



中华人民共和国国家标准

GB/T XXXX. 12—XXXX

现场设备集成 第12部分：行规 HART 和 WirelessHART

Field Device Integration (FDI) - Part 12: Profiles - HART and WirelessHART

IEC 62769-7:2021, IDT

（征求意见稿）

（本稿完成日期：2022-01）

XXXX - XX - XX 发布

XXXX - XX - XX 实施

国家市场监督管理总局
国家标准化管理委员会

目 次

前言	II
1 范围	1
2 规范性引用文件	1
3 术语、定义、缩略语	1
3.1 术语和定义	1
3.2 缩略语	1
3.3 约定	1
4 CP 9/1(HART®) 或 CP 9/2(WirelessHART®)行规.....	2
4.1 概述	2
4.2 目录行规	2
4.3 将包与 CP 9/1 设备关联	3
4.4 信息模型映射	4
4.5 拓扑元素	5
4.6 方法	14
附录 A (规范性附录) 拓扑扫描架构.....	28
A.1 概述	28
A.2 IdentificationType	28
A.3 AddressTypeTP	30
A.4 AddressTypeIP	31
A.5 AddressTypeTDMA	32
A.6 AddressType	32
A.7 ConnectionPointType	33
A.8 NetworkType	33
A.9 Network	34
附录 B (规范性附录) 传输服务参数.....	35
B.1 概述	35
B.2 B.2 receiveData	35
B.3 B.3 sendData	35
B.4 B.4 TransferResultDataT	35
B.5 B.5 TransferSendDataT	36
参考文献	37
表 1 CommunicationProfile 定义	2
表 2 行规设备的目录值.....	3

表 3	协议版本信息.....	3
表 4	设备类型目录映射.....	4
表 5	ProtocolType HART 定义.....	4
表 6	继承的 DeviceType 特性映射.....	4
表 7	标识参数.....	5
表 8	ConnectionPointType HART_TP 定义.....	6
表 9	ConnectionPointType HART_IP 定义.....	9
表 10	ConnectionPointType HART_TDMA 定义.....	11
表 11	方法 Connect 的参数.....	14
表 12	方法 Disconnect 的参数.....	15
表 13	方法 Transfer 的参数.....	16
表 14	方法 GetPublishedData 的参数.....	17
表 15	方法 SetAddress 的参数.....	17
表 16	方法 Connect 的参数.....	19
表 17	方法 Disconnect 的参数.....	20
表 18	方法 Transfer 的参数.....	21
表 19	方法 GetPublishedData 的参数.....	23
表 20	方法 SetAddress 的参数.....	24
表 A.1	IdentificationT 的特性.....	29
表 A.2	AddressTypeTP 的元素.....	31
表 A.3	AddressTypeIP 的元素.....	31
表 A.4	AddressTypeTDMA 的元素.....	32
表 A.5	AddressT 的元素.....	33
表 A.6	ConnectionPointT 的元素.....	33
表 A.7	TNetworkT 的元素.....	34
表 B.1	TransferResultDataT 的特性.....	35
表 B.2	TransferSendDataT 的特性.....	36

前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》和 GB/T 1.2—2020《标准化工作导则 第2部分：以 ISO/IEC 标准化文件为基础的标准化文件起草规则》的规定起草。

本文件是GB/T XXXX的第7部分。GB/T XXXX已经发布了以下部分：

- 第1部分：概述；
- 第2部分：客户端；
- 第3部分：服务器；
- 第4部分：包；
- 第5部分：信息模型。

本文件等同采用 IEC 62769-7: 2021《现场设备集成（FDI）第7部分：通信设备》，文件类型由 IEC 的技术标准调整为我国的国家标准。

本文件由中国机械工业联合会提出。

本文件由全国工业过程测量控制和自动化标准化技术委员会（SAC/TC124）归口。

本文件起草单位：

本文件主要起草人：卢铁林

本文件为首次发布。

引 言

现场设备集成（FDI）系列标准是指导工业过程测量控制等相关活动的重要技术标准，GB/T XXXX旨在确立适用于设备集成的规范准则，拟由十二个部分构成。

- 第1部分：概述；
- 第2部分：客户端；
- 第3部分：服务器；
- 第4部分：包；
- 第5部分：信息模型；
- 第6部分：技术映射；
- 第7部分：通信设备；
- 第8部分：行规 基金会现场总线H1；
- 第9部分：行规 基金会现场总线HSE；
- 第10部分：行规 PROFIBUS
- 第11部分：行规 PROFINET；
- 第12部分：行规 HART和WirelessHART。

现场设备集成 第12部分：行规 HART®和 WirelessHART®

1 范围

本文件规定了IEC 62769对IEC 61784-1_CP 9/1 HART®和IEC 61784-1_CP 9/2 WirelessHART®的FDI行规。

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，仅所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。

- IEC 62541-100: 2015 OPC统一架构 第100部分：设备接口
- IEC 62769-4 现场设备集成(FDI) 第4部分：FDI包
- IEC 62769-5 现场设备集成(FDI) 第5部分：FDI信息模型
- IEC 62769-7 现场设备集成(FDI) 第7部分：FDI通信设备

3 术语、定义、缩略语

3.1 术语和定义

XXXX界定的以及下列术语和定义适用于本文件。

3.2 缩略语

下列缩略语适用于本文件。

CP	Communication profile	通信行规
CPF	Communication profile family	通信行规族
EDD	Electronic Device Description	电子设备描述
EDDL	Electronic Device Description Language	电子设备描述语言
FDI	Field Device Integration	现场设备集成
FSK	Frequency-Shift-Keying	频移键控
HCF	HART Communication Foundation	HART通信基金会
ID	Identification	标识
IM	Information Model	信息模型
IP	Internet protocol	因特网协议
PDU	Protocol data unit	协议数据单元
PSK	Phase-Shift-Keying	相移键控
TCP	Transmission Control Protocol	传输控制协议
UDP	User Datagram Protocol	用户数据报协议
XML	Extensible markup language	可扩展标记语言

3.3 约定

3.3.1 EDDL 语法

本文件规定了作为FDI通信包的一部分的EDD组件的内容。EDDL语法使用Courier New字体。EDDL语法用于方法签名、变量、数据结构和组件声明。

3.3.2 XML 语法

XML语法示例使用Courier New字体。XML语法用于描述XML文档模式。

例如: `<xs:simpleType name="ExampleT">`

3.3.3 大写

IEC 62769 系列使用大写的术语来强调这些术语具有 FDI 特定的含义。

其中一些术语使用首字母缩略词作为前缀, 例如:

- FDI 客户端, 或
- FDI 服务器。

其中一些术语是复合术语, 例如:

- 通信服务器, 或
- 行规包。

参数名称或属性连接到一个术语中, 其中初始术语在该术语中以大写字母开头, 例如:

- ProtocolSupportFile, 或
- ProtocolType。

参数名称或属性也可使用下划线字符连接两个或多个术语来构造, 例如:

- PROFILE_ID, 或
- HART_Network。

4 CP 9/1 (HART®) 或 CP 9/2 (WirelessHART®) 行规

4.1 概述

在IEC 62769中FDI规范中的行规文档规定了描述通信服务器、网关和设备的FDI包所需的协议细节。

4.2 目录行规

4.2.1 协议支持文件

设备信息文件为设备支持的动态运行时数据提供元数据。该元数据是包含在EDD的信息子集。轻量级网关或服务器实现可从包中提取设备信息文件, 以便以最小实现开销交换运行时设备信息。设备信息文件不会取代EDD部分的需求, 因为设备信息文件仅包含EDD信息的子集, 且不提供任何用户接口元素。

表1规定了设备信息文件的格式。

表1 设备信息文件

部分	内容
Content Type	无规定
Root Namespace	无规定
Source Relationship	http://fdi-cooperation.com/2010/relationship/attachment-protocol
Filename	无规定

FCG AG21073规定了设备信息文件。

4.2.2 CommunicationProfile 定义

IEC 62769-4为Catalog XML模式定义了CommunicationProfileT字符串类型。表2定义了此枚举的CP 9/1特定值。

表2 CommunicationProfile 定义

CommunicationProfile	说明
hart_fsk	支持 FSK 物理层的 CP 9/1 设备类型（一对线缆上的频移键控）
hart_psk	支持 PSK 物理层的 CP 9/1 设备类型（一对线缆上的相移键控）。支持 PSK 的设备还需要固有支持 FSK。因此 PSK 将始终只与 FSK 结合使用。
hart_wirelesshart	支持无线物理层（设备和网关之间的通信）的 CP 9/2 设备类型。
hart_ip	支持 IP 的 CP 9/1 设备类型（这些设备同时支持 TCP 和 UDP）。
hart_rs485	支持 EIA-485 数字通信的 CP 9/1 设备类型。
hart_ir	支持红外物理层的 CP 9/1 设备类型（设计用于透传给 FSK 主设备，仅作为信息用于指示设备支持红外连接）。

注：单个 CP 9/1 设备可能支持多个 CP。

4.2.3 行规设备

行规包应提供设备配置文件的目录值，如果没有特定的设备描述，则 FDI 服务器能够使用通用的设备描述。表 3 中的定义侧重于与供应商无关的目录内容。

表3 行规设备的目录值

元素	属性	内容
PackageType	—	行规
DeviceModel	—	空

Manufacturer	—	空
--------------	---	---

4.2.4 协议版本信息

IEC 62769-4 为目录 XML 模式定义了一个名为 InterfaceT 的元素类型。元素类型 InterfaceT 包含一个名为 Version 的元素，该元素应提供有关所应用的通信协议行规的版本信息。该值必须遵循在元素类型 VersionT 中定义的 IEC 62769-4 定义的版本信息模式。4.2.4 描述了如何为设备目录中的 CP 9/1 或 CP 9/2 应用当前已知的协议版本。一般规则是根据 IEC 62769-4 对版本信息中未在当前已知协议版本中使用的部分使用值“0”。协议版本信息见表 4。

表4 协议版本信息

协议版本	InterfaceT 版本值
HART 通用修订版 5	5.0.0
HART 通用修订版 6	6.0.0
HART 通用修订版 7	7.0.0
包中定义的协议版本仅供参考，不得用于确定包对设备的兼容性或适用性。	

4.3 将包与 CP 9/1 设备关联

4.3.1 设备类型标识映射

CP 9/1 设备类型由 Manufacturer, Model 和 DeviceRevision 唯一标识。这些参数用于将给定的设备实例与 FDI 设备包关联。根据表 5，这些参数映射到 FDI 设备包目录。

表5 设备类型目录映射

目录元素	CP 映射(见表 A.1)
InterfaceT 的 Manufacturer 元素 (IEC 62769 4: -, 第 E.11 条)	Manufacturer 字符串格式“0xdddd”，其中 dddd 是十六进制格式的 MANUFACTURER_ID。
InterfaceT 的 DeviceMode 元素 (IEC 62769 4: -, 第 E.11 条)	Model 字符串格式“0xdddd”，其中 dddd 是十六进制格式的 DEVICE_TYPE。
DeviceRevision 元素 ListOfSupportedDeviceRevisionsT (IEC 62769-4: -, 第 E.21 条)	DeviceRevision 字符串格式“x.0.0”，其中 x 是十进制格式的 DEVICE_REVISION (无前导零)。

4.3.2 设备类型修订版本映射

每种设备类型按照5.3.1进行标识。若没有与DeviceRevision匹配的包，只要现场设备的设备修订版本等于或大于FDI包中指定的设备修订版本，相应制造商和型号的任何CP 9/1的FDI包应始终与现场设备兼容。

4.4 信息模型映射

4.4.1 ProtocolType 定义

表6定义了用于标识CP 9/1网络通信的ProtocolType。

表6 ProtocolType HART 定义

属性	值				
BrowseName	HART				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
继承 IEC 62541-100 中定义的 ProtocolType 的特性。					

4.4.2 DeviceType 映射

每个设备类型都继承DeviceType的特性。表7定义了从DeviceType继承特性的映射。

表7 继承的 DeviceType 特性映射

特性	基础映射
SerialNumber	设备的唯一 ID，映射到 IdentificationT 的 SERIAL_NUMBER。
RevisionCounter	配置更改计数器，映射到 IdentificationT 的 REV_COUNTER。
Manufacturer	取自 FDI 包目录的字符串（Manufacturer 名称来自 PackageT）
Model	取自 FDI 包目录的字符串（DeviceTypeT 的名称，是本地化名称）。
DeviceManual	条目文本字符串（不支持） ^a
DeviceRevision	设备的设备修订版本，映射到 IdentificationT 的 REVICE_REVISION。
SoftwareRevision	设备的软件修订版本，映射到 IdentificationT 的 SOFTWARE_REVISION。
HardwareRevision	设备的硬件修订版本，映射到 IdentificationT 的 HARDWARE_REVISION。
^a 设备手册作为 FDI 设备包的附件被公开。	

4.4.3 FunctionalGroup 标识定义

根据IEC 62541-100的定义，在FDI服务器托管的信息模型中的每个设备描述应包含被称为标识的协议特定的FunctionalGroup。这个FunctionalGroup组织在设备类型实例中创建的变量。表8定义了CP 9/1的FunctionalGroup标识。

表8 标识参数

BrowseName	DataType	可选/强制
MANUFACTURER_ID	UInt16	强制
DEVICE_TYPE	UInt16	强制
DEVICE_REVISION	UInt8	强制
UNIVERSAL_REVISION	UInt8	可选
SERIAL_NUMBER	UInt24	可选
HARDWARE_REVISION	UInt8	可选
SOFTWARE_REVISION	UInt8	可选
REVISION_COUNTER	UInt16	可选

4.5 拓扑元素

4.5.1 ConnectionPoint 定义

4.5.1.1 概述

CP 9/1设备最多能支持五种用于网络通信的不同ConnectionPoint类型。

4.5.1.2 HART_TP5, HART_TP6, HART_TP7

ConnectionPoint类型HART_TP5、HART_TP6和HART_TP7应用于标识CP 9/1令牌传递通信网络，并在表9定义。HART_TP5、HART_TP6和HART_TP7都包含相同的特性，但每个特性都为某些特性提供不同的限定信息（如下所述）。4.2.3中描述的协议版本（UNIVERSAL_REVISION）可用于帮助确定三种令牌传递ConnectionPoint类型中哪一种最合适。CP 9/1令牌传递通信可用于各种物理层。FSK、PSK、RS485和红外物理层连接均应使用HART_TP连接类型。ConnectionPoint类型HART_TP5、HART_TP6和HART_TP7是IEC 62769-5中定义的抽象类型ConnectionPointType的子类型。

DevAddr特性应为设备的长地址（5字节），并且是与现场设备通信所必需的唯一参数。

DevMfg特性应为2字节的Manufacturer ID，并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevType特性应为2字节扩展设备类型，并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevRev特性应为设备修订版本，并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevTag特性应为HART协议版本6或7的设备的长位号。或者为协议版本5的设备的位号。DevTag特性有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。HART_TP5连接点应将DevTag长度限制为8个字符。HART_TP6和HART_TP7连接点应将DevTag的长度限制为32个字符。

DevPollAddr特性应为轮询地址，并可用于识别位于特定轮询地址的设备。对于DevPollAddr特性，HART_TP5连接点应限制在0到15之间。而对于DevPollAddr特性，HART_TP6连接点应限制在0到31之间。对于DevPollAddr特性，HART_TP7连接点应限制在0到63之间。

对于向前兼容性，较低版本的HART_TP连接点是兼容的，可用于较高版本的通用设备连接。例如，若遇到将来的HART通用版本8设备，并且FDI服务器中没有可用的HART_TP8，则HART_TP7应兼容并用于连接设备。如果由于任何原因，协议版本（即通用版本）是未知的，则可使用HART_TP5连接点，并将向前兼容以后的通用版本。

表9 ConnectionPointType HART_TP 定义

属性	值				
BrowseName	ConnectionPoint_HART_TP5 或 ConnectionPoint_HART_TP6 或 ConnectionPoint_HART_TP7				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
继承 IEC 62769-5 中定义的 ConnectionPointType 的特性。					
HasProperty	变量	DevAddr	UInt40	PropertyType	强制
HasProperty	变量	DevMfg	UInt16	PropertyType	可选
HasProperty	变量	DevType	UInt16	PropertyType	可选
HasProperty	变量	DevRev	UInt16	PropertyType	可选
HasProperty	变量	DevTag	String	PropertyType	可选
HasProperty	变量	DevPollAddr	UInt8	PropertyType	可选

ConnectionPoint 类型 HART_TP5, HART_TP6 和 HART_TP7 应通过包含在可驱动 HART 网络的与通信设备相关的 FDI 包中的 EDD 元素进行描述。实际的 ConnectionPoint 特性由组合在一个 COLLECTION 中，名为 ConnectionPoint_HART_TP5 或 ConnectionPoint_HART_TP6 或 ConnectionPoint_HART_TP7 的结构 VARIABLE 来声明。以下 EDDL 源代码是描述 TP5 Connection Point 的示例。

```

COMPONENT ConnectionPoint_HART_TP5
{
    LABEL "HART TP Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_TP5;
}

VARIABLE DevAddr
{
    LABEL "Address";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(5);
    HANDLING READ & WRITE;
}

```

```
VARIABLE DevMfg
{
  LABEL "Manufacturer";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(2);
  HANDLING READ & WRITE;
}

VARIABLE DevType
{
  LABEL "Device Type";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(2);
  HANDLING READ & WRITE;
}

VARIABLE DevRev
{
  LABEL "Device Revision";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER;
  HANDLING READ & WRITE;
}

VARIABLE DevTag
{
  LABEL "Tag";
  CLASS DEVICE;
  TYPE ASCII(32);
  HANDLING READ & WRITE;
}

VARIABLE DevPollAddr
{
  LABEL "Poll Address";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER
  {
    MAX_VALUE 15; //Define appropriate max value for various revisions
  }
  HANDLING READ & WRITE;
}
```

```

COLLECTION ConnectionPoint_TP5
{
  LABEL "Connection Point";
  MEMBERS
  {
    ADDRESS, DevAddr, "Device Address";
    MFG, DevMfg, "Manufacturer";
    DEV_TYPE, DevType, "Device Type";
    DEV_REV, DevRev, "Device Revision";
    TAG, DevTag, "Device Tag";
    POLL_ADDR, DevPollAddr, "Poll Address";
  }
}

```

4.5.1.3 HART_IP

ConnectionPoint类型HART_IP应用于识别CP 9/1 IP网络通信，定义见表10。HART_IP通信可用于多种物理层，以太网连接应全部采用HART_IP连接类型。将来开发的其他物理层也可使用HART_IP连接类型。ConnectionPoint类型HART_IP是IEC 62769-5中定义的抽象类型ConnectionPointType的子类型。

IPAddress特性应指示用于连接的IP地址（最多16字节）。

IPVersion特性应指示用于连接的IP版本（4或6）。

IPPort特性应为连接的IP端口号。用于HART IP的端口号缺省值是5094。

DevAddr特性应为设备的长地址（5字节）。

DevMfg特性应为2字节的Manufacturer ID，并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevType特性应为2字节扩展设备类型，并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevRev特性应为设备修订版本，并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevTag特性应为长位号（最多32个字符），并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

表10 ConnectionPointType HART_IP 定义

属性	值				
BrowseName	ConnectionPoint_HART_IP				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
继承 IEC 62769-5 中定义的 ConnectionPointType 的特性。					
HasProperty	变量	IPAddress	ByteString	PropertyType	强制
HasProperty	变量	IPVersion	UInt8	PropertyType	强制
HasProperty	变量	IPPort	UInt16	PropertyType	强制
HasProperty	变量	DevAddr	UInt40	PropertyType	强制
HasProperty	变量	DevMfg	UInt16	PropertyType	可选
HasProperty	变量	DevType	UInt16	PropertyType	可选
HasProperty	变量	DevRev	UInt16	PropertyType	可选
HasProperty	变量	DevTag	String	PropertyType	可选

ConnectionPoint 类型 HART_IP 应由与通信设备相关的 FDI 包中包含的 EDD 元素进行描述，该 FDI 包可驱动 CP 9/1 网络。实际的 ConnectionPoint 特性由 VARIABLE 结构来声明，这些结构在一个名为 ConnectionPoint_HART_IP 的 COLLECTION 中组合在一起。以下 EDDL 源代码是描述 IP Connection Point 的示例。

```

COMPONENT ConnectionPoint_HART_IP
{
    LABEL "HART IP Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_IP;
}

ARRAY IPAddress
{
    LABEL "IP Address";
    CLASS DEVICE;
    TYPE OCTET(16);
    HANDLING READ & WRITE;
}

VARIABLE IPVersion
{
    LABEL "IP Version";
    CLASS DEVICE;
    TYPE ENUMERATED
    {
        { 4, "IPv4" },
        { 6, "IPv6" }
    }
    HANDLING READ & WRITE;
}

```

```

}

VARIABLE IPPort
{
    LABEL "IP Port";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER (2);
    DEFAULT_VALUE 5 094;
    HANDLING READ & WRITE;
}

COLLECTION ConnectionPoint_IP
{
    LABEL "Connection Point";
    MEMBERS
    {
        IPADDRESS, IPAddress, "IP Address";
        IPVERSION, IPVersion, "IP Version";
        IPPORT, IPPort, "IP Port";
        ADDRESS, DevAddr, "Device Address";
        MFG, DevMfg, "Manufacturer";
        DEV_TYPE, DevType, "Device Type";
        DEV_REV, DevRev, "Device Revision";
        TAG, DevTag, "Device Tag";
    }
}

```

4.5.1.4 HART_TDMA

ConnectionPoint 类型 HART_TDMA 应用于标识 CP 9/2 时分媒体访问网络通信,定义见表 11。HART_TDMA 通信可用于多种物理层。ConnectionPoint 类型 HART_TDMA 是 IEC 62769-5 中定义的抽象类型 ConnectionPoint 类型的一个子类型。WirelessHART 连接应全部采用 HART_TDMA 连接类型。将来开发的其他物理层也可使用 HART_TDMA 连接类型。

Network 特性应为网络的网络 ID。

DevAddr 特性应为设备的长地址（5 字节）。

DevMfg 特性应为 2 字节的 Manufacturer ID, 并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevType 特性应为 2 字节扩展设备类型, 并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevRev 特性应为设备修订版本, 并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

DevTag 特性应为长位号（最多 32 个字符）, 并有助于将扫描列表中的活动设备分配给脱机占位符的过程实现自动化。

表11 ConnectionPointType HART_TDMA 定义

属性	值				
BrowseName	ConnnectionPoint_HART_TDMA				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
继承 IEC 62769-5 中定义的 ConnectionPointType 的特性。					
HasProperty	变量	Network	UInt16	PropertyType	强制
HasProperty	变量	DevAddr	UInt40	PropertyType	强制
HasProperty	变量	DevMfg	UInt16	PropertyType	可选
HasProperty	变量	DevType	UInt16	PropertyType	可选
HasProperty	变量	DevRev	UInt16	PropertyType	可选
HasProperty	变量	DevTag	String	PropertyType	可选

ConnectionPoint 类型 HART_TDMA 应由与通信设备相关的 FDI 包中包含的 EDD 元素进行描述,该 FDI 包可驱动 CP 9/2 网络。实际的 ConnectionPoint 特性由 VARIABLE 结构来声明,这些结构在一个名为 ConnectionPoint_HART_TDMA 的 COLLECTION 中组合在一起。以下 EDDL 源代码是描述 TDMA Connection Point 的示例。

```

COMPONENT ConnectionPoint_HART_TDMA
{
    LABEL "HART TDMA Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_TDMA;
}

VARIABLE Network
{
    LABEL "Network ID";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER (2);
    HANDLING READ & WRITE;
}

```

```

}

COLLECTION ConnectionPoint_TDMA
{
  LABEL "Connection Point";
  MEMBERS
  {
    NETWORK, Network, "Network ID";
    ADDRESS, DevAddr, "Device Address";
    MFG,      DevMfg,  "Manufacturer";
    DEV_TYPE, DevType, "Device Type";
    DEV_REV,  DevRev,  "Device Revision";
    TAG,      DevTag,  "Device Tag";
  }
}

```

4.5.2 通信设备定义

根据IEC 62769-7，每个FDI通信包应包含一个描述通信设备的EDD元素。以下的EDDL源代码是描述FDI通信服务器的示例。

```

COMPONENT CommunicationServer_HART
{
  LABEL "HART Communication Server";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK_COMPONENT;
  COMPONENT_RELATIONS { Communication_Device_Setup_HART}
}

COMPONENT_RELATION Communication_Device_Setup_HART
{
  LABEL "Relation between Device and Communication Device";
  RELATION_TYPE CHILD_COMPONENT;
  COMPONENTS
  {
    CommunicationDevice_HART{AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 4;
}

```

上述EDDL结构的语义在IEC 62769-7中进行了描述。FDI服务器和FDI通信服务器将使用EDDL COMPONENT创建CommunicationServerType类型的实例，如IEC 62769-7中所述。

根据IEC 62769-7，每个FDI通信包应包含至少一个EDD元素，描述至少一个通信设备组件。下面的EDDL源代码是CP 9/1通信设备的示例。

```

COMPONENT CommunicationDevice_HART
{
  LABEL "HART Communication Device";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK_COMPONENT;
  COMPONENT_RELATIONS { ServiceProviderRelation_HART}
  BYTE_ORDER BIG_ENDIAN;
}

```

```

COMPONENT_RELATION ServiceProviderRelation_HART
{
  LABEL "Relation to communication service provider";
  RELATION_TYPE CHILD_COMPONENT;
  COMPONENTS
  {
    CommunicationServiceProvider_HART {AUTO_CREATE 1;}
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 1;
}

```

上述EDDL结构的语义在IEC 62769-7中描述。FDI服务器和FDI通信服务器将使用EDDL COMPONENT创建CommunicationServerChannelType类型的实例，如IEC 62769-7所述。

4.5.3 通信服务提供者定义

根据IEC 62769-7，每个FDI通信包应包含至少一个EDD元素，描述至少一个通信服务提供者组件。下面的EDDL源代码是CP 9/1通信服务提供者组件的示例。

组件引用（ConnectionPoint_HART_IP）对应于4.5.1中的相关连接点定义之一。属性BYTE_ORDER值将根据协议设置。

```

COMPONENT CommunicationServiceProvider_HART
{
  LABEL "HART Communication Service Provider";
  CAN_DELETE FALSE;
  CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
  COMPONENT_RELATIONS { ServiceProviderConnectionPointRelation_HART }
  BYTE_ORDER BIG_ENDIAN;
}

COMPONENT_RELATION ServiceProviderConnectionPointRelation_HART
{
  LABEL "Relation between communication service and connection point";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING
  {
    IPAddress
  }
  COMPONENTS
  {
    ConnectionPoint_HART_IP { AUTO_CREATE 1; }
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 1;
}

```

上述EDDL结构的语义在IEC 62769-7中描述。FDI服务器和FDI通信服务器将使用EDDL COMPONENT创建IEC 62769-7中所述的CommunicationServiceType类型的实例。

4.5.4 网络定义

根据IEC 62769-7，每个FDI通信包应至少包含一个EDD元素，以使用组件结构描述网络配置约束。下面的EDDL源代码是一个描述网络的示例。

```

COMPONENT Network_HART
{
  LABEL "HART Network";
  CAN_DELETE TRUE;
  CLASSIFICATION NETWORK;
  COMPONENT_RELATIONS { NetworkConnectionPointRelation_HART }
}

COMPONENT_RELATION NetworkConnectionPointRelation_HART
{
  LABEL "Relation between network and connection point";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING {DevPollAddr}
  COMPONENTS
  {
    ConnectionPoint_HART_TP5
    {
      MAXIMUM_NUMBER 16;
    },
    ConnectionPoint_HART_TP6
    {
      MAXIMUM_NUMBER 32;
    },
    ConnectionPoint_HART_TP7
    {
      MAXIMUM_NUMBER 64;
    }
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER 64; //Defined to limitations of the comm. device
}

```

上述EDDL结构的语义在IEC 62769-7中描述。FDI服务器和FDI通信服务器将使用EDDL COMPONENT创建NetworkType类型的实例，如IEC 62769-5所述。一些通信设备可以将多个IO卡组织成单独的网络，而另一些通信设备可以呈现出包含所有IO卡上的所有连接点的单个网络。

4.6 方法

4.6.1 FDI 通信服务器的方法

4.6.1.1 概述

FDI通信服务器应根据4.6.1中描述的方法签名和信息模型实现服务。

4.6.1.2 Connect

方法Connect的参数见表12。

Signature:

```

Connect (
  [in]ByteString CommunicationRelationId,
  [in]ByteString LongAddress,
  [out]Int32      ServiceError);

```

表12 方法 Connect 的参数

参数	说明
CommunicationRelationId	该参数值包含设备 ConnectionPoint 的 nodeId，表示设备与直接连接到 FDI 通信服务器硬件的物理网络之间的连接。nodeId 允许查找直接的父子关系。
LongAddress	该参数名应与为 ConnectionPoint 定义的相应属性名匹配，该属性名由 4.5.1 中规定的相应 EDD 元素描述。参数保存设备的长（5 字节）地址。该值可通过方法扫描获得。
ServiceError	0: OK / 执行结束，已成功建立连接 -1: Connect Failed/被呼叫方取消 -2: Connect Failed/未知的服务 ID -3: Connect Failed/设备未找到 -4: Connect Failed/无效的设备节点地址 -5: Connect Failed/无效的设备标识
注1: IEC 62769-7，将方法Connect的参数AddressData定义为Variant数组。表中定义的LongAddress参数按照上面指定的顺序表示为Variant数组的条目。 注2: IEC 62769-7，将参数DeviceInformation定义为协议特定的参数列表，方法Connect在其中存储结果数据。DeviceInformation参数定义为Variant数组。该DeviceInformation参数不用于CP 9/1设备。	

4.6.1.3 Disconnect

方法Disconnect的参数见表13。

Signature:

```
Disconnect (
    [in]ByteString CommunicationRelationId,
    [out]Int 32ServiceError);
```

表13 方法 Disconnect 的参数

参数	说明
CommunicationRelationId	该参数值包含设备 ConnectionPoint 的 nodeId，代表设备与直接连接到 FDI 通信服务器硬件的物理网络之间的连接。nodeId 允许查找直接的父子关系。
ServiceError	0: OK / 断开连接成功 -1: Disconnect Failed/不存在通信关系 -2: Disconnect Failed/无效的通信关系标识符

4.6.1.4 Transfer

方法Transfer的参数见表14。

Signature

```
Transfer (
  [in]ByteString  CommunicationRelationId,
  [in]UInt16      Command,
  [in]ByteString  Request,
  [out]ByteString Reply,
  [out]Int32      ServiceError);
```

表14 方法 Transfer 的参数

参数	说明
CommunicationRelationId	该参数值包含 ConnectionPoint 的 NodeId, 代表设备与直接连接到 FDI 通信服务器硬件的物理网络之间的连接。 NodeId 允许查找直接的父子关系。
Command	要发送的 CP9/1 命令的命令号。FDI 通信服务器需要根据 CP9/1 协议规范以适当的格式生成和发送 PDU (例如: 负责命令扩展)。
Request	表示 CP9/1 命令请求中的数据字节的字节数组 (例如: 在字节计数之后开始, 在校验和之前结束)。
Reply	表示 CP9/1 命令响应中的数据字节的字节数组 (例如: 在字节计数之后开始, 在校验和之前结束)。
ServiceError	0: OK /执行完毕 -1: Transfer Failed/被呼叫方取消 -2: Call Failed /未知的服务 ID -3: Transfer Failed/不存在通信关系 -4: Transfer Failed/无效的通信关系标识符 -5: Transfer Failed/无效的请求内容 -6: Transfer Failed/无效的应答格式
<p>注1: IEC 62769-7将传输方法的SendData参数定义为Variant数组。表中定义的参数Command和Request按照上面指定的顺序表示为Variant数组的项。</p> <p>注2: IEC 62769-7将传输方法的ReceiveData参数定义为Variant数组。表中定义的参数Reply表示为Variant数组的项。</p>	

4.6.1.5 GetPublishedData

CP 9/1突发事件和事件通知消息表示在 IEC 62769-7中定义的未经请求的消息。方法GetPublishedData的参数见表15。

Signature:

```

GetPublishedData(
    [in] ByteString CommunicationRelationId,
    [out] UInt16 Command,
    [out] ByteString Reply,
    [out] DateTime TimeStamp,
    [out] Int32 ServiceError);

```

表15 方法 GetPublishedData 的参数

参数	说明
CommunicationRelationId	该参数值包含 ConnectionPoint 的 NodeId, 代表在信息模型中设备和物理网络之间的连接。
Command	要发送的 CP9/1 命令的命令号. FDI 通信服务器需要根据 CP9/1 协议标准解析各种格式的 PDU(例如: 处理命令扩展).
Reply	表示已发布的 CP9/1 命令响应中的数据字节的字节数组(例如: 在字节计数之后开始, 在校验和之前结束)。
TimeStamp	捕获已发布数据的时间。
注: ServiceError	0: OK /执行结束, REPLY 包含结果 -1: GetPublishedData Failed/被呼叫方取消 -2: Call Failed/未知的服务 ID -3: GetPublishedData Failed/不支持 -4: GetPublishedData Failed/不存在的通信关系 -5: GetPublishedData Failed/无效的通信关系标识符 -8: GetPublishedData Failed/没有发布的突发/事件数据
注: IEC 62769-7将GetPublishedData方法的ReceiveData参数定义为Variant数组. 表中定义的参数Command和Reply按照上面指定的顺序表示为Variant数组的条目。	

4.6.1.6 SetAddress

方法SetAddress的参数见表16。

Signature

```

SetAddress (
    [in] UInt8 OldPollAddress,
    [in] UInt8 NewPollAddress,
    [out] Int32 ServiceError);

```

表16 方法 SetAddress 的参数

参数	说明
OldPollAddress	该参数值保存设备的当前地址。允许值为 0..63。
NewPollAddress	该参数值保存设备的新地址。允许值为 0..63。
ServiceError	0: OK /执行成功结束 -1: SetAddress Failed/被呼叫方取消 -2: Call Failed/未知的服务 ID -3: SetAddress Failed/未初始化 -4: SetAddress Failed/未连接到网络 -5: SetAddress Failed/找不到响应 OldAddress 的设备 -6: SetAddress Failed/重复地址错误 -7: SetAddress Failed/设备不接受新地址 -8: SetAddress Failed/无效的 OldAddress (在语法、数据类型、数据格式等方面) -9: SetAddress Failed/无效的 NewAddress 在语法、数据类型、数据格式等方面) -10: SetAddress Failed/连接状态下不可能
注1: IEC 62769-7将SetAddress方法的OldAddress参数定义为Variant数组。表中定义的OldPollAddress参数表示为Variant数组的项。 注2: IEC 62769-7将SetAddress方法的NewAddress参数定义为Variant数组。表中定义的NewPollAddress参数表示为Variant数组的项。	

4.6.1.7 Scan

IEC 62769-7中规定的方法签名适用。附录A中规定了相应的TopologyScanResult模式。

4.6.1.8 ResetScan

IEC 62769-7中规定的方法签名适用。附录A中规定了相应的TopologyScanResult模式。

4.6.2 网关的方法

4.6.2.1 概述

4.6.2中描述的方法签名应在包含通信设备定义的与网关关联的FDI包中包含的 EDD元素（IEC 62769-4）中实现。

4.6.2.2 Connect

方法Connect的参数见表17。

Signature:

```

METHOD BeginConnect (
  DD_String      CommunicationRelationId,
  DD_String      LongAddress,
  unsigned long  ServiceID,
  unsigned long  &DelayForNextCall,
  long           &ServiceError)
{
  DEFINITION {<Gateway specific implementation>}
}

```

```

METHOD EndConnect (
  DD_String      CommunicationRelationId,
  unsigned long  ServiceID,
  unsigned long  &DelayForNextCall,
  long           &ServiceError);
{
  DEFINITION {<Gateway specific implementation>}
}

```

```

METHOD CancelConnect (
  DD_String      CommunicationRelationId,
  unsigned long  ServiceID,
  long           &ServiceError);
{
  DEFINITION {<Gateway specific implementation>}
}

```

表17 方法 Connect 的参数

参数	说明
CommunicationRelationId	该参数值包含 ConnectionPoint 的 nodeId, 代表设备与直接连接到 FDI 通信服务器硬件的物理网络之间的连接。nodeId 允许查找直接的父子关系。
LongAddress	该参数名应与为 ConnectionPoint 定义的相应特性名匹配, 该特性名由 5.5.1 中指定的相应 EDD 元素描述。参数保存设备的长(5 字节)地址。该值可通过方法 Scan 获得。
ServiceId	服务事务代码建立服务请求和相应的响应之间的关系。
DelayForNextCall	该值以毫秒为单位指定延迟时间, 以限制 EndConnect 调用周期, 该周期不得超过参数值中指定的时间。
ServiceError	1: OK /函数异步启动, 必须使用 EndConnect 对结果轮询 0: OK /执行完毕, 已成功建立连接 -1: Connect Failed/被呼叫方取消 -2: Call Failed/未知的服务 ID -3: Connect Failed/设备未找到

	-4: Connect Failed/无效的设备地址
	-5: Connect Failed/无效的设备标识

4.6.2.3 Disconnect

方法Disconnect的参数见表18。

Signature:

```
METHOD Disconnect (
  DD_String  communicationRelationId,
  Long      &serviceError)
{
  DEFINITION {<Gateway specific implementation>}
}
```

表18 方法Disconnect的参数

参数	说明
CommunicationRelationId	该参数值包含 ConnectionPoint 的 nodeId, 代表设备与直接连接到 FDI 通信服务器硬件的物理网络之间的连接。nodeId 允许查找直接的父子关系。
ServiceError	0: OK /Disconnect 成功结束 -1: Disconnect Failed/不存在的通信关系 -2: Disconnect Failed/无效的通信关系标识符

4.6.2.4 Transfer

方法Transfer的参数见表19。

Signature:

```

METHOD BeginTransfer (
  DD_String      CommunicationRelationId,
  unsigned short Command,
  DD_String      Request,
  DD_String      &Reply,
  unsigned long  ServiceId,
  unsigned long  &DelayForNextCall,
  long           &ServiceError);
{
  DEFINITION {<Gateway specific implementation>}
}

```

```

METHOD EndTransfer (
  DD_String      CommunicationRelationId,
  DD_String      &Reply,
  unsigned long  ServiceId,
  unsigned long  &DelayForNextCall,
  long           &ServiceError);
{
  DEFINITION {<Gateway specific implementation>}
}

```

```

METHOD CancelTransfer (
  DD_String      CommunicationRelationId,
  unsigned long  ServiceId,
  long           &ServiceError);
{
  DEFINITION {<Gateway specific implementation>}
}

```

表19 方法 Transfer 的参数

参数	说明
CommunicationRelationId	该参数值包含 ConnectionPoint 的 nodeId, 代表设备与直接连接到 FDI 通信服务器硬件的物理网络之间的连接。nodeId 允许查找直接的父子关系。
Command	要发送的 CP9/1 命令的命令号。 FDI 通信服务器需要根据 CP9/1 协议规范以适当的格式生成和发送 PDU (例如: 处理命令扩展)。
Request	表示 CP9/1 命令请求中的数据字节的字节数组 (例如: 在字节计数之后开始, 在校验和之前结束)。
Reply	表示 CP9/1 命令响应中的数据字节的字节数组 (例如: 在字节计数之后开始, 在校验和之前结束)。
ServiceId	服务事务代码建立服务请求和相应的响应之间的关系。
DelayForNextCall	该值以毫秒为单位指定延迟时间, 以限制 EndTransfer 调用周期, 该周

	期不得超过参数值中指定的时间。
ServiceError	<p>1: OK /函数异步启动, 必须使用 EndTransfer 对结果轮询</p> <p>0: OK /执行结束, REPLY 包含结果</p> <p>-1: Transfer Failed /被呼叫方取消</p> <p>-2: Call Failed /未知的服务 ID</p> <p>-3: Transfer Failed/不存在的的通信关系</p> <p>-4: Transfer Failed/无效的通信关系标识符</p> <p>-5: Transfer Failed/ 无效的 REQUEST 内容</p> <p>-6: Transfer Failed/ 无效的 REPLY 格式</p>

4.6.2.5 GetPublishedData

CP9/1突发和事件通知消息表示IEC 62769-7定义的未经请求的消息。方法GetPublishedData的参数见表20。

Signature

```

METHOD BeginGetPublishedData (
  DD_String      CommunicationRelationId,
  unsigned short &Command,
  DD_String      &Reply,
  TIME           &TimeStamp
  unsigned long  ServiceId,
  unsigned long  &DelayForNextCall,
  long           &ServiceError);
{
  DEFINITION {<Gateway specific implementation>}
}

```

```

METHOD EndGetPublishedData (
  DD_String      CommunicationRelationId,
  unsigned short &Command,
  DD_STRING      &Reply,
  TIME           &TimeStamp
  unsigned long  ServiceId,
  unsigned long  &DelayForNextCall,
  long           &ServiceError);
{
  DEFINITION {<Gateway specific implementation>}
}

```

```

METHOD CancelGetPublishedData (
  DD_String      CommunicationRelationId,
  unsigned long  ServiceId,
  long           &ServiceError);
{
  DEFINITION {<Gateway specific implementation>}
}

```

表20 方法 GetPublishedData 的参数

参数	说明
CommunicationRelationId	该参数值包含 ConnectionPoint 的 NodeId, 表示信息模型中设备和物理网络之间的连接。
Command	发布的 CP9/1 命令的命令号。 FDI 通信服务器需要根据 CP9/1 协议标准以各种格式解析 PDU (例如: 处理命令扩展)。
Reply	表示已发布的 CP9/1 命令响应中的数据字节的字节数组 (例如: 在字节计数之后开始, 在校验和之前结束)。
TimeStamp	捕获的已发布数据的时间。
ServiceId	服务事务代码建立服务请求和相应的响应之间的关系。
DelayForNextCall	该值以毫秒为单位指定延迟时间, 以限制 EndGetPublishedData 调用周期, 其周期不得超过参数值中指定的时间。

ServiceError	1: OK /函数异步启动, 必须使用 EndGetPublishedData 对结果轮询 0: OK /执行结束, ReceivedData 和 TimeStamp 包含结果 -1: GetPublishedData Failed/被呼叫方取消 -2: Call Failed/未知的服务 ID -3: GetPublishedData Failed/不支持 -4: GetPublishedData Failed/不存在的通信关系 -5: GetPublishedData Failed/无效的通信关系标识符 -8: GetPublishedData Failed/未发布突发信息
--------------	---

4.6.2.6 SetAddress

方法SetAddress的参数见表21。

Signature

```

METHOD BeginSetAddress (
    unsigned char  OldPollAddress,
    unsigned char  NewPollAddress,
    unsigned long  ServiceId,
    unsigned long  &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

```

```

METHOD EndSetAddress (
    unsigned long  ServiceId,
    unsigned long  &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

```

```

METHOD BeginSetAddress (
    unsigned long  ServiceId,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

```

表21 方法 SetAddress 的参数

参数	说明
OldPollAddress	参数值保存设备的当前地址。允许的值为 0..63。
NewPollAddress	参数值保存设备的新地址。允许的值为 0..63。

ServiceId	服务事务代码建立服务请求和相应的响应之间的关系。
DelayForNextCall	该值以毫秒为单位指定延迟时间，以限制 EndTransfer 调用周期，其周期不得超过参数值中指定的时间。
ServiceError	<p>1: OK /函数异步启动，必须使用 EndSetAddress 对结果轮询</p> <p>0: OK /执行成功结束</p> <p>-1: SetAddress Failed/被呼叫方取消</p> <p>-2: Call Failed/未知的服务 ID</p> <p>-3: SetAddress Failed/未初始化</p> <p>-4: SetAddress Failed/未连接到网络</p> <p>-5: SetAddress Failed/找不到响应 OldAddress 的设备</p> <p>-6: SetAddress Failed/重复地址错误</p> <p>-7: SetAddress Failed/设备不接受新地址</p> <p>-8: SetAddress Failed/无效的 OldAddress (在语法、数据类型、数据格式等方面)</p> <p>-9: SetAddress Failed/无效的 NewAddress (在语法、数据类型、数据格式等方面)</p> <p>-10: SetAddress Failed/连接状态下不可能</p>

4.6.2.7 Scan

IEC 62769-7中规定的方法签名适用。HART网关业务逻辑应使用以下EDD定义创建扫描结果，该定义应包含在通信设备COMPONENT定义中。在COMPONENT中的EDD LIST也包含与XML模式中定义的数据结构相对应的设备列表。此LIST应通过通信设备组件的SCAN_LIST属性来引用。附录A中规定了相应的TopologyScanResult模式

```
VARIABLE DevAddr
{
  LABEL "Address";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(5);
  HANDLING READ & WRITE;
}

VARIABLE DevMfg
{
  LABEL "Manufacturer";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(2);
  HANDLING READ & WRITE;
}
```

```

VARIABLE DevType
{
  LABEL "Device Type";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(2);
  HANDLING READ & WRITE;
}

VARIABLE DevRev
{
  LABEL "Device Revision";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER;
  HANDLING READ & WRITE;
}

VARIABLE DevTag
{
  LABEL "Tag";
  CLASS DEVICE;
  TYPE ASCII(32);
  HANDLING READ & WRITE;
}

COLLECTION ScanItemType_TP
{
  MEMBERS
  {
    ADDRESS, DevAddr;
  }
}

COLLECTION DeviceIdentificationType
{
  MEMBERS
  {
    MFG,      DevMfg;
    DEV_TYPE, DevType;
    DEV_REV,  DevRev;
    TAG,      DevTag;
  }
}

COLLECTION ScanResultType
{
  MEMBERS
  {
    ScanItem,          ScanItemType_TP;
    DeviceIdentification, DeviceIdentificationType;
  }
}

LIST TopologyScanResult
{
  TYPE ScanResultType;
  CAPACITY 64; //Defined to limitation of the comm. device
}

```


}

4.6.2.8 ScanNext

IEC 62769-7中规定的方法签名适用。HART网关业务逻辑应使用4.6.2.7中定义的不同EDD定义创建扫描结果。附录A中规定了相应的TopologyScanResult模式。

附录 A

(规范性)

拓扑扫描模式

A.1 概述

附录A中规定的拓扑扫描结果模式描述了HART特定格式的方法扫描参数TopologyScanResult。XML文档内容和结构应与描述拓扑的信息模型设计概念相对应，以便使连接到网络的物理设备和FDI服务器托管的信息模型之间实现通用匹配。

A.2 IdentificationType

元素内容对应于“FunctionalGroup 标识”。IdentificationType 的 XML 模式是：

```

<xs:complexType name="IdentificationT">
  <xs:attribute name="MANUFACTURER_ID" type="xs:unsignedShort"
use="required">
  </xs:attribute>
  <xs:attribute name="DEVICE_TYPE" type="xs:unsignedShort"
use="required">
  </xs:attribute>
  <xs:attribute name="UNIVERSAL_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="DEVICE_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="SERIAL_NUMBER" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="16777216"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="HARDWARE_REVISION" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedByte">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="32"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="SOFTWARE_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="REV_COUNTER" type="xs:unsignedShort" default="0">
  </xs:attribute>
  <xs:attribute name="TAG" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="32"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

```

表A.1描述了IdentificationType的属性。

表A.1 IdentificationT 的属性

属性	说明
MANUFACTURER_ID	制造商标识号。 对于 HART 通用修订版本 7 或更高版本的设备： 设备的 2 字节制造商代码，以命令 0、命令 11 或命令 21 的 字节 17 和 18 返回。 对于 HART 修订版本 6 或较低版本的设备： 最高有效字节应固定为 0，最低有效字节在命令 0 或命令 11

	或命令 21 的字节 1 中返回。
DEVICE_TYPE	与资源关联的制造商型号。 对于 HART 通用修订版本 7 或更高版本的设备： 2 字节设备类型以命令 0、命令 11 或命令 21 的字节 1 和 2 中返回。 对于 HART 修订版本 6 或较低版本的设备： 最高有效字节应固定为 0，最低有效字节在命令 0 或命令 11 或命令 21 的字节 2 中返回。
UNIVERSAL_REVISION	与设备关联的通用 HART 修订版。 在命令 0 或命令 11 或命令 21 的字节 4 中返回。
DEVICE_REVISION	与设备关联的制造商设备修订版本号。 在命令 0 或命令 11 或命令 21 的字节 5 中返回
SERIAL_NUMBER	设备的唯一序列号。设备的 3 字节唯一 ID。 以命令 0、命令 11 或命令 21 的字节 9 到 11 返回。
HARDWARE_REVISION	与设备关联的制造商硬件修订版本。 在命令 0、命令 11 或命令 21 的字节 7（仅高 5 位）中返回。
SOFTWARE_REVISION	与设备关联的制造商固件修订版本。 在命令 0 或命令 11 或命令 21 的字节 6 中返回
REV_COUNTER	与设备关联的配置更改计数器。 对用于 HART 通用修订版本 6 或更高版本的设备： 以命令 0、命令 11 或命令 21 的字节 14 和 15 返回。 对于 HART 修订版本 5 或更低版本的设备： -1（未定义）。
TAG	与设备关联的已配置标识位号。 对用于 HART 通用修订版本 6 或更高版本的设备： 在命令 20 或命令 21 中返回的 32 个字符长位号。 对于 HART 修订版本 5 或更低版本的设备： 命令 11 或命令 13 中返回 8 字符位号。

A.3 AddressTypeTP

AddressTypeTP 的 XML 模式是：

```
<xs:complexType name="AddressTypeTP">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:element name="DevPollAddr" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="63"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

```

表 A.2 描述了 AddressTypeTP 的元素。

表A.2 AddressTypeTP 的元素

元素	说明
DevAddr	保存关联设备的 5 字节 HART 地址。
DevPollAddr	保存用于轮询关联设备的 HART 短地址。

A.4 AddressTypeIP

AddressTypeIP的XML模式为:

```

<xs:simpleType name="IPv4T">
  <xs:restriction base="xs:string">
    <xs:pattern
      value="((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.) {3} (25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)"/>
    </xs:restriction>
  </xs:simpleType>

<xs:simpleType name="IPv6T">
  <xs:restriction base="xs:string">
    <xs:pattern value="([A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-9]{1,4}"/>
    </xs:restriction>
  </xs:simpleType>

<xs:complexType name="AddressTypeIP">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:choice>
          <xs:element name="IPv4Address" type="hart:IPv4T">
          </xs:element>
          <xs:element name="IPv6Address" type="hart:IPv6T">
          </xs:element>
        </xs:choice>
        <xs:element name="IPPort" type="xs:unsignedShort">
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

表A.3描述了AddressTypeIP的元素。

表A.3 AddressTypeIP 的元素

元素	说明
DevAddr	保存关联设备的 5 字节 HART 地址。
IPAddress	保存关联设备的网络 IP 地址。
IPPort	保存用于关联设备的 IP 端口号。

A.5 AddressTypeTDMA

AddressTypeTDMA的XML模式为:

```
<xs:complexType name="AddressTypeTDMA">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:element name="NetworkID" type="xs:unsignedShort">
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

表A.4描述了AddressTypeTDMA的元素。

表A.4 AddressTypeTDMA 的元素

元素	说明
DevAddr	保存关联设备的 5 字节 HART 地址。
IPAddress	保存关联设备的网络 IP 地址。
IPVersion	保存用于关联设备（版本 4 或 6）的 IP 版本号。
IPPort	保存用于关联设备的 IP 端口号。

A.6 AddressType

AddressType 的 XML 模式为:

```
<xs:complexType name="AddressT">
  <xs:choice>
    <xs:element name="AddressTP" type="hart:AddressTypeTP">
    </xs:element>
    <xs:element name="AddressIP" type="hart:AddressTypeIP">
    </xs:element>
    <xs:element name="AddressTDMA" type="hart:AddressTypeTDMA">
    </xs:element>
  </xs:choice>
</xs:complexType>

<xs:complexType name="LongAddressT">
  <xs:sequence>
```

```

<xs:element name="DevAddr">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="([A-Fa-f0-9]{10})"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

表 A.5 描述了 AddressType 的元素。

表A.5 AddressT 的元素

元素	说明
AddressTP	令牌传递地址保存用于 4.5.1.2 中描述的 ConnectionPointType_HART_TP5, 6, 或 7 的地址信息。
AddressIP	IP 地址保存 4.5.1.3 中描述的 ConnectionPointType_HART_IP 中描述的地址信息。
AddressTDMA	TDMA 地址保存 4.5.1.3 中描述的 ConnectionPointType_HART_TDMA 中描述的地址信息。

A.7 ConnectionPointType

ConnectionPointType 的 XML 模式是:

```

<xs:complexType name="ConnectionPointT">
  <xs:sequence>
    <xs:element name="Identification" type="hart:IdentificationT">
    </xs:element>
    <xs:element name="Address" type="hart:AddressT">
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

表 A.6 中描述了 ConnectionPointType 的元素。

表A.6 ConnectionPointT 的元素

元素	说明
Identification	元素数据保存设备类型标识数据。与信息模型 (IEC 62769-5) 相比, ConnectionPoint 不包含或引用设备类型标识数据。但是, 为了支持 FDI 主机系统查找与所连接设备匹配的包, 此模式将设备类型标识与 ConnectionPoint 关联。
Address	地址元素保存用于唯一寻址网络中设备的特定信息。地址将是 AddressType 描述的三种可能类型之一。

A.8 NetworkType

元素类型描述单个网络的完整扫描结果，因为每个“通信设备”实例提供的扫描方法与网络实例的关系为一一对应的。

NetworkType的XML模式是：

```
<xs:complexType name="NetworkT">
  <xs:sequence>
    <xs:element name="ConnectionPoint" type="hart:ConnectionPointT"
      maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

表 A.7 中描述了 NetworkType 的元素。

表A.7 NetworkT 的元素

元素	说明
ConnectionPoint	ConnectionPoint 元素保存总线扫描操作期间找到的网络连接设备的地址和标识。

A.9 Network

后续元素层次结构用于根据拓扑的信息模型（IEC 62769-5）表示返回扫描结果。

Network 元素的 XML 模式是：

```
<xs:element name="Network" type="hart:NetworkT">
</xs:element>
```


附 录 B (规范性) 传输服务参数

B.1 概述

IEC 62769-2 中规定的直接访问服务，允许用户界面插件（UIP）直接与设备交换数据。直接数据交换意味着设备和 UIP 之间交换的数据可不会反映在信息模型中。IEC 62769-6 定义的接口 IDirectAccess 对应于 IEC 62769-2 规定的直接访问服务。接口 IDirectAccess 定义的函数 BeginTransfer 和 EndTransfer 需要传递协议规定的信息。下面的 XML 模式记录了协议规定。

B.2 receiveData

下面描述的元素包含通过 IDirectAccess 函数 Endtransfer 和 GetPublishedData 定义的返回值返回的数据。

```
<xs:element name="receiveData" type="hart:TransferResultDataT">
</xs:element>
```

B.3 sendData

下面描述的元素包含要通过 IDirectAccess 函数 Begintransfer 定义的参数 sendData 提交的数据。

```
<xs:element name="sendData" type="hart:TransferSendDataT">
</xs:element>
```

B.4 TransferResultDataT

XML 元素类型定义应用于 EndTransfer 和 GetPublishedData 定义的返回值的返回的服务参数数据格式。TransferResultDataType 中定义的属性名与 4.6.1.4 和 4.6.2.4 中指定的 Transfer 方法参数的名称相同。

```
<xs:complexType name="TransferResultDataT">
  <xs:attribute name="COMMAND" type="xs:unsignedInt" use="required">
  </xs:attribute>
  <xs:attribute name="REPLY" type="xs:hexBinary" use="required">
  </xs:attribute>
</xs:complexType>
```

表B.1 TransferResultDataT 的属性

属性	说明
COMMAND	该属性对应于 CP9/1 命令号。
REPLY	该属性对应于 CP9/1 命令数据字段中的应答数据字节。

B.5 TransferSendDataT

XML 元素类型定义服务参数数据格式，该格式应用于 BeginTransfer 定义的 SendData 参数。在 TransferSendDataType 中定义的属性名与 4.6.1.4 和 4.6.2.4 中指定的 Transfer 方法参数的名称相同。

```
<xs:complexType name="TransferSendDataT">
  <xs:attribute name="COMMAND" type="xs:unsignedInt" use="required">
  </xs:attribute>
  <xs:attribute name="REQUEST" type="xs:hexBinary" use="required">
  </xs:attribute>
</xs:complexType>
```

表B.2 TransferSendDataT 的属性

属性	说明
COMMAND	该属性对应于 CP9/1 命令号。
REQUEST	该属性对应于 CP9/1 命令数据字段中的 REQUEST 数据字节。

参 考 文 献

- IEC 61784-1 工业通信网络-行规-第1部分：现场总线行规 (Industrial communication networks - Profiles - Part 1: Fieldbus Profiles)
- IEC 61784-2 工业通信网络-行规-第2部分：基于ISO/IEC/IEEE 8802-3实时网络附加的现场总线行规 (Industrial communication networks - Profiles - Part 2: Additional Fieldbus Profiles for real-time networks based on ISO/IEC/IEEE 8802-3)
- IEC 61804 (all parts) 过程控制和电气设备描述语言(EDDL)的功能块(FB) (Function blocks (FB) for process control and Electronic Device Description Language (EDDL))
- IEC 62769-1 现场设备集成(FDI) 第1部分：概述 (Field device integration (FDI) - Part 1: Overview)
- IEC 62769-2 现场设备集成(FDI) 第2部分：FDI客户端(Field device integration(FDI) - Part 2: FDI Client)
- IEC 62769-3 现场设备集成(FDI) 第3部分：FDI服务器 (Field device integration (FDI) - Part 3: FDI Server)
- IEC 62769-6 现场设备集成(FDI) 第6部分：FDI技术映射 (Field device integration (FDI) - Part 6: FDI Technology Mapping)
- HCF_SPEC-13 HART®现场通信协议规范 (HART® Field Communication Protocol Specification)
- HCF_SPEC-99 命令总结规范 (Command Summary Specification)
- HCF_SPEC-127 通用命令规范 (Universal Command Specification)
- HCF_SPEC-151 通用实施命令规范 (Common Practice Command Specification)
- IETF RFC 768 用户数据表协议 (User Datagram Protocol)
- IETF RFC 793 传输控制协议 (Transmission Control Protocol)
-