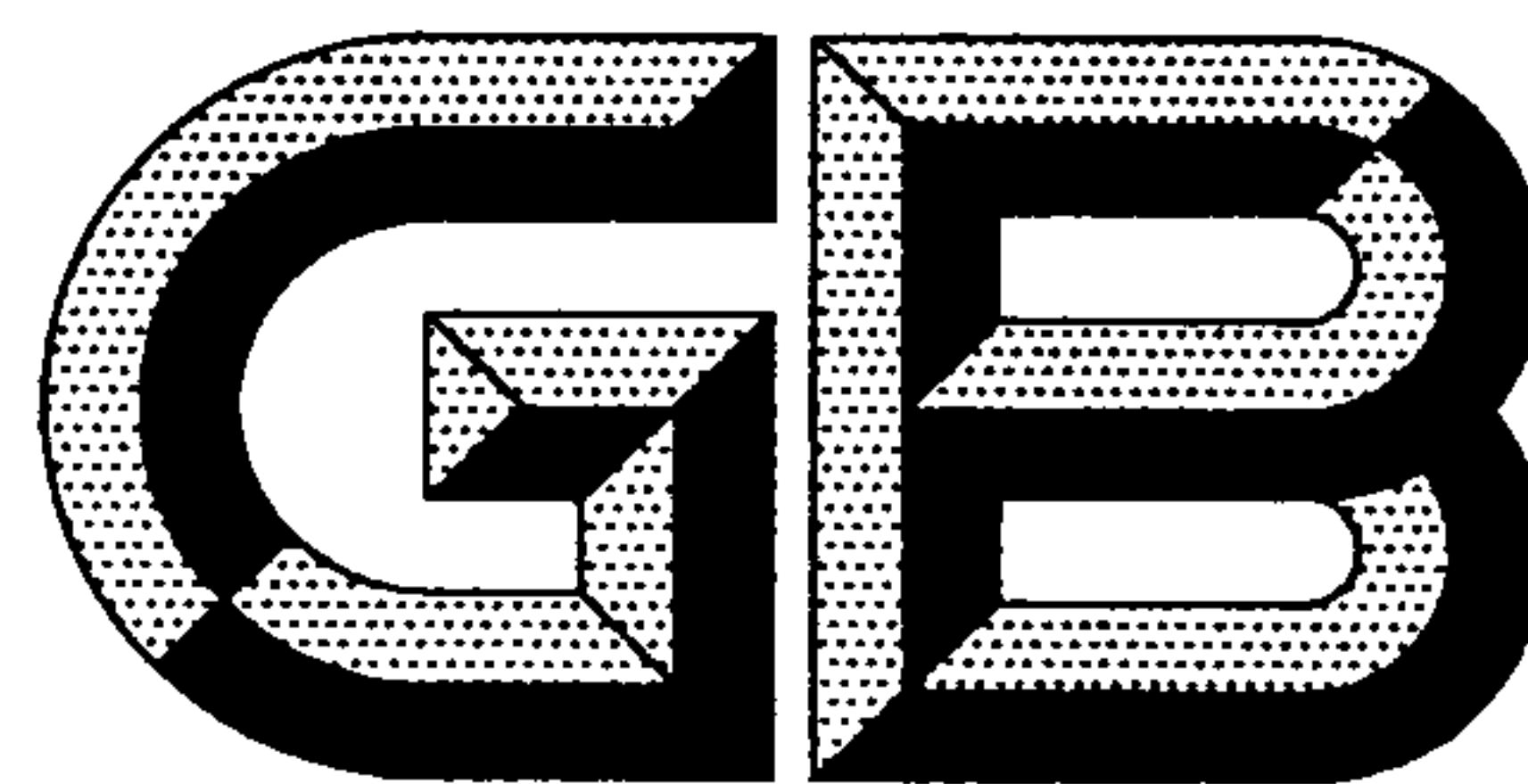




中华人民共和国国家标准

GB/T 33905.2—2017

智能传感器 第2部分：物联网应用行规

Intelligent sensor—Part 2: Application profile for internet of things

广东省网络空间安全协会受控资料

2017-07-31 发布

2018-02-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目次

前言 V

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 物联网应用规范 2

4.1 概述 2

4.2 对象模型 2

4.3 标识对象(类代码=0x01) 2

4.4 寄存器对象(类代码=0x02) 4

4.5 离散的输入点对象(类代码=0x03) 5

4.6 离散的输出点对象(类代码=0x04) 6

4.7 模拟输入点对象(类代码=0x05) 7

4.8 模拟输出点对象(类代码=0x06) 8

4.9 物体感知对象(类代码=0x07) 9

4.10 参数对象(类代码=0x08) 10

4.11 参数组对象(类代码=0x09) 12

4.12 组对象(类代码=0x0A) 13

4.13 离散输入组对象(类代码=0x0B) 13

4.14 离散输出组对象(类代码=0x0C) 14

4.15 离散组对象(类代码=0x0D) 15

4.16 模拟输入组对象(类代码=0x0E) 16

4.17 模拟输出组对象(类代码=0x0F) 17

4.18 模拟组对象(类代码=0x10) 19

4.19 位置传感器对象(类代码=0x11) 20

4.20 位置控制器的监控对象(类代码=0x12) 21

4.21 位置控制器对象(类代码=0x13) 24

4.22 电机数据对象(类代码=0x20) 27

4.23 控制监视器对象(类代码=0x14) 27

4.24 交流/直流驱动器对象(类代码=0x15) 29

5 抽象语法规范 33

5.1 传输格式规范 33

5.2 抽象语法符号 33

5.3 控制网络数据规范 33

5.4 数据类型规范/词典 35

6 应用程序传输语法:紧凑型编码 38

6.1 紧凑型编码 38

6.2 紧凑型编码的约束条件 39

6.3 举例	39
7 数据类型报告	45
7.1 对象数据描述	45
7.2 基本数据类型报告	46
7.3 构建数据类型报告	47
参考文献	51
图 1 定长位串紧凑型编码的比特排列规则	42
图 2 BYTE 定长字符串紧凑型编码的例子	42
图 3 WORD 定长字符串紧凑型编码的例子	43
图 4 DWORD 定长字符串紧凑型编码的例子	43
图 5 LWORD 定长字符串紧凑型编码的例子	43
图 6 结构类型规范正规编码的例子	47
图 7 结构类型规范缩略编码的例子	48
图 8 数组类型规范正规编码的例子 1	48
图 9 数组类型规范正规编码的例子 2	49
图 10 数组类型规范缩略编码的例子 1	49
图 11 数组类型规范缩略编码的例子 2	50
表 1 标识对象实例属性	2
表 2 设备类型	4
表 3 寄存器对象实例属性	5
表 4 离散的输入点对象实例属性	5
表 5 离散的输出点对象实例属性	6
表 6 模拟输入点对象实例属性	7
表 7 模拟输出点对象实例属性	8
表 8 物体感知对象实例属性	9
表 9 参数对象实例属性	10
表 10 参数组对象实例属性	12
表 11 组对象实例属性	13
表 12 离散输入组对象实例属性	14
表 13 离散输出组对象实例属性	15
表 14 离散组对象实例属性	16
表 15 模拟输入组对象实例属性	16
表 16 模拟输出组对象实例属性	18
表 17 模拟组对象实例属性	19
表 18 位置传感器对象实例属性	20
表 19 位置控制器的监控对象实例属性	21
表 20 位置控制器的监控对象原点(Home)和指针(Index)属性	22
表 21 位置控制器的监控对象(Registration)属性	23
表 22 位置控制器的监控对象轴跟随(Axis Following)属性	24
表 23 位置控制器对象 Profile 属性	24

表 24	电机数据对象实例属性	27
表 25	控制监督对象实例属性	27
表 26	交流/直流驱动对象实例属性	30
表 27	交流/直流驱动器对象基本量的名称和单位符号	33
表 28	BOOLEAN(布尔)编码	39
表 29	BOOL(布尔)值紧凑型编码的例子	39
表 30	SignedInteger(符号整型)值的编码	39
表 31	SignedInteger(符号整型)值的紧凑型编码举例	39
表 32	UnsignedInteger(无符号整型)值的编码	40
表 33	UnsignedInteger(无符号整型)的紧凑型编码举例	40
表 34	FixedLengthReal(定长实数)值的编码	40
表 35	REAL(实数)值的紧凑型编码举例	40
表 36	LREAL(长实数)值的紧凑型编码举例	40
表 37	时间值的编码	41
表 38	STRING 值	41
表 39	STRING2 值	41
表 40	STRINGN 值	41
表 41	SHORT_STRING 值	41
表 42	STRING 值的紧凑型编码举例	42
表 43	STRING2 值的紧凑型编码举例	42
表 44	SHORT_STRING 类型	42
表 45	一维 ARRAY(数组)紧凑型编码的例子	43
表 46	多维 ARRAY(数组)紧凑型编码的例子	44
表 47	STRUCTURE(结构)紧凑型编码的例子	44
表 48	基本数据类型的标识代码和描述	46
表 49	结构类型规范正规编码的例子	47

前 言

GB/T 33905《智能传感器》分为 5 部分：

- 第 1 部分：总则；
- 第 2 部分：物联网应用行规；
- 第 3 部分：术语；
- 第 4 部分：性能评定方法；
- 第 5 部分：检查和例行试验导则。

本部分是 GB/T 33905 的第 2 部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

本部分由中国机械工业联合会提出。

本部分由全国工业过程测量控制和自动化标准化技术委员会(SAC/TC 124)归口。

本部分起草单位：福建上润精密仪器有限公司、机械工业仪器仪表综合技术经济研究所、北京国电智深控制技术有限公司、山东省计算中心(国家超级计算济南中心)、济南市长清计算机应用公司、重庆市伟岸测器制造股份有限公司、施耐德电气(中国)有限公司。

本部分主要起草人：林仁祥、梅恪、田雨聪、李刚、岳宗龙、唐田、杜佳琳、戈剑、柳晓菁、王成城、周鸣乐、欧文辉、汪付强。

广东省网络空间安全协会受控资料

智能传感器 第2部分：物联网应用行规

1 范围

GB/T 33905 的本部分规定了物联网应用使用的智能传感器、执行器、二进制设备以及其他装置用于操作、调试、维护和诊断的基本设备参数集。本部分还规定了抽象语法规则、应用程序传输语法和数据类型报告。

本部分适用于物联网中应用的智能传感器。本部分也适用于其他类型的传感器(前提是预先对其差异进行考虑)。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

ISO/IEC 10646 信息技术 通用多重八位组编码字符集[Information technology—Universal coded character set(UCS)]

3 术语和定义

下列术语和定义适用于本文件。

3.1

兼容性 compatibility

当设备按照本部分的规定而设计时,无需加以改动即可与其他设备相互连接和使用的程度(例如:机械、电气、功能)。

3.2

接口 interface

系统与另一系统(或系统的某些部分)之间的公共边界,信息通过该公共边界传递。

3.3

接口系统 interface system

实现设备之间通信的一组机械、电气和功能元件。

3.4

本地控制 local control

设备接受指令的一种方式,即通过其本地(面板或背板)控制来接受以使设备能执行各种任务。

3.5

远程控制 remote control

设备接受的一种方式,即通过其电气接口连接来接受以使设备能执行不同的任务。

3.6

可编程 programmable

设备的一种特性,即能接收指令来改变其内部程序以执行一个或多个特定任务。

3.7

系统 system

由相互作用、相互依赖的若干组成部分结合而成的,具有特定功能的有机整体。而且这个有机整体又是其从属的更大系统的组成部分。

3.8

信号 signal

信息的物理表现。

注:在本部分中,这是指通常在一般意义下称之为“信号”的一种狭义的定义,并在下文中都是指数字式电信号而言。

4 物联网应用规范

4.1 概述

本部分主要面向由测量、激励以及与控制器互连的传感器组成的物联网系统,也可直接用于中小规模的物联网系统。本部分规定了一套形式化描述方法,以便于智能传感器在物联网中进行信息交换,描述方法涉及:语法结构、公共数据结构、状态信息、组态方式等。本部分以典型智能传感器的功能描述为基础,可标准化同一系统中不同制造商设备的行为。

本部分除定义了各种与智能传感器相关的信息外,还对传感接口功能进行了扩展和进一步解释,但同时保持了标准的兼容性。

应用本部分并不能百分之百地覆盖用户的应用层次上对系统兼容性的要求。

注:本部分规定之外的传感器类型,可参照近似类型进行自定义。

4.2 对象模型

对象模型体现智能传感器的虚拟构架。智能传感器被模型化为对象的集合。每一个对象是相关的服务、属性和行为的集合。服务是对象执行的程序。属性是由数据体现的、量化的对象的特征。对象的行为是指示对象如何响应特殊事件。

4.3 标识对象(类代码=0x01)

标识对象应当提供关于设备的标志和一般信息。标识对象应出现于所有的设备中。标识对象应支持如表 1 规定的实例属性。

表 1 标识对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	必要	获取 (Get)	厂商 ID (Vendor ID)	UINT (无符号整型)	每个厂商的唯一标识	厂商 ID 唯一地标识一个具体的厂商。值 0 不使用
2	必要	获取 (Get)	设备类型 (Device Type)	UINT (无符号整型)	产品通用类型的标识	设备类型应当用于标识正在使用中的特定设备的设备行规。设备行规定义了设备应当实现的最小要求和一般选项。见表 2

表 1（续）

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
3	必要	获取 (Get)	产品代码 (Product C)	UINT (无符号整型)	各厂商特殊产品的标识	被分配了产品代码的厂商应当通过设备类型识别特定产品。每个厂商应当分配这个代码给每一个设备。典型的,产品代码反映一个或多个产品目录/型号
4	必要	获取 (Get)	版本 (Revision)	STRUCT (结构体)	标识对象的实例体现的版本	由主要版本和次要版本组成的版本属性应当标识标识对象所体现的条目的版本。 一个符合标准的产品的 主要版本字段和次要版本 字段都不能为 0。主要和 次要版本被代表性地显示 为 major.minor。次要版本 应为以 0 开头的 3 位数字。 主要版本限制为 7 比特位。 第 8 比特位被保留并且应 为 0
			主版本 (Major Revision)	USINT (无符号双整型)		
			次版本 (Minor Revision)	USINT (无符号单整型)		
5	必要	获取 (Get)	状态 (Status)	WORD (字)	设备状态摘要	这个属性应当体现当前整个设备的状态。它的值随着设备状态的变化而变化。 状态属性是 WORD 类型
6	必要	获取 (Get)	序列号 (Serial Number)	UDINT (无符号双整型)	设备序列号	这个属性提供了一个号码和厂商 ID 连接起来组成每个设备的唯一标识符。 每个厂商应当有责任保证它所有设备序列号的唯一性
7	必要	获取 (Get)	产品名称 (Product Name)	SHORT_STRING (短字符串)	人可读的标识	本文本串是由产品代码 (属性 3) 所表示的产品或者产品系列的简短描述。同一个产品代码可以有不同的产品名称串。这个串中最多包含 32 个 8 位字符。每个字符应当在 0x20~0x7E 范围内

设备类型见表 2。

表 2 设备类型

设备类型名称	设备类型号
一般设备	0x00
控制站	0x01
AC 驱动器	0x02
电机过载保护器	0x03
限位开关	0x04
感应接近开关	0x05
光电传感器	0x06
一般用途离散 I/O	0x07
编码器	0x08
旋转变压器	0x09
一般用途模拟 I/O	0x0A
为将来使用保留	0x0B
通信适配器	0x0C
条形码扫描器	0x0D
可编程逻辑控制器	0x0E
为将来使用保留	0x0F
定位控制器	0x10
称重秤	0x11
信息显示器	0x12
DC 驱动器	0x13
伺服驱动器	0x14
接触器	0x15
电机启动器	0x16
软启动器	0x17
人机接口	0x18
气动阀	0x19
物理层	0x32

4.4 寄存器对象(类代码 = 0x02)

寄存器对象被赋地址范围从 1 bit 到 64 kbit。它可以做为一个生产者(输入)寄存器或者用户(输出)寄存器。

、寄存器对象支持表 3 规定的实例属性。

表 3 寄存器对象实例属性

属性 ID	实现 必要性	访问 规则	名称	数据 类型	属性描述	值语义
1	必要	获取(Get)	Bad Flag	BOOL(布尔型)	如果设置 1,那么属性 4 可以包含无效的、坏的或者其他误用的数据	0=好 1=坏
2	必要	设置(Set)	Direction	BOOL(布尔型)	数据传输的 Direction	0=生产者寄存器 1=用户寄存器
3	必要	设置(Set)	Size	UINT (无符号整型)	寄存器数据的大小以比特表示	数据大小以比特表示
4	必要	有条件的 ^a	Data	比特队列	被传送的数据	^b
^a 设置的访问规则是可选的,如果属性 2,那么 Direction=1;如果 Direction=0,那么访问规则可获取。 ^b 所有已编码比特应当是排列的 LSO。						

4.5 离散的输入点对象(类代码=0x03)

离散的输入点(DIP)对象模拟了产品中的离散输入。在应用中,对象可以被用于简单的 toggle 开关的设备或者复杂的离散的 I/O 控制设备。离散的输入点对象应生产数据到网络上。离散的输入点接口是实际的输入点,比如开关或者 screw 终端。输入应当被采样并且数据被存储在对象的值属性中。离散的输入值的采样应通过外部命令被触发。

离散的输入点对象支持表 4 规定的实例属性。

表 4 离散的输入点对象实例属性

属性 ID	实现 必要性	访问规则	名称	数据 类型	属性描述	值语义
1	可选	获取(Get)	属性数	USINT(无符号单整型)	这个产品支持的数	—
2	可选	获取(Get)	属性列表	USINT(无符号单整型)队列	产品支持的属性列表	—
3	必要	获取(Get)	值	BOOL(布尔型)	输入点值	0=off 1=on
4	可选	获取(Get)	地位	BOOL(布尔型)	输入点地位	0=OK 1=产品特定警告或者地位
5	可选	设置(Set)	Off_On Delay	UINT(无符号整型)	从 off 到 on 的转换的滤波时间是 0~65 535 μs ^a	默认值 0
6	可选	设置(Set)	On_Off Delay	UINT(无符号整型)	从 on 到 off 的转换的滤波时间是 0~65 535 μs ^b	默认值 0
^a 在 on 状态被记录到值属性之前,输入应在由 Off_On Delay 属性规定的全部滤波时间的数量上。 ^b 在 off 状态被记录到值属性之前,输入应在由 On _Off Delay 属性规定的全部滤波时间的数量上。						

4.6 离散的输出点对象(类代码=0x04)

离散的输出点 (DOP)对象是产品中的离散输出的模型化。输出可被外界设备使用。

离散的输出点接口是实际的输出点,例如继电器或者 LED。输出能够从这个对象的值属性中读出并且被应用到输出端子(比如螺丝端子)。

离散的输出点对象支持表 5 规定的实例属性。

表 5 离散的输出点对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取 (Get)	属性数 (Number of Attribute)	USINT(无符号单整型)	在产品中支持的属性数	—
2	可选	获取 (Get)	属性列 (Attribute List)	USINT(无符号单整型)队列	产品支持的属性的列表	—
3	必要	设置 (Set)	值 (Value)	BOOL(布尔型)	输出点值	0=off 1=on
4	可选	获取 (Get)	状态 (Status)	BOOL(布尔型)	输出点状态	0=OK 1=失效或者报警
5	可选	设置 (Set)	故障动作 (Fault Action)	BOOL(布尔型)	在可恢复的故障状态对输出值采取的动作	0=故障值属性 1=保持最后状态
6	可选	设置 (Set)	故障值 (Fault Value)	BOOL(布尔型)	在故障状态属性使用的用户定义的值	0=off 1=on
7	可选	设置 (Set)	空闲动作 (IdleActioin)	BOOL(布尔型)	在可恢复的故障状态对输出值采取的动作	0=空闲值属性 1=保持最后状态
8	可选	设置 (Set)	空闲值 (Idle Value)	BOOL(布尔型)	在空闲状态属性使用的用户定义的值	0=off 1=on
9	可选	设置 (Set)	Run_Idle_Command	BOOL(布尔型)	产生 Receive_Idle 或者 Receive_Ready_to_Run 事件	0=Receive_Idle 1=Receive_Ready_to_Run
10	可选	设置 (Set)	闪烁 (Flash)	BOOL(布尔型)	如果点是在 on,那么周期性地闪烁输出	0=无闪烁 1=闪烁
11	可选	设置 (Set)	闪烁速率 (Flash Rate)	USINT(无符号单整型)	闪烁属性的闪烁速率	无符号正整数显示以 Hz 为单位的频率 比如 1=1 Hz
12	可选	获取 (Get)	对象状态 (Object State)	USINT(无符号单整型)	对象状态	1=不存在 2=可用的 3=空闲 4=就绪 5=运行 6=可恢复的故障 7=不可恢复的故障 255=保留

4.7 模拟输入点对象(类代码=0x05)

模拟输入点(AIP)对象是产品中模拟输入的模型化。它可以用于如:单个模拟点那样简单的应用,也可以用于如:模拟 I/O 控制设备那样复杂的应用。术语“输入”是从网络的角度定义的。一个输入应向网络生产数据。

模拟输入点接口是接到实际的输入点,比如热电偶或者压力传感器。采样输入并且将数据储存在对象的 VALUE(值)属性中。模拟输入值的取样是由外部命令(输入状态改变,循环数据触发器等)触发的。模拟输入对象应支持表 6 规定的类属性。

表 6 模拟输入点对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值的语义
1	可选	获取 (Get)	属性数 (Number of Attributes)	USINT(无符号单整型)	被支持的属性数	—
2	可选	获取 (Get)	属性列表 (Attributes List)	USINT(无符号单整型)队列	该点支持的属性列表	—
3	必要	获取 (Get)	值(Value)	INT(整型)或者基于属性 8	模拟输入值。数据类型缺省值为 INT 但是可以基于属性 8 改变	—
4	可选	获取 (Get)	状态 (Status)	BOOL(布尔型)	点在没有报警或者故障时运行	0=OK 1=报警状态
5	可选	获取 (Get)	所有者厂商 ID (Owner Vender ID)	UINT(无符号整型)	通道所有者的厂商 ID	—
6	可选	获取 (Get)	所有者序列号 (Owner Seiral Number)	UDINT(无符号双整型)	通道所有者的 32 位序列号	—
7	可选	设置 (Set)	输入范围 (Input Range)	USINT(无符号单整型)	点运行的输入范围	0=-10 V~10 V 1=0~5 V 2=1 V~5 V 3=0~10 V 4=4 mA~20 mA 5=-15 mV~75 mV 6=-15 mV~30 mV 255=保留
8	可选	设置 (Set)	值数据类型 (Value Data Type)	USINT(无符号单整型)	确定值的数据类型	0=INT 1=REAL 2=USINT 3=SINT 4=DINT 5=LINT 6=UINT 7=UDINT 8=ULINT 9=LREAL 100=厂商特定的

4.8 模拟输出点对象(类代码=0x06)

模拟输出点(AOP)对象是产品中模拟输出的点级属性和服务的模型化。它可以被用于模型化电压输出或者模拟 I/O 控制设备的部件。术语“输出”是从网络的角度定义的。一个输出将被来自网络的数据使用。

模拟输出点接口是实际的输出点,比如电机驱动的阀(MOV)或者线性定位器。它从这个对象的值属性读取输出,并且施加到输出端子(比如螺丝端子)。

模拟输出点对象应支持表 7 规定的类属性。

表 7 模拟输出点对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取 (Get)	属性数	USINT(无符号单整型)	属性支持的数	0~255
2	可选	获取 (Get)	属性列表	ARRAY of USINT(无符号单整型数组)	由点支持的属性的列表	—
3	必要	设置 (Set)	值	INT(整型)或根据属性 8	模拟输出值。给 INT 的数据类型缺省,但根据属性 8 可以改变	
4	可选	获取 (Get)	状态	BOOL(布尔型)	点运行时无报警或故障	0=OK 1=报警状态
5	可选	获取 (Get)	所有者厂商 ID	UINT(无符号整型)	通道所有者厂商标识	—
6	可选	获取 (Get)	所有者厂商 ID	UDINT(无符号双整型)	通道所有者的 32 位序列号	—
7	可选	设置 (Set)	输出范围	USINT(无符号单整型)	所用输出通道规定的输出范围	0=4 mA~20 mA 1=0~10 V 2=0~20 mA 3=-10 V~10 V 4=0~5 V 5=1 V~5 V 6=-5 V~5 V 255=备用
8	可选	设置 (Set)	值数据类型	USINT(无符号单整型)	确定值的数据类型	0=INT 1=REAL 2=USINT 3=SINT 4=DINT 5=LINT 6=UINT 7=UDINT 8=ULINT 9=LREAL 100=厂商特定的

表 7（续）

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
9	可选	设置 (Set)	故障状态	USINT(无符号单整型)	输出值趋于失效或故障	0=保持最后状态 1=低限 2=高限 3=用户规定值
10	可选	设置 (Set)	空闲状态	USINT(无符号单整型)	输出值趋于空闲状态	—
11	可选	设置 (Set)	故障值	INT(整型)或根据属性 8	用户定义的故障模式时的输出值。若故障状态=3,使用用户规定的值	—
12	可选	设置 (Set)	空闲值	INT(整型)或根据属性 8	用户定义的空闲模式时的输出值。若空闲状态=3,使用用户规定的值	—
13	可选	设置 (Set)	命令	BOOL(布尔型)	改变 AOP 的状态至空闲模式或运行模式	0=空闲 1=运行
14	可选	获取 (Get)	对象状态	USINT(无符号单整型)	对象的状态	1=不存在 2=可用 3=空闲 4=就绪 5=运行 6=可恢复故障 7=不可恢复故障 255=保留

4.9 物体感知对象(类代码=0x07)

该对象感知现实世界目标的存在或不存在。
物体感知对象支持的实例属性见表 8。

表 8 物体感知对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	必要	获取 (Get)	输出	BOOL(布尔型)	0=切换元件开 1=切换元件闭	—
2	可选	获取 (Get)	属性数	USINT(无符号单整型)	支持的属性数	—
3	可选	获取 (Get)	属性列表	ARRAY of USINT (无符号单整型数组)	属性支持的列表	—

表 8 (续)

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
4	可选	获取(Get)	诊断	BOOL(布尔型)	0=OK 1=故障	—
5	可选	设置(Set)	on 延迟	UINT(无符号整型)	0~65 535 ms	—
6	可选	设置(Set)	off 延迟	UINT(无符号整型)	0~65 535 ms	—
7	可选	设置(Set)	单触发延迟 (One Shot Delay)	UINT(无符号整型)	0~65 535 ms	—
8	可选	设置(Set)	操作方式	BOOL(布尔型)	0=规定的输出属性 1=输出属性取反	—
9	可选	设置(Set)	灵敏度	USINT(无符号单整型)	0~255	—
10	可选	获取(Get)	目标边界	USINT(无符号单整型)	0~255	—
11	可选	获取(Get)	背景边界	USINT(无符号单整型)	0~100	—
12	可选	设置(Set)	最小检测距离	UINT(无符号整型)	0~65 535 mm	—
13	可选	设置(Set)	最大检测距离	UINT(无符号整型)	0~65 535 mm ^a	—
14	可选	获取(Get)	检测距离	UINT(无符号整型)	0~65 535 mm	—
^a 最大检测距离应大于或等于最小检测距离。						

4.10 参数对象(类代码=0x08)

参数对象提供为设备配置数据用的已知公用接口。该对象也提供对设备每个单独配置的参数进行定义及描述所必需的所有信息。

该对象提供参数的全部描述使设备能充分标识可配置的参数,它包括最小及最大值,以及可供人读出描述参数的文本信息串。

对每个可配置参数都有一个该对象类的实例,它需要或期望有一个公用配置的接口。每个实例都被链接到可配置参数中的一个,它可能是设备内其他对象之一的一个属性。变更参数对象的参数值属性(实例参数 1),将通过链接路径属性(属性值由参数对象实例指出)示出属性值内产生相应的变化。

参数对象只支持示于表 9 中的实例属性。

表 9 参数对象实例属性

属性 ID	实现必要性	访问规则	基本型/整个型	名称	数据类型	属性的描述	值语义
1	必要	^a	基本	参数值	数据类型在描述符中规定	参数的实际值。数据类型在描述符、数据类型及数据大小中规定	^{a, b}
2	必要	设置(Set)	基本	链路路径大小	USINT(无符号单整型)	链路路径大小属性,若该属性为零,则规定为无链路	属性 3 的字节数

表 9（续）

属性 ID	实现必要性	访问规则	基本型/整个型	名称	数据类型	属性的描述	值语义
3	必要	设置 (Set)	基本	链路路径	路径段的 ARRAY (数组)	从该参数值被取回处到对象的路径	链路路径限制在 255 字节内
				段类型/端口	BYTE(字节)	网络及传输层	—
				段地址	路径(根据包含在段类型/端口内的数据格式)	—	—
4	必要	获取 (Get)	基本	描述符	WORD(字)	参数描述	—
5	必要	获取 (Get)	基本	数据类型	USINT(无符号单整型)	数据类型码	—
6	必要	获取 (Get)	基本	数据大小	USINT(无符号单整型)	属性 1 内字节数, 参数值	—
7	有条件	获取 (Get)	基本	参数名字符串	SHORT_STRING (短字符串)	人可读的字符串, 代表着参数名, 例如“frequency # 1”	最大字符数为 16
8	有条件 ^c	获取 (Get)	全部	单位字符串	SHORT_STRING (短字符串)	工程单位字符串	最大字符数为 4
9	有条件 ^c	获取 (Get)	全部	帮助字符串	SHORT_STRING (短字符串)	帮助字符串	最大字符数为 64
10	有条件 ^c	获取 (Get)	全部	最小值	与属性 1 同	对可设置的属性 1 参数值有效的最小实际值	—
11	有条件	获取 (Get)	全部	最大值	与属性 1 同	对可设置的属性 1 参数值有效的最大实际值	—
12	有条件	获取 (Get)	全部	缺省值	与属性 1 同	对可设置的属性 1 参数值有效的缺省值	—
13	有条件	获取 (Get)	全部	标度变换乘法因子	UINT(无符号整型)	标度变换公式的乘法因子	—
14	有条件	获取 (Get)	全部	标度变换除法因子	UINT(无符号整型)	标度变换公式的除法因子	—
15	有条件	获取 (Get)	全部	标度变换基数	UINT(无符号整型)	标度变换公式的基数	—
16	有条件	获取 (Get)	全部	标度变换偏移	INT(整型)	标度变换公式的偏移	—

表 9（续）

属性 ID	实现必要性	访问规则	基本型/整个型	名称	数据类型	属性的描述	值语义
17	有条件	获取 (Get)	全部	乘法因子链接	UINT (无符号整型)	乘法因子源的对象实例数	—
18	有条件	获取 (Get)	全部	除法因子链接	UINT (无符号整型)	除法因子源的对象实例数	—
19	有条件	获取 (Get)	全部	基数链接	UINT (无符号整型)	基数源的对象实例数	—
20	有条件	获取 (Get)	全部	偏移链接	UINT (无符号整型)	偏移源的对象实例数	—
21	有条件	获取 (Get)	全部	小数精度	USINT (无符号单整型)	当显示换算的工程值时,规定的十进制小数位数。也用于确定实际增量值,因此当增加一位时,会对该精度换算的工程值产生变化	—
^a 访问规则定义在实例属性 4 描述符的比特 4 中。若比特 4 为 0,则访问规则为设置 (Set),参数值可以读写。若比特 4 为 1,则访问规则为获取 (Get),参数值只读。 ^b 数据类型规定:实例属性 4 (描述符)、5 (数据类型)及 6 (数据大小)。 ^c 设备内参数对象实例要么为全部,要么为基本。若为基本,则这些实例属性为不可用的并将不被实现。若所有参数对象实例为全部,则这些实例参数将是必须的,并被实现。若某一设备不需应用属性,则可用表内所列出的缺省值。							

4.11 参数组对象 (类代码 = 0x09)

参数组对象将在设备内对参数组进行标识并提供访问,参数组对象也将对有关参数集 [设置 (Set)] 提供方便的访问。

参数组对象支持如表 10 所规定的实例属性。

表 10 参数组对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	必要	获取 (Get)	组名字符串	SHORT _ STRING (短字符串)	表示组名可读的字符串 (如 set-up, frequency set)	最大字符数 = 16
2	必要	获取 (Get)	组内成员数	UINT (无符号整型)	组内参数数	—
3	必要	获取 (Get)	组内第 1 个参数号	UINT (无符号整型)	参数实例数	—
4	必要	获取 (Get)	组内第 2 个参数号	UINT (无符号整型)	参数实例数	—
n	必要	获取 (Get)	组内第 (n - 2) 个参数号	UINT (无符号整型)	参数实例数	—

4.12 组对象(类代码=0x0A)

组对象与其他对象绑定在一起,具有代表性的有离散与模拟,含有 AIP、AOP、AIG、AOG、DIP、DOP、DOG 或 DIG。对象绑定成组将支持同一属性与服务,它们在产品中应用于所有绑定的对象(输入与输出二者,离散与模拟二者)。

在上电时,建立组的一览表以及/或将点捆绑成组,作为绑定的一览表是静态的而且是组的 Get-only 属性。

- 组对象可应用于:
- 在有多个模拟及/或离散输入及输出点以及/或组之间的时候,可共享多个属性。
 - 用可影响所有组内成员的支持服务进行更有效地访问数据。

组对象支持的实例属性规定于表 11。

表 11 组对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取(Get)	属性数	USINT(无符号单整型)	由组支持的属性数	—
2	可选	获取(Get)	属性列表	USINT OFF ARRAY(无符号单整型数组)	支持的属性的列表	—
3	可选	获取(Get)	绑定实例的数	USINT(无符号单整型)	组内的点数	—
4	可选	获取(Get)	绑定	ARRAY of STRUCT: UINT(结构数组: 无符号整型)	组内的实例列表,类 ID,实例 ID	—
5	可选	获取(Get)	状态	BOOL(布尔型)	组运行于无报警或故障	0=OK 1=报警状态
6	可选	获取(Get)	所有者厂商 ID	UINT(无符号整型)	组所有者的厂商 ID	—
7	可选	获取(Get)	所有者序列号	UINT(无符号整型)	组所有者的 32 位序列号	—

4.13 离散输入组对象(类代码=0x0B)

离散输入组(DIG)对象应绑定在设备的一组离散输入点上。绑定到组的所有点应共享包含在组里的所有属性。如果一个属性被多个离散输入点(DIP)共享(这些点有同样的属性和同样的属性值),那么这个属性可能包含在一个离散输入组里。一个离散输入点可能被绑定到多个离散输入组。

绑定到组的离散输入点列表应在上电时建立,因为绑定列表是静态的,并且是组的 Get-only 属性。离散输入组对象可能被使用:

- 当在许多输入点共享一个单独属性时;
- 为了更有效地访问数据。

离散输入组对象应支持表 12 中规定的实例属性。

表 12 离散输入组对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取(Get)	属性数(Number of Attributes)	USINT(无符号单整型)	设备支持的属性数	—
2	可选	获取(Get)	属性列表(Attribute ListAttribute List)	ARRAY of USINT(无符号单整型数组)	设备支持的属性的列表	—
3	可选	获取(Get)	已绑定实例数(Number of Bound Instances)	USINT(无符号单整型)	绑定到该组的点数	—
4	可选	获取(Get)	绑定(Binding)	ARRAY of UINT(无符号整型数组)	绑定到该组的所有点的列表实例 ID	Class DIP Instance # X Class DIP Instance # Y...
5	可选	获取(Get)	状态(Status)	BOOL(布尔型)	该组中所有离散输入点的状态	0=OK 1=产品特定的报警或状态
6	可选	获取(Get)	Off_On 延时(Off_On Delay)	UINT(无符号整型)	从 off 到 on 的滤波时间 0~65 535 μs ^a	—
7	可选	获取(Get)	On_Off 延时(On_Off Delay)	UINT(无符号整型)	从 on 到 off 的滤波时间 0~65 535 μs ^b	—
^a 在 on 状态被记录到值属性之前,输入应在由 Off_On Delay 属性规定的全部滤波时间的数量上。 ^b 在 off 状态被记录到值属性之前,输入应在由 On_Off Delay 属性规定的全部滤波时间的数量上。						

4.14 离散输出组对象（类代码=0x0C）

对象应绑定在设备的一组离散输入点上。绑定到组的所有点应共享包含在组里的所有属性。如果一个属性被多个离散输出点(DOP)共享,那么这个属性可能包含在一个离散输出组里。一个离散输出点可能被绑定到多个离散输出组。

绑定到组的离散输入点列表应在上电时建立,因为绑定列表是静态的,并且是组的 Get-only 属性。

离散输出组对象可能被使用:

- 当在许多输出点共享一个单独属性时;
- 为了更有效地访问数据。

离散输出组对象应支持表 13 中规定的实例属性。

表 13 离散输出组对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取(Get)	属性数(Number of Attributes)	USINT(无符号单整型)	设备支持的属性数	---
2	可选	获取(Get)	属性列表(Attribute ListAttribute List)	ARRAY of USINT(无符号单整型数组)	设备支持的属性列表	—
3	可选	获取(Get)	已绑定实例数(Number of Bound Instances)	USINT(无符号单整型)	绑定到该组的点数	—
4	可选	获取(Get)	绑定(Binding)	ARRAY of UINT(无符号整型数组)	绑定到该组的所有点的列表实例 ID	—
5	可选	获取(Get)	状态(Status)	BOOL(布尔型)	该组中所有离散输入点的状态	0=OK 1=专用产品报警或状态
6	必要	设置(Set)	命令(Command)	BOOL(布尔型)	该组中所有 DOP 改变状态到空闲模式或运行模式	0=空闲 1=运行
7	可选	设置(Set)	错误状态(Fault State)	BOOL(布尔型)	可恢复故障后的输出状态	0=错误值属性 1=保持最后状态
8	可选	设置(Set)	错误值(Fault Value)	BOOL(布尔型)	用户定义的用于错误状态属性的值	0=关 1=开
9	可选	设置(Set)	空闲状态(Idle State)	BOOL(布尔型)	空闲时的输出状态	0=空闲值属性 1=保持最后状态
10	可选	设置(Set)	空闲值(Idle Value)	BOOL(布尔型)	用户定义的用于空闲状态属性的值	0=关 1=开

4.15 离散组对象(类代码=0x0D)

离散组(DG)对象应绑定其他的离散对象,包括 DIP、DOP、DIG 或 DOG。绑定到 DG 的对象(输入和输出)应支持同样的属性和服务。

你可以在上电时建立绑定到 DG 的离散组和/或点的列表,因为绑定列表是静态的,且是 DG 的 Get-only 属性。

- DG 对象可被用于:
- 当一个单独的属性在许多离散输入和输出点和/或组内共享时;
 - 为了更有效地访问数据。

DG 对象应支持表 14 中规定的实例属性。

表 14 离散组对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取(Get)	属性数 (Number of Attributes)	USINT (无符号单整型)	组支持的属性数	—
2	可选	获取(Get)	属性列表 (Attribute ListAttribute List)	ARRAY of USINT (无符号单整型数组)	被支持的属性列表	—
3	可选	获取(Get)	已绑定实例数 (Number of Bound Instances)	USINT (无符号单整型)	组内点数量	—
4	可选	获取(Get)	绑定 (Binding)	ARRAY of STRUCT: UINT (结构体数组: 无符号整型)	所有实例组类 ID 的列表	—
5	可选	获取(Get)	状态 (Status)	BOOL (布尔型)	组无报警或错误下操作	0=OK 1 = 报警状态

4.16 模拟输入组对象 (类代码=0x0E)

模拟输入组 (AIG) 对象应绑定在设备的一组模拟输入点上。绑定到组的所有点应共享包含在组里的所有属性。如果一个属性被多个模拟输入点 (AIP) 共享 (这些点有同样的属性和同样的属性值), 那么这个属性可能包含在一个模拟输入组里。一个模拟输入点可能被绑定到多个模拟输入组。

绑定到组的模拟输入点列表应在上电时建立, 因为绑定列表是静态的, 并且是组的 Get-only 属性。

模拟输入组对象可能被使用:

- 当在许多输入点共享一个单独属性时;
- 为了更有效地访问数据。

模拟输入组对象应支持表 15 中规定的实例属性。

表 15 模拟输入组对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取(Get)	属性数 (Number of Attributes)	USINT (无符号单整型)	组支持的属性数	—
2	可选	获取(Get)	属性列表 (Attribute ListAttribute List)	ARRAY of USINT (无符号单整型数组)	被支持的属性的列表	—
3	可选	获取(Get)	已绑定实例数 (Number of Bound Instances)	USINT (无符号单整型)	组内的点数	—
4	可选	获取(Get)	绑定 (Binding)	ARRAY of UINT (无符号整型数组)	绑定到该组的所有点的列表实例 ID	—

表 15（续）

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
5	可选	获取(Get)	状态(Status)	BOOL(布尔型)	无报警或错误的组操作	0=OK 1=报警状态
6	可选	获取(Get)	所有者厂商 ID(Owner Vendor ID)	UINT(无符号整型)	组所有者的厂商 ID	—
7	可选	获取(Get)	所有者序列号(Owner Serial Number)	UDINT(无符号双整型)	32 位组所有者序列号	—
8	可选	设置(Set)	值数据类型(Value Data Type)	USINT(无符号单整型)	规定 AIP 值的数据类型	0=INT 1=REAL 2=USINT 3=SINT 4=DINT 5=LINT 6=UINT 7=UDINT 8=ULINT 9=LREAL 100 = 厂商特定的
9	可选	设置(Set)	温度模式(Temp Mode)	BOOL(布尔型)	报告温度值时使用的温度单位	0=摄氏温度 1=华氏温度

4.17 模拟输出组对象(类代码=0x0F)

模拟输出组(AOG)对象应绑定在设备的一组模拟输出点上。绑定到组的所有点应共享包含在组里的所有属性。如果一个属性被多个模拟输出点(AOP)共享(这些点有同样的属性和同样的属性值),那么这个属性可能包含在一个模拟输出组里。一个模拟输出点可能被绑定到多个模拟输出组。

绑定到组的模拟输出点列表应在上电时建立,因为绑定列表是静态的,并且是组的 Get-only 属性。

模拟输出组对象可能被使用:

- 当在许多输出点共享一个单独属性时;
- 为了更有效地访问数据。

模拟输出组对象应支持表 16 中规定的实例属性。

表 16 模拟输出组对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取(Get)	属性数 (Number of Attributes)	USINT(无符号单整型)	组支持的属性数	—
2	可选	获取(Get)	属性列表 (Attribute ListAttribute List)	ARRAY of USINT(无符号单整型数组)	组支持的属性的列表	—
3	可选	获取(Get)	已绑定实例数 (Number of Bound Instances)	USINT(无符号单整型)	组内的绑定数量	—
4	可选	获取(Get)	绑定 (Binding)	ARRAY of UINT(无符号整型数组)	绑定到该组的所有点的列表实例 ID	—
5	可选	获取(Get)	状态 (Status)	BOOL(布尔型)	无报警或错误的组操作	0=OK 1=报警状态
6	可选	获取(Get)	所有者厂商 ID (Owner Vendor ID)	UINT(无符号整型)	组所有者的厂商 ID	—
7	可选	获取(Get)	所有者序列号 (Owner Serial Number)	UDINT(无符号双整型)	32 位组所有者序列号	—
8	可选	设置(Set)	值数据类型 (Value Data Type)	USINT(无符号单整型)	规定任何绑定值的数据类型	0=INT 1=REAL 2=USINT 3=SINT 4=DINT 5=LINT 6=UINT 7=UDINT 8=ULINT 9=LREAL 100=厂商特定的
9	必要	设置(Set)	命令 (Command)	BOOL(布尔型)	改变 AOP 状态到空闲模式或运行模式	0=空闲 1=运行
10	可选	设置(Set)	故障状态 (Fault State)	BOOL(布尔型)	可恢复故障的输出状态	0=故障值属性 1=保持最后状态
11	可选	设置(Set)	故障值 (Fault Value)	INT(整型)或基于属性 8	用于故障状态属性的用户定义值	—
12	可选	设置(Set)	空闲状态 (Idle State)	BOOL(布尔型)	空闲时的输出状态	0=空闲值属性 1=保持最后状态
13	可选	设置(Set)	空闲值 (Idle Value)	INT(整型)或基于属性 8	用于空闲状态属性的用户定义值	—

4.18 模拟组对象(类代码=0x10)

模拟组(AG)对象应绑定其他的模拟对象,包括 AIP、AOP、AIG 或 AOG。绑定到 AG 的对象(输入和输出)应支持同样的属性和服务。

绑定到 AG 的模拟组和/或点列表应称为绑定列表。绑定列表应在上电时建立。绑定列表是静态的,且是 AG 对象的 Get-only 属性。

- AG 对象可被用于:
- 当一个单独的属性在许多模拟输入和输出点和/或组内共享时;
 - 为了更有效地访问数据。

AG 对象应支持表 17 中规定的实例属性。

表 17 模拟组对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取(Get)	属性数(Number of Attributes)	USINT(无符号单整型)	组支持的属性数	—
2	可选	获取(Get)	属性列表(Attribute ListAttribute List)	ARRAY of USINT(无符号单整型数组)	被支持的属性的列表	—
3	可选	获取(Get)	已绑定实例数(Number of Bound Instances)	USINT(无符号单整型)	组内点数量	—
4	可选	获取(Get)	绑定(Binding)	ARRAY of STRUCT: UINT (结构体数组:无符号整型)	所有组内实例列表,类 ID,实例 ID	—
5	可选	获取(Get)	状态(Status)	BOOL(布尔型)	组无警报或错误下操作	0=OK 1=报警状态
6	可选	获取(Get)	所有者厂商 ID(Owner Vendor ID)	UINT(无符号整型)	组所有者的厂商 ID	—
7	可选	获取(Get)	所有者序列号(Owner Serial Number)	UDINT(无符号双整型)	32 位组所有者序列号	—
8	可选	设置(Set)	值的数据类型(Value Data Type)	USINT(无符号单整型)	决定所有绑定对象值的数据类型,如果组被绑定到 AG,则通过另一个绑定传播	0=INT 1=REAL 2=USINT 3=SINT 4=DINT 5=LINT 6=UINT 7=UDINT 8=ULINT 9=LREAL 10~99=保留 100 = 厂商特定的

4.19 位置传感器对象(类代码=0x11)

位置传感器对象(Position Sensor Object)将产品中的绝对位置传感器进行模型化。对象的行为应扩展基本位置传感器性能,以包括零偏移和位置边界检查(凸轮开关)。

位置传感器对象接口是为了实现实际传感器硬件,如绝对数字编码器、模拟旋转变压器或其他绝对位置输入设备。

位置传感器对象应支持表 18 中规定的实例属性。

表 18 位置传感器对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取 (Get)	属性数 (Number of Attributes)	USINT(无符号单整型)	产品中支持的属性数	—
2	可选	获取 (Get)	属性列表 (Attribute ListAttribute List)	ARRAY of USINT(无符号单整型数组)	产品中被支持的属性的列表	—
3	必要	获取 (Get)	值 (Value)	UDINT(无符号双整型)	当前位置	被分辨率和零偏移属性修改的物理位置
4	可选	获取 (Get)	凸轮 (CAM)	BOOL(布尔型)	虚拟凸轮开关值	0=off 1=on
5	可选	获取 (Get)/ 设置 (Set)	值比特分辨率 (Value Bit Resolution)	USINT(无符号单整型)	位置传感器分辨率。可能大于或小于物理位置传感器分辨率	可用的位置传感设备的分辨率。缺省应是位置传感硬件的实际分辨率。 设置此值为 7,限制值为 0~127(在应用 Zero Offset 以前)
6	可选	获取 (Get)/ 设置 (Set)	零偏移 (Zero Offset)	UDINT(无符号双整型)	属性零偏移的值,从属于分辨率 (Resolution)属性	这个属性应在应用分辨率属性后,偏移值属性。应有效地为值 (Value) 属性设置零点。 缺省值为 0
7	可选	获取 (Get)/ 设置 (Set)	凸轮下限 (CAM Low Limit)	UDINT(无符号双整型)	虚拟凸轮的开关下限	缺省值为 0
8	可选	获取 (Get)/ 设置 (Set)	凸轮上限 (CAM High Limit)	UDINT(无符号双整型)	虚拟凸轮的开关上限	缺省值为 0
9	可选	获取 (Get)/ 设置 (Set)	设置零 (SetZero)	BOOL(布尔型)	自动零控制	信号的上升沿设置零偏移 (Zero Offset) 为当前位置

4.20 位置控制器的监控对象(类代码=0x12)

4.20.1 监视(Supervisor)属性

位置控制器对象应支持表 19 中规定的实例属性。

表 19 位置控制器的监控对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	可选	获取(Get)	属性数 (Number of Attributes)	USINT(无符号单整型)	设备支持的属性总数	返回值在 0~255 的范围内
2	可选	获取(Get)	属性列表 (Attribute List)	ARRAY of USINT(无符号单整型数组)	返回设备支持的属性列表数组	数组大小由属性 1 定义
3	必要	获取(Get)	轴实例 (Axis Instance)	USINT(无符号单整型)	返回实例或轴数量。此属性由 I/O 集使用,来确定为 Position Controller Supervisor 或 Position Controller 属性设置或返回哪个实例数据	在范围 1~7 内 1=轴 1 2=轴 2
4	—	—	保留	—	—	—
5	必要	获取(Get)	常见故障 (General Fault)	BOOL(布尔型)	常见故障标志。此位应是设备中所有故障条件属性的逻辑或。当故障条件消除时,此位应重置	1=故障条件存在
6	必要	设置(Set)	输入命令集合类型 (Input Command Assembly Type)	USINT(无符号单整型)	设置由控制设备传送的命令集。轮询响应的后四个字节可能被此属性定义	有效集合类型代码为 1~31 1=集合 01 2=集合 02 等
7	必要	设置(Set)	输出应答集合类型 (Output Response Assembly Type)	USINT(无符号单整型)	设置返回到控制设备的响应集。轮询响应的后四个字节可能被此属性定义	有效集合类型代码为 1~31 1=集合 01 2=集合 02
8	可选	获取(Get)	故障输入 (Fault Input)	BOOL(布尔型)	故障输入标志	1=故障输入起作用
9	可选	设置(Set)	故障输入动作 (Fault Input Action)	BOOL(布尔型)	当故障输入成为活动代码时采取的动作	0=伺服系统关闭 1=硬停机 2=平滑停机 3=无动作 动作代码 128~255 供特定厂商的动作使用

4.20.2 原点(Home)和指针(Index)属性

位置控制器对象应支持表 20 中规定的原点(Home)和指针(Index)属性。

表 20 位置控制器的监控对象原点(Home)和指针(Index)属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
10	可选	设置(Set)	返回原点动作 (Home Action)	USINT(无符号单整型)	返回原点输入动作代码。当已待命原点输入被触发时采取动作	0=伺服系统关闭 1=硬停机 2=平滑停机 3=无动作 4=门分度 动作代码 128~255 供特定厂商的动作使用
11	可选	设置(Set)	返回原点动作等级 (Home Active level)	BOOL(布尔型)	返回原点有效级标志用于返回原点输入有效级编程	0=有效级低 1=有效级高
12	可选	获取(Get)/ 设置(Set)	返回原点待命 (Home Arm)	BOOL(布尔型)	返回原点触发标志用于等待返回原点输入	1=待命返回原点输入 0=触发已发生
13	可选	设置(Set)	分度动作 (Index Action)	USINT(无符号单整型)	分度输入动作代码	0=伺服系统关闭 1=硬停机 2=平滑停机 3=无动作 动作代码 128~255 供特定厂商的动作使用
14	可选	获取(Get)/ 设置(Set)	分度动作等级 (Index Active Level)	BOOL(布尔型)	用于对分度输入动作级编程	0=有效级低 1=有效级高
15	可选	获取(Get)/ 设置(Set)	分度待命 (Index Arm)	BOOL(布尔型)	分度触发待命标志用于等待分度输入	1=待命分度输入 0=触发已发生
16	可选	获取(Get)	返回原点输入等级 (Home Input Level)	BOOL(布尔型)	返回原点输入的实际等级	0=返回原点输入级低 1=返回原点输入级高
17	可选	获取(Get)	原 点 位 置 (Home Posi- tion)	DINT(双整型)	原点触发位置反映在返回原点输入被触发时的位置	值在范围 0x80000001~ 0x7FFFFFFF
18	可选	获取(Get)	分度位置 (Index Position)	DINT(双整型)	分度触发位置反映在分度输入被触发时的位置	值在范围 0x80000001~ 0x7FFFFFFF

4.20.3 寄存器(Registration)属性

位置控制器的监控对象应支持表 21 中规定的寄存器(Registration)属性。

表 21 位置控制器的监控对象(Registration)属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
19	可选	设置(Set)	对准定位动作 (Registration Action)	USINT (无符号单整型)	对准定位输入动作应定义当对准定位输入被触发时所发生的动作	0=伺服系统关闭 1=硬停机 2=平滑停机 3=无动作 4=转对准定位原点补偿值 5=转对准定位绝对位置量 动作代码 128 ~ 255 供特定厂商的动作使用
20	可选	设置(Set)	对准定位动作等级 (Registration Active level)	BOOL (布尔型)	对准定位触发动作等级标志用于对准定位输入动作等级编程	0=动作级低 1=动作级高
21	可选	获取(Get)/设置(Set)	对准定位 (Registration Arm)	BOOL (布尔型)	对准定位触发待命标志用于等待对准定位输入	1 = 待命对准定位输入 0 = 已发生对准定位触发
22	可选	获取(Get)	对准定位输入等级 (Registration Input Level)	BOOL (布尔型)	对准定位输入的动作等级	0 = 对准定位动作级低 1 = 对准定位动作级高
23	可选	设置(Set)	对准定位原点补偿 (Registration Offset)	DINT (双整型)	根据对准定位动作代码定义一个位置值作为补偿位置或绝对位置	值在范围 0x80000001 ~ 0x7FFFFFFF
24	可选	获取(Get)	对准定位位置 (Registration Position)	DINT (双整型)	对准定位触发位置应反映对准定位输入被触发时的位置	值在范围 0x80000001 ~ 0x7FFFFFFF

4.20.4 轴跟随(Axis Following)属性

位置控制器的监控对象应支持表 22 中的轴跟随(Axis Following)属性。

表 22 位置控制器的监控对象轴跟随(Axis Following)属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
25	可选	设置 (Set)	随动启用 (Follow Enable)	BOOL(布尔型)	启用使轴随动起作用	0=跟踪不起作用 1=跟踪生效
26	可选	设置 (Set)	随动轴 (Follow Axis)	USINT(无符号单整型)	规定随动的轴	0=无随动 1~255 指定参与随动的轴数
27	可选	设置 (Set)	随动除法因子 (Follow Divisor)	DINT(双整型)	用此值除以随动轴的位置计算定位命令位置	—
28	可选	设置 (Set)	随动乘法因子 (Follow Multiplier)	DINT(双整型)	用此值乘以跟踪轴位置计算定位命令位置	—

4.21 位置控制器对象(类代码=0x13)

位置控制器对象应执行控制器输出速度配置的控制,并处理向电动机驱动器单元、限位开关等的输出和从电动机驱动器单元、限位开关的输入。

位置控制器对象应支持表 23 中规定的 Profile 属性。

表 23 位置控制器对象 Profile 属性

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
1	必要	获取 (Get)	属性数(Number of Attributes)	USINT(无符号单整型)	返回设备支持的属性数	返回值在范围 0~255
2	必要	获取 (Get)	属性列表 (Attribute List)	UDINT(无符号双整型)	返回设备支持的属性列表数组	属性 1 定义的数组大小
3	可选	设置 (Set)	模式 (Mode)	UDINT(无符号双整型)	运行模式	0=位置模式(缺省值) 1=速度模式 2=力矩模式
4	可选	设置 (Set)	定位单位 (Position Units)	DINT(双整型)	定位单位比率值是一个定位单位相当于实际使用编码器的计数脉冲数	此值仅可设置为正数。 缺省值=1
5	可选	设置 (Set)	运动单位 (Profile Units)	DINT(双整型)	运动单位比率值是一个速度、加速或减速单位,它相当于每秒或每二次方秒实际使用编码器的计数脉冲数	此值仅可设置为正数。 缺省值=1
6	必要	设置 (Set)	目标位置 (Target Position)	DINT(双整型)	所定义的运动目标位置,单位为定位单位	设置范围为 0x80000001~ 0x7FFFFFFF

表 23 (续)

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
7	必要	设置 (Set)	目标速度 (TargetVelocity)	DINT(双整型)	所定义的目标运动速度,单位为每秒运动单位	此值仅可设置为正数
8	必要	设置 (Set)	加速 (Acceleration)	DINT(双整型)	运动加速度应以每二次方秒多少个运动单位定义	此值仅可设置为正数
9	可选	设置 (Set)	减速 (Deceleration)	DINT(双整型)	运动减速度应以每二次方秒多少个运动单位定义	此值仅可设置为正数
10	可选	设置 (Set)	增量位置标志 (Incremental Position Flag)	BOOL(布尔型)	增量位置或相对位置标志	如果设为 0,则目标位置(属性 6)应解释为绝对位置。如果设为 1,则目标位置应解释为增量或相对位置
11	必要	获取 (Get)/ 设置 (Set)	轨迹开始/结束 (Trajectory Start/Complete)	BOOL(布尔型)	运动轨迹开始和结束标志	设置运动轨迹的起点。当运动结束时,清除读数
12	可选	获取 (Get)	目标位置在位 (On Target Position)	BOOL(布尔型)	目标位置在位标志表示电动机处于目标位置的不动作距离带(死区)内	当实际位置等于(目标位置±死区)时,读设置。如果目标位置改变,则清除
13	可选	获取 (Get)/ 设置 (Set)	实际位置 (Actual Position)	DINT(双整型)	实际绝对位置值应等于其单位为定位单位的实际位置。设置为再定义实际位置	当设置时,此值范围为:0x80000001~0x7FFFFFFF
14	可选	获取 (Get)	实际速度 (Actual Velocity)	DINT(双整型)	实际速度的单位为每秒运动单位	恒为正数
15	可选	获取 (Get)	已命令位置 (Commanded Position)	DINT(双整型)	此值应等于位置的瞬时计算值	当被读时,此值可能范围为 0x80000001~0x7FFFFFFF
16	可选	获取 (Get)	已命令速度 (Commanded Velocity)	DINT(双整型)	此值应等于速度的瞬时计算值,其单位为每秒运动单位	值的读数应为正数值
17	可选	设置 (Set)	允许 (Enable)	BOOL(布尔型)	允许输出	置 1: 允许驱动和反馈。 置 0(清除):禁止

表 23 (续)

属性 ID	实现必要性	访问规则	名称	数据类型	属性描述	值语义
18	可选	设置 (Set)	运动类型 (Type)	USINT(无符号单整型)	运动类型代码定义运动轨迹的类型	0=梯形 1=S-曲线 2=抛物线 128~255=厂商指定的运动类型
19	可选	设置 (Set)	运动增益 (Gain)	DINT(双整型)	此属性给出非梯形运动轨迹(如 S-曲线)的增益值。此增益值的实现和功能是厂商专用的	范围由厂商定义
20	可选	设置 (Set)	平滑停车 (Smooth Stop)	BOOL(布尔型)	电动机平滑停车	设置电动机立即以可编程的减速速率强制减速至零速
21	可选	设置 (Set)	硬停车 (Hard Stop)	BOOL(布尔型)	电动机硬停车	设置电动机立即以最大减速速率强制减速至零速
22	可选	设置 (Set)	点动速度 (Jog Velocity)	DINT(双整型)	定义点动速度, 单位:每秒运动单位	此值仅被设置为正数
23	可选	获取 (Get)/ 设置 (Set)	方向 (Direction)	BOOL(布尔型)	瞬时方向	0=反方向 1=正方向 在速度模式下,设置此值来改变方向
24	可选	设置 (Set)	参考方向 (Reference Direction)	BOOL(布尔型)	定义正方向	0=正方向为顺时针 1=反方向为逆时针
25	可选	获取 (Get)/ 设置 (Set)	力矩 (Torque)	DINT(双整型)	输出力矩	设置此值来改变力矩(仅在力矩模式下),或读出当前力矩值。 0=无力矩输出 范围由厂商定义
26	可选	设置 (Set)	正力矩限制 (Positive Torque Limit)	DINT(双整型)	此值设置正方向上最大允许力矩输出	此值应仅被设置为正数。范围由厂商定义
27	可选	设置 (Set)	负力矩限制 (Negative Torque Limit)	DINT(双整型)	此值设置反方向的最大允许力矩输出	此值仅被设置为正数。范围由厂商定义
28	—	—	保留	—	—	—

4.22 电机数据对象（类代码=0x20）

本对象提供电机参数数据库的服务。

电机数据属性应支持表 24 中规定的类属性。

电机数据实例属性将随着对象所表示的电机类型不同而变化。因此实例属性 3“电机类型”是必需的。它的值将决定适用于该电机类型的电机专用属性。对所有的电机类型,属性 1~5 是相同的。

表 24 电机数据对象实例属性

属性 ID	实现 必要性	访问规则	名称	数据类型	值语义
1	可选	Get	属性编号 (NumAttr)	USINT(无符号单整型)	所支持的属性编号
2	可选	Get	属性 (Attributes)	Array of USINT (无符号单整型数组)	所支持的属性表
3	必要	Set/Get	电机类型 (MotorType)	USINT(无符号单整型)	0=非标准电机 1=PM DC 电机 2=FC DC 电机 3=PM 同步电机 4=FC 同步电机 5=开关磁阻电机 6=绕线转子感应电机 7=鼠笼式感应电机 8=步进电机 9=正弦 PM BL 电机 10=梯形 PM BL 电机
4	可选	Set/Get	目录号 (CatNumber)	SHORT-STRING (短字符串)	制造商电机目录号(铭牌号) 最大 32 个字符
5	可选	Set/Get	制造商 (Manufacture)	SHORT-STRING (短字符串)	制造商名称 最大 32 个字符

4.23 控制监视器对象(类代码=0x14)

本对象是将电机控制设备系列中各种设备的管理功能模型化。电机控制设备的行为在表 25 中被描述。控制监视器对象应支持表 25 所规定的实例属性。

表 25 控制监督对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	值语义
1	可选	Get	属性编号 (NumAttr)	USINT(无符号单整型)	所支持的属性编号
2	可选	Get	属性 (Attributes)	USINT(无符号单整型)阵列	所支持的属性列表

表 25 (续)

属性 ID	实现必要性	访问规则	名称	数据类型	值语义
3	必要	Set/Get	Run1	BOOL(布尔型)	—
4	可备	Set/Get	Run2	BOOL(布尔型)	—
5	可选	Set/Get	NetCtrl	BOOL(布尔型)	就地或者从网络请求 Run/Stop 控制 0=就地控制 1=网络控制 Run/Stop 控制的实际状态应反映在属性 15 CtrlFromNet 中
6	可选	Get	状态 (State)	USINT (无符号单整型)	0=制造商特定 1=启动 2=未就绪 3=就绪 4=使能 5=停止 6=出错停止 7=出错
7	仅对驱动器和伺服电机是必需的	Get	Running1	BOOL(布尔型)	1=使能与 Run1 或者 停止与 Running1 或者 出错停止与 Running1 0=其他状态
8	可选	Get	Running2	BOOL(布尔型)	1=允许与 Run2 或者 停止与 Running2 或者 出错停止与 Running2 0=其他状态
9	可选	Get	就绪 (Ready)	BOOL(布尔型)	1=就绪或允许或停止 0=其他状态
10	仅对驱动器和伺服电机是必需的	Get	出错 (Faulted)	BOOL(布尔型)	1=发生出错(闭锁) 0=不存在出错
11	可选	Get	警告 (Warning)	BOOL(布尔型)	1=警告(未闭锁) 0=不存在警告 如果警告不被支持,该属性通常为 0
12	仅对驱动器和伺服电机是必需的	Set/Get	出错加复位 (FaultRst)	BOOL(布尔型)	0->1=出错·复位 0=无动作

表 25 (续)

属性 ID	实现必要性	访问规则	名称	数据类型	值语义
13	可选	Get	出错代码 (FaultCode)	UINT (无符号整型)	如果处在出错(Faulted) 状态, 出错代码将指出导致转变到出错状态的故障。 如果发生多个故障, 供应商将选择报告其中一个出错, 其他的将丢失。 如果不是处在出错(Faulted) 状态, 出错代码指的将是导致转变到出错状态的最后一个故障。 出错代码的上电状态由供应商规定。 驱动器的出错代码与启动器的出错代码是不同的, 参照相应的设备规范中出错代码的内容
14	可选	Get	报警代码 (WarnCode)	UINT (无符号整型)	代码字指出存在报警。 如果存在多个报警, 显示的将是最小的代码值。 注意驱动器和伺服电机的出错代码与启动器的出错代码是不同的, 参照相应的设备规范中出错代码的内容
15	可选	Get	允许从网络控制 (CtrlFromNet)	BOOL(布尔型)	Run/Stop 控制源的状态: 0=就地控制 1=允许从网络控制
16	可选	Set/Get	DN 故障 (DNFaultMode)	USINT (无符号单整型)	链接丢失的行动: 0=出错+停止 1=忽略(报警可选) 2=供应商特定
17	可选	Set/Get	强制出错/脱扣 (ForceFault/ Trip)	BOOL(布尔型)	0->1=强制
18	可选	Get	强制状态 (ForceStatus)	BOOL(布尔型)	0=非强制 非 0 值=强制

4.24 交流/直流驱动器对象(类代码=0x15)

交流/直流驱动器对象应对交流或直流驱动器特定的功能(如速度斜率, 力矩控制等)模型化。
交流/直流驱动器对象应支持表 26 所规定的实例属性。

表 26 交流/直流驱动对象实例属性

属性 ID	实现必要性	访问规则	名称	数据类型	值语义
1	可选	Get	属性数量 (NumAttr)	USINT(无符号单整型)	所支持的属性数量
2	可选	Get	属性 (Attributes)	Array of USINT(无符号单整型数组)	所支持的属性列表
3	可选	Get	在基准值 (AtReference)	BOOL(布尔型)	1=驱动器实际值在该状态下的基准值上 (速度或者力矩基准值)
4	必要	Set/Get	网络基准 (NetRef)	BOOL(布尔型)	请求就地或者通过网络的力矩或者速度的基准。 0=非 DN 控制设定基准值 1=DN 控制设定基准值
5	可选	Set/Get	网络进程 (NetProc)	BOOL(布尔型)	请求本地或者来自网络的过程控制基准值。 0=在本地控制下设定值 1=在网络控制下设定基准值
6	必要	Set/Get	驱动器模式 (DriveMode)	USINT(无符号单整型)	0=供应商规定的模式 1=开环速度(频率) 2=闭环速度控制 3=力矩控制 4=过程控制(即 PI) 5=位置控制
7	必要	Get	实际速度 (SpeedActual)	INT(整型)	实际驱动器速度(最佳近似)单位: RPM/2 ^{SpeedScale} 其中速度(SpeedScale)即属性 22
8	必要	Set/Get	速度基准值 (SpeedRef)	INT(整型)	速度基准值单位:RPM/2 ^{SpeedScale} 其中速度标度(SpeedScale)即属性 22
9	可选	Get	实际电流 (CurrentActual)	INT(整型)	实际电机相电流单位:100 mA/2 ^{CurrentScale} 其中电流标度 CurrentScale 即属性 23
10	可选	Set/Get	电流限幅 (CurrentLimit)	INT(整型)	电机相电流限幅单位:100 mA/2 ^{CurrentScale} 其中电流标度(CurrentScale)即属性 23
11	可选	Get	实际力矩 (TorqueActual)	INT(整型)	实际力矩单位: Nm/2 ^{TorqueScale} 其中力矩标度(TorqueScale)即属性 24
12	可选	Set/Get	力矩基准值 (TorqueRef)	INT(整型)	力矩基准值单位: Nm/2 ^{TorqueScale} 其中力矩标度(TorqueScale)即属性 24
13	可选	Get	过程实际值 (ProcessActual)	INT(整型)	实际过程控制值: %/2 ^{ProcessScale} 其中过程标度(ProcessScale)即属性 25

表 26 (续)

属性 ID	实现必要性	访问规则	名称	数据类型	值语义
14	可选	Set/Get	过程基准值 (ProcessRef)	INT(整型)	过程控制基准值： $\%/2^{\text{ProcessScale}}$ 其中过程标度(ProcessScale)即属性 25
15	可选	Get	实际功率 (PowerActual)	INT(整型)	实际输出功率单位： $\text{W}/2^{\text{PowerScale}}$ 其中功率标度(PowerScale)即属性 26
16	可选	Get	输入电压 (InputVoltage)	INT(整型)	输入电压单位： $\text{V}/2^{\text{VoltageScale}}$ 其中电压标度(VoltageScale)即属性 27
17	可选	Get	输出电压 (OutputVoltage)	INT(整型)	输出电压单位： $\text{V}/2^{\text{VoltageScale}}$ 其中电压标度(VoltageScale)即属性 27
18	可选	Set/Get	加速时间 (AccelTime)	UINT(无符号整型)	加速时间从 0 到速度高限的时间 (HighSpdLimit)单位： $\text{ms}/2^{\text{TimeScale}}$ 其中时间标度(TimeScale)即属性 28 负方向加速时间的选择由供应商规定
19	可选	Set/Get	减速时间 (DecelTime)	UINT(无符号整型)	减速时间从 0 到速度高限的时间 (HighSpdLimit)单位： $\text{ms}/2^{\text{TimeScale}}$ 其中时间标度(TimeScale)即属性 28 负方向减速时间选择由供应商规定
20	可选	Set/Get	低速限幅 (LowSpdLimit)	UINT(无符号整型)	低速限制单位： $\text{RPM}/2^{\text{SpdScale}}$ 其中速度标度(SpeedScale)即属性 22
21	可选	Set/Get	高速限幅 (HighSpdLimit)	UINT(无符号整型)	最大速度限制单位： $\text{RPM}/2^{\text{SpeedScale}}$ 其中速度标度(SpeedScale)即属性 22
22	可选 ^a	Set/Get	速度标度 (SpeedScale)	SINT(单整型)	速度标度因子标度过程如下： 标度的速度= $\text{RPM}/2^{\text{SpeedScale}}$ 缩放范围： $-128\sim127$
23	可选 ^a	Set/Get	电流标度 (CurrentScale)	SINT(单整型)	电流标度因子标度过程如下： 标度的电流= $\text{A}/2^{\text{CurrentScale}}$ 范围： $-128\sim127$
24	可选 ^a	Set/Get	力矩标度 (TorqueScale)	SINT(单整型)	力矩标度因子标度过程如下： 标度的力矩= $\text{Nm}/2^{\text{TorqueScale}}$ 范围： $-128\sim127$
25	可选 ^a	Set/Get	过程标度 (ProcessScale)	SINT(单整型)	过程标度因子标度过程如下： 标度的过程= $\%/2^{\text{ProcessScale}}$ 范围： $-128\sim127$
26	可选 ^a	Set/Get	功率标度 (PowerScale)	SINT(单整型)	功率标度因子标度过程如下： 标度的功率= $\text{W}/2^{\text{PowerScale}}$ 范围： $-128\sim127$

表 26 (续)

属性 ID	实现必要性	访问规则	名称	数据类型	值语义
27	可选 ^a	Set/Get	电压标度 (VoltageScale)	SINT(单整型)	电压标度因子标度过程如下： 标度的电压 = $V/2^{\text{VoltageScale}}$ 范围：-128~127
28	可选 ^a	Set/Get	时间标度 (TimeScale)	SINT(单整型)	时间标度因子标度过程如下： 标度的时间 = $\text{ms}/2^{\text{TimeScale}}$ 范围：-128~127
29	可选	Get	基准值来自网络 (RefFromNet)	BOOL(布尔型)	力矩/速度基准值的状态 0=本地力矩/速度基准值 1=来自链路的力矩/速度基准值
30	可选	Get	过程基准值来自网络 (ProcFromNet)	BOOL(布尔型)	过程控制基准值的状态 0=本地过程基准值 1=通过连接确定的过程基准值
31	可选	Set/Get	励磁电流或电压 (FieldIorV)	BOOL(布尔型)	为直流电机选择励磁电压或励磁电流控制 0=电压控制(开环) 1=电流控制(直流电机励磁)
32	可选	Set/Get	励磁电压比 (FieldVoltRatio)	UINT(无符号整型)	直流电机电压控制
33	可选	Set/Get	磁场电流设定点 (FieldCurSetPt)	UINT(无符号整型)	直流电机励磁电流设定值单位： $A/2^{\text{CurrentScale}}$ 其中电流标度(CurrentScale)即属性 23
34	可选	Set/Get	FieldWkEnable	BOOL(布尔型)	允许/禁止直流电机弱励 0=禁止(电流控制方式的直流电机) 1=使能
35	可选	Get	实际励磁电流 (FieldCurActual)	INT(整型)	直流电机实际励磁电流单位： $A/2^{\text{CurrentScale}}$ 其中电流标度(CurrentScale)即属性 23
36	可选	Set/Get	最小励磁电流 (FieldMinCur)	INT(整型)	直流电机最小励磁电流单位： $A/2^{\text{CurrentScale}}$ 其中电流标度(CurrentScale)即属性 23
^a 在交流/直流驱动器对象中使用的基本单位见表 27。					

表 27 交流/直流驱动器对象基本量的名称和单位符号

量的名称	单位符号
转速	r/min
电流	A
力矩	Nm
过程控制	%
功率	W
电压	V
时间	s

5 抽象语法规范

5.1 传输格式规范

开放系统架构中位于底部的各层,与分散的各功能单元之间用户数据的传输密切相关。在这些层中,用户数据被简单地看作为八位字节的序列。然而,应用层实体要求操作的是具有相当复杂程度的数据类型的值。为了实现应用层和底部各层之间相互独立,数据类型需要以抽象语法符号的形式加以规范。

运用一个或更多的算法(被称为编码规则)对抽象语法的补充,可以决定承载应用层数据值的底层八位字节的值。抽象语法与一组传输规则的结合将产生一个具体的传输语法。

注:本部分的用户需要阅读并理解 GB/T 16262—2006、GB/T 16263—2006、GB/T 15969.3 和本部分第 5 章。阅读本部分时,请参照这些文档。

5.2 抽象语法符号

本部分中的数据类型定义应按照 GB/T 16262—2006 抽象语法符号 1(ASN.1)的格式书写。这些类型定义可作为 ASN.1 中模块“网络数据类型”的一部分。所以,为了指示这些定义包含于此模块中,ASN.1 声明的开始部分应写为:

IOT network DataTypes DEFINITIONS ::= BEGIN

ASN.1 声明的结束是关键字“END”。

随后的抽象定义将由控制网络数据类型的集合组成。此外,为了基于已有的类型扩展或派生新的数据类型,需作出相应的定义规定,并包含在“类型词典”中。

5.3 控制网络数据规范

符号[typeId]用于直接派生、枚举、子范围和结构化位串数据类型,它意味着标记应从 VariableDictionaryEntry 条目下的“type”域中取得。

Network Data ::= CHOICE{ElementaryData, DerivedData}
ElementaryData ::= CHOICE{
 BOOL,
 FixedLengthInteger,
 FixedLengthReal,

```

        AnyTime,
        AnyDate,
        AnyString,
        FixedLengthBitString}
DerivedData ::= CHOICE {
    DirectlyDerivedData,
    EnumeratedData,
    SubrangeData,
    StructuredBitStringData,
    ARRAY,
    STRUCT,
    FunctionBlockData}
DirectlyDerivedData ::= [typeId] NetworkData
EnumeratedData ::= [typeId] USINT
SubrangeData ::= [typeId] FixedLengthInteger
StructuredBitStringData ::= [typeId] FixedLengthBitString
FixedLengthInteger ::= CHOICE {SignedInteger, UnsignedInteger}
SignedInteger ::= CHOICE {SINT, INT, DINT, LINT}
UnsignedInteger ::= CHOICE {USINT, UINT, UDINT, ULINT}
FixedLengthReal ::= CHOICE {REAL, LREAL}
AnyTime ::= CHOICE {ITIME, TIME, FTIME, LTIME}
AnyDate ::= CHOICE {DATE, TIME_OF_DAY, DATE_AND_TIME}
AnyString ::= CHOICE {STRING, STRING2}
FixedLengthBitString ::= CHOICE {BYTE, WORD, DWORD, LWORD}
BOOL ::= [PRIVATE 1] IMPLICIT BOOLEAN
SINT ::= [PRIVATE 2] IMPLICIT OCTET STRING - 1 octet
INT ::= [PRIVATE 3] IMPLICIT OCTET STRING - 2 octets
DINT ::= [PRIVATE 4] IMPLICIT OCTET STRING - 4 octets
LINT ::= [PRIVATE 5] IMPLICIT OCTET STRING - 8 octets
USINT ::= [PRIVATE 6] IMPLICIT OCTET STRING - 1 octet
UINT ::= [PRIVATE 7] IMPLICIT OCTET STRING - 2 octets
UDINT ::= [PRIVATE 8] IMPLICIT OCTET STRING - 4 octets
ULINT ::= [PRIVATE 9] IMPLICIT OCTET STRING - 8 octets
REAL ::= [PRIVATE 10] IMPLICIT OCTET STRING - 4 octets
LREAL ::= [PRIVATE 11] IMPLICIT OCTET STRING - 8 octets
STIME ::= [PRIVATE 12] IMPLICIT DINT
DATE ::= [PRIVATE 13] IMPLICIT UINT
TIME_OF_DAY ::= [PRIVATE 14] IMPLICIT UDINT
DATE_AND_TIME ::= [PRIVATE 15] IMPLICIT SEQUENCE {
    time_of_day UDINT,
    date UINT }

```

STRING ::= [PRIVATE 16] IMPLICIT SEQUENCE {
 charcount UINT,
 stringcontents OCTET STRING} -one octet per character
 BYTE ::= [PRIVATE 17] IMPLICIT OCTET STRING - 1 octet
 WORD ::= [PRIVATE 18] IMPLICIT OCTET STRING - 2 octets
 DWORD ::= [PRIVATE 19] IMPLICIT OCTET STRING - 4 octets
 LWORD ::= [PRIVATE 20] IMPLICIT OCTET STRING - 8 octets
 STRING2 ::= [PRIVATE 21] IMPLICIT SEQUENCE {
 charcount UINT,
 string2contents OCTET STRING} - 2 octets/ character
 FTIME ::= [PRIVATE 22] IMPLICIT DINT
 LTIME ::= [PRIVATE 23] IMPLICIT LINT
 ITIME ::= [PRIVATE 24] IMPLICIT INT
 STRINGN ::= [PRIVATE 25] IMPLICIT SEQUENCE {
 charsize UINT,
 charcount UINT,
 stringNcontents OCTET STRING} - N octets/ character
 SHORT_STRING ::= [PRIVATE 26] IMPLICIT SEQUENCE {
 charcount USINT,
 stringcontents OCTET STRING} -one octet per character
 ARRAY ::= SEQUENCE OF NetworkData - All of same base type
 STRUCT ::= SEQUENCE OF NetworkData - May be different types
 FunctionBlockData ::= SET {
 inputs [0] IMPLICIT STRUCT OPTIONAL,
 outputs [1] IMPLICIT STRUCT OPTIONAL,
 controlInputs [2] IMPLICIT STRUCT OPTIONAL,
 controlOutputs [3] IMPLICIT STRUCT OPTIONAL}

5.4 数据类型规范/词典

对象的定义可以包括定义属性的文字。属性都应具有对象规范中的数据类型。属性的数据类型可能是包含于本部分中定义的数据类型,或可能是对于本部分的对象特定的扩展。下面的定义为数据提供了类型规范,同时也提供了基于已有类型的扩展或派生一个新的数据类型结构。

Dictionary ::= CHOICE {VariableDictionary, TypeDictionary}
 VariableDictionary ::= SEQUENCE OF VariableDictionaryEntry
 VariableDictionaryEntry ::= SEQUENCE {
 name AnyString,
 id FixedLengthInteger,
 type TypeID,
 ranges SEQUENCE OF Subrange, - for arrays
 accessPrivilege BOOL {READ_ONLY(0), READ_WRITE(1)}

TypeID ::= OCTET STRING -- ASN.1 encoded tag value of the
-- DataTypeSpecification module

Subrange ::= SEQUENCE {
 minValue FixedLengthInteger,
 maxValue FixedLengthInteger}

TypeDictionary ::= SEQUENCE OF TypeDictionaryEntry

TypeDictionaryEntry ::= SEQUENCE {
 name AnyString,
 type TypeID,
 spec DataTypeSpecification}

DataTypeSpecification ::= CHOICE {
 alt [PRIVATE 0] IMPLICIT AlternateTypeSpec,
 bool [PRIVATE 1] IMPLICIT NULL, -- BOOL
 sint [PRIVATE 2] IMPLICIT NULL, -- SINT
 int [PRIVATE 3] IMPLICIT NULL, -- INT
 dint [PRIVATE 4] IMPLICIT NULL, -- DINT
 lint [PRIVATE 5] IMPLICIT NULL, -- LINT
 usint [PRIVATE 6] IMPLICIT NULL, -- USINT
 uint [PRIVATE 7] IMPLICIT NULL, -- UINT
 udint [PRIVATE 8] IMPLICIT NULL, -- UDINT
 ulint [PRIVATE 9] IMPLICIT NULL, -- ULINT
 real [PRIVATE 10] IMPLICIT NULL, -- REAL
 lreal [PRIVATE 11] IMPLICIT NULL, -- LREAL
 stime [PRIVATE 12] IMPLICIT NULL, -- STIME
 date [PRIVATE 13] IMPLICIT NULL, -- DATE
 tod [PRIVATE 14] IMPLICIT NULL, -- TIME_OF_DAY
 dat [PRIVATE 15] IMPLICIT NULL, -- DATE_AND_TIME
 str1 [PRIVATE 16] IMPLICIT NULL, -- STRING
 byte [PRIVATE 17] IMPLICIT NULL, -- BYTE
 word [PRIVATE 18] IMPLICIT NULL, -- WORD
 dword [PRIVATE 19] IMPLICIT NULL, -- DWORD
 lword [PRIVATE 20] IMPLICIT NULL, -- LWORD
 str2 [PRIVATE 21] IMPLICIT NULL, -- STRING2
 ftime [PRIVATE 22] IMPLICIT NULL, -- FTIME
 ltime [PRIVATE 23] IMPLICIT NULL, -- LTIME
 itime [PRIVATE 24] IMPLICIT NULL, -- ITIME
 strN [PRIVATE 25] IMPLICIT NULL, -- STRINGN
 shstr [PRIVATE 26] IMPLICIT NULL, -- SHORT_STRING

```
constructedData CHOICE {
    abbrevStruc    [0]      IMPLICIT AbbreviatedStrucTypeSpec,
    abbrevArr      [1]      IMPLICIT AbbreviatedArrayTypeSpec,
    frmlStruc      [2]      IMPLICIT FormalStrucTypeSpec,
    frmlArr        [3]      IMPLICIT FormalArrayTypeSpec,
    expBitStr      [4]      IMPLICIT ExpandedFixedLenBitStrTypeSpec,
    expStr1        [5]      IMPLICIT ExpandedStringTypeSpec,
    expStr2        [6]      IMPLICIT ExpandedString2TypeSpec}
}
```

AbbreviatedStrucTypeSpec ::= UINT

AbbreviatedArrayTypeSpec ::= DataTypeSpecification

FormalStrucTypeSpec ::= SEQUENCE OF DataTypeSpecification

```
FormalArrayTypeSpec ::= SEQUENCE {
    lowBound    [0]    IMPLICIT FixedLengthInteger, -- rray Lower Bound
    highBound   [1]    IMPLICIT FixedLengthInteger, -- Array Upper Bound
    dataType    DataTypeSpecification }
```

```
ExpandedFixedLenBitStrTypeSpec ::= SEQUENCE {
    bitStrType    DataTypeSpecification -- BYTE, WORD, DWORD, or LWORD
    bitFields     [7]    IMPLICIT BitFieldDef}
```

```
BitFieldDef ::= SEQUENCE OF {
    bitDef [2] IMPLICIT OCTET STRING} -- Length is always 2 octets.
-- First octet contains starting
-- Bit Position. Trailing octet
-- contains the number of bits.
```

ExpandedStringTypeSpec ::= UINT -- String Length In Octets

ExpandedString2TypeSpec ::= UINT -- String Length In Octets

```
AlternateTypeSpec ::= CHOICE {
    directlyDerivedTypeSpec [0]    IMPLICIT    TypeID,
    subrangeTypeSpec        [1]    IMPLICIT    SubrangeTypeSpec ,
    enumeratedTypeSpec       [2]    IMPLICIT    EnumeratedTypeSpec,
    fbTypeSpec               [3]    IMPLICIT    FBTypeSpec}
```

```
SubrangeTypeSpec ::= SEQUENCE{
    baseType    TypeID,                -- NOTE: minValue and maxValue
    minValue    FixedLengthInteger,    -- shall be within the range
    maxValue    FixedLengthInteger}    -- of baseType values
```

```
EnumeratedTypeSpec ::= SEQUENCE OF AnyString
```

```
BitNameDefintion ::= SEQUENCE {
    bitName AnyString,
    bitNumber USINT}
```

```
FBTypeSpec ::= SET{
    inputs          [0]    IMPLICIT FbtElementSpec    OPTIONAL,
    outputs         [1]    IMPLICIT FbtElementTypeSpec OPTIONAL,
    controlInputs   [2]    IMPLICIT FbtElementTypeSpec OPTIONAL,
    controlOutputs  [3]    IMPLICIT FbtElementTypeSpec OPTIONAL}
FbtElementTypeSpec ::= SEQUENCE OF ElementSpec
```

```
ElementSpec ::= SEQUENCE {
    name AnyString,
    ypespec ElementTypeSpec}
```

```
ElementTypeSpec ::= CHOICE {
    [0]    IMPLICIT    TypeID,
    [1]    IMPLICIT    SubrangeTypeSpec,
    [2]    IMPLICIT    EnumeratedTypeSpec,
    [3]    IMPLICIT    FormalArrayTypeSpec,
    [4]    IMPLICIT    ExpandedStringTypeSpec,
    [5]    IMPLICIT    ExpandedString2TypeSpec}
```

下面的 END 声明将结束 ASN.1 模块的定义。

END

6 应用程序传输语法:紧凑型编码

6.1 紧凑型编码

本章讲述了第 5 章数据类型值中定义的数据类型,在控制网络上被编码/传输时可采用的方法。传输语法由抽象的语法定义和一组特定的编码规则组成。对于应用程序用户数据,应单独定义一组编码规则(紧凑型编码),这就是紧凑型传输语法。

紧凑型编码规则应首先遵从 ASN.1 8825 中定义的编码规则。然后它可以采用最优化规则,从最外部的服务数据单元(SDU)开始对每一个连续打包服务数据单元(SDU)进行处理。紧凑型编码规则应定义一个更高效的编码机制以减少在设备间传输的信息量。

紧凑型编码值与 ASN.1 编码值产生的差异在于删除了描述信息类型和长度的域。ASN.1 编码值中的标签(TAG)和长度(LENGTH)元素不应在控制网络上被传送。而且,紧凑型编码规则指出其八位字节排序规则和 ASN.1 中所见是相反的。

根据下节列出的条件,可以运用通用规则将 ASN.1 编码值转化成紧凑型编码值。通用规则如下:

——去除标识符八位字节;

- 去除 ASN.1 规定的“标签”八位字节；
- 去除长度八位字节；
- 去除 ASN.1 规定的“长度”八位字节；
- 倒转多八位字节的字节顺序。

6.2 紧凑型编码的约束条件

- 使用紧凑型编码的变量值的表示可能会有以下限制：
- 变量类型应为定长并且没有条件域或选择域；
 - 已知变量的编码应用从该变量类型规范中给出的一个恒定八位字节码表示。

6.3 举例

6.3.1 布尔编码

布尔编码用单个八位字节来表示，见表 28。

表 28 BOOLEAN(布尔)编码

如果值为：	则：
FALSE(假)	字节中 bit 0=0 (‘00’H)
TRUE(真)	字节中 bit 0=1 (‘01’H)

一个 FALSE 布尔值可由表 29 表示。

表 29 BOOL(布尔)值紧凑型编码的例子

八位字节编号	第一位
布尔	00

6.3.2 符号整型编码

符号整型编码的表示见表 30。

表 30 SignedInteger(符号整型)值的编码

八位字节编号	第一位	第二位	第三位	第四位	第五位	第六位	第七位	第八位
SINT(单整型)	0LSB							
INT(整型)	0LSB	1LSB						
DINT(双整型)	0LSB	1LSB	2LSB	3LSB				
LINT(长整型)	0LSB	1LSB	2LSB	3LSB	4LSB	5LSB	6LSB	7LSB

表 31 举例说明了一个其值为 0x12345678 的双整型变量的编码。

表 31 SignedInteger(符号整型)值的紧凑型编码举例

八位字节编号	第一位	第二位	第三位	第四位
DINT(双整型)	78	56	34	12

6.3.3 无符号整型编码

无符号整型编码的表示见表 32。

表 32 UnsignedInteger(无符号整型)值的编码

八位字节编号	第一位	第二位	第三位	第四位	第五位	第六位	第七位	第八位
USINT(无符号单整型)	0LSB							
UINT(无符号整型)	0LSB	1LSB						
UDINT(无符号双整型)	0LSB	1LSB	2LSB	3LSB				
ULINT(无符号长整型)	0LSB	1LSB	2LSB	3LSB	4LSB	5LSB	6LSB	7LSB

表 33 举例说明了一个其值为 0xAABBCCDD 的无符号双整型变量的编码。

表 33 UnsignedInteger(无符号整型)的紧凑型编码举例

八位字节编号	第一位	第二位	第三位	第四位
UDINT (无符号双整型)	DD	CC	BB	AA

6.3.4 定长实数型编码

定长实数型编码的表示见表 34。

表 34 FixedLengthReal(定长实数)值的编码

八位字节编号	第一位	第二位	第三位	第四位	第五位	第六位	第七位	第八位
REAL(实数)	0LSB	1LSB	2LSB	3LSB				
LREAL(长实数)	0LSB	1LSB	2LSB	3LSB	4LSB	5LSB	6LSB	7LSB

表 35 举例说明了一个其值为 Float1=10.0 的实数变量的编码[此值采用了 GB/T 15969.3 中的符号。在 ASN.1 中此值={‘41200000’H},在 IEEE 中此值=(1.25×2³, 指数是 130 (以 127 为基), 小数是 25)]。

表 35 REAL(实数)值的紧凑型编码举例

八位字节编号	0LSB	1LSB	2LSB	3LSB
REAL(实数)	00	00	20	41

表 36 举例说明了一个其值为 Float2=−10.0 的长实数变量的编码[在 ASN.1 中此值={‘C059000000000000’H},在 IEEE 中此值=(1.562 5×2⁶, 指数是 1 029 (以 1 023 为基), 小数是 5 625)]。

表 36 LREAL(长实数)值的紧凑型编码举例

八位字节编号	0LSB	1LSB	2LSB	3LSB	4LSB	5LSB	6LSB	7LSB
LREAL(长实数)	00	00	00	00	00	00	59	C0

6.3.5 时间编码

时间编码的表示见表 37。

表 37 时间值的编码

八位字节编号	第一位	第二位	第三位	第四位	第五位	第六位	第七位	第八位
TIME(时间)	0LSB	1LSB	2LSB	3LSB				
DATE(日期)	0LSB	1LSB						
TIME_OF_DAY(一天中的时间)	0LSB	1LSB	2LSB	3LSB				
DATE_AND_TIME(日期和时间)	0LSB	1LSB	2LSB	3LSB	0LSB	1LSB		
	Time	Time	Time	Time	Date	Date		
FTIME[持续时间 (高精度)]	0LSB	1LSB	2LSB	3LSB				
LTIME[持续时间 (长)]	0LSB	1LSB	2LSB	3LSB	4LSB	5LSB	6LSB	7LSB

6.3.6 字符串编码

STRING、STRING2、STRINGN 和 SHORT_STRING 的紧凑型编码的例子如下。
注：根据国际通常的字符串要求,对用户提供的字符串数据,其首选字符串类型是 String2。
表 38 给出了一个 STRING 值的编码的通用例子。

表 38 STRING 值

字符串	内容 (字符计数)		内容 (string 内容)
STRING (字符串)	0LSB	1LSB	0LSB

表 39 给出了一个 STRING2 值的编码的通用例子。

表 39 STRING2 值

字符串	内容 (字符计数)		内容 (string2 内容)
STRING2 (字符串 2)	0LSB	1LSB	0LSB

表 40 给出了一个 STRINGN 值的编码的通用例子。

表 40 STRINGN 值

字符串	内容(字符大小)		内容(字符计数)		内容 (stringN 内容)	
STRINGN(字符串)	0LSB	1LSB	0LSB	1LSB	0LSB	NLSB

表 41 给出了一个 SHORT_STRING 值的编码的通用例子。

表 41 SHORT_STRING 值

字符串	内容 (字符计数)		内容 (short_string 内容)
SHORT_STRING(短字符串)	0LSB	1LSB	0LSB

表 42 举例说明了一个其内容为“Mill”的字符串变量的编码。所有字符串变量的编码例子都有介绍。字符编码在 ISO/IEC 10646 中有规定。用十六进制表示 8 位编码相当于{'4D696C6C'H}。

表 42 STRING 值的紧凑型编码举例

字符串	内容（字符计数）		内容（string 内容）			
STRING（字符串）	04	00	4D	69	6C	6C

表 43 将“Mill”作为 STRING2 类型编码。

表 43 STRING2 值的紧凑型编码举例

字符串	内容（字符计数）		内容（string2 内容）							
STRING2(字符串 2)	04	00	4D	00	69	00	6C	00	6C	00

表 44 将“Mill”作为 SHORT_STRING 类型编码。

表 44 SHORT_STRING 类型

字符串	内容（字符计数）	内容（short_string 内容）			
SHORT_STRING（短字符串）	04	4D	69	6C	6C

6.3.7 定长位串编码

下面给出了 BYTE、WORD、DWORD、LWORD 数据值的紧凑型编码的例子。图 1 举例说明了有关定长位串紧凑型编码的比特排列规则。

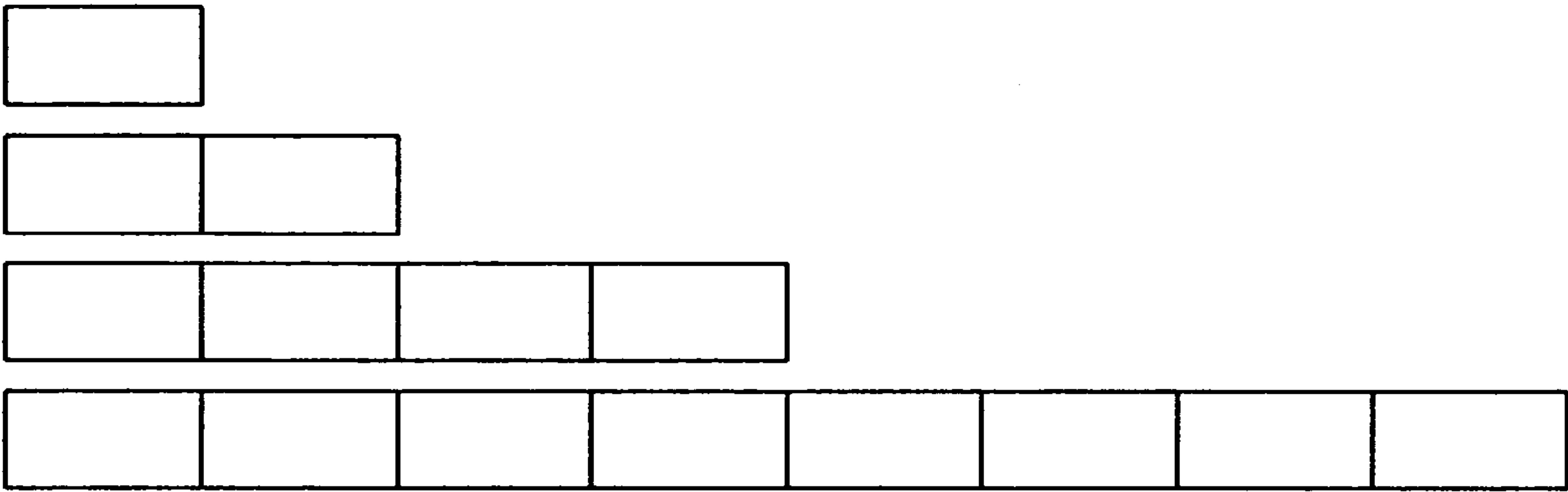


图 1 定长位串紧凑型编码的比特排列规则

图 2～图 5 举例说明了 BYTE、WORD、DWORD、LWORD 的编码。

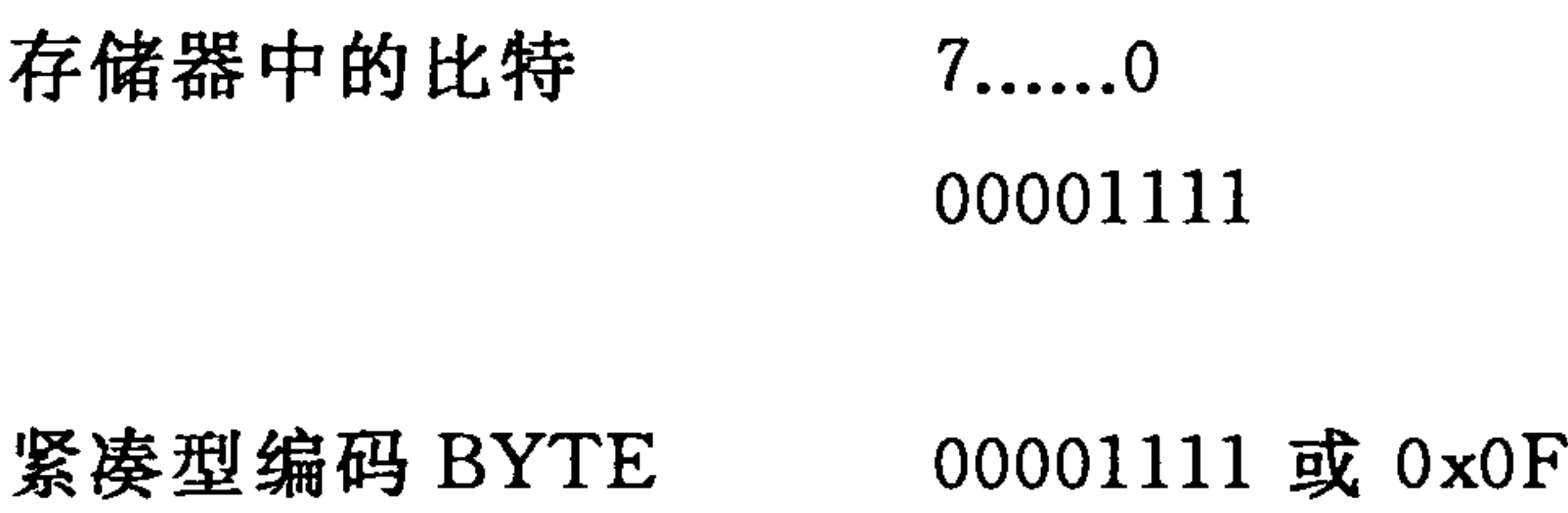


图 2 BYTE 定长字符串紧凑型编码的例子

存储器中的比特 15.....0
 00001111 11001111

紧凑型编码 WORD 11001111 00001111 或 0xCF0F

图 3 WORD 定长字符串紧凑型编码的例子

存储器中的比特 31.....0
 00001111 11001111 10110110 00111110

紧凑型编码 DWORD 00111110 10110110 11001111 00001111 或 0x3EB6CF0F

图 4 DWORD 定长字符串紧凑型编码的例子

存储器中的比特 63.....0
 11110000 11001111 10110110 00111110 11110000 00101101 00011110 00001111

紧凑型编码 LWORD 00001111 00011110 00101101 11110000 00111110 10110110 11001111 11110000
 或 0x0F1E2DF03EB6CFF0

图 5 LWORD 定长字符串紧凑型编码的例子

6.3.8 数组编码

6.3.8.1 一维数组编码

数组编码应采用对每个元素的数据类型的编码规则,并且连接组成数组的元素。数组元素的编码值应按照相应 ASN.1 类型或变量规范规定的相同顺序来编码。这些元素可以是任何数据类型。

对控制网络中一维数组的定义,ASN.1 的样式是:

ARRAY ::= SEQUENCE OF { array_dimension_low_bound,
 array_dimension_high_bound, NetworkData
}

假定如下数组定义::

ARRAY1 ::= SEQUENCE OF {array_dimension_low_bound := 0,
 array_dimension_high_bound := 1, UINT}

在这个数组定义中插入 UINT 值{1,2},就产生表 45 所示的编码。

表 45 一维 ARRAY(数组)紧凑型编码的例子

八位字节编号	第一位	第二位	第三位	第四位
ARRAY (数组)	01	00	02	00

6.3.8.2 二维数组编码

ARRAY ::= SEQUENCE OF { array_dimension_low_bound,
array_dimension_high_bound,

SEQUENCE OF { array_dimension_low_bound,
array_dimension_high_bound, NetworkData } }

6.3.8.3 三维数组编码

ARRAY ::= SEQUENCE OF { array_dimension_low_bound,
array_dimension_high_bound,
SEQUENCE OF { array_dimension_low_bound,
array_dimension_high_bound,
SEQUENCE OF { array_dimension_low_bound,
array_dimension_high_bound, NetworkData } } }

由于控制网络数据会包含基本数据或派生数据,所以在传送数组的值之前需要定义一种新的类型或变量规范。

多维数组在传输时应被视为一维数组。数组元素的次序通过端节点遵循的打包/拆包顺序予以保持。遵循该次序应可以访问数组第 N 维,也可以访问其他维的所有值。

这可以通过以下方法完成,首先将所有维度设定到它们的起始索引值来访问数组。此后第 N 维沿着其所有的索引值递增。当第 N 维度的索引值到达终点时,递增第(N-1)维度,同时第 N 维设定到起始索引值。这一过程不断重复,直到所有数组维度达到索引范围的终点。这样数组的结果就被打包成一维数组放到报文缓冲器中。采用同样的步骤可以将一维数组拆包成多维数组。

下面所示的数据流为二维数组按照紧凑型编码规则打包成一维数组时情形,产生表 46 所示的编码。

ARRAY1 [0..1 , 0..2] of UINT := { { 1, 2, 3 },
{ 4, 5, 6 } }

表 46 多维 ARRAY(数组)紧凑型编码的例子

八位字节编号	第一位	第二位	第三位	第四位	第五位	第六位	第七位	第八位	第九位	第十位	第十一位	第十二位
ARRAY (数组)	01	00	02	00	03	00	04	00	05	00	06	00

6.3.9 结构编码

结构编码应对每个元素采用数据类型的编码规则,连接这些元素以组成结构。

结构元素的编码值应按照相应 ASN.1 类型或变量规则规定的相同顺序来编码。这些元素可以是任何数据类型。

STRUCT ::= SEQUENCE OF NetworkData -可能是不同类型数据。

由于控制网络数据会包含基本数据或派升数据,所以在传送结构的值之前需要定义一种新的类型或变量规范。

假设如下结构定义:

newStruct ::= SEQUENCE { BOOL,UINT,DINT}

在这个结构中插入值 {TRUE,'1234'H,'56789ABC'H},就产生表 47 所示的编码。

表 47 STRUCTURE(结构)紧凑型编码的例子

八位字节编号	第一个	第二个	第三个	第四个	第五个	第六个	第七个
newStruct(新结构)	01	34	12	BC	9A	78	56

6.3.10 复杂编码数据格式

6.3.10.1 举例

下面的例子说明了更复杂的数据格式应怎样来打包。例 1 显示结构数组的打包。例 2 说明了带有数组元素的结构怎样来打包。

6.3.10.2 例 1: 结构中数组的编码

```
STRUCT1 ::= SEQUENCE OF {
    UINT          ele1;
    USINT         ele2;
    USINT         ele3;
    USINT         ele4;
    UINT          ele5
}

ARRAY1 [ 0..1 , 0..2 ] of STRUCT1 := {
    { { 1, 2, 3, 4, 5 }, { 6, 7, 8, 9, 10 },
      { 11, 12, 13, 14, 15 } },
    { { 15, 14, 13, 12, 11 }, { 10, 9, 8, 7, 6 },
      { 5, 4, 3, 2, 1 } } }
```

得出如下数据流:

```
[01][00][02][03][04][05][00] [06][00][07][08][09][0A][00]
[0B][00][0C][0D][0E][0F][00] [0F][00][0E][0D][0C][0B][00]
[0A][00][09][08][07][06][00] [05][00][04][03][02][01][00]
```

6.3.10.3 例 2: 带有数组元素的结构编码

```
STRUCT2 ::= SEQUENCE OF {
    UINT ele1;
    ARRAY [ 0..2 ] of USINT array2;
    UINT ele5;
}

STRUCT2 := { 1, { 2, 3, 4 }, 5 }
```

得出如下数据流:

```
[01][00] [02][03][04] [05][00]
```

7 数据类型报告

7.1 对象数据描述

为了报告一种特定属性的数据类型,或将实际数据和类型信息一起传送,对象可以选择执行一种机制。本章定义了传送数据类型信息的方法。

数据类型信息的规范按控制网络定义的优选法,利用 GB/T 16262—2006 和 GB/T 16263—2006 中规定的 ASN.1 的方法论,编码在本部分 5.4——数据类型规范/词典中规定的 DataTypeSpecification

(数据类型规范)生成。控制网络定义的最优化如下：

——NULL 类型的长度字节没有被编码。例如“abc [PRIVATE 1] IMPLICIT NULL”的编码就是 0xC1（一个没有长度字节的标签）。

7.2 基本数据类型报告

基本数据类型应用表 48 中定义的标识代码来标识。这些代码应定义出 DataTypeSpecification(数据类型规范)生产中原始成员的编码。控制网络明确 ASN.1 NULL 类型不报告成长度字节为 0。

表 48 基本数据类型的标识代码和描述

数据类型名称	数据类型代码（十六进制）	数据类型描述
BOOL(布尔型)	C1	逻辑量,值为 TRUE 和 FALSE
SINT(单整型)	C2	8 位符号整型值（单整型）
INT(整型)	C3	16 位符号整型值（整型）
DINT(双整型)	C4	32 位符号整型值（双整型）
LINT(长整型)	C5	64 位符号整型值（长整型）
USINT(无符号单整型)	C6	8 位无符号整型值（无符号短整型）
UINT(无符号整型)	C7	16 位无符号整型值（无符号整型）
UDINT(无符号双整型)	C8	32 位无符号整型值（无符号双整型）
ULINT(无符号长整型)	C9	64 位无符号整型值（无符号长整型）
REAL(实数)	CA	32 位浮点值（实数）
LREAL(长实数)	CB	64 位浮点值（长实数）
STIME(同步时间)	CC	同步时间
DATE(日期)	CD	日期
TIME_OF_DAY(一天中的时间)	CE	一天中的时间
DATE_AND_TIME(日期和时间)	CF	日期和时间
STRING 字符串	D0	字符串（每字符 1 字节）
BYTE 字节	D1	长度为 8 的位串
WORD(字)	D2	长度为 16 位的位串
DWORD(双字)	D3	长度为 32 位的位串
LWORD(长字)	D4	长度为 64 位的位串
STRING2(字符串 2)	D5	字符串（每字符 2 字节）
FTIME[持续时间（高精度）]	D6	持续时间（高精度）
LTIME[持续时间（长）]	D7	持续时间（长）
ITIME[持续时间（短）]	D8	持续时间（短）
STRINGN(字符串 N)	D9	字符串（每字符 N 字节）
SHORT_STRING(短字符串)	DA	字符串（每字符 1 字节, 1 字节长度指示符）

7.3 构建数据类型报告

7.3.1 结构类型定义

7.3.1.1 概述

控制网络定义了两种不同的方法,来报告结构类型定义:

- 正规编码 (FormalStrucTypeSpec);
- 缩略编码 (AbbreviatedStrucTypeSpec)。

正规编码用于提供整个结构定义的详细报告,包括所有组件数据类型的完整定义。缩略编码用于规定结构定义的一种简短形式。这种简短形式不包括和结构组件有关的数据类型。

7.3.1.2 结构类型信息的正规编码

下面 2 个例子说明了结构类型规范的正规编码。实际上这是一个在本部分 5.4——数据类型规范/词典中定义的 FormalStrucTypeSpec 生成的编码实例。

7.3.1.3 示例 1

表 49 举例说明了以下结构定义的编码。

STRUCT ::= SEQUENCE OF { BOOL, UINT, DINT }

表 49 结构类型规范正规编码的例子

结构类型	类型长度	组件类型	组件类型	组件类型
—	—	BOOL	UINT	DINT
A2	03	C1	C7	C4

注：零(0)长度字节不跟随 DataTypeSpecification(数据类型规范)生成中的 IMPLICIT NULL 类型。

7.3.1.4 示例 2

图 6 举例说明了以下结构定义的编码。

STRUCT_MAIN ::= SEQUENCE OF { UINT, STRUCT_SUB, INT }
with subelement STRUCT_SUB defined as:
STRUCT_SUB ::= SEQUENCE OF { UINT, SINT, INT }

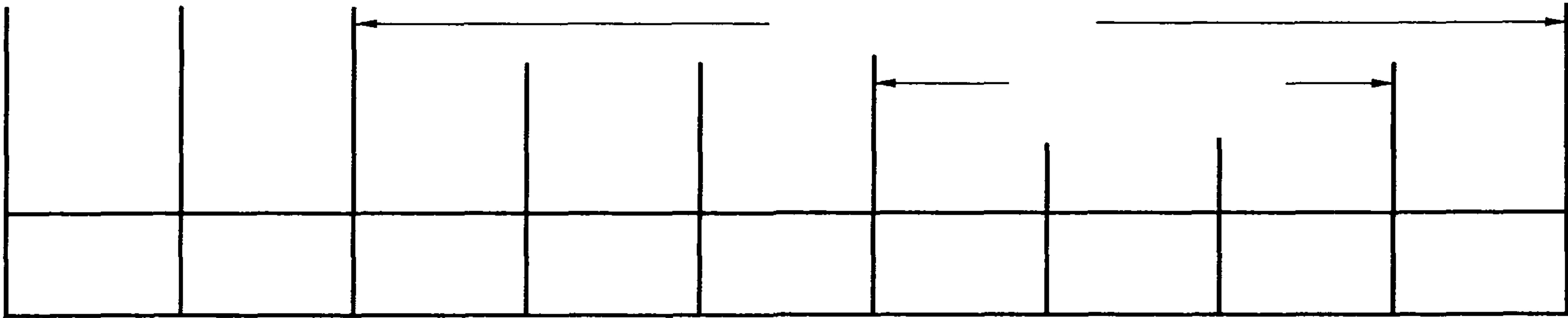


图 6 结构类型规范正规编码的例子

7.3.1.5 结构类型信息的缩略编码

下面例子说明了结构类型规范的缩略编码。实际上这是一个在本部分 5.4——数据类型规范/词典

中定义的 AbbreviatedStrucTypeSpec 生成的编码实例。

在 AbbreviatedStrucTypeSpec 生成中定义的 UNIT 应以一个 16 位循环冗余校验(CRC)值起始化。应用程序逻辑可藉此判断结构形式是否已经改变。用以产生 CRC 的字节流是实际的正规编码 (FormalStrucTypeSpec) 结构类型规范。

7.3.1.6 例子

图 7 举例说明了之前提过的结构定义的缩略编码。

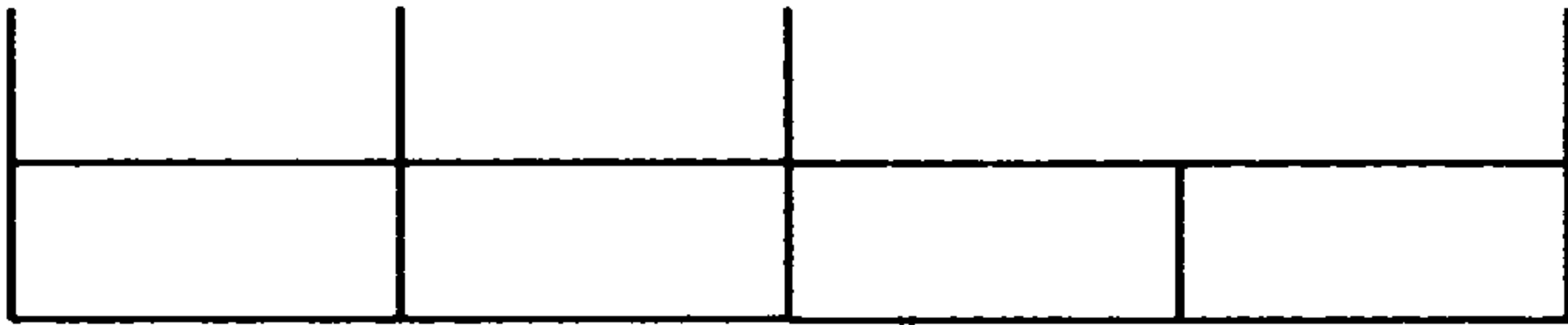


图 7 结构类型规范缩略编码的例子

注：从正规编码类型规范产生 CRC 值为 0x26C7 的运算规则为:[A2][07][C7][A2][03][C7][C2][C3][C3]。

7.3.2 数组类型定义

控制网络定义了两种不同的方法,来报告数组类型定义:

- 正规编码 (FormalArrayTypeSpec);
- 缩略编码 (AbbreviatedArrayTypeSpec)。

注：正规编码用于提供整个数组定义的详细报告,包括数据内容和数组维度。缩略编码用于规定数组定义的一种简短形式。这种简短形式不包括指定数组维度的信息。

7.3.3 数组类型信息的正规编码

7.3.3.1 示例 1

- 注 1: 本节不含规范性要求。
- 注 2: 下面示例说明了结构类型规范的正规编码。实际上这是一个在本部分 5.4——数据类型规范/词典中定义的 FormalArrayTypeSpec 生成的编码实例。

图 8 举例说明了以下数组定义的正规编码。

```
ARRAY1 ::= SEQUENCE OF { array_dimension_low_bound := 0,
                           array_dimension_high_bound := 9,
                           UINT }
```

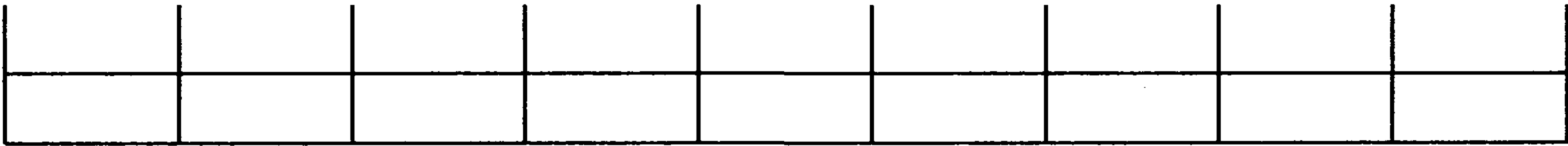


图 8 数组类型规范正规编码的例子 1

注：零(0)长度字节不跟随 DataTypeSpecification(数据类型规范)生成中的 IMPLICIT NULL 类型。

7.3.3.2 示例 2

图 9 举例说明了以下数组定义的编码。

```
ARRAY1 ::= SEQUENCE OF { array_dimension_low_bound := 0,
                           array_dimension_high_bound := 19,
                           SEQUENCE OF { array_dimension_low_bound := 0,
                                           array_dimension_high_bound := 255,
```

STRUCT_ELE } }

其中 STRUCT_ELE 定义如下：

STRUCT_ELE ::= SEQUENCE OF { UINT, SINT, INT }

正规编码:[A3][13][80][01][00][81][01][13][A3][0B][80][01][00][81][01][FF][A2][03][C7][C2][C3]

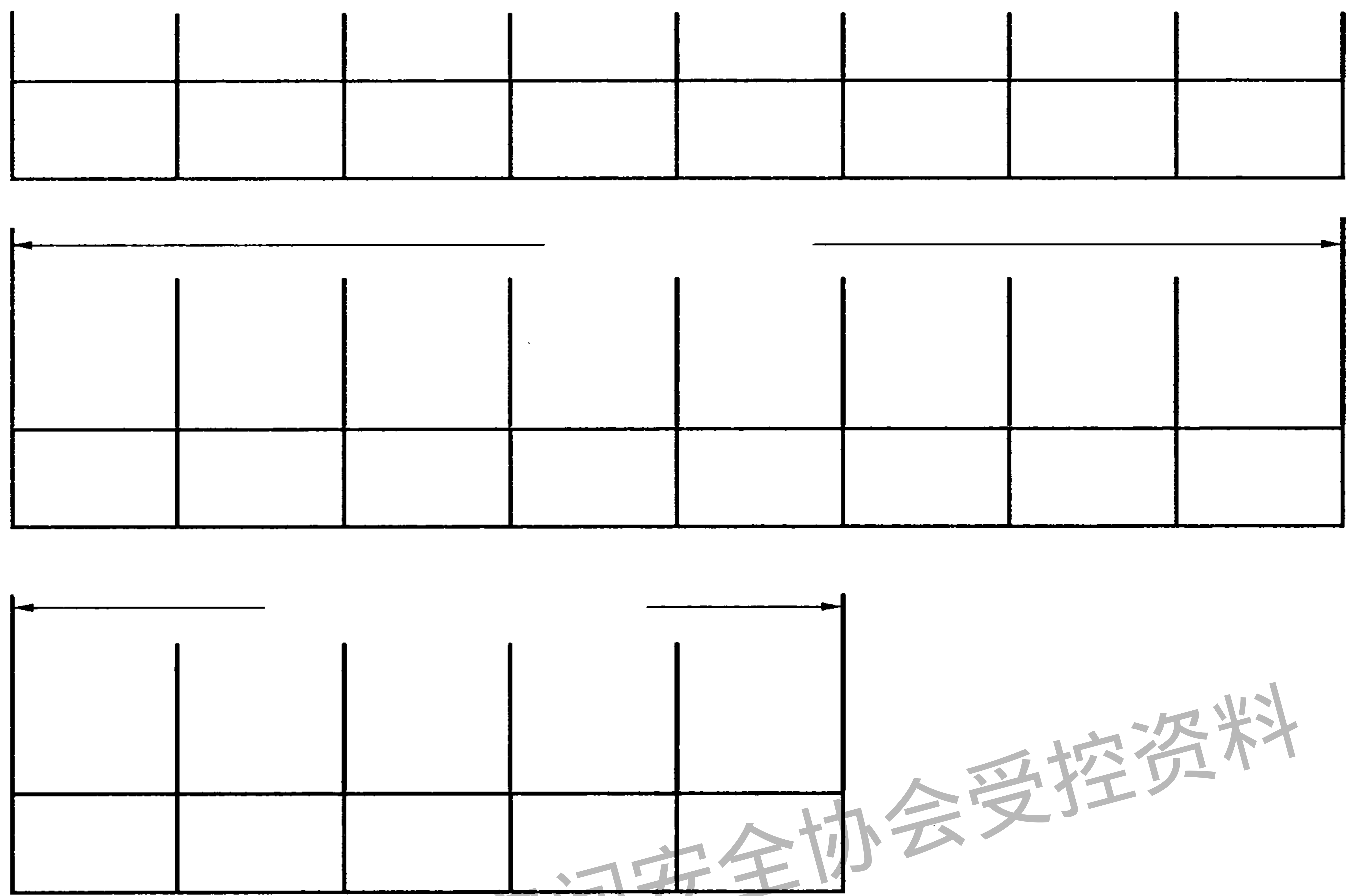


图 9 数组类型规范正规编码的例子 2

注：零(0)长度字节不跟随 DataTypeSpecification(数据类型规范)生成中的 IMPLICIT NULL 类型。

7.3.4 数组类型信息的缩略编码

7.3.4.1 示例 1

注 1：本节不含规范性要求。
注 2：数组类型的缩略编码不包括指定数组维度的信息。实际上这是一个在本部分 5.4——数据类型规范/词典中定义的 AbbreviatedArrayTypeSpec 生成的编码实例。
下列数组定义的缩略编码如图 10 所示。

ARRAY2 ::= SEQUENCE OF { array_dimension_low_bound := 0,
array_dimension_high_bound := 9,
UINT }

数组类型	类型长度	UINT
A1	01	C7

图 10 数组类型规范缩略编码的例子 1

注：零(0)长度字节不跟随 DataTypeSpecification(数据类型规范)生成中的 IMPLICIT NULL 类型。

7.3.4.2 示例 2

图 11 举例说明了以下数组定义的缩略编码。

ARRAY ::= SEQUENCE OF { array_dimension_low_bound := 0,
array_dimension_high_bound := 19,
SEQUENCE OF { array_dimension_low_bound := 0,
array_dimension_high_bound := 899,
STRUCT_ELE } }

其中 STRUCT_ELE 定义如下：
STRUCT_ELE ::= SEQUENCE OF { UINT, SINT, INT }

数组类型	类型长度						
		数组类型	类型长度	结构类型	类型长度	含CRC的UINT	
A1	06	A1	A1	A0	02	59	51

图 11 数组类型规范缩略编码的例子 2

参 考 文 献

- [1] GB/T 16262.1—2006 信息技术 抽象语法记法—(ASN.1) 第1部分:基本记法规范 (ISO/IEC 8824-1:2002, IDT)
- [2] GB/T 16262.2—2006 信息技术 抽象语法记法—(ASN.1) 第2部分:信息客体规范 (ISO/IEC 8824-2:2002, IDT)
- [3] GB/T 16262.3—2006 信息技术 抽象语法记法—(ASN.1) 第3部分:约束规范 (ISO/IEC 8824-3:2002, IDT)
- [4] GB/T 16262.4—2006 信息技术 抽象语法记法—(ASN.1) 第4部分:ASN.1 规范的参数化 (ISO/IEC 8824-4:2002, IDT)
- [5] GB/T 16263.1—2006 信息技术 ASN.1 编码规则 第1部分:基本编码规则(BER)、正则编码规则(CER)和非典型编码规则(DER)规范 (ISO/IEC 8825-1:2002, IDT)
- [6] GB/T 16263.2—2006 信息技术 ASN.1 编码规则 第2部分:紧缩编码规则(PER)规范 (ISO/IEC 8825-2:2002, IDT)
- [7] GB/T 15969.3—2005 可编程序控制器 第3部分:编程语言 (IEC 61131-3:2002, IDT)

广东省网络空间安全协会受控资料

广东省网络空间安全协会受控资料

中 华 人 民 共 和 国
国 家 标 准
智能传感器 第2部分:物联网应用行规
GB/T 33905.2—2017

*

中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)

网址 www.spc.net.cn
总编室:(010)68533533 发行中心:(010)51780238
读者服务部:(010)68523946

中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

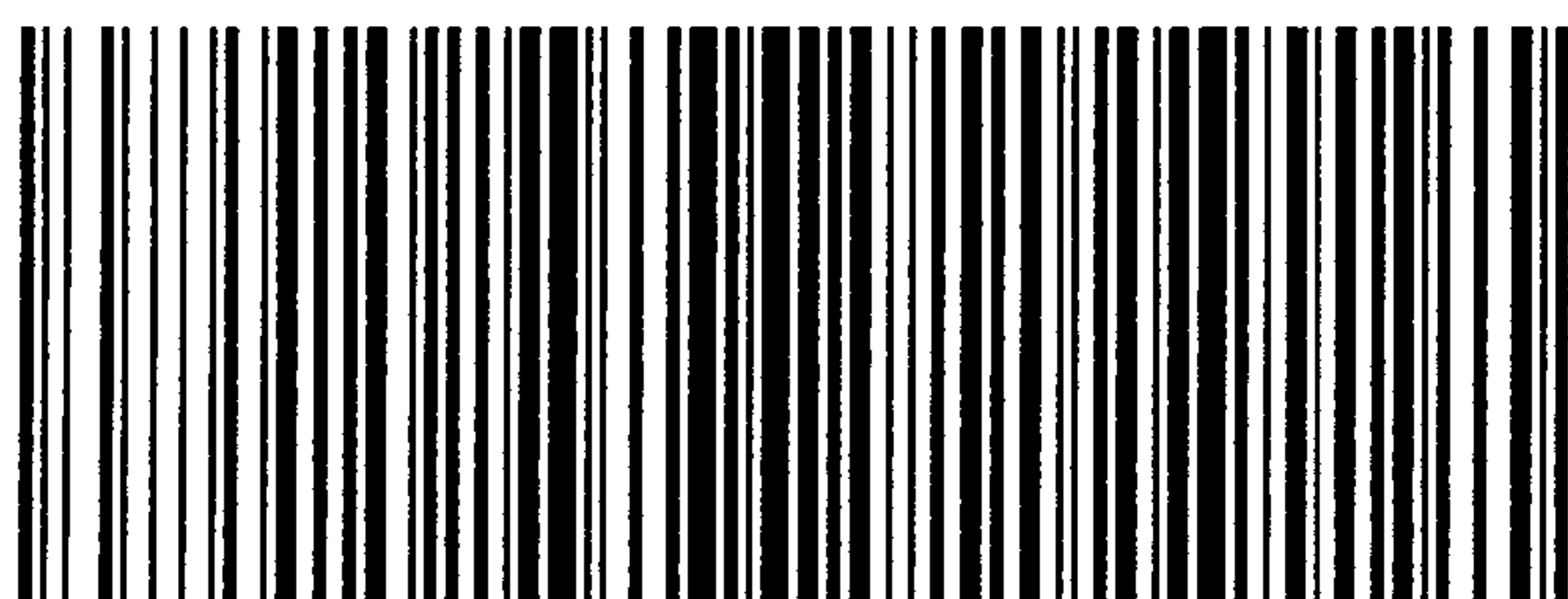
*

开本 880×1230 1/16 印张 3.75 字数 103 千字
2017年8月第一版 2017年8月第一次印刷

*

书号: 155066·1-56181 定价 51.00 元

如有印装差错 由本社发行中心调换
版权专有 侵权必究
举报电话:(010)68510107



GB/T 33905.2—2017