..N
INIT ^{a)}	初始化输出	0	0	0	空系统值
IDLE	没有动作	0	0	—	—
WAITING	请求来自远程通信伙伴的变量	0	0	-1	—
CHECKING	检验数据类型匹配	0	0	—	—
HAVE-DATA	存放数据	1	0	0	新的数据
ERROR	指示错误	0	1	b)	—

—指示 FB 输出“未变化”。

a) INIT 是冷启动状态。

b) 错误码放置在状态输出中。

c) 见图 10。

7.5 编程数据采集

7.5.1 USEND/URCV 功能块

PLC 通信功能编程数据采集使用 USEND 功能块和 URCV 功能块。(参阅图 21、图 22 和图 23)

一个 USEND 功能块和一个 URCV 功能块的对应实例提供了一个 PLC 功能编程数据采集实例。

USEND 实例发送数据给 URCV 实例,URCV 实例可以用自己的应用程序处理该数据。当请求的 USEND 实例从它的 SD-i 输入中获取数据并传输给相应的 URCV 实例时,先前收到的数据被重写。URCV 实例经由它的 RD-i 输出传送接收到的数据给应用程序。无论何时应用程序请求它的 USEND 实例传送数据,这种情况都会发生。无论何时 URCV 实例只要得到一个新数据,它就会传送新接收到的数据。当收到新数据后,它通知应用程序。图 20 中的数据流图说明了这种情况。

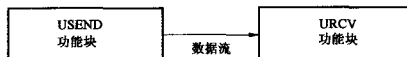


图 20 编程数据采集数据流

SD-i 和 RD-i 参数是可扩展的。至少在 FB USEND 中的 SD..1 输入和在 FB URCV 中的 RD..1 是存在的。USEND 实例的 SD-i 输入的数量及每个数据类型与对应的 URCV 实例的 RD-i 输出是兼容的。

一个 USEND 实例发送数据给一个 URCV 实例,这就是说,如果 ID 参数值引用相同的通信通道,并且 R_ID 参数值在该通信通道范围内相等,则它们是对应的实例。

注:如果系统提供的通信通道支持一对多或一对全部的连接,则这些功能块可用于对从一个通信伙伴到多个通信伙伴的数据采集功能进行编程。

如果一个错误发生,ERROR 输出产生一个周期的脉冲以指示错误,并且 STATUS 输出包含该错误码。

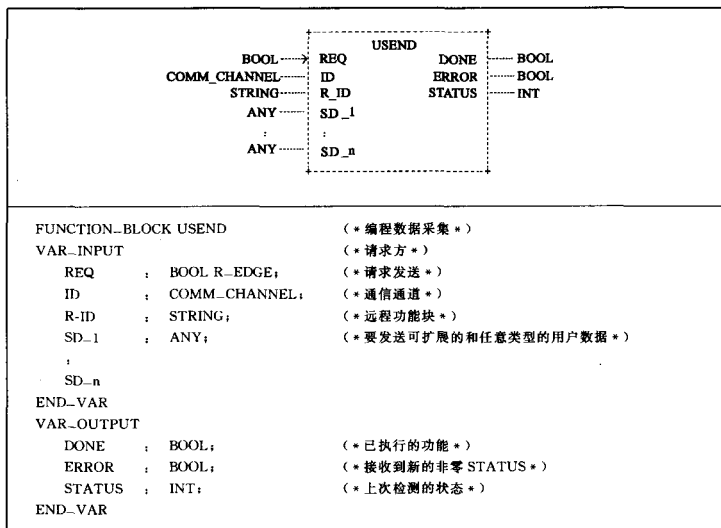


图 21 USEND 功能块

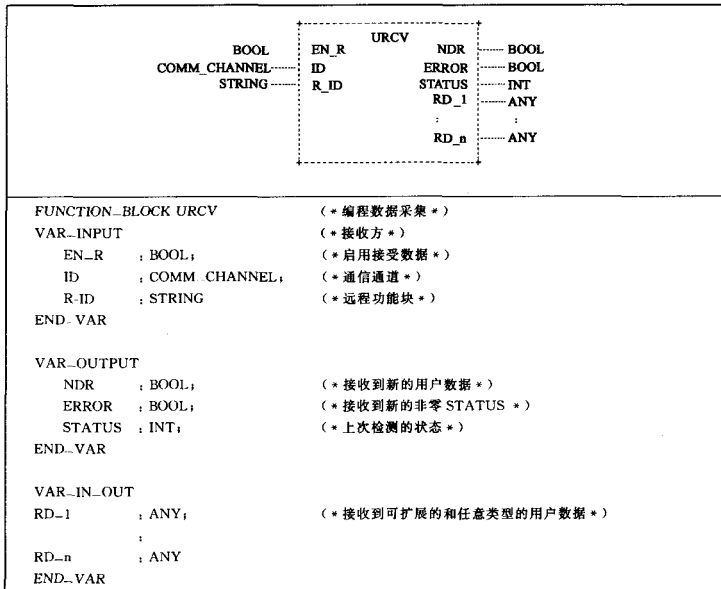


图 22 URCV 功能块

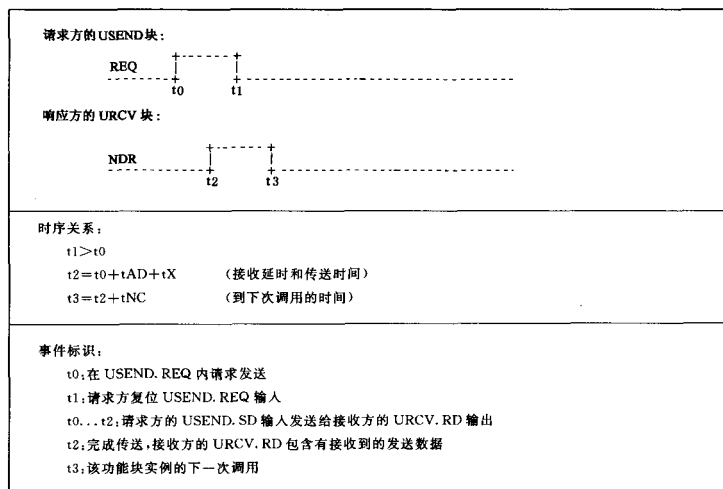


图 23 USEND 和 URCV 功能块的时序图

图 24 所示的状态图描述了 USEND 功能块的算法。表 31 和表 32 描述了状态图的转换和在各个状态内要执行的动作以及 USEND 功能块输出的设置。

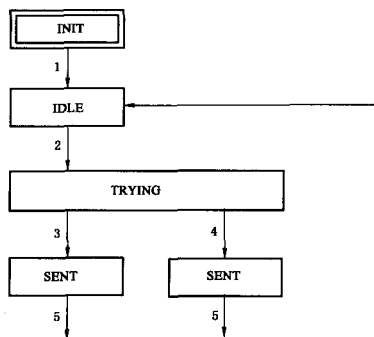


图 24 USEND 功能块的状态图

表 31 USEND 功能块状态图的转换

转换	条 件
1	完成初始化
2	在 REQ 输入的上升沿
3	通信系统指示“发送到远程通信伙伴”
4	通信系统指示“不能向远程通信伙伴发送”或检测到有其他通信问题
5	在该实例的下一个调用后

表 32 USEND 功能块状态图的动作表

状态	动作	FB 输出		
		DONE ^{a)}	ERROR ^{c)}	STATUS
INIT ^{a)}	初始化输出	0	0	0
IDLE	没有动作	0	0	—
TRYING	发送由 SD_i 输入给出的数据给远程通信伙伴	—	—	—
SEND	清除出错指示	1	0	0
ERROR	显示出错	0	1	b)

—指示 FB 输出“未变化”。

a) INIT 是冷启动状态。

b) 错误码放在状态输出中。

c) 见图 10。

图 25 所示的状态图描述了 URCV 功能块的算法。表 33 和表 34 描述了状态图的转换和在各个状态内要执行的动作以及 URCV 功能块输出的设置。

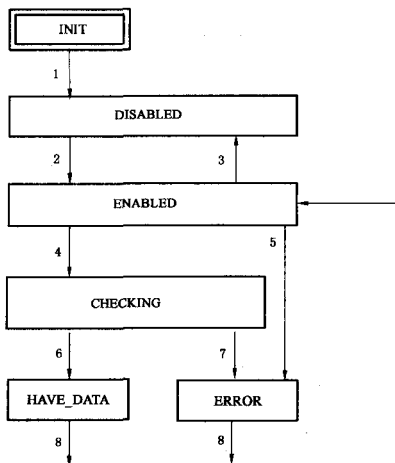


图 25 URCV 功能块的状态图

表 33 URCV 状态图的转换

转换	条 件
1	完成初始化
2	EN_R=1
3	EN_R=0
4	接收到来自远程通信伙伴的数据
5	检测到通信问题
6	USEND 的 SD_i 的数据类型和 URCV 的 RD-i 的数据类型相匹配
7	USEND 的 SD_i 的数据类型和 URCV 的 RD-i 的数据类型不匹配
8	在该实例的下一个调用之后

表 34 URCV 功能块状态图的动作表

状态	动作	FB 输出			
		NDR ^{a)}	ERROR ^{a)}	STATUS	RD-1...RD..n
INIT ^{a)}	初始化输出	0	0	0	空系统值
DISABLED	没有动作	0	0	—	—
ENABLED	没有动作	—	—	—	—
CHECKING	检验数据类型匹配	—	—	—	—
HAVE_DATA	存放数据	1	0	0,9	新的数据
ERROR	指示有出错	0	1	b)	—
—指示 FB 输出“未变化”。 a) INIT 是冷启动状态。 b) 错误码放置在状态输出中。 c) 见图 10。					

7.5.2 BSEND/BRCV 功能块

一个 BSEND 功能块和一个 BRCV 功能块的对应实例提供一个 PLC 编程数据采集功能的实例。(参阅图 26、图 27 和图 28)

BSEND 实例发送数据给 BRCV 实例,它可以用自己的应用程序处理此数据。当请求的 BSEND 实例从它的 SD-*i* 输入中取得数据并传输这个数据给相应的 BRCV 实例时,先前收到的数据被重写。BRCV 实例通过它的 RD-*i* 输出传输收到的数据给应用程序。此情况发生在应用程序要求它的 BSEND 实例发送时。当 BRCV 实例已完成了以前的请求并得到新数据时,它传送收到的新数据。当新数据到达后,它通知应用程序。

BSEND 实例的 SD-1 输入和 BRCV 实例 RD-1 输出都是 ANY 数据类型,并按照字节顺序进行解释。

注:在控制器中该数据的表达典型地依赖于具体实现。如果传送的数据类型不是字节数组的数据,则通信伙伴应在任何数据的解释上达成一致。这就限定了使用这些通信功能块的程序的互操作性。

BSEND 实例发送给 SD-1 输入中给出的数据的字节数应等同于其 LEN 输入中给出的数。如果 LEN 输入有 0 值,则传送 SD-*i* 输入给出的整个变量。RD-1 输出应能够存储要发送的数据。BRCV 实例的 LEN 输出应包含有 RD-1 输出中接收到的字节数。如果有以下情况,则将发出出错信号:

- 在 BSEND 实例的 LEN 输入中的值大于连接到 SD-1 输入变量的总字节长度;
- 或者接收到的数据的总字节长度大于连接到 BRCV 实例的 RD-1 输出变量的总字节变量长度。

BSEND 实例的数据传送周期起始于 REQ 输入的上升沿,当 DONE 或 ERROR 输出为 1 时结束。在一个周期内,要传送的全部数据块被完全传送后,DONE 输出将为 1。在数据传送期间内,执行者能限制包含有要传送数据的变量的存取。当 NDR 输出为值 1 时,BRCV 实例的 RD-1 输出是有效的。

一个 BSEND 实例传送数据给一个 BRCV 实例;这表明,如果 ID 参数值引用相同的通信通道,并且 R-ID 参数值在该通信通道内是相等的,则它们是相对应的实例。

如果在 BSEND 实例的 R 输入中检测到上升沿,则数据传送被取消。如果数据传送被取消,则 BRCV 实例的 ERROR 和 STATUS 输出被设置。此时,在 RD-1 和 LEN 输出中的值没有被定义。

如果有错误发生,ERROR 输出一个周期的脉冲指示有出错,并且 STATUS 输出包含有此出错代码。

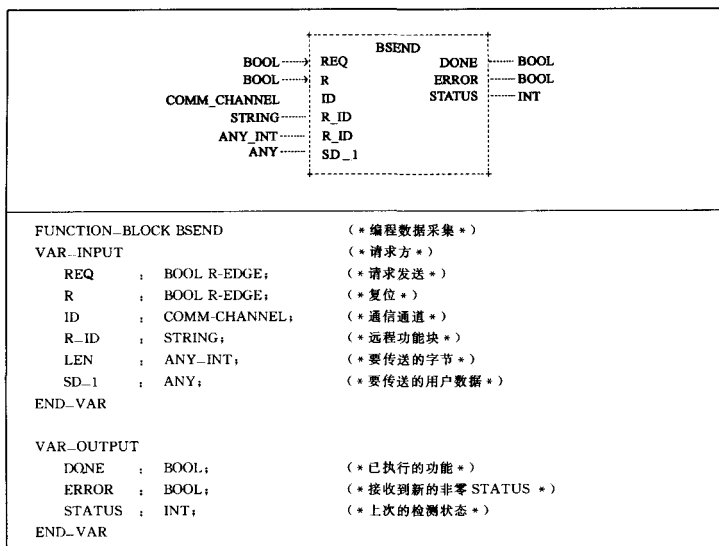


图 26 BSEND 功能块

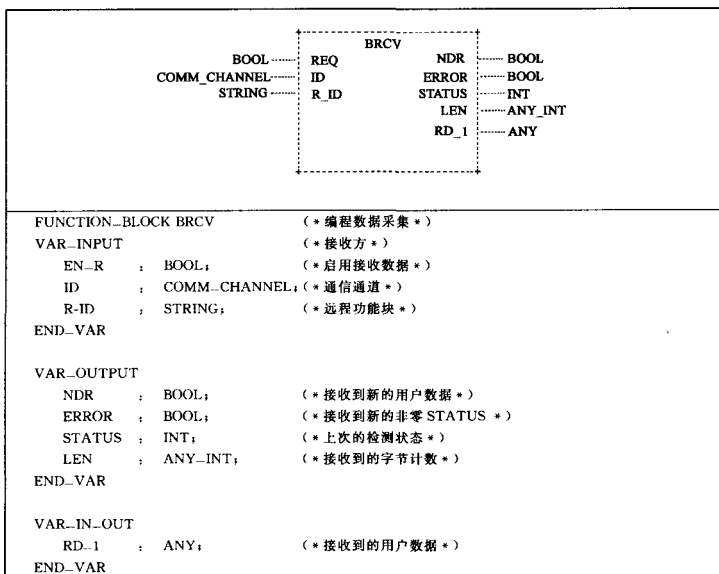


图 27 BRCV 功能块

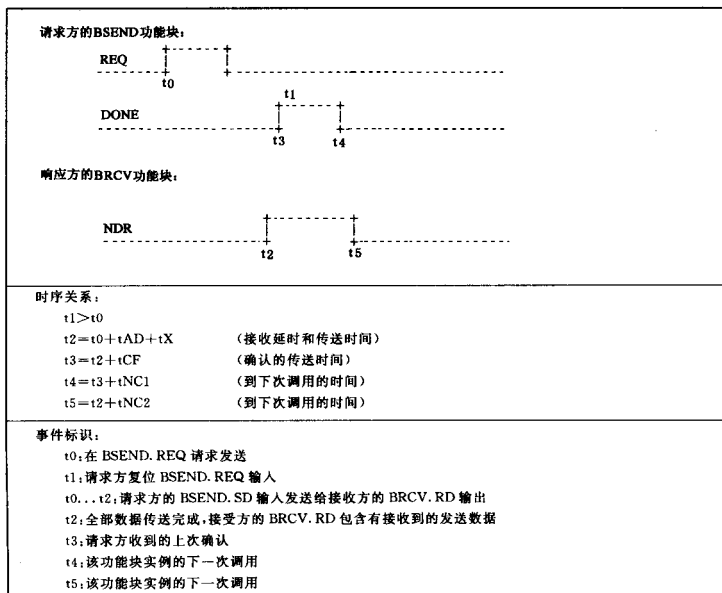


图 28 BSEND 和 BRCV 功能块的时序图

图 29 所示的状态图描述了 BSEND 功能块的算法。表 35 和表 36 描述了状态图的转换和在各个状态内要执行的动作以及 BSEND 功能块输出的设置。

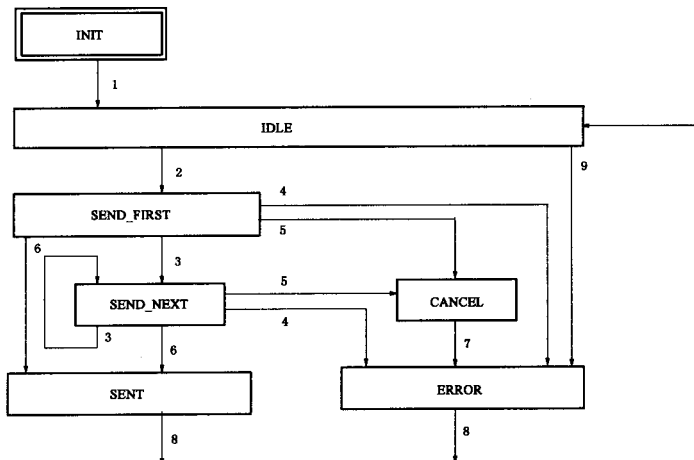


图 29 BSEND 功能块的状态图

表 35 BSEND 状态图的转换

转换	条 件
1	完成初始化
2	在 REQ 输入的上升沿
3	接收到肯定的确认,并有更多的数据要发送
4	接收到否定的确认或检测到有通信问题
5	在 R 输入的上升沿
6	接收到肯定的确认,没有更多数据要发送
7	即时
8	在该实例的下一调用之后
9	检测到有通信问题

表 36 BSEND 状态图的动作表

状态	动作	FB 输出		
		DONE ^{a)}	ERROR ^{a)}	STATUS
INIT ^{a)}	初始化输出	0	0	0
IDLE	没用动作	0	0	—
SEND_FIRST	发送由 SD_1 输入给出的第一个数据块到远程通信伙伴,发送最大总长度为 LEN 字节	—	1	—
SEND_NEXT	发送由 SD_1 输入给出的下一数据块到远程通信伙伴,发送最大总长度为 LEN 字节	—	—	—
SENT	清除出错指示	1	0	0
CANCEL	停止数据传送	—	—	—
ERROR	指示错误	0	1	b)
—指示 FB 输出“未变化”。 a) INIT 是冷启动状态。 b) 错误码放置在状态输出中。 c) 见图 10。				

图 30 所示的状态图描述了 BRCV 功能块的算法。表 37 和表 38 描述了状态图的转换和在各个状态内要执行的动作以及 BRCV 功能块输出的设置。

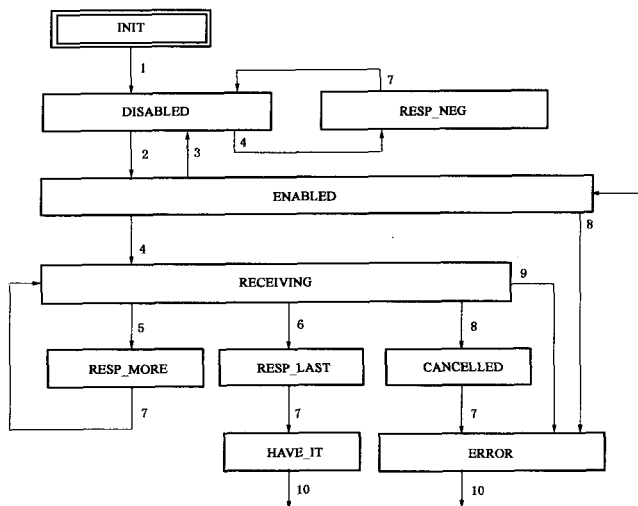


图 30 BRCV 功能块的状态图

表 37 BRCV 状态图的转换

转换	条 件
1	完成初始化
2	EN_R=1
3	EN_R=0
4	接收到来自远程通信伙伴的数据
5	跟随更多的数据为真
6	跟随更多的数据为假
7	即时
8	检测到的通信问题
9	接收到清除数据传送的指示
10	在该实例的下一调用之后

表 38 BRCV 状态图的动作表

状态	动作	FB 输出			
		NDR ^{a)}	ERROR ^{a)}	STATUS	RD-1, LEN
INIT ^{a)}	初始化输出	0	0	0	空系统值
DISABLED	没有动作	0	0	—	—
RESP_NEG	发送否定的响应	—	—	—	—
ENABLED	没有动作	—	—	—	—
RECEIVING	校验可存储的数据, 并储存在所给定的 索引中	—	—	^{a)}	新的数据 ^{d)}
RESP_MORE	发送肯定的响应	—	—	—	—
RESP_LAST	发送肯定的响应	—	—	—	—
CANCELLED	发送肯定的响应	—	—	5	—
HAVE_IT	存放数据	1	0	0	新的数据
ERROR	指示出错	0	1	^{b)}	
—指示 FB 输出“未变化”。 a) INIT 是冷启动状态。 b) 错误码放置在状态输出中。 c) 见图 10。 d) 在 STATUS 输出被设置为一1 的情况下,新数据被放置在 RD-1 输出中。					

7.6 参数控制

PLC 通信功能参数控制使用 WRITE 功能块。(参阅图 31 和图 32)

WRITE 功能块一个实例提供 PLC 功能参数控制的一个实例。

ID 参数标识到远程通信伙伴的通信通道。

WRITE 功能块的 VAR-i 输入包含一个能被远程通信伙伴作为其变量标识符(存取路径)的字符串。SD-i 输入引用所写人的变量值,该变量值是由 VAR-i 输入标识的。由远程通信伙伴将值写入到这些变量。VAR-i 和 SD-i 参数是可扩展的。至少 VAR-1 和 SD-1 应该是存在的。

远程通信伙伴的每一个变量应与在 WRITE 实例的 SD-i 输入中编程的变量有相同的数据类型。

如果远程通信伙伴是一台 PLC,则具有存取路径的变量和具有直接表达式的变量可以用 WRITE 功能块存取。具有存取路径的变量可参见 PLC 编程语言的 VAR_ACCESS 结构(见 GB/T 15969.3 的 2.7.1)。在该结构中指定的存取名称应作为 VAR-i 输入中的变量标识符使用。如果一个具有直接表达式的变量能用 WRITE 功能块存取,则 VAR-i 输入将包含有直接表达式的字符串,例如 %IW17。在一个 WRITE 功能块实例的调用中,有存取路径的变量和直接表达式的变量可以混合使用。

如果一个变量是通过在程序内部说明的存取路径写入,(见 GB/T 15969.3)则应使用 REMOTE_VAR 功能。在 SC_ID 输入中应使用程序实例的名称,在 NAME 输入中应使用变量名,例如,为了将变量 AB12 写入到程序 DO7 中,REMOTE_VAR 将调用 REMOTE_VAR 功能(2,“DO7”,“AB12”,“”)。

如果写入一个结构变量的子元素或一个数组元素,则使用 REMOTE_VAR 功能的 SUB 输入来标识该 VAR-i 输入的子元素或元素。

如果发生错误,ERROR 输出产生一个周期的脉冲以指示错误,并且 STATUS 输出包含有此出错代码。

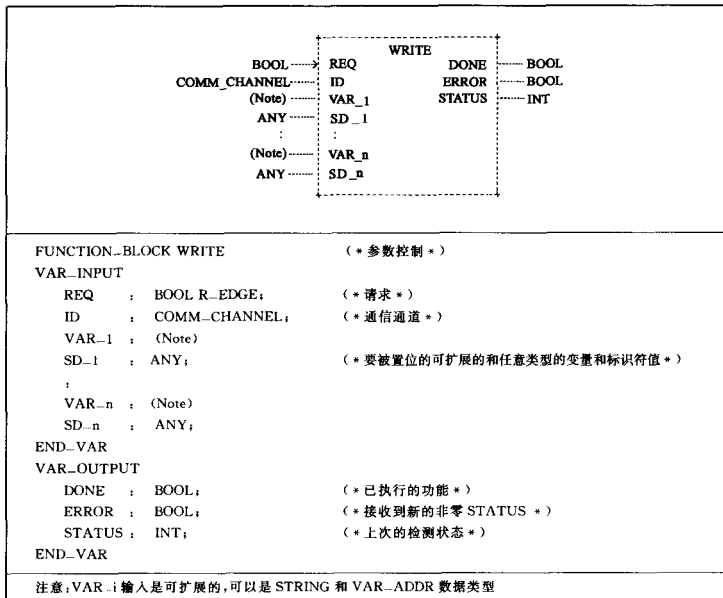


图 31 WRITE 功能块

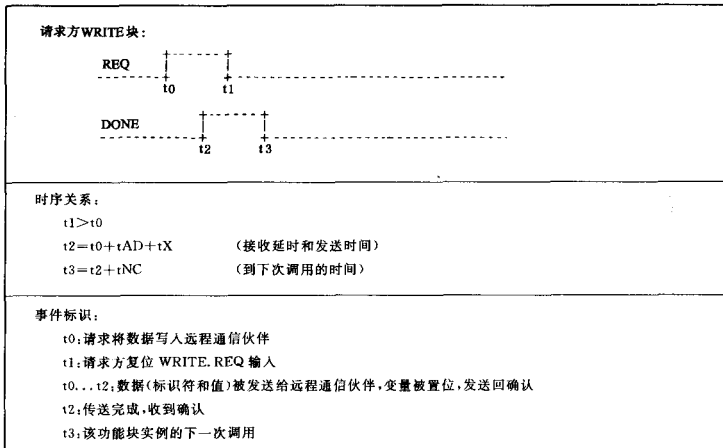


图 32 WRITE 功能块的时序图

图 33 所示的状态图描述了 WRITE 功能块的算法。表 39 和表 40 描述了状态图的转换和在各个状态内要执行的动作以及 WRITE 功能块输出的设置。

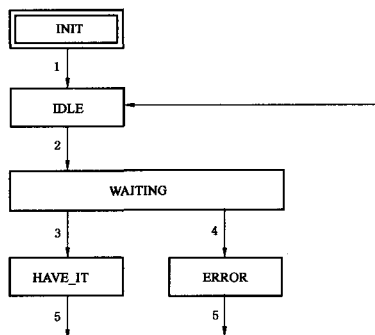


图 33 WRITE 功能块的状态图

表 39 WRITE 状态图的转换

转换	条 件
1	完成初始化
2	在 REQ 输入的上升沿
3	远程通信伙伴的肯定响应
4	远程通信伙伴的否定响应或检测到有其他通信问题
5	在该实例的下一调用之后

表 40 WRITE 状态图的动作表

状态	动作	FB 输出		
		DONE ^{a)}	ERROR ^{c)}	STATUS
INIT ^{a)}	初始化输出	0	0	0
IDLE	没有动作	0	0	...
WAITING	请求把变量写到远程通信伙伴	—	—	-1
HAVE_IT	指示成功	1	0	0
ERROR	指示出错	0	1	b)

—表示 FB 的输出“没有改变”。

a) INIT 是冷起动状态。

b) 出错代码放置在状态输出中。

c) 见图 10。

7.7 互锁控制

PLC 通信互锁控制功能使用 SEND 功能块和 RCV 功能块。(参阅图 34、图 35 和图 36)

相应的一个 SEND 和一个 RCV 功能块类型实例提供一个 PLC 互锁控制的功能。如果 ID 参数的值引用相同的通信通道并且 R_ID 参数的值在这个通信的通道范围内是相等的,则两个 SEND 和 RCV 功能块的实例是相对应的。

SEND 实例请求 RCV 实例执行一个应用操作并将操作结果通知给 SEND 实例。这里包括两个方

面,SEND 和 RCV 实例应用程序同步以及在它们之间交换信息。此功能可影响到从一个应用程序到另外一个应用程序的一个远程过程调用。

互锁控制功能是由 SEND 功能块实例的 REQ 输入的上升沿请求的。在请求时,SEND 实例从它的 SD_i 输入取得数据并且发送此数据到对应的 RCV 实例。

当 RCV 实例的 EN_R 输入的值为 1 时,它能够从对应的 SEND 实例接收到数据,并且执行应用程序的预期操作。当 RCV 实例从 SEND 实例接收到数据时,它通过它的 RD_i 输出传送接收到的数据到应用程序。RCV 实例的 NDR 输出产生一个周期的脉冲指示接收到新的数据。在执行应用程序的预期操作以后,通过 SD_i 输入取得结果数据,而且在 RCV 实例的 RESP 输入的上升沿启动响应。

当 SEND 实例接收该响应时,它发送接收到的数据到它的 RD_i 输出。SEND 实例的 NDR 输出产生一个周期的脉冲指示新数据准备就绪。

在 SEND 实例的 R 输入的上升沿复位 SEND 和相应的 RCV 实例。

SEND 实例的 SD_i 输入的每个数据类型和数量与对应的 RCV 实例的输出应是一致的。对 RCV 实例的 SD_i 输入和 SEND 实例的 RD_i 输出必需是相同的。SEND 和 RCV 功能块的 SD_i 输入和 RD_i 输出是可扩展的。发送数据或者响应数据或两者均可以是空,也就是说用户不需要编程任何 SD_i 输入和对应的 RD_i 输出。

如果接收到的数据不能匹配 RCV 功能块的 RD_i 输出或者发生一个错误,则 ERROR 输出产生一个周期的脉冲指示一个错误,并且 STATUS 输出含有此出错代码。

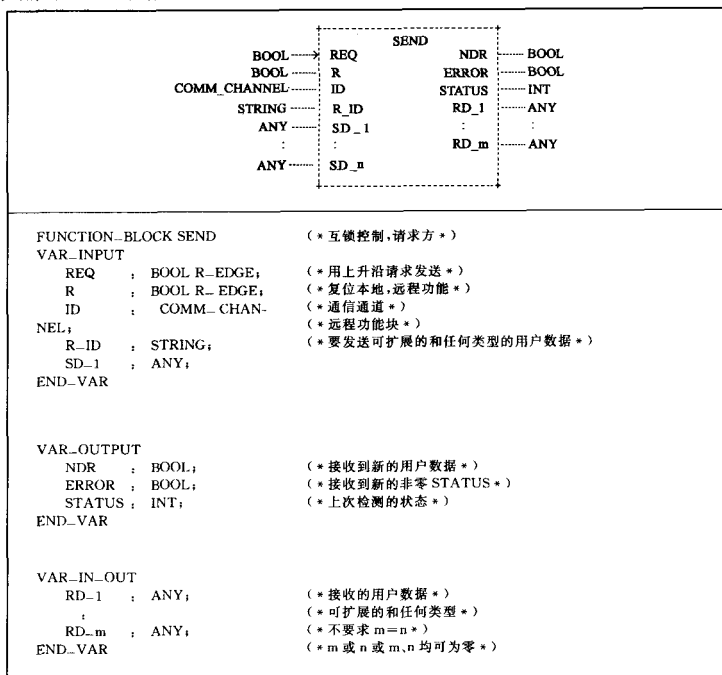


图 34 SEND 功能块

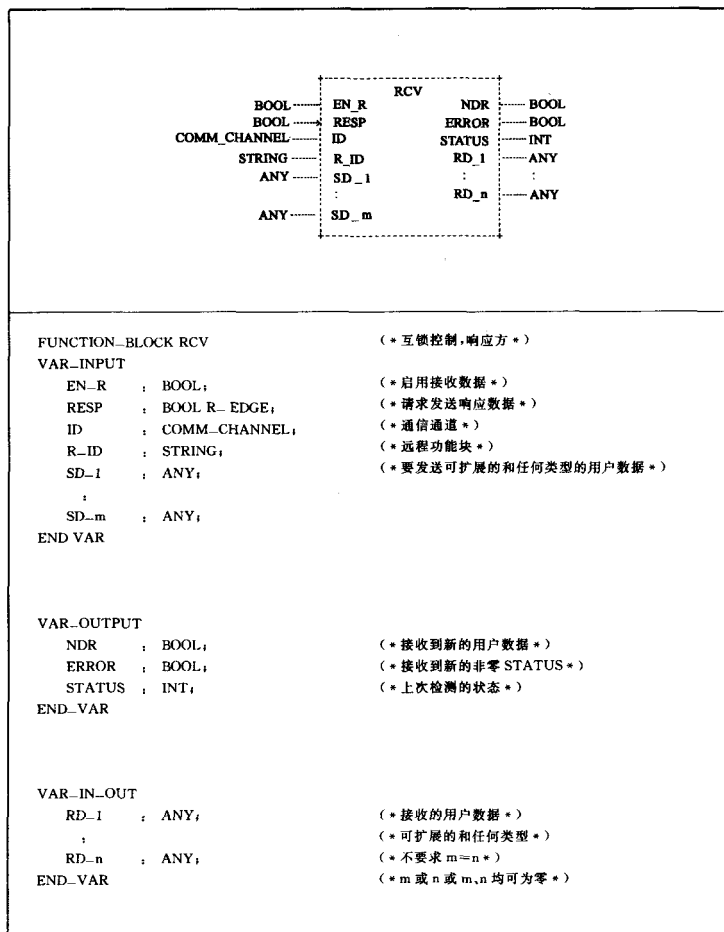


图 35 RCV 功能块

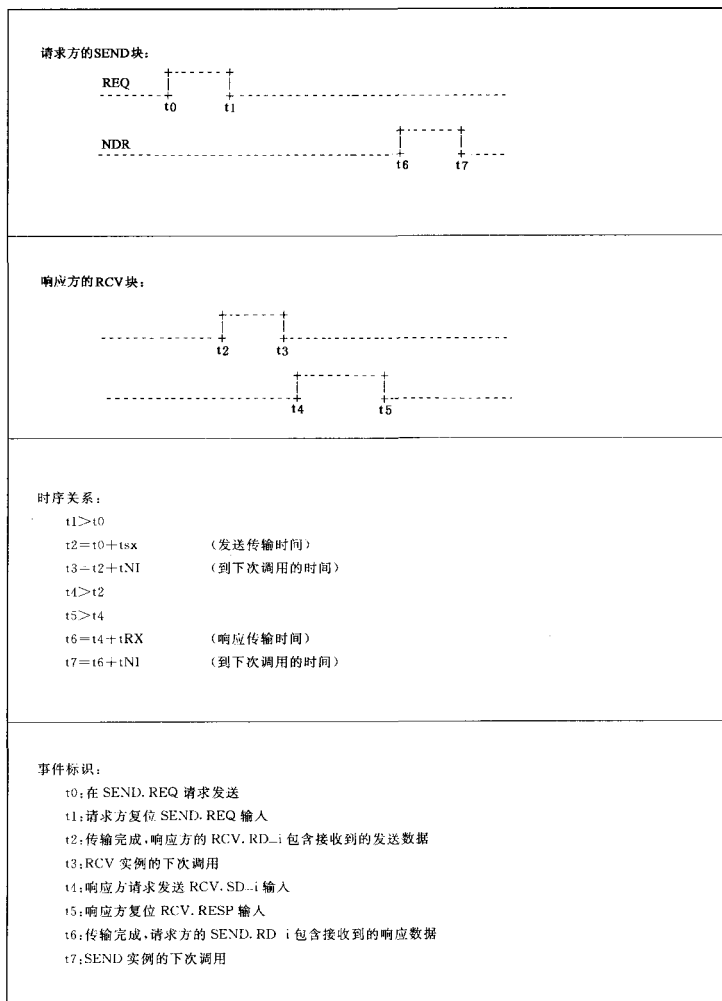


图 36 SEND 和 RCV 功能块的时序图

图 37 所示的状态图描述 SEND 功能块的算法。表 41 和表 42 描述状态图的转换和在各个状态内执行的动作以及 SEND 功能块输出的设置。

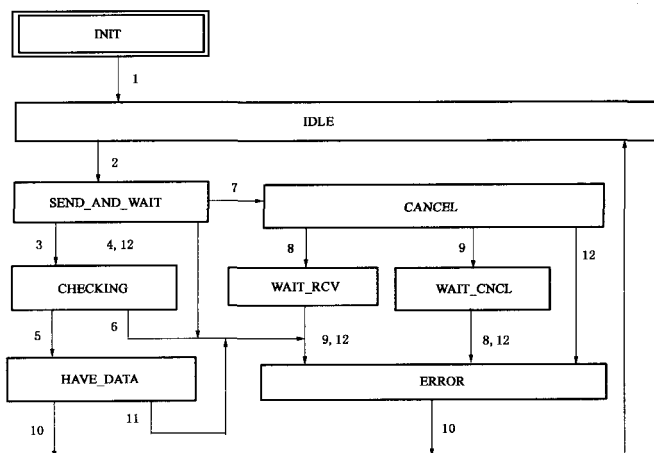


图 37 SEND 功能块的状态图

表 41 SEND 状态图的转换

转换	条 件
1	完成初始化
2	在 REQ 输入的上升沿
3	RCV 的肯定响应
4	RCV 的否定响应
5	接收数据的数据类型与和 RD ₁ 到 RD _m 相匹配
6	接收数据的数据类型与和 RD ₁ 到 RD _m 不匹配
7	在 R 输入的上升沿
8	对复位请求的肯定或否定响应
9	RCV 的肯定或否定响应
10	实例的下次调用之后
11	检测出本地资源的问题
12	检测出通信问题

表 42 SEND 状态图的动作表

状态	动作	FB 输出			
		NDR ^{c)}	ERROR ^{c)}	STATUS	RD_1...RD_m
INIT ^{a)}	初始化输出	0	0	0	空系统值
IDLE	没有动作	0	0	—	—
SEND AND WAIT	发送数据到 RCV, 首先评估转换 7。	—	—	—1	—
CHECKING	检验数据类型的匹配	—	—	—	—
HAVE_DATA	存放数据在实例中	1	0	0	来自 RCV 的新数据
ERROR	指示出错	0	1	b)	—
CANCEL	请求复位 RCV	—	—	5	—
WAIT_RCV	没有动作	—	—	—	—
WAIT_CNCL	没有动作	—	—	—	—

—表示 FB 的输出“没有改变”。

a) INIT 是冷起动状态。

b) 出错代码放在状态输出中。

c) 见图 10。

图 38 所示的状态图描述 RCV 功能块的算法。表 43 和表 44 描述状态图的转换和在各个状态内执行的动作以及 RCV 功能块输出的设置。

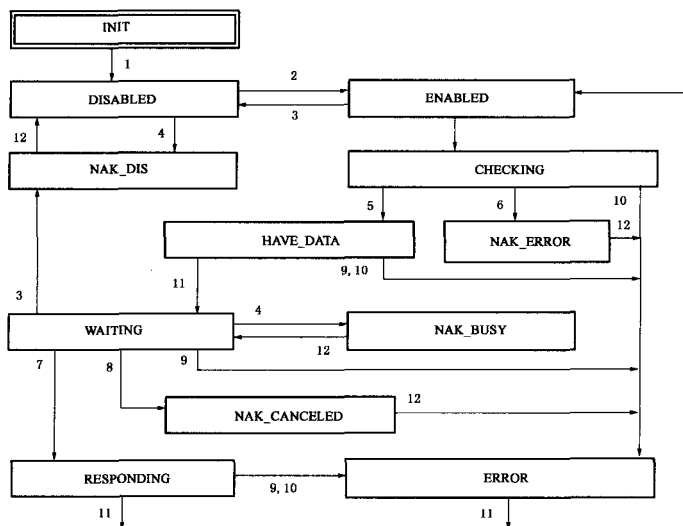


图 38 RCV 功能块的状态图

表 43 RCV 状态图的转换

转换	条 件
1	完成初始化
2	EN-R=1
3	EN-R=0
4	当从 SEND 接收到数据时
5	接收数据的数据类型与 RD-1 到 RD-n 相匹配
6	接收数据的数据类型与 RD-1 到 RD-n 不匹配
7	RESP 输入的上升沿
8	当由 SEND 请求复位时
9	检测出通信问题
10	检测出本地资源问题
11	下次调用实例之后
12	真,即条件为空

表 44 RCV 状态图的动作表

状态	动作	FB 输出			
		NDR ^{a)}	ERROR ^{a)}	STATUS	RD-1...RD-n
INIT ^{a)}	初始化输出	0	0	0	空系统值
DISABLED	没有动作	0	0	—	—
ENABLED	没有动作	0	0	—	—
CHECKING	检验数据类型的匹配	—	—	—	—
HAVE_DATA	存放数据	1	0	0	来自 SEND 的新数据
NAK_ERROR	发送否定响应到 SEND	—	—	—	—
WAITING	没有动作	0	0	—	—
NAK_BUSY	发送否定响应到 SEND	—	—	—	—
NAK_CANCELED	发送肯定响应到复位请求,然后发送否定响应到 SEND	—	—	—	—
RESPONDING	发送用户数据和肯定响应到 SEND	—	—	—	—
ERROR	指示出错	0	1	b)	—
NAK_DIS	发送否定响应到 SEND	—	—	—	—

表示 FB 的输出“没有改变”。

a) INIT 是冷起动状态。

b) 出错代码是放在状态输出中。

c) 见图 10。

7.8 编程报警报告

PLC 通信的编程报警功能使用 NOTIFY 功能块和 ALARM 功能块。(参阅图 39、图 40 和图 41) 一个 NOTIFY 功能块的实例或一个 ALARM 功能块的实例提供一个 PLC 编程报警功能的实例。

使用 ALARM 功能块可以编程 PLC 来报告一个具有确认能力的报警报文,或者使用 NOTIFY 功能块编程 PLC 来报告一个不具有应答能力的报警报文。

在 EVENT 输入的上升沿报告一个到来事件的报警报文,在 EVENT 输入的下降沿报告一个离去的事件。用 EV_ID 输入给出该事件的原因,事件的严重程度在 SEVERITY 的功能块输入中给出。严重性的范围从 0 到 127。0 代表最高的严重性,64 代表一般严重性,127 代表最低的严重性。另外,在 SD_i 输入处的数据可用于说明发生事件的细节。

ALARM 功能块的 ACK_UP 输出将包含一个指示,不管到来的事件是否通过远程通信的伙伴确认。ACK_DN 输出包含离去事件的指示。仅最新的事件确认应设置确认输出。对 EVENT 输入和 ACK_DN 输出的下降沿,同样的行为将是真。

ID 参数标识对于远程通信伙伴的通信通道。如果该通信系统提供支持“一对多”或“一对所有”连接的通信通道,则这些功能块可用来编程从一台 PLC 到多台 PLC 的报警报告功能。在这种情况下,第一个将有效的确认设置成适当的 ALARM 确认输出。

在报警报文中,发送数据能提供附加的信息。发送数据可能是空的,也就是说 SD_i 输入没有被编程。

如果发生错误,ERROR 输出产生一个周期脉冲指示出错,STATUS 输出包含此出错代码。

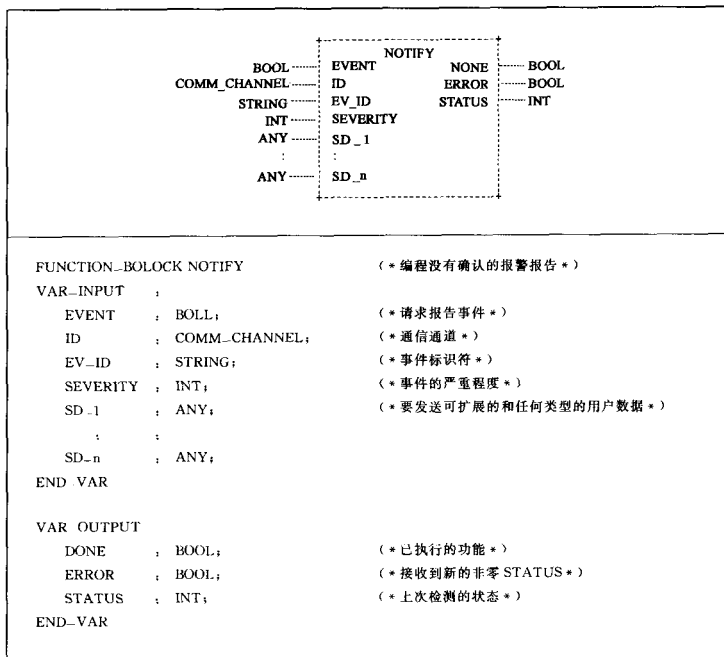


图 39 NOTIFY 功能块

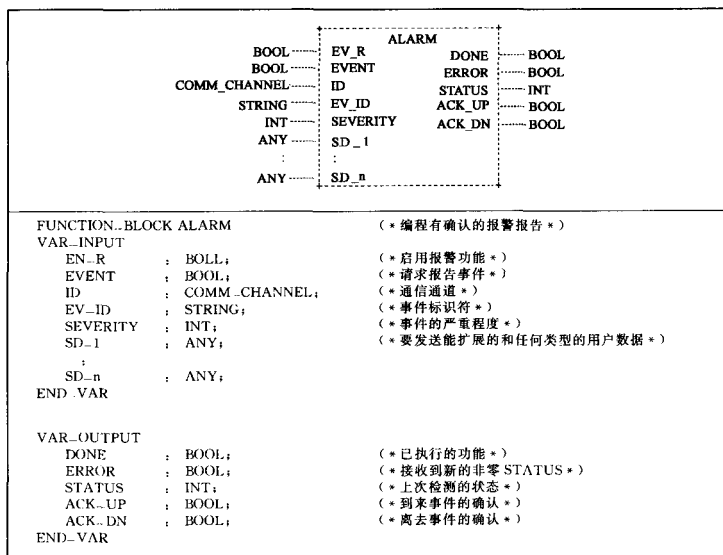


图 40 ALARM 功能块

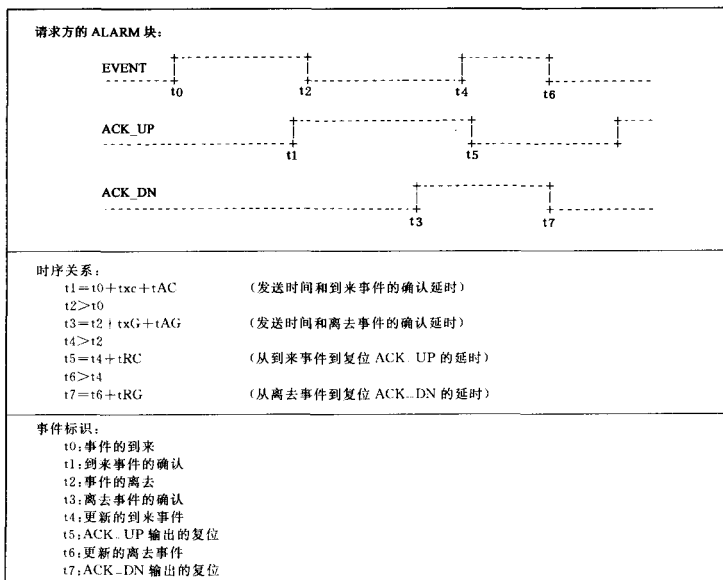


图 41 ALARM 功能块的时序图

图 42 所示的状态图描述 NOTIFY 功能块的算法。表 45 和表 46 描述状态图的转换和在各个状态内执行的动作以及 NOTIFY 功能块输出设置。

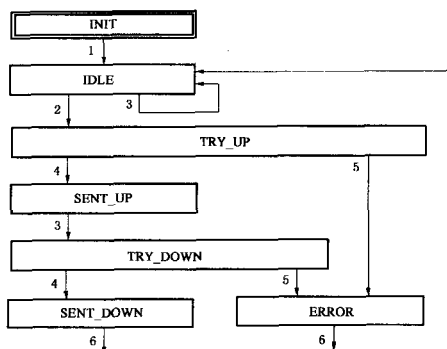


图 42 NOTIFY 功能块的状态图

表 45 NOTIFY 状态图的转换

转换	条 件
1	完成初始化
2	EVENT 输入的上升沿
3	EVENT 输入的下降沿
4	通信系统指示“发送到远程通信伙伴”
5	通信系统指示“不能发送到远程通信伙伴”
6	下次调用此实例之后

表 46 NOTIFY 状态图的动作表

状态	动作	FB 输出		
		DONE ^{a)}	ERROR ^{c)}	STATUS
INIT ^{a)}	初始化输出	0	0	0
IDLE	没有动作	0	0	0
TRY_UP	请求对远程通信伙伴报告到来的警报。	0	0	—
SENT_UP	没有动作	1	0	—
TRY_DOWN	请求对远程通信伙伴报告离去的警报。	0	0	—
SENT_DOWN	没有动作	1	0	—
ERROR	出错指示	0	1	b)

—表示 FB 的输出“没有改变”。

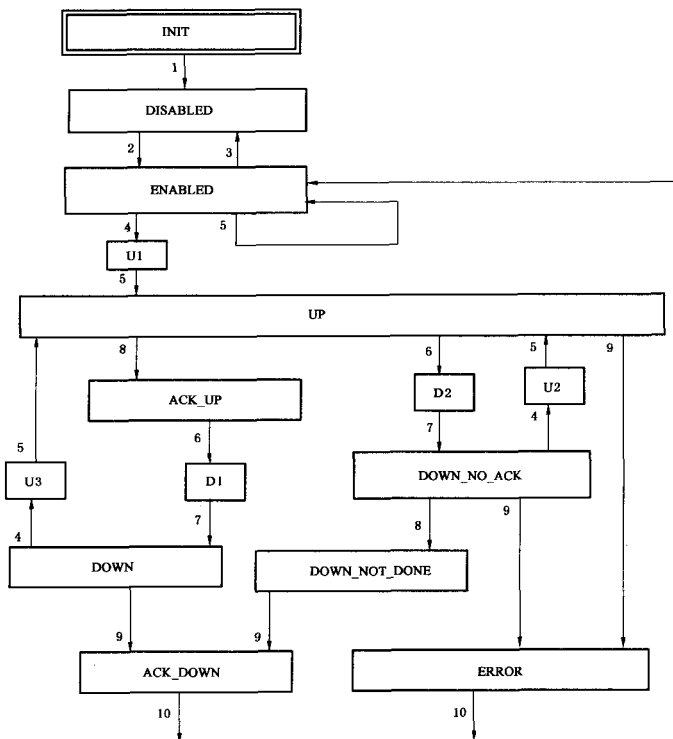
a) INIT 是冷起动状态。

b) 出错代码放在状态输出中。

c) 见图 10。

图 43 所示的状态图描述 ALARM 功能块的算法。表 47 和表 48 描述状态图的转换和在各个状态

内执行的动作以及 ALARM 功能块输出的设置。



注：如果底层在试发送事件通知时检测到有错误，则状态标记 U1、U2、U3、D1 和 D2 也可能转换到 ERROR 状态。

图 43 ALARM 功能块的状态图

表 47 ALARM 状态图的转换

转换	条 件
1	完成初始化
2	EN_R=1
3	EN_R=0
4	EVENT 输入的上升沿
5	由通信系统发送到来事件的通知
6	EVENT 输入的下降沿
7	由通信系统发送离去事件的通知
8	接受到来事件报警的确认
9	接受离去事件报警的确认
10	下次调用这个实例之后

表 48 ALARM 状态图的动作表

状态	动作	FB 输出				
		DONE ^{c)}	ERROR ^{c)}	STATUS	ACK UP	ACK DN
INIT ^{a)}	初始化输出	0	0	0	0	0
DISABLED	没有动作	0	0	—	0	0
ENABLED	没有动作	0	0	0	—	—
U1,U2,U3	请求对远程通信伙伴报告到来的报警	0	0	—	0	—
UP	没有动作	1	—	—	—	—
ACK UP	肯定地确认接收到的确认	—	0	0	1	—
D1,D2	请求对远程通信伙伴报告离去的报警	0	0	—	—	0
DOWN NO ACK	没有动作	1	—	—	—	—
DOWN	没有动作	1	—	—	—	—
DOWN_NOT_DONE	没有动作	0	—	—	—	—
ACK_DOWN	肯定地确认接收到的确认	—	0	0	—	1
ERROR	指出出错并且否定地确认接收到的确认	0	1	b)	0	0
—表示 FB 的输出“不改变”。 a) INIT 是冷起动状态。 b) 出错代码放在状态输出中。 c) 见图 10。						

7.9 连接管理

PLC 通信功能连接管理使用 CONNECT 功能块。(参阅图 44 和图 45)

一个 CONNECT 功能块的实例提供 PLC 连接管理功能的一个实例。该通信功能允许在调用的通信伙伴和远程通信伙伴之间建立起一个连接。

PLC 能请求一个远程通信伙伴在该 PLC 和远程通信伙伴之间建立起一个连接。远程通信伙伴是用自己的名称来标识(名称由远程通信伙伴实现者指定)。定义到一个远程通信伙伴的通信通道。

远程通信伙伴应判定是否建立了连接。

如果一个 CONNECT 功能块的调用建立起一个远程通信伙伴的通信通道,则 ID 输出应提供通信通道标识符,它能用作其他通信功能块的 ID 输入,这些其他通信功能块与这个远程通信伙伴进行通信。

作为一个客户机,远程通信伙伴自身也能使用此连接用于它的通信。远程通信伙伴通过调用一个 CONNECT 功能块,以引用已经建立起连接的通信伙伴,就能得到 ID 参数必需的值。

如果出现一个错误,ERROR 输出产生一个周期的脉冲来指出出错,并且在 STATUS 输出中含有此出错代码。

由本地 PLC 在通信通道的一方或者两方,也能建立起相应的通信通道。在这种情况下,如果在具有参数 EN C 固定为 1 的应用程序循环的开始时调用用于此通信通道的 CONNECT 功能块,则 PLC 应显示相同的行为特性。实现者将为此通信通道定义怎样获得 ID 参数的适当的值。

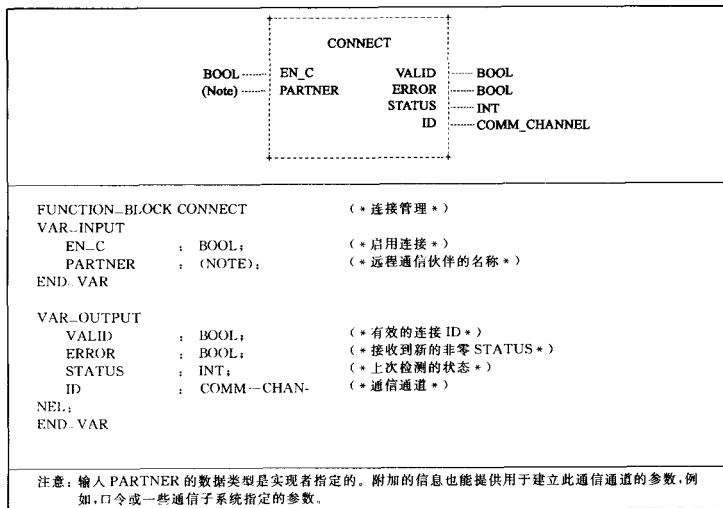


图 44 CONNECT 功能块

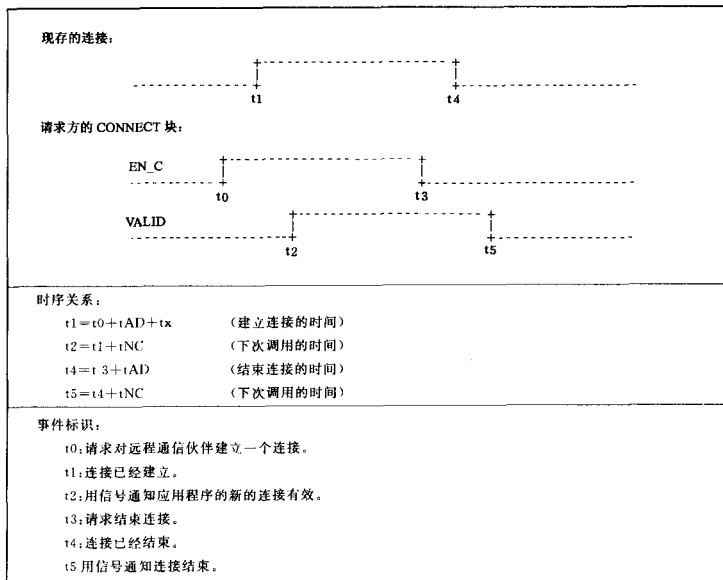


图 45 CONNECT 功能块的时序图

图 46 表示的状态图描述 CONNECT 功能块的算法。表 49 和表 50 描述状态图的转换和在各个状态内执行的动作以及 CONNECT 功能块输出的设置。

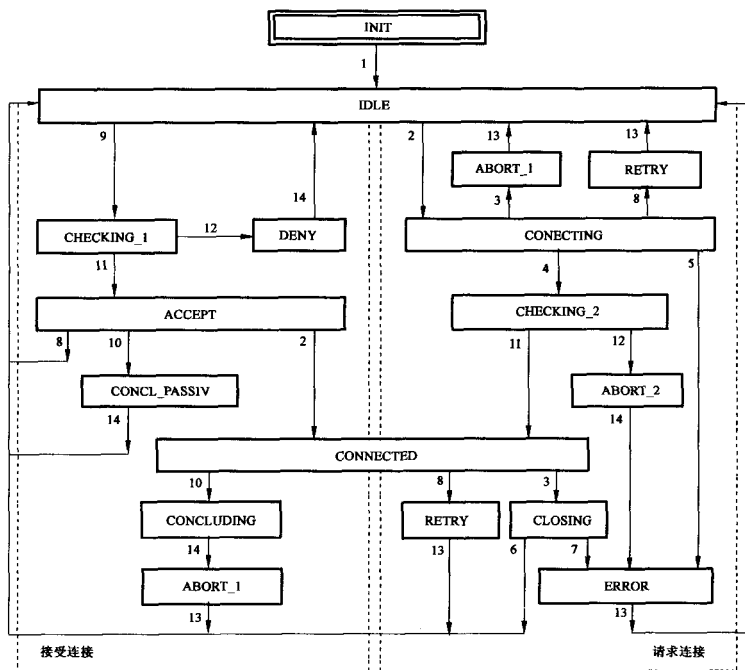


图 46 CONNECT 功能块的状态图

表 49 CONNECT 状态图的转换

转换	条 件
1	完成初始化
2	EN_C=1
3	EN_C=0
4	从远程通信伙伴接收到请求建立一个新连接的肯定确认。
5	从远程通信伙伴接收到请求建立一个新连接的否定确认。
6	从远程通信伙伴接收到请求结束连接的肯定确认。
7	从远程通信伙伴接收到请求结束连接的否定确认。
8	连接丢失
9	接收到对远程通信伙伴建立一个新连接的请求
10	接收到对远程通信伙伴结束连接的一个请求
11	接收建立一个连接
12	拒绝建立一个连接
13	下次调用
14	立即

表 50 CONNECT 状态图的动作表

状态	条件	FB 输出			
		VALID	ERROR ^{a)}	STATUS	ID ^{c)}
INIT ^{a)}	初始化输出	0	0	0	NG
IDLE	没有动作	0	—	—	NG
CHECKING_1	检查是否能建立连接	0	—	—	NG
DENY	对请求连接发送否定响应	0	—	—	NG
ACCEPT	接收建立连接并且发送肯定响应	0	0	—	NG
CONCL_PASSIV	对请求结束连接发送肯定响应	0			NG
CONNECTIONG	请求对远程通信伙伴的连接	0	0	—1	NG
ABORT_1	请求中止通信	0	—	—	NG
CHECKING_2	检查是否能建立连接	0	0	—	NG
ABORT_2	请求中止通信	0		—	NG
CONNECTED	没有动作	1	0	0	OK
RETRY	没有动作	0	1	1	NG
CLOSING	请求结束连接	0	0	—1	NG
CONCLUDING	对结束连接发送肯定响应	0	0		NG
ERROR	指示出错	0	1	b)	NG

—表示 FB 的输出“没有改变”。

a) INIT 是冷起动状态。

b) 出错代码放在状态输出中。

c) NG 表示“无效连接 ID”,OK 表示“有效连接 ID”。

d) 见图 10。

7.10 通信功能块的应用举例

7.10.1 建立一个通信通道

两台 PLC 要建立一个通信通道。两台 PLC 都要用此通道实现客户机和服务器功能,即两台 PLC 都需要为它们通信功能块的 ID 参数获得到适当的值。在此例中,假定数据类型 COMM_CHANNEL 用于通信通道的柄(handle)或索引(Index)。本例使用在 GB/T 15969.3 中定义的结构文本编程语言。

PLC1 应用程序的摘录:

(* 例如在一个应用程序的开始,说明数据和 CONNECT 实例 *)

VAR

TO_PLC2;COMM_CHANNEL;(* 由 CONNECT 设置的变量 *)

CO1;CONNECT;(* CONNECT 实例说明 *)

END_VAR;

(* 程序主体内的某个部分 *)

CO1(EN_C:=1,PARTNER:= 'PLC2');(* 调用 CONNECT 实例 CO1,并且建立一个到 PLC2 的通信通道 *)

IF CO1.ERROR THEN (* 推荐定义对一些错误的处理 *)

IF CO1.VALID THEN (* 存储引用的通信通道 *)

TO_PLC2:=CO1.ID;

PLC2 应用程序的摘录:

(* 例如在一个应用程序的开始,说明数据和 CONNECT 实例 *)

VAR

TO_PLC1:COMM_CHANNEL;(* 由 CONNECT 设置的变量 *)

CO2:CONNECT;(* CONNECT 实例说明 *)

END_VAR;

(* 程序主体内的某个部分 *)

CO2(EN_C:1,PARTNER:='PLC1');(* 调用 CONNCT 实例 CO2,并且建立一个到 PLC1 的通信通道 *)

IF CO2.ERROR THEN (* 推荐定义对一些错误的处 *)

IF CO2.VALID THEN (* 存储引用的通信通道 *)

TO PLC1:=CO2.ID;

PLC1 和 PLC2 应用程序的这些部分几乎是相同的。通信通道由 PLC 早先调用它的 CONNECT 实例所建立,以后调用的 PLC 得到早已建立的通信通道。

7.10.2 传送数据

两台 PLC 用一个已经建立的通信通道进行通信。使用 USEND 和 URCV 功能块从 PLC1 到 PLC2 传送一些数据。以下应用程序部分说明怎样才能完成这个功能。应用程序怎样提供和处理被传送的数据,在此例中没有提到。本例使用 GB/T 15969.3 中定义的结构文本编程语言给出。

使用一个 USEND 功能块实例发送数据的 PLC1 应用程序的摘录:

(* 例如在一个功能块定义中,说明数据和 USEND 实例 *)

VAR

SENDREQ:BOOL; (* 请求发送的标志 *)

TO_PLC2: COMM_CHANNEL; (* 允许使用到 PLC2 通信通道的变量 *)

SDAT1: ARRAY[0..20] OF BYTE; (* 要发送的数据说明 *)

SDAT2:REAL;

US1:USEND; (* USEND 实例说明 *)

END_VAR

(* 功能块主体内的某个部分 *)

US1(REQ:=SENDREQ,ID:=TO PLC2,R ID='PACK1,SD_1:=SDAT1,SD_2:=SDAT2);

(* 调用 USEND 实例 US1 且将在 SENDREQ 布尔值的上升沿发送数据 *)

IF US1.ERROR THEN(* 推荐定义对一些错误的处理 *);

使用一个 URCV 功能块实例来接受由 PLC1 发送数据的 PLC2 应用程序:

(* 说明数据和 URCV 实例,例如在一个功能块的定义中 *)

VAR

TO_PLC1:COMM_CHANNEL;(* 允许使用到 PLC1 通信通道的变量 *)

RDAT1:ARRAY[0..20] OF BYTE;(* 说明将被存储的数据的变量,变量的数量及其数据类型必

须与发出数据一致 *)

```
RDAT2:REAL;
UR1:URCV;          (* 说明 URCV 实例 *)
S:REAL;             (* 说明一个任意浮点的变量 *)
END_VAR;
```

(* 功能块主体内的某个部分 *)

```
UR1(EN_C:=1,ID:=TO_PLC1,R_ID='PACK1',RD_1=RDAT1,RD_2:=RDAT2);
(* 调用 URCV 实例 UR1 以等待来自 PLC1 的数据 *)
IF UR1.NDR THEN (* 处理接收到的数据 *)
BEGIN
S:=S+RDAT2; (* 例如将接收到的浮点数加入到变量中 *)
IF UR1.ERROR THEN (* 推荐定义对一些错误的处理 *)
```

7.10.3 使用计时器监控通信

一台 PLC(PLC1)要从 PLC2 请求一个处理功能。它使用一个 SEND 功能块的实例传送这个请求,它等待来自 PLC2 的响应。但是如果 PLC2 不能在 5 s 内响应,则复位这个请求。给出的例子用 GB/T 15969.3 中定义的结构文本编程语言。

使用一个 SEND 功能块实例来请求处理功能的 PLC1 应用程序摘录:

(* 说明数据和 SEND 实例,例如在一个功能块的定义中 *)

```
VAR
SREQ:BOOL;          (* 发送请求的标志 *)
FCT1:INT;            (* 要请求的功能码 *)
DAT1:REAL;           (* 这个功能的第一个参数 *)
RDAT1:INT;           (* 响应参数 *)
SR1:SEND;            (* FB SEND 实例 *)
M1:RS;               (* 保存计时器的 IN 参数 *)
T1:TON;              (* 用于 SEND 超时控制的计时器 *)
END_VAR;
```

(* 功能块主体内的某个部分 *)

```
SR1(REQ:=SREQ, (* 在 SREQ 布尔值的上升沿请求 *)
R:=T1.Q, (* 超时复位,例如计时器超时 *)
ID:=TO_PLC2,R_ID='ORD1', (* 识别远程伙伴和 RCV 实例 *)
SD_1:=FCT1,SD_2:=DAT1, (* 用于处理功能的数据 *)
RD_1:=RDAT1); (* 用于结果的变量 *)
M1(S:=SREQ, (* 当请求发送时设置 *)
R1:=T1.Q OR SR1.NDR OR SR1.ERROR);
(* 当计时器超时或 SEND 得到正确的(NDR=1)或错误的(ERROR=1)响应时复位 *)
T1(IN:=M1.Q1, (* 在请求期间保持计时器 5 秒钟动作 *)
PT:=T#5s);
```

使用 GB/T 15969.3 中定义的功能块图的相同的程序部分如图 47 所示。

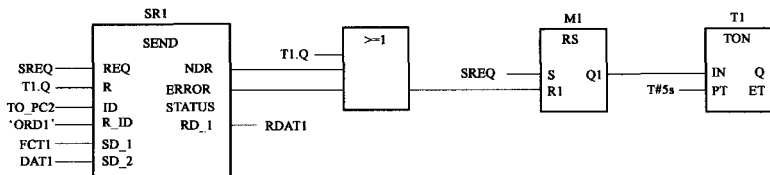


图 47 功能块图语言的举例

8 一致性和实现者特指的特性和参数

8.1 一致性

根据 GB/T 15969.1 中的定义,一个声称完整地或部分地符合本标准要求的 PLC 系统,其行为特征应符合如下描述。

一致性声明应包括在与系统配套的文件中或者由系统自身生成。一致性声明的格式是:

“本系统符合下列特性部分的要求”,接着为以下列格式的一致性表。

表 头

表编号	特性编号	特性描述

表和特性编号及其特性描述取自 GB/T 15969 有关的章节给出的表。表头和相关的表是取自表 51。

表 51 相符的表头和相关的表

表头	用于表中特性的相关表
PLC 状态	表 1~表 9
应用特指的特性	表 10~表 20, 除外表 15
PLC 通信功能块	表 21

一个 PLC 系统符合本标准必须的要求是:

- 为了完成本标准中规定的特性,不需要包括替代或其他特性。
- 应配套详细说明所有实现者特指的特性或参数值的文本,其内容列在表 52 和附录 A 的表 A30 中。
- 应配套单独描述任何在本标准中被禁止的或未被规定的通信关系特性的文本,如描述“在本标准中定义的 PLC 通信的扩展”的特性。
- 不使用在第 7 章中为实现者定义特性所定义的任何功能或功能块名称,因为实现者定义的特性与在这一条文中所描述的不同。

8.2 实现特指的特性和参数

在本标准中定义的实现特指的特性和参数以及与他们最相关的子条文都列在表 52 中。

表 52 实现特指的特性和参数

条目	实现特指的特性和参数
6.1	PLC 子系统的主要和次要故障的定义
6.1.8	PLC 提供状态信息的子系统数量
6.1.8	PLC 提供状态信息的子系统类型
6.1.8	包含 PLC 和它的子系统名称的字符串大小的最大值
6.1.8	对实现者特指的子系统,在子元素 STATE 中的值的语义
6.1.8	状态信息子元素 SPECIFIC 的大小
6.1.8	状态信息子元素 SPECIFIC 的语义
6.2.2	存取有直接表达式的变量的限制
6.2.2	存取有直接表达式的变量的算法
6.2.2	PLC 支持的能不间断地存取的每种数据类型的条件(大小、位置等)
6.2.6	支持的 I/O 状态的值
6.2.6	如果需要实现者特指的值则定义输出
6.2.6	如果需要用户指定的值则指定输出机制的定义。
6.2.7	其他语言元素,例如可装载的功能类型或功能块类型和下载与上载它们的条件及限制。
6.2.7	当一个客户正在下载到 PLC 时,另外的客户用这个 PLC 能做些什么
7.2	通信功能块使用 COMM-CHANNEL 类型的 ID 输入识别远程通信伙伴
7.2	REMOTE_VAR 功能的 VAR-ADDR 输出的数据类型和对使用 REMOTE_VAR 功能输出的限制
7.2	在 REMOTE_VAR 功能的 SCOPE 参数中使用执行者支持的名称范围
7.2	映象实现者特指的名称范围到通信系统
7.2	由一个通信功能块调用支持的 SD-i, RD-i, 和 VAR-i 参数的数量
7.2	通信功能块的 STATUS 输出的代码比-1 小或者比 20 大的含义
7.2	如果用 CONNECT 功能块不明显地受控,则用另外的(如果有的话)控制时间的方法建立通信通道
7.3	FB STATUS 和 FB USTATUS 的 LOCAL 输出的附加状态信息所使用的长度和语义
7.9	使用在通信功能块的 ID 输入,用户怎样能得到通信通道的信息

附录 A

(标准的附录)

映象到 ISO/IEC 9506-5

A1 概述

此附录规定 PLC 通信功能对 GB/T 16979.1 中定义的 MMS 对象和服务以及 ISO/IEC 9506-5 中定义的扩展的映象。当 PLC 使用 ISO/IEC 9506-5 中定义的抽象语法进行通信时,要使用此映象。

引用标准

- GB/T 9387.1—1998 信息技术 开放系统互连 基本参考模型:基本模型
GB/T 16979.1—1996 工业自动化系统 制造报文规范 第1部分:服务定义 修订1:数据交换。
GB/T 16720.2—1996 工业自动化系统 制造报文规范 第2部分:协议规范 修订1:数据交换。
ISO/IEC 9506-5:1999 工业自动化系统 制造报文规范 第5部分:用于可编程序控制器的伴同标准

引用其他标准的定义

- GB/T 9387.1—1998 信息技术 开放系统互连 基本参考模型:基本模型
应用实体(application entity)(AE)
GB/T 16979.1—1996,工业自动化系统 制造报文规范 第1部分:服务定义
域(domain)
程序调用(program invocation)(PI)
虚拟制造设备(Virtual Manufacturing Device)(VMD)(条文7)

缩写词

- | | |
|-----|-----------|
| Cnf | 通信服务的确认原语 |
| Ind | 通信服务的指示原语 |
| Req | 通信服务的请求原语 |
| Rsp | 通信服务的响应原语 |
| VMD | 虚拟制造设备 |

A2 应用特指的功能

A2.1 设备检验

PLC 可以通过 MMS 的 Status 服务提供一些非常通用的状态给所请求的设备。此状态的内容是由 MMS 定义的。实现者可以提供包含附加信息的 Local Detail(最大 128 个位的位串)。PLC 可以使用 MMS Unsolicited Status 服务启动此相同信息的未经请求的状态报告。

注:Local Detail 的 16 个最低编号的位可能提供如同 P-PLC STATE 变量中相同的信息。

PLC 可以通过 MMS Named Variables 或 MMS Unnamed Variables 或两者提供 PLC 及其子系统的详细状态信息(见 6.1.8)。这些变量可以用 MMS Read 服务来读出。对于这些变量的规定见 6.1.8。

A2.2 数据采集

包含在 PLC 中的数据用 PLC 变量表达。使用存取路径语言结构将所选择的 PLC 变量映象到 MMS Named Variables 上(见 A3.2)。此外,直接表达式的变量可以由提实现者特指的约定映象到 MMS Unnamed Variables。使用 MMS Read 和 InformationReport 服务,可以将这些变量的内容提供给客户机。

A2.3 参数控制

由对 PLC 变量写入的值来控制 PLC 的操作,即参数控制。操作的变化由应用程序或其他本地机制来决定。MMS Write 服务用于对 PLC 变量写入一个新值。

A2.4 互锁控制

客户机请求服务器执行一个应用操作并将操作结果通知客户机,即互锁控制。SEND 和 RCV 功能块用于提供这种功能。这些功能映象成 MMS 对象和服务,见 A4。

A2.5 用户应用间的同步

用户可能需要同步服务。此 PLC 功能由通信功能互锁控制(见 6.2.3 和 A2.4)提供。

A2.6 报警报告

当一个预定的条件满足时,PLC 能够向客户机发出报警报文。ALARM 和 NOTIFY 功能块用于提供此功能。这些功能映象成 MMS 对象和服务,见 A4。

A2.7 应用程序执行和 I/O 控制

程序执行和 I/O 控制使用 GB/T 15969.3(见 6.2)的语言元素配置和资源。配置和资源被映象成 MMS Program Invocation 对象。

与资源有关的 I/O(输入和输出)的状态用各种 MMS 服务指定,这些 MMS 服务使用 ISO/IEC 9506-5 中定义的 I/O State Parameter 并在 Program Invocation 上运行。

A2.8 应用程序传送

应用程序传送允许客户机为了归档或检验而上载可编程存储器的全部或部分内容或者为了恢复 PLC 成已知的状态而下载它。可以被上载或下载的可编程内存的部分按 6.2.7 中的规定。这些可装载的单元被映象到 MMS Domain 对象。它们用 MMS Domain 服务传送。

A2.9 连接管理

MMS Environment 和 General Management 服务用于管理连接。

A3 PLC 对象映象

ISO/IEC 9506 对通信支持一种面向对象的观念。用 ISO/IEC 9506-5 的抽象语法通信的 PLC 被映象成一个虚拟设备。此虚拟设备包括对远程通信伙伴可视的所有对象和从远程伙伴请求的所有服务。

A3.1 VMD

一个 PLC、PLC 的某一部分或若干 PLC 的集合均可以映象为一个 VMD。对 VMD 的通信应符合 ISO/IEC 9506-5。

实现者应规定此映象。本标准对 VMD 的属性未规定任何限制。

A3.2 命名变量

每个存取路径被映象为一个 MMS 的 Named Variable。MMS 的 Named Variable 的属性如下:

对象:Named Variable

——属性:Variable Name——变量名,它是存取路径说明中的存取名称。如果存取路径定义引用直接表达式的变量、程序输入或输出、配置或资源的全局变量(见 GB/T 15969.3 的 2.7.1),则此名称有 VMD-Specific 范围。如果存取路径定义是在程序级(见 GB/T 15969.3 的 2.5.3),则此名称应是该域的域专用名称,该域与包含此引用变量的程序有关见 A2.9)。

——属性:Reference to Access Control List——引用存取控制表,如果存取路径中包含 READ-ONLY 语言描述,则此对象将引用 Access Control List 对象 M_ReadOnly,要不然引用 M_NonDeletable。

——属性:Type Description——类型描述,它是在根据表 A1 映象到 MMS 的存取路径说明中规定的的数据类型。

表 A1 类型描述映象

编号	存取路径数据类型关键字	MMS 类型描述类别和大小	注
1	BOOL	布尔数	4
2	SINT	8 位整数	4
3	INT	16 位整数	4
4	DINT	32 位整数	4
5	LINT	64 位整数	4
6	USINT	8 位无符号数	4
7	UINT	16 位无符号数	4
8	UDINT	32 位无符号数	4
9	ULINT	64 位无符号数	4
10	REAL	32 位浮点数(小数点在第 8 位后)	4
11	LREAL	64 位浮点数(小数点在第 11 位后)	4
12	TIME	32 位无符号数	4
13	DATE	二进制—时间真	1
14	TIME_OF_DATE.TOD	二进制—时间假	1
15	DATE_AND_TIME.DT	二进制—时间真	1
16	STRING[N]	8 位位组串 N	2
17	BYTE	8 位位串	4
18	WORD	16 位位串	4
19	DWORD	32 位位串	4
20	LWORD	64 位位串	4
21	Enumerated specification	整数	3
22	Subrange specification	子区域的基本数据类型	
23	ARRAY	数组	
24	STRUC	结构	
注 1: 值的真或假分别指示数据被包含或不被包含 注 2: 实现者应规定允许的最大长度, 给出的 N 即此串的最大的尺寸 注 3: 为了保持所有可能的计算值, 选择整数的大小 注 4: 给出的大小是固定的			

用 REMOTE VAR 功能可以寻址变量。SCOPE 和 SC_ID 参数的映象如表 A2:

表 A2 SCOPE 和 SC_ID 参数的映象

参数 SCOPE 的值	MMS 命名范围	参数 SC_ID 的使用
0	VMD	不使用
1	VMD	不使用
2	域	域名
3	域	域名
10	VMD	不使用
11	域	域名
12	与应用有关	AA 名

A3.3 非命名变量

对象:Unnamed Variable

——属性:Address——地址,这种地址参数具有值“Symbolic Address”. Symbolic Address 参数的值是 PLC 变量的直接表达式。

——属性:Type Description——类型描述,它是根据表 A3 映射的直接表达式的大小前缀而得出的数据类型。

表 A3 直接表达式的大小前缀

编号	直接表达式的大小前缀	MMS 类型描述和大小
1	(无)	布尔数
2	X	布尔数
3	B	8 位位串
4	W	16 位位串
5	D	32 位位串
6	L	64 位位串
注: GB/T 15969.3 的编程语言定义符号变量的表示方法,而另一种方法是直接表示数据元素与 PLC 的输入、输出或内存结构中物理的或逻辑的位置的关系。非名变量对象用于存取有直接表达式的变量。		

A3.4 程序调用

对象:Program Invocation

——属性:Program Invocation Name——程序调用名,它是一个与配置或资源有关的名称。

——属性:Reusable——可重用,该属性应为 TRUE。

——属性:Additional Detail——附加细则,它由 ISO/IEC 9506-5 中定义的如下属性组成:

1) Independent——如果相关的 PLC 语言元素是一个配置,则此属性为真值否则它具有假值。

2) Constraint:Independent=FALSE

Program Invocation Reference——程序调用引用,此属性具有配置标识符的值。

3) I/O 状态

实现者为建立附加的程序调用对象可以定义附加含义。

A3.5 域

对象:Domains

——属性:Domain Name——域名,它是相关的可装载单元的名称。

——属性:List of Subordinate Objects——附属对象表,如果命名变量 P_DDATE 由实现者支持,则出现此属性。

实现者为建立附加的域对象可以定义附加方法。

A4 映射到 MMS 对象和服务的通信功能块

通信功能块使用有转换和动作的状态图来描述。在附录的本部分中,涉及到 PLC 通信系统的所有转换和动作被映射到 ISO/IEC 9506-5。

A4.1 使用通信通道

ISO/IEC 9506-5 仅提供一对一的通信通道。

在下面几个条文中叙述的所有请求和所有响应均使用由通信功能块的 ID 参数引用的应用连接来发送。指示或确认仅对某些通信功能块实例有作用,在这些实例中使用由其 ID 参数引用的应用连接来接收。

A4.2 数据类型兼容规则

当接收到一个协议数据单元并请求检验所接收的对象的数据类型是否与正在接收的功能块的 RD_i 输出参数处编程的应用程序变量的数据类型相一致,如下的规则是适用的。

a) 如果在 RD_i 参数处编程的应用程序变量是有符号或无符号的整数数据类型,且对接收到的任何有符号或无符号的整数类型的变量在存贮它的值时没有丢失信息,则通过类型检查。此应用程序变量将得到所接收的变量对象的值。

b) 如果在 RD_i 参数处程序的应用程序变量是 BYTE、WORD、DWORD 或 LWORD 数据类型,且对接收到的任何位串类型变量,其长度不大于应用程序变量的位长度,则通过类型检查。此应用程序变量将得到所接收变量对象的值,并存放所接收的位串的位 0 到应用程序变量的位 0 中,如此等等。如果它的长度超过了,则应用程序变量的值填写为 0。

c) 如果在 RD_i 参数处编程的应用程序变量是 STRING 数据类型,且对接收到的任何变量其字节长度不大于应用程序变量的字节长度,则通过类型检查。此应用程序变量 STRING 将得到所接收变量对象的值,并存放所接收变量的字节 1 到应用程序变量 STRING 的字节 1 中,如此等等。此字符串的长度将设置为所接收到的并存放的字节数。

d) 如果所接收的变量对象具有八位位组串的数据类型且应用程序变量的字节长度不小于所接收的八位位组串的长度,则通过类型检查。应用程序变量将得到所接收的八位位组串的值,并存放所接收的变量的字节 1 到应用程序变量的字节 1 中,如此等等。如果所接收到的变量的长度超过了,则应用程序变量的值填为 0。

e) 如果在 RD_i 参数处编程的应用程序变量不是在规则 a) 到 c) 中描述的数据类型而且也不适用于规则 d),此时如果所接收的 MMS 变量对象的数据类型符合使用表 A1 和表 A3 的应用程序变量的数据类型,则通过类型检查。此应用程序变量将得到所接收的变量对象的值。

A4.3 设备检验

STATUS 功能块使用作为客户机的 Status 服务。(参阅表 A4)

表 A4 STATUS 状态图的转换映象

转 换	条 件
3	远程通信伙伴的肯定响应: Status.cnf(+)
4	远程通信伙伴的否定响应或发现通信问题: Status.cnf(-)或低层指示出错

表 A5 STATUS 状态图的动作映象

状 态	动 作
WAITING	请求远程设备的状态信息: Status Req: Extended Derivation: FALSE
HAVE_DATA	存放状态信息在实例中: Status.cnf(+): VMD 逻辑状态到 LOG 输出 VMD 物理状态到 PHYS 输出 本地细节到 LOCAL 输出

USTATUS 功能块使用作为接收方的 Unsolicited Status 服务。(参阅表 A6 和表 A7)

表 A6 USTATUS 状态图的转换映象

转 换	条 件
4	接收到远程通信伙伴的状态信息: Unsolicited Status. Ind
5	发现通信问题: 低层指示出错

表 A7 USTATUS 状态图的动作映象

状 态	动 作
HAVE_DATA	存放状态信息在实例中： Unsolicited Status. Ind VMD 逻辑状态到 LOG 输出 VMD 物理状态到 PHYS 输出 本地细节到 LOCAL 输出

A4.4 轮询数据采集

READ 功能块使用作为客户机的 Read 服务。(参阅表 A8 和表 A9)

在 VAR_i 输入中的要作为变量标识符读出的 REMOTE_VAR 功能的字符串和输出被映象为 Read 请求的 Variable Access Specification 参数。VAR_i 输入的每个标识符形成 List of Variable Selection 的一个元素。

如果 VAR_i 输出的字符串以“%”字符开始,则应读取一个带 Symbolic Address 的 Unnamed Variable 对象,此 VAR_i 输入的字符串被用在 Symbolic Address 参数中。否则,将读出一个 VMD-Specific 变量。Object Name 参数中的 Name Scope 参数应设定为 VMD-specific,此字符串用作标识符。

如果 REMOTE-VAR 功能输出被用来标识要读的远程变量,则 Object Name 参数的 Name Scope 参数应该设定为 REMOTE-VAR 功能的 SCOPE 输入中的给出值,并用表 A2 来映象。如果范围 (Scope) 是 VMD-Specific 或 AA-Specific,则 NAME 输入的值用作标识符。如果范围是域专用,则 SC_ID 输入的值用作域标识符,且此 NAME 输入值作为项 (item) 标识符。

如果所获得的变量要被读出,则 Variable Access Specification 的 Alternate Access 选项用来说明要读出一个结构的哪个子元素或一个数组的哪个元素要被读出。List of Alternate Access Selections 应精确地包含一个元素。如果 REMOTE_VAR 功能的 SUB 输入是一个整数(有符号或无符号的),则它应用作要读出的数组元素的索引。如果 SUB 输入是一个非空的字符串,则它应用作要读出的子元素的部件名。

表 A8 READ 状态图的转换映象

转 换	条 件
3	远程通信伙伴的肯定响应： Read.cnf(+)
4	远程通信伙伴的否定响应或发现有其他通信问题 Read.cnf(-)或低层指示出错

表 A9 READ 状态图的动作映象

状 态	动 作
WAITING	从远程设备请求变量： Read.Req Specification With Result=FALSE Variable Access Specification
CHECKING	检验数据类型的匹配： 检查 List of Access Results 的所有元素是否与 RD_i 输出说明一致： 如果 Success 是假,则检查失败。 如果 Kind of Data 参数与相关的 RD_i 输出的数据类型不匹配(见 A4.2),则检查失败。 如果检查出一个数据元素失败,则整个检查失败。
HAVE_DATA	存放数据： 存放 List of Access Results 的所有元素在 RD_i 输出中。

A4.5 编程数据采集

USEND 功能块使用作为请求方的 Information Report 服务。(参阅表 A10 和表 A11)

一个 USEND 功能块的实例用 AA-specific 范围定义一个 Named Variable 对象。此对象的属性如下:

对象:Named Variable

——属性:Variable Name——变量名,在 R-ID 输入参数中的字符串是带有 AA-Specific 范围的变量名。

——属性:Reference to Access Control List——引用存取控制表,此对象应引用 Access Control List 对象 M-ReadOnly。

——属性:Type Description——类型描述,如果仅给出一个发送数据参数,则 Named Variable 对象的类型就是使用表 A1 和表 A3 的发送数据的类型;如果编程的发送数据参数多于一个,则 Named Variable 对象的类型是一个包含以发送数据作为分量的结构,这些分量的次序和类型与发送数据参数给出的相同。

表 A10 USEND 状态图的转换映象

转 换	条 件
3	通信系统指示“Sent to Remote Communication Partner”; 低层指示无出错
4	通信系统指示“Cannot Sent to Remoter Communication Partner”或发现其他通信问题; 低层指示出错

表 A11 USEND 状态图的动作映象

状 态	动 作
TRYING	发送给 SD-i 输入上的数据到远程通信伙伴: Information Report. Req Variable Access Specification Kind of Variable=NAMED Name Named Scope=AA-SPECIFIC Item Identifier=Content of R-ID input List of Access Results

URCV 功能块使用作为接收方的 InformationReport 服务。(参阅表 A12 和表 A13)

R-ID 中的字符串用作将被 URCV 功能块接收的带有 AA-Specific 范围的 Named Variable 对象的名。如果仅给一个数据输出 RD 1,则此参数的类型将用所接收的变量的类型来检查,见 A4.2。如果给出多个数据输出 RD_1...RD_n,则所接收的变量对象的类型应是一个包含用于不同参数作为分量的数据的结构,这些分量的次序和类型与在功能块实例中给出的相同。

表 A12 URCV 状态图的转换映象

转 换	条 件
4	接收到来自远程通信伙伴的数据: InformationReport.Ind Variable Access Specification List of Variable Variable Specification Name Name Scope = AA-SPECIFIC Item Identifier = Value of R-ID Parameter List of Access Results
5	发现通信伙伴的问题 低层指示出错
6	USEND 的 SD_i 类型与 URCV 的 RD_i 的数据类型匹配; Access Result 参数的数据参数与 URCV 实例的 RD_i 输出的读数和数据类型匹配(见 A4.2)
7	USEND 的 SD_i 的数据类型与 URCV 的 RD_i 的数据类型不匹配; Access Result 参数的数据参数与 URCV 实例的 RD_i 输出的读数和数据类型不匹配(见 A4.2)

表 A13 URCV 状态图的动作映象

状 态	动 作
CHECKING	检验数据类型匹配: 检查 List of Access Results 的所有元素是否与 RD_i 输出说明一致; 如果 Success 是假,则检查失败。 如果 Kind of Data 参数与相关的 RD_i 输出的数据类型不匹配 (见 A4.2),则检查失败。 如果有一个数据元素的检查失败,则整个检查失败。
HAVE_DATA	存放数据: 存放 List of Access Result 的所有数据元素在 RD_i 输出中

BSEND 功能块使用作为客户机的 Write 服务。(参阅表 A14 和表 A15)

被写的带 AA-Specific 范围的 Named Variable 对象由对应的 BRCV 功能块的实例来定义。此对象的属性如下:

对象:Named Variable

——属性:Variable Name——变量名,在 R-ID 输入参数中字符串是带有 AA-Specific 范围的变量名。

——属性:Reference to Access Control List——引用存取控制表,此对象应引用 Access Control-List 对象 M_ReadOnly。

——属性:Type Description——类型描述,它是一个结构,包括无符号 32 位整数数据类型的索引、数据类型布尔的 more_follows 标志并作为八位位组串发送的数据。

```

Structure{
  Components{
    {Component Name="Index",
      Component Type=unsigned_32},
  }
}

```

```
{Component Name="More_Follows",
Component Type=boolean},
{Component Name="Data",
Component Type=Octet_String}}}
```

表 A14 BSEND 状态图的转换映象

转 换	条 件
3	接收到肯定确认,并发送更多的数据: Write.cnf(+)
4	接收到否定确认或检查出通信问题: Write.cnf(-)或低层指示出错
6	接收到肯定确认,并无更多的数据发送: Write.cnf(+)
9	检查出通信问题: 低层指示出错

表 A15 BSEND 状态图的动作映象

状 态	动 作
SEND_FIRST	发送 SD.1 输入中的第 1 个数据块给远程通信伙伴,发送全部的最大 LEN 字节: Write.Req Variable Access Specification List of Data With Component Index=1 (=element number of the first byte in the byte array to send) Component More_Follows=true if not all data to send are contained in component Data Component Data=the first bytes out of the data to Send
SEND_NEXT	发送 SD.1 输入中的下一个数据块给远程通信伙伴,发送全部的最大 LEN 字节 Write.Req Variable Access Specification List of Data With Component Index=Element number of the first byte of the octet string in the byte array to send if Component More_Follows=true not all data still to send are contained in component Data Component Data=bytes out of the data to send starting with the byte the number of which is contained in component Index
CANCEL	停止数据传送: Write.Req Variable Access Specification List of Data With component Index=0 With component More_Follows=false With component Data=null

BRCV 功能块使用作为服务器的 Write 服务。(参阅表 A16 和表 A17)

R_ID 输入中的字符串是由 BRCV 功能块定义的带 AA-Specific 范围的 Named Variable 对象的名。

表 A16 BRCV 状态图的转换映象

转 换	条 件
4	接收到来自远程通信伙伴的数据: Write_Ind
5	跟随更多的数据是真: Component More_Follows=true
6	跟随更多的数据是假: Component More_Follows=false
7	接收到取消数据传送的指示: Write_Ind With Component Index=0
9	检查出通信问题: 低层指示出错

表 A17 BRCV 状态图的动作映象

状 态	动 作
RECEIVING	检验数据类型匹配: 如果刚接收到的分量数据的八位位组串的长度+包含在刚收到的分量索引减去 1 中的值大于 RD-1 字节数组的元素个数时,则检查失败
RESP_MORE	发送肯定响应: Write_Rsp(+)
RESP_LAST	发送肯定响应: Write_Rep(+)
HAVE_IT	存放数据: 从具有 Component Index 中号的元素开始,成功地存放 Component Data 的字节
CANCELLED	发送肯定响应: Write_Rep(+)

A4.6 参数控制

Write 功能块使用作为客户机的 Write 服务。(参阅表 A18 和表 A19)

在 VAR_i 输入中的要作为变量标识符写入的 REMOTE_VAR 功能的字符串和输出被映象为 Write 请求的 Variable Access Specification 参数。VAR_i 输入的每个标识符形成 List of Variable Selection 的一个元素。

如果 VAR_i 输入的字符串以“%”字符开始,则一个带 Symbolic Address 的 Unnamed Variable 对象将被写入。此 VAR_i 参数的 Name Scope 参数将设定的 VMD-Specific。此字符串用作标识符。

如果 REMOTE_VAR 功能输出被用来标识要被写的远程变量,则 Object Name 参数的 Name Scope 参数应该设定为 REMOTE_VAR 功能的 SCOPE 输入中的给出值,并用表 A2 来映象。如果此范围是 VMD-Specific 或 AA-Specific,则 NAME 输入的值用作标识符。如果此范围是域专用,则 SC_ID 输入的值应用作域的标识符,且此 NAME 输入值作为项(item)标识符。

如果所获得的变量要被写入,则 Variable Access Specification 的 Alternate Access 选项用来说明

一个结构的哪一个子元素或一个数组的哪一个元素要被写入。List of Alternate Access Selection 应精确地包含一个元素。如果 REMOTE_VAR 功能的 SUB 输入是一个整数(有符号的或无符号的),则它应该用作要写入的数组元素的索引。如果 SUB 输入是一个非空字符串,则应用作要写入的子元素的分量名。

SD_i 输入中的数据给出 Write 请求的 List of Data 参数。

表 A18 WRITE 状态图的转换映象

转 换	条 件
3	远程通信伙伴的肯定响应: Write.cnf(+)
4	来自远程通信伙伴的否定响应或发现其他通信问题: Write.cnf(-) 低层指示出错

表 A19 WRITE 状态图的动作映象

转 换	条 件
WAITING	请求写变量到远程通信伙伴: Write.Req Variable Access Specification List of Data

A4.7 互锁控制

SEND 功能块使用作为客户机的 Exchange Data 服务。(参阅表 A20 和表 A21)

R_ID 输入中的字符串用作 AA-Specific 数据交换对象的项标识符。Data Exchange 请求的 List of Request Data 参数中的数据输入 SD_1...SD_n 的次序和类型与 SEND 功能块调用中给出的相同。Data Exchange 确认的 List of Response Data 参数的数据依 SEND 功能块调用中给出的次序存放在数据输出 RD_1...RD_m 中。

表 A20 SEND 状态图的转换映象

转 换	条 件
3	来自 RCV 的肯定响应: Exchange Data.cnf(+)
4	来自 RCV 的否定响应: Exchange Data.cnf(-)
5	所接收到的数据的数据类型与 RD_1 到 RD_m 相匹配: List of Response Data 参数的数据与 RD_1 到 RD_m 相匹配
6	所接收到的数据的数据类型与 RD_1 到 RD_m 不匹配: List of Response Data 参数的数据与 RD_1 到 RD_m 不匹配
8	对复位请求的肯定或否定响应: Cancel.Cnf
9	来自 RCV 的肯定或否定响应: Exchange Data.Cnf
12	发现通信问题: 低层指示出错

表 A21 SEND 状态图的动作映象

状 态	动 作
SEND_AND_WAIT	发送数据到 RCV; ExchangeData. Req Data Exchange Name Name Scope = AA-SPECIFIC Item Identifier = Value of R-ID parameter List of Request Data
CHECKING	检验数据类型匹配; 检查 ExchangeData.cnf(+) 的 List of Response Data 参数的数据类型, 见 A4.2
CANCEL	请求复位 RCV; Cancel. Req Original Invoke ID

RCV 功能块使用作为服务器的 Exchange Data 服务。(参阅表 A22 和表 A23)

一个 RCV 功能块的实例定义带 AA-Specific 范围的 Data Exchange 对象。此对象的属性如下:

对象: Data Exchange

——属性: Data Exchange Name——数据交换名, 在 R-ID 输入参数中的字符串是带 AA-Specific 范围的 Data Exchange 对象的名。

——属性: Reference to Access Control List——引用存取控制表, 此对象将引用 Access Control List 对象 M_NonDeletable。

——属性: List of Request Type Specifications——请求类型说明表, 此表准确地包含 RD-i 输出中用表 A1 和表 A3 给出的接收数据的类型。

——属性: List of Response Type Specifications——响应类型说明表, 此表准确地包含 RD-i 输入中用表 A1 和表 A3 给出的发送数据的类型。

——属性: Linked——此属性应有值 FALSE

表 A22 RCV 状态图的转换映象

转 换	条 件
4	当接收来自 SEND 的数据时: ExchangeData. Ind Data Exchange Name Name Scope = AA-SPECIFIC Item Identifier = Value of R-ID Parameter List of Request Data
5	接收的数据的数据类型与 RD-1 到 RD-n 相匹配: List of Request Data Parameter 的数据与 RD-1 到 RD-n 相匹配
6	接收的数据的数据类型与 RD-1 到 RD-n 不匹配: List of Request Data Parameter 的数据与 RD-1 到 RD-n 不匹配
8	当由 SEND 请求复位时: Cancel. Ind Original Invoke ID
9	发现通信问题: 低层指示出错

表 A23 RCV 状态图的动作映象

状 态	动 作
CHECKING	检验数据类型匹配: 检查 List of Request Type Specification 属性和 List of Request Data 参数(见 A4.2)的一致性验证的结果。
NAK_ERROR	发送否定响应到 SEND: ExchangeData.Rsp(-) Error Class=DEFINITION Error Code=TYPE_INCONSISTENT
NAK_BUSY	发送否定响应到 SEND: ExchangeData.Rsp(-) Error Class=SERVIEC Error Code=OBJECT_STATE_CONFLICT
NAK_CANCELLED	发送肯定响应到复位请求并发送否定响应到 SEND: Cancel.Rsp(+) ExchangeData.Rsp(-) Error Class=SERVIEC-PREEMPT Error Code=CANCEL
RESPONDING	发送带用户数据的肯定响应到 SEND: ExchangeData.Rsp(+) List of Response Data=Data given at the SD-i input
NAK_DIS	发送否定响应到 SEND: ExchangeData.Rsp(-) Error Class=ACCESS Error Code=OBJECT_NON_EXISTENT

A4.8 编程报警报告

NOTIFY 和 ALARM 功能块使用作为请求方的 Event Notification 服务。此外,ALARM 功能块使用作为服务器的 Acknowledge Event Notification 服务。(参阅表 A24~表 A27)

NOTIFY 或 ALARM 功能块的一个实例定义带 AA-Specific 范围的 Named Variable 对象。此变量对象将在 Action Result 参数的 Confirmed Servie Response 选项的 Read 响应中被传输。此对象的属性如下:

对象:Named Variable

——属性:Variable Name——变量名,在 EV_ID 输入参数中给出的字符串是带 AA-Specific 范围的变量名。

——属性:Reference to Access Control List——引用存取控制表,此对象将引用 Access Control List 对象 M_ReadOnly。

——属性:Type Decription——类型描述,如果仅给出一个发送数据参数,则此果 Named Variable 对象的类型就是使用表 A1 和表 A3 的发送数据的类型。如果编程了若干个发送数据参数,则 Named Variable 对象的类型是一个包含以发送数据作为分量的结构,这些分量的次序和类型与给出的发送数据参数相同。

NOTIFY 或 ALARM 功能块的一个实例定义带 AA-Specific 范围的 Event Condition 对象。此对象的属性如下:

对象: Event Condition

——属性: Event Condition Name——事件条件名, 此属性具有 EV_ID 输入参数的值。

——属性: Event Condition Class——事件条件类别, 此属性具有值 MONITORED。

——属性: Reference to Access Control List——引用存取控制表, 此对象引用 Access Control List 对象 M_NonDeletable。

——属性: Priority——优先权, 此属性的值应由实现者指定。

——属性: Severity——强度, 此属性从功能块调用的 SEVERITY 输入参数获得。

——属性: List of Event Enrolment Reference——事件登记引用表, 此属性包含对由此功能块实例预先定义的 Event Enrolment 对象的引用。

——属性: Enabled——可用, 此属性从功能块调用的 EN_R 输入参数获得。

——属性: Alarm Summary Reports——报警摘要报告, 此属性具有值 FALSE。

——属性: Monitored Variable Reference——监视变量引用, 此属性应具有值 UNSPECIFIED。

——属性: Evaluation Interval——评估时间间隔, 不支持此属性, 因为在 ISO/IEC 9506-5 的任何一致性类别中不请求参数 CBB 和 CEI。

NOTIFY 或 ALARM 功能块的一个实例定义带 AA-Specific 范围的 Event Enrolment 对象, 此对象属性如下:

对象: Event Enrolment

——属性: Event Enrolment Name——事件登记名, 此属性具有 EV_ID 输入参数的值。

——属性: Reference to Access Control List——引用存取控制表, 此对象将引用 Access Control List 对象 M_NonDeletable。

——属性: Enrolment Class——登记类别, 此属性具有值 NOTIFICATION。

——属性: Event Condition Reference——事件条件引用, 引用由此功能块对象预先定义的 Event Condition。

——属性: Notification Lost——通知丢失, 此属性的值由实现者指定。

——属性: Event Action Reference——事件动作引用, 此属性包含对由此功能块实例预先定义的 Event Action 对象的引用。

——属性: Duration——持续时间, 此属性具有值 PERMANENT。

——属性: Alarm Acknowledgement Rule——报警确认规则, 如果 NOTIFY 功能块被编程, 此属性具有值 NONE; 如果编程了 ALARM 功能块, 则此属性具有值 ACK_ALL。

如果在 NOTIFY 或 ALARM 功能块的实例中编程了一个或多个事件数据输入 SD_i, 则此实例定义带 AA-Specific 范围的 Event Action 对象。此对象的属性如下:

对象: Event Action

——属性: Event Action Name——事件动作名, 此属性具有 EV_ID 输入参数的值。

——属性: Reference to Access Control List——引用存取控制表, 此对象将引用 Access Control List 对象 M_NonDeletable。

——属性: Confirmed Service Request——确认的服务请求, Read 服务被用作 Event Action。此服务的变元应是:

——属性: Specification With Result-FALSE。

——属性: Variable Access Specification——此变元应指定功能块实例的相关变量。

——属性: List of Modifier——此属性应是空。

——属性: List of Event Enrolment Reference——此属性包含对由功能块实例预先定义的 Event Enrolment 对象的引用。

表 A24 NOTIFY 状态图的转换映象

转 换	条 件
4	通信系统指示“Sent to remote Communication partner”; 低层指示无出错
5	通信系统指示“Cannot Sent to remote Communication partner”; 低层指示出错

表 A25 NOTIFY 状态图的动作映象

状 态	动 作
TRY_UP	如果 Event Condition 处于 DISABLED 状态,则将报告一个带状态码 10 的出错,否则请求给远程通信伙伴报告到来的报警; EventNotification. Req Event Enrolment Name Name Scope = AA-SPECIFIC Item Identifier = Value of EV_ID parameter Event Condition Name Name Scope = AA-SPECIFIC Item Identifier = Value of EV_ID parameter Severity = Value of SEVERITY parameter Current State = ACTIVE Alarm Acknowledge Rule = NONE Action Result Event Action Name Name Scope = AA-SPECIFIC Item Identifier = Value of EV_ID parameter Success Confirmed Service Response
TRY_DOWN	如果 Event Condition 处于 DISABLED 状态,则将报告一个带状态码 10 的出错,否则请求给远程通信伙伴报告过去的报警; EventNotification. Req Event Enrolment Name Name Scope = AA-SPECIFIC Item Identifier = Value of EV_ID parameter Event Condition Name Name Scope = AA-SPECIFIC Item Identifier = Value of EV_ID parameter Severity = Value of SEVERITY parameter Current State = IDLE Transition Time = Time the raising edge of the EVENT parameter was detected Alarm Acknowledge Rule = NONE Action Result Event Action Name Name Scope = AA-SPECIFIC Item Identifier = Value of EV_ID parameter Success Confirmed Service Response

表 A26 ALARM 状态图的转换映象

转 换	条 件
6	接收到来的报警报告的确认: AcknowledgeEventNotification. Ind Event Enrolment Name Name Scope=AA-SPECIFIC Item Identifier=Value of EV-ID parameter Acknowledged State=ACTIVE
7	接收离去的报警报告的确认: AcknowledgeEventNotification. Ind Event Enrolment Name Name Scope=AA-SPECIFIC Item Identifier=Value of EV-ID parameter Acknowledged Stae=IDLE

表 A27 ALARM 状态图的动作映象

状 态	动 作
U1,U2,U3	如果 Event Condition 处于 DISABLED 状态,则将报告一个带状态代码 10 的出错,否则请求将到来的报警报告给远程通信伙伴: EventNotification. Req Event Enrolment Name Name Scope= AA-SPECIFIC Item Identifier= Value of EV-ID parameter Event Condition Name Name Scope= AA-SPECIFIC Item Identifier= Value of EV-ID parameter Severity= Value of SEVERITY parameter Current State= ACTIVE NO-ACK-A Transition Time=Time the raising edge of the EVENT parameter was detected Alarm Acknowledge Rule= ACK-ALL Action Result Event Action Name Name Scope=AA-SPECIFIC Item Identifier= Value of EV-ID parameter Success Confirmed Service Response
ACK..UP	肯定地确认所接收的确认: AcknowledgeEventNotification. Rsp(+) 设置由 EV ID 参数标识的所有事件登记的状态给 ACTIVE ACKED

表 A27(完)

状 态	动 作
D2	如果 Event Condition 处于 DISABLED 状态,则将报告一个带状态码 10 的出错,否则请求将离去的报警报告给远程通信伙伴: 参阅状态 U1,但是: EventNotification. Req Current State=IDLE_NO_ACK_A
D1	如果 EventCondition 处于 DISABLED 状态,则将报告一个带状态码 10 的出错,否则请求将离去的报警报告给远程通信伙伴;参阅状态 U1,但是: EventNotification. Req Current State=IDLE ACKED
ACK_DOWN	肯定地确认所接收到的确认: AcknowledgeEventNotification. Rep(+) 设置由 EV_ID 参数标识的所有事件登记的状态给 IDLE ACKED
ERROR	否定地确认所接收到确认: AcknowledgeEvent Notification. Rep(-) Error Class=SERVICE Error Code=OBJECT-STATE-CONFLICT

A4.9 连接管理

CONNECT 功能块使用作为客户机和服务器的 Initiate 服务和 Conclude 服务。(参阅表 A28 和表 A29)

PARTNER 输入被用来标识需要与之连接的应用实体。

表 A28 CONNECT 状态图的转换映象

转 换	条 件
4	接收到来自远程通信伙伴的建立一个新连接请求的肯定的确认: Initiate.cnf(+)
5	接收到来自远程通信伙伴的建立一个新连接请求的否定的确认: Initiate.cnf(-)
6	接收到来自远程通信伙伴的结束此连接请求的肯定的确认: Conclude.cnf(+)
7	接收到来自远程通信伙伴的结束此连接请求的否定的确认: Conclude.cnf(-)
8	连接丢失,例如应用关系的丢失。
9	接收到一个新的建立与远程通信伙伴连接请求: Initiate. Ind
10	接收到一个结束与远程通信伙伴连接请求: Conclude. Ind

表 A29 CONNECT 状态图的动作映象

CHECKING_1	检查是否可能建立连接; 检查在 Initiate. Ind 原语中标识的限制能否被接收,或能否建议其他的值。
DENY	发送一个否定的响应给连接请求: Initiate. Rep(-) Error Type Error class=INITIATE Error code=OTHER
ACCEPT	接收建立连接并发送一个肯定的响应: Initiate. Rsp(+)
CONCL_PASSIV	发送一个肯定的响应给结束连接: Conclude. Rsp(+)
CONNECTING	请求与远程通信伙伴连接: Initiate. Req
ABORT_1	请求中止通信: Abrot. Req
CHECKING_2	检查此连接能否建立: 检查此应用关系能否建立
ABORT_2	请求中止通信: Abort. Req
CLOSING	请求结束此连接 Conclude. Req
CONCLUDING	发送肯定响应给结束连接: Conclude. Req(+)

A5 实现特指的特性和参数

在本附录中定义的实现者特指的特性和参数以及与每个最有关的子条文列于表 A30 中。

表 A30 实现特指的特性和参数

条文	实现特指的特性和参数
A6.1	映象 PLC 系统到 ISO/IEC 9506-5 的 VMD
A6.2	当用 ISO/IEC 9506-5 的抽象语法通信时,对数据类型 STRING 的数据所允许的最大长度。
A6.4	建立附加 Program Invocation 对象的方法
A6.5	建立附加 Domain 对象的方法
A7	用 FB CONNECT 建立的通信通道是可否再用
A7.6	MMS Event Condition 对象的 Priority 属性的值
A7.6	MMS Event Enrolment 对象的 Notification Lost 属性的值

附录 B

(标准的附录)

使用 GB/T 16720.2 的 PLC 行为特性

B1 PLC 通信映射到 MMS

本附录规定 PLC 在使用 GB/T 16720.2 定义的抽象语法进行通信时的行为特性。

GB/T 16979.1 规定 VMD 的模型,同时还规定 VMD 怎样提供服务。ISO/IEC 9506-5 扩展了用于 PLC 的 VMD,例如增加 I/O 状态的概念。为了规定 PLC 的行为特性还扩展了由 PLC 提供的服务。

通常的办法是,PLC 就象是正在使用缺省值以及 ISO/IEC 9506-5 定义的抽象语法进行通信,该缺省值在 ISO/IEC 9506-5 已经扩展的服务适用场合中的缺省值。

对于 CreateProgramInvocation 服务,将使用如表 B1 的缺省:

表 B1 Create ProgramInvocation 服务的缺省

GB/T 15969.3 的 PLC 语言元素	独立的属性	程序调用引用属性
Configuration	TRUE	
Resource	FALSE	配置的 Program Invocation 的标识符
Others	TRUE	

对于如下的服务,PLC 的作用应符合由 ISO/IEC 9506-5 定义的、带有在表 B2 中使用的 I/O State 值的服务扩展,包括有关 Independent and Program Invocation Reference 属性的服务过程扩展。

表 B2 用于 I/O State 参数的 Program Invocation 服务的缺省

MMS 服务	I/O State 值
Start	0-Controlled
Resume	0-Controlled
Stop	3-Implementer State
Kill	3-Implementer State

不使用为 GetProgramInvocation Attributes 服务定义的服务过程扩展。

B2 实现特指的特性和参数

在本标准的本附录中使用的实现者特指的特性和参数以及对每个最有关的子条被列于表 B3 中。

表 B3 实现者特指的特性和参数

条文	实现特指的特性和参数
A6.1	PLC 系统映射到 GB/T 16979.1 的 VMD
A6.2	当用 GB/T 16979.1 通信时,对数据类型 STRING 的数据所允许的最大长度。
A6.4	建立附加 Program Invocation 对象的方式
A6.5	建立附加 Domain 对象的方式
A7	使用 FB CONNECT 建立的通信通道是可否使用还是不可使用
A7.6	MMS Condition 对象的 Priority 属性的值
A7.6	MMS Event Enrolment 对象的 Notification Lost 属性的值