



中华人民共和国国家标准

GB/T 41970—2022



智能工厂数控机床互联接口规范

Interconnection interface specification of numerical control machine
tools in smart factory

2022-10-12 发布

2022-10-12 实施

国家市场监督管理总局
国家标准化管理委员会 发布

目 次

前言	VII
引言	VIII
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	4
5 总体要求	4
5.1 数控机床互联接口体系架构	4
5.1.1 组成	4
5.1.2 应用系统层	4
5.1.3 NC-Link 接口层	5
5.1.4 数控机床层	5
5.1.5 连接关系	5
5.2 适配器、代理器和数据传输接口功能要求	5
5.2.1 适配器要求	5
5.2.2 代理器要求	5
5.2.3 数据传输接口功能要求	5
5.3 数控机床要求	5
5.4 接口使用的通用技术要求	6
5.5 接口的信息交互要求	6
5.5.1 概述	6
5.5.2 上行信息交互	6
5.5.3 下行信息交互	7
5.5.4 信息交互的一般要求	7
5.5.5 适配器与代理器的交互要求	7
5.5.6 适配器与数控机床交互要求	7
5.5.7 应用系统与代理器的交互要求	7
6 数控机床模型定义	8
6.1 模型组成	8
6.2 对象定义	9
6.2.1 数据类型	9
6.2.2 根对象	9
6.2.3 设备对象	10

6.2.4	组件对象	11
6.2.5	数据对象	12
6.2.6	采样通道对象	13
6.2.7	内构方法对象	14
7	数据项定义	15
7.1	概述	15
7.2	设备对象的数据项	15
7.3	组件对象的数据项	15
7.4	数据对象的数据项	16
8	数据传输接口定义	23
8.1	数据传输接口概述	23
8.1.1	通信方式	23
8.1.2	数据传输接口明细表	24
8.1.3	消息标识符	28
8.1.4	终端标识符	28
8.1.5	数据类型的访问	28
8.1.6	数据传输单元	28
8.1.7	状态码及错误码定义	29
8.2	数据传输接口定义	32
8.2.1	注册请求数据传输接口	32
8.2.2	注册响应数据传输接口	33
8.2.3	终端探测请求数据传输接口	34
8.2.4	终端探测响应数据传输接口	36
8.2.5	版本号校对请求数据传输接口	38
8.2.6	版本号校对响应数据传输接口	39
8.2.7	模型侦测请求数据传输接口	40
8.2.8	模型侦测响应数据传输接口	41
8.2.9	模型设置请求数据传输接口	42
8.2.10	模型设置响应数据传输接口	42
8.2.11	数据查询请求数据传输接口	43
8.2.12	数据查询响应数据传输接口	49
8.2.13	数据设置请求数据传输接口	55
8.2.14	数据设置响应数据传输接口	60
8.2.15	数据采样数据传输接口	63
8.2.16	状态通知数据传输接口	64
8.2.17	动态采样注册请求数据传输接口	65
8.2.18	动态采样注册响应数据传输接口	67

8.2.19	动态采样注销请求数据传输接口	68
8.2.20	动态采样注销响应数据传输接口	69
8.2.21	动态数据采样数据传输接口	70
8.2.22	内构方法调用数据传输接口	71
8.2.23	内构方法进度数据传输接口	72
8.2.24	内构方法结果数据传输接口	73
8.2.25	内构方法控制请求数据传输接口	75
8.2.26	内构方法控制响应数据传输接口	76
8.2.27	事件注册请求数据传输接口	77
8.2.28	事件注册响应数据传输接口	80
8.2.29	事件注销请求数据传输接口	81
8.2.30	事件注销响应数据传输接口	82
8.2.31	事件数据数据传输接口	83
8.3	网络连接错误场景	85
8.3.1	概述	85
8.3.2	连接错误情景一	85
8.3.3	连接错误情景二	86
8.3.4	连接错误情景三	87
8.3.5	连接错误情景四	89
8.3.6	连接错误情景五	91
9	测试与评价	93
9.1	测试要求	93
9.1.1	测试目的	93
9.1.2	测试系统组成	93
9.1.3	测试对象的连接	94
9.2	测试内容	94
9.2.1	连接测试	94
9.2.2	测试用例	94
9.2.3	模型侦测与模型设置测试	94
9.2.4	数据查询与数据设置测试	95
9.2.5	数据采样测试	95
9.2.6	事件功能测试	96
9.2.7	动态采样功能测试	96
9.2.8	状态通知测试	96
9.2.9	内构方法功能测试	96
9.3	测试结果评价	97
9.3.1	连接测试	97

9.3.2	模型文件规范性检查	97
9.3.3	适配器和代理器功能测试	97
9.4	测试记录	98
附录 A	(资料性) 数控机床模型样式文件示例	99
附录 B	(资料性) 数控机床模型描述示例	138
附录 C	(资料性) 测试记录表	148
图 1	NC-Link 接口体系架构	4
图 2	数控机床模型对象构成及各对象间关系	8
图 3	数据传输接口“请求/响应”通信方式	23
图 4	数据传输接口“订阅/发布”通信方式	24
图 5	适配器终端探测流程	35
图 6	单值类型数据项查询请求参数形式化定义	45
图 7	字典类型数据项查询请求参数形式化定义	46
图 8	列表类型数据项查询请求参数形式化定义	47
图 9	单值类型数据项查询响应参数形式化定义	50
图 10	字典类型数据项查询响应参数形式化定义	51
图 11	列表类型数据项查询响应参数形式化定义	52
图 12	单值类型数据项设置请求参数形式化定义	56
图 13	字典类型数据项设置请求参数形式化定义	57
图 14	列表类型数据项设置请求参数形式化定义	58
图 15	数据项设置响应参数形式化定义	61
图 16	过滤器参数形式化定义	79
图 17	事件体数据参数形式化定义	84
图 18	连接错误情景一	85
图 19	连接错误情景二	86
图 20	连接错误情景三	88
图 21	连接错误情景四	90
图 22	连接错误情景五	92
图 23	测试系统和测试对象连接方式	93
图 B.1	数控机床逻辑模型	139
表 1	根对象属性列表	9
表 2	设备对象属性列表	10
表 3	组件对象属性列表	11
表 4	数据对象属性列表	12
表 5	采样通道对象属性列表	13

表 6	内构方法对象属性列表	14
表 7	设备对象的数据项	15
表 8	组件对象的数据项	15
表 9	组件对象的 type 属性的部分组合定义	16
表 10	物理量数据项	16
表 11	属性 units 常用的非默认数据单位	17
表 12	通用数据项	18
表 13	专用数据项	19
表 14	数据对象 type 属性取特定值时 value 属性的定义	21
表 15	数据对象 type 属性取特定值时仅使用 value 属性的组合定义	22
表 16	数据对象 type 属性取特定值时 value 属性和设备/组件对象的 type 属性的组合定义	22
表 17	数据传输接口指令列表	24
表 18	请求执行状态码取值	29
表 19	错误原因列表	29
表 20	状态码、错误原因及错误码在数据传输单元中的规范性定义	32
表 21	注册请求数据传输单元定义	32
表 22	注册响应数据传输单元定义	33
表 23	终端探测请求数据传输单元定义	35
表 24	终端探测响应数据传输单元定义	36
表 25	适配器描述	37
表 26	版本号校对请求数据传输单元定义	38
表 27	版本号校对响应的数据传输单元定义	39
表 28	模型侦测请求数据传输单元定义	40
表 29	模型侦测响应数据传输单元定义	41
表 30	模型设置请求数据传输单元定义	42
表 31	模型设置响应数据传输单元定义	43
表 32	数据查询请求数据传输单元定义	44
表 33	数据查询响应数据传输单元定义	49
表 34	数据设置请求数据传输单元定义	55
表 35	数据设置响应数据传输单元定义	61
表 36	数据采样数据传输单元定义	63
表 37	状态通知数据传输单元定义	65
表 38	动态采样注册请求数据传输单元定义	66
表 39	动态采样注册响应数据传输单元定义	67
表 40	动态采样注销请求数据传输单元定义	68
表 41	动态采样注销响应数据传输单元定义	69
表 42	动态数据采样数据传输单元定义	70

表 43	内构方法调用数据传输单元定义	72
表 44	内构方法进度数据传输单元定义	73
表 45	内构方法结果数据传输单元定义	74
表 46	内构方法控制请求数据传输单元定义	75
表 47	内构方法控制响应数据传输单元定义	76
表 48	事件注册请求数据传输单元定义	77
表 49	事件注册响应数据传输单元定义	80
表 50	事件注销数据传输单元定义	81
表 51	事件注销响应数据传输单元定义	82
表 52	事件数据数据传输单元定义	83
表 53	连接错误情景三错误状态的数据传输单元	88
表 54	错误情景四错误状态的数据传输单元	91
表 55	连接测试项目	94
表 56	模型侦测与模型设置测试项目	95
表 57	数据查询与数据设置测试项目	95
表 58	数据采样测试项目	95
表 59	事件功能测试项目	96
表 60	动态采样功能测试项目	96
表 61	状态通知测试项目	96
表 62	内构方法功能测试项目	97
表 63	适配器和代理器功能测试评价项目	97
表 C.1	测试记录表	148

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》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国机械工业联合会提出。

本文件由全国机床数控系统标准化技术委员会(SAC/TC 367)归口。

本文件起草单位：华中科技大学、武汉华中数控股份有限公司、中国机床工具工业协会、沈机(上海)智能系统研发设计有限公司、北京航空航天大学、华工法利莱切焊系统工程股份有限公司、中南民族大学、智能云科信息科技有限公司、广州数控设备有限公司、科德数控股份有限公司、嘉泰数控科技有限公司、北京兰光创新科技有限公司、华为技术有限公司、艾普工华科技(武汉)有限公司、联想(北京)有限公司、中国电子科技网络信息安全有限公司、北京亚鸿世纪科技发展有限公司。

本文件主要起草人：路松峰、陈吉红、金健、朱建新、汤学明、惠恩明、王萧、张航军、路直、胡瑞琳、黄云鹰、彭翀、高连生、江哲夫、赵振宇、徐科、张晓、唐建锐、王声文、李敏之、朱铎先、周亚灵、杜俊志、于辰涛、刘波、肖远军、唐威。



引 言

随着信息技术的快速发展,传统制造业开始向智能制造过渡,智能制造的关键在于数控机床的信息交互。但是数控机床属于多源异构系统,具有很大的差异性,给信息的交互带来困难,同时也形成了上层应用系统与数控机床交互的瓶颈。为此,智能工厂需要统一规范的数控机床互联协议,其核心是互联的模型和接口规范。

目前国内外提出了一些数控装备互联标准或协议,但这些标准和协议尚不能满足数字孪生需要的数据要求,为此依据国内工业互联的需求和发展趋势,应制定满足智能工厂数控机床互联的相关接口规范。

智能工厂数控机床互联接口规范(简称 NC-Link)正是在此背景下提出并制定的。NC-Link 用于连接数控机床与应用系统,将数控机床属性数据、参数以及运行信息传递至应用系统或将应用系统的控制信息传递至数控机床,从而实现信息交互。此外,智能工厂的数控机床在互联时也需要采集和使用其他测量、物流等数控机床附属设备的数据,因此智能工厂的数控机床互联接口也会涵盖这些附属设备的接口。

智能工厂数控机床互联接口规范

1 范围

本文件规定了智能工厂数控机床互联接口规范(以下简称 NC-Link),包括总体要求、数控机床模型定义、数据项定义、数据传输接口定义及测试与评价。

本文件适用于数控机床及其附属设备。智能工厂中其他数字化设备可参照使用。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;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。

GB/T 25069—2010 信息安全技术 术语

ECMA-404:2017 JSON 数据交互语法 (The JSON Data Interchange Syntax)

3 术语和定义

GB/T 25069—2010 界定的以及下列术语和定义适用于本文件。

3.1

应用系统 application system

对数控机床的属性数据、参数数据及运行数据进行分析、管理和监控或对数控机床进行控制的软硬件系统。

注:常见的应用系统有 ERP、MES 等。

3.2

适配器 adapter

软件组件或硬件设备,用于连接代理器和数控机床,实现从数控装备采集其属性数据、参数数据或运行数据并进行格式解析和转换,以统一格式传送到代理器,或接收从代理器传来的控制信息,按照指定的数控机床要求转换并传递到对应的数控机床。

注:由于数控机床与适配器为一对一对应,本文件中使用适配器来代表数控机床。

3.3

代理器 agent

软件组件或硬件设备,用于连接适配器和应用系统,实现从适配器采集数据并分发到应用系统,或接收从应用系统传来的控制信息并下载到适配器。

3.4

终端 terminal

应用系统或连接了适配器的数控机床。

注:终端由系统内唯一的终端标识符来标识。

3.5

数据传输接口 data transmission interface

应用系统与代理器、适配器与代理器之间的通信规则

注：该规则包括请求/应答/发布等指令的名称、格式，以及指令的各项参数。

3.6

数控机床模型 numerical control machine model

采用树状的面向对象方式对数控机床的结构、属性和数据进行描述的结果。

注 1：其表现形式为数控机床样式文件。

注 2：具体数控机床的模型是依据当前数控机床的实际特点裁剪而得到的，称为数控机床模型实例，其表现形式为数控机床模型文件。

3.7

上行 upload

数据由数控机床经适配器、代理器，传输至应用系统。

3.8

下行 download

数据由应用系统经过代理器、适配器传输至数控机床。

3.9

运行信息 running information

数控机床运行时产生的电流、位置等数据。

注：由于这些数据包含了语义信息，也称运行数据。出于使用习惯，不同场合可能会称“运行数据”或“运行信息”。

3.10

样式文件 schema file

基于《IETF JSON 框架草案 01, JSON 框架:描述 JSON 文档的媒体类型》语法的，描述所定义对象语法规则的文件。

注：样式文件也称 Schema 文件或数控机床样式文件。

3.11

模型文件 model file

符合样式文件描述的数控机床模型文件。

注：模型文件内容为 JSON 对象。也称 Probe 文件。

3.12

数据项 data item

数控机床模型定义中各对象的属性在使用中规定的取值组合单元。

3.13

单值数据类型 single value data type

一个有效的 JSON 数据类型。

注：单值数据类型包括数值 value、字符串 string、数字 number、对象 object、数组 array、布尔 Boolean、NULL。

3.14

键值对 key-value pair

数据库存储最简单的组织形式。“键”为存储的数据的编号，“值”为存储的数据。

3.15

字典数据类型 dictionary data type

一个 JSON 对象，是键值对的集合。

3.16

列表数据类型 list data type

一个JSON数组。

注：列表中的每一个元素可是字典数据类型、单值数据类型或列表数据类型。

3.17

设备 device

数控机床或其附属设备。

3.18

数据传输单元 data transfer unit;DTU

应用系统、代理器和适配器三者之间交互的数据单元。

注：数据传输单元分为请求数据传输单元(Request DTU)、响应数据传输单元(Response DTU)和采样数据传输单元(Sample DTU)

3.19

请求/响应 request/response

应用系统、代理器、适配器三者之间的通信方式。该通信方式下,应用系统通过代理器向适配器发出“Request DTU”,适配器收到该请求数据传输单元后,通过代理器,向应用系统返回相应的“Response DTU”。

3.20

订阅/发布 subscribe/publish

应用系统、代理器、适配器三者之间的通信方式。通过订阅/发布通信模式,三者之间传递数据传输单元,完成应用系统与适配器之间的信息传输或控制功能。

3.21

状态通知 state notify

用于终端得知其他终端状态发生变更的机制。代理器在检测到终端的连接状态发生变更时,向监听该终端状态的终端发送状态变更消息。

3.22

离线 offline

代理器在连续若干时间间隔内,未收到终端发布的任何数据传输单元,则认定该终端当前处于离线状态。

适配器与应用系统通信时,如果在一定时间间隔内未收到该应用系统的数据传输单元,则认定应用系统当前处于离线状态。

在应用系统、代理器和适配器网络连接中,若任意两者之间网络中断,则连接双方互相认定对方当前处于离线状态。

注：时间间隔由应用系统实现者定义。

3.23

在线 online

应用系统在连续若干个时间间隔内,接收到适配器发布的任何数据传输单元,则认定该适配器当前处于在线状态。

适配器与应用系统通信时,如果在一定时间段内接收到应用系统发布的任何数据传输单元,则认定应用系统当前处于在线状态。

注：时间间隔由应用系统实现者定义。

4 缩略语

下列缩略语适用于本文件。

CRM:客户关系管理(Customer Relationship Management)

ERP:企业资源计划(Enterprise Resource Planning)

JSON:JavaScript 对象表示法(JavaScript Object Notation)

MES:制造执行系统(Manufacturing Execution System)

PLM:产品生命周期管理(Product Lifecycle Management)

PDM:产品数据管理(Product Data Management)

SCM:软件配置管理(Software Configuration Management)

5 总体要求

5.1 数控机床互联接口体系架构

5.1.1 组成

NC-Link 接口体系架构由数控机床层、NC-Link 接口层和应用系统层组成(见图 1)。

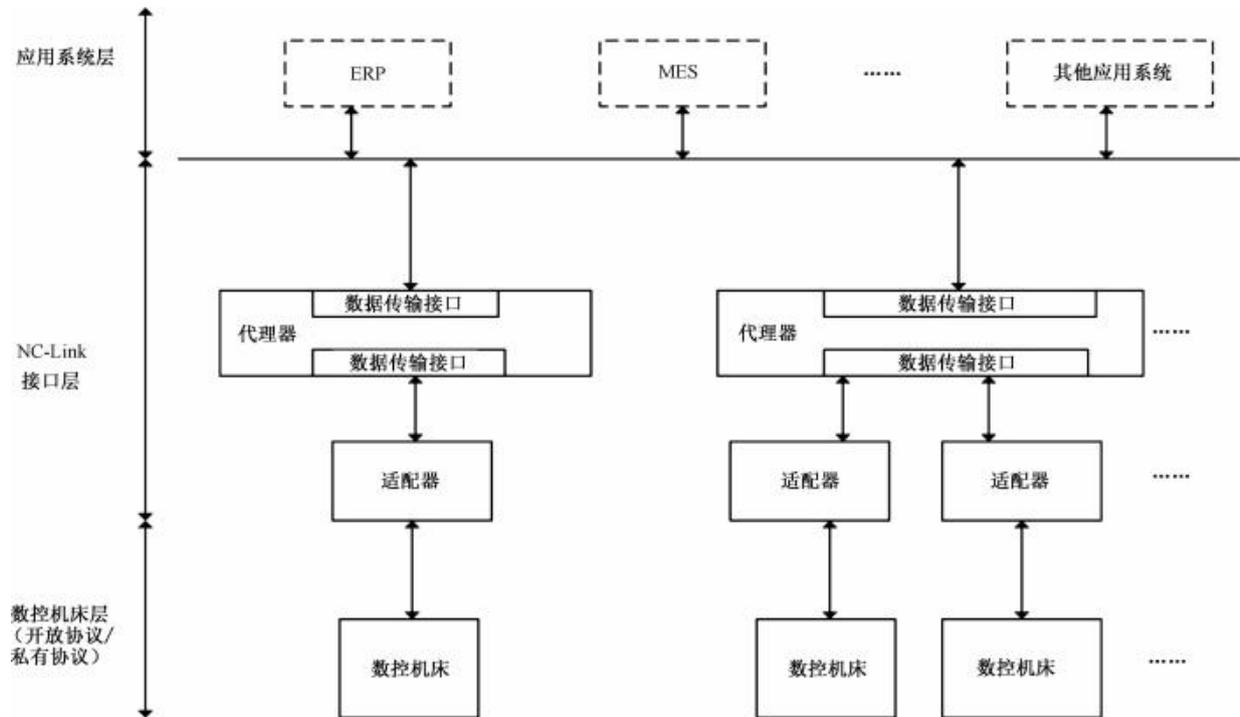


图 1 NC-Link 接口体系架构

5.1.2 应用系统层

应用系统层应为信息应用方和/或控制指令发出方,可是一个或多个应用系统,如 ERP、MES、PLM、PDM、SCM、CRM 等。应用系统层不属于本文件的内容。

5.1.3 NC-Link 接口层

NC-Link 接口层为应用系统层与数控机床层的信息交互中介方,由若干独立的适配器、代理器组成。NC-Link 接口层的信息交互由具体的数据传输接口来实现。

5.1.4 数控机床层

数控机床层为信息提供方和/或受控方,可以是一台或多台数控机床及其附属设备(附属设备属于数控机床的构成部分,不单独运行,未在图 1 中单独标识)。数控机床层不属于本文件的内容。

5.1.5 连接关系

数控机床、适配器、代理器、应用系统之间的连接关系如下:

- a) 数控机床与适配器为一对一连接关系;
- b) 一台适配器仅与一台代理器连接,一台代理器可与一台或多台适配器连接;
- c) 一台代理器可与一个或多个应用系统连接;
- d) 代理器包含与适配器和应用系统数据传输接口。

5.2 适配器、代理器和数据传输接口功能要求

5.2.1 适配器要求

适配器应具有装备驱动、模型解析和数据通信的功能。其中,装备驱动功能应能实现从数控机床中采集数据和将代理器下传的数据传送到数控机床;模型解析功能应能依据数控机床模型实例将适配器要上传的数据进行格式化封装,或把代理器下传的数据依据数控机床模型进行信息映射,即转换成数控机床可识别的指令;数据通信功能应能接收代理器下传的数据,并能把从数控机床采集到的数据经数控机床模型解析后传送到代理器。

注:不同厂家、不同型号的数控机床的适配器的驱动模块可能具有差异性,本文件对数控机床驱动模块与机床控制器之间的数据传输接口不作规定。

适配器应先完成与之连接数控机床的数控机床模型文件的存储,再进行其他信息交互。

适配器可置于数控机床的数控系统内,也可置于数控系统外。

5.2.2 代理器要求

代理器应支持转发应用系统和适配器的订阅请求和发布请求。

代理器应具有缓存应用系统下传数据或适配器上传数据的能力。

注:本文件不规定缓存的大小,缓存大小由具体使用者规定。

代理器应能将数据传到应用系统或适配器。

代理器应能存储向其注册的适配器和应用系统的身份信息。

代理器应能为数控机床和应用系统提供身份认证、访问控制和传输安全服务。

5.2.3 数据传输接口功能要求

数据传输接口功能应包含代理器与应用系统的数据传输接口和代理器与适配器的数据传输接口。数据传输接口应符合第 8 章的规定。

5.3 数控机床要求

数控机床应能对外提供其自身的参数数据、属性数据和运行数据,应能够提供组合数据,组合数据

应符合 6.2.6.2 的规定。

数控机床应能接收来自应用系统的数据。

5.4 接口使用的通用技术要求

基于 NC-Link 的加工单元、车间或工厂等应用场景,其布局可是集中式或分布式,其进行信息交互过程中涉及到的以下环节应分别符合第 6 章、第 7 章、第 8 章、第 9 章的相关要求:

- a) 数控机床模型定义;
- b) 数据项定义;
- c) 数据传输接口定义;
- d) 测试与评价。

5.5 接口的信息交互要求

5.5.1 概述

NC-Link 支持上行和下行两种信息交互方式。

上行信息交互支持“请求/响应”和“订阅/发布”两种方式。

上行或下行信息交互应符合 5.5.4、5.5.5、5.5.6 和 5.5.7 的要求,“请求/响应”和“订阅/发布”模型应符合 8.1.1 的要求。

5.5.2 上行信息交互

5.5.2.1 概述

上行信息交互可用来传递应用系统需要的数据,是数据从适配器到代理器,再到应用系统的过程,分为“请求/响应”和“订阅/发布”两种方式。

5.5.2.2 请求/响应

“请求/响应”可获取数控机床的属性数据、参数数据和运行数据。

“请求/响应”的交互流程一般为:

- a) 应用系统监听代理器的响应,该响应对应于应用系统可能发送的请求;
- b) 适配器监听代理器的请求,该请求为代理器可能发出的请求;
- c) 应用系统向代理器发送请求;
- d) 代理器向适配器发送请求;
- e) 适配器接收请求,进行处理,将结果封装,向代理器发送响应;
- f) 代理器接收适配器发送的响应,向应用系统发送响应;
- g) 应用系统接收响应,获取数据。

5.5.2.3 订阅/发布

“订阅/发布”可获取数控机床连续的属性数据、参数数据和运行数据;使用“订阅/发布”的数据传输接口包括数据采样、动态数据采样、内构方法进度。数据采样和动态数据采样的数据格式应为组合数据,组合数据应由采样通道对象定义,组合数据的格式和采样通道对象的要求应符合 6.2 的规定。内构方法进度数据格式应符合 8.2.23 的规定。

“订阅/发布”的交互流程一般为:

- a) 适配器周期性地向代理器发布数据;

- b) 应用系统向代理器订阅指定数据；
- c) 代理器接收指定数据，并发送给订阅该数据的应用系统；
- d) 应用系统接收订阅数据。

5.5.3 下行信息交互

下行信息交互用来传递数控机床需要的数据，是数据从应用系统到代理器，再到适配器的过程，方式为“请求/响应”，其交互流程一般为：

- a) 应用系统监听代理器的响应，该响应对应于数据下传命令；
- b) 适配器监听代理器的数据下传指令；
- c) 应用系统向代理器发送数据下传指令；
- d) 代理器向适配器发送数据下传指令；
- e) 适配器接收数据下传指令，解析后传送到对应的数控机床；
- f) 适配器将数控机床的执行结果发送给代理器；
- g) 代理器将数控机床的执行结果发送给应用系统。

5.5.4 信息交互的一般要求

应按照第 6 章的规定对每个数控机床建立数控机床模型，并生成数控机床模型文件。

适配器应在信息交互前完成与其对应的数控机床的模型文件的存储。

信息交互前，应用系统应获取需要与其交互的数控机床模型，获取流程如下：

- a) 应用系统向代理器订阅信息模型请求的响应；
- b) 适配器向代理器订阅信息模型请求；
- c) 应用系统向代理器发送信息模型请求；
- d) 代理器向适配器发送信息模型请求；
- e) 适配器接收信息模型请求，向代理器发布信息模型响应，并将其存储的数控机床模型文件发送给代理器；
- f) 代理器接收适配器发布的信息模型响应，获取数控机床模型文件，向应用系统发送信息模型响应，并将数控机床模型文件发送给应用系统；
- g) 应用系统接收信息模型响应，获取数控机床模型文件。

当数控机床模型文件发生更改或版本号变化时，适配器应向代理器发送版本校对消息；代理器收到版本校对消息后应将其转发给订阅该消息的应用系统；应用系统接收到版本校对消息后，应重新按流程获取新的数控机床模型。

5.5.5 适配器与代理器的交互要求

适配器应与代理器进行网络连接，其网络连接的通信协议宜为 TCP/IP。适配器应向代理器注册，注册方法应符合第 8 章的规定。

5.5.6 适配器与数控机床交互要求

数控机床可采用任意协议（如开放协议或私有协议）与适配器建立通信关系。本文件不对其协议作规定。

5.5.7 应用系统与代理器的交互要求

应用系统应与代理器进行网络连接，其网络连接的通信协议宜为 TCP/IP。应用系统应向与其连接的代理器注册，注册方法应符合第 8 章的规定。

6 数控机床模型定义

6.1 模型组成

数控机床模型采用 JSON 进行描述,相关示例见附录 A。数控机床模型一般由根对象、设备对象、组件对象、数据对象、采样通道对象和内构方法对象组成。各对象的构成以及各对象间的关系见图 2。

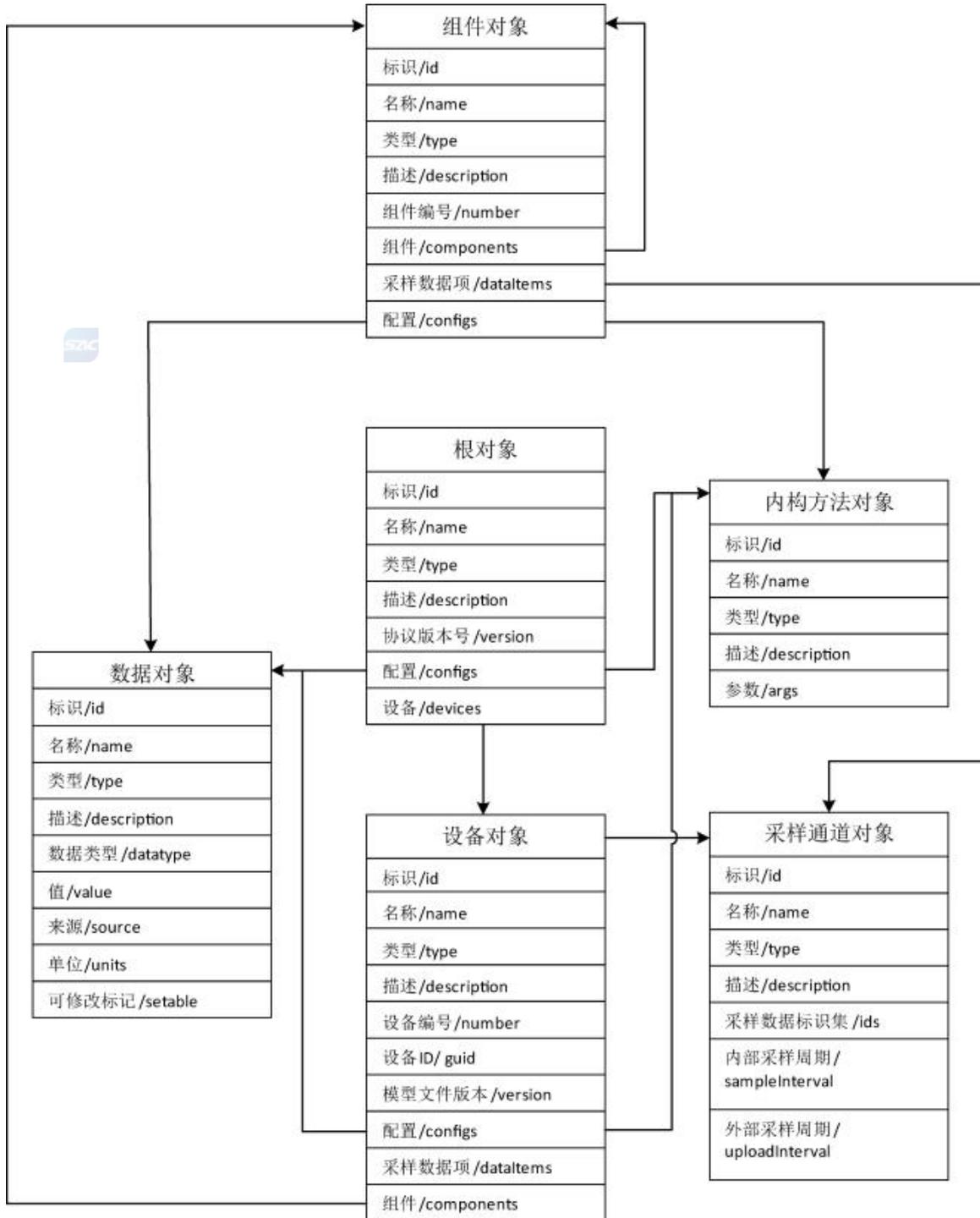


图 2 数控机床模型对象构成及各对象间关系

数控机床模型描述示例见附录 B。

根对象是数控机床模型的最外层对象；设备对象用来描述数控机床或其附属设备的配置、组件和可提供的数据项；组件对象用来描述设备对象中的组件信息，组件对象可包含其他组件对象；数据对象用于在数控机床模型文件中描述数控机床中的各种参数以及所有可采样数据；采样通道对象用来描述当前可采样的数据和采样周期；内构方法对象用来描述当前已定义的内构方法。

6.2 对象定义

6.2.1 数据类型

NC-Link 中所使用的数据类型可包括单值数据类型、字典数据类型、列表数据类型、文件类型 file。用户可根据需要对数据类型进行扩充。

单值数据类型 value、string、number、array、object、null 等应遵循 ECMA-404:2017 的定义。其中，string 类型的数值应采用 UTF-8 编码，number 类型的数值可由使用者根据数据实际用途动态转换成整数或浮点数。

字典数据类型 dict 是一种基于 JSON object 的数据类型，应支持以下操作：

- get_keys: 查询 dict 类型数据的所有键 key；
- get_value: 查询 dict 类型数据键 key 对应的值；
- set_value: 修改 dict 类型数据键 key 对应的值；
- get_attributes: 查询 dict 类型数据键 key 对应值的数据类型；
- add: 向 dict 中增加一项数据；
- delete: 删除 dict 中键 key 对应的一项数据。

列表数据类型 list 是一种基于 JSON array 的数据类型，应支持以下操作：

- get_length: 查询 list 的长度；
- get_value: 查询 list 类型数据对应位置的值；
- set_value: 修改 list 类型数据对应位置的值；
- get_attributes: 查询 list 类型数据对应位置的数据类型；
- add: 向 list 中指定位置增加一项数据；
- delete: 删除 list 中指定位置的一项数据。

dict 和 list 两种数据类型的使用方法应符合 8.2.11、8.2.12、8.2.13、8.2.14 的要求。

文件类型 file 是针对存储信息的特殊编码方式，用于识别内部储存的资料。

6.2.2 根对象

根对象包括标识、名称、类型、描述、协议版本号、配置和设备等属性，各属性的定义见表 1。

表 1 根对象属性列表

属性名称	含义描述	数据类型	是否可选
标识/id	根对象的标识号，应用系统内唯一	string	必选
名称/name	模型的名称，用易于理解的词语或词语的组合表示	string	可选

表 1 根对象属性列表 (续)

属性名称	含义描述	数据类型	是否可选
类型/type	只能为 NC_LINK_ROOT ^a	string	必选
描述/description	根对象的详细描述	string	可选
协议版本号/version	NC-Link 协议版本号, 形式为 1.x.x	string	必选
配置/configs ^b	数据对象或内构方法对象的数组, 描述根对象的多个配置信息	array	可选
设备/devices ^c	设备对象数组, 描述多个设备对象	array	必选

^a 表示根对象类型, 为固定的 string 类型数据。
^b 由一个或多个“数据对象”或“内构方法对象”形成的数组, “数据对象”的定义见 6.2.5, 配置中的数据对象不应作为采样数据源, 应用系统可查询或修改配置中的数据对象, “内构方法对象”的定义见 6.2.7。
^c 由一个或多个“设备对象”构成的数组, “设备对象”的定义见 6.2.3。

示例:

```
{
  "id": "6F9609FF-8B86-D111-C42D-00C04FC964FF",
  "name": "NC_LINK_PROBE",
  "type": "NC_LINK_ROOT",
  "description": "A NC_LINK instance given by NC_LINK Group",
  "version": "1.1.0",
  "configs": [{数据对象 1}, {数据对象 2}, {内构方法对象 1}, {内构方法对象 2}],
  "devices": [{设备对象 1}, {设备对象 2}]
}
```



6.2.3 设备对象

设备对象包括标识、名称、类型、描述、设备编号、设备 ID、模型文件版本、配置、采样数据项和组件等属性, 各属性的定义见表 2。

表 2 设备对象属性列表

属性名称	含义描述	数据类型	是否可选
标识/id	数控机床的标识号, 应用系统内唯一	string	必选
名称/name	数控机床的名称, 用易于理解的词语或词语的组合表示	string	可选
类型/type	数控机床类型之一, 如 MACHINE、ROBOT 等	string	必选
描述/description	对数控机床的详细描述	string	可选

表 2 设备对象属性列表（续）

属性名称	含义描述	数据类型	是否可选
设备编号/number	数控设备的编号,用以区分同类数控设备	string	可选
设备 ID/guid	设备唯一标识	string	可选
模型文件版本/version	模型文件版本号,形式为 1.x.x	string	必选
配置/configs ^a	包含数据对象和采样通道对象数组,描述数控机床的多个配置信息和可采集的数据	array	可选
采样数据项/dataItems ^b	数据对象数组,描述数控机床采样数据源	array	可选
组件/components	由组件对象形成的数组,描述设备对象下的多个组件	array	可选

^a 由一个或多个“数据对象”或“内构方法对象”形成的数组,“数据对象”的定义见 6.2.5,配置中的数据对象不应作为采样数据源,应用系统可查询或修改配置中的数据对象,“内构方法对象”的定义见 6.2.7。

^b 描述一个数控机床下所有可采集的数据,一个采样数据项中数据对象仅当其 id 被采样通道选取,才会实际被采集。

示例:

```
{
  "id": "7F9609FF-8B86-D111-C42D-00C04FC964FF",
  "name": "VMC-3Axis",
  "type": "MACHINE",
  "description": "3Axis example from NC_LINK Group",
  "configs": [{数据对象 1},{数据对象 2},{采样通道对象 1},{采样通道对象 2}],
  "dataItems": [{数据对象 1},{数据对象 2}],
  "number": "0001", "guid": "dev_uuid_1", "version": "1.3",
  "components": [{组件对象 1},{组件对象 2}]
}
```

6.2.4 组件对象

组件对象可包括标识、名称、类型、描述、组件编号、配置、采样数据项和组件,各属性的定义见表 3。

表 3 组件对象属性列表

属性名称	含义描述	数据类型	是否可选
标识/id	组件的标识号,Probe 文件中唯一	string	必选
名称/name	组件的名称,用易于理解的词语或词语的组合表示	string	可选
类型/type	组件类型之一,如 AXIS、SCREW、TOOL 等	string	必选
描述/description	对组件的详细描述。	string	可选

表 3 组件对象属性列表（续）

属性名称	含义描述	数据类型	是否可选
组件编号/number	组件的编号,用以区分同类组件	string	可选
配置/configs ^a	数据对象数组,描述组件的配置信息	array	可选
采样数据项/dataItems ^b	数据对象数组,描述组件采样数据源	array	可选
组件/components	子组件,是组件对象形成的数组	array	可选

^a 由一个或多个“数据对象”或“内构方法对象”形成的数组,“数据对象”的定义见 6.2.5,配置中的数据对象不应作为采样数据源,应用系统可查询或修改配置中的数据对象,“内构方法对象”的定义见 6.2.7。

^b 描述一个组件下所有可采集的数据,一个采样数据项中数据对象仅当其 id 被采样通道选取,才会实际被采集。组件对象嵌套用来表示对象间的层次关系。

示例:

```
{
  "name": "axis", "id": "axis", "type": "AXIS", "number": "0001", "description": "AXIS-X",
  "configs": [{数据对象 1}, {数据对象 2}],
  "dataItems": [{数据对象 1}, {数据对象 2}],
  "components": [{组件对象 1}, {组件对象 2}]
}
```

6.2.5 数据对象

数据对象包括标识、名称、类型、描述、数据类型、值、来源、单位和可修改标记等属性,各属性的定义见表 4。

表 4 数据对象属性列表

属性名称	含义描述	数据类型	是否可选
标识/id	数据对象的标识号,Probe 文件中唯一	string	必选
名称/name	数据对象的名称,用于区分同类型的数据	string	可选
类型/type	数据对象类型之一,例如 CURRENT, VERSION 等	string	必选
描述/description	数据对象的详细描述	string	可选
数据类型/datatype	list 或 dict,未指定此属性时,代表数据类型为单值数据类型	string	可选
值/value ^a	数据对象的数值,可为任何 JSON value	value	必选
来源/source ^b	描述数据来源于哪个组件	string	可选
单位/units	数据单位,仅当数据有单位且与缺省值不一致时才需要指定	string	可选

表 4 数据对象属性列表（续）

属性名称	含义描述	数据类型	是否可选
可修改标记/setable	是否可修改,缺省值为 false,不应修改	Boolean	可选
<p>^a 根据 type 的取值,value 可能取任意 JSON value 能表达的值。在数据对象中,仅当对象的值已知且固定不变时,才可在对象定义的时候给 value 赋值,该值可直接通过 Probe 文件获取,当然也可通过查询指令进行查询。</p> <p>^b source 是为了方便用户在不用全部详细描述所有组件的情况下,根据已列出和未列出组件的类型描述数据的来源。source 应指定从根对象到数据所在组件对象的唯一路径上所有对象的类型,相同类型对象,通过“对象类型#对象编号”的方式进行描述,不同类型之间用冒号“:”分割。</p>			

示例:

```
{ "id": "ver", "type": "VERSION", "value": "1.0" }
{ "id": "sp", "type": "SPEED_RATE", "units": "METER/SECOND" }
{ "id": "reg1", "type": "VARIABLE", "datatype": "list", "setable": "true" }
{ "id": "p1", "type": "PARAMETER", "datatype": "dict",
  "source": " NCLINK_ROOT:MACHINE#0001:AXIS"
}
```

6.2.6 采样通道对象

6.2.6.1 概述

采样通道对象包括标识、名称、类型、描述、采样数据标识集、内部采样周期和外部采样周期等属性,各属性的定义见表 5。

表 5 采样通道对象属性列表

属性名称	含义描述	数据类型	是否可选
标识/id	采样通道对象的标识号,Probe 文件内唯一	string	必选
名称/name	采样通道对象的名称	string	可选
类型/type	只能为 SAMPLE_CHANNEL	string	必选
描述/description	采样通道对象的详细描述	string	可选
采样数据标识集/ids	该通道中采集的数据的 id	array	必选
内部采样周期/sampleInterval	适配器采集机床数据的时间间隔,缺省单位为毫秒	number	必应
外部采样周期/uploadInterval	适配器向代理器发送数据的时间间隔,缺省单位为毫秒	number	必选

示例:

```
{
  "id": "cn1", "type": "SAMPLE_CHANNEL",
  "ids": [{"id": "id1"}, {"id": "id2"}, {"id": "id3"}], "sampleInterval": 10,
```

```
"uploadInterval": 100
}
```

一个数控机床模型文件可定义多个采样通道对象,采样数据标识集中所列的 id 应已在某个数控机床的数据项中进行了定义。

6.2.6.2 组合数据

组合数据是不同数据 id 的数据按照时间顺序组成的数据块,组合数据的数据 id 应由采样数据标识集指定。数据块内同一行内不同数据 id 的数据应为同一时间产生的数据,不同行的数据产生的时间应不相同,相邻行数据的产生时间的间隔由内部采样周期指定。

采样通道中返回的数据应为组合数据。在 6.2.6.1 中的示例,将会采集 id1、id2 和 id3 三个数据,分别表示位置、位移和电流,则采集到的组合数据如果为 $\{\{100, 200, 10\}, \{105, 205, 10\}, \{108, 204, 12\}, \dots\}$,则表示在采集开始的第 0 ms 数控机床对象 cn1 的位置、位移和电流分别为 100 mm、200 mm、10 A,第 10 ms 时的位置、位移和电流分别为 105 mm、205 mm、10 A,第 20 ms 时的位置、位移和电流分别为 108 mm、204 mm、12 A 等。

6.2.7 内构方法对象

6.2.7.1 概述

内构方法对象包括标识、名称、类型、描述、参数等属性,各属性的定义见表 6。

表 6 内构方法对象属性列表

属性名称	含义描述	数据类型	是否可选
标识/id	内构方法对象的标识号,Probe 文件内唯一	string	必选
名称/name	内构方法对象的名称,用易于理解的词语或词语的组合表示	string	可选
类型/type	只能为 METHOD	string	必选
描述/description	内构方法的详细描述	string	可选
参数/args	内构方法需要的参数描述	array	必选

示例:

```
{
  "id": "method001", "name": "ftpcall",
  "type": "METHOD",
  "description": "Call a ftp server",
  "args": {
    "protocol": "FTP",
    "address": "10.10.56.1:21",
    "user": "user1",
    "password": "111111",
    "source": "/gcodes/O1",
    "destination": "/h/lnc8/prog/O1",
```



```

"operation": "push"
}
}

```

一个数控机床模型文件可定义多个内构方法对象,内构方法对象只能在根对象的配置 configs 中进行定义,见 6.2.2。

7 数据项定义

7.1 概述

对设备对象、组件对象、数据对象中数据项的约定取值进行定义和描述。

7.2 设备对象的数据项

设备对象的数据项见表 7。

表 7 设备对象的数据项

type 属性取值	含义描述
AGV	自动搬运车
CLEAN	清洗装备
MACHINE	机床
MEASURE	检测装备
PRODUCTLINE	自动化产线
WAREHOUSE	自动料库

7.3 组件对象的数据项

组件对象的数据项见表 8。

表 8 组件对象的数据项

type 属性取值	含义描述
AUXILIARY	辅助系统
AXIS ^a	轴
COMPONENT	其他通用部件
CONTROLLER	控制器
FIXTURE	夹具
MOTOR	电机
SCREW	丝杠
SENSOR	传感器

表 8 组件对象的数据项（续）

type 属性取值	含义描述
SERVO_DRIVER	伺服驱动器
TOOL_MAGAZINE	刀库
^a 当 type 属性取值 AXIS 时,组件对象的 number 属性的取值见表 9。	

当组件对象的 type 属性为特定的取值时,应使用组件对象的 number 属性进行组合定义。部分组合定义见表 9。

表 9 组件对象的 type 属性的部分组合定义

对象 type 属性取值	对象 number 属性取值	含义描述
AXIS	X, Y, Z	线性轴单轴
	X1, Y1, Z1 ~ Xn, Yn, Zn	线性轴多轴
	A, B, C	旋转轴单轴
	A1, B1, C1 ~ An, Bn, Cn	旋转轴多轴
	U, V, W	摆动轴
	J1, J2, J3 ~ Jn	关节轴
	J1.1...1, J1.1...i ~ J1.2.1...1, J1.2.1...j ~ Jn.1...1 ~ Jn.k...m	关节轴含子轴

7.4 数据对象的数据项

数据对象的数据项可分为物理量数据项、通用数据项和专用数据项。

物理量数据项见表 10。

表 10 物理量数据项

type 属性取值	value 属性数据类型	value 属性数据单位	含义描述
ACCELERATION	number	mm/s ²	加速度
ANGLE	number	rad	角位置
ANGULAR_ACCELERATION	number	rad/s ²	角加速度
ANGULAR_VELOCITY	number	rad/s	角速度
CONCENTRATION	number	m/m	浓度
CONDUCTIVITY	number	S/m	导电力
CURRENT	number	A	电流
DISPLACEMENT	number	mm	位移
ENERGY	number	kW · h	功耗

表 10 物理量数据项 (续)

type 属性取值	value 属性数据类型	value 属性数据单位	含义描述
FLOW	number	L/s	流体瞬时流量
FREQUENCY	number	Hz	频率
LENGTH	number	mm	长度
MASS	number	kg	质量
PERIOD	number	s	周期
POSITION	number	mm	位置
POWER	number	W	功率
POWER_FACTOR	number	—	功率因数
PRESSURE	number	Pa	压强
RESISTANCE	number	Ω	电阻
ROTATION_SPEED	number	r/min	转速
SPEED	number	mm/min	速度
TEMPERATURE	number	$^{\circ}\text{C}$	温度
TORQUE	number	$\text{N} \cdot \text{m}$	扭矩
VISCOSITY	number	$\text{Pa} \cdot \text{s}$	黏度
VOLT_AMPERE	number	VA	视在功率

数据对象的可选属性 units 数据单位,仅当数据有单位且与缺省值不一致时才需要指定。常用的非默认数据单位见表 11。

表 11 属性 units 常用的非默认数据单位

默认数据单位	常用的非默认数据单位	含义描述
A	mA, μA	电流
mm	m, μm	位移
$\text{W} \cdot \text{h}$	$\text{kW} \cdot \text{h}$, $\text{W} \cdot \text{s}$	功耗
L/s	mL/s	流体瞬时流量
Hz	kHz, MHz, GHz	频率
mm	m	长度
kg	g, mg	质量
W	kW, mW	功率

表 11 属性 units 常用的非默认数据单位 (续)

默认数据单位	常用的非默认数据单位	含义描述
Ω	kΩ, MΩ	电阻
mm/min	m/min	速度
°C	°F	温度
VA	kVA	视在功率

通用数据项见表 12。

表 12 通用数据项

type 属性取值	value 属性数据类型	数据单位	含义描述
CATAGORY ^a	string	—	类型
CREATE_TIME	string	—	创建时间
CREATOR	string	—	创建者
IP	string	—	网络 IP 地址
MANUFACTURER	string	—	厂商
MODEL	string	—	型号
NAME	string	—	名称
PARAMETER	dict	—	参数
SERIAL_NUMBER	number	—	编号
STATUS ^b	string	—	设备、组件运行状态
UPPER_LIMIT	number	—	数量上限
USER_ROLE ^b	string	—	用户角色
VARIABLE	list	—	运行变量
VERSION	string	—	版本信息
WARNING ^c	object	—	报警信息
WORK_MODE ^b	string	—	工作模式

^a 当 type 属性取值 CATAGORY 时,数据对象 value 属性的取值定义见表 16。
^b 当 type 属性取值 STATUS、WORK_MODE、USER_ROLE 时,数据对象 value 属性的取值定义见表 15。
^c 当 type 属性取值 WARNING 时,具体的 JSON 对象的取值定义见表 14。

专用数据项见表 13。

表 13 专用数据项

type 属性取值	value 属性数据类型	数据单位	含义描述
AUTO_MODE	Boolean	—	自动模式
AXIS_BACKLASH	number	mm	轴反向间隙
AXIS_LOAD	number	kg	轴负载
AXIS_TOREFPOINT	Boolean	—	轴回参考点
BLOCK_SKIP_MODE	Boolean	—	跳段模式
CHIP_REMOVAL ^a	string	—	排屑模式
COMPACITY	number	—	刀库容量
CONSOLE	string	—	控制台
COOLANT_ON	Boolean	—	冷却液开
COORDINATE ^b	object	—	坐标系
CYCLE_START	Boolean	—	循环启动
DOOR_CLOSED	Boolean	—	门关闭
EDIT_MODE	Boolean	—	编辑模式
EMG	Boolean	—	急停
FEED_HOLD	Boolean	—	进给保持
FEED_OVERRIDE	number	—	进给倍率
FEED_SPEED	number	mm/s	进给速度
FILE	file	—	文件
FIXTURE_MODE ^a	string	—	夹具状态
HANDLE_MODE	Boolean	—	手摇模式
INCREMENTAL_MODE	Boolean	—	增量模式
JOG_MODE	Boolean	—	手动模式
LIGHTING_ON	Boolean	—	照明灯亮
LINE_NUMBER	string	—	程序行号
LUBRICANT_ON	Boolean	—	润滑液开
MACHINE_LOCKED	Boolean	—	机床锁定
MDI_MODE	Boolean	—	MDI 模式
MST_LOCKED	Boolean	—	MST 锁定
OPT_STOP_MODE	Boolean	—	选停模式

表 13 专用数据项 (续)

type 属性取值	value 属性数据类型	数据单位	含义描述
PATH_LEFT_LENGTH	number	mm	剩余进给
PART ^b	object	—	工件
PART_COUNT	string	—	加工件数
PERIOD	number	s	周期
PROGRAM	string	—	主程序名
PROGRAM_MODE	string	—	程序模式
PROGRAM_NUMBER	number	—	当前程序号
RAPID_OVERRIDE	number	—	快移倍率
REF_MODE	Boolean	—	回零模式
RESET_MODE	Boolean	—	复位
RESIDUAL	number	—	剩余量
SHELF_UNIT	list	—	仓位
SINGLE_BLOCK_MODE	Boolean	—	单段模式
SITE	string	—	AGV 工作站点
SPINDLE_MODE ^a	string	—	主轴工作模式
SPINDLE_OVERRIDE	number	—	主轴倍率
SPINDLE_SPEED	number	r/min	主轴速度
SUBPROGRAM	string	—	子程序名
TOOL	list	—	刀具
TOOL_CHANGE ^a	string	—	换刀模式
TOOL_CHANGING	Boolean	—	正在换刀
TOOL_PARAM ^b	object	—	刀具参数
TOOL_NUMBER	number	—	当前刀具号
TOOL_POSITION	number	—	当前刀位
WARNING ^b	object	—	报警信息
^a 当 type 属性取值 CHIP_REMOVAL、FIXTURE_MODE、SPINDLE_MODE、TOOL_CHANGE 时,数据对象 value 属性的取值定义见表 15。 ^b 当 type 属性取值 COORDINATE、PART、TOOL_PARAM、WARNING 时,具体的 object 的取值定义见表 14。			

当数据对象 type 属性为特定取值时,使用数据对象 value 属性进行组合定义,value 属性的数据类型为 object。组合定义见表 14。

表 14 数据对象 type 属性取特定值时 value 属性的定义

type 属性取值	value 属性取值	数据类型	含义描述
COORDINATE	name	string	坐标名,如 G54、G55 等
	x,y,z	number	可选线性轴坐标系数据
	a,b,c	number	可选旋转轴、摆动轴坐标系数据
	x1,y1,z1 ~ xn,yn,zn	number	多通道线性轴坐标系数据
	a1,b1,c1 ~ an,bn,cn	number	多通道旋转轴、摆动轴坐标系数据
PART	id	string	工件编号
	name	string	名称
	material	string	材料
	length	number	长
	width	number	宽
	height	number	高
	snap_shot	file	零件图片文件
	geometry_file	file	零件 CAD 模型文件
	key_value	dict	自定义键值对的测量参数
TOOL_PARAM	type	string	刀具类型
	id	string	刀具编号
	name	string	刀具名称
	kind	string	刀具类型
	radius	number	刀具半径
	length	number	刀具长度
	time_usage	number	使用时间
	max_time_usage	number	最大使用时长
	count_usage	number	使用次数
	max_count_usage	number	最大使用次数
	length_process	number	切削里程
	max_length_process	number	最大切削里程
	radius_abaration	number	刀具半径补偿
	length_abaration	number	刀具长度补偿
key_value	dict	自定义键值对的刀具参数	
WARNING	number	string	报警编号
	text	string	报警内容
	emerge_time	number	报警产生时间
	key_value	dict	无或多组自定义键值对报警数据

当数据对象 type 属性为特定取值时,应使用数据对象 value 属性和设备/组件对象的 type 属性进

行组合定义。value 属性的数据类型为 string。

仅使用数据对象 value 属性进行组合定义的,见表 15。

表 15 数据对象 type 属性取特定值时仅使用 value 属性的组合定义

type 属性取值	value 属性取值	含义描述
CHIP_REMOVAL	forward	排屑正转
	stop	排屑停止
	reverse	排屑反转
FIXTURE_MODE	loose	夹具松开
	tight	夹具卡紧
SPINDLE_MODE	forward	主轴正转
	stop	主轴停止
	reverse	主轴反转
STATUS	running	运行
	free	空闲
	holding	进给保持(例如,在自动产线中,设备等待上料时,就处于 holding 状态)
TOOL_CHANGE	auto	自动换刀模式
	manual	手动换刀模式
USER_ROLE	operator	操作工
	manager	车间管理员
	machine_manufacturer	机床厂家
	cnc_manufacturer	数控厂家
	system_manager	系统管理员
WORK_MODE	manual	手动模式
	auto	自动模式

同时使用数据对象 value 属性和设备/组件对象的 type 属性进行组合定义的,见表 16。

表 16 数据对象 type 属性取特定值时 value 属性和设备/组件对象的 type 属性的组合定义

数据对象 type 属性取值	设备/组件对象 type 属性取值	数据对象 value 属性取值	含义描述
CATAGORY	AXIS	rotary	旋转轴
		linear	线性轴
	MACHINE	lathe	车床
		milling	铣床
		grinder	磨床
		planer	刨床
		other	其他

8 数据传输接口定义

8.1 数据传输接口概述

8.1.1 通信方式

本文件中的数据传输接口有“请求/响应”和“订阅/发布”两种接口通信方式。

数据传输接口“请求/响应”通信方式见图 3,该通信方式用于应用系统主动向代理器或代理器主动向适配器请求获得数据或请求设置数据,适配器或代理器在收到请求后被动做出响应。应用系统通过请求数据传输接口向代理器发送“Request DTU”;代理器通过存储转发接收到的“Request DTU”,使用请求数据传输接口向适配器发送“Request DTU”。当适配器通过请求数据传输接口收到“Request DTU”后,适配器通过请求内容对应的响应接口向代理器发送“Response DTU”;代理器存储转发接收到的“Response DTU”,使用响应接口向应用系统发送“Response DTU”。存储时间由代理器实现者规定。

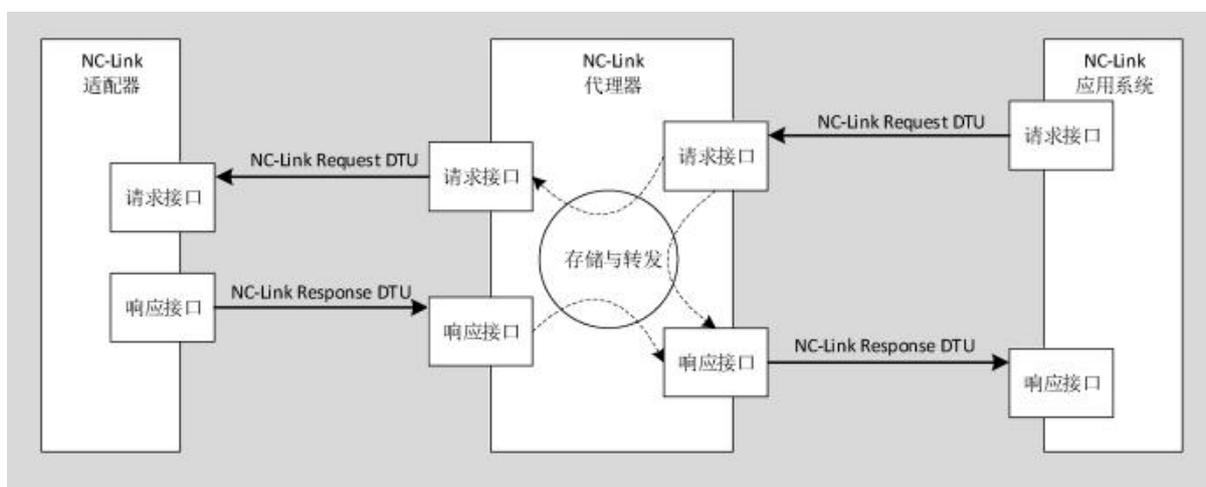


图 3 数据传输接口“请求/响应”通信方式

数据传输接口“订阅/发布”通信方式见图 4,该通信方式用于适配器主动向代理器发布数据和应用系统主动向代理器订阅数据。当代理器拥有应用系统所订阅的数据时,代理器向应用系统发布该数据。适配器每过一定周期主动向代理器发布“Sample DTU”。代理器向订阅了该数据的应用系统存储转发“Sample DTU”。订阅过该数据的应用系统通过对应数据传输接口从代理器处获取“Sample DTU”。没有订阅该内容的应用系统可通过数据传输接口向代理器订阅内容。

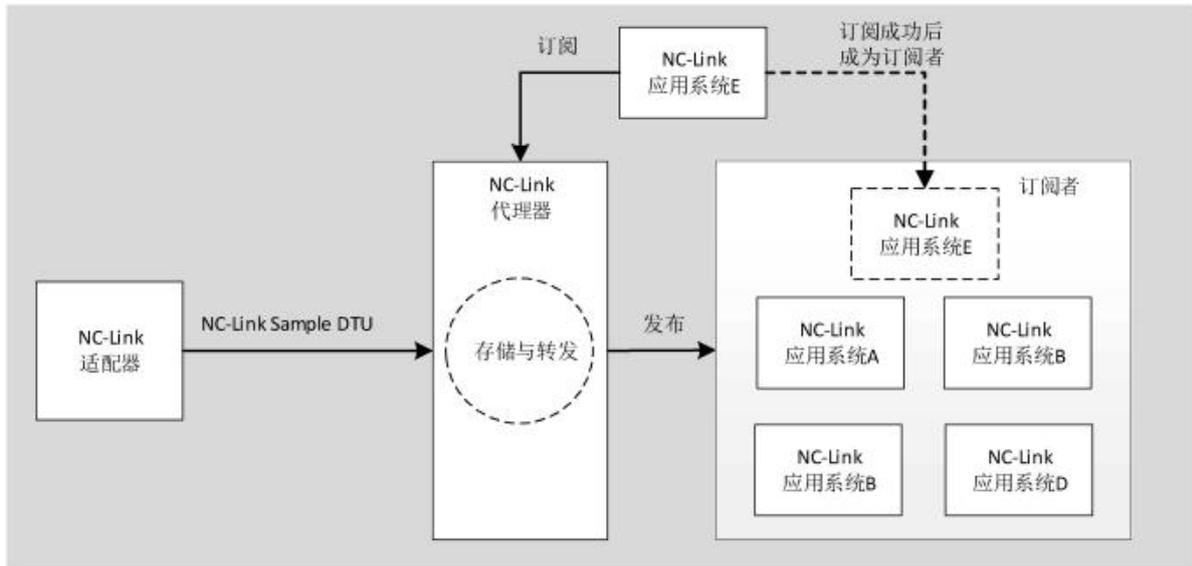


图 4 数据传输接口“订阅/发布”通信方式

8.1.2 数据传输接口明细表

数据传输接口通过具体的指令实现信息交互。数据传输接口指令列表见表 17。

表 17 数据传输接口指令列表

数据传输接口名称	数据传输接口指令	含义	备注
注册请求数据传输接口	Register/Request	适配器向代理器进行注册,代理器可实现对适配器的设备管理功能、设备认证功能、访问权限管理功能等	适配器、代理器、应用系统应支持该指令
注册响应数据传输接口	Register/Response/cli_uid	代理器对适配器注册的响应	适配器、代理器、应用系统应支持该指令
终端探测请求数据传输接口	Discovery/Request/ex_cid	应用系统向代理器请求目前在线或已注册的适配器	代理器、应用系统可支持该指令
终端探测响应数据传输接口	Discovery/Response/ex_cid	代理器向应用系统返回 Discovery/Request 请求的响应	代理器、应用系统可支持该指令
版本号校对请求数据传输接口	Probe/Version/dev_uid	适配器向代理器或应用系统给出当前的 NC-Link 模型文件版本号。如果代理器或应用系统缓存的 dev_uid 设备的模型文件的版本号不一致,则应通过 Probe/Query/Request/dev_uid 接口重新获取 dev_uid 设备的模型文件	适配器、代理器、应用系统应支持该指令

表 17 数据传输接口指令列表（续）

数据传输接口名称	数据传输接口指令	含义	备注
版本号校对响应数据传输接口	Probe/Version/Response/dev_uuid	代理器或应用系统向适配器返回 Probe/Version/dev_uuid 的响应	适配器、代理器、应用系统可支持该指令
模型侦测请求数据传输接口	Probe/Query/Request/dev_uuid	应用系统或代理器向适配器请求 dev_uuid 设备的模型文件。当代理器实现设备管理、用户访问权限管理、历史数据服务等功能时,可能需要数控机床的模型文件	适配器、代理器、应用系统应支持该指令
模型侦测响应数据传输接口	Probe/Query/Response/dev_uuid	应用系统或代理器对适配器 Probe/Query/Request/dev_uuid 接口请求的响应	适配器、代理器、应用系统应支持该指令
模型设置请求数据传输接口	Probe/Set/Request/dev_uuid	应用系统通过代理器,请求更新适配器的模型文件	适配器、代理器、应用系统可支持该指令
模型设置响应数据传输接口	Probe/Set/Response/dev_uuid	适配器通过代理器,向应用系统返回更新模型文件请求 Probe/Set/Request/dev_uuid 的处理结果	适配器、代理器、应用系统可支持该指令
数据查询请求数据传输接口	Query/Request/dev_uuid	应用系统通过代理器,向适配器查询由 dev_uuid 标识的数控机床的某项或多项数据	适配器、代理器、应用系统应支持该指令
数据查询响应数据传输接口	Query/Response/dev_uuid	适配器通过代理器,向应用系统返回 Query/Request/dev_uuid 查询请求的结果信息	适配器、代理器、应用系统应支持该指令
数据设置请求数据传输接口	Set/Request/dev_uuid	应用系统通过代理器,向适配器设置(或更新)由 dev_uuid 标识的数控机床的某项或多项数据	适配器、代理器、应用系统可支持该指令
数据设置响应数据传输接口	Set/Response/dev_uuid	适配器通过代理器,向应用系统返回 Set/Request/dev_uuid 设置请求的结果信息	适配器、代理器、应用系统可支持该指令
数据采样数据传输接口	Sample/dev_uuid	适配器通过代理器向应用系统主动推送由 dev_uuid 所标识的数控机床的运行时数据。具体的数据项和推送频率,由模型文件定义	适配器、代理器、应用系统可支持该指令
状态通知数据传输接口	Notify/State/tid	代理器在检测到终端的连接状态发生变更时,向监听该终端状态通知数据传输接口的终端发送状态通知消息	适配器、代理器、应用系统可支持该指令

表 17 数据传输接口指令列表 (续)

数据传输接口名称	数据传输接口指令	含义	备注
动态采样注册请求数据传输接口	Register/Sample/Request/ <i>dev_uuid</i>	用于代理器或应用系统向适配器注册动态采样。动态采样使得应用系统或代理器在不更新数控机床模型文件的情况下,具有增加采样通道的能力。适配器依据动态采样注册请求,动态分配采样通道资源,启动采样任务	适配器、代理器、应用系统可支持该指令
动态采样注册响应数据传输接口	Reigster/Sample/Response/ <i>dev_uuid</i> /	用于适配器向应用系统发送动态采样注册响应的 NC-Link 数据传输单元。适配器通过该数据传输接口向代理器或应用系统返回动态采样注册结果	适配器、代理器、应用系统可支持该指令
动态采样注销请求数据传输接口	Unregister/Sample/Request/ <i>dev_uuid</i>	用于代理器或应用系统向适配器注销动态采样。用于注销已注册的动态采样通道,注销后,采样任务终止,注销的动态采样通道资源被回收	适配器、代理器、应用系统可支持该指令
动态采样注销响应数据传输接口	Unreigster/Sample/Response/ <i>dev_uuid</i>	用于适配器向应用系统发送动态采样注销响应的 NC-Link 数据传输单元。适配器通过该数据传输接口向代理器或应用系统返回动态采样注销结果	适配器、代理器、应用系统可支持该指令
动态数据采样数据传输接口	Sample/ <i>dev_uuid</i> / <i>ex_cid</i> /sample_channel_id	用于适配器通过代理器向应用系统主动推送由/ <i>dev_uuid</i> / <i>ex_cid</i> / <i>sample_channel_id</i> 所标识的数控机床的运行时数据。具体的数据项和推送频率,在动态采样注册时设置。其中 <i>dev_uuid</i> 是发送该消息的适配器的 GUID, <i>ex_cid</i> 是接收该消息的应用系统的终端标识符, <i>sample_channel_id</i> 是该适配器中模型文件给出的合法的采样通道标识符	适配器、代理器、应用系统可支持该指令
内构方法调用数据传输接口	Method/Call/ <i>dev_uuid</i>	用于代理器或应用系统向适配器下达调用内构方法指令。内构方法可提供文件管理功能、常见的控制功能模块(如循环启动、复位、停止等),也可由用户自定义控制功能模块	适配器、代理器、应用系统可支持该指令

表 17 数据传输接口指令列表（续）

数据传输接口名称	数据传输接口指令	含义	备注
内构方法进度数据传输接口	Method/Status/ <i>dev_uuid</i>	用于适配器向代理器或应用系统发送内构方法执行进度。适配器通过发送内构方法执行进度,告知应用系统当前的内构方法是否仍在正常执行	适配器、代理器、应用系统可支持该指令
内构方法结果数据传输接口	Method/Result/ <i>dev_uuid</i>	用于适配器向代理器或应用系统发送内构方法执行结果。适配器通过发送内构方法执行结果,告知应用系统当前的内构方法是否执行成功	适配器、代理器、应用系统可支持该指令
内构方法控制请求数据传输接口	Method/Control/Request/ <i>dev_uuid</i>	用于代理器或应用系统向适配器发送内构方法控制请求指令。代理器或应用系统通过发送内构方法控制指令,可临时改变内构方法的执行	适配器、代理器、应用系统可支持该指令
内构方法控制响应数据传输接口	Method/Control/Response/ <i>dev_uuid</i>	当适配器在接口 Method/Control/Request/ <i>dev_uuid/ex_cid</i> 收到代理器或应用系统发布的内构方法控制请求后。通过该数据传输接口向代理器或应用系统返回内构方法控制请求结果	适配器、代理器、应用系统可支持该指令
事件注册请求数据传输接口	Register/Event/Request/ <i>dev_uuid</i>	用于代理器或应用系统向适配器注册事件。事件使得应用系统或代理器在不更新数控机床模型文件的情况下,具有获知指定数控机床数据变化的能力。通过使用事件,适配器可在数据发生特定变化时及时通知代理器或应用系统	适配器、代理器、应用系统可支持该指令
事件注册响应数据传输接口	Reigster/Event/Response/ <i>dev_uuid</i>	用于适配器向应用系统发送事件注册响应的 DTU。适配器通过该数据传输接口向代理器或应用系统返回事件注册结果	适配器、代理器、应用系统可支持该指令
事件注销请求数据传输接口	Unregister/Event/Request/ <i>dev_uuid</i>	用于代理器或应用系统向适配器注销事件。用于注销已注册的事件,注销后,事件任务终止,注销的事件资源被回收	适配器、代理器、应用系统可支持该指令
事件注销响应数据传输接口	Unreigster/Event/Response/ <i>dev_uuid</i>	用于适配器向应用系统发送事件注销响应的 DTU。适配器通过该数据传输接口向代理器或应用系统返回事件注销结果	适配器、代理器、应用系统可支持该指令

表 17 数据传输接口指令列表（续）

数据传输接口名称	数据传输接口指令	含义	备注
事件数据数据传输接口	Event/dev_uuid	用于适配器向代理器或应用系统发送事件数据。适配器发送事件数据时表明事件被触发,代理器或应用系统可及时获得数据改变的消息	适配器、代理器、应用系统可支持该指令

8.1.3 消息标识符



消息标识符的键定义字符串“@id”,消息标识符的值用来标识区分不同的数据传输单元。应用系统每次进行请求操作,应在“Request DTU”中给出形如 {"@id": "some_mid_xxx"}的键值对。代理器在存储转发该“Request DTU”时,不能修改该键值对。适配器在收到该“Request DTU”,处理完成后,生成相对应的“Response DTU”返回给应用系统时,应在“Response DTU”中携带该 {"@id": "some_mid_xxx"}键值对。应用系统使用 {"@id": "some_mid_xxx"}键值对作为请求/响应的匹配依据。

键值对 {"@id": "some_mid_xxx"}应由应用系统生成,键值对中的值“some_mid_xxx”应具有唯一性。

8.1.4 终端标识符

终端标识符包括适配器标识符和应用系统标识符,这两种标识符应在整个系统中唯一。

注:“整个系统”是指包括至少一个代理器,与该代理器连接的所有适配器和与该代理器连接的所有应用系统。

当适配器和应用系统连接到代理器时,应携带唯一的终端标识符。

对于适配器,其标识符应使用模型文件中的 GUID 作为终端标识符;对于应用系统,其标识符应是系统中唯一的。

“dev_uuid”表示通用的(非特定的)适配器的终端标识符变量,“ex_cid”表示通用的(非特定的)应用系统的终端标识符变量。“dev_uuid_x”表示某个特定的适配器的终端标识符变量,“ex_cid_x”表示某个特定的应用系统的终端标识符变量。其中的“x”代表任何数字和字母。

8.1.5 数据类型的访问

单值类型的数据项只能以整个数据项为单位进行访问,不能单独访问其中的子对象/子域/索引项等。例如,不能访问字符串中的某个字符或子串,不能访问数组中某个索引项或某子域,不能访问对象中的某个子对象。

字典类型的数据项可单独访问其中的任意元素,包括每一个键和值。

列表类型的数据项可通过索引访问列表中的每一个元素或子域。

8.1.6 数据传输单元

数据传输单元是一个 JSON 对象,是数据传输接口指令中的指令参数。每个数据传输接口有定义的数据传输单元,内容为键值对。键为关键字,是一个字符串,不应被更改。值为该关键字所传达的内容,由适配器、代理器、应用系统设置,数据类型应是单值数据类型、字典数据类型、列表数据类型中的一种。每一个数据传输单元中应包含消息标识符和终端标识符。数据传输单元通过消息标识符区分不同的消息,通过终端标识符识别发送/接收信息的终端。

8.1.7 状态码及错误码定义

当接收者不支持发送者的数据访问请求,或发生其他通信故障时,应在“Response DTU”中通过“code”和“reason”键值对给出响应的指示。键“code”的取值见表 18 定义。

表 18 请求执行状态码取值

值	含义	类型
OK	表示请求被正确执行,数据有效	string
NG	表示请求被执行,但执行失败	string

表 19 给出了 NC-Link 数据传输单元中,表示错误原因的键“reason”的取值定义,以及对应的错误码的键“error”的取值定义,错误原因及错误码在数据传输单元中的规范性定义见表 20。

表 19 错误原因列表

错误原因	错误码	含义	场景
Success	0	正常、成功	
Out of Range	101	超出数组的索引范围	应用系统发出“RequestDTU”请求: 由适配器返回给应用系统。当应用系统请求/设置适配器中数组或列表类型的数据时,如果数组访问越界,则适配器返回该错误码
Parameter Missing	102	参数错误	应用系统发出“RequestDTU”请求: 当适配器接收到应用系统的数据请求,经检查发现该数据请求需要参数,或对数据的操作需要参数,而 RequestDTU 中没有给出所需要的参数,则向应用系统返回该错误码。 例如:当更新/设置某个数据项时,没有给出待写入的新值;应用系统请求适配器执行某个动作,但没有给出该动作要求的参数
Unavailable	202	数据或服务不可用	应用系统发出“RequestDTU”请求: a) 由适配器返回给应用系统,指示应用系统所请求的数据项不可用或不存在; b) 由代理器返回给应用系统,指示适配器不可使用,或是所请求的数据项不可访问
Timeout	203	请求超时	应用系统发出“RequestDTU”请求: a) 当适配器接收到应用系统的数据请求,内部处理超时的情况下,由适配器返回给应用系统; b) 当代理器没有在设定的时间范围内收到适配器的 ResponseDTU,可向应用系统返回该错误码

表 19 错误原因列表（续）

错误原因	错误码	含义	场景
Permission Denied	301	权限错误	应用系统发出“RequestDTU”请求： a) 当适配器接收到应用系统的数据请求，经检查发现无访问权限（读/写权限），则向应用系统返回该错误码； b) 代理器在向适配器转发该数据请求前，经检查发现该应用系统没有访问指定的适配器的权限，或没有访问适配器中数据项的权限，则向应用系统返回该错误码
Node not exists	401	访问节点不存在	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 所请求的节点不存在
No index passed in	402	未输入索引值	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 需要对 list 类型数据进行访问时，但没有输入 index 值
No key passed in	403	未输入键值	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 需要对 dict 类型数据进行访问时，但没有输入 key 值
Memory configured error	404	内存分配错误	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 所请求的功能已没有可分配的内存
Value type error	405	值类型错误	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 键值对中的值类型错误
unsupported operation get_length	406	不支持 get_length 操作	应用系统发出“Request DTU”请求： 当适配器或代理器对 DTU 进行解析后发现该 DTU 为数据请求 DTU，但所请求数据项类型不支持 get_length 操作
unsupported operation get_attributes	407	不支持 get_attributes 操作	应用系统发出“Request DTU”请求： 当适配器或代理器对 DTU 进行解析后发现该 DTU 为数据请求 DTU，但所请求数据项类型不支持 get_attributes 操作
unsupported operation get_keys	408	不支持 get_keys 操作	应用系统发出“Request DTU”请求： 当适配器或代理器对 DTU 进行解析后发现该 DTU 为数据请求 DTU，但所请求数据项类型不支持 get_keys 操作
arguments nimiety	409	传入参数过多错误	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 中所传入的参数多于所需参数数量
key error	410	字典键错误	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 键值对中的键类型错误

表 19 错误原因列表（续）

错误原因	错误码	含义	场景
Attribute error	411	属性错误	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 键值对属性错误
syntax error	412	语法错误	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 格式语法错误
keys of arguments error	413	传入参数键错误	应用系统发出“Request DTU”请求： 当适配器对 DTU 进行解析后发现该 DTU 键值对中的键的关键词不符合该数据传输接口的要求
unsupported terminal identifier	414	不合法的终端标识符	适配器或应用系统发出“Request DTU”注册请求： 当代理器对 DTU 进行解析后发现该 DTU 中的终端标识符不合法
Insufficient dynamic sampling resources	415	动态采样资源已满	应用系统发出“Request DTU”动态采样注册请求： 当适配器对 DTU 进行解析后发现该 DTU 所需的动态采样资源大于目前适配器可分配资源
Insufficient event resources	416	事件资源已满	应用系统发出“Request DTU”事件注册请求： 当适配器对 DTU 进行解析后发现该 DTU 所需的事件资源大于目前适配器可分配资源
Insufficient method resources	417	内构方法资源已满	应用系统发出“Request DTU”内构方法注册请求： 当适配器对 DTU 进行解析后发现该 DTU 所需的内构方法资源大于目前适配器可分配资源
method action type error	418	内构方法控制指令错误	应用系统发出“Request DTU”内构方法控制指令： 当适配器对 DTU 进行解析后发现该 DTU 中关键字 action 所对应的值不为 stop、resume、cancel
filter type error	419	过滤器类型错误	应用系统发出“Request DTU”事件注册请求： 当适配器对 DTU 进行解析后发现该 DTU 中关键字 filter 中 type 的种类不在 change、equal、range、less、greater 或 include 中
trigger type error	420	触发器类型错误	应用系统发出“Request DTU”事件注册请求： 当适配器对 DTU 进行解析后发现该 DTU 中关键字 trigger 中 type 的种类不在 always、normal、once 中
filter type and arguments not matched	421	过滤器类型与过滤器参数不匹配	应用系统发出“Request DTU”事件注册请求： 当适配器对 DTU 进行解析后发现该 DTU 中过滤器关键词与过滤器类型不匹配。如 change 类型过滤器中有“from”
用户自定义	小于 0	用户自定义	所有小于 0 的错误码及对应的错误原因由用户自定义

表 20 状态码、错误原因及错误码在数据传输单元中的规范性定义

键	值(示例)	值类型	要求
code	NG	string	该键值对应存在
reason	Out of Range	string	该键值对可选存在
error	101	number	该键值对可选存在

8.2 数据传输接口定义

8.2.1 注册请求数据传输接口

8.2.1.1 数据传输接口指令

注册请求数据传输接口指令为: Register/Request。

8.2.1.2 用途

注册请求数据传输接口用于终端第一次连接代理器时向代理器发送注册信息。

8.2.1.3 用法

终端使用本数据传输接口发送消息,代理器监听本数据传输接口接收消息。注册指令交互采用“请求/响应”模式。

终端第一次连接代理器时,通过本数据传输接口向代理器发送注册请求消息;在连接到代理器时,应使用其终端标识符在代理器中注册。代理器使用此终端标识符来管理该终端的接入和后续访问控制。

8.2.1.4 数据传输单元定义

注册请求数据传输单元定义见表 21。

表 21 注册请求数据传输单元定义

键	值(示例)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对可存在
cli_uuid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对对应存在
privateInfo	null	object	属于实现的自定义范畴	该键值对可存在

8.2.1.5 数据传输单元示例

示例 1: 适配器向代理器注册

适配器通过注册请求数据传输接口发送如下“Request DTU”。

```
{
  "@id": "some_mid_001",
```

```
"cli_uuid": "dev_uuid "
```

示例 2：应用系统向代理器注册

应用系统通过注册请求数据传输接口发送如下“Request DTU”。

```
{
  "@id": "some_mid_001",
  "cli_uuid": "ex_cid "
}
```

8.2.2 注册响应数据传输接口

8.2.2.1 数据传输接口指令

注册响应数据传输接口指令为：Register/Response/cli_uuid。其中“cli_uuid”有两种：“dev_uuid”是请求注册的适配器的 GUID；“ex_cid”是请求注册的应用系统的终端标识符。

8.2.2.2 用途

当代理器在注册响应数据传输接口收到适配器或应用系统发送的注册请求后，通过本数据传输接口向请求注册的适配器或应用系统返回注册结果响应。

8.2.2.3 用法

终端监听本数据传输接口接收消息，代理器使用本数据传输接口发送消息。注册指令交互采用“请求/响应”模式。

代理器在注册响应数据传输接口收到终端发送的注册请求后，判断终端标识符是否合法。对于不合法的适配器终端标识符，代理器可拒绝该适配器的连接、数据的转发等。对于合法的终端标识符，代理器应接受注册，并将终端之间的“Request/Response DTU”进行正确的存储转发。对于不合法的应用系统终端标识符，代理器可拒绝连接、拒绝访问、限制其访问权限等。

当代理器不具有用户管理功能时，代理器应通过终端标识符来区分不同的应用系统；当代理器具有用户管理功能时，只有在应用系统提供了终端标识符所对应的合法安全验证时才能注册成功。

只有注册成功，终端才能通过代理器进行有效信息交互。当注册不成功时，代理器可拒绝与该终端的连接，可不向该终端返回注册结果信息。当注册成功时，代理器应向该终端返回注册成功信息。在任意时刻，代理器对同一个终端标识符只维护一个有效连接。

8.2.2.4 数据传输单元定义

注册响应数据传输单元定义见表 22。

表 22 注册响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	some_mid_001	string	应符合 8.1.3 的要求	该键值对可存在
cli_uuid	dev_uuid	string	应符合 8.1.4 的要求	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可选存在

表 22 注册响应数据传输单元定义（续）

键	值(示例值)	值类型	描述	要求
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
privateInfo	null	object	属于实现的自定义范畴	该键值对可存在

8.2.2.5 数据传输单元示例

示例 1：适配器注册响应

代理器从注册响应数据传输接口向适配器发送如下“Response DTU”。

```
{
  "@id": "some_mid_001",
  "cli_uuid": "dev_uuid",
  "code": "NG",
  "reason": "Permission Denied",
  "error": 301
}
```

示例 2：应用系统注册响应

代理器从注册响应数据传输接口向应用系统发送如下“Response DTU”。

```
{
  "@id": "some_mid_001",
  "cli_uuid": "ex_cid",
  "code": "NG"
  "reason": "Permission Denied"
}
```



8.2.3 终端探测请求数据传输接口

8.2.3.1 数据传输接口指令

终端探测请求数据传输接口指令为:Discovery/Request/ex_cid。其中“ex_cid”是发送探测请求的应用系统的终端标识符。

8.2.3.2 用途

应用系统通过该数据传输接口从代理器获得已经注册的适配器列表。应用系统在任何时候均可向代理器进行适配器终端探测请求。

图 5 给出了适配器终端探测流程。

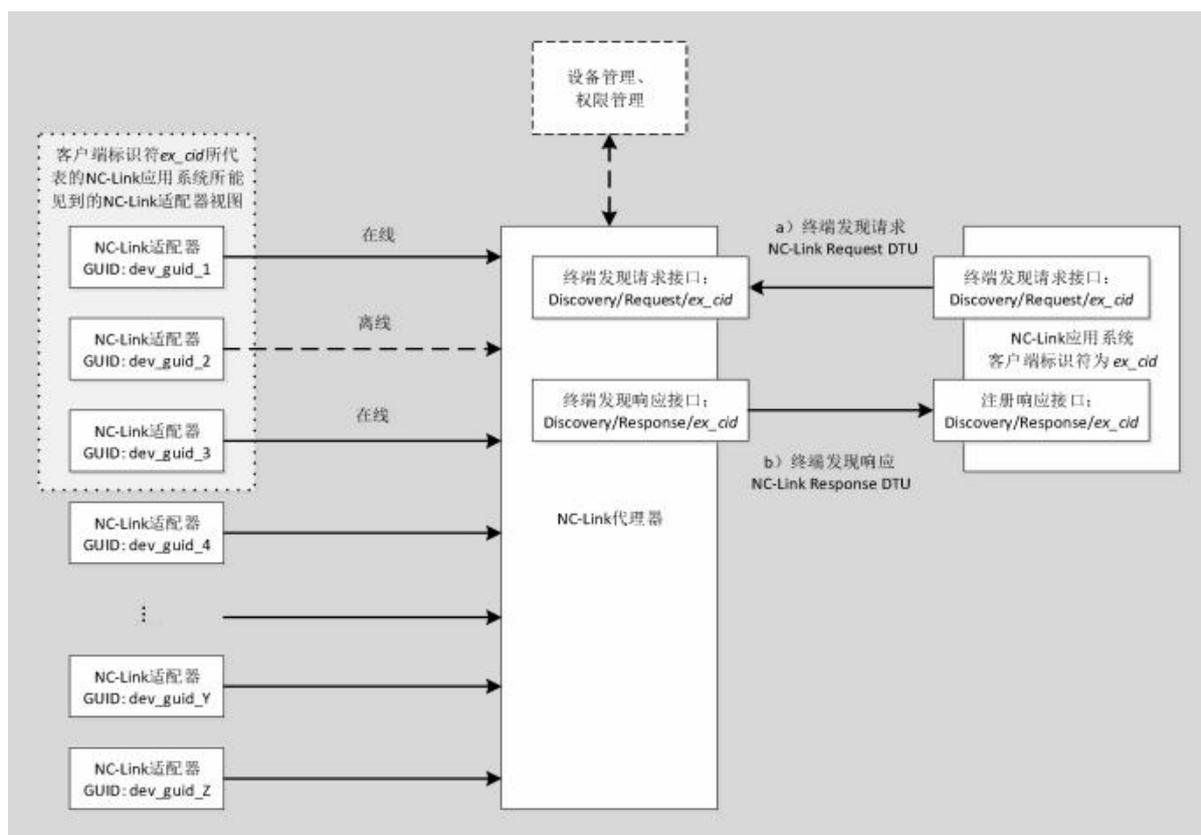


图 5 适配器终端探测流程

8.2.3.3 用法

应用系统使用本数据传输接口发送消息,代理器监听本数据传输接口接收消息。终端探测请求数据传输接口指令交互采用“请求/响应”模式。应用系统通过终端探测请求数据传输接口发送终端探测请求数据传输单元。

8.2.3.4 数据传输单元定义

终端探测请求数据传输单元定义见表 23。

表 23 终端探测请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	some_mid_001	string	应符合 8.1.3 的要求	该键值对应存在
guid	ex_cid	string	应符合 8.1.4 的要求	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.3.5 数据传输单元示例

示例:

终端标识符为“ex_cid”的应用系统向代理器进行终端探测请求,其“Request DTU”如下。

```
{
  "@id": "some_mid_001",
  "guid": "ex_cid"
}
```

8.2.4 终端探测响应数据传输接口

8.2.4.1 数据传输接口指令

终端探测响应数据传输接口指令为:Discovery/Response/*ex_cid*。其中“*ex_cid*”是发送探测请求的应用系统的终端标识符。

8.2.4.2 用途

代理器在终端探测响应数据传输接口收到终端探测请求后,向该应用系统返回当前已注册的、具有访问权限的代理器列表,列表是JSON对象描述的“Response DTU”。

8.2.4.3 用法

代理器使用本数据传输接口发送消息,应用系统监听本数据传输接口接收消息。终端探测响应数据传输接口指令交互采用“请求/响应”模式。

代理器通过本数据传输接口发送终端探测请求“Response DTU”,应用系统通过本数据传输接口接收“Response DTU”。图5适配器终端探测流程中,当标识符为“*ex_cid*”的应用系统具备访问GUID分别为*dev_uuid_1*、*dev_uuid_2*、*dev_uuid_3*的适配器的权限时,代理器将通过终端探测响应数据传输接口向应用系统的终端探测响应数据传输接口返回“Response DTU”。JSON列表中仅包含GUID为*dev_guid_1*、*dev_guid_2*、*dev_guid_3*的适配器信息。

8.2.4.4 数据传输单元定义

终端探测响应数据传输单元定义见表24。



表 24 终端探测响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可选存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
guid	<i>ex_cid</i>	string	应符合 8.1.4 的要求	该键值对应存在
deviceList	null	JSON 数组对象	键“deviceList”对应的值是一个JSON数组对象,数组中的每一个元素也是一个JSON对象,描述标识符“ <i>ex_cid</i> ”所代表的应用系统具有访问权限的适配器。表 25 给出了适配器的描述	该键值对应存在,值可为 null,值为 null 时表示无法访问任何适配器
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

适配器描述见表 25。

表 25 适配器描述

键	值(示例值)	值类型	描述	要求
guid	<i>dev_uuid_1</i>	string	值“ <i>dev_uuid_1</i> ”代表一个适配器	该键值对应存在
status	online	string	键“status”的取值应是“online”、“offline”、“unknown”三者之一,分别表示在线状态、离线状态、未知状态	该键值对应存在
lastAliveTimeS- tamp	<i>seconds</i>	string	键“lastAliveTimeStamp”的值“ <i>seconds</i> ”是代理器与“ <i>dev_uuid_1</i> ”所代表的适配器最后一次交互的时间的 string 表述,是代理器上精确到秒的本地时间	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.4.5 数据传输单元示例

示例:

代理器返回给应用系统的终端探测响应数据传输单元如下。

```
{
  "@id": "some_mid_001",
  "code": "OK",
  "guid": "ex_cid",
  "deviceList": [
    {
      "guid": "dev_uuid_1",
      "status": "online",
      "lastAliveTimeStamp": "1611287021",
      "privateInfo": null
    },
    {
      "guid": "dev_uuid_2",
      "status": "offline",
      "lastAliveTimeStamp": "1611176010",
      "privateInfo": null
    },
    {
      "guid": "dev_uuid_3",
      "status": "online",
      "lastAliveTimeStamp": "1611287160"
    }
  ]
}
```

8.2.5 版本号校对请求数据传输接口

8.2.5.1 数据传输接口指令

版本号校对请求数据传输接口指令为：Probe/Version/*dev_uuid*。其中“*dev_uuid*”是发送版本号校对请求的适配器的 GUID。

8.2.5.2 用途

版本号校对请求数据传输接口用于 GUID 为“*dev_uuid*”的适配器向代理器或应用系统发布当前的 NC-Link 模型文件版本号。

8.2.5.3 用法

适配器使用本数据传输接口发送消息，代理器使用本数据传输接口接收并存储转发消息，应用系统监听本数据传输接口接收消息。版本号校对请求数据传输接口指令交互采用“请求/响应”模式。

适配器通过本数据传输接口发送版本号校对请求数据传输单元，代理器和应用系统通过本数据传输接口接收“Request DTU”。

代理器通过本数据传输接口接收到“Request DTU”后，存储并转发给应用系统。当适配器重启或修改了模型文件时，适配器通过本数据传输接口向代理器发布新的模型文件版本号。

代理器应缓存适配器的模型文件版本号，应能感知到模型文件版本号的变化。应用系统可缓存适配器的模型文件版本号，可实现感知模型文件版本号发生变化的功能。

8.2.5.4 数据传输单元定义

版本号校对请求数据传输单元定义见表 26。

表 26 版本号校对请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对可存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
version	v1.0.1	string	该键值表示模型版本号	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.5.5 数据传输单元示例

示例：

适配器在版本号校对请求数据传输接口发布如下“Request DTU”。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "version": "v1.0.1",
  "privateInfo": null
}
```

8.2.6 版本号校对响应数据传输接口

8.2.6.1 数据传输接口指令

版本号校对响应数据传输接口指令为：`Probe/Version/Response/dev_uuid/ex_cid`。其中“*dev_uuid*”是发送版本号校对请求的适配器的 GUID，“*ex_cid*”是应用系统的终端标识符。

8.2.6.2 用途

当代理器或应用系统收到适配器的版本号校对请求数据传输单元后，可通过版本号校对响应数据传输接口向适配器返回响应，表示接收到版本号。也可不向适配器发送对应的“Response DTU”。

8.2.6.3 用法

适配器监听本数据传输接口接收消息，代理器使用本数据传输接口接收并存储转发消息，应用系统使用本数据传输接口发送消息。版本号校对响应数据传输接口指令交互采用“请求/响应”模式。

代理器或应用系统，在收到适配器的版本号校对请求数据传输单元后，通过本数据传输接口发布版本号校对响应数据传输单元，告知适配器已经收到了当前的版本号。适配器可处理或忽略该“Response DTU”。

8.2.6.4 数据传输单元定义

版本号校对响应数据传输单元是一个 JSON 对象，定义见表 27。

表 27 版本号校对响应的数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对可存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对可存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可存在
version	v1.0.1	string	表示成功收到的版本号	该键值对可存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.6.5 数据传输单元示例

示例：

代理器或应用系统在版本号校对响应数据传输接口发布如下“Response DTU”。

```
{
  "@id": "some_mid_001",
  "code": "OK",
  "guid": "dev_uuid",
  "version": "v1.0.1",
  "privateInfo": null
}
```



8.2.7 模型侦测请求数据传输接口

8.2.7.1 数据传输接口指令

模型侦测请求数据传输接口指令为:Probe/Query/Request/*dev_uuid*/*ex_cid*。其中“*dev_uuid*”是接收模型侦测的适配器的 GUID,“*ex_cid*”是应用系统的终端标识符。

8.2.7.2 用途

模型侦测请求数据传输接口用于应用系统或代理器向适配器请求当前的模型文件,以获得适配器当前的数据和服务能力。

8.2.7.3 用法

适配器监听本数据传输接口接收消息,代理器使用本数据传输接口接收并存储转发消息,应用系统使用本数据传输接口发送消息。模型侦测请求数据传输接口指令交互采用“请求/响应”模式。

当适配器通过版本号校对请求数据传输接口执行了版本号校对请求功能后,如代理器或应用系统当前没有缓存该适配器的模型文件,或收到的版本号与当前缓存的模型文件的版本号不一致,代理器或应用系统将通过模型侦测请求数据传输接口,发布模型文件获取请求。

8.2.7.4 数据传输单元定义

模型侦测请求数据传输单元定义见表 28。

表 28 模型侦测请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.7.5 数据传输单元示例

示例:

代理器或应用系统在模型侦测请求数据传输接口发布如下“Request DTU”。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "privateInfo": null
}
```



8.2.8 模型侦测响应数据传输接口

8.2.8.1 数据传输接口指令

模型侦测响应数据传输接口指令为: Probe/Query/Response/*dev_uuid*/*ex_cid*。其中“*dev_uuid*”是接收模型侦测的适配器的 GUID,“*ex_cid*”是应用系统的终端标识符。

8.2.8.2 用途

当适配器在模型侦测请求数据传输接口收到代理器发送的模型侦测请求数据传输单元并处理完成后,在模型侦测响应数据传输接口发布模型侦测响应数据传输单元,以告知请求者适配器当前的数据和服务能力。

8.2.8.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。模型侦测响应数据传输接口指令交互采用“请求/响应”模式。

当适配器在模型侦测请求数据传输接口上接收到模型侦测请求后,通过本数据传输接口发布模型侦测响应数据传输单元。

8.2.8.4 数据传输单元定义

模型侦测响应数据传输单元定义见表 29。

表 29 模型侦测响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可选存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
probe	{……}	object	值“{……}”是适配器当前的模型文件的文本描述	如键“code”的值为“OK”,该键值对应存在;如键“code”的值不为“OK”,该键值对可存在

8.2.8.5 数据传输单元示例

示例:

适配器在模型侦测响应数据传输接口发布如下“Response DTU”。

```
{
  "@id": "some_mid_001",
  "code": "OK",
  "probe": "{……}"
}
```

8.2.9 模型设置请求数据传输接口

8.2.9.1 数据传输接口指令

模型设置请求数据传输接口指令为:Probe/Set/Request/dev_uuid/ex_cid。其中“dev_uuid”是接收该消息的适配器的 GUID,“ex_cid”是发送该消息的应用系统的终端标识符。

8.2.9.2 用途

应用系统通过本数据传输接口设置/更新适配器的模型文件。应用系统应保证模型文件的有效性与合法性。

8.2.9.3 用法

适配器监听本数据传输接口接收消息,代理器使用本数据传输接口接收并存储转发消息,应用系统使用本数据传输接口发送消息。模型设置请求数据传输接口交互采用“请求/响应”模式。

应用系统应保证模型文件的有效性与合法性。适配器在接收到新的模型文件时,应进行合法性检查,可拒绝或接收设置请求。适配器在设置/更新模型文件,使得模型文件生效后,宜通过版本号校对请求数据传输接口完成一次版本号校对工作。

8.2.9.4 数据传输单元定义

模型设置请求数据传输单元定义见表 30。

表 30 模型设置请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	some_mid_001	string	应符合 8.1.3 的要求	该键值对应存在
probe	{……}	object	值“{……}”是适配器当前的模型文件的文本描述	该键值对应存在,且值是一个合法的模型文件的 JSON 对象的文本描述。适配器应校验模型文件的合法性,可拒绝模型设置请求

8.2.9.5 数据传输单元示例

示例:

代理器或应用系统在模型设置请求数据传输接口发布如下“Request DTU”。

```
{
  "@id": "some_mid_001",
  "probe": "{……}"
}
```



8.2.10 模型设置响应数据传输接口

8.2.10.1 数据传输接口指令

模型设置响应数据传输接口指令为:Probe/Set/Response/dev_uuid/ex_cid。其中“dev_uuid”是

接收该消息的适配器的 GUID,“*ex_cid*”是发送该消息的应用系统的终端标识符。

8.2.10.2 用途

当适配器在模型设置请求数据传输接口收到代理器或应用系统发布的模型设置请求数据传输单元并处理完成后,在模型设置响应数据传输接口发布模型设置响应数据传输单元,以告知请求者返回模型设置请求的处理结果。

8.2.10.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。模型设置响应数据传输接口指令交互采用“请求/响应”模式。

当适配器在模型设置请求数据传输接口上接收到模型侦测请求后,通过本数据传输接口发布模型设置响应数据传输单元。

8.2.10.4 数据传输单元定义

模型设置响应数据传输单元定义见表 31。

表 31 模型设置响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可选存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在

8.2.10.5 数据传输单元示例

示例:

适配器在模型设置响应数据传输接口发布如下“Response DTU”。

```
{
  "@id": "some_mid_001",
  "code": "OK"
}
```

8.2.11 数据查询请求数据传输接口

8.2.11.1 数据传输接口指令

数据查询请求数据传输接口指令为:Query/Request/*dev_uuid/ex_cid*。其中“*dev_uuid*”是接收该消息的适配器的 GUID,“*ex_cid*”是发送该消息的应用系统的终端标识符。

8.2.11.2 用途

用于代理器或应用系统向适配器进行数据查询。如代理器实现了历史数据服务,也可用于应用系统向代理器查询指定适配器的历史数据。

8.2.11.3 用法

适配器监听本数据传输接口接收消息,代理器使用本数据传输接口接收并存储转发消息,应用系统使用本数据传输接口发送消息。数据查询请求数据传输接口指令交互应采用“请求/响应”模式。

当适配器在该请求数据传输接口接收到数据查询请求数据传输单元后,应解析该 DTU,并依据当前有效的模型文件检查请求的合法性和有效性。并根据检查结果决定接受或拒绝数据查询请求。当该应用系统无权查询该项数据或查询指令错误时,检查结果为拒绝。无论是接受还是拒绝,适配器应通过数据查询响应数据传输接口向应用系统给出“Response DTU”。

8.2.11.4 数据传输单元定义

数据查询请求数据传输单元定义见表 32。单值类型数据项、字典类型数据项及列表类型数据项查询请求参数形式化定义分别见图 6、图 7 和图 8。

表 32 数据查询请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
ids	[{"id": "010203"}, {"id": "010214"}, "params": { "....." }]	array	数组中每个元素应是 JSON 对象,一个 JSON 对象描述如何查询一个数据项。对应于不同的三种 NC-Link 数据类型,分别有三种不同的查询请求参数形式: a) 单值数据类型,其要求应与图 6 相符; b) 字典数据类型,其要求应与图 7 相符; c) 列表数据类型,其要求应与图 8 相符	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

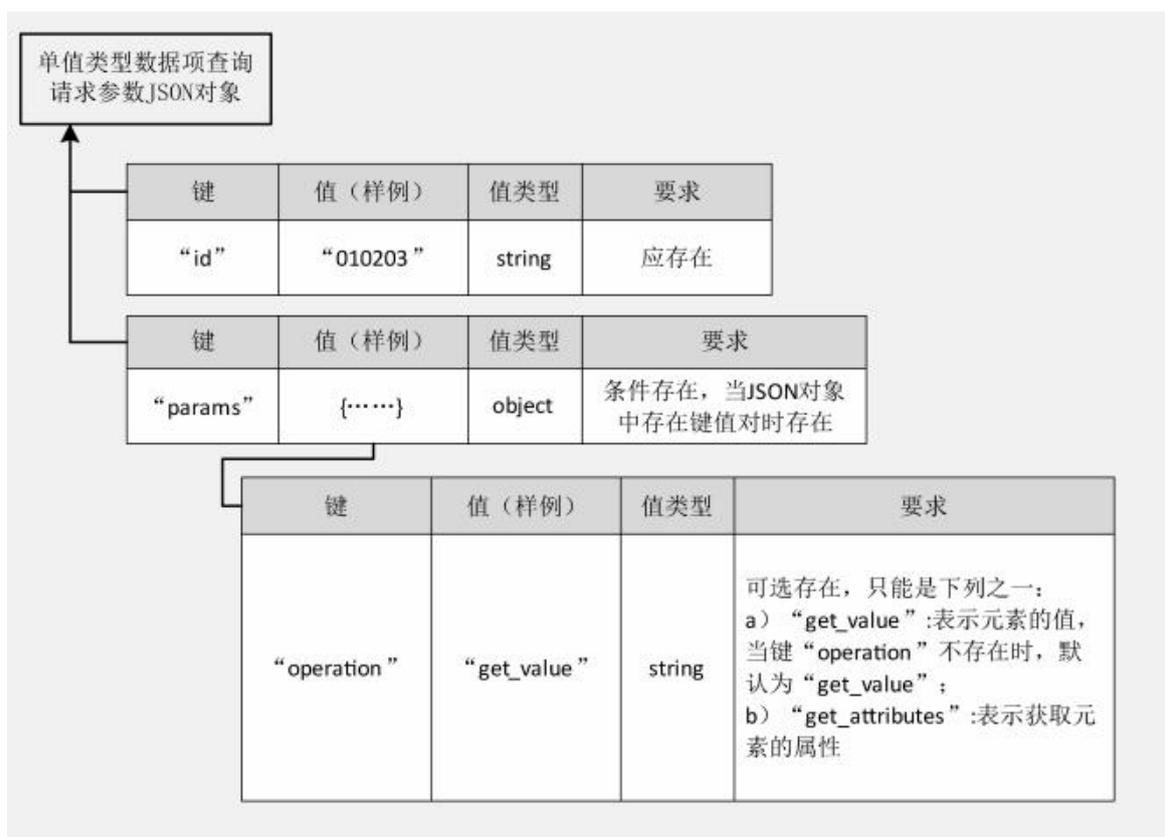


图6 单值类型数据项查询请求参数形式化定义

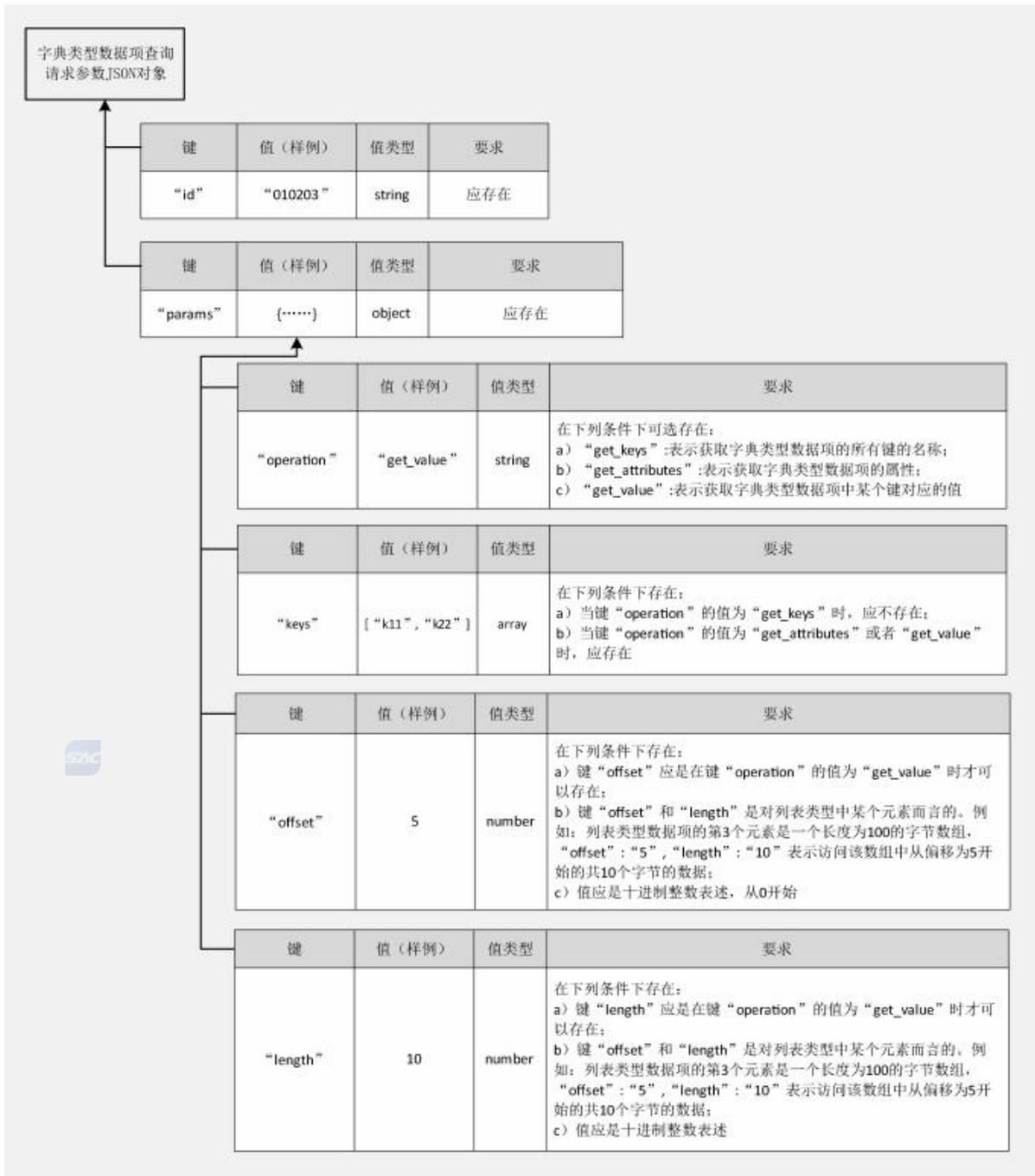


图 7 字典类型数据项查询请求参数形式化定义

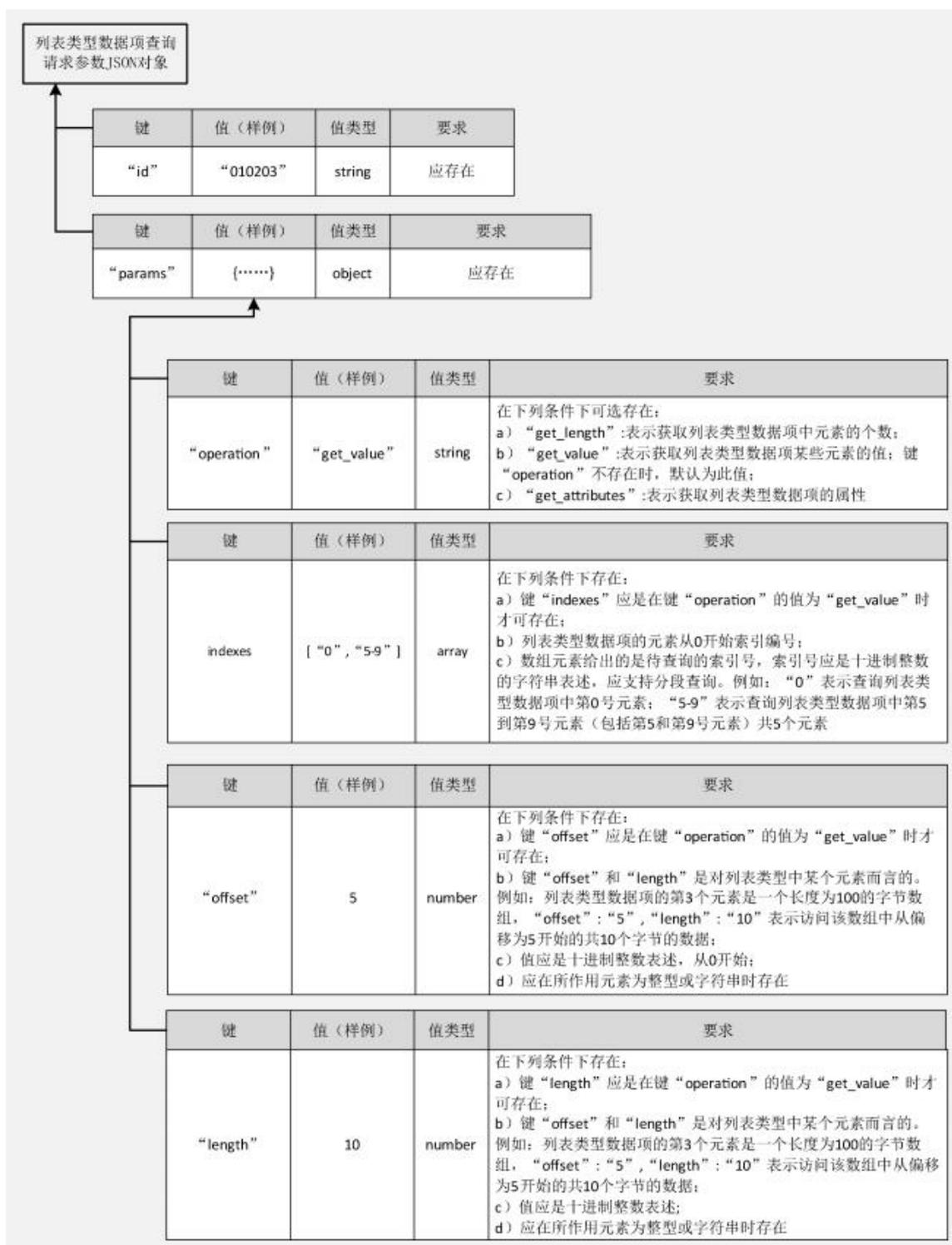


图 8 列表类型数据项查询请求参数形式化定义

8.2.11.5 数据传输单元示例

示例 1：单值类型数据项查询请求

终端标识符为“ex_cid”的应用系统查询终端标识符为“dev_uuid”的适配器中“id”分别为“010203”和“010204”的两

个单值类型数据项。

应用系统在数据查询请求数据传输接口发布如下“Request DTU”。

```
{
  "@id" : "some_mid_x1",
  "guid" : "dev_uuid",
  "ids" : [{"id" : "010203"}, {"id" : "010204"}]
}
```

示例 2: 字典类型数据项查询请求

终端标识符为“ex_cid”的应用系统查询终端标识符为“dev_uuid”的适配器中“id”为“011213”的字典类型数据项。

a) 查询“id”为“011213”的字典类型数据项的所有键；

应用系统在数据查询请求数据传输接口发布如下“Request DTU”。

```
{
  "@id" : "some_mid_x2",
  "guid" : "dev_uuid",
  "ids" : [ {"id" : "011213", "params" : {"operation" : "get_keys"}} ]
}
```

b) 查询“id”为“011213”的字典类型数据项中键分别为 k1、k2 的数据的属性；

应用系统在数据查询请求数据传输接口发布如下“Request DTU”：

```
{
  "@id" : "some_mid_x3",
  "guid" : "dev_uuid",
  "ids" : [ {"id" : "011213", "params" : {"operation" : "get_attributes", "keys":["k1", "k2"]} } ]
}
```

c) 查询“id”为“011213”的字典类型数据项中键分别为 k1、k2 的值。

应用系统在数据查询请求数据传输接口发布如下“Request DTU”：

```
{
  "@id" : "some_mid_x4",
  "guid" : "dev_uuid",
  "ids" : [ {"id" : "011213", "params" : {"operation": "get_value", "keys":["k1", "k2"]} } ]
}
```

示例 3: 列表类型数据项查询请求

终端标识符为“ex_cid”的应用系统查询终端标识符为“dev_uuid”的适配器中“id”为“012223”的列表类型数据项。

a) 查询“id”为“012223”的列表类型数据项的长度；

应用系统在数据查询请求数据传输接口发布如下“Request DTU”：

```
{
  "@id" : "some_mid_x5",
  "guid" : "dev_uuid",
  "ids" : [ {"id" : "012223", "params" : {"operation": "get_length"}} ]
}
```

b) 查询“id”为“012223”的列表类型数据项中索引号为 0、5、6、7、8 的元素。

应用系统在数据查询请求数据传输接口发布如下“Request DTU”：

```
{
  "@id" : "some_mid_x6",
  "guid" : "dev_uuid",
  "ids" : [ {"id" : "012223", "params": {"operation": "get_value", "indexes":["0", "5-8"]} } ]
}
```

}

8.2.12 数据查询响应数据传输接口

8.2.12.1 数据传输接口指令

数据查询响应数据传输接口指令为:Query/Response/*dev_uuid*/*ex_cid*。其中“*dev_uuid*”是发送该消息的适配器的 GUID,“*ex_cid*”是接收该消息的应用系统的终端标识符。

8.2.12.2 用途

当适配器在数据查询请求数据传输接口收到用于代理器或应用系统发布的数据查询请求后,通过本数据传输接口向请求查询的代理器或应用系统返回查询结果响应。如代理器实现了历史数据服务,本数据传输接口也可用于代理器向应用系统返回查询结果响应。

8.2.12.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。数据查询响应数据传输接口指令交互应采用“请求/响应”模式。

当适配器接收到数据查询请求数据传输单元后,应解析该 DTU,并依据当前有效的模型文件检查请求的合法性和有效性。并根据检查结果决定接受或拒绝数据查询请求。当该应用系统无权查询该项数据或查询指令错误时,检查结果为拒绝接受。无论是接受还是拒绝,适配器应通过数据查询响应数据传输接口向应用系统给出“Response DTU”。

8.2.12.4 数据传输单元定义

数据查询响应数据传输单元定义见表 33。单值类型数据项、字典类型数据项及列表类型数据项查询响应参数形式化定义分别见图 9、图 10 和图 11。

表 33 数据查询响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

表 33 数据查询响应数据传输单元定义 (续)

键	值(示例值)	值类型	描述	要求
values	[{“id”:“010203”, “values”:[12.5], “code”:“OK”}, {“id”:“010214”, “code”:“NG”, “code”:“Permission Denied”}]	array	数组中每个元素应是 JSON 对象,一个 JSON 对象描述针对一个数据项的查询结果。对应于不同的三种 NC-Link 数据类型,分别有三种不同的查询结果参数形式: a) 单值类型,其要求应与图 9 相符; b) 字典类型,其要求应与图 10 相符; c) 列表类型,其要求应与图 11 相符	该键值对应存在

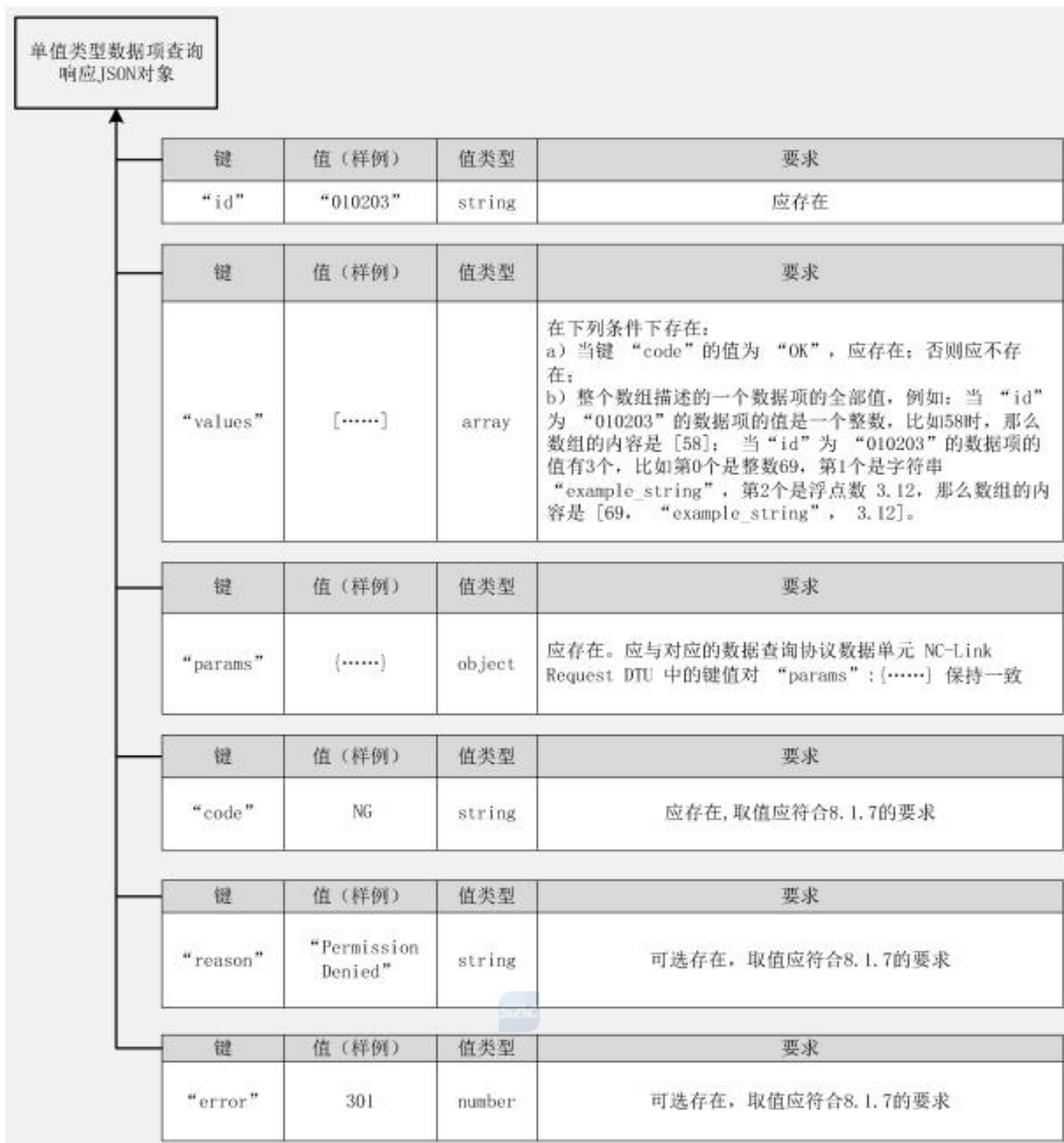


图 9 单值类型数据项查询响应参数形式化定义



图 10 字典类型数据项查询响应参数形式化定义



图 11 列表类型数据项查询响应参数形式化定义

8.2.12.5 数据传输单元示例

示例 1: 单值类型数据项查询响应

终端标识符为“ex_cid”的应用系统查询终端标识符为“dev_uuid”的适配器中“id”分别为“010203”和“010204”的两个单值类型数据项。其中，“id”为“010203”的单值类型数据项的当前值为 [2, "str1", [1, 2, 3], 100.23]，“id”为“010204”的单值类型数据项的当前值不可用。

适配器在数据查询响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x1",
  "guid" : "dev_uuid",
  "values" : [
    { "id" : "010203", "values" : [2, "str1", [1, 2, 3], 100.23], "code" : "OK" },
    { "id" : "010204", "code" : "NG", "reason" : " Unavailable", "error" : 301 }
  ]
}
```

示例 2：字典类型数据项查询响应

终端标识符为“*ex_cid*”的应用系统查询终端标识符为“*dev_uuid*”的适配器中“id”为“011213”的字典类型数据项。该字典类型数据项有三个键，分别为“k1”“k2”“k3”，对应的属性分别为“key_attr_1”“key_attr_2”“key_attr_3”，对应的值分别为 number 值 1234，string“kv_string”，字节数组 [1, 2, 3, 4]。

a) 查询“id”为“011213”的字典类型数据项的所有键的响应

适配器在数据查询响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x2",
  "guid" : "dev_uuid",
  "values" : [
    {
      "id" : "011213",
      "code" : "OK",
      "params" : { "operation" : "get_keys" },
      "values" : [ "k1", "k2", "k3" ]
    }
  ]
}
```

b) 查询“id”为“011213”的字典类型数据项中键分别为 k1、k2 的数据的属性的响应

适配器在数据查询响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x3",
  "guid" : "dev_uuid",
  "values" : [
    {
      "id" : "011213",
      "code" : "OK",
      "params" : { "operation" : "get_attributes", "keys": ["k1", "k2"] },
      "values" : [ "key_attr_1", "key_attr_2" ]
    }
  ]
}
```

- c) 查询“id”为“011213”的字典类型数据项中键分别为 k1、k2、k3 的值适配器在数据查询响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x4",
  "guid" : "dev_uuid",
  "values" : [
    {
      "id" : "011213",
      "code" : "OK",
      "params" : {"operation" : "get_value", "keys":["k1", "k2", "k3"]},
      "values" : [1234, "kv_string", [1, 2, 3, 4]]
    }
  ]
}
```

示例 3: 列表类型数据项查询响应

终端标识符为“ex_cid”的应用系统查询终端标识符为“dev_uuid”的适配器中“id”为“012223”的列表类型数据项,该数据项的值有 10 个元素,示例值为 [100,234, 20,30,40,50,"str_v_6", [1.1, 2.2, 3.3],8,9,10]。

- a) 查询“id”为“012223”的列表类型数据项的长度的响应适配器在数据查询响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x5",
  "guid" : "dev_uuid",
  "values" : [
    {
      "id" : "012223",
      "code" : "OK",
      "params" : {"operation" : "get_length"},
      "values" : [10]
    }
  ]
}
```

- b) 查询“id”为“012223”的列表类型数据项中索引号为 0、4、5、6、7 的元素的响应适配器在数据查询响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x6",
  "guid" : "dev_uuid",
  "values" : [
    {
      "id" : "012223",
      "code" : "OK",
      "params" : {"operation" : "get_value", "indexes" : ["0", "4-7"]},
```

```

    "values" : [100.234, 50, "str_v_6", [1.1, 2.2, 3.3], 8]
  }
]
}

```

8.2.13 数据设置请求数据传输接口

8.2.13.1 数据传输接口指令

数据设置请求数据传输接口指令为: Set/Request/*dev_uuid*/*ex_cid*。其中“*dev_uuid*”是接收该消息的适配器的 GUID,“*ex_cid*”是发送该消息的应用系统的终端标识符。

8.2.13.2 用途

用于代理器或应用系统向适配器设置指定数控机床模型文件中指定数据项的值。

8.2.13.3 用法

适配器监听本数据传输接口接收消息,代理器使用本数据传输接口接收并存储转发消息,应用系统使用本数据传输接口发送消息。数据设置请求数据传输接口指令交互采用“请求/响应”模式。

当适配器在该请求数据传输接口接收到数据设置请求数据传输单元后,应解析该 DTU,并依据当前有效的模型文件检查请求的合法性和有效性。并根据检查结果决定接受或拒绝数据设置请求。当该应用系统无权设置该项数据或设置指令错误时,检查结果为拒绝。无论是接受还是拒绝,适配器应通过数据设置请求数据传输接口向应用系统给出“Response DTU”。

8.2.13.4 数据传输单元定义

数据设置请求数据传输单元定义见表 34。单值类型数据项、字典类型数据项及列表类型数据项设置请求参数形式化定义分别见图 12、图 13 和图 14。

表 34 数据设置请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
values	<pre> [{ "id": "010203", "params": { } } { "id": "010214", "params": { } }] </pre>	array	<p>数组中每个元素应是 JSON 对象,一个 JSON 对象描述如何设置一个数据项。对应于不同的三种 NC-Link 数据类型,分别有三种不同的设置请求参数形式:</p> <p>a) 单值类型,其要求应与图 12 相符;</p> <p>b) 字典类型,其要求应与图 13 相符;</p> <p>c) 列表类型,其要求应与图 14 相符</p>	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

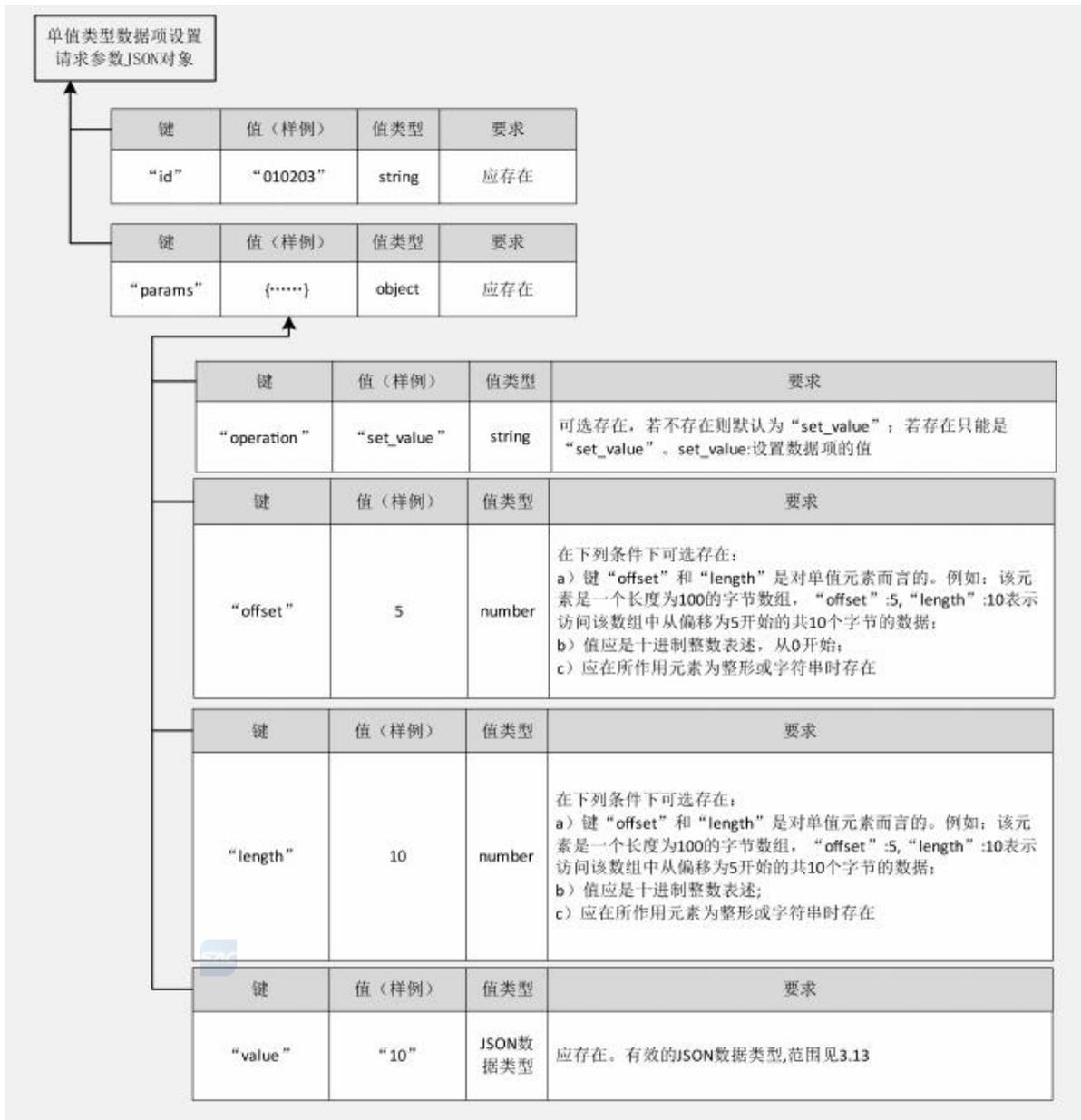


图 12 单值类型数据项设置请求参数形式化定义



图 13 字典类型数据项设置请求参数形式化定义

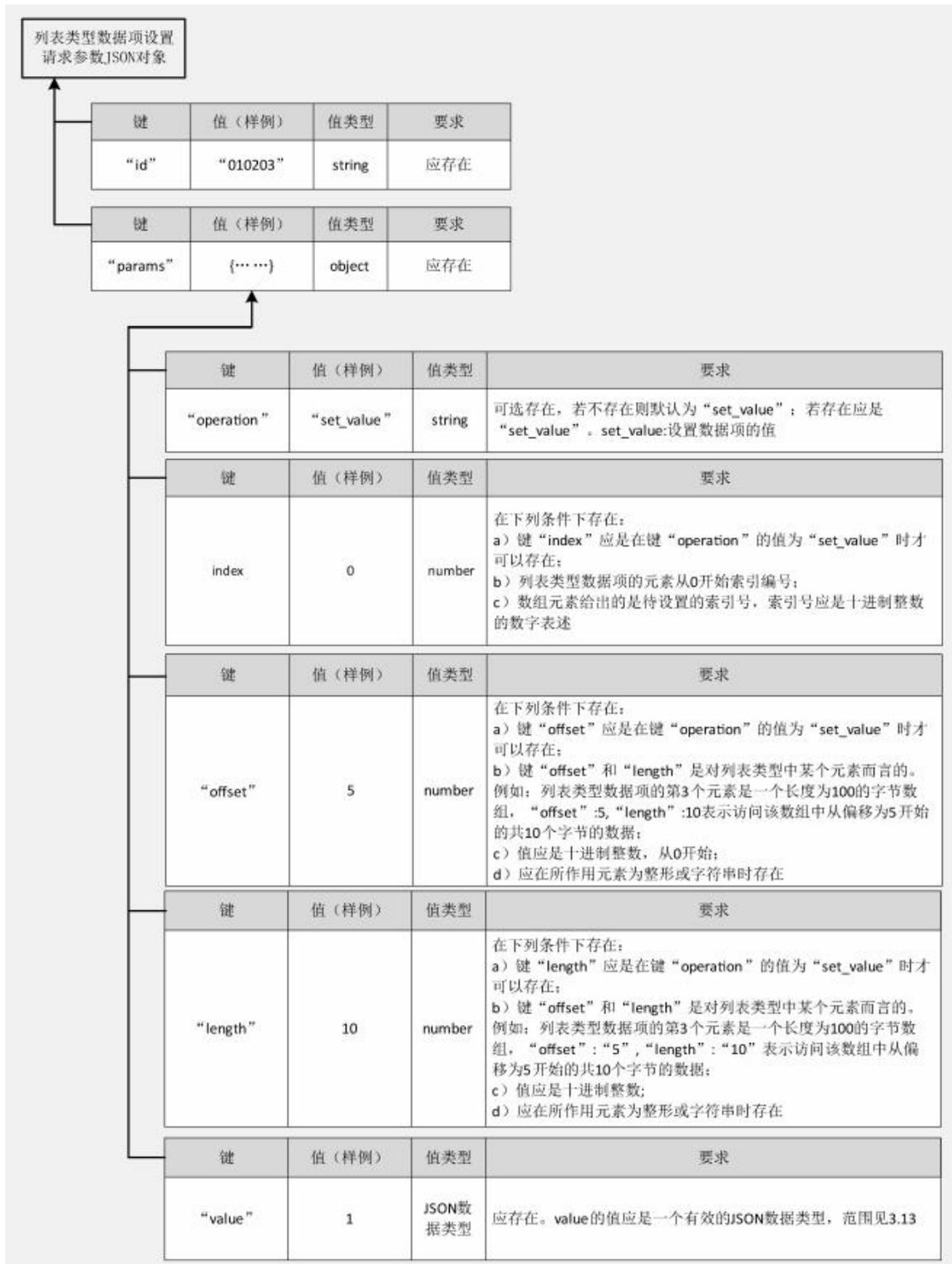


图 14 列表类型数据项设置请求参数形式化定义

8.2.13.5 数据传输单元示例

示例 1: 单值类型数据项设置请求

终端标识符为“*ex_cid*”的应用系统设置终端标识符为“*dev_uuid*”的适配器中数据项“000006”的值为 12.5、数据项“000008”的值为 22.3。

应用系统在数据设置请求数据传输接口发布如下“Request DTU”。

```
{
  "@id" : "some_mid_x1",
  "guid" : "dev_uuid",
  "values" : [
    {"id": "000006", "params": {"value": 12.5}},
    {"id": "000008", "params": {"operation": "set_value", "value": 22.3}}
  ]
}
```

示例 2: 字典类型数据项设置请求

终端标识符为“*ex_cid*”的应用系统查询终端标识符为“*dev_uuid*”的适配器中“id”为“011213”的字典类型数据项。

- a) 设置“id”为“011213”的字典类型数据项的键 k1 的值为 2,设置“id”为“000003”的字典类型数据项的键 k1 的值为 2。

应用系统在数据设置请求数据传输接口发布如下“Request DTU”。

```
{
  "@id" : "some_mid_x2",
  "guid" : "dev_uuid",
  "values" : [
    {"id": "011213", "params": {"operation": "set_value", "key": "k1", "value": 2}},
    {"id": "000003", "params": {"key": "k1", "value": 2}}
  ]
}
```

- b) 在“id”为“011213”的字典类型数据项中添加键“AAA”,值为“111”。

应用系统在数据设置请求数据传输接口发布如下“Request DTU”。

```
{
  "@id" : "some_mid_x3",
  "guid" : "dev_uuid",
  "values" : [
    {"id": "011213", "params": {"operation": "add", "key": "AAA", "value": "111"}},
  ]
}
```

- c) 删除“id”为“011213”的字典类型数据项中的键“AAA”

应用系统在数据设置请求数据传输接口发布如下“Request DTU”。

```
{
  "@id" : "some_mid_x4",
  "guid" : "dev_uuid",
  "values" : [
    {"id": "011213", "params": {"operation": "delete", "key": "AAA"}},
  ]
}
```

}

示例 3：列表类型数据项设置请求

终端标识符为“*ex_cid*”的应用系统查询终端标识符为“*dev_uuid*”的适配器中“id”为“012223”的列表类型数据项。

设置“id”为“012223”的列表类型数据项中第零号元素从偏移为 5 开始的共 10 个字节的数据为 16。

应用系统在数据设置请求数据传输接口发布如下“Request DTU”。

```
{
  "@id" : "some_mid_x5",
  "guid" : "dev_uuid",
  "values" : [
    { "id" : "012223",
      "params" : {
        "operation" : "set_value",
        "index" : 0,
        "offset" : 5,
        "length" : 10,
        "value" : 16
      }
    }
  ]
}
```

8.2.14 数据设置响应数据传输接口

8.2.14.1 数据传输接口指令

数据设置响应数据传输接口指令为: Set/Response/*dev_uuid*/*ex_cid*。其中“*dev_uuid*”是发送该消息的适配器的 GUID,“*ex_cid*”是接收该消息的应用系统的终端标识符。

8.2.14.2 用途

当适配器在数据设置请求数据传输接口收到用于代理器或应用系统发布的数据设置请求后。通过本数据传输接口向代理器或应用系统返回设置结果响应。

8.2.14.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。数据设置响应数据传输接口指令交互应采用“请求/响应”模式。

当适配器接收到数据设置请求数据传输单元后,应解析该 DTU,并依据当前有效的模型文件检查请求的合法性和有效性。并根据检查结果决定接受或拒绝数据设置请求。当该应用系统无权设置该项数据或设置指令错误时,检查结果为拒绝。无论是接受还是拒绝,适配器应通过数据设置响应数据传输接口向应用系统给出“Response DTU”。

8.2.14.4 数据传输单元定义

数据设置响应数据传输单元定义见表 35。数据项设置响应参数形式化定义见图 15。

表 35 数据设置响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	"some_mid_001	string	应符合 8.1.3 的要求	该键值对应存在
guid	"dev_uuid	string	应符合 8.1.4 的要求	该键值对应存在
results	[{ "id": "010203", "code": "OK" }, { "id": "010214", "code": "NG", "reason": "Permission Denied", "error": 301, "params": {·····} }]	array	数组中每个元素应是 JSON 对象, 一个 JSON 对象描述针对一个数据项的设置结果。对于不同的三种 NC-Link 数据类型, 共有一种设置结果参数形式, 其要求应与图 15 相符	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

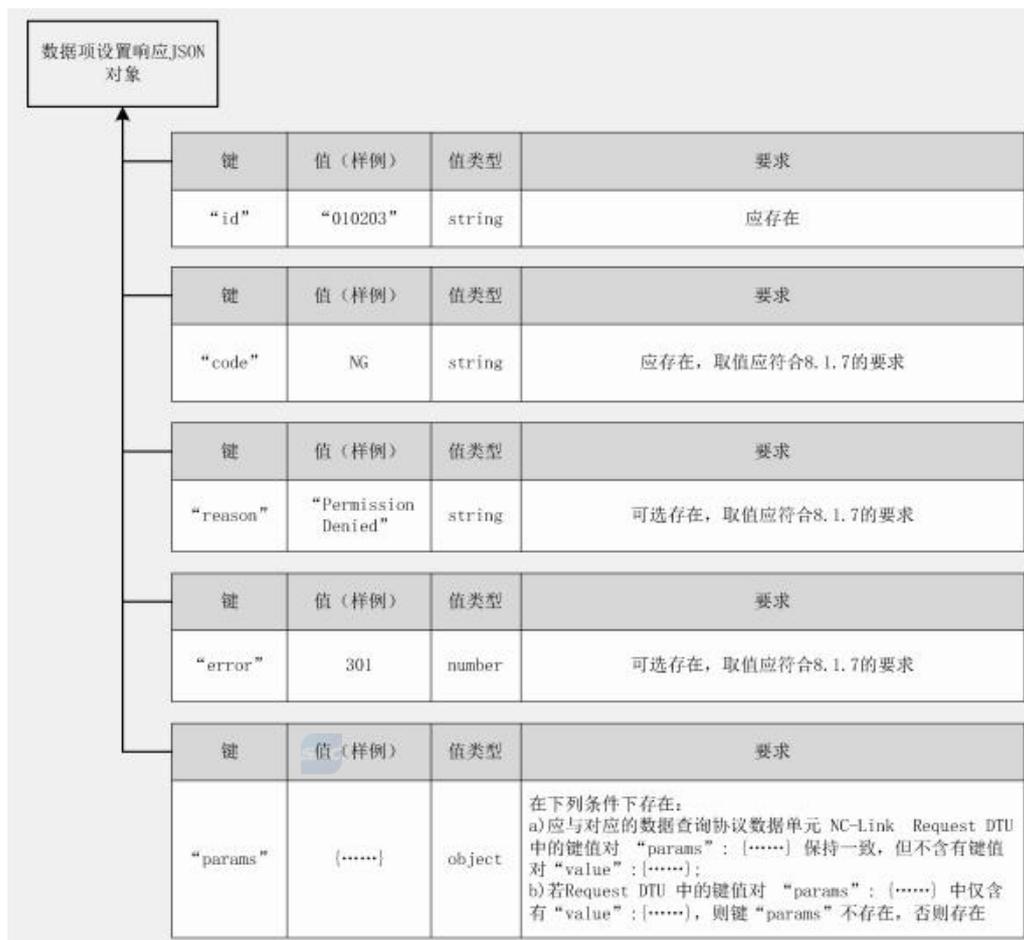


图 15 数据项设置响应参数形式化定义

8.2.14.5 数据传输单元示例

示例 1: 单值类型数据项设置响应

终端标识符为“*ex_cid*”的应用系统设置终端标识符为“*dev_uuid*”的适配器中数据项“000006”的值为 12.5、数据项“000008”的值为 22.3。其中，“id”为“000006”的单值类型数据项设置成功，“id”为“000006”的单值类型数据项设置时访问权限错误。

适配器在数据设置响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x1",
  "guid" : "dev_uuid",
  "results" : [
    { "id": "000006", "code": "OK" },
    { "id": "000008", "code": "NG", "reason": "Permission Denied", "error": 301,
      "params" : { "operation": "set_value" }
    }
  ]
}
```

示例 2: 字典类型数据项设置响应

终端标识符为“*ex_cid*”的应用系统设置终端标识符为“*dev_uuid*”的适配器中“id”为“011213”的字典类型数据项。成功设置“id”为“011213”的字典类型数据项的键 k1、k2 的值。

适配器在数据设置响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x2",
  "guid" : "dev_uuid",
  "results" : [
    {
      "id" : "011213",
      "code" : "OK",
      "params" : { "operation" : "set_values", "key": ["k1", "k2"] }
    }
  ]
}
```

示例 3: 列表类型数据项设置响应

终端标识符为“*ex_cid*”的应用系统查询终端标识符为“*dev_uuid*”的适配器中“id”为“012223”的列表类型数据项。成功设置“id”为“012223”的列表类型数据项中第零号元素从偏移为 5 开始的共 10 个字节的数据。

应用系统在数据设置响应数据传输接口发布如下“Response DTU”。

```
{
  "@id" : "some_mid_x5",
  "guid" : "dev_uuid",
  "values" : [
    { "id" : "012223",
```

```

    "code": "OK",
    "params": {
      "operation": "set_value",
      "index": ["0"],
      "offset": 5,
      "length": 10
    }
  }
]
}

```

8.2.15 数据采样数据传输接口

8.2.15.1 数据传输接口指令

数据采样数据传输接口指令为: `Sample/dev_uuid/sample_channel_id`。其中“`dev_uuid`”是发送该消息的适配器的 GUID,“`sample_channel_id`”是该适配器中模型文件给出的合法的采样通道标识符。

8.2.15.2 用途

数据采样数据传输接口指令用于获取数控机床终端的采样数据,适配器每个周期均会向代理器或应用系统发布采样数据。

8.2.15.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。如代理器提供历史数据服务,则代理器也使用本数据传输接口发送消息。数据采样指令的交互采用“订阅/发布”模式。

实施者可自主在数控机床模型文件中定义多个采样通道,每个采样通道可有多个数据采样点,每个采样数据点对应数控机床终端中的一个数据源。实施者可自主在数控机床模型文件中定义发布周期。

8.2.15.4 数据传输单元定义

数据采样数据传输单元定义见表 36。

表 36 数据采样数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对可存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对存在
id	<i>cn1</i>	string	标识采样通道。对应数控机床文件中的采样通道 id	该键值对存在。长度小于等于 16 字节。应与数控机床模型文件中的定义一致

表 36 数据采集数据传输单元定义（续）

键	值(示例值)	值类型	描述	要求
beginTime	1535642875900	string	本次采样数据的开始采样时间	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在
data	[{"data": [414.262, 414.262, 414.262]}, {"data": [414.262, 414.262, 414.262]}, {"data": [0, 0, 0]}, {"data": [0, 0, 0]},]	array	一个采样通道中所有的采样数据。该数组中的每一个 JSON 对象标识一个采样点。采样点使用 "data": [.....] 来表示采样数据集	该键值对应存在。采样点数据集的顺序和数目应与采样通道模型文件的采样点一致

8.2.15.5 数据传输单元示例

示例：

适配器(经由代理器)向应用系统提供了采样通道 id 为“cn1”，起始采样时间为“1535642875900”的四个采样点的值。每个采样点对应一个数据项。其中每个数据项有 3 个值，3 个值分别表示该采样点在不同时间点的实际值。

```
{
  "id": "cn1",
  "guid": "dev_uuid",
  "beginTime": "1535642875900",
  "data": [
    {"data": [414.262, 414.262, 414.262]},
    {"data": [414.262, 414.262, 414.262]},
    {"data": [0, 0, 0]},
    {"data": [0, 0, 0]},
  ]
}
```

8.2.16 状态通知数据传输接口

8.2.16.1 数据传输接口指令

状态通知数据传输接口指令为:Notify/State/tid。其中，“tid”是终端标识，可是适配器的 GUID 或应用系统的终端标识。

8.2.16.2 用途

状态通知数据传输接口用于将终端状态发生变更的消息通知其他终端。

8.2.16.3 用法

终端监听指定终端标识 tid 对应的状态通知数据传输接口接收消息。代理器在检测到终端标识 tid

所对应的终端的连接状态发生变更时,向监听该终端状态通知数据传输接口的终端发送状态通知消息。

8.2.16.4 数据传输单元定义

状态通知数据传输单元定义见表 37。

表 37 状态通知数据传输单元定义

键	值(示例值)	值类型	描述	要求
tid	<i>dev_001</i>	string	状态发生变更的终端的标识,与指令中的“tid”一致	该键值对可存在
state	offline	string	“tid”所指定的终端状态变更后的状态,只能是“offline”或“online”,其中,“offline”表示终端状态由在线变为离线;“online”表示终端状态由离线变为在线	该键值对可存在

8.2.16.5 数据传输单元示例

示例:

标识“*dev_001*”对应的终端设备状态发生变化,状态由在线变为离线。

```
{
  "tid": "dev_001",
  "state": "offline"
}
```

8.2.17 动态采样注册请求数据传输接口

8.2.17.1 数据传输接口指令

动态采样注册请求数据传输接口指令为: Register/Sample/Request/*dev_uuid*/*ex_cid*。其中“*dev_uuid*”是接收该消息的适配器的 GUID,“*ex_cid*”是发送该消息的应用系统的终端标识符。

8.2.17.2 用途

用于代理器或应用系统向适配器注册动态采样。动态采样使应用系统或代理器在不更新数控机床模型文件的情况下,具有增加采样通道的能力。适配器依据动态采样注册请求,动态分配采样通道资源,启动采样任务。

8.2.17.3 用法

适配器监听本数据传输接口接收消息,代理器使用本数据传输接口接收并存储转发消息,应用系统使用本数据传输接口发送消息。动态采样注册请求数据传输接口指令交互应采用“请求/响应”模式。

当适配器在该请求数据传输接口接收到动态采样注册请求“Request DTU”后,应解析该 DTU,依据当前有效的模型文件检查请求的合法性和有效性,并根据检查结果决定接受或拒绝动态采样注册请求。当该应用系统无权添加动态采样通道,或动态采样注册请求格式错误时,适配器拒绝该请求。无论是接受还是拒绝,适配器均通过动态采样注册响应数据传输接口向请求注册的应用系统给出“Response DTU”。DTU 中包含动态采样通道的采样配置,适配器依据此配置,动态分配采样通道资源。

源,启动采样任务。

8.2.17.4 数据传输单元定义

动态采样注册请求数据传输单元定义见表 38。

表 38 动态采样注册请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
sampleInterval	20000	number	内部采样周期,用来描述适配器采集机床数据的时间间隔,缺省单位为毫秒	该键值对应存在
uploadInterval	20000	number	外部采样周期,用来描述适配器向代理器发送数据的时间间隔,缺省单位为毫秒	该键值对应存在
ids	[{"id": "010302"}, {"id": "010305"}, {"id": "010306"}, {"id": "010307"}]	array	数组中每个元素应是 JSON 对象,一个 JSON 对象描述该通道中采集的数据的 id	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.17.5 数据传输单元示例

示例:

代理器或应用系统在动态采样注册请求数据传输接口发布如下“Request DTU”,用于采集“id”为“010302”,“010305”,“010306”,“010307”的数据项的数据。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid "
  "sampleInterval": 20000,
  "uploadInterval": 20000,
  "ids": [
    {"id": "010302"},
    {"id": "010305"},
    {"id": "010306"},
    {"id": "010307"}
  ]
}
```

8.2.18 动态采样注册响应数据传输接口

8.2.18.1 数据传输接口指令

动态采样注册响应数据传输接口指令为：`Reigster/Sample/Response/dev_uuid/ex_cid`。其中“`dev_uuid`”是发送该消息的适配器的 GUID，“`ex_cid`”是接收该消息的应用系统的终端标识符。

8.2.18.2 用途

用于适配器向应用系统发送动态采样注册响应的 DTU。适配器通过本数据传输接口向代理器或应用系统返回动态采样注册结果。

8.2.18.3 用法

适配器使用本数据传输接口发送消息，代理器使用本数据传输接口接收并存储转发消息，应用系统监听本数据传输接口接收消息。动态采样注册响应数据传输接口指令交互采用“请求/响应”模式。

当适配器接收到动态采样注册请求数据传输单元后，适配器应通过动态采样注册响应数据传输接口向应用系统给出“Response DTU”。适配器若在规定的时间内未能收到终端标识符为“`ex_cid`”的应用系统的 DTU，则会主动注销该采样通道。

8.2.18.4 数据传输单元定义

 动态采样注册响应数据传输单元定义见表 39。

表 39 动态采样注册响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<code>some_mid_001</code>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<code>dev_uuid</code>	string	应符合 8.1.4 的要求	该键值对应存在
id	<code>channel_01</code>	string	动态采样通道对象的标识号	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可选存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.18.5 数据传输单元示例

示例：

适配器在动态采样注册响应数据传输接口发布如下“Response DTU”，用于表示设置成功，并把该动态采样数据通道“id”设置为“`channel_01`”。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "id": "channel_01",
```

```
"code": "OK"
}
```

8.2.19 动态采样注销请求数据传输接口

8.2.19.1 数据传输接口指令

动态采样注销请求数据传输接口指令为: `Unregister/Sample/Request/dev_uuid/ex_cid`。其中“`dev_uuid`”是接收该消息的适配器的 GUID,“`ex_cid`”是发送该消息的应用系统的终端标识符。

8.2.19.2 用途

用于代理器或应用系统向适配器注销动态采样通道,注销后,采样任务终止,被注销的动态采样通道资源被回收。

8.2.19.3 用法

适配器监听本数据传输接口接收消息,代理器使用本数据传输接口接收并存储转发消息,应用系统使用该数据传输接口发送消息。动态采样注销请求数据传输接口指令交互应采用“请求/响应”模式。

当适配器在该请求数据传输接口接收到动态采样注销请求数据传输单元后,应解析该 DTU,并依据当前有效的模型文件检查请求的合法性和有效性。并根据检查结果决定接受或拒绝动态采样注销请求。当该应用系统无权注销该动态采样通道或动态采样注册请求消息格式错误时,检查结果为拒绝。无论是接受还是拒绝,适配器均通过动态采样注销响应数据传输接口向应用系统给出“Response DTU”。

8.2.19.4 数据传输单元定义

动态采样注销请求数据传输单元定义见表 40。

表 40 动态采样注销请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
id	<i>channel_01</i>	string	动态采样对象的标识号	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.19.5 数据传输单元示例

示例:

代理器或应用系统在动态采样注销请求数据传输接口发布如下“Request DTU”,用于注销“guid”为“`dev_uuid`”的适配器中“id”为“`channel_01`”的动态采样通道。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid ",
  "id": "channel_01"
}
```

8.2.20 动态采样注销响应数据传输接口

8.2.20.1 数据传输接口指令

动态采样注销响应数据传输接口指令为:Unreigster/Sample/Response/*dev_uuid*/*ex_cid*。其中“*dev_uuid*”是发送该消息的适配器的 GUID,“*ex_cid*”是接收该消息的应用系统的终端标识符。

8.2.20.2 用途

用于适配器向应用系统发送动态采样注销响应的 DTU。适配器通过本数据传输接口向代理器或应用系统返回动态采样注销结果。

8.2.20.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。动态采样注销响应数据传输接口指令交互应采用“请求/响应”模式。

当适配器接收到动态采样注销请求数据传输单元后,通过动态采样注销响应数据传输接口向应用系统给出“Response DTU”。

8.2.20.4 数据传输单元定义

动态采样注销响应数据传输单元定义见表 41。

表 41 动态采样注销响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可选存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.20.5 数据传输单元示例

示例:

适配器在动态采样注销响应数据传输接口发布如下“Request DTU”,用于取消“guid”为“*dev_uuid*”的适配器中“id”为“*channel_01*”的动态采样通道。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "code": "OK"
}
```

8.2.21 动态数据采样数据传输接口

8.2.21.1 数据传输接口指令

动态数据采样数据传输接口指令为: `Sample/dev_uuid/ex_cid/sample_channel_id`。其中“`dev_uuid`”是发送该消息的适配器的 GUID,“`ex_cid`”是接收该消息的应用系统的终端标识符,“`sample_channel_id`”是该适配器中模型文件给出的合法的采样通道标识符。

8.2.21.2 用途

动态数据采样数据传输接口指令用于获取数控机床终端的动态采样数据。在完成动态采样注册后,适配器每个周期均会向代理器或应用系统发布动态采样数据。

8.2.21.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。如代理器提供历史数据服务,则代理器也使用本数据传输接口发送消息。动态数据采样数据传输接口指令的交互采用“订阅/发布”模式。实施者可自主在数控机床模型文件中定义多个采样通道,每个采样通道可有多个数据采样点,每个采样数据点对应数控机床终端中的一个数据源。实施者可自主在数控机床模型文件中定义发布周期。

8.2.21.4 数据传输单元定义

动态数据采样数据传输单元定义见表 42。

表 42 动态数据采样数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<code>some_mid_001</code>	string	应符合 8.1.3 的要求	该键值对可存在
guid	<code>dev_uuid</code>	string	应符合 8.1.4 的要求	该键值对应存在
id	<code>cn1</code>	string	标识采样通道。对应数控机床文件中的采样通道 id	该键值对应存在。长度小于等于 16 字节。应与数控机床模型文件中的定义一致
data	[{ "data": [414.262, 414.262, 414.262] }, { "data": [414.262, 414.262, 414.262] }, { "data": [0, 0, 0] }, { "data": [0, 0, 0] },]	array	一个采样通道中所有的采样数据。该数组中的每一个 JSON 对象标识一个采样点。采样点使用 "data": [.....] 来表示采样数据集	该键值对应存在。采样点数据集的顺序和数目应与采样通道模型文件的采样点一致
beginTime	<code>1535642875900</code>	string	本次采样数据的开始采样时间	该键值对应存在
privateInfo	<code>null</code>	object	属于协议实现的自定义范畴	该键值对可存在

8.2.21.5 数据传输单元示例

示例：

适配器(经由代理器)向应用系统提供了采样通道 id 为“cn1”,起始采样时间为“1535642875900”的四个采样点的值。每个采样点对应一个数据项。其中每个数据项有 3 个值,3 个值分别表示该采样点在不同时间点的实际值。

```
{
  "id": "cn1",
  "beginTime": "1535642875900",
  "data": [
    {"data": [414.262, 414.262, 414.262]},
    {"data": [414.262, 414.262, 414.262]},
    {"data": [0, 0, 0]},
    {"data": [0, 0, 0]},
  ]
}
```

8.2.22 内构方法调用数据传输接口

8.2.22.1 数据传输接口指令

内构方法调用数据传输接口指令为:Method/Call/dev_uuid/ex_cid。其中“dev_uuid”是接收该消息的适配器的 GUID,“ex_cid”是发送该消息的应用系统的终端标识符。

8.2.22.2 用途

用于代理器或应用系统向适配器下达调用内构方法指令。内构方法可提供文件管理功能,可提供常见的控制功能模块(如循环启动、复位、停止等),也可由用户自定义控制功能模块。

8.2.22.3 用法

适配器监听本数据传输接口接收消息,代理器使用本数据传输接口接收并存储转发消息,应用系统使用本数据传输接口发送消息。内构方法调用数据传输接口指令交互应采用“请求/响应”模式。

当应用系统或代理器发送该数据传输接口的指令后,应用系统或代理器监听内构方法进度数据传输接口和内构方法结果数据传输接口接收到的数据。当适配器在本数据传输接口接收到内构方法调用数据传输单元以后,应解析该 DTU,并依据当前有效的模型文件检查请求的合法性和有效性,并根据检查结果决定接受或拒绝指令调用。适配器如果接受,则根据 DTU 的内构方法对象的标识号调用对应的内构方法,并且通过内构方法进度数据传输接口向应用系统给出“Response DTU”。当内构方法执行完毕后通过内构方法结果数据传输接口向应用系统给出“Response DTU”。适配器如果拒绝,则通过内构方法结果数据传输接口向应用系统给出“Response DTU”。

8.2.22.4 数据传输单元定义

内构方法调用数据传输单元定义见表 43。



表 43 内构方法调用数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
id	<i>method_01</i>	string	内构方法对象的标识号	该键值对应存在
args	{.....}	object	参数列表,用于传入调用的内构方法	该键值对可存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.22.5 数据传输单元示例

示例:

代理器或应用系统在内构方法调用数据传输接口发布如下“Request DTU”,用于注册标识号“id”为“*method_01*”的内构方法对象。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "id": "method_01",
  "args": {
    "protocol": "FTP",
    "address": "10.10.56.1:21",
    "user": "user1",
    "password": "111111",
    "source": "/gcodes/O1",
    "destination": "/h/lnc8/prog/O1",
    "operation": "push"
  }
}
```

8.2.23 内构方法进度数据传输接口

8.2.23.1 数据传输接口指令

内构方法进度数据传输接口指令为:Method/Status/*dev_uuid*/*ex_cid*。其中“*dev_uuid*”是发送该消息的适配器的 GUID,“*ex_cid*”是接收该消息的应用系统的终端标识符。

8.2.23.2 用途

用于适配器向代理器或应用系统发送内构方法执行进度。适配器通过发送内构方法执行进度,告知应用系统当前的内构方法是否仍在正常执行。

8.2.23.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统

监听本数据传输接口接收消息。内构方法进度数据传输接口指令交互应采用“订阅/发布”模式。

当适配器在本数据传输接口接收到内构方法调用数据传输单元并接受后,适配器定期通过内构方法进度数据传输接口向应用系统给出“Sample DTU”。当应用系统在发送内构方法调用 DTU 后,应一直订阅内构方法进度数据传输接口。当应用系统通过内构方法结果数据传输接口接收到内构方法结果 DTU 后,结束对内构方法进度数据传输接口的调用。

8.2.23.4 数据传输单元定义

内构方法进度数据传输单元定义见表 44。

表 44 内构方法进度数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
id	<i>method_01</i>	string	内构方法对象的标识号	该键值对应存在
status	executing	string	描述内构方法执行状态。 waiting: 正在等待执行; executing: 正在执行; stopped: 暂停; sleep: 未被调用	该键值对应存在
progress	30	number	描述内构方法执行进度百分比	该键值对可存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.23.5 数据传输单元示例

示例:

适配器(经由代理器)向应用系统提供了在内构方法进度数据传输接口发布如下“Sample DTU”,用于发布标识号“id”为“*method_01*”的内构方法对象的当前状态以及完成进度。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "id": "method_01 ",
  "status": "executing",
  "process": 30
}
```

8.2.24 内构方法结果数据传输接口

8.2.24.1 数据传输接口指令

内构方法结果数据传输接口指令为:Method/Result/*dev_uuid/ex_cid*。其中“*dev_uuid*”是发送该消息的适配器的 GUID,“*ex_cid*”是接收该消息的应用系统的终端标识符。

8.2.24.2 用途

用于适配器向代理器或应用系统发送内构方法执行结果。适配器通过发送内构方法执行结果,告

知应用系统当前的内构方法是否执行成功。

8.2.24.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。内构方法结果数据传输接口指令交互采用“请求/响应”模式。

当适配器在收到内构方法调用数据传输单元后,适配器如果接受,则当内构方法执行完毕后应通过本数据传输接口向应用系统给出“Response DTU”。适配器如果拒绝,应通过内构方法结果数据传输接口向应用系统给出“Response DTU”。当应用系统或代理器发送内构方法调用数据传输单元后,应订阅内构方法结果数据传输接口。

8.2.24.4 数据传输单元定义

内构方法结果数据传输单元定义见表 45。

表 45 内构方法结果数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
id	<i>method_01</i>	string	内构方法对象的标识号	该键值对应存在
result	finished	string	描述内构方法执行结束时的状态。 finished: 结束; canceled: 取消; error: 错误	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可存在
return	null	object	内构方法执行完毕后的返回值	该键值对可存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.24.5 数据传输单元示例

示例:

适配器在内构方法结果数据传输接口发布如下“Response DTU”,用于返回“id”为“*method_01*”的内构方法对象的执行结果以及返回值。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "id": "method_01",
  "result": "finished",
  "code": "OK",
  "return": [
    {
```



```

    "name": "O1",
    "size": "100"
  },
  {
    "name": "AC",
    "directory": "true"
  }
]
}

```

8.2.25 内构方法控制请求数据传输接口

8.2.25.1 数据传输接口指令

内构方法控制请求数据传输接口指令为：Method/Control/Request/dev_uuid/ex_cid。其中“dev_uuid”是接收该消息的适配器的 GUID，“ex_cid”是发送该消息的应用系统的终端标识符。

8.2.25.2 用途

用于代理器或应用系统向适配器发送内构方法控制请求指令。代理器或应用系统通过发送内构方法控制指令，可临时改变内构方法的执行。

8.2.25.3 用法

适配器监听本数据传输接口接收消息，代理器使用本数据传输接口接收并存储转发消息，应用系统使用本数据传输接口发送消息。内构方法控制请求数据传输接口指令交互应采用“请求/响应”模式。

当适配器在收到内构方法控制请求数据传输单元后，应解析该 DTU，并依据当前有效的模型文件检查请求的合法性和有效性，并根据检查结果决定接受或拒绝内构方法控制请求。当该应用系统无权控制该内构方法或内构方法控制请求格式出错，检查结果为拒绝。无论是接受还是拒绝，适配器应通过内构方法控制响应数据传输接口向应用系统给出“Response DTU”。

8.2.25.4 数据传输单元定义

内构方法控制请求数据传输单元定义见表 46。

表 46 内构方法控制请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	some_mid_001	string	应符合 8.1.3 的要求	该键值对应存在
guid	dev_uuid	string	应符合 8.1.4 的要求	该键值对应存在
id	method_01	string	内构方法对象的标识号	该键值对应存在
action	stop	string	描述内构方法控制指令。stop: 暂停; resume: 恢复; cancel: 取消	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.25.5 数据传输单元示例

示例：

代理器或应用系统在内构方法控制请求数据传输接口发布如下“Request DTU”，用于请求控制“id”为“method_01”的内构方法对象，“action”为“stop”，暂停内构方法的调用。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "id": "method_01",
  "action": "stop"
}
```

8.2.26 内构方法控制响应数据传输接口

8.2.26.1 数据传输接口指令

内构方法控制响应数据传输接口指令为：Method/Control/Response/dev_uuid/ex_cid。其中“dev_uuid”是发送该消息的适配器的 GUID，“ex_cid”是接收该消息的应用系统的终端标识符。

8.2.26.2 用途

当适配器在内构方法控制请求数据传输接口收到代理器或应用系统发布的内构方法控制请求后，通过本数据传输接口向代理器或应用系统返回内构方法控制请求结果。

8.2.26.3 用法

适配器监听本数据传输接口接收消息，代理器应使用本数据传输接口接收并存储转发消息，应用系统使用本数据传输接口发送消息。内构方法控制响应数据传输接口指令交互应采用“请求/响应”模式。

当适配器接收到内构方法控制请求数据传输单元，且完成或拒绝内构方法控制指令后，适配器通过内构方法控制响应数据传输接口向应用系统给出“Response DTU”。

8.2.26.4 数据传输单元定义

内构方法控制响应数据传输单元定义见表 47。

表 47 内构方法控制响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	some_mid_001	string	应符合 8.1.3 的要求	该键值对应存在
guid	dev_uuid	string	应符合 8.1.4 的要求	该键值对应存在
id	method_01	string	内构方法对象的标识号	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可存在
error	301	number	应符合 8.1.7 的要求	该键值对可存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.26.5 数据传输单元示例

示例：

适配器在内构方法控制响应数据传输接口发布如下“Response DTU”，用于返回“id”为“method_01”的内构方法对象的控制响应结果。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "id": "method_01 ",
  "code": "OK"
}
```

8.2.27 事件注册请求数据传输接口

8.2.27.1 数据传输接口指令

事件注册请求数据传输接口指令为：Register/Event/Request/dev_uuid/ex_cid。其中“dev_uuid”是接收该消息的适配器的 GUID，“ex_cid”是发送该消息的应用系统的终端标识符。

8.2.27.2 用途

用于代理器或应用系统向适配器注册事件。事件使应用系统或代理器在不更新数控机床模型文件的情况下，具有获知指定数控机床数据变化的能力。通过使用事件，适配器可在数据发生特定变化时及时通知代理器或应用系统。适配器依据事件注册请求，动态分配事件通道资源，启动事件任务。

8.2.27.3 用法

适配器监听本数据传输接口接收消息，代理器使用本数据传输接口接收并存储转发消息，应用系统使用本数据传输接口发送消息。事件注册请求数据传输接口指令交互采用“请求/响应”模式。

当适配器在该请求数据传输接口接收到事件注册请求数据传输单元后，应解析该 DTU，并依据当前有效的模型文件检查请求的合法性和有效性。并根据检查结果决定接受或拒绝事件注册请求。当该应用系统无权进行事件注册或事件注册请求格式错误，检查结果为拒绝。无论是接受还是拒绝，适配器均通过事件注册响应数据传输接口向应用系统给出“Response DTU”。DTU 中包含请求的数据项和事件过滤器的配置，适配器依据此配置，动态分配事件通道资源，启动事件任务。

8.2.27.4 数据传输单元定义

事件注册请求数据传输单元定义见表 48。过滤器参数形式化定义见图 16。

表 48 事件注册请求数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	some_mid_001	string	应符合 8.1.3 的要求	该键值对应存在
guid	dev_uuid	string	应符合 8.1.4 的要求	该键值对应存在
id	010203	string	数据项/配置项在模型文件中的 id	该键值对应存在

表 48 事件注册请求数据传输单元定义 (续)

键	值(示例值)	值类型	描述	要求
key	k1	string	字典类型数据,应在此字段中填入监听元素对应的 key。一个事件只能监听一个单值元素	该键值对可存在
index	1	number	列表类型数据,应在此字段中填入监听元素对应的 index。一个事件只能监听一个单值元素	该键值对可存在
filters	[{"type": "range", "from": 10, "to": 20}, {"type": "equal", "value": 100}]	array	数组中的每一个 JSON 对象描述一个过滤器,每个过滤器之间为“或”的关系。按照数组中过滤器顺序依次进行匹配。一个 filters 每次最多触发一个事件。 未指定任何过滤器的事件,适配器默认为事件配置一个 change 类型的过滤器。过滤器参数形式应与图 16 相符	该键值对可存在
triggerType	always	string	事件触发类型,分为 always、normal、once 三种类型: a) always 表示总是产生事件。第一次满足过滤器条件时产生第一次事件;每次值发生变化且值仍然满足过滤器条件时均会产生事件;第一次订阅事件时,若当前值已满足过滤器条件也会向订阅者发送事件; b) normal 表示由不满足过滤器条件变为满足过滤器条件时产生事件。第一次满足过滤器条件时产生第一次事件;每次值发生变化时,若前一次值不满足过滤器条件,变化后满足过滤器条件,则产生事件。若前一次值满足过滤器条件,变化后无论是否满足过滤器条件,不会产生事件;第一次订阅事件时,若当前值已满足过滤器条件也会向订阅者发送事件; c) once 表示只产生一次。仅在第一次满足过滤器条件时产生,一次事件之后事件不再产生,适配器中的事件资源被释放,事件被注销;第一次订阅事件时,若当前值已满足过滤器条件也会向订阅者发送事件	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

事件注册请求过滤器 JSON对象			
键	值（样例）	值类型	要求
"type"	"change"	string	过滤器类型 change: 变化时产生事件, 为默认值 equal: 相等时产生事件 range: 进入范围时产生事件 less: 小于时产生事件 greater: 大于时产生事件 include: 包含时产生事件
"from"	1	number	匹配范围起始值, 不包含此值, 仅出现在range类型过滤器中
"to"	123	number	匹配范围结束值, 不包含此值, 仅出现在range类型过滤器中
"value"	1	number, string, boolean	出现在equal类型过滤器中时可为数字类型, 字符串类型, 布尔类型 在less或greater类型过滤器中出现时只能为数字类型
"values"	[1,2,34,5]	array	仅出现在include类型的过滤器中。用来判断监听的数据是否包含于values数组当中。 数组中的每个元素应是数字类型、字符串类型、布尔类型中的一种
"variance"	2	number	变化幅度 适用于change/range/less/greater类型的过滤器 当指定此值时, 除了满足过滤器条件, 还要求 $ABS(\text{当前值} - \text{前一次记录值}) \geq \text{变化幅度}$ 前一次记录值指: 前一次触发事件时记录的值, 事件第一次触发时不判断变化幅度

图 16 过滤器参数形式化定义

8.2.27.5 数据传输单元示例

示例:

代理器或应用系统在事件注册请求数据传输接口发布如下“Request DTU”,注册事件,当数据项“id”为“010302”的数据只要满足在大于 10 且小于 20 或等于 100 时产生事件。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid "
  "id": "010302",
  "filters": [
    {"type": "range", "from": 10, "to": 20},
```

```

        {"type": "equal", "value": 100}
    ],
    "triggerType": "always"
}

```

8.2.28 事件注册响应数据传输接口

8.2.28.1 数据传输接口指令

事件注册响应数据传输接口指令为：`Reigster/Event/Response/dev_uuid/ex_cid`。其中“`dev_uuid`”是发送该消息的适配器的 GUID，“`ex_cid`”是接收该消息的应用系统的终端标识符。

8.2.28.2 用途

用于适配器向应用系统发送事件注册响应的 DTU。适配器通过本数据传输接口向代理器或应用系统返回事件注册结果。

8.2.28.3 用法

适配器使用本数据传输接口发送消息，代理器使用本数据传输接口接收并存储转发消息，应用系统监听本数据传输接口接收消息。事件注册响应数据传输接口指令交互应采用“请求/响应”模式。

当适配器接收到事件注册请求数据传输单元以后，适配器通过事件注册响应数据传输接口向应用系统给出“Response DTU”。适配器若在持续时间内未能收到终端标识符为“`ex_cid`”的应用系统的 DTU，则会主动注销该事件通道。

8.2.28.4 数据传输单元定义

事件注册响应数据传输单元定义见表 49。

表 49 事件注册响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
id	<i>event_01</i>	string	事件 id, 由适配器生成, 后续事件以此 id 作为标识	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可选存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.28.5 数据传输单元示例

示例：

适配器在事件注册响应数据传输接口发布如下“Response DTU”，用于返回事件注册响应，包含事件“id”为

“event_01”，注册结果“code”为“OK”。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "id": "event_01",
  "code": "OK"
}
```

8.2.29 事件注销请求数据传输接口

8.2.29.1 数据传输接口指令

事件注销请求数据传输接口指令为：Unregister/Event/Request/dev_uuid/ex_cid。其中“dev_uuid”是接收该消息的适配器的 GUID，“ex_cid”是发送该消息的应用系统的终端标识符。

8.2.29.2 用途

用于代理器或应用系统向适配器注销事件。注销后，事件任务终止，被注销的事件资源被回收。

8.2.29.3 用法

适配器监听本数据传输接口接收消息，代理器使用本数据传输接口接收并存储转发消息，应用系统使用本数据传输接口发送消息。事件注销请求数据传输接口指令交互应采用“请求/响应”模式。

当适配器在该请求数据传输接口接收到事件注销请求数据传输单元以后，应解析该 DTU，并依据当前有效的模型文件检查请求的合法性和有效性。并根据检查结果决定接受或拒绝数据设置请求。无论是接受还是拒绝，适配器均通过事件注销响应数据传输接口向应用系统给出“Response DTU”。

8.2.29.4 数据传输单元定义

事件注销请求数据传输单元定义见表 50。

表 50 事件注销数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	some_mid_001	string	应符合 8.1.3 的要求	该键值对应存在
guid	dev_uuid	string	应符合 8.1.4 的要求	该键值对应存在
id	event_01	string	事件 id	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.29.5 数据传输单元示例

示例：

代理器或应用系统在事件注销请求数据传输接口发布如下“Request DTU”，注销事件“id”为“event_01”。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid ",
  "id": "event_01"
}
```

8.2.30 事件注销响应数据传输接口

8.2.30.1 数据传输接口指令

事件注销响应数据传输接口指令为: `Unregister/Event/Response/dev_uuid/ex_cid`。其中“`dev_uuid`”是发送该消息的适配器的 GUID,“`ex_cid`”是接收该消息的应用系统的终端标识符。

8.2.30.2 用途

用于适配器向应用系统发送事件注销响应的 DTU。适配器通过本数据传输接口向代理器或应用系统返回事件注销结果。

8.2.30.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。事件注销响应数据传输接口指令交互采用“请求/响应”模式。

当适配器接收到事件注销请求数据传输单元以后,通过事件注销响应数据传输接口向应用系统给出“Response DTU”。应用系统可忽略此消息。

8.2.30.4 数据传输单元定义

事件注销响应数据传输单元定义见表 51。

表 51 事件注销响应数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_001</i>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<i>dev_uuid</i>	string	应符合 8.1.4 的要求	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	Permission Denied	string	应符合 8.1.7 的要求	该键值对可选存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在

8.2.30.5 数据传输单元示例

示例:

适配器在事件注销响应数据传输接口发布如下“Response DTU”。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "code": "NG",
  "reason": "Permission Denied",
  "error": 301
}
```



8.2.31 事件数据数据传输接口

8.2.31.1 数据传输接口指令

事件数据数据传输接口指令为: `Event/dev_uuid/ex_cid`。其中“`dev_uuid`”是发送该消息的适配器的 GUID,“`ex_cid`”是接收该消息的应用系统的终端标识符。

8.2.31.2 用途

用于适配器向代理器或应用系统发送事件数据。发送时表明事件被触发,代理器或应用系统可及时获得数据改变的消息。

8.2.31.3 用法

适配器使用本数据传输接口发送消息,代理器使用本数据传输接口接收并存储转发消息,应用系统监听本数据传输接口接收消息。事件数据数据传输接口指令交互采用“订阅/发布”模式。当适配器在接受事件注册请求后,根据事件注册请求中的配置分配资源,当监听的数据项发生的变化满足过滤器要求后,通过事件数据数据传输接口向终端标识符为“`ex_cid`”的应用系统给出“Sample DTU”。

8.2.31.4 数据传输单元定义

事件数据数据传输单元定义见表 52。事件体数据参数形式化定义见图 17。

表 52 事件数据数据传输单元定义

键	值(示例值)	值类型	描述	要求
@id	<code>some_mid_001</code>	string	应符合 8.1.3 的要求	该键值对应存在
guid	<code>dev_uuid</code>	string	应符合 8.1.4 的要求	该键值对应存在
id	<code>event_01</code>	string	事件 id	该键值对应存在
time	1120302139	number	事件产生的毫秒级别时间戳	该键值对应存在
event	{ "id": "1023213", "value": 5 }	object	事件体包含描述触发事件的信息。事件体数据参数形式应与图 17 相符	该键值对应存在
privateInfo	null	object	属于协议实现的自定义范畴	该键值对可存在



图 17 事件体数据参数形式化定义

8.2.31.5 数据传输单元示例

示例：

适配器(经由代理器)向应用系统提供了在事件数据数据传输接口发布如下“Sample DTU”,用于发布事件“id”为“event_01”被触发时的值。

```
{
  "@id": "some_mid_001",
  "guid": "dev_uuid",
  "id": "event_01 ",
  "time": 19202912 ,//事件产生时间戳
  "event": {
    "id": "012023",
    "index": "1"
    "oldValue": 10,
    "value": 5
  }
}
```

8.3 网络连接错误场景

8.3.1 概述

应用系统通过代理器与适配器交互时,在任何时候均存在以下两个连接:

- 应用系统与代理器之间的连接;
- 适配器与代理器之间的连接。

8.3.2 连接错误情景一

该错误情景见图 18,Time-1 时刻,应用系统 A 到代理器 A 之间网络连接错误,适配器 A 到代理器 A 之间连接正常。此时没有正在发送、处理、等待接收的 DTU。

该情景下:

- 应用系统 A 无法将“Request DTU”送达至代理器 A,也无法接收由代理器 A 发送的“Response DTU”;
- 代理器 A 无法将“Response DTU”送达至应用系统 A,也无法接收由应用系统 A 发送的“Request DTU”。但代理器 A 可接收与其通信连接正常终端发送的“Resquest/Response DTU”,代理器 A 也可将“Resquest/Response DTU”送达至与其通信连接正常的终端;
- 适配器 A 可将“Response DTU”送达至代理器 A,也可接收由代理器 A 发送的“Request DTU”。

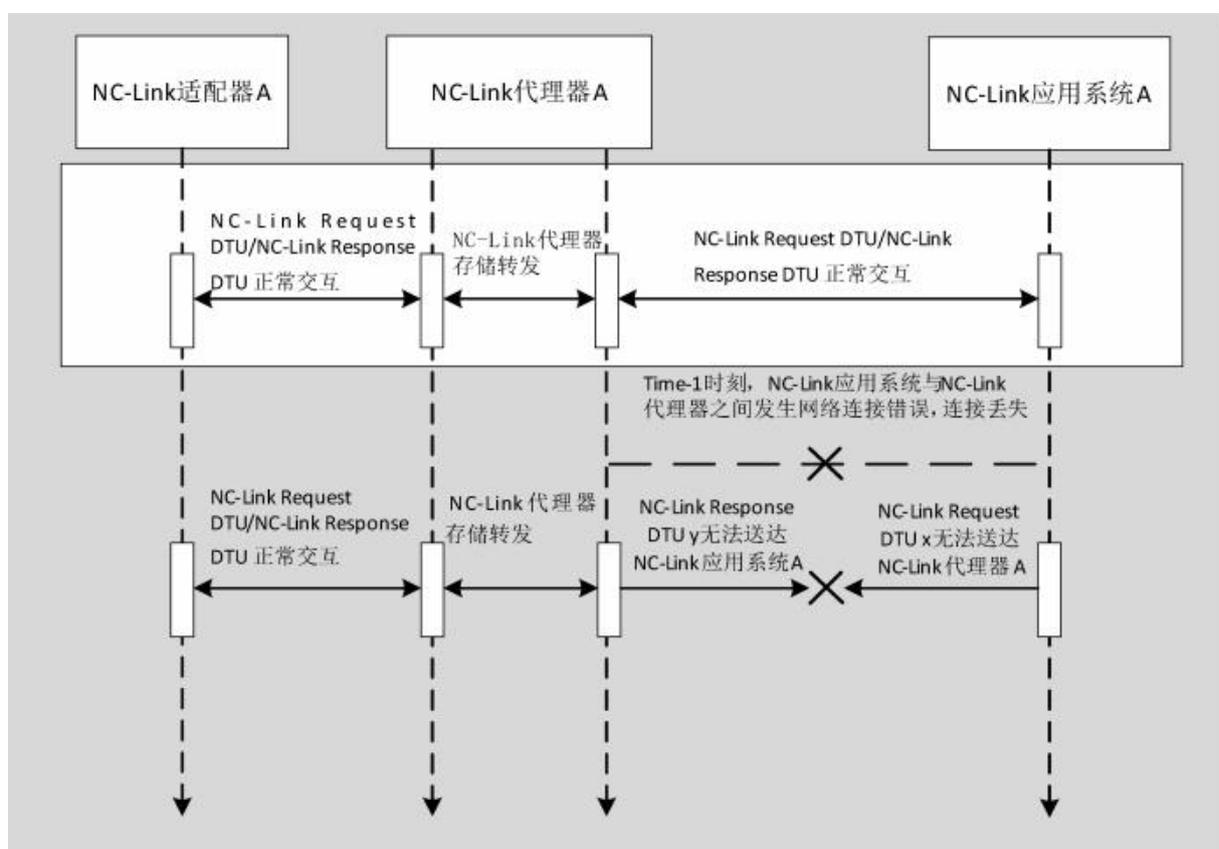


图 18 连接错误情景一

应用系统可选择缓存无法发送的“Request DTU”当前请求,尝试重新建立连接并发送该请求数据单元。也可选择丢弃该“Request DTU”。代理器可缓存由适配器发送给应用系统的上一条“Request DTU”所对应的“Response DTU”,当应用系统恢复连接时,代理器再转发给应用系统这条“Response DTU”,也可丢弃该“Response DTU”。

8.3.3 连接错误情景二

该错误情景见图 19, Time-1 时刻,在应用系统 A 向代理器 A 送达“Request DTU x”后,且代理器 A 向应用系统 A 发送“Request DTU x”所对应的“Response DTU x”前,应用系统 A 到代理器 A 之间网络连接错误,适配器 A 到代理器 A 之间连接正常。此时没有正在发送、处理、等待接收的 DTU。

该情景下:

- a) 应用系统 A 无法给代理器 A 送达新的“Request DTU y”,也无法接收代理器 A 发送的“Response DTU x”;
- b) 代理器 A 可接收与代理器 A 通信连接正常的终端发送的新的“Request DTU”,也可送达新的“Response DTU”到与代理器 A 通信连接正常的终端;
- c) 适配器 A 可接收代理器 A 发送的新的“Request DTU”,也可送达新的“Response DTU”到代理器 A。

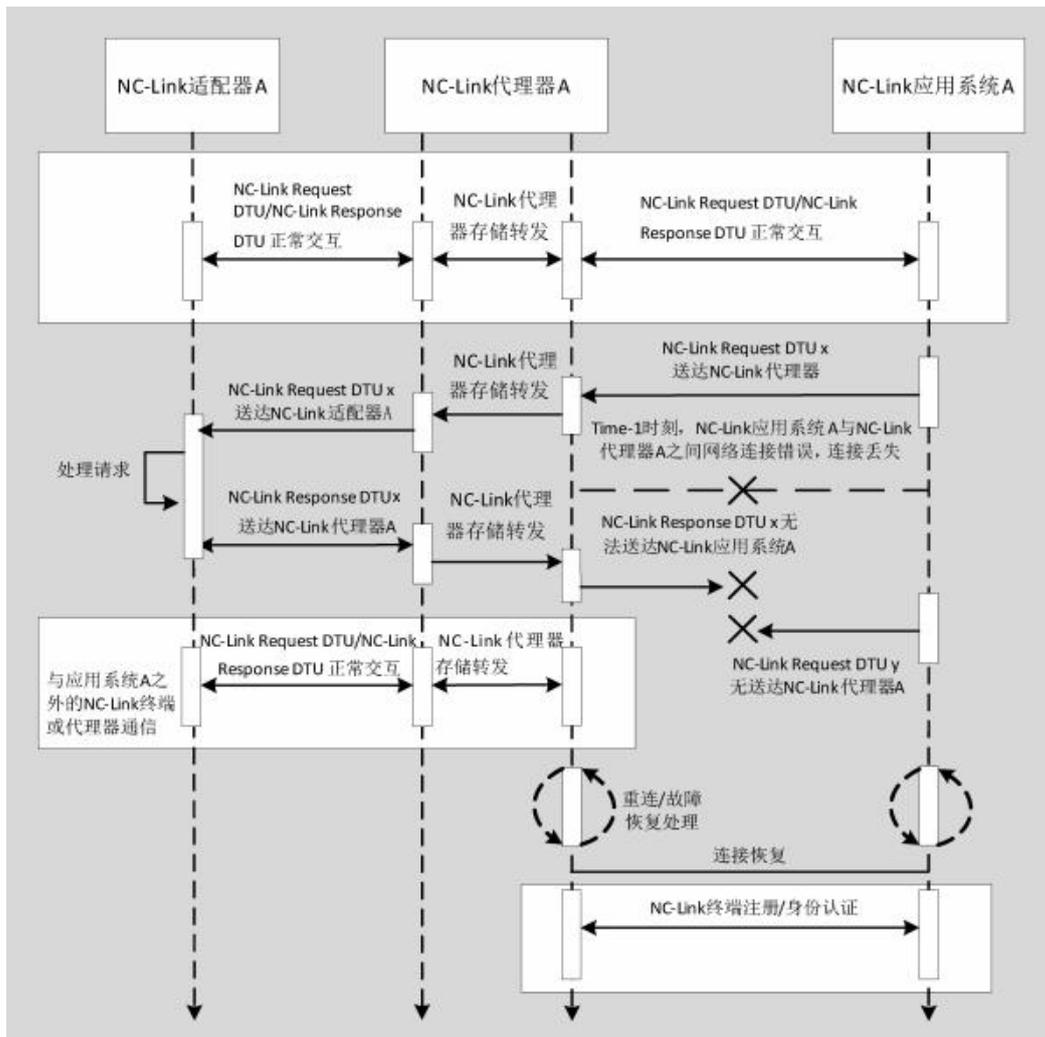


图 19 连接错误情景二

应用系统可选择缓存无法发送的“Request DTU”当前请求,尝试重新建立连接并发送该“Request DTU”。也可选择丢弃该“Request DTU”。代理器可缓存由适配器发送给应用系统的上一条“Request DTU”所对应的“Response DTU”,当应用系统恢复连接时,代理器再转发给应用系统这条“Response DTU”。也可丢弃该“Response DTU”。

8.3.4 连接错误情景三

该错误情景见图 20,Time-1 时刻,在代理器 A 向适配器 A 送达“Request DTU x”前,应用系统 A 到代理器 A 之间网络连接正常,适配器 A 到代理器 A 之间网络连接错误。此时没有正在发送、处理、等待接收的 DTU。

该情景下:

- a) 应用系统 A 可给代理器 A 送达的“Request DTU”,也可接收代理器 A 发送的“Response DTU”;
- b) 代理器 A 无法将“Request DTU x”送达到适配器 A,也无法接收由适配器 A 发送的“Response DTU y”,但代理器 A 可接收与其通信连接正常的终端发送的“Request/Response DTU”,代理器 A 也可送达“Request/Response DTU”到与其通信连接正常的终端;
- c) 适配器 A 也无法送达“Response DTU”到代理器 A,适配器也无法接收到代理器 A 发送的“Request DTU”。



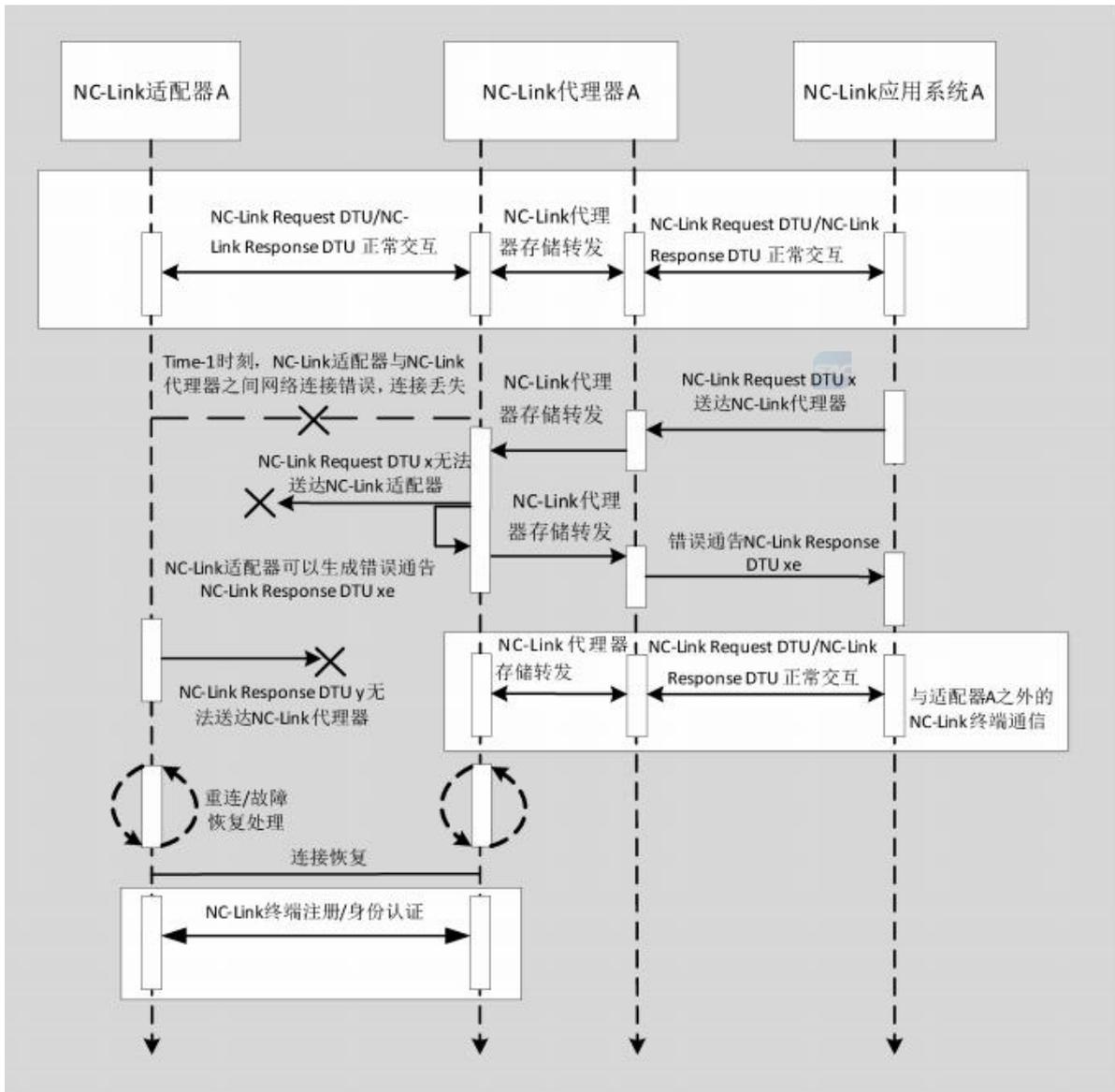


图 20 连接错误情景三

代理器可缓存应用系统发出的“Request DTU”，在连接重新建立起后再次发送，也可丢弃该“Request DTU”，但应告知应用系统，当前与适配器连接错误状态。代理器向应用系统报告错误状态的数据传输单元见表 53。

表 53 连接错误情景三错误状态的数据传输单元

键	值(示例值)	值类型	描述	要求
@id	some_mid_123	string	值“some_mid_123”对应于应用系统发送的最近一个 RequestDTU 中的值	该键值对应存在
dev_uuid	dev_uuid_2	string	应符合 8.1.4 的要求	该键值对应存在

表 53 连接错误情景三错误状态的数据传输单元（续）

键	值(示例值)	值类型	描述	要求
cli_uuid	<i>ex_cid</i>	string	应符合 8.1.4 的要求	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	DEV_UNREACHABLE	string	应符合 8.1.7 的要求	该键值对应存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
privateInfo	null	JSON 对象	属于协议实现的自定义范畴	该键值对可存在

8.3.5 连接错误情景四

该错误情景见图 21, Time-1 时刻, 适配器接收到“Request DTU x”, 在处理请求时, 应用系统到代理器之间网络连接正常, 适配器到代理器之间网络连接错误。此时没有正在发送、处理、等待接收的 DTU。

该情景下:

- a) 应用系统 A 可给与其通信连接正常的代理器送达的“Request DTU”, 也可接收通信连接正常的代理器发送的“Response DTU”;
- b) 代理器 A 无法将“Request DTU x”送达到适配器 A, 也无法接收由适配器 A 发送的“Response DTU y”, 但代理器 A 可接收与其通信连接正常的终端发送的“Request/Response DTU”, 代理器 A 也可送达“Request/Response DTU”到与其通信连接正常的终端;
- c) 适配器 A 也无法送达“Response DTU”到代理器 A, 适配器也无法接收到代理器 A 发送的“Request DTU”。



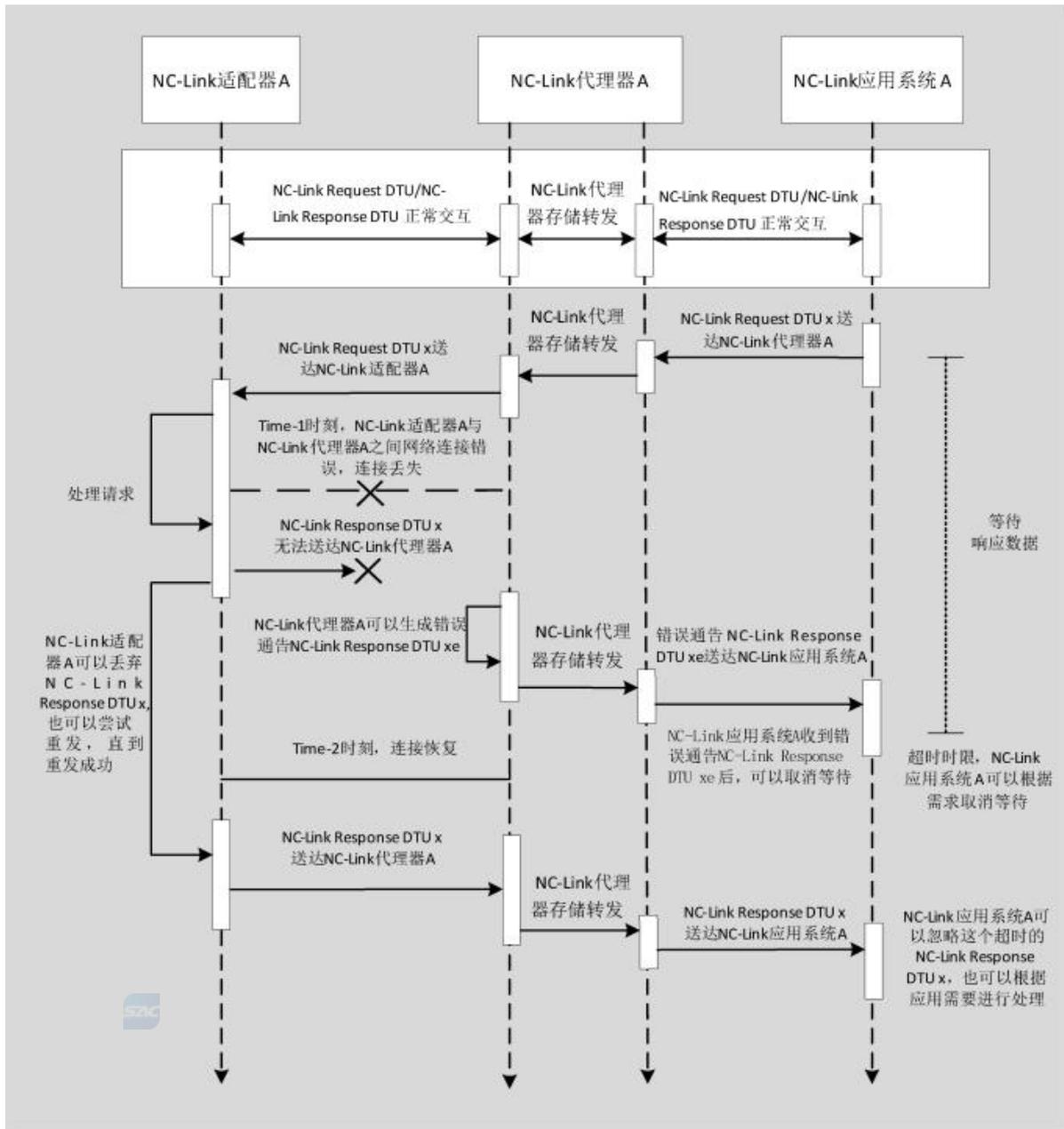


图 21 连接错误情景四

代理器可缓存应用系统发出的“Request DTU”，也可丢弃该 DTU，但应告知应用系统，当前与适配器网络连接错误。代理器向应用系统报告错误状态的数据传输单元“Response DTU xe”，见表 54。应用系统收到错误通告“Response DTU xe”后，可取消等待。等待时间超过时限，应用系统可根据需求取消等待。适配器可丢弃“Response DTU x”，也可尝试重发，直到重发成功。应用系统可忽略超时的“Response DTU x”，也可根据应用需要进行处理。

表 54 错误情景四错误状态的数据传输单元

键	值(示例值)	值类型	描述	要求
@id	<i>some_mid_123</i>	string	值“ <i>some_mid_123</i> ”对应于应用系统发送的最近一个 RequestDTU 中的值	该键值对应存在
dev_uuid	<i>dev_uuid_2</i>	string	应符合 8.1.4 的要求	该键值对应存在
cli_uuid	<i>ex_cid</i>	string	应符合 8.1.4 的要求	该键值对应存在
code	NG	string	应符合 8.1.7 的要求	该键值对应存在
reason	DEV _ UNREACH- ABLE	string	应符合 8.1.7 的要求	该键值对应存在
error	301	number	应符合 8.1.7 的要求	该键值对可选存在
privateInfo	null	JSON 对象	属于协议实现的自定义范畴	该键值对可存在

8.3.6 连接错误情景五

该错误情景见图 22, Time-1 时刻, 应用系统 A 到代理器 A、适配器 A 到代理器 A 之间的网络连接均错误。这包括代理器 A 无法工作或存在其他形式失能。此时没有正在发送、处理、等待接收的数据单元。

该情景下:

- a) 应用系统 A 无法送达“Request DTU”到代理器 A, 也无法接收到由代理器 A 发送的“Response DTU”;
- b) 代理器 A 无法将“Request DTU”送达到适配器 A, 也无法接收由适配器 A 发送的“Response DTU”, 但代理器 A 可接收与其通信连接正常的终端发送的“Request/Response DTU”, 代理器 A 也可送达“Request/Response DTU”到与其通信连接正常的终端;
- c) 适配器 A 也无法送达“Response DTU”到代理器 A, 适配器 A 也无法接收到代理器 A 发送的“Request DTU”。

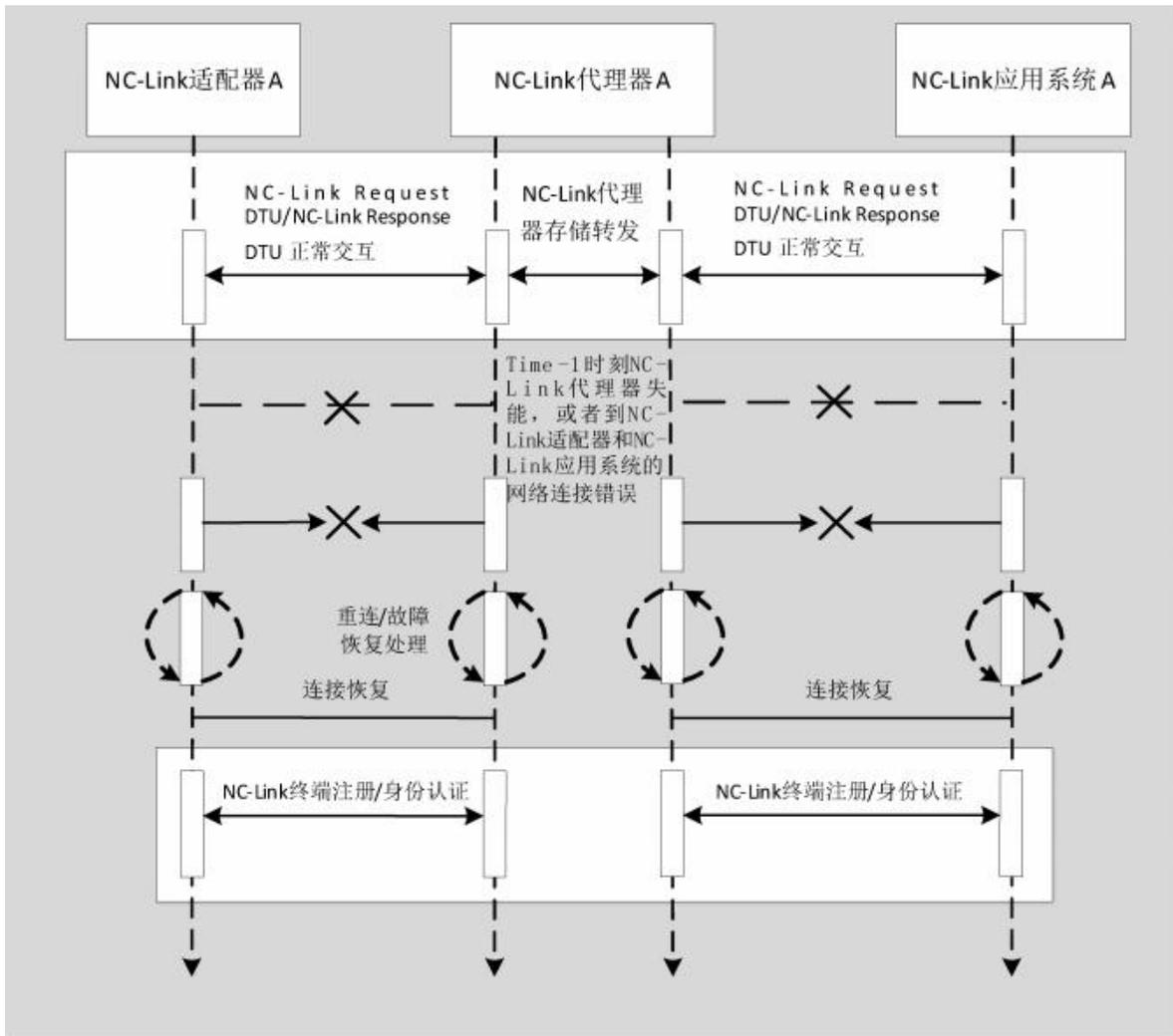


图 22 连接错误情景五

适配器和应用系统可选择不断重试进行连接,也可选择放弃连接,建议实施者考虑和处理网络的抖动和临时性失效的场景。

当应用系统 A 无法发送“Request DTU”时,应用系统 A 实施者可根据应用需求,缓存当前请求,尝试重新建立连接并发送该“Request DTU”。也可选择丢弃当前“Request DTU”,尝试重新建立连接并发送应用业务层决定的新的请求。

当适配器 A 无法发送“Response DTU”时,适配器 A 可缓存该“Response DTU”,尝试重新建立连接并发送该“Response DTU”。也可选择丢弃该“Response DTU”。

当代理器 A 恢复正常工作或应用系统 A 到代理器 A、适配器 A 到代理器 A 之间的网络连接先后恢复正常时,适配器 A 应重新进行终端注册,应用系统 A 可重新进行终端探测,也可不进行终端探测。



9 测试与评价

9.1 测试要求

9.1.1 测试目的

对按照 NC-Link 标准开发的适配器和代理器进行符合性测试。

9.1.2 测试系统组成

9.1.2.1 概述

测试系统由应用测试系统、标准代理器、标准适配器、待测代理器、待测适配器及与标准适配器和待测适配器连接的数控机床组成,具体见图 23。其中,适配器可集成在数控机床中如连接方式 2~连接方式 4 中的适配器。

根据实现形式的不同,测试系统中各组成部分的连接方式共有四种,分别对应四种不同的测试对象(图 23 中虚线框内)。具体见图 23。

测试时,应用测试系统向所连接的标准代理器或待测代理器发送各项测试指令,接收从标准代理器或待测代理器返回的响应数据,形成各数据项序列,以此对测试对象进行测试,给出通过、不通过以及通过率等结果。

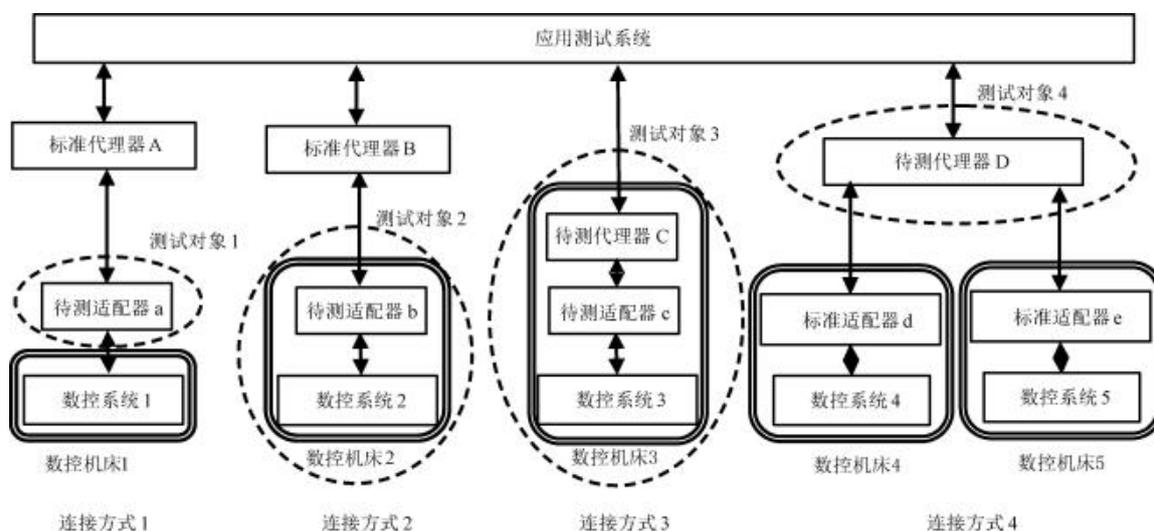


图 23 测试系统和测试对象连接方式

9.1.2.2 应用测试系统

应用测试系统是专门测试代理器和适配器而配置在应用系统端的软件系统,能够向代理器发送测试指令、接收响应数据,并与测试用例进行对比,验证测试项目的正确性,还能对获取的数控机床模型文件的语法和数据项内容的规范性进行评价。

9.1.2.3 标准代理器

标准代理器是具备表 17 中列出的代理器应支持和可支持的各项通信接口功能特性的代理器,能够

与符合本文件要求的适配器、代理器和应用系统相连并正常工作。

9.1.2.4 标准适配器

标准适配器是具备表 17 中列出的适配器应支持的各项通信接口功能各必选项功能特性的适配器,并具备与所连接的数控机床相适应的表 17 中列出的适配器可支持的各项通信接口功能所有可选项功能特性。

9.1.3 测试对象的连接

测试对象共有四种连接方式(见图 23),各连接方式的适用场景分别如下:

- a) 连接方式 1:适用于对独立的待测适配器进行测试,如图 23 中的待测适配器 a。该待测适配器与对应的数控机床连接,并通过标准代理器与应用测试系统连接;
- b) 连接方式 2:适用于待测适配器与数控机床为集成关系的场景,如图 23 中的适配器 b。该数控机床通过标准代理器与应用测试系统连接;
- c) 连接方式 3:适用于待测适配器和待测代理器与数控机床为集成关系的场景,如图 23 中的适配器 c 和代理器 C。该测试对象应与应用测试系统直接连接;
- d) 连接方式 4:适用于对独立的待测代理器进行测试,如图 23 中的代理器 D。该代理器分别与标准适配器(连接相应的数控机床)和应用测试系统连接。该连接方式下,数控机床与标准适配器之间应能互相匹配支持。

9.2 测试内容

9.2.1 连接测试

按 9.1.2.1 组建测试系统,统一上电。

应用测试系统按表 55 数据传输接口功能的次序,执行“终端注册数据传输接口”与“终端探测数据传输接口”的测试验证。如测试对象的连接测试不通过,则不应再进行后续的测试内容。

表 55 连接测试项目

序号	数据传输接口功能	数据传输接口描述	测试验证内容
1	终端注册 Register/Request	应用测试系统向代理器注册	应用测试系统分别向标准代理器 A、标准代理器 B、待测代理器 C、待测代理器 D(四种情况)注册,检查是否注册成功
2	终端探测 Discovery/Request	应用测试系统向代理器请求已经注册的数控机床列表	应用测试系统分别向标准代理器 A、标准代理器 B、待测代理器 C、待测代理器 D 请求终端设备列表,检查是否能分别发现待测适配器 a、待测适配器 b、待测适配器 c、标准适配器 d 和标准适配器 e

9.2.2 测试用例

测试用例应满足数据传输接口功能和数据传输接口参数(的)组合的测试需求。

9.2.3 模型侦测与模型设置测试

模型侦测请求/响应数据传输接口与模型设置请求/响应数据传输接口的测试内容见表 56。

表 56 模型侦测与模型设置测试项目

序号	数据传输接口功能	数据传输接口描述	测试验证内容
1	模型侦测 Probe/Query/Request	应用测试系统向适配器请求数控机床的模型文件	测试验证内容包括以下两部分： a) 应用测试系统分别通过标准代理器 A、标准代理器 B、待测代理器 C、待测代理器 D 向待测适配器 a、待测适配器 b、待测适配器 c、标准适配器 d 和标准适配器 e 获取数控机床 1~数控机床 5 的模型文件，检查是否能成功获取； b) 应用测试系统对机床模型文件的语法和数据项定义使用的规范性进行验证，检查是否符合要求
2	模型设置 Probe/Set/Request	应用测试系统向适配器更新数控机床的模型文件	测试验证内容包括以下两部分： a) 应用测试系统分别更新待测适配器 a、待测适配器 b、待测适配器 c、标准适配器 d 和标准适配器 e 中的机床模型文件版本，检查是否成功； b) 检查应用测试系统是否收到版本号校对请求，且版本号已经更新

对获取的模型文件内容的语法进行规范性验证，具体应符合第 6 章的要求。如数控机床模型文件中出现语法问题，应更正后重新测试；如数控机床模型文件中定义的数据项不符合要求，或不符合所连接装备的实际情况，也应更正后重新测试。

9.2.4 数据查询与数据设置测试

数据查询请求/响应数据传输接口的测试内容见表 57。setable 属性为真的数据对象的取值进行更新的数据设置请求/响应数据传输接口测试的内容可参照执行。

表 57 数据查询与数据设置测试项目

序号	数据传输接口功能	数据传输接口描述	测试验证内容
1	数据查询 Query/Request	应用测试系统查询数控机床模型文件中定义的数据项的当前值	对获取的数据项的值进行测试验证，检查是否正确
2	数据设置 Set/Request	应用测试系统对数控机床模型文件中定义的数据项设置新的值	对设置的数据项新的值进行测试验证，检查是否有效

9.2.5 数据采样测试

数据采样数据传输接口的测试内容见表 58。

表 58 数据采样测试项目

序号	数据传输接口功能	数据传输接口描述	测试验证内容
1	数据采样 Sample	应用测试系统对数控机床模型文件中定义的采样通道中的数据项进行采样	对获取的系列采样数据进行测试验证，检查是否正常

9.2.6 事件功能测试

事件注册请求/响应数据传输接口、事件数据数据传输接口和事件注销请求/响应数据传输接口的测试内容见表 59。

表 59 事件功能测试项目

序号	数据传输接口功能	数据传输接口描述	测试验证内容
1	事件注册/注销 Register/Event/ RequestUnregister/Sample/Request	应用测试系统定义事件,进行注册/注销	检查事件的注册/注销是否成功
2	事件数据 Event	应用测试系统查询事件数据	查询事件数据序列,检查是否正确

9.2.7 动态采样功能测试

动态采样注册请求/响应数据传输接口、数据采样数据传输接口和动态采样注销请求/响应数据传输接口的测试内容见表 60。

表 60 动态采样功能测试项目

序号	数据传输接口功能	数据传输接口描述	测试验证内容
1	动态采样注册/注销 Register/Sample/ RequestUnregister/Sample/Request	应用测试系统定义动态采样,进行注册/注销	检查动态采样注册/注销是否成功
2	数据采样 Sample	应用测试系统对动态采样注册成功的数据项进行数据采样	查询注册的动态采样数据,获取采样数据序列,检查是否正确

9.2.8 状态通知测试

状态通知数据传输接口的测试内容见表 61。

表 61 状态通知测试项目

序号	数据传输接口功能	数据传输接口描述	测试验证内容
1	状态通知 Notify/State	应用测试系统接收适配器离线的状态通知	检查在适配器离线后,能否接收到适配器离线的状态通知

9.2.9 内构方法功能测试

内构方法调用数据传输接口、内构方法进度数据传输接口、内构方法结果数据传输接口和内构方法控制请求/响应数据传输接口的测试内容见表 62。



表 62 内构方法功能测试项目

序号	数据传输接口功能	数据传输接口描述	测试验证内容
1	内构方法调用/控制请求 Method/CallMethod/Control /Request	应用测试系统调用已经定义的内构方法功能,并发送内构方法控制请求	检查能否成功调用内构方法,能否获取状态信息,能否发送新的控制数据,能否获取最终的执行结果

9.3 测试结果评价

9.3.1 连接测试

连接测试见图 23,应用测试系统分别依次向待测适配器 a、待测适配器 b、待测适配器 c、标准适配器 d 和标准适配器 e 发送终端注册、终端探测等数据传输接口指令,若应用测试系统均能得到正确有效的响应,则判定对应的测试对象中的待测代理器和待测适配器通过该项测试,否则判定相应的待测代理器和待测适配器测试不通过该项测试。

9.3.2 模型文件规范性检查

按照第 6 章模型定义的要求,对模型文件进行语法的规范化检查,包括模型文件所支持的标准版本号、对象定义、规范引用的数据项定义等。如模型文件符合语法规则,则判定该模型文件通过检查,否则判定该模型文件不通过检查。

9.3.3 适配器和代理器功能测试

按照图 23 连接的待测适配器和待测代理器,其功能测试内容及结果评价见表 63。

表 63 适配器和代理器功能测试评价项目

序号	测试内容	评价
1	能否对数控机床模型文件中定义的数据对象进行正常的数据项读/写操作	若能正常读/写操作,则测试通过,否则为不通过
2	能否对数控机床模型文件中定义的采样通道对象进行数据采样	若能正常采样,则测试通过,否则为不通过
3	能否对数控机床模型文件中定义的数据对象进行事件注册、事件数据查询和注销	若能正常使用事件功能,则测试通过,否则为不通过
4	能否对数控机床模型文件中定义的数据对象进行动态采样注册、数据采样和注销	若能正常使用动态采样功能,则测试通过,否则为不通过
5	能否对数控机床模型文件中定义的内构方法调用、控制请求,以及能否获取内构方法调用的进度和结果信息	若能正常使用内构方法功能,则测试通过,否则为不通过
6	能否实现状态通知	若能正常接收到适配器离线的状态通知,则测试通过,否则为不通过

其中,连接方式 1、连接方式 2 和连接方式 3 中的待测适配器,其数据传输接口功能与连接的数控机床有关,如数控机床模型文件中没有定义采样通道对象或内构方法,则表 63 中的第 2 项和第 5 项内容可不进行测试;如数控机床可能不需要或不支持事件和动态采样的特性,则表 63 中的第 3 项和第 4 项内容为可选测试。

连接方式 4 中的待测代理器应进行表 63 中的各项测试内容。

9.4 测试记录

测试完成后应将测试内容及测试结果等信息汇总记录。附录 C 给出了测试记录的样表。



附 录 A
(资料性)
数控机床模型样式文件示例

下面给出了数控机床模型样式文件的示例。

示例：

```
{
  "$schema": "http://JSON-schema.org/draft-07/schema#",
  "$id": "http://www.nclink.com.cn/draft-01/schema#",
  "description": "智能工厂数控机床互联接口规范",
  "type": [
    "object",
    "boolean"
  ],
  "anyOf": [
    {
      "$ref": "#/definitions/NC_Probe"
    },
    {
      "$ref": "#/definitions/CMD_Register_Request"
    },
    {
      "$ref": "#/definitions/CMD_Probe_Version"
    },
    {
      "$ref": "#/definitions/CMD_Discovery_Response"
    },
    {
      "$ref": "#/definitions/CMD_Probe_Query_Response"
    },
    {
      "$ref": "#/definitions/CMD_Probe_Set_Request"
    },
    {
      "$ref": "#/definitions/CMD_Probe_Set_Response"
    },
    {
      "$ref": "#/definitions/CMD_Query_Request"
    },
  ],
}
```

```

    {
      "$ref": "#/definitions/CMD_Query_Response"
    },
    {
      "$ref": "#/definitions/CMD_Probe_Set_Request"
    },
    {
      "$ref": "#/definitions/CMD_Probe_Set_Response"
    },
    {
      "$ref": "#/definitions/CMD_Sample"
    }
  ],
  "definitions": {
    "NC_Probe": {
      "description": "数控机床模型描述",
      "type": "object",
      "required": [
        "id",
        "type",
        "version",
        "devices"
      ],
      "properties": {
        "name": {
          "type": "string"
        },
        "id": {
          "type": "string"
        },
        "type": {
          "enum": [
            "NC_LINK_ROOT"
          ]
        },
        "version": {
          "type": "string",
          "pattern": "-[0-9]{1}.[0-9]{1}.[0-9]{1}$",
          "description": "NC-Link 协议版本号,例如: 1.2.3"
        }
      }
    }
  }

```

```

    "description": {
      "type": "string"
    },
    "configs": {
      "$ ref": "#/definitions/ConfigsWithMethod"
    },
    "devices": {
      "type": "array",
      "description": "多个数控设备",
      "minItems": 0,
      "items": {
        "$ ref": "#/definitions/Device"
      }
    }
  },
  "Device": {
    "type": "object",
    "description": "一台数控设备",
    "required": [
      "id",
      "type",
      "version"
    ],
    "properties": {
      "name": {
        "type": "string"
      },
      "id": {
        "type": "string"
      },
      "type": {
        "enum": [
          "AVG",
          "CLEAN",
          "MACHINE",
          "MEASURE",
          "PRODUCTLINE",
          "WAREHOUSE"
        ]
      }
    }
  }
}

```

```

    },
    "description": {
      "type": "string"
    },
    "number": {
      "type": "string"
    },
    "guid": {
      "type": "string"
    },
    "version": {
      "type": "string",
      "pattern": "^-[0-9]{1}.[0-9]{1}.[0-9]{1}$",
      "description": "模型文件版本号,例如: 1.2.3"
    },
    "configs": {
      "$ ref": "#/definitions/ConfigsWithChannel"
    },
    "dataItems": {
      "$ ref": "#/definitions/DataItems"
    },
    "components": {
      "$ ref": "#/definitions/Components"
    }
  }
},
"ConfigsWithMethod": {
  "type": "array",
  "description": "设备层配置信息,可包含采样通道",
  "minItems": 0,
  "items": {
    "anyOf": [
      {
        "$ ref": "#/definitions/DataItem"
      },
      {
        "$ ref": "#/definitions/Method"
      }
    ]
  }
}

```

```

},
"ConfigsWithChannel": {
  "type": "array",
  "description": "设备层配置信息,可包含采样通道",
  "minItems": 0,
  "items": {
    "anyOf": [
      {
        "$ref": "#/definitions/DataItem"
      },
      {
        "$ref": "#/definitions/SampleChannel"
      }
    ]
  }
},
"ConfigsGen": {
  "$ref": "#/definitions/DataItems"
},
"DataItems": {
  "type": "array",
  "description": "数据项数组",
  "minItems": 0,
  "items": {
    "$ref": "#/definitions/DataItem"
  }
},
"postiveInteger": {
  "type": "integer",
  "minimum": 1
},
"DataItem": {
  "type": "object",
  "description": "一个数据项",
  "required": [
    "id",
    "type"
  ],
  "properties": {
    "name": {

```

```

    "type": "string"
  },
  "id": {
    "type": "string"
  },
  "type": {
    "$ref": "#/definitions/ValueTypeNames"
  },
  "units": {
    "$ref": "#/definitions/unitType"
  },
  "value": {
    "$ref": "#/definitions/ValueTypes"
  },
  "source": {
    "type": "string"
  },
  "description": {
    "type": "string"
  },
  "datatype": {
    "enum": [
      "list",
      "dict"
    ]
  },
  "setable": {
    "type": "boolean",
    "default": false
  }
}
},
"Method": {
  "type": "object",
  "description": "内构方法对象",
  "required": [
    "id",
    "type"
  ],
  "properties": {

```

```

    "name": {
      "type": "string"
    },
    "id": {
      "type": "string"
    },
    "type": {
      "enum": [
        "SAMPLE_CHANNEL"
      ]
    },
    "description": {
      "type": "string"
    },
    "args": {
      "$ ref": "#/definitions/DICT"
    }
  }
},
"SampleChannel": {
  "type": "object",
  "description": "采样通道对象",
  "required": [
    "id",
    "type",
    "sampleInterval",
    "uploadInterval",
    "ids"
  ],
  "properties": {
    "name": {
      "type": "string"
    },
    "id": {
      "type": "string"
    },
    "type": {
      "enum": [
        "SAMPLE_CHANNEL"
      ]
    }
  }
}

```



```

    },
    "description": {
      "type": "string"
    },
    "sampleInterval": {
      "$ref": "#/definitions/postiveInteger"
    },
    "uploadInterval": {
      "$ref": "#/definitions/postiveInteger"
    },
    "ids": {
      "type": "array",
      "minItems": 0,
      "items": {
        "type": "object",
        "required": [
          "id"
        ],
        "properties": {
          "id": {
            "type": "string"
          }
        }
      }
    }
  },
  "Components": {
    "type": "array",
    "description": "多个组件组成的组件组",
    "minItems": 0,
    "items": {
      "$ref": "#/definitions/Component"
    }
  },
  "Component": {
    "type": "object",
    "description": "单个组件及其数据项",
    "required": [
      "id",

```



```

    "type"
  ],
  "properties": {
    "name": {
      "type": "string"
    },
    "id": {
      "type": "string"
    },
    "guid": {
      "$ ref": "#/definitions/DeviceId"
    },
    "type": {
      "enum": [
        "AUXILIARY",
        "AXIS",
        "COMPONENT",
        "CONTROLLER",
        "FIXTURE",
        "MOTOR",
        "SCREW",
        "SENSOR",
        "SERVO_DRIVER",
        "TOOL_MAGAZINE"
      ]
    },
    "description": {
      "type": "string"
    },
    "number": {
      "type": "string"
    },
    "configs": {
      "$ ref": "#/definitions/ConfigsGen"
    },
    "dataItems": {
      "$ ref": "#/definitions/DataItems"
    },
    "components": {
      "$ ref": "#/definitions/Components"
    }
  }
}

```



```

    }
  }
},
"Reason": {
  "type": "string"
},
"Code": {
  "enum": [
    "OK",
    "NG"
  ]
},
"Error": {
  "type": "integer"
},
"CMD_Register_Request": {
  "type": "object",
  "required": [
    "deviceid"
  ],
  "properties": {
    "deviceid": {
      "$ref": "#/definitions/DeviceId"
    }
  }
},
"CMD_Probe_Version": {
  "type": "object",
  "required": [
    "version"
  ],
  "properties": {
    "version": {
      "type": "string"
    }
  }
},
"CMD_Discovery_Response": {
  "type": "object",
  "required": [

```

```

    "code"
  ],
  "properties": {
    "code": {
      "$ ref": "#/definitions/Code"
    },
    "reason": {
      "$ ref": "#/definitions/Reason"
    },
    "deviceids": {
      "type": "array",
      "minItems": 0,
      "items": {
        "$ ref": "#/definitions/DeviceId"
      }
    }
  }
},
"CMD_Probe_Query_Response": {
  "type": "object",
  "required": [
    "code"
  ],
  "properties": {
    "code": {
      "$ ref": "#/definitions/Code"
    },
    "reason": {
      "$ ref": "#/definitions/Reason"
    },
    "probe": {
      "$ ref": "#/definitions/NC_Probe"
    }
  }
},
"CMD_Probe_Set_Request": {
  "type": "object",
  "required": [
    "probe"
  ],

```

```

    "properties": {
      "probe": {
        "$ ref": "#/definitions/NC_Probe"
      }
    }
  },
  "CMD_Probe_Set_Response": {
    "type": "object",
    "required": [
      "code"
    ],
    "properties": {
      "code": {
        "$ ref": "#/definitions/Code"
      },
      "reason": {
        "$ ref": "#/definitions/Reason"
      }
    }
  },
  "Query_Params": {
    "type": "object",
    "properties": {
      "operation": {
        "enum": [
          "get_length",
          "get_keys",
          "get_value",
          "get_attribute"
        ],
        "default": "get_value"
      },
      "indexes": {
        "type": "array",
        "minItems": 0,
        "items": {
          "type": "string",
          "pattern": "~(0|[1-9][0-9]*)~(0|[1-9][0-9]*)?"
        }
      }
    }
  },

```

```

"keys": {
  "type": "array",
  "minItems": 0,
  "items": {
    "type": "string"
  }
},
"offset": {
  "$ref": "#/definitions/nonNegativeIntegerDefault0"
},
"length": {
  "$ref": "#/definitions/nonNegativeIntegerDefault0"
}
},
"CMD_Query_Request": {
  "type": "object",
  "required": [
    "ids"
  ],
  "properties": {
    "ids": {
      "type": "array",
      "minItems": 0,
      "items": {
        "type": "object",
        "required": [
          "id"
        ],
        "properties": {
          "id": {
            "$ref": "#/definitions/DeviceId"
          },
          "params": {
            "$ref": "#/definitions/Query_Params"
          }
        }
      }
    }
  }
}

```

```

    },
    "CMD_Query_Response": {
      "type": "object",
      "required": [
        "values"
      ],
      "properties": {
        "values": {
          "type": "array",
          "minItems": 0,
          "items": {
            "type": "object",
            "required": [
              "id",
              "code",
              "values"
            ],
            "properties": {
              "id": {
                "type": "string"
              },
              "code": {
                "$ref": "#/definitions/Code"
              },
              "reason": {
                "$ref": "#/definitions/Reason"
              },
              "values": {
                "$ref": "#/definitions/ValueTypes"
              },
              "offset": {
                "$ref": "#/definitions/nonNegativeIntegerDefault0"
              },
              "length": {
                "$ref": "#/definitions/nonNegativeIntegerDefault0"
              },
              "params": {
                "$ref": "#/definitions/Query_Params"
              }
            }
          }
        }
      }
    }
  }

```

```

    }
  }
},
"Set_params": {
  "type": "object",
  "required": ["values"],
  "properties": {
    "operation": {
      "enum": [
        "add",
        "delete",
        "set_value"
      ],
      "default": "set_value"
    },
    "indexes": {
      "type": "array",
      "minItems": 0,
      "items": {
        "type": "string",
        "pattern": "^(0|[1-9][0-9]*)-(0|[1-9][0-9]*)?"
      }
    },
    "keys": {
      "type": "array",
      "minItems": 0,
      "items": {
        "type": "string"
      }
    },
    "offset": {
      "$ref": "#/definitions/nonNegativeIntegerDefault0"
    },
    "length": {
      "$ref": "#/definitions/nonNegativeIntegerDefault0"
    },
    "values": {
      "anyOf": [
        {

```

```

        "type": "string"
      },
      {
        "type": "number"
      }
    ]
  }
}
},
"nonNegativeInteger": {
  "type": "integer",
  "minimum": 0
},
"nonNegativeIntegerDefault0": {
  "allOf": [
    { "$ref": "#/definitions/nonNegativeInteger" },
    { "default": 0 }
  ]
},
"CMD_Set_Request": {
  "type": "object",
  "required": [
    "values"
  ],
  "properties": {
    "values": {
      "type": "array",
      "minItems": 0,
      "items": {
        "type": "object",
        "required": [
          "id"
        ],
        "properties": {
          "id": {
            "$ref": "#/definitions/DeviceId"
          },
          "offset": {
            "$ref": "#/definitions/nonNegativeIntegerDefault0"
          }
        }
      }
    }
  }
}

```

```

    "length": {
      "$ref": "#/definitions/nonNegativeIntegerDefault0"
    },
    "params": {
      "$ref": "#/definitions/Set_params"
    }
  }
}
},
"CMD_Set_Response": {
  "type": "object",
  "required": [
    "results"
  ],
  "properties": {
    "results": {
      "type": "array",
      "minItems": 0,
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "string"
          },
          "code": {
            "$ref": "#/definitions/Code"
          },
          "reason": {
            "$ref": "#/definitions/Reason"
          },
          "params": {
            "params": {
              "$ref": "#/definitions/Set_params"
            }
          }
        }
      }
    }
  }
}
}

```

```

    }
  },
  "CMD_Sample": {
    "type": "object",
    "required": [
      "id",
      "beginTime",
      "data"
    ],
    "properties": {
      "id": {
        "type": "string"
      },
      "beginTime": {
        "$ref": "#/definitions/TimeStamp"
      },
      "data": {
        "type": "array",
        "minItems": 0,
        "items": {
          "type": "object",
          "properties": {
            "data": {
              "type": "array",
              "minItems": 0,
              "items": {
                "$ref": "#/definitions/ValueTypes"
              }
            }
          }
        }
      }
    }
  },
  "DeviceId": {
    "anyOf": [
      {
        "type": "string"
      },
      {

```

```

    "type": "string",
    "pattern":
"^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$ ",
    "description": "UUID"
  }
]
},
"ValueTypeName": {
  "enum": [
    "ACCELERATION",
    "ANGLE",
    "ANGULAR_ACCELERATION",
    "ANGULAR_VELOCITY",
    "CONCENTRATION",
    "CONDUCTIVITY",
    "CURRENT",
    "DISPLACEMENT",
    "ENERGY",
    "FLOW",
    "FREQUENCY",
    "LENGTH",
    "MASS",
    "PERIOD",
    "POSITION",
    "POWER",
    "POWER_FACTOR",
    "PRESSURE",
    "RESISTANCE",
    "ROTATION_SPEED",
    "SPEED",
    "TEMPERATURE",
    "TORQUE",
    "VISCOSITY",
    "VOLT_AMPERE",
    "CATAGORY",
    "CREATE_TIME",
    "CREATOR",
    "IP",
    "MANUFACTURER",
    "MODEL",

```

"NAME",
"PARAMETER",
"SERIAL_NUMBER",
"STATUS",
"UPPER_LIMIT",
"USER_ROLE",
"VARIABLE",
"VERSION",
"WARNING",
"WORK_MODE",
"AUTO_MODE",
"AXIS_BACKLASH",
"AXIS_LOAD",
"AXIS_TOREFPOINT",
"BLOCK_SKIP_MODE",
"CHIP_REMOVAL",
"COMPACITY",
"CONSOLE",
"COOLANT_ON",
"COORDINATE",
"CYCLE_START",
"DOOR_CLOSED",
"EDIT_MODE",
"EMG",
"FEED_HOLD",
"FEED_OVERRIDE",
"FEED_SPEED",
"FILE",
"FIXTURE_MODE",
"HANDLE_MODE",
"INCREMENTAL_MODE",
"JOG_MODE",
"LIGHTING_ON",
"LINE_NUMBER",
"LUBRICANT_ON",
"MACHINE_LOCKED",
"MDI_MODE",
"MST_LOCKED",
"OPT_STOP_MODE",
"PATH_LEFT_LENGTH",



```

"PART",
"PART_COUNT",
"PROGRAM",
"PROGRAM_MODE",
"PROGRAM_NUMBER",
"RAPID_OVERRIDE",
"RATE",
"REF_MODE",
"RESET_MODE",
"RESIDUAL",
"SINGLE_BLOCK_MODE",
"SPINDLE_OVERRIDE",
"SPINDLE_SPEED",
"SUBPROGRAM",
"TOOL",
"TOOL_CHANGE",
"TOOL_CHANGING",
"TOOL_PARAM",
"TOOL_NUMBER",
"TOOL_POSITION"
]
},

"unitType": {
  "description": "单位类型",
  "enum": [
    "AMPERE",
    "CELSIUS",
    "COUNT",
    "DEGREE",
    "DEGREE/SECOND",
    "DEGREE/SECOND^2",
    "HERTZ",
    "JOULE",
    "KILOGRAM",
    "LITER",
    "LITER/SECOND",
    "MILLIMETER",
    "MILLIMETER/SECOND",
    "MILLIMETER/SECOND^2",

```

```

    "MILLIMETER_3D",
    "NEWTON",
    "NEWTON_METER",
    "PASCAL",
    "PERCENT",
    "PH",
    "REVOLUTION/MINUTE",
    "SECOND",
    "VOLT",
    "WATT",
    "OHM",
    "SOUND_LEVEL",
    "SIEMENS/METER",
    "MICRO_RADIAN",
    "PASCAL_SECOND",
    "VOLT_AMPERE",
    "VOLT_AMPERE_REACTIVE",
    "WATT_SECOND",
    "DECIBEL",
    "CENTIPOISE",
    "DEGREE/MINUTE",
    "FAHRENHEIT",
    "KILOWATT",
    "KILOWATT_HOUR",
    "MILLIMETER/MINUTE",
    "RADIAN",
    "RADIAN/MINUTE",
    "RADIAN/SECOND",
    "RADIAN/SECOND^2",
    "REVOLUTION/SECOND",
    "OTHER"
  ]
},
"TimeStamp": {
  "anyOf": [
    {
      "format": "date-time"
    },
    {
      "type": "integer"
    }
  ]
}

```

```

    },
    {
      "type": "string"
    }
  ]
},
"ValueTypes": {
  "anyOf": [
    {
      "type": "null"
    },
    {
      "$ref": "#/definitions/ACCELERATION"
    },
    {
      "$ref": "#/definitions/ANGLE"
    },
    {
      "$ref": "#/definitions/ANGULAR_ACCELERATION"
    },
    {
      "$ref": "#/definitions/ANGULAR_VELOCITY"
    },
    {
      "$ref": "#/definitions/CONCENTRATION"
    },
    {
      "$ref": "#/definitions/CONDUCTIVITY"
    },
    {
      "$ref": "#/definitions/CURRENT"
    },
    {
      "$ref": "#/definitions/DISPLACEMENT"
    },
    {
      "$ref": "#/definitions/ENERGY"
    },
    {
      "$ref": "#/definitions/FLOW"
    }
  ]
}

```

```

    },
    {
      "$ref": "#/definitions/FREQUENCY"
    },
    {
      "$ref": "#/definitions/LENGTH"
    },
    {
      "$ref": "#/definitions/MASS"
    },
    {
      "$ref": "#/definitions/PERIOD"
    },
    {
      "$ref": "#/definitions/POSITION"
    },
    {
      "$ref": "#/definitions/POWER"
    },
    {
      "$ref": "#/definitions/POWER_FACTOR"
    },
    {
      "$ref": "#/definitions/PRESSURE"
    },
    {
      "$ref": "#/definitions/RESISTANCE"
    },
    {
      "$ref": "#/definitions/ROTATION_SPEED"
    },
    {
      "$ref": "#/definitions/SPEED"
    },
    {
      "$ref": "#/definitions/TEMPERATURE"
    },
    {
      "$ref": "#/definitions/TORQUE"
    },
    },

```

```
{
  "$ ref": "#/definitions/VISCOSITY"
},
{
  "$ ref": "#/definitions/VOLT_AMPERE"
},
{
  "$ ref": "#/definitions/CATAGORY"
},
{
  "$ ref": "#/definitions/CREATE_TIME"
},
{
  "$ ref": "#/definitions/CREATOR"
},
{
  "$ ref": "#/definitions/IP"
},
{
  "$ ref": "#/definitions/MANUFACTURER"
},
{
  "$ ref": "#/definitions/MODEL"
},
{
  "$ ref": "#/definitions/NAME"
},
{
  "$ ref": "#/definitions/PARAMETER"
},
{
  "$ ref": "#/definitions/SERIAL_NUMBER"
},
{
  "$ ref": "#/definitions/STATUS"
},
{
  "$ ref": "#/definitions/UPPER_LIMIT"
},
{
```

```

    "$ref": "#/definitions/USER_ROLE"
  },
  {
    "$ref": "#/definitions/VARIABLE"
  },
  {
    "$ref": "#/definitions/VERSION"
  },
  {
    "$ref": "#/definitions/WARNING"
  },
  {
    "$ref": "#/definitions/WORK_MODE"
  },
  {
    "$ref": "#/definitions/AUTO_MODE"
  },
  {
    "$ref": "#/definitions/AXIS_BACKLASH"
  },
  {
    "$ref": "#/definitions/LOAD"
  },
  {
    "$ref": "#/definitions/TOREFPOINT"
  },
  {
    "$ref": "#/definitions/BLOCK_SKIP_MODE"
  },
  {
    "$ref": "#/definitions/CHIP_REMOVAL"
  },
  {
    "$ref": "#/definitions/COMPACITY"
  },
  {
    "$ref": "#/definitions/CONSOLE"
  },
  {
    "$ref": "#/definitions/COOLANT_ON"
  }

```



```
},  
{  
  "$ ref": "#/definitions/COORDINATE"  
},  
{  
  "$ ref": "#/definitions/CYCLE_START"  
},  
{  
  "$ ref": "#/definitions/DOOR_CLOSED"  
},  
{  
  "$ ref": "#/definitions/EDIT_MODE"  
},  
{  
  "$ ref": "#/definitions/EMG"  
},  
{  
  "$ ref": "#/definitions/FEED_HOLD"  
},  
{  
  "$ ref": "#/definitions/FEED_OVERRIDE"  
},  
{  
  "$ ref": "#/definitions/FEED_SPEED"  
},  
{  
  "$ ref": "#/definitions/FILE"  
},  
{  
  "$ ref": "#/definitions/FIXTURE_MODE"  
},  
{  
  "$ ref": "#/definitions/HANDLE_MODE"  
},  
{  
  "$ ref": "#/definitions/INCREMENTAL_MODE"  
},  
{  
  "$ ref": "#/definitions/JOG_MODE"  
},
```

```
{
  "$ref": "#/definitions/LIGHTING_ON"
},
{
  "$ref": "#/definitions/LINE_NUMBER"
},
{
  "$ref": "#/definitions/LUBRICANT_ON"
},
{
  "$ref": "#/definitions/MACHINE_LOCKED"
},
{
  "$ref": "#/definitions/MDI_MODE"
},
{
  "$ref": "#/definitions/MST_LOCKED"
},
{
  "$ref": "#/definitions/OPT_STOP_MODE"
},
{
  "$ref": "#/definitions/PATH_LEFT_LENGTH"
},
{
  "$ref": "#/definitions/PART"
},
{
  "$ref": "#/definitions/PART_COUNT"
},
{
  "$ref": "#/definitions/PROGRAM"
},
{
  "$ref": "#/definitions/PROGRAM_MODE"
},
{
  "$ref": "#/definitions/PROGRAM_NUMBER"
},
{
```

```

    "$ ref": "#/definitions/RAPID_OVERRIDE"
  },
  {
    "$ ref": "#/definitions/RATE"
  },
  {
    "$ ref": "#/definitions/REF_MODE"
  },
  {
    "$ ref": "#/definitions/RESET_MODE"
  },
  {
    "$ ref": "#/definitions/RESIDUAL"
  },
  {
    "$ ref": "#/definitions/SINGLE_BLOCK_MODE"
  },
  {
    "$ ref": "#/definitions/SPINDLE_MODE"
  },
  {
    "$ ref": "#/definitions/SPINDLE_OVERRIDE"
  },
  {
    "$ ref": "#/definitions/SPINDLE_SPEED"
  },
  {
    "$ ref": "#/definitions/SUBPROGRAM"
  },
  {
    "$ ref": "#/definitions/TOOL"
  },
  {
    "$ ref": "#/definitions/TOOL_CHANGE"
  },
  {
    "$ ref": "#/definitions/TOOL_CHANGING"
  },
  {
    "$ ref": "#/definitions/TOOL_PARAM"
  }

```

```

    },
    {
      "$ref": "#/definitions/TOOL_NUMBER"
    },
    {
      "$ref": "#/definitions/TOOL_POSITION"
    }
  ]
},
"ACCELERATION": {
  "type": "number",
  "description": "加速度,毫米每二次方秒"
},
"ANGLE": {
  "type": "number",
  "description": "角位置,弧度"
},
"ANGULAR_ACCELERATION": {
  "type": "number",
  "description": "角加速度,弧度每二次方秒"
},
"ANGULAR_VELOCITY": {
  "type": "number",
  "description": "角速度,弧度每秒"
},
"CONCENTRATION": {
  "type": "number",
  "description": "浓度,百分比"
},
"CONDUCTIVITY": {
  "type": "number",
  "description": "导电力,西门子每米"
},
"CURRENT": {
  "type": "number",
  "description": "电流,安培"
},
"DISPLACEMENT": {
  "type": "number",
  "description": "位移,毫米"
}

```

```
},  
"ENERGY": {  
  "type": "number",  
  "description": "功耗,千瓦时"  
},  
"FLOW": {  
  "type": "number",  
  "description": "流体瞬时流量,升每秒"  
},  
"FREQUENCY": {  
  "type": "number",  
  "description": "频率,赫兹"  
},  
"LENGTH": {  
  "type": "number",  
  "description": "长度,毫米"  
},  
"MASS": {  
  "type": "number",  
  "description": "质量,千克"  
},  
"PERIOD": {  
  "type": "number",  
  "description": "周期,秒"  
},  
"POSITION": {  
  "type": "number",  
  "description": "位置,毫米"  
},  
"POWER": {  
  "type": "number",  
  "description": "功率,瓦特"  
},  
"POWER_FACTOR": {  
  "type": "number",  
  "description": "功率因数"  
},  
"PRESSURE": {  
  "type": "number",  
  "description": "压强,帕斯卡"
```

```

},
"RESISTANCE": {
  "type": "number",
  "description": "电阻,欧姆"
},
"ROTATION_SPEED": {
  "type": "number",
  "description": "转速,转每分钟"
},
"SPEED": {
  "type": "number",
  "description": "速度,毫米每分钟"
},
"TEMPERATURE": {
  "type": "number",
  "description": "温度,摄氏度"
},
"TORQUE": {
  "type": "number",
  "description": "扭矩,牛顿米"
},
"VISCOSITY": {
  "type": "number",
  "description": "黏度,帕斯卡秒"
},
"VOLT_AMPERE": {
  "type": "number",
  "description": "视在功率,伏安"
},
"CREATE_TIME": {
  "type": "string",
  "description": "创建时间"
},
"CREATOR": {
  "type": "string",
  "description": "创建者"
},
"MANUFACTURER": {
  "type": "string",
  "description": "厂商"
}

```

```

},
"MODEL": {
  "type": "string",
  "description": "型号"
},
"NAME": {
  "type": "string",
  "description": "名称"
},
"NUMBER": {
  "type": "number",
  "description": "编号"
},
"PARAMETER": {
  "type": "array"
},
"STATUS": {
  "oneOf": [
    {
      "type": "integer",
      "enum": [
        0,
        1,
        2
      ],
      "description": "运行、空闲、进给保持"
    },
    {
      "type": "string",
      "enum": [
        "running",
        "free",
        "holding"
      ],
      "description": "运行、空闲、进给保持"
    }
  ]
},
"VERSION": {

```

```

    "type": "string",
    "description": "版本"
  },
  "WARNING": {
    "type": "object",
    "properties": {
      "number": {
        "type": "string",
        "description": "报警编号"
      },
      "text": {
        "type": "string",
        "description": "报警内容"
      }
    },
    "other": {
      "type": "object"
    }
  }
},

"CONSOLE": {
  "type": "string",
  "description": "控制台"
},
"COORDINATE": {
  "anyOf": [
    {
      "type": "object",
      "properties": {
        "x": {
          "type": "number"
        },
        "y": {
          "type": "number"
        },
        "z": {
          "type": "number"
        }
      }
    }
  ]
}

```

```

    }
  },
  {
    "type": "object",
    "properties": {
      "a": {
        "type": "number"
      },
      "b": {
        "type": "number"
      },
      "c": {
        "type": "number"
      }
    }
  },
  {
    "type": "object",
    "properties": {
      "u": {
        "type": "number"
      },
      "v": {
        "type": "number"
      },
      "w": {
        "type": "number"
      }
    }
  }
],
  "description": "坐标系"
},
"FEED_OVERRIDE": {
  "type": "number",
  "description": "进给倍率"
},
"FEED_SPEED": {
  "type": "number",
  "description": "进给速度,毫米每秒"
}

```

```
},  
"FILE": {  
  "$ref": "#/definitions/DICT",  
  "description": "文件"  
},  
"LINE_NUMBER": {  
  "type": "string",  
  "description": "程序行号"  
},  
"PATH_LEFT_LENGTH": {  
  "type": "number",  
  "description": "剩余进给,毫米"  
},  
"PART": {  
  "type": "object",  
  "description": "工件",  
  "properties": {  
    "id": {  
      "type": "string",  
      "description": "编号"  
    },  
    "name": {  
      "type": "string",  
      "description": "名称"  
    },  
    "material": {  
      "type": "string",  
      "description": "材料"  
    },  
    "length": {  
      "type": "number",  
      "description": "长度"  
    },  
    "width": {  
      "type": "number",  
      "description": "宽度"  
    },  
    "height": {  
      "type": "number",  
      "description": "高度"  
    }  
  }  
}
```

```

    },
    "snap_shot": {
      "$ ref": "#/definitions/FILE",
      "description": "零件图片文件"
    },
    "geometry_file": {
      "$ ref": "#/definitions/FILE",
      "description": "零件 CAD 模型文件"
    },
    "other": {
      "type": "object"
    }
  }
},
"PART_COUNT": {
  "type": "string",
  "description": "加工件数"
},
"PROGRAM": {
  "type": "string",
  "description": "主程序名"
},
"PROGRAM_NUMBER": {
  "type": "number",
  "description": "当前程序号"
},
"SITE": {
  "type": "string",
  "description": "AGV 工作站点"
},
"SPINDLE_OVERRIDE": {
  "type": "number",
  "description": "主轴倍率"
},
"SUBPROGRAM": {
  "type": "string",
  "description": "子程序名"
},
"TOOL": {

```

```

    "$ ref": "#/definitions/LIST",
    "description": "刀具"
  },
  "TOOLPARAM": {
    "type": "object",
    "description": "刀具参数",
    "properties": {
      "id": {
        "type": "string",
        "description": "刀具编号"
      },
      "kind": {
        "type": "string",
        "description": "需再确认"
      },
      "radius": {

        "type": "string",
        "description": "半径"
      },
      "length": {
        "type": "string",
        "description": "长度"
      },
      "time_usage": {
        "type": "number",
        "description": "使用时间"
      },
      "count_usage": {
        "type": "number",
        "description": "使用次数"
      },
      "length_process": {
        "type": "number",
        "description": "切削里程"
      },
      "radius_abaration": {
        "type": "string",
        "description": "半径补偿"
      },
      "length_abaration": {

```

```

        "type": "string",
        "description": "长度补偿"
    }
}
},
"TOOL_NUMBER": {
    "type": "number",
    "description": "当前刀具号"
},
"TYPE": {
    "type": "string",
    "description": "类型"
},
"VARIABLE": {
    "$ ref": "# /definitions/LIST",
    "description": "运行变量"
},
"WORK_MODE": {
    "enum": [
"manual", "auto"
], "description": "运行模式"
},
"dict": {
    "type": "object"
},
"LIST": {
    "type": "array"
}
},
"default": true
}

```

附录 B

(资料性)

数控机床模型描述示例

本附录给出了一个数控机床模型描述示例,该示例描述了一台数控机床的逻辑模型,给出了一个完整的数控机床模型文件。

数控机床逻辑模型见图 B.1。其中:

- 图 B.1 中,各个节点所代表的对象的描述只提取了关键的信息,比如 id、type,详细的对象描述见对应的 JSON 格式描述的对象;
- 图 B.1 中, id 为“0103”所描述的设备对象 devices[0]的 id 为“010302”“010303”“010304”“010305”“010306”“010307”,共 6 个数据项对象。为了使得文档清晰,省略了其中的“010304”“010305”“010306”这 3 个数据对象对应的节点;
- 图 B.1 中, id 为“0103”所描述的设备对象 devices[0]的 id 为“010308”“010309”“010310”“010311”“010312”,共 5 个组件对象。其中“010308”“010309”“010310”“010311”这 4 个组件对象的逻辑结构所表达的语义是一致的,为了使得文档清晰,图中只给出了“010308”对象的节点的下层结构;
- 图 B.1 中, id 为“0103”所描述的设备对象的组件对象“010312”下层的 id 为“01031201”“01031202”“01031203”“01031204”“01031205”“01031206”“01031207”“01031215”,共 8 个配置对象。为了使得文档清晰,图中只给出了“01031201”“01031202”“01031203”“01031215”共 4 个配置对象对应的节点;
- 在数控机床模型文件示例中, id 为“sample_channel0”的采样通道对象有 ids 数组,数组中的每一个“键-值对”JSON 对象均代表一个采样点。例如:{"id": "010302"}这个 JSON 对象表示将从 id 为“010302”的数据项对象中读取机床状态信息。



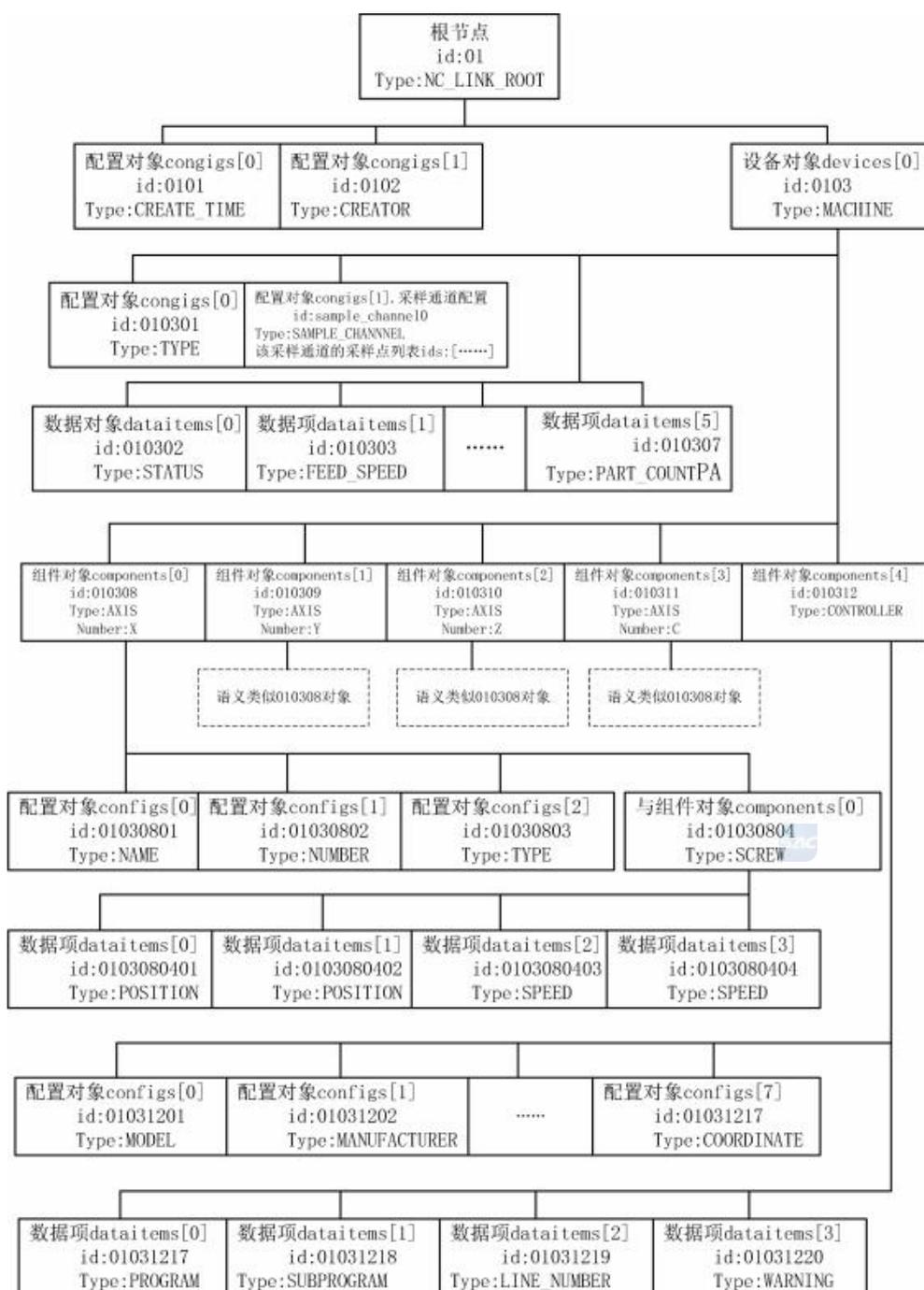


图 B.1 数控机床逻辑模型

数控机床模型文件示例如下：

```
{
  "id": "01",
  "type": "NC_LINK_ROOT",
  "name": "数控机床模型文件",
  "devices": [
```

```

{
  "type": "MACHINE",
  "id": "0103",
  "name": "数控机床",
  "description": "数控机床",
  "version": "1.0",
  "configs": [
    {
      "id": "010301",
      "type": "TYPE",
      "name": "机床类型"
    },
    {
      "id": "sample_channel0",
      "type": "SAMPLE_CHANNEL",
      "name": "采样通道",
      "sampleInterval": 2000,
      "uploadInterval": 2000,
      "ids": [
        {
          "id": "010302"
        },
        {
          "id": "010303"
        },
        {
          "id": "010305"
        },
        {
          "id": "010306"
        },
        {
          "id": "010307"
        },
        {
          "id": "01031220"
        }
      ]
    }
  ]
}

```

```
"id": "sample_channel1",
"type": "SAMPLE_CHANNEL",
"name": "采样通道",
"sampleInterval": 100,
"uploadInterval": 300,
"ids": [
  {
    "id": "0103080402"
  },
  {
    "id": "0103080404"
  },
  {
    "id": "0103090402"
  },
  {
    "id": "0103090404"
  },
  {
    "id": "0103100402"
  },
  {
    "id": "0103100404"
  },
  {
    "id": "01031217"
  },
  {
    "id": "01031218"
  },
  {
    "id": "01031219"
  }
]
},
"dataItems": [
  {
    "id": "010302",
    "name": "机床状态",
```



```

        "type": "STATUS"
    },
    {
        "id": "010303",
        "name": "进给速度",
        "type": "FEED_SPEED"
    },
    {
        "id": "010305",
        "name": "进给倍率",
        "type": "FEED_OVERRIDE"
    },
    {
        "id": "010306",
        "name": "主轴倍率",
        "type": "SPDL_OVERRIDE"
    },
    {
        "id": "010307",
        "name": "加工件数",
        "type": "PART_COUNT"
    }
],
"components": [
    {
        "type": "AXIS",
        "number": "X",
        "id": "010308",
        "name": "X 轴",
        "description": "",
        "configs": [
            {
                "id": "01030803",
                "name": "轴类型",
                "type": "TYPE",
                "value": "linear"
            }
        ]
    },
    {

```

```

    "type": "SCREW",
    "id": "01030804",
    "name": "X 轴丝杠",
    "description": "",
    "dataItems": [
      {
        "id": "0103080402",
        "name": "实际位置",
        "type": "POSITION"
      },
      {
        "id": "0103080404",
        "name": "实际速度",
        "type": "SPEED"
      }
    ]
  },
  {
    "type": "AXIS",
    "number": "Y",
    "id": "010309",
    "name": "Y 轴",
    "description": "",
    "configs": [
      {
        "id": "01030903",
        "name": "轴类型",
        "type": "TYPE",
        "value": "linear"
      }
    ],
    "components": [
      {
        "type": "SCREW",
        "id": "01030904",
        "name": "Y 轴丝杠",
        "description": "",
        "dataItems": [

```

```

    {
      "id": "0103090402",
      "name": "实际位置",
      "type": "POSITION"
    },
    {
      "id": "0103090404",
      "name": "实际速度",
      "type": "SPEED"
    }
  ]
}
]
},
{
  "type": "AXIS",
  "number": "Z",
  "id": "010310",
  "name": "Z 轴",
  "description": "",
  "configs": [
    {
      "id": "01031003",
      "name": "轴类型",
      "type": "TYPE",
      "value": "linear"
    }
  ],
  "components": [
    {
      "type": "SCREW",
      "id": "01031004",
      "name": "Z 轴丝杠",
      "description": "",
      "dataItems": [
        {
          "id": "0103100402",
          "name": "实际位置",
          "type": "POSITION"
        },

```

```

        {
            "id": "0103100404",
            "name": "实际速度",
            "type": "SPEED"
        }
    ]
}
]
},
{
    "type": "AXIS",
    "number": "C",
    "id": "010311",
    "name": "C 轴",
    "description": "",
    "configs": [
        {
            "id": "01031101",
            "name": "轴名",
            "type": "NAME",
            "value": "C"
        },
        {
            "id": "01031102",
            "name": "轴号",
            "type": "NUMBER",
            "value": 5
        },
        {
            "id": "01031103",
            "name": "轴类型",
            "type": "TYPE",
            "value": "rotary"
        }
    ],
    "components": [
        {
            "type": "MOTOR",
            "id": "01031104",
            "name": "C 轴电机",

```

```

      "description": "",
      "dataItems": [
        {
          "id": "0103110402",
          "name": "实际位置",
          "type": "POSITION"
        },
        {
          "id": "0103110404",
          "name": "实际速度",
          "type": "SPEED"
        }
      ]
    }
  ],
},
{
  "type": "CONTROLLER",
  "id": "010312",
  "name": "数控系统",
  "description": "",
  "configs": [
    {
      "id": "01031201",
      "type": "MODEL",
      "name": "数控系统型号"
    },
    {
      "id": "01031202",
      "type": "manufacturer",
      "name": "数控系统厂家"
    },
    {
      "id": "01031207",
      "type": "TOOL",
      "name": "刀具参数",
      "dataType": "LIST",
      "settable": true
    },
    {
      "id": "01031215",
      "type": "COORDINATE",

```

```
    "name": "坐标系",
    "dataType": "LIST",
    "settable": true
  }
],
"dataItems": [
  {
    "id": "01031217",
    "type": "PROGRAM",
    "name": "主程序名",
    "dataType": "DICT"
  },
  {
    "id": "01031218",
    "type": "SUBPROGRAM",
    "name": "子程序名",
    "dataType": "DICT"
  },
  {
    "id": "01031219",
    "type": "LINE_NUMBER",
    "name": "指令行号"
  },
  {
    "id": "01031220",
    "type": "WARNING",
    "name": "报警"
  }
]
}
]
}
```

附 录 C

(资料性)

测试记录表

本附录给出了测试记录表的样例,测试记录见表 C.1。

表 C.1 测试记录表

基本信息					
测试对象	<input type="checkbox"/> 1 适配器 <input type="checkbox"/> 2 机床含适配器 <input type="checkbox"/> 3 机床含适配器和代理器 <input type="checkbox"/> 4 代理器				
送测单位		测试对象版本			
测试系统		测试系统版本			
测试日期		测试人员			
测试评价					
测试项目		测试结果		备注	
连接测试	终端注册	必选项	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
	终端探测	必选项	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
	评价结论	连接测试: <input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		若连接测试不通过,则测试对象测试不通过	
模型文件数据项定义检查	组件对象数量	规范类型		项	
		自定义类型		项	
	数据对象数量	规范类型		项	
		自定义类型		项	
	采样通道数量	适配器提供		个	
内构方法数量	适配器提供		个		
数据传输接口功能测试	模型侦测	必选项	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
	模型设置	必选项	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
	数据查询	必选项	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
	状态通知	必选项	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
	—	测试对象 1~3 选填	测试对象 4 选填		
	数据设置	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
	采样通道采样	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
	内构方法	调用	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过	
		状态	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过	
		控制	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过	
		结果	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过	
	事件	注册	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过	
		查询	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过	
		注销	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过	
	动态采样	注册	<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过	
采样		<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		
注销		<input type="checkbox"/> 支持 <input type="checkbox"/> 不支持	<input type="checkbox"/> 通过 <input type="checkbox"/> 不通过		

表 C.1 测试记录表 (续)

测试过程记录：			
测试评价结论：			
测试评价人员签字：		日期：	

