

嵌入式 DeviceNet 接口卡
技术说明书

Ver 2.1

上海泗博自动化技术有限公司

DeviceNet产品开发中心

2006-9-2

1.1 模块介绍

本产品在DeviceNet中为仅组2服务器。
外形尺寸特别紧凑，为 58mm * 58 mm。

连接支持：

- ① 显式报文
- ② I0轮询
- ③ 位选通
- ④ 状态改变

波特率和Mac ID可以通过编码开关进行设置，同时支持波特率自动侦听和Mac ID的软件设置。

1.1.1 原理框图

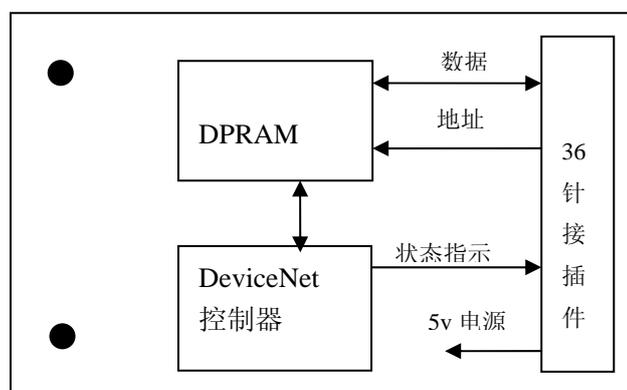
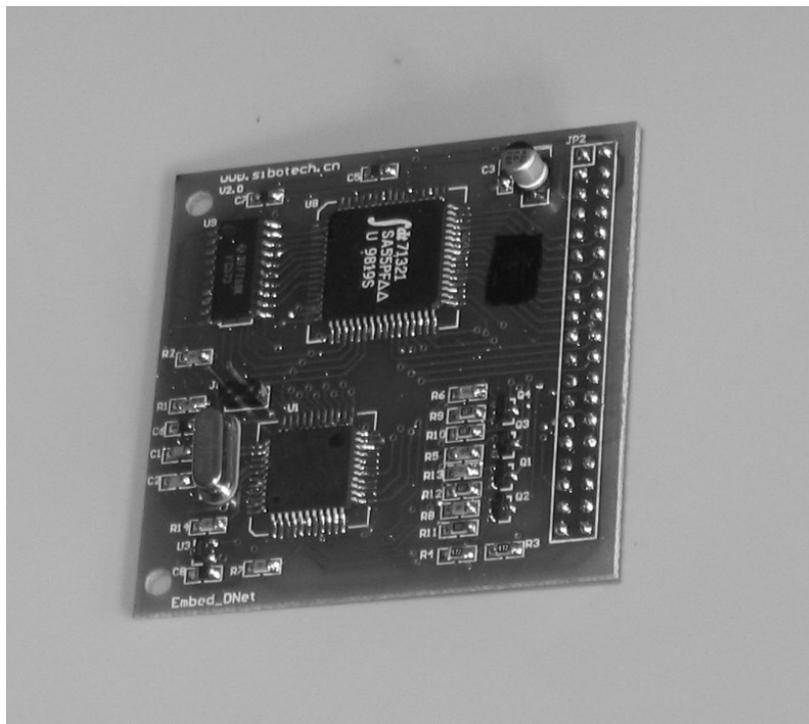


图1 原理框图

1.1.2 应用接口

本产品可以提供二种方法与设备进行数据交换：双口RAM、串行口。
双口RAM版后缀为P，串行口版后缀为S。

1.1.3 板卡一览



1.1.4 DeviceNet 连接器



图2 DeviceNet 连接器

管脚定义如下所示：

管脚	描述
1	V-
2	CAN_L
3	SHIELD
4	CAN_H
5	V+

表1 DeviceNet 连接器描述

1.1.5 板间连接器定义 (J1)

采用间距 2.54mm 的双排插针。



图3 间距 2mm 排针

管脚定义:

管脚	描述	管脚	描述
1, 5	VCC	2, 6	GND
3	网络状态红灯	4	网络状态绿灯
7	模块状态红灯	8	模块状态绿灯
9	A0	10	A1
11	A2	12	A3
13	A4	14	A5
15	A6	16	A7
17	A8	18	A9
19	D0	20	D1
21	D2	22	D3
23	D4	24	D5
25	D6	26	D7
27	BUSY_R	28	INT_R
29	OE_R	30	R/W_R
31	CE_R	32	RST1 复位信号
33	A10_R	34	悬空
35	串行口TXD	36	串行口RXD
37	CAN_TXD	38	CAN_RXD

表2 板间接插件接口

1.2 功能及操作

1.2.1 模块 DeviceNet 对象模型图

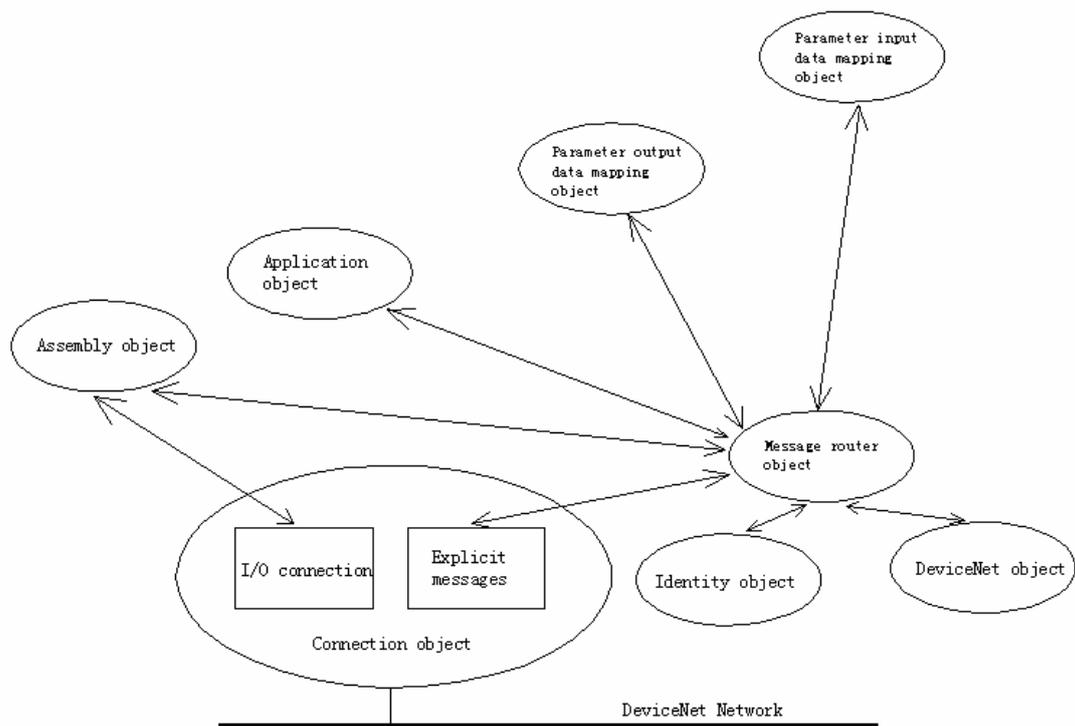


图4 Object Class

在DeviceNet网络中，每个节点都有一个Mac ID(网络地址)，Mac ID介于0~63。在网络中每个节点的Mac ID都是唯一的。另外在DeviceNet网络可以支持三种波特率分别为：125, 250 和500 kbit / sec。在同一个网络中所有参与通信的节点必须为相同的波特率。

本产品可以支持Mac ID和波特率通过软件设定。

1.2.2 波特率

波特率为自动监听，或者通过用户板卡与嵌入式卡通信设置。

1.2.3 Mac ID

Mac ID通过网络和双口RAM进行设置

1.3 LED 指示

本产品LED指示包括DeviceNet通信和串行口(UART)通信。

1.3.1 DeviceNet 通信指示

使用两个红绿双色灯，分别代表模块灯和网络灯（上方为模块灯MS下方为网络灯NS）。指示灯显示说明如下表：

表3 模块状态指示灯 (MS)

指 示 灯 状 态	含 义
关闭	未供电
绿色	工作正常
绿色闪烁	出错，不完全或不正确配置
红色闪烁	可恢复的故障

红	不可恢复的故障
红—绿闪烁	正在进行自检

表4 网络状态指示灯 (NS)

指示灯状态	含义
关闭	未通过重复MAC ID检测或DeviceNet电路未供电
绿色闪烁	设备在线但没有已建立的连接
绿色	设备在线且已建立了连接
红色闪烁	一个或多个I/O连接已经超时
红	设备检测到一个错误，且不能进行链路通信

1.4 DPRAM Memory map (仅对双口RAM版有效)

2kB 的 DPRAM 被分为以下几个存储区：

假定双口RAM基址为 DP_start。

序号	内存地址	名称	功能描述
1	DP_start + 0x7fd	MEM_CTRL	存储控制
2	DP_start + 0x0 ~ 0xff	OUTPUT_data	网络(PLC)输出(即设备从 PLC 接收的数据)
3	DP_start + 0x100 ~ 0x1ff	INPUT_data	网络(PLC)输入 (即设备发送给 PLC 的数据)
4	DP_start + 0x7E0	INPUT_LENGTH	网络输入有效数据长度 ^{注1} ，双字节
5	DP_start + 0x7E2	OUTPUT_LENGTH	网络输出有效数据长度，双字节
6	DP_start + 0x7FB	INPUT_data_lock	网络输入数据保护
7	DP_start + 0x7FC	OUTPUT_data_lock	网络输出数据保护
8	DP_start + 0x7DE	OUTPUT_CRCH	网络输出数据 CRC 校验高字节
9	DP_start + 0x7DF	OUTPUT_CRCL	网络输出数据 CRC 校验低字节
10	DP_start + 0x7E4	SOFTWARE_VERSION	软件版本
11	DP_start + 0x7E6	APP_CARD_SERIAL	用户卡序列号 (四字节, MSB FIRST)
12	DP_start + 0x7EA	APP_CARD_ID	用户卡身份识别号
13	DP_start + 0x7F0	CONFIG_MAC_ID	DeviceNet 网络地址设置字节
14	DP_start + 0x7F1	CONFIG_BAUD_RATE	DeviceNet 网络波特率设置字节
15	DP_start + 0x7FF	INTR	接口卡发送命令字节
16	DP_start + 0x7FE	INTL	接口卡接收命令字节 (保留)
17	DP_start + 0x7B0	PARAM1_data	DeviceNet 参数 1 映射
18	DP_start + 0x7B2	PARAM2_data	DeviceNet 参数 2 映射

19	DP_start + 0x7B4	PARAM3_data	DeviceNet 参数 3 映射
20	DP_start + 0x7B6	PARAM4_data	DeviceNet 参数 4 映射

表5 双口 RAM 数据定义表

注 1: 网络输入/输出 有效数据长度等同于 DeviceNet 连接输入/输出数据长度, 非常重要! 缺省设置为 32。数据为双字节数, 高位默认为零。

内存控制字节

上电以后, 嵌入式接口卡 (以下简称接口卡) 将尽快开始双口 RAM 自检, 双口 RAM 自检期间, MEM_CTRL 将为 0x55, 这个时候请不要往双口 RAM 里写任何输入数据, 否则自检将不能通过, 接口卡将可能挂起。

自检成功, 该字节置为 0x10, 这时接口卡将等待用户板卡向双口 RAM 写入配置数据, 大约等待 2s 左右, 这段时间如果没有写入数据, 板卡调用缺省值。

用户板卡写完后, 应置该字节内容为 0xee, 接口卡将根据该字节判断配置信息是否有效。

网络输出数据缓冲区

用户从该缓冲区内读数据。该数据区的有效数据的字节数由 网络输出有效数据长度 定义。注意该数据区有双重保护。

首先, 如果接口卡正在更新数据, OUTPUT_data_lock 将为非零值 (比如 1)。此时用户卡不要去读数据, 以免读到不完整的数据。

此外, 数据更新完成后, 接口卡将计算 CRC 校验值, 并放置在 OUTPUT_CRCH, OUTPUT_CRCL 中。用户可以使用, 也可以不使用 CRC。

网络输入数据缓冲区

用户向该缓冲区内写数据。该数据区的有效数据的字节数由 网络输入有效数据长度 定义。注意该数据区有保护。

首先, 如果接口卡正在更新数据, INPUT_data_lock 应非零值 (比如 1)。此时用户卡不要去写数据, 以免数据不完整。

DeviceNet 网络地址设置字节

用于设定 DeviceNet 网络地址。如果从网络接收到修改地址的指令并成功执行, 该地址将被修改为新设定的地址。

DeviceNet 网络波特率设置字节

用于设定 DeviceNet 网络波特率。接口卡模块支持自动波特率监听。

数值	波特率
0	125K
1	250K
2	500K
>=3	自动波特率侦听

发送命令字节

如果 DeviceNet 地址被成功修改, 该命令字节将被置为 0x11。(17) 接口卡写该字节将产生一个中断信号 (INTR)。用户卡读取该字节将清除中断信号。

接收命令字节
保留。

1.5 接口之间数据存取说明

上电后, DeviceNet处理器先判断RXD口线(Pin 36)的状态, 如果是低电平, 默认为串行口通信方式, 否则进行双口RAM自检。自检通过以后, 嵌入式DeviceNet接口卡进入正常工作状态, 采用并口通信模式。如果没有监测到板上有双口RAM, 也用串口模式通信。

如果用户希望自行配置 DeviceNet 网络配置信息, 首先检查内存控制字节: MEM_CTRL, (地址偏移量 0x7fd, 参见地址表), 当这个内存单元值等于(0x10), 表示双口 RAM 自检已通过, 可以写入配置信息。

注意: 千万要等到存储控制字节 = 0x10 (16) 以后, 才可以写入配置, 否则双口 RAM 自检将无法通过。用户需要在 2 秒内写完配置信息, 写完以后, 将内存控制字节置为 0xEE, 表示用户配置完成。此时嵌入式接口卡将调用用户配置。

客户如果使用自身的配置, 通过写 0xEE 使能以后, 必须注意配置的完整性。

配置信息至少包括 1. DeviceNet 地址, 0x7f0

2. DeviceNet 通信波特率

DeviceNet 网络输入字节长度 0x7E0, 0x7E1

网络输入字节 0x7E0 必须写零, 0x7E1 写入用户需要设置的长度。

3. DeviceNet 网络输出字节长度 0x7E2, 0x7E3

网络输出字节 0x7E2 必须写零, 0x7E3 写入用户需要设置的长度。

1.5.1 双口 RAM 通信

IO数据的存取说明:

用户卡更新DPRAM的输出数据空间, 模块需要数据时直接向DPRAM输出数据空间取数据; 输入数据空间由模块负责更新数据。

- 1) 输入数据空间 (提供给用户卡进行消费的数据), 用户需要读取此空间的数据时先判断是否被锁。
- 2) 输出数据空间 (用户卡提供给模块的消耗的数据), 用户需要在向此空间写数据时将此空间锁上, 模块只有在此空间没有锁定时才有权读取此空间的数据。
- 3) 参数的设置: 模块将设置命令+参数号+数据放入命令窗口, 并产生INTR中断告知用户卡进行处理。
- 4) 用户卡发现数据改变了, 就向DPRAM 0x7FEH中写数据产生INTL中断, 模块通过COS向主站报告 (如果DeviceNet配置了该连接的话)。

1.5.2 串行通信说明

串口通信是用户需要修改EDS文件, 将输入输出均变为8个字节。如果不知道如何修改, 请联系泗博公司。

串口通信时, 支持用户4个双字节参数。同时接口卡的输入/输出 8个字节 正是这四个参数的顺序排列。

- 1) 读取参数命令：0x0E+参数号
- 2) 设置参数命令：0x10+参数号+数据，用户卡接收到设置命令将参数数据改变后返回成功响应。

表6 请求命令格式

读取参数	设置参数
0x0E	0x10
参数号	参数号
BCC	数据长度
	数据低字节
	数据高字节
	BCC

表7 响应命令格式

读取参数成功响应	设置参数成功响应
0x06	0x06
参数号	
数据长度	
数据低字节	
数据高字节	
BCC	

读取参数错误响应	设置参数错误响应
0x15	0x15

注意：

- ◇ 模块与用户卡有两种方式（DPRAM和串口），但不能同时进行通信。另外当模块通过串口与用户进行通信时IO输入输出为8字节，通过DPRAM与用户卡进行通信时IO输入输出用户可选。
- ◇ 由协议所限模块在IO输入输出超过8字节时不能同时配置位选通连接。
- ◇ 产品Profile文件及EDS文件附录

附录 A: Profile 文件

DeviceNet 信息

产品代码:0x0A
设备类型为:0x0

DeviceNet 对象

类代码	对象
0x01	标识对象
0x02	报文路由对象
0x03	DeviceNet 对象
0x04	组合对象
0x05	连接对象

产品特定对象

类代码	对象
0xAB	应用对象
0xAA	参数对象
0x2B	应答处理对象

标识对象—类代码 0x01

类属性不支持

实例属性:

属性 ID	访问规则	名称	数据类型	属性值
1	读	生产商 ID	UINT	
2	读	设备类型	UINT	
3	读	产品代码	UINT	
4	读	版本 主要版本 次要版本	结构: UINT UINT	1 1
5	读	状态	WORD	0=没有分配 1=已分配
6	读	序列号	UDINT	唯一号码
7	读	产品名称: 字符串长度 ASCII 码串	结构: USINT STRING	
8	读	状态	USINT	0=不存在 1=设备自检 2=待机 3=可操作 4=重要可恢复故障 5=重要不可恢复故障

标识对象支持的公共服务:

服务代码	类	实例	服务名称
------	---	----	------

0X0E	不支持	支持	单个属性值(get_attribute_single)
0X05	不支持	支持	复位(reset)

报文路由对象—类代码 0X02

不支持任何实例属性

DeviceNet 对象—类代码 0X03

类属性

属性 ID	访问规则	名称	数据类型	属性值
1	读	版本	INT	

实例属性:

属性 ID	访问规则	名称	数据类型	属性值
1	读/设置	MAC ID	UINT	0—63
2	读	波特率	USINT	0=125K 1=250K 2=500K
5	读	分配信息: 分配连接选择 主站 MAC ID	结构: BYTE UINT	分配字节 地址=0—63 255=未分配
6	读	MAC ID 开关变 化	BOOL	0=没有变化 1=从最近一次上 电或复位以来已 经改变
8	读	MAC ID 开关值	USINT	范围 0-63

分配字节:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
N/a	应答抑制	循环	状态改变	多点轮询	位选通	轮询信息	显式信息

DeviceNet 对象支持的公共服务

服务代码	类	实例	服务名称
0X0E	支持	支持	读单个属性值(get_attribute_single)
0X10	不支持	支持	设置个属性值(set_attribute_single)
0X4B	不支持	支持	分配主从连接组 (allocate_master/slave_connection_set)
0X4C	不支持	支持	释放主从连接组 (release_master/slave_connection_set)

组合对象—类代码 0x04

不支持类属性

实例 A3,双口 RAM 输入 8byte (缺省)

属性 ID	访问规则	名称	数据类型	属性值
3	读	数据	Array of UINT	N/A

实例 B4h,双口 RAM 输入 32byte

属性 ID	访问规则	名称	数据类型	属性值
3	读	数据	Array of UINT	N/A

实例 A4,双口 RAM 输出 8byte (缺省)

属性 ID	访问规则	名称	数据类型	属性值
3	读	数据	Array of UINT	N/A

实例 B5h,双口 RAM 输出 32byte

属性 ID	访问规则	名称	数据类型	属性值
3	读	数据	Array of UINT	N/A

组合对象支持的公共服务:

服务代码	类	实例	服务名称
0X0E	不支持	支持	单个属性值(get_attribute_single)

连接对象一类代码 0x05

不支持类属性

连接对象实例 1 属性 (显示报文连接)

属性 ID	访问规则	名称	数据类型	属性值
1	读	状态	USINT	0=不存在 3=已建立
2	读	实例类型	USINT	0=显式报文
3	读	传输类触发	BYTE	0x83(服务器、类 3)
4	读	生产连接 ID	UINT	10xxxxx011(xxxxx=address)
5	读	消费连接 ID	UINT	10xxxxx100(xxxxx=address)
6	读	初始通信特征	BYTE	0x21 2=从站显式报文(生产,源) 1=主站显式请求(消费,目的)
7	读	生产连接数据大小	UINT	
8	读	消费连接数据大小	UINT	
9	读/设置	期望信息包速率	UINT	最小时间单位为 10ms (缺省 2500ms)
12	读/设置	看门狗定时器动作	USINT	1=自动删除 3=延迟删除
13	读	生产连接路径长度	UINT	0
14	读	生产连接路径		空
15	读	消费连接路径长度	UINT	0
16	读	消费连接路径		空

连接对象实例 2 属性(I/O 轮询连接)

属性 ID	访问规则	名称	数据类型	属性值
1	读	状态	USINT	0=不存在 1=正在配置 3=已建立 4=过时
2	读	实例类型	USINT	0=I/O 报文
3	读	传输类触发	BYTE	如果分配选择=轮询 或 如果分配选择=!轮询&& 应答抑制: 则为 0x82(服务器类 2)

				如果分配选择=! 轮询&& 应答抑制: 则为 0x80(服务器类 0)
4	读	生产连接 ID	UINT	01111xxxxx(xxxxx=address)
5	读	消费连接 ID	UINT	10xxxxx101(xxxxx=address)
6	读	初始通信特征	BYTE	0x21 2=主站 POLL 响应 1=主站 POLL 消费
7	读	生产连接数据大小	UINT	256
8	读	消费连接数据大小	UINT	256
9	读/设置	期望信息包速率	UINT	最小时间单位为 10ms
12	读/设置	看门狗定时器动作	USINT	0=转变为超时 1=自动删除 2=自动复位
13	读	生产连接路径长度	UINT	7
14	读/设置	生产连接路径	EPATH	[63hex][hex string]
15	读	消费连接路径长度	UINT	7
16	读/设置	消费连接路径	EPATH	[63hex][hex string]

连接对象实例 3 属性(位选通连接)

属性 ID	访问规则	名称	数据类型	属性值
1	读	状态	USINT	0=不存在 1=正在配置 3=已建立 4=过时
2	读	实例类型	USINT	0=I/O 报文
3	读	传输类触发	BYTE	0x83
4	读	生产连接 ID	UINT	01111xxxxx(xxxxx=address)
5	读	消费连接 ID	UINT	10xxxxx101(xxxxx=address)
6	读	初始通信特征	BYTE	0x02
7	读	生产连接数据大小	UINT	1...8
8	读	消费连接数据大小	UINT	0...8
9	读/设置	期望信息包速率	UINT	最小时间单位为 10ms
12	读/设置	看门狗定时器动作	USINT	0=转变为过时 1=自动删除 2=自动复位
13	读	生产连接路径长度	UINT	3
14	读/设置	生产连接路径	EPATH	[63hex][hex string]
15	读	消费连接路径长度	UINT	3
16	读/设置	消费连接路径	EPATH	[63hex][hex string]

连接对象实例 4 属性 (I/O 状态改变/循环连接)

属性 ID	访问规则	名称	数据类型	属性值
1	读	状态	USINT	0=不存在 1=正在配置 3=已建立 4=过时
2	读	实例类型	USINT	0=I/O 报文
3	读	传输类触发	BYTE	循环: 0x03 循环应答抑制: 0x00

				状态改变: 0x13 状态改变应答抑制: 0x01
4	读	生产连接 ID	UINT	01101xxxxx(xxxxx=address)
5	读	消费连接 ID	UINT	10xxxxx010(xxxxx=address)
6	读	初始通信特征	BYTE	0x01
7	读	生产连接数据大小	UINT	0...8
8	读	消费连接数据大小	UINT	0...8
9	读/设置	期望信息包速率	UINT	最小时间单位为 10ms
12	读/设置	看门狗定时器动作	USINT	0=转变为过时 1=自动删除 3=延迟删除
13	读	生产连接路径长度	UINT	7
14	读/设置	生产连接路径	EPATH	[63hex][hex string]
15	读	消费连接路径长度	UINT	5
16	读/设置	消费连接路径	EPATH	[63hex][hex string]
17	读/设置	生产禁止时间	UINT	最小时间单位=10ms

连接对象支持的公共服务:

服务代码	类	实例	服务名称
0x0E	不支持	支持	读单个属性(get_attribute_single)
0x10	不支持	支持	设置单个属性(set_attribute_single)

应用对象-类代码 0xAB

实例属性:

属性 ID	访问规则	名称	数据类型	属性值
1	读	用户卡序列号	UDINT	
2	读	用户卡厂商 ID	UINT	
3	读	DeviceNet 软件版本	UINT	
4	读	IN I/O 长度	UINT	
5	读	OUT I/O 长度	UINT	
6	读/设置	Produced IO assembly instance	USINT	0xA4
7	读/设置	Consumed IO assembly instance	USINT	0xA3

应用对象支持的公共服务:

服务代码	类	实例	服务名称
0x0E	不支持	支持	读单个属性(get_attribute_single)
0x10	不支持	支持	设置单个属性(set_attribute_single)

参数对象-类代码 0xAA

类属性不支持

实例属性:

属性 ID	访问规则	名称	数据类型	属性值
1	读/设置	参数 1	UINT	
2	读/设置	参数 2	UINT	
3	读/设置	参数 3	UINT	
4	读/设置	参数 4	UINT	

参数对象支持的公共服务:

服务代码	类	实例	服务名称
0x0E	不支持	支持	读单个属性(get_attribute_single)
0x10	不支持	支持	设置单个属性(set_attribute_single)

应答处理对象- 类代码 0X2B

类属性不支持

应答处理对象支持下列实例属性：

属性 ID	访问规则	名称	数据类型	属性值
1	读/设置	应答定时器	UINT	16
2	读/设置	重发次数限制	SINT	1
3	读/设置	状态改变生产连接实例	UINT	4

应答处理对象支持的公共服务：

服务代码	类	实例	服务名称
0x0E	不支持	支持	读单个属性值 (get_attribute_single)
0x10	不支持	支持	设置单个属性值 (set_attribute_single)

附录 B: EDS 文件

\$ DeviceNet Electronic Data Sheet
\$ Electronic Data Sheet
\$ Copyright (C) 1998 S-S Technologies Inc.

[File]

DescText = "Sibotech_Dnet_Adaptor";
CreateDate = 12-29-03;
CreateTime = 08:57:44;
ModDate = 08-17-06;
ModTime = 10:33:05;
Revision = 1.1;

[Device]

VendCode = 1016;
VendName = "Shanghai Sibotech Automation Co. Ltd.";
ProdType = 0;
ProdTypeStr = "Generic Device";
ProdCode = 10;
MajRev = 1;
MinRev = 1;
ProdName = "DNet_Adaptor_SiboTech";
Catalog = "Generic DeviceNet Device Powered by Sibotech";

[IO_Info]

Default = 0x0001;

PollInfo = 0x0001, 1, 1;

COSInfo = 0x0005, 1, 1;

Input1=
32, 0, 0x000F,
"Dflt Input Assy",
6, "20 04 24 B4 30 03",
"Default Input Assmblly";

Output1=
32, 0, 0x000F,
"Dflt Output Assy",
6, "20 04 24 B5 30 03",

"Default Output Assembly";

[ParamClass]

MaxInst = 11;
Descriptor = 0x0003;
CfgAssembly = 0;

[Params]

Param1=
0,
6, "20 AA 24 01 30 01",
0x0020, 2, 2,
"data1",
"",
"parameter from application",
0, 65535, 0,
1, 1, 1, 0,
0, 0, 0, 0,
0;

Param2=
0,
6, "20 AA 24 01 30 02",
0x0020, 2, 2,
"data2",
"",
"parameter from application",
0, 65535, 0,
1, 1, 1, 0,
0, 0, 0, 0,
0;

Param3=
0,
6, "20 AA 24 01 30 03",
0x0020, 2, 2,
"data3",
"",
"parameter from application",
0, 65535, 0,
1, 1, 1, 0,
0, 0, 0, 0,
0;

```
Param4=  
0,  
6, "20 AA 24 01 30 04",  
0x0020, 2, 2,  
"data4",  
"",  
"parameter from application",  
0, 65535, 0,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

```
Param5=  
0,  
6, "20 AB 24 01 30 01",  
0x0010, 6, 4,  
"application card serial",  
"",  
"information for application",  
0, 65536, 0,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

```
Param6=  
0,  
6, "20 AB 24 01 30 02",  
0x0010, 2, 2,  
"application card id",  
"",  
"information for application card",  
0, 65535, 0,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

```
Param7=  
0,  
6, "20 AB 24 01 30 03",  
0x0010, 2, 2,  
"software version",  
"",  
"information for software version",
```

```
0, 65535, 0,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

```
Param8=  
0,  
6, "20 AB 24 01 30 04",  
0x0010, 2, 2,  
"input length",  
"",  
"information for io",  
0, 65535, 0,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

```
Param9=  
0,  
6, "20 AB 24 01 30 05",  
0x0010, 2, 2,  
"output length",  
"",  
"information for io",  
0, 65535, 0,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

```
Param10=  
0,  
6, "20 AB 24 01 30 06",  
0x0000, 8, 1,  
"active produced assembly",  
"",  
"active assembly",  
0, 255, 0,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

```
Param11=  
0,  
6, "20 AB 24 01 30 07",
```

```
0x0000, 8, 1,  
"active consumed assembly",  
"",  
"active assembly",  
0, 255, 0,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

[EnumPar]

[Groups]

上海泗博自动化技术有限公司 提供

www.sibotech.cn

Tel: 021-5102 8348