



	1
	1
	1
	1
	1
	2
1	RS485	2
2	RS232	2
	3
1	3
2	4
	4
1	4
2	5
3	5
	6
1	6
2	7
3	8
	8
1	8
2	Modbus	9
3	10
4	11
5	12
	14
	14



JY-DAM0816
JUYING ELECTRONIC

DAM0816数采控制器

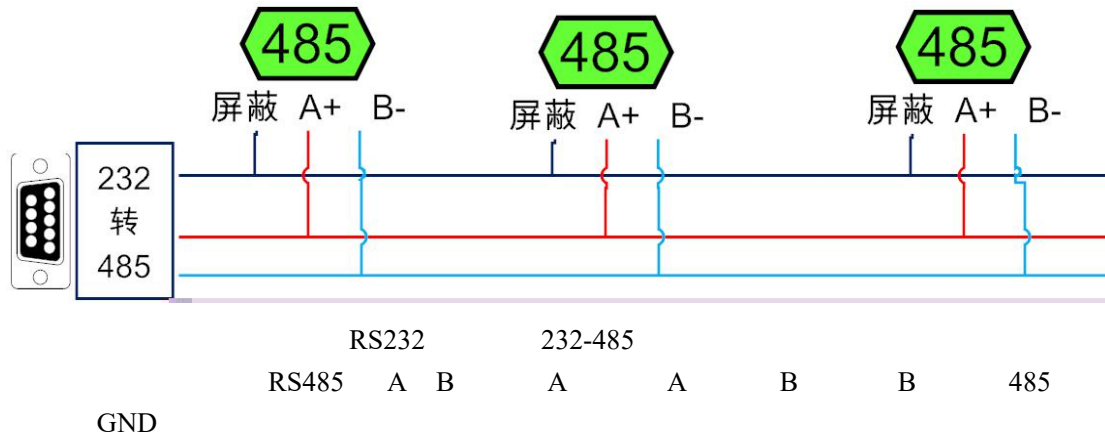
- 供电电压: DC 7-30V
- 输出: 250V 10A触点
- 通信协议: MODBUS RTU/ASCII
- 通信端口: RS232 RS485 RF 网口 WIFI USB

电源 RS485 RS232 COM1 D01 D02 D03 D04 D05 D06 D07 D08 COM

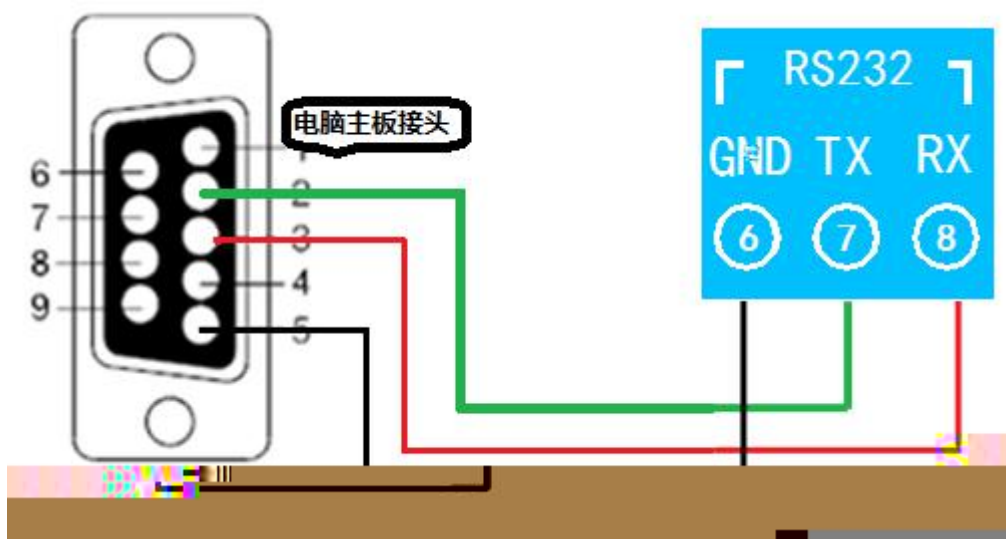
+ - PB A+ B- GND TX RX

① ② Pow Run ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱

1 R 485



2 R 232

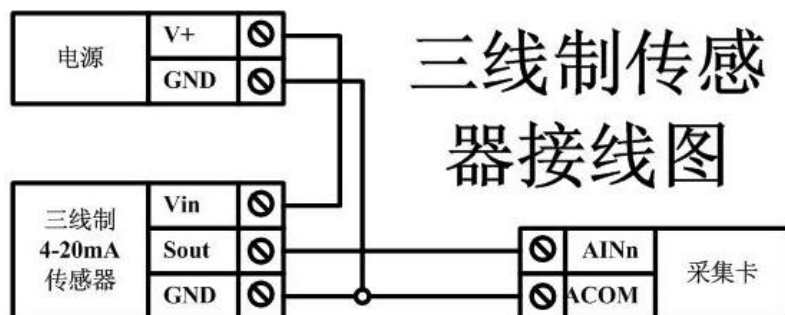
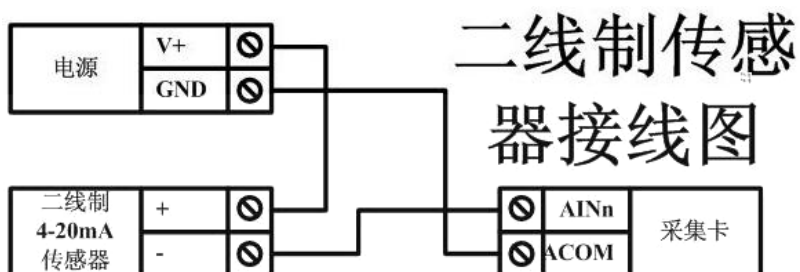


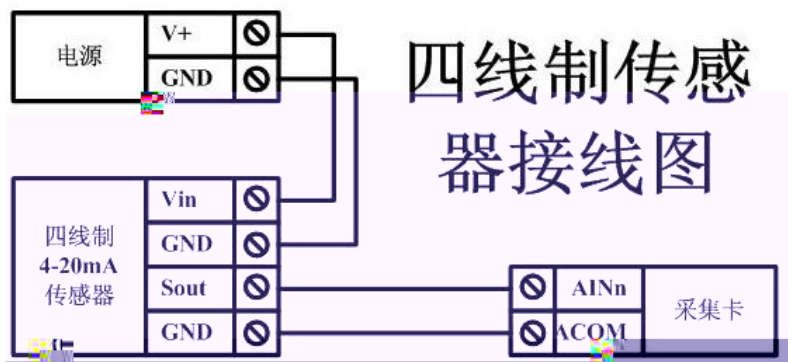
1

AI1-AI8

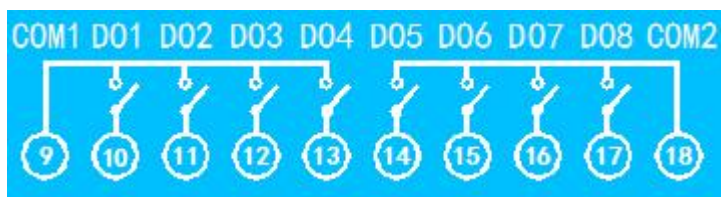
ACOM-

ACOM+





2

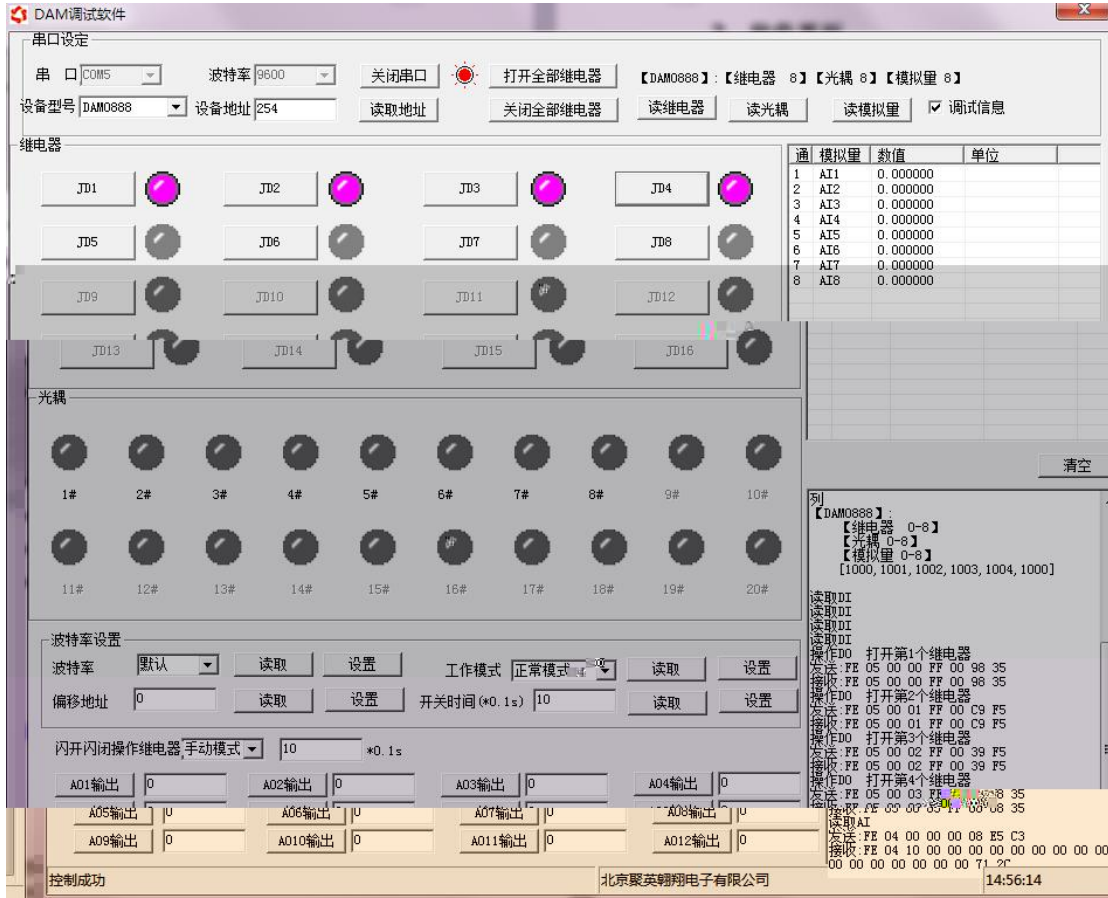


1

<http://www.juyingele.com.cn/software/software/> DAM

_____ .rar

2



串口设定

串 口 COM5 波特率 9600 关闭串口 打开全部继电器 【DAM0888】: 【继电器 8】 【光耦 8】 【模拟量 8】

设备型号 DAM0888 设备地址 254 读取地址 关闭全部继电器 读继电器 读光耦 读模拟量 调试信息

继电器

JD1 JD2 JD3 JD4 JD5 JD6 JD7 JD8 JD9 JD10 JD11 JD12 JD13 JD14 JD15 JD16

光耦

1# 2# 3# 4# 5# 6# 7# 8# 9# 10# 11# 12# 13# 14# 15# 16# 17# 18# 19# 20#

波特率设置

波特率 默认 读取 设置 工作模式 正常模式 读取 设置

偏移地址 0 读取 设置 开关时间(*0.1s) 10 读取 设置

闪开闪闭操作继电器,手动模式 10 *0.1s

A01输出 0 A02输出 0 A03输出 0 A04输出 0 A05输出 0 A06输出 0 A07输出 0 A08输出 0 A09输出 0 A10输出 0 A11输出 0 A12输出 0

控制成功 北京聚英翱翔电子有限公司 14:56:14

通	模拟量	数值	单位
1	AI1	0.000000	
2	AI2	0.000000	
3	AI3	0.000000	
4	AI4	0.000000	
5	AI5	0.000000	
6	AI6	0.000000	
7	AI7	0.000000	
8	AI8	0.000000	

```

【DAM0888】:
【继电器 0-8】
【光耦 0-8】
【模拟量 0-8】
[1000, 1001, 1002, 1003, 1004, 1000]

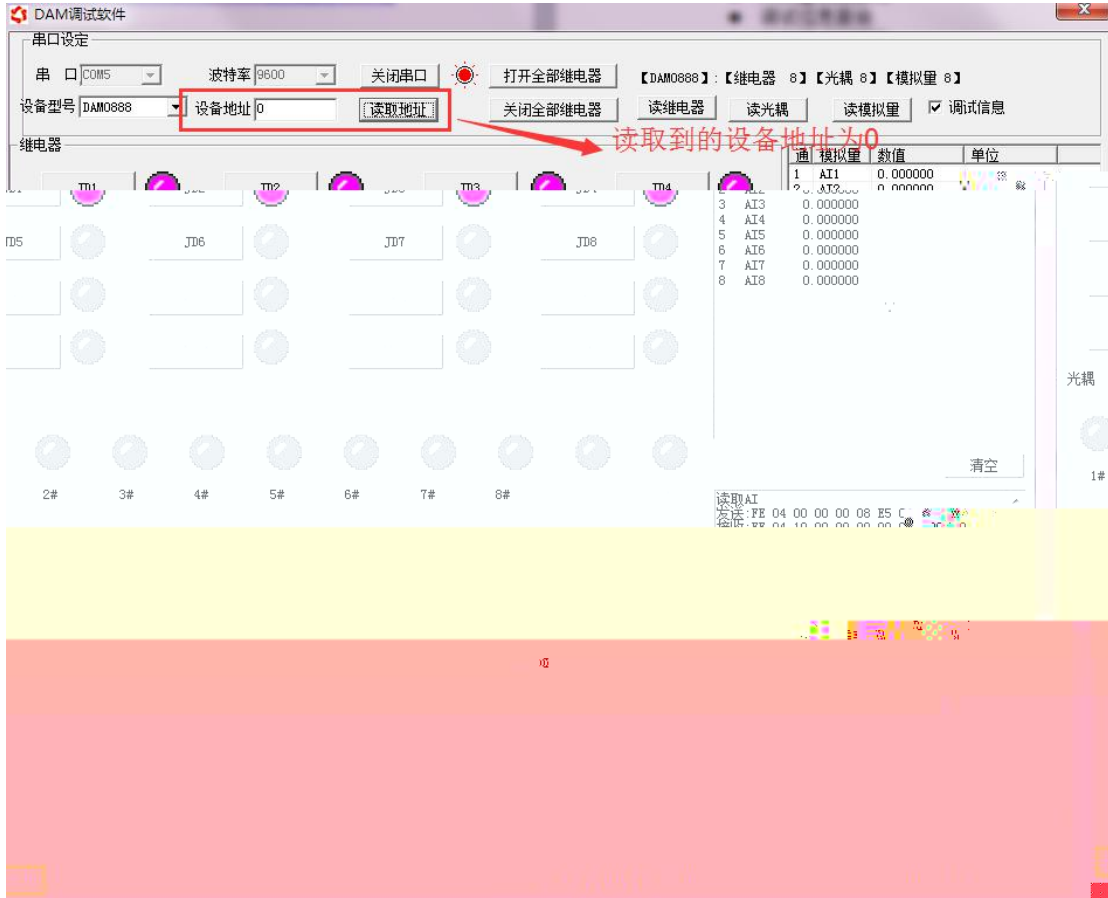
读取DI
读取DI
读取DI
读取DI
操作DO 打开第1个继电器
发送: FE 05 00 00 FF 00 98 35
接收: FE 05 00 00 FF 00 98 35
操作DO 打开第2个继电器
发送: FE 05 00 01 FF 00 C9 F5
接收: FE 05 00 01 FF 00 C9 F5
操作DO 打开第3个继电器
发送: FE 05 00 02 FF 00 39 F5
接收: FE 05 00 02 FF 00 39 F5
操作DO 打开第4个继电器
发送: FE 05 00 03 FF 00 08 35
接收: FE 05 00 03 FF 00 08 35
读取AI
发送: FE 04 00 00 00 08 E5 C3
接收: FE 04 10 00 00 00 00 00 00
00 00 00 00 00 00 00 71 2c
    
```

3



254

0



1

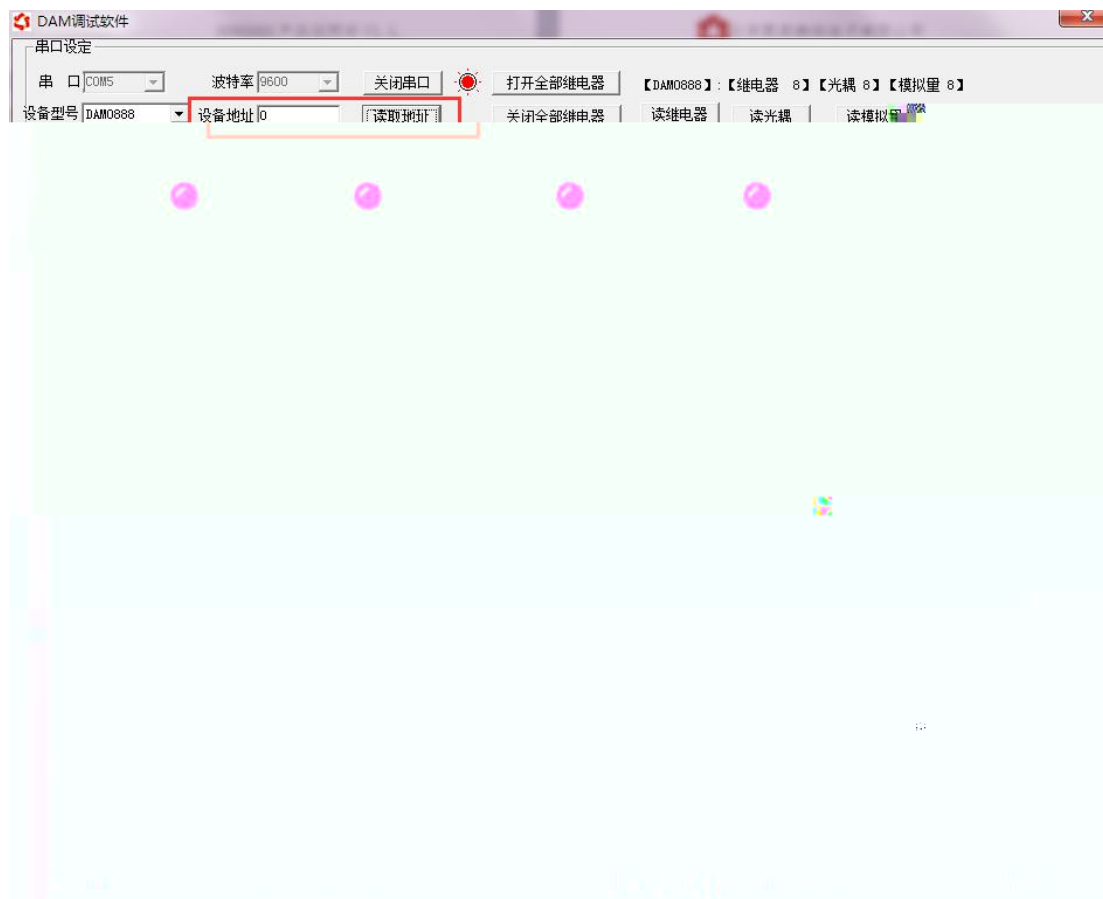
DAM

0

254

0

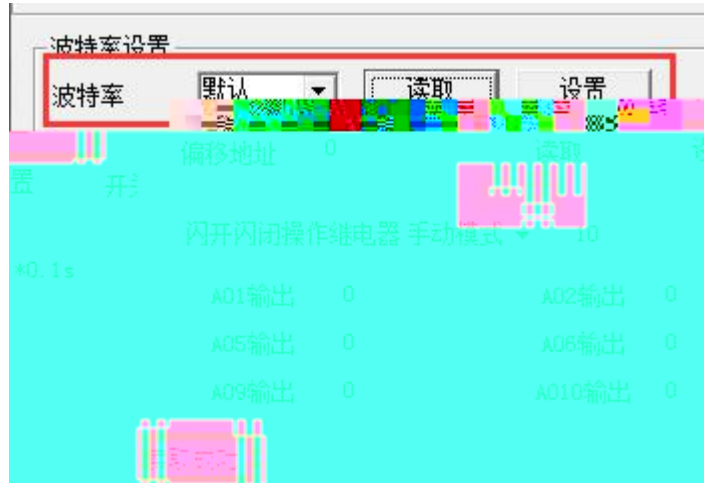
254



DAM



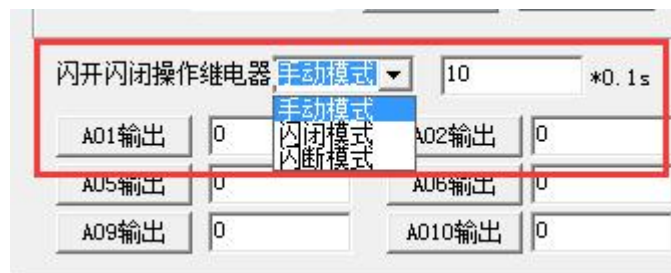
2



3

= *0.1 1 1.

= *0.1



1

modbus
 MODBUS Modbus
<http://www.juyingele.com.cn/software/software/Modbus%20POLL> .rar

2 M

2 4

2	
4	AD

1	1	0x0001	
2		0x0002	
3		0x0003	
4		0x0004	
5		0x0005	
6		0x0006	
7		0x0007	
8		0x0008	
1	4	3x0001	
2		3x0002	
3		3x0003	
4		3x0004	
5		3x0005	
6		3x0006	
7		3x0007	
8		3x0008	
9		3x0009	
10		3x0010	
11		3x0011	
12		3x0012	
13		3x0013	
14		3x0014	
15		3x0015	
16		3x0016	
		4x1001	0
			RS232
			0-5
			RS485
		4x1002	
		4x1003	= +

	FE 05 00 06 00 00 39 C4
	FE 05 00 07 FF 00 29 F4
	FE 05 00 07 00 00 68 04
	FE 02 02 00 00 AD AC
1	FE 04 00 00 00 01 25 C5
	FE 04 02 00 00 AD 24
2	FE 04 00 01 00 01 74 05
3	FE 04 00 02 00 01 84 05
4	FE 04 00 03 00 01 D5 C5
5	FE 04 00 04 00 01 64 04
6	FE 04 00 05 00 01 35 C4
7	FE 04 00 06 00 01 C5 C4
8	FE 04 00 07 00 01 94 04
9	FE 04 00 08 00 01 A4 07
10	FE 04 00 09 00 01 F5 C7
11	FE 04 00 0A 00 01 05 C7
12	FE 04 00 0B 00 01 54 07
13	FE 04 00 0C 00 01 E5 C6
14	FE 04 00 0D 00 01 B4 06
15	FE 04 00 0E 00 01 44 06
16	FE 04 00 0F 00 01 15 C6

5

FE 05 00 00 FF 00 98 35

FE			
05	05		
00 00			
FF 00			
98 35	CRC16	6	CRC16

FE 05 00 00 FF 00 98 35

FE			
05	05		
00 00			
FF 00			
98 35	CRC16	6	CRC16

FE 01 00 00 00 08 29 C3

FE			
01	01		
00 00			
00 08			
29 C3	CRC16	6	CRC16

FE 01 01 00 61 9C

FE			
01	01		0x81
01			$1+(n-1)/8$
00		Bit0: Bit1: Bit7:	
61 9C	CRC16	6	CRC16

= *0.001

FE 04 00 00 00 01 25 C5

FE			
04	04		
00 00			
00 01			
25 C5	CRC16		

FE 04 02 00 00 AD 24

FE			
04	04		0x82
02			
00 00	AD	0x0227	551
AD 24	CRC16		

