

模拟量输入CAN总线智能变送器

8路模拟量输入信号转 CAN 总线及 RS485 总线模块：ISO CAN-I

产品特点

- 热电偶/0-20mA/0-10V等模拟量输入转CAN总线输出
- 支持CAN2.0A帧格式，支持CAN OPEN通讯协议
- CANBUS采用隔离及保护电路，隔离电压1500VDC
- 支持模拟量信号转RS485，支持modbus RTU协议
- 模拟信号采样精度优于 0.02% (@25 °C)
- 8路模拟量共地输入，输入信号类型及幅度可订制
- 宽电源电压供电范围：9-30VDC
- 外形美观，可靠性高，使用方便，DIN 35标准安装
- 工业级温度范围: - 40 ~ +85 °C

典型应用

- 温度信号测量、监视和控制
- 工业现场信号隔离及长线传输
- 热电偶信号隔离转换及长线传输
- 模拟信号A/D转换、调整及远程变送
- 小信号的测量及工业现场数据的获取与记录
- 工业自动化设备、机器人CANBUS系统控制
- 数控机床、DCS、智能充电站CAN总线数据采集
- 智能家居CANBUS总线远程数据采集
- 医疗、汽车电控装置及工控智能化产品开发

第一章 概述

SunYuan ISO CAN-I 系列标准工业级DIN35导轨安装CAN总线模拟量输入智能变送器是一种将隔离模拟量输入信号转换成隔离CANBUS信号及非隔离RS485信号的转换模块，产品内部包括干扰抑制滤波电路，DC-DC隔离电路，CANBUS信号隔离转换电路和输入模拟量校准电路等，可实现热电偶信号/模拟0-20mA/0-5V/0-10V标准信号采集。

ISO CAN-I 产品按工业标准设计制造，辅助电源宽电压供电范围：9-30VDC，CANBUS采用隔离及保护电路，隔离电压1500VDC，抗干扰能力强，可靠性高。工作温度范围- 40°C ~ +85°C。模拟输入信号与电源通信电路隔离，隔离电压3000VDC。

ISO CAN-I 是基于单片机的智能监测和数据采集系统，所有的用户设定的偏移值及倍数均储存在非易失性存储器里，产品支持CAN2.0A帧格式，支持CAN OPEN通讯协议，节点ID波特率可由用户设置，能与其他厂家的设备挂在同一CAN总线上，8路模拟量输入信号共地，输入范围及类型可由用户指定，模块同时支持RS485/232现场总线数据采集，用户可通过标准modbus RTU协议读取模拟量输入数据。广泛用于汽车电子、医疗设备、电力仪表、数控机床、机器人、电池充放电控制系统、智能家居及工控智能化行业。

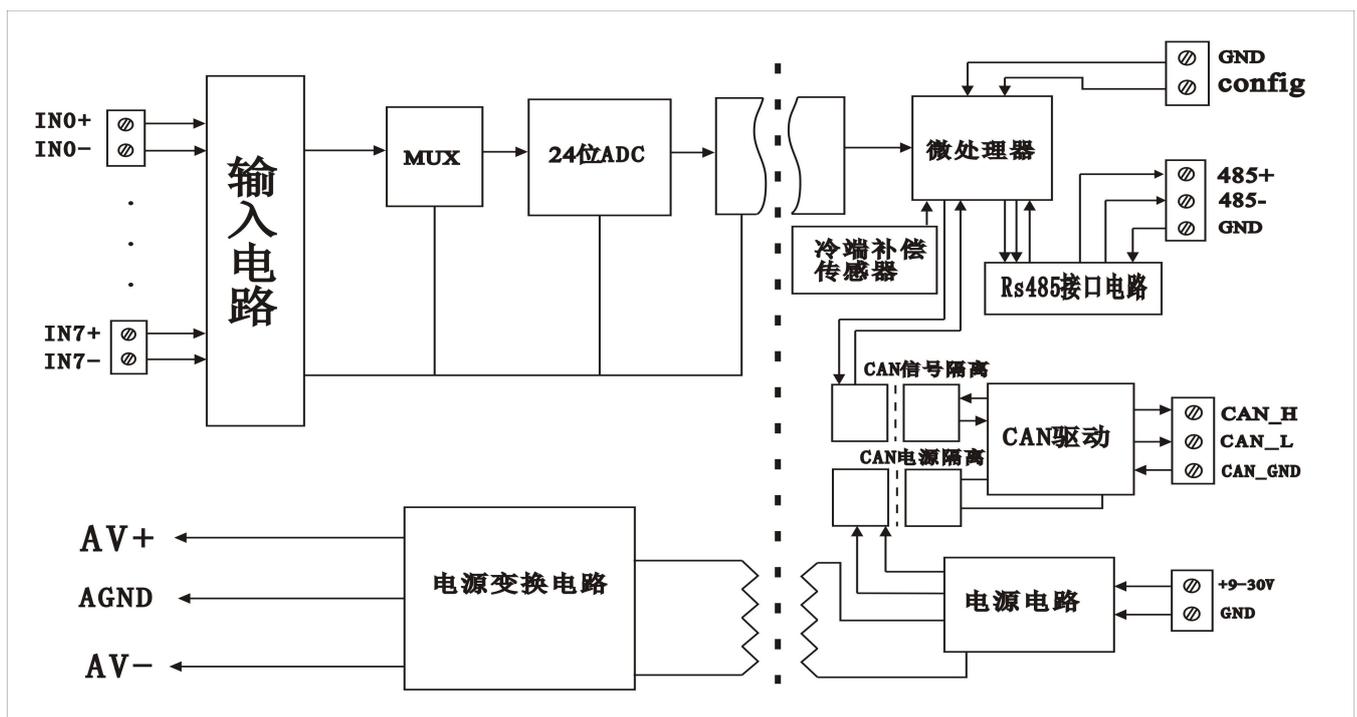


图 1 ISO-CAN-I原理框图

功能简介: ISO CAN-I 可采集8路热电偶/0-24mA/0-±5V/0-±10V模拟输入信号, 支持CAN总线标准CAN open协议

1、模拟信号输入采用 24 位 ADC 转换芯片, 产品出厂前所有输入信号偏移及量程已全部校准

通讯接口: 1 路隔离 CAN 总线接口, 支持 CAN2.0A 帧格式, 1 路 RS485 总线接口

通讯协议: 支持 CAN open 通讯协议, 支持 modbus RTU 协议, 能实现与多种品牌的 PLC、HMI、计算机监控系统进行网络通讯

2、抗干扰

内部包括浪涌保护电路、干扰抑制滤波电路, DC-DC 隔离电路, CANBUS 信号隔离转换电路

产品选型

ISO CAN - I□

输入热电偶信号

或模拟量电压电流信号值



T: J/K/B/E/R/S/T 热电偶

I1: 4-20mA

I2: 0-20mA

I4: 0-5V

I5: 0-10V

I6: 1-5V

I7: 0-±5V

I8: 用户自定义

I9: 0-±20mA

I10: 0-±10V

选型举例 1: 型号: **ISO CAN-I1** 表示 4-20mA 信号输入

选型举例 2: 型号: **ISO CAN-I4** 表示 0-5V 信号输入



图 2 ISO CAN-I 产品图片

通用参数

(typical @ +25°C, Vs为24VDC)

输入类型: 热电偶信号输入/模拟量电流信号输入/模拟量电压信号输入

精度: 0.02%@25°C

输入失调: 热电偶信号输入 ±0.5 uV/°C

电流输入 ±0.5 uA/°C

电压输入 ±0.25mV/°C

温度漂移: ±20 ppm/°C (±30 ppm/°C, 最大)

输入阻抗: 热电偶输入 3G

电流输入 50R (4-20mA/0-24mA 电流输入)

电压输入 920K (0-10V)

通讯:

CAN总线: 支持标准CAN open协议, 可插拔接口

节点地址及波特率由用户设定

RS485总线: 支持标准modbus RTU协议, 节点地址与can节点地址相同, 地址及波特率可由用户设定

工作电源: +9 — 30 VDC 宽供电范围, 内部有防反接和过压保护电路

功率消耗: 小于1W

工作温度: -40 — +85°C

工作湿度: 10 — 90% (无凝露)

存储温度: -55 — +85°C

存储湿度: 10 — 95% (无凝露)

隔离耐压: 模块与 CAN 总线: 1.5KVDC/1 分钟, 漏电流小于 1mA

模拟量输入与电源及 RS485: 3KVDC/1 分钟, 漏电流小于 1mA

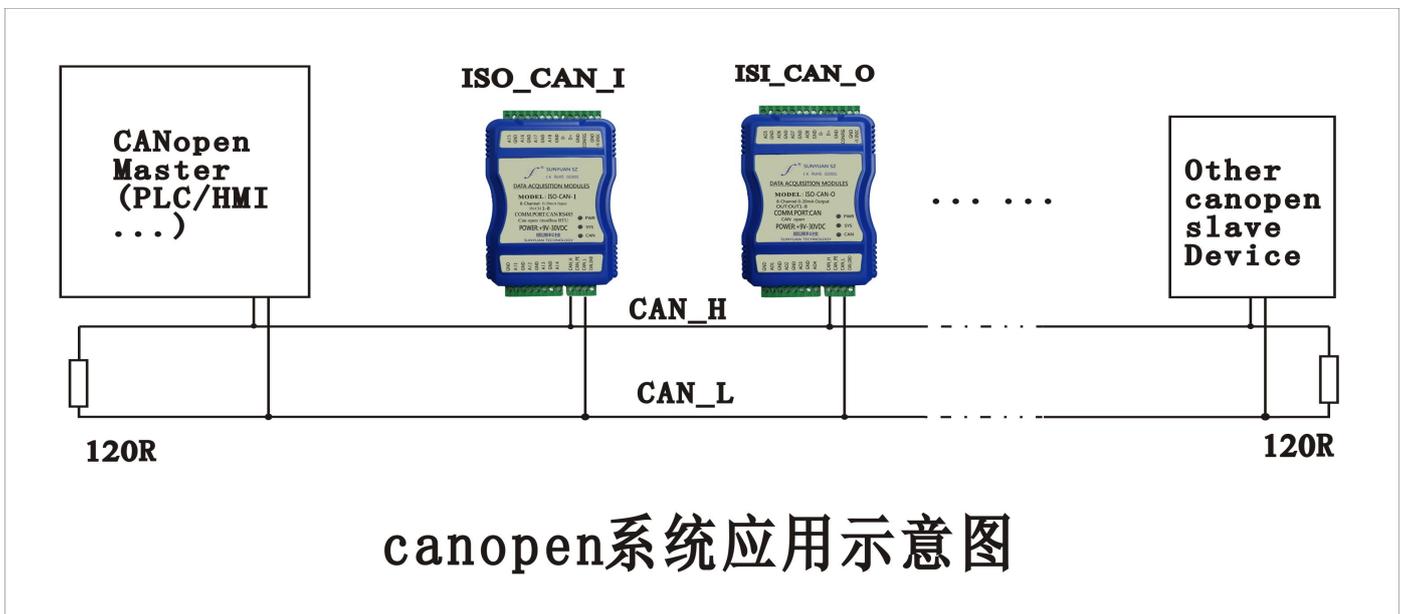
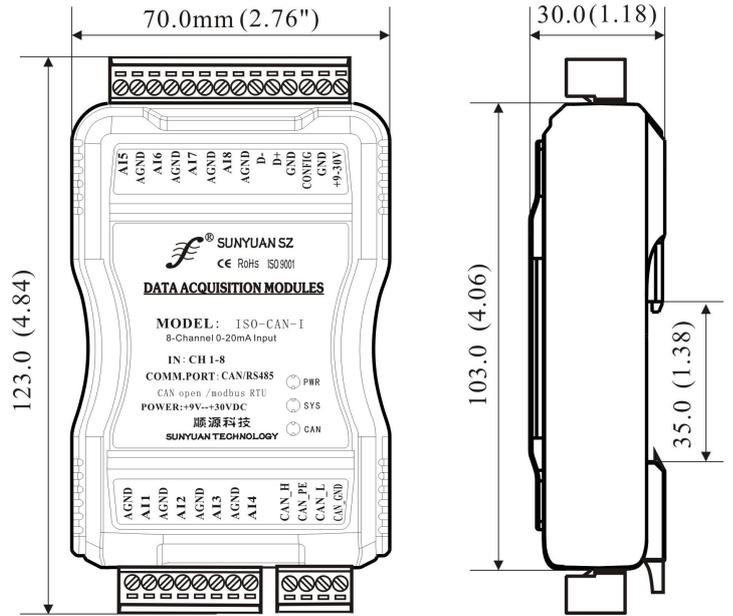
耐冲击电压: 1.5KVAC, 1.2/50us(峰值)

外形尺寸: 123 x 70 x 30 mm

重量: 约83克

外形尺寸及引脚定义

引脚	名称	描述
1	+9-30VDC	输入电源正极
2	GND	输入电源地
3	CONF	配置管脚
4	GND	输入电源地
5	D+	RS485+
6	D-	RS485-
7	AGND	输入地
8	AI8	输入 8
9	AGND	输入地
10	AI7	输入 7
11	AGND	输入地
12	AI6	输入 6
13	AGND	输入地
14	AI5	输入 5
15	AGND	输入地
16	AI4	输入 4
17	AGND	输入地
18	AI3	输入 3
19	AGND	输入地
20	AI2	输入 2
21	AGND	输入地
22	AI1	输入 1
23	CAN_H	CAN 总线 H 信号线
24	CAN_PE	CAN 总线保护接地
25	CAN_L	CAN 总线 L 信号线
26	CAN_GND	CAN 总线信号地



产品使用

电源连接

ISO CAN-I 支持工业现场常见的+9~30V DC 直流电源，为保证模块可靠工作建议使用 12V 或 24V 直流稳压电源。

系统状态指示灯

模块具有 1 个 SYS 指示灯、1 个 PWR 指示灯和 1 个 CAN 指示灯来指示设备的运行状态，这 3 种指示灯的具体指示功能及状态如下表所示。

指示灯	状态	指示状态	
PWR	不亮	电源供电故障	
	绿色常亮	设备供电正常	
SYS	绿色快闪	CAN open 预操作状态	
	绿色慢闪	CAN open 启动状态	
CAN	绿色慢闪	数据量小、速率慢	
	绿色快闪	数据量大、速率快	
	红色常亮	总线故障	

初始化ISO-CAN-I模块

在使用模块之前，必须为 ISO-CAN-I 模块分配一个独一无二的 ID 地址代码，地址代码为 16 进制数，取值在 0x00 和 0x7F 之间。所有全新的 ISO-CAN-I 模块使用 RS485 总线接口初始设置，如下所示：

地址代码为 01H

波特率 9600 bps

禁止校验和

ASCII 字符通讯协议

由于新模块的地址代码都是一样的，他们的地址将会和其他模块矛盾，所以当你组建系统时，你必须重新配置每一个模拟输出模块的 ID 地址。可以在接好 ISO-CAN-I 模块电源线和 RS-485 通讯线后，通过配置命令来修改 ISO-CAN_I 模块的 ID 地址及波特率。而在修改 ID 地址及波特率之前，必须让模块先进入配置状态，否则无法修改。

让模块进入配置状态的方法

ISO-CAN-I 模块都有一个特殊的标为 CONFIG 的管脚。将 CONFIG 管脚短路接到地线(GND 管脚)后，再接通电源，此时模块进入配置状态。在这个状态时，模块支持 ASCII 字符通讯协议,模块的配置如下：

(1) 串口：

地址代码为 00H

波特率 9600 bps

禁止校验和

ASCII 字符通讯协议

这时，可以通过配置命令来修改 ISO-CAN-I 模块的配置参数。在不确定某个模块的具体配置时，也可以通过安装配置跳线，使模块进入配置状态，再对模块进行重新配置。

第二章 ASCII 字符通讯协议

简介

当控制器以 ASCII 字符通讯协议进行通讯时，一个信息中的每 8Bit 作为 2 个 ASCII 字符传输（如传送数字 34，将分别传送 3 和 4 对应的 ASCII 码 0x33 和 0x34），这种模式的主要优点是无限定命令字符间时间间隔。

字符协议命令集

命令由一系列字符组成，如首码、地址 ID，变量、可选校验和字节和一个用以显示命令结束符 (cr)。SYAD08T 模块不支持广播地址，所以主机一次只控制一个 SYAD08T 模块。

命令格式：(Leading Code) (Addr) (Command) [data] [checksum] (cr)

(Leading code) 命令开始识别符。所有命令都需要一个命令开始识别符，如 %, \$, #, @, ... 等。

1- 字符

(Addr) 模块的地址代码，如果下面没有指定，取值范围从 00~FF（十六进制）。

2- 字符

(Command) 命令代码或变量值。

1- 字符

[data] 命令参数。

可变长度

[checksum] 校验和，为可选参数，只有在启用校验和时，才需要此选项。 2- 字符

(cr) 命令结束识别符，(cr)作为回车结束符，它的值为0x0D。 1- 字符

校验和用来检查主机与模块通信是否正确。当启用校验和时，命令与应答都必须附加校验和 [Checksum] 参数。它占2个字符。校验和字符放置在命令或响应字符之后，回车符之前。

计算方法：求之前所发所有字符的ASCII码数值之和，然后与十六进制数0xFF相与。所得结果为两位十六进制数。

应用举例：禁止校验和(checksum)

用户命令 \$022(cr)

模块应答 !02000600 (cr)

启用校验和(checksum)

用户命令 \$022B8 (cr)

模块应答 !02000640AD (cr)

'\$' = 0x24 '0' = 0x30 '2' = 0x32

B8=(0x24+0x30+0x32+0x32) AND 0xFF

'!' = 0x21 '0' = 0x30 '2' = 0x32 '4' = 0x34 '6' = 0x36

AD=(0x21+0x30+0x32+0x30+0x30+0x30+0x36+0x34+0x30) AND 0xFF

常用模拟输入模块命令

- 1、读模拟输入模块数据
- 2、读输入模块通道 N 数据
- 3、CJC 状态命令
- 4、CJC 偏移校准
- 5、配置模块
- 6、读配置状态
- 7、偏移校准
- 8、满刻度校准
- 9、启用或禁止通道命令
- 10、读通道状态命令
- 11、热电偶断线检测命令
- 12、读模块名称

命令的应答

应答信息取决于各种各样的命令。应答也由几个字符组成，包括首代码，变量和结束标识符。应答信号的首代码有两种，'!' 或 '>' 表示有效的命令而 '?' 则代表无效。通过检查应答信息，可以监测命令是否有效

注意：1、需确保地址正确，如地址错误，目标模块不做响应。

2、命令须为大写字母。

1、读模拟输入模块数据命令

功能：以当前配置的数据格式，从模拟输入模块中读回模拟输入数据。

命令语法：#AA(cr)

参数说明：# 分界符。

AA 模块地址，取值范围 00H~FFH(十六进制)。

(cr) 结束符，上位机回车键 (ODH)。

响应语法：>(data)(cr) 命令有效。

?AA(cr) 命令无效或非法操作。

参数说明：> 分界符。

(data) 代表读回的数据。数据格式为 16 进制补码，根据偏移量及倍数计算得到，与 CAN 端数据一致。

(cr) 结束符，上位机回车键 (ODH)。

其他说明：如地址错误或通讯错误，目标模块不做响应。

输入为热电偶时，如果热电偶断线，则输出为最大值（满刻度值）。可以发送断线检测命令\$AAB(cr)检查是否断线（参看命令集第 11 条）。

应用举例： 用户命令 #23(cr)

模块应答 >[0]=+1372.0427

[1]=+1372.0427

[2]=+1372.0427

[3]=+1372.0427

[4]=+1372.0427

[5]=+1372.0427

[6]=+1372.0427

[7]=+1372.0427

说明：在地址 23H 模块上输入是+1372.0472 度

2、读模块通道 N 数据命令

功能：以当前配置的数据格式，从模块中读回通道 N 的模拟输入值。

命令语法：**#AA(cr)**

参数说明：**#** 分界符。

AA 模块地址，取值范围 00H~FFH。

N 通道代号 0或1

(cr) 结束符，上位机回车键（ODH）。

响应语法：**>(data)(cr)** 命令有效。

?AA(cr) 命令无效或非法操作或通道被关闭。

参数说明：**>** 分界符。

(data) 代表读回的通道 N 的数据。数据格式为 16 进制补码，根据偏移量及倍数计算得到，与 CAN 端数据一致。

(cr) 结束符，上位机回车键（ODH）。

其他说明：如地址错误或通讯错误，目标模块不做响应。

应用举例：用户命令 **#230(cr)**

模块应答 **>[0]=+1372.0427(cr)**

说明：在地址 23H 模块上 通道 0 的输入是热电偶温度值。

3、CJC 状态命令

功能：读 CJC 传感器的数值，返回数值单位是℃。

命令语法：**\$AA3(cr)**

参数说明：**\$** 分界符。

AA 模块地址，取值范围 00H~FFH。

3 指示 CJC 状态命令。

(cr) 结束符，上位机回车键（ODH）。

响应语法：**>(data)(cr)** 命令有效。

?AA(cr) 命令无效或非法操作，或输入没有配置成热电偶信号输入。

参数说明：**>** 分界符，表示命令有效。

? 分界符，表示命令无效。

(data) 代表 CJC 传感器的数值，单位是℃。格式组成：一位“+” or “-” 符号，接着是 5 个十进制数字位和一个小数点位。小数点是固定的，分辨率是 0.1℃。

AA 代表输入模块地址

(cr) 结束符，上位机回车键（ODH）。

其他说明：如地址错误或通讯错误，目标模块不做响应。

应用举例：用户命令 **\$233(cr)**

模块应答 **> [0]=+30.7 [1]=+32.2 (cr)**

说明：在地址 23H 模块上 CJC 传感器的温度是 30.7 与 32.2℃。

4、CJC 偏移校准命令

说明：CJC（冷端补偿）偏移校准可用于校对 CJC 偏移误差。

命令语法：**\$AA9(Counts)(cr)**

参数说明：**\$** 分界符。

AA 模块地址，取值范围 00H~FFH。

(Counts) CJC 偏移值的大小，出厂默认值为 0000。命令为 4 字符（十六进制），带有+ 或 - 的符号，范围为 0000 到 FFFF。每一个计数大约等于 0.125℃。

(cr) 结束符，上位机回车键（ODH）。

响应语法：**!AA(cr)** 命令有效。

?AA(cr) 命令无效或非法操作，或输入没有配置成热电偶信号输入。

应用举例： 用户命令 %239+0028(cr)
模块应答 !23(cr)

说明：对地址 23H 模块 CJC 偏移补偿，偏移温度为+0028= 2 * 16 + 8 = 40, 40 * 0.125℃ = 5℃

5、配置模拟输入模块命令

功能：设置一个模块的地址，输入范围，波特率，数据格式，校验和状态。配置信息储存在非易失性存储器 EEPROM 里。

命令语法：%AANNTCCFF(cr)

参数说明：% 分界符。

AA 模块地址，取值范围 00H~7FH。

NN 代表模块新的 16 进制地址，数值 NN 的范围从 00 到 7F。

TT 用 16 进制代表类型编码(只读), 具体如下所示：

- 0x00: 0-20ma/4-20ma
- 0x01: 0-10ma/0-+-10ma
- 0x02: 0-+-1ma
- 0x08: 自定义电流
- 0x10: 0-5V/0-+-5V
- 0x11: 0-10V/0-+-10V
- 0x12: 0-2.5V
- 0x13: 0-+-1V
- 0x14: 0-+-500mv
- 0x15: 0-+-100mv
- 0x16: 0-75mv
- 0x18: 自定义电压
- 0x1B:0-+-50mv
- 0x1D:0-+-15mv
- 0x1E:0-24v
- 0x1F:0-30mv
- 0x2E:J 50mV 0—760 度
- 0x2F:K 45mV 0—1000 度
- 0x20:T 25mV —100—+400
- 0x21:E 78mV 0—1000
- 0x22:R 22mv 500—1750
- 0x23:S 20mV 500—1750
- 0x24:B 15mV 500—1800

CC 用 16 进制代表波特率编码，详见表 2。

波特率代码	串口波特率
x1	300bps
x2	600bps
x3	1200bps
x4	2400bps
x5	4800bps
x6	9600bps
x7	19200bps
x8	38400bps
	CAN 波特率
0x	1000Kbps
1x	800Kbps
2x	500Kbps
3x	250Kbps
4x	125Kbps

5x	100Kbps
6x	50Kbps
7x	20Kbps
8x	10Kbps

表 2 波特率代码(低 4 位表示串口波特率, 高 4 位表示 CAN 波特率)

FF 用 16 进制的 8 位代表数据格式, 校验和。注意从 bits2 到 bits5 不用必须设置为零。

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit2	Bit 1	Bit 0
------	-------	-------	-------	-------	------	-------	-------

表 3 数据格式, 校验和代码

Bit7: 保留位, 必须设置为零

Bit6: 校验和状态, 必须为 0: 禁止;

Bit5-bit2: 不用, 一般设置为零。

Bit1-bit0: 数据格式位。 必须为 0: 工程单位(Engineering Units)

(cr) 结束符, 上位机回车键 (ODH)。

响应语法: !AA(cr) 命令有效。

?AA(cr) 命令无效或非法操作, 或在改变波特率或校验和前, 没有安装配置跳线。

参数说明: ! 分界符, 表示命令有效。

? 分界符, 表示命令无效。

AA 代表模块地址。

(cr) 结束符, 上位机回车键 (ODH)。

其他说明: 注 3: 如果用户要重新配置模块的地址、波特率, 则必须安装配置跳线, 使模块进入配置状态, 此时模块地址为 00H, 即 AA=00H, NN 等于新的地址。否则将返回错误信号。

注 4: 执行配置命令后, 从机将新的地址作为响应数据返回给主机。

如地址错误或通讯故障, 目标模块不做响应。

应用举例: 用户命令 %0011000600(cr)

模块应答 !11(cr)

说明: % 分界符。

00 表示你想配置的模拟输出模块原始地址为 00H。

11 表示新的模块 16 进制地址为 11H。

00 类型代码(只读, 不可设置)。

06 表示波特率 50kbps。

00 表示数据格式为工程单位, 禁止校验和。

6、读配置状态命令

功能: 读取相应模块的配置信息。

命令语法: \$AA2(cr)

参数说明: \$ 分界符。

AA 模块地址, 取值范围 00H~7FH。

2 表示读配置状态命令

(cr) 结束符, 上位机回车键 (ODH)。

响应语法: !AATTCCFF(cr) 命令有效。

?AA(cr) 命令无效或非法操作。

参数说明: ! 分界符。

AA 代表输入模块地址

TT 代表类型编码。见表 4

CC 代表波特率编码。见表 3

FF 见表 5

(cr) 结束符, 上位机回车键 (ODH)。

其他说明: 如地址错误或通讯错误, 目标模块不做响应。

应用举例: 用户命令 \$302(cr)

模块应答 !302F0600(cr)

说明：！ 分界符。
30 表示模拟输入模块地址为30H。
2F 表示输入为K型电偶，测量范围0℃~1000℃。
06 表示波特率9600 bps。
00 表示数据格式为工程单位，禁止校验和。

7、偏移校准命令

功能：校准一个模块的通道N的零点偏移。

命令语法：**\$AA1N(cr)**

参数说明：\$ 分界符。
AA 模块地址，取值范围00H~FFH。
1 表示偏移校准命令
N 通道代号0~7
(cr) 结束符，上位机回车键(ODH)。

响应语法：!AA(cr) 命令有效。
?AA(cr) 命令无效或非法操作。

参数说明：! 分界符，表示命令有效。
? 分界符，表示命令无效。
AA 代表输入模块地址
(cr) 结束符，上位机回车键(ODH)。

其他说明：产品出厂时已经校准，用户无需校准即可直接使用。

当对一个模拟输入模块校准时，先校准偏移，再校准增益。

在校准时，模拟输入模块需在要校准的通道上加上合适的输入信号。不同的输入范围需要不同的输入电压或电流。具体校准方法请看校准模块章节。

如地址错误或通讯错误，目标模块不做响应。

应用举例： 用户命令 **\$2310(cr)**
 模块应答 **!23 (cr)**

说明：对地址23H模块的通道0进行偏移校准。

8、增益校准命令

功能：校准一个输入模块通道N的增益。

命令语法：**\$AAON(cr)**

参数说明：\$ 分界符。
AA 模块地址，取值范围00H~FFH。
0 表示增益校准命令
N 通道代号0~7
(cr) 结束符，上位机回车键(ODH)。

响应语法：!AA(cr) 命令有效。
?AA(cr) 命令无效或非法操作。

参数说明：! 分界符，表示命令有效。
? 分界符，表示命令无效。
AA 代表输入模块地址
(cr) 结束符，上位机回车键(ODH)。

其他说明：产品出厂时已经校准，用户无需校准即可直接使用。

当对一个模拟输入模块校准时，先校准偏移，再校准增益。

在校准时，模拟输入模块需要连上合适的输入信号。不同的输入范围需要不同的输入电压，具体指标可参考下文校准模块部分的说明。

如地址错误或通讯错误，目标模块不做响应。

应用举例： 用户命令 **\$2303(cr)**
 模块应答 **!23 (cr)**

说明：对地址23H模块的通道3进行增益校准。

9、启用或禁止通道命令

命令功能：启动或禁止目标模块的数据采集通道

命令语法: **\$AA5VV(cr)**

参数说明: **\$** 分界符。

AA 目标模块地址, 取值范围 00~FF(十六进制)。

5 启动或禁止模块的数据采集通道命令关键字。

VV 两位16进制数, 第一位数的3~0位代表7~4通道
第二位数的3~0位代表3~0通道
位值为 0: 禁止通道
位值为 1: 启用通道

(cr) 结束符, 上位机回车键 (ODH)。

响应语法: **!AA(cr)** 命令有效。

?AA(cr) 命令无效或非法操作

参数说明: **!** 分界符, 表示命令有效。

? 分界符, 表示命令无效。

AA 目标模块地址。

(cr) 结束符, 上位机回车键 (ODH)。

其他说明: 如地址错误或通讯错误, 目标模块不做响应。

应用举例: 用户命令 **\$08537(cr)**

模块应答 **!08(cr)**

说明: 设置目标模块通道值为 0x37。

3即0011, 表示启用通道5和4, 禁止通道7和6。

7即0111, 表示启用通道2、1和0, 禁止通道3。

10、读通道状态命令

功能: 读一个模块的通道状态命令。

命令语法: **\$AA6(cr)**

参数说明: **\$** 分界符。

AA 模块地址, 取值范围 00H~FFH。

6 表示读通道状态命令

(cr) 结束符, 上位机回车键 (ODH)。

响应语法: **!AAVV(cr)** 命令有效。

?AA(cr) 命令无效或非法操作

参数说明: **!** 分界符, 表示命令有效。

? 分界符, 表示命令无效。

AA 代表输入模块地址。

VV 两个16进制数, 数的7~0位代表7~0通道

位值为 0: 禁止通道

位值为 1: 启用通道

(cr) 结束符, 上位机回车键 (ODH)。

其他说明: 如地址错误或通讯错误, 目标模块不做响应。

应用举例: 用户命令 **\$186(cr)**

模块应答 **!18FF(cr)**

说明: 当前通道状态值为 0xFF。

0xFF即1111和1111, 表示地址18H的模块所有通道都已经启用。

11、热电偶断线检测命令

功能: 检测热电偶是否断线。

命令语法: **\$AAB(cr)**

参数说明: **\$** 分界符。

AA 模块地址, 取值范围 00H~FFH。

B 热电偶断线检测命令。

(cr) 结束符, 上位机回车键 (ODH)。

响应语法: **!AAVV(cr)** 两个16进制数, 数的7~0位代表7~0通道

位值为 0: 通道热电偶没断线。

位值为 1: 通道热电偶断线。

?AA(cr)命令无效或非法操作，或输入没有配置成热电偶信号输入。

参数说明：! 分界符，表示命令有效。

? 分界符，表示命令无效。

AA 代表输入模块地址。

(cr) 结束符，上位机回车键 (ODH)。

其他说明：如地址错误或通讯错误，目标模块不做响应。

应用举例： 用户命令 \$06B(cr)

模块应答 !0600(cr)

说明：地址 06H 模块上热电偶没有断线。

12、读模块名称命令

功能：读取相应模块的名称。

命令语法：\$AAM(cr)

参数说明：\$ 分界符。

AA 模块地址，取值范围 00H~FFH。

M 表示读模块名称命令

(cr) 结束符，上位机回车键 (ODH)。

响应语法：!AA(ModuleName)(cr) 命令有效。

?AA(cr) 命令无效或非法操作

参数说明：! 分界符，表示命令有效。

? 分界符，表示命令无效。

AA 代表输入模块地址。

(ModuleName) 模块名称可以为 ISO_CAN_I 等等，代表你使用的模块型号

(cr) 结束符，上位机回车键 (ODH)。

其他说明：如地址错误或通讯错误，目标模块不做响应。

应用举例： 用户命令 \$08M(cr)

模块应答 !08ISO_CAN_I(cr)

说明：在地址 08H 模块为 ISO_CAN_I。

注：以上指令在配置模式下有效

正常工作状态下从模块读到的数据格式与 CAN 端的数据格式一致，说明如下：

模拟输入模块使用的数据格式为 16 进制的补码(Twos complement)

其值根据 CAN 端设置的倍数与偏移量计算得到：

例 1：模块类型为热电偶，测量得的温度为 1.123 度，CAN 端设置的倍数为 10 倍，偏移量为 0，则数据为 11

例 2：模块类型为 0—10V 电压输入，测量得到的输入电压为 9.999V，CAN 端设置的倍数为 1000 倍，偏移量为 0，则数据为 9999。

正常工作状态下模块除了支持ASCII字符协议外还支持modbus RTU协议。

产品寄存器说明

地址 4X	数据内容	属性	数据说明
40001	IN0	只读	第 0 通道测量值
40002	IN1	只读	第 1 通道测量值
40003	IN2	只读	第 2 通道测量值
40004	IN3	只读	第 3 通道测量值
40005	IN4	只读	第 4 通道测量值
40006	IN5	只读	第 5 通道测量值
40007	IN6	只读	第 6 通道测量值
40008	IN7	只读	第 7 通道测量值
30001	模块名称	只读	高位：0x01 低位：0x08
30002	通道状态	读/写	高位：0x00 低位：通道状态 (0xFF)

表 5 Modbus RTU 寄存器说明

快速上手

本章将使用 CAN 总线分析仪演示模块的使用，CAN 总线分析仪由用户自行选择购买

ISO CAN-I 模块默认上电后切换至 CAN open 模式的 05 启动状态，默认模块上电后可将采集到的模拟量数据定时发送到 can 总线。

例如：默认 ISO CAN-I 的 Node ID 为 1（以下节点号均为 1），则主站设备可接收到一条节点状态数据，帧 ID 为 0x701，标准帧，数据为 0x05，表示当前状态

序号	传输方	时间标	状态	报文描述	报文内容	CAN帧	
7	接收	13:49:		NMT Err	Node ID	Node Status	帧ID:00000701 数据帧 标准帧 DLC:01 Data:05
					0x1	Operational	

标准 CANopen 模式上电启动报文

ISO CAN-I 模块满足标准 CAN open CIA 301 协议，是标准的 CAN open 从站设备。ISO CAN-I 启动后将主动发出一帧数据给主站，帧 ID 为 0x700+Node ID。

例如：ISO CAN-I 的 Node ID 为 1（以下节点号均为 1），则主站设备可接收到一条节点状态数据，帧 ID 为 0x701，帧数据为 0x7F。

序号	传输方	时间标	状态	报文描述	报文内容	CAN帧	
8	接收	13:58:		NMT Err	Node ID	Node Status	帧ID:00000701 数据帧 标准帧 DLC:01 Data:7F
					0x1	Pre-operational	

对象字典 0x1016 设置主站心跳时间（BIT0-BIT15 表示以毫秒为单位的心跳时间，BIT16-BIT22 为主站 ID），默认为 0，默认不开启超时功能；0x1016 通过 SDO 配置后，节点开始监测主站心跳，当在设定的时间内未监测到主站心跳时，从站进入 Pre-operational 模式，并停止所有输入通道的数据采集。

NMT 状态切换

ISO CAN-I 模块接收由主站发出的操作指令，帧 ID 为 0x000，DLC 为 2，帧数据第一个字节为命令符，第二个字节为节点号（若为 00 则控制全部节点）。

例如：ISO CAN-I 的 Node ID 为 1，命令 ISO CAN-I 进入操作状态（01），则 NMT 命令帧 ID 为 0x000，帧数据为 0x01,0x01。

主站发送数据 01 01，ISO CAN-I 模块进入 05 启动状态。

序号	传输方	时间标	状态	报文描述	报文内容	CAN帧	
7	接收	14:05:		NMT Err	Node ID	Node Status	帧ID:00000701 数据帧 标准帧 DLC:01 Data:7F
					0x1	Pre-operational	
8	发送	14:05:	成功	NMT M	NMT Command	Node	帧ID:00000000 数据帧 标准帧 DLC:02 Data:01 01
					Start Remote Node	0x1	
9	接收	14:05:		NMT Err	Node ID	Node Status	帧ID:00000701 数据帧 标准帧 DLC:01 Data:05
					0x1	Operational	

PDO 命令

ISO CAN-I 模块采用 PDO（Process Data Object，过程数据对象）将采集到的模拟输入量数据发送到 CAN 总线。TPDO 用于表示输入通道的数据，共有 2 个 TPDO。其中 TPDO1，帧 ID 为 0x180+Node ID，代表前 4 个通道。TPDO2，帧 ID 为 0x280+Node ID，代表后 4 个通道。

TPDO1:

COB-ID	数据			
0x180 + NODE-ID	Byte0 Byte1	Byte2 Byte3	Byte4 Byte5	Byte6 Byte7
	通道 1	通道 2	通道 3	通道 4

TPDO2:

COB-ID	数据			
0x280 + NODE-ID	Byte0 Byte1	Byte2 Byte3	Byte4 Byte5	Byte6 Byte7
	通道 5	通道 6	通道 7	通道 8

通道 1—通道 8 数据 = 模拟量输入 * 倍数 + 偏移。

用户通过SDO命令设置倍数与偏移量时必须根据输入数据的值仔细计算，确保不超过2字节有符号数能表示的最小最大数（-32768—+32767），例如：模块为0—±10V输入电压采集模块时，可以设定倍数为1000，采集的数据不会超过表示范围，但如果倍数设定为10000则数据可能会超过表示范围。

以下为模块TPDO发送的数据实例：

2996	接收	17:03:39	PDO	<table border="1"> <thead> <tr> <th>PDO Object</th> <th>Node ID</th> <th>PDO Data</th> </tr> </thead> <tbody> <tr> <td>PDO1 (发送)</td> <td>0x1</td> <td>0x5C 05 5C 05 5C 05 5C 05</td> </tr> </tbody> </table>	PDO Object	Node ID	PDO Data	PDO1 (发送)	0x1	0x5C 05 5C 05 5C 05 5C 05	帧ID:00000181 数据帧 标准帧 DLC:08 Data:5C 05 5C 05 5C 05 5C 05
PDO Object	Node ID	PDO Data									
PDO1 (发送)	0x1	0x5C 05 5C 05 5C 05 5C 05									
2997	接收	17:03:39	PDO	<table border="1"> <thead> <tr> <th>PDO Object</th> <th>Node ID</th> <th>PDO Data</th> </tr> </thead> <tbody> <tr> <td>PDO2 (发送)</td> <td>0x1</td> <td>0x5C 05 5C 05 5C 05 5C 05</td> </tr> </tbody> </table>	PDO Object	Node ID	PDO Data	PDO2 (发送)	0x1	0x5C 05 5C 05 5C 05 5C 05	帧ID:00000281 数据帧 标准帧 DLC:08 Data:5C 05 5C 05 5C 05 5C 05
PDO Object	Node ID	PDO Data									
PDO2 (发送)	0x1	0x5C 05 5C 05 5C 05 5C 05									

SDO 命令

服务数据对象 SDO（Service Data Objects）主要用来访问节点的对象字典，可以直接对ISO CAN-I的参数进行读写配置。配置后重新上电生效。

修改系统模式

ISO CAN-I模块可通过发送 SDO 命令写参数，修改模块的使用模式。该参数位于对象字典索引 0x2400、子索引 0x00，数据类型 Unsigned8。

COB-ID	数据							
0x600 + NODE-ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
	指令符	索引		子索引	数据	0x00	0x00	0x00
	0x2F	0x00	0x24	0x00				

数据	模式
0x02	默认模式
0x03	标准 CAN open 模式

修改输入模拟量的倍数与偏移量：

ISO CAN-I 模块可通过发送 SDO 命令写参数，修改模块每一路输出的倍数与偏移量。该参数位于对象字典索引 0x2401 至 2408、子索引 0x00，数据类型 Unsigned32。

每一路的倍数与偏移量均可自由配置为不同数值。

倍数的数据类型为 Unsigned16，默认值为 0x01。

偏移量的数据类型为 signed16，默认值为 0x0。

COB-ID	数据							
0x600 + NODE-ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
	指令符	索引		子索引	倍数		偏移量	
	0x23	0x01	0x24	0x00	0x01	0x00	0x00	0x00

索引	第几路
0x2401	第一路
0x2402	第二路

0x2403	第三路
0x2404	第四路
0x2405	第五路
0x2406	第六路
0x2407	第七路
0x2408	第八路

配置操作举例

CAN open 模式配置操作举例：（以节点号为 1 举例）默认模式上电后：

1、模块配置为标准 CANopen 模式

报文：ID： 0x601 数据： 2F 00 24 00 03 00 00 00， 设置模块为标准 CANopen模式。

序号	传输方	时间标	状态	报文描述	报文内容	CAN帧
16	发送	14:35:	成功	Initiate S	Node ID:00000601 数据帧 标准帧 DLC:08 Data:2F 00 24 00 03 00 00 00	0x1
17	接收	14:35:		Initiate S	Node ID:00000581 数据帧 标准帧 DLC:08 Data:60 00 24 00 00 00 00 00	0x1

2、修改第 1 路输入的倍数与偏移量

报文：ID： 0x601 数据： 23 01 24 00 0A 00 00 00， 设置第 1 路输入的倍数为10， 即 16 进制的 0x0A； 偏移量为0， 即 16 进制的 0x00。

91	发送	17:16:00	成功	Initiate	Node ID: 0x1 Object Index: 0x2401 Object Sub-index: 0x0 Transfer T: Expedite	帧ID:00000601 数据帧 标准帧 DLC:08 Data:23 01 24 00 0A 00 00 00
92	接收	17:16:00		PDO	PDO Object: PDO1 (发送) Node ID: 0x1 PDO Data: 0x5C 05 5C 05 5C 05 5C 05	帧ID:00000181 数据帧 标准帧 DLC:08 Data:5C 05 5C 05 5C 05 5C 05
93	接收	17:16:00		Initiate	Node ID: 0x1 Object Index: 0x2401 Object Sub-index: 0x0	帧ID:00000581 数据帧 标准帧 DLC:08 Data:60 01 24 00 00 00 00 00

重新上电后， 配置生效

使用操作举例

CAN open 模式操作举例：（以节点号为 1 举例）配置完重新上电后：

1、NMT 状态切换

报文：ID： 0x000 数据： 01 01， 设置模块为 05 启动状态。

序号	传输方	时间标	状态	报文描述	报文内容	CAN帧
1	发送	14:47:	成功	NMT Module	NV Start	帧ID:00000000 数据帧 标准帧 DLC:02 Data:01 01

2、TPDO数据解析举例:

2	接收	17:33:51		PDO	PDO Object: PDO2 (发送) Node ID: 0x1 PDO Data: 0x98 35 98 35 98 35 98 35	帧ID:00000281 数据帧 标准帧 DLC:08 Data:98 35 98 35 98 35 98 35
3	接收	17:33:51		PDO	PDO Object: PDO1 (发送) Node ID: 0x1 PDO Data: 0x98 35 98 35 98 35 98 35	帧ID:00000181 数据帧 标准帧 DLC:08 Data:98 35 98 35 98 35 98 35

上述数据是K型热电偶输入模块采集的数据,因为没有接热电偶,所以实际输出数据为K偶最大温度值,且每个通道的数据相同,模块已经通过SDO设置为倍数为10,偏移量为0:实际输出值为0x3598(16进制),转换为10进制为13720,由于倍数设定为10,所以实际温度为1372.0℃。

附: ISO CAN- I 对象字典
通讯参数区

索引	子索引	名称	类型	属性	默认值	说明
0x1000	0	Device Type	UINT32	RO	0x00000001	设备类型
0x1001	0	Error Register	UINT8	RO	0	当前错误类型
0x1003	0	number of errors	UINT8	RW	0	-
	1~4	standard error field	UINT32	RO	0	历史紧急错误代码
0x1005	0	COB-ID SYNC	UINT32	RW	0x80	同步报文 COB-ID
0x1006	0	CommunicationCycle Period	UINT32	RW	0x2710	通讯循环周期
0x1007	0	Sync WindowsLength	UINT32	RW	0	同步窗口长度
0x1008	0	GCAN-4068Name	VIS_STR	RO	ISO-CAN-IN	制造商设备名称
0x1009	0	GCAN4068Hardware Version	VIS_STR	RO	V1.0	硬件版本
0x100A	0	GCAN4068Software Version	VIS_STR	RO	V1.0	软件版本
0x100C	0	Guard Time	UINT16	RW	0	保护时间
0x100D	0	Life Time Factor	UINT8	RW	3	保护时间乘数因子
0x1010	0	largest supported Sub-Index	UINT8	RO	0x1	-
	1	save all parameters	UINT32	RW	0x0	-
0x1011	0	largest supported Sub-Index	UINT8	RO	0x1	-
	1	restore all default para.	UINT32	RW	0x0	-
0x1014	0	COB-ID Emergency message	UINT32	RW	\$NODEID+ 0x80	紧急报文 COB-ID
0x1016	0	Number Of Entries	UINT8	RO	0x1	消费者心跳数量
	1	Consumer Heartbeat Time #1	UINT32	RW	0x0	消费者心跳时间及ID
0x1017	0	ProducerHeartbeat Time	UINT16	RW	0x03E8	生产者心跳时间
0x1018	0	Highest sub-index supported	UINT24	RO	4	-
	1	Vendor-ID	UINT32	RO	0x000001	
	2	Product code	UINT32	RO	0x00000002	产品代码
	3	Revision number	UINT32	RO	0x00000003	修订码
	4	Serial number	UINT32	RO	0x00000004	序列码

TPDO参数区

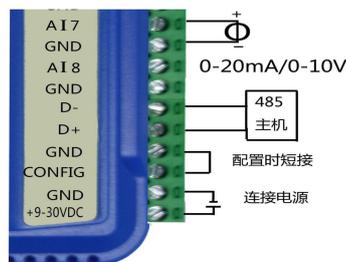
索引	子索引	名称	类型	属性	默认值	说明
0x1800	0	Highest Subindex Supported	UINT8	RO	6	TPDO1 通讯参数
	1	COB-ID	UINT32	RW	0x180+nodeID	TPDO1 的 COB-ID
	2	Transmission type	UINT8	RW	0xFF	TPDO1 的传输类型
	3	Inhibit timer	UINT16	RW	0x0064	禁止时间
	5	Eventtimer	UINT16	RW	0x0014	事件计时器
0x1801	0	Highest SubIndex Supported	UINT8	RO	6	TPDO2 通讯参数
	1	COB-ID	UINT32	RW	0x280+nodeID	TPDO2 的 COB-ID
	2	Transmission type	UINT8	RW	0xFF	TPDO2 的传输类型
	3	Inhibit timer	UINT16	RW	0x0064	禁止时间

	5	Eventtimer	UINT16	RW	0x0014	事件计时器
0x1A00	0	Number of Entries	UINT8	RO	4	TPDO1 映射参数
	1	TPDO1 mapping app object 1	UINT32	RW	0x20100110	
	2	TPDO1 mapping app object 2	UINT32	RW	0x20100210	
	3	TPDO1 mapping app object 3	UINT32	RW	0x20100310	
	4	TPDO1 mapping app object 4	UINT32	RW	0x20100410	
0x1A01	0	Number of Entries	UINT8	RO	4	TPDO2 映射参数
	1	TPDO2 mapping app object 5	UINT32	RW	0x20100510	
	2	TPDO2 mapping app object 6	UINT32	RW	0x20100610	
	3	TPDO2 mapping app object 7	UINT32	RW	0x20100710	
	4	TPDO2 mapping app object 8	UINT32	RW	0x20100810	

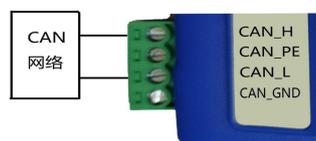
厂商自定义参数区

索引	子索引	名称	类型	属性	默认值	说明
0x2010	0	Number of Entries	UINT8	RO	8	
	1	VALUE_CH1	UINT16	RO	0	第 1 路输入值
	2	VALUE_CH2	UINT16	RO	0	第 2 路输入值
	3	VALUE_CH3	UINT16	RO	0	第 3 路输入值
	4	VALUE_CH4	UINT16	RO	0	第 4 路输入值
	5	VALUE_CH5	UINT16	RO	0	第 5 路输入值
	6	VALUE_CH6	UINT16	RO	0	第 6 路输入值
	7	VALUE_CH7	UINT16	RO	0	第 7 路输入值
	8	VALUE_CH8	UINT16	RO	0	第 8 路输入值
0x2400	0	Type	UINT8	RW	0x02	模式切换
0x2401	0	CH1	UINT32	RW	0x00000001	第一路倍数, 偏移值
0x2402	0	CH2	UINT32	RW	0x00000001	第二路倍数, 偏移值
0x2403	0	CH3	UINT32	RW	0x00000001	第三路倍数, 偏移值
0x2404	0	CH4	UINT32	RW	0x00000001	第四路倍数, 偏移值
0x2405	0	CH5	UINT32	RW	0x00000001	第五路倍数, 偏移值
0x2406	0	CH6	UINT32	RW	0x00000001	第六路倍数, 偏移值
0x2407	0	CH7	UINT32	RW	0x00000001	第七路倍数, 偏移值
0x2408	0	CH8	UINT32	RW	0x00000001	第八路倍数, 偏移值

应用接线图



电源,配置,RS485及输入接线示意图



CAN接线示意图