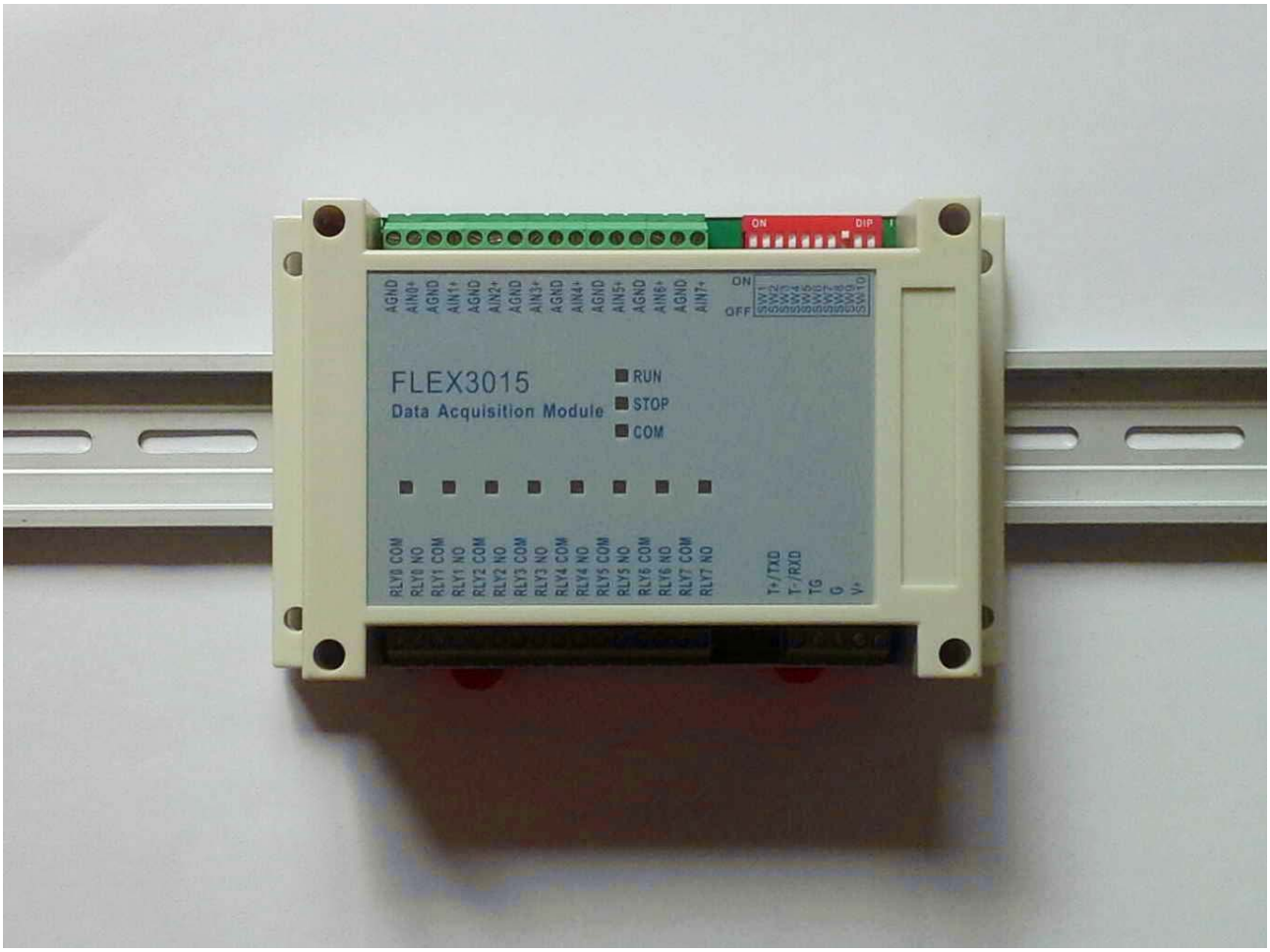
FLEX3015热敏电阻采集模块用户手册



目 录

1 产品介绍

FLEX-3015 模拟量采集模块是 FLEX-3000 系列智能测控模块之一，广泛应用于各种工业现场，提供了模拟量信号的采集以及转换，线性处理并转换成线性化的数据值，经 RS-485 总线传送到控制器。FLEX-3015 具有八个测量通道，可订制为各种型号的热敏电阻，0-1V，0-5V，0-10V，0-10mA，0-20mA，4-20mA 等常用模拟量输入通道。模块内部各处理单元之间提供了 1500V 的电气隔离，有效的防止模块因外界高压冲击而损坏，为工厂自动化以及楼宇自动化提供了高效的解决方案。模块主要特点如下：

- 8 通道模拟量输入
- 输入信号支持各种型号的热敏电阻，0-1V，0-5V，0-10V，0-10mA，0-20mA，4-20mA（需订货时说明）
- 宽电压范围输入（18-36V DC），功耗低
- RS-485 网络连接，支持 Modbus RTU/ASCII 协议
- 内置看门狗，运行稳定可靠
- 安装方便，标准导轨卡装或螺钉固定
- 宽温度范围运行

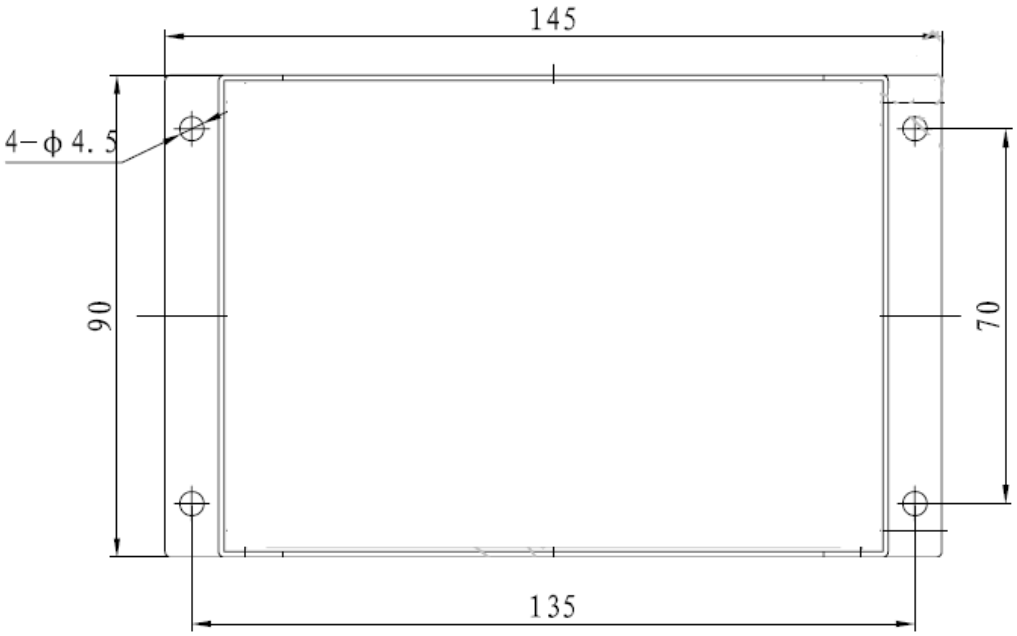
表 1　技术参数	
输入通道数量	8 输入通道
通讯接口	RS-485 光电隔离，ESD 保护，通讯距离：1200 米
通讯协议	Modbus RTU/ASCII，ADAM。支持无请求主动输出
传感器类型	各种型号的热敏电阻，0-1V，0-5V，0-10V，0-10mA，0-20mA，4-20mA（需订货时说明）
传感器连接方式	单端输入
输入类型/范围/精度	见表 2
隔离电压	模拟量输入对通讯 1500V 外部供电对通讯 1500V
采样速率	8通道/每秒（三次滤波）
分辨率	12 Bit，详见表 2
电源电压	18~36V DC
功耗	小于 2W @ 24VDC
储存环境	-40~85℃ 湿度<95%
运行环境	-20~85℃ 湿度<95%
外型尺寸	145*90*40mm
安装方式	螺钉固定/导轨

表 2　传感器类型				
传感器类型	测量范围	数据范围	分辨率	精度
NTC R(25℃)=10K,B25/50=3600	-40.0 ~ 125.0 ℃	-400~1250	优于 0.2℃	0.5%
NTC R(25℃)=5K,B25/50=3950	-40.0 ~ 125.0 ℃	-400~1250	优于 0.2℃	0.5%
NTC R(25℃)=10K,B25/50=3600	-40.0 ~ 125.0 ℃	-400~1250	优于 0.2℃	0.5%

模拟量输入	0-1V	0-4000	0.25mV@12Bit	0.1%
模拟量输入	0-5V	0-4000	1.25mV@12Bit	0.1%
模拟量输入	0-10V	0-4000	2.5mV@12Bit	0.1%
模拟量输入	0-10mA	0-4000	2.5uA@12Bit	0.1%
模拟量输入	0-20mA(4-20mA)	0-4000(800-4000)	5uA@12Bit	0.1%
NTC YSI-B-Mix-10K	-40.0 ~ 125.0 ℃	-400~1250	优于0.2℃	0.5%
NTC YSI-H-Mix-10K	-40.0 ~ 125.0 ℃	-400~1250	优于0.2℃	0.5%
NTC R(25℃)=10K,B25/50=3455	-40.0 ~ 125.0 ℃	-400~1250	优于0.2℃	0.5%
NTC R(25℃)=10K,B25/50=3435	-40.0 ~ 125.0 ℃	-400~1250	优于0.2℃	0.5%
NTC R(25℃)=10K,B25/50=3977	-40.0 ~ 125.0 ℃	-400~1250	优于0.2℃	0.5%
NTC R(25℃)=5K,B25/50=3470	-40.0 ~ 125.0 ℃	-400~1250	优于0.2℃	0.5%
NTC R(25℃)=10K,B25/50=3470	-40.0 ~ 125.0 ℃	-400~1250	优于0.2℃	0.5%
客户定制	客户定制		12Bit	

2 电气连接及安装

外形尺寸



端子说明

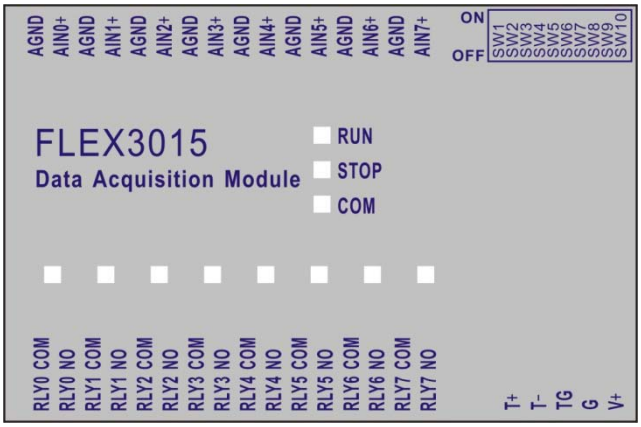
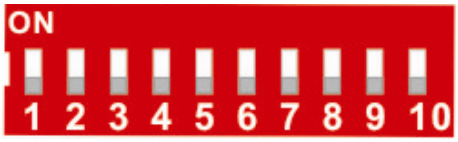


图2 接线端子

端子名称	说明	端子名称	说明	端子名称	说明
AGND	模拟量输入公共端	RLY0 COM	NC(未定义)	T+	RS485通讯正/RS232-TXD
AIN0+	模拟量输入通道0	RLY0 NO	NC(未定义)	T-	RS485通讯负/RS232-RXD
AGND	模拟量输入公共端	RLY1 COM	NC(未定义)	TG	RS485通讯地/RS232-GND
AIN1+	模拟量输入通道1	RLY1 NO	NC(未定义)	G	电源地
AGND	模拟量输入公共端	RLY2 COM	NC(未定义)	V+	电源正
AIN2+	模拟量输入通道2	RLY2 NO	NC(未定义)		
AGND	模拟量输入公共端	RLY3 COM	NC(未定义)	SW1	拨码开关1
AIN3+	模拟量输入通道3	RLY3 NO	NC(未定义)	SW2	拨码开关2
AGND	模拟量输入公共端	RLY4 COM	NC(未定义)	SW3	拨码开关3
AIN4+	模拟量输入通道4	RLY4 NO	NC(未定义)	SW4	拨码开关4
AGND	模拟量输入公共端	RLY5 COM	NC(未定义)	SW5	拨码开关5
AIN5+	模拟量输入通道5	RLY5 NO	NC(未定义)	SW6	拨码开关6
AGND	模拟量输入公共端	RLY6 COM	NC(未定义)	SW7	拨码开关7
AIN6+	模拟量输入通道6	RLY6 NO	NC(未定义)	SW8	拨码开关8
AGND	模拟量输入公共端	RLY7 COM	NC(未定义)	SW9	拨码开关9
AIN7+	模拟量输入通道7	RLY7 NO	NC(未定义)	SW10	拨码开关10

拨码开关

模块具有一个 10 位的拨码开关，拨码开关第 1-8 位：用于设置 Modbus 地址，可设置范围为 1-255。



拨码开关序号	1	2	3	4	5	6	7	8
地址位	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
地址=0(注1)	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
地址=1	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
地址=2	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF

地址=3	OFF	OFF	OFF	OFF	OFF	OFF	ON	ON
……	……	……	……	……	……	……	……	……
地址=127	OFF	ON	ON	ON	ON	ON	ON	ON
地址=128	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
地址=129	ON	OFF	OFF	OFF	OFF	OFF	OFF	ON
……	……	……	……	……	……	……	……	……
地址=254	ON	ON	ON	ON	ON	ON	ON	OFF
地址=255	ON	ON	ON	ON	ON	ON	ON	ON

注：当拨码开关全部为 OFF 时，模块的地址由内部寄存器设置，详见通信协议章节。

第 9,10 位用于设置模块的运行状态。具体如下：

拨码开关序号	9	10	模块状态
	OFF	OFF	运行模式
	OFF	**ON**	**设置模式**
	ON	OFF	运行模式
	ON	ON	运行模式

模块处于设置模式时，模块的 Modbus 地址默认为 0，通信配置默认为：9600,N,8,1（9600bps，无校验位，8 个数据位，一个停止位），通信协议默认为 Modbus-RTU，以方便用户与模块进行通信。这是设置模式与运行模式的唯一区别。

运行模式时，如果模块的外部拨码开关设置的 Modbus 地址为 0，则实际的 Modbus 地址由模块内部的地址寄存器决定；如果模块的外部拨码开关设置的 Modbus 地址不为 0，模块的 Modbus 地址由拨码开关第 1-8 位（或模块内部寄存器）决定；通信配置（波特率，校验位，通信协议）由模块内部寄存器的设置。

电源连接（供电为 18-36V 宽范围直流，建议使用 24V 直流电源）

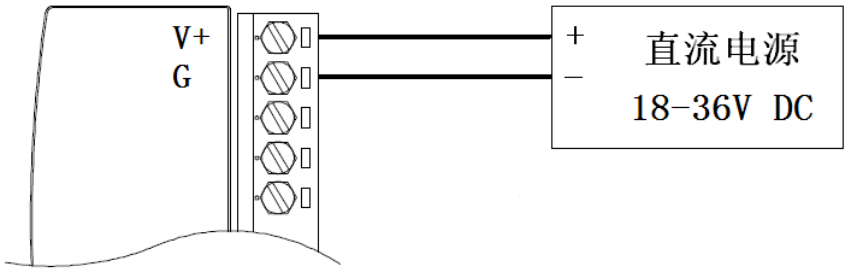


图3 电源接线

注：可连接 18~36V 直流电源供电，但电源功率必须满足模块要求。如在标准 24V 供电的情况下，模块功耗小于 2W，在选择电源时要选择大于 2W 的电源模块。当为多个模块供电的情况下，电源功率应大于(2W*模块数)。

传感器连接

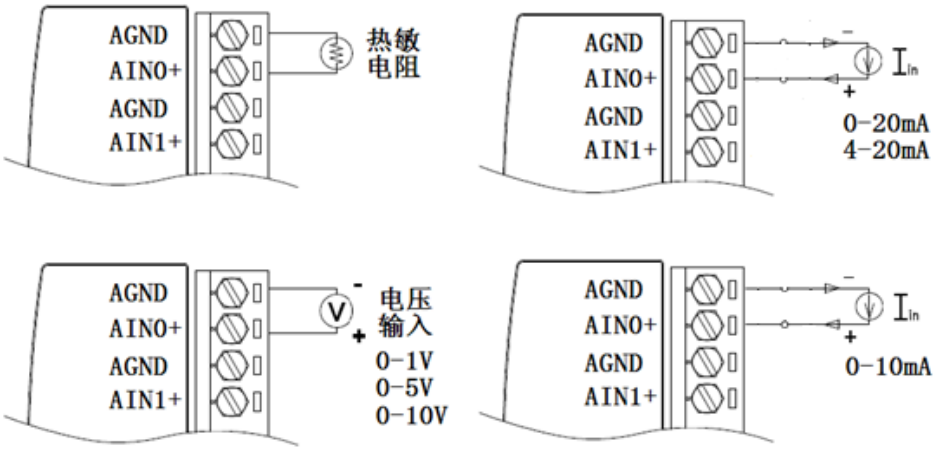


图4 传感器接线(请根据订货型号连接输入信号)

通讯连接

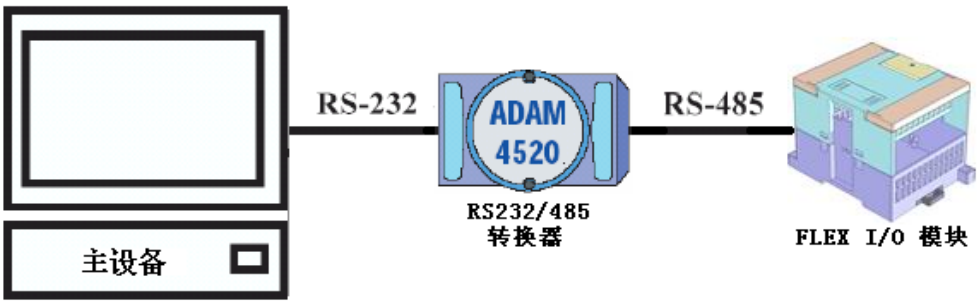


图5 主设备与模块通过RS232/RS485转换器连接

注：图中 RS232/485 转换器使用 ADAM4520 作为示意，可根据需要自行选择。

RS485 接口连接：

表4 RS485半双工连接					
主设备（一般为PC）			RS232/485转换器		FLEX模块
RS232引脚定义	DB9	DB25	主设备连接侧	FLEX模块侧	
RX	2	2	见所选择的RS232/485	A/A+	T+
TX	3	3	转换器用户手册	B/B-	T-
COM(公共地)	5	7		COM	TG

RS232 接口连接：

主设备（一般为PC,PLC）			FLEX模块
RS232引脚定义	DB9	DB25	
RX	2	2	T+
TX	3	3	T-
COM(公共地)	5	7	TG

注 1：终端电阻应根据通讯电缆的特性阻抗选择，一般情况下选择 Rt=120Ω

注 2：通讯电缆的屏蔽层可与 FLEX 模块通讯地(TG)连接

选型订购

系列	代码1	传感器类型
FLEX3015-	A	NTC R(25℃)=10K,B25/50=3600 -40.0 ~ 125.0 ℃
		NTC R(25℃)=5K,B25/50=3950 -40.0 ~ 125.0 ℃
		NTC R(25℃)=10K,B25/50=3892 -40.0 ~ 125.0 ℃
		NTC YSI-B-Mix-10K -40.0 ~ 125.0 ℃
		NTC YSI-H-Mix-10K -40.0 ~ 125.0 ℃
		NTC R(25℃)=10K,B25/50=3455 -40.0 ~ 125.0 ℃
		NTC R(25℃)=10K,B25/50=3435 -40.0 ~ 125.0 ℃
		NTC R(25℃)=10K,B25/50=3977 -40.0 ~ 125.0 ℃
		NTC R(25℃)=5K,B25/50=3470 -40.0 ~ 125.0 ℃
		NTC R(25℃)=10K,B25/50=3470 -40.0 ~ 125.0 ℃
		NTC R(25℃)=10K,B25/50=3950 -40.0 ~ 125.0 ℃
	D	模拟量输入 0-1V
	E	模拟量输入 0-5V
	F	模拟量输入 0-10V
	G	模拟量输入 0-10mA
	H	模拟量输入 0-20mA/4-20mA
	I	客户定制

代码2	继电器输出
C	有继电器控制输出
N	无继电器控制输出

代码3	通信接口
A	RS-485，Modbus RTU/ASCII协议
B	RS-232，Modbus RTU/ASCII协议
C	客户订制通信接口或协议

型号举例：

FLEX3015系列，传感器NTC R(25℃)=10K,B25/50=3600 -40.0 ~ 125.0 ℃，无继电器控制输出。RS-485，Modbus RTU/ASCII协议。选型代码如下：

FLEX3015-	A	N	A

3 通讯命令集

3.1 Modbus 通讯协议

寄存器	功能号	功能说明	读/写	有效值
0x0000	03/04	通道 0 测量值	读	对于温度，读出数据为温度实际值*10。读出数据后需要除以 10，代表当前的温度值。负数以补码表示。比如读出数据转换为十进制为 251，则 251/10=25.1 度。 对于电压或者电流信号，读出数据的范围为 0-4000，与电
0x0001	03/04	通道 1 测量值	读	
0x0002	03/04	通道 2 测量值	读	
0x0003	03/04	通道 3 测量值	读	
0x0004	03/04	通道 4 测量值	读	

0x0005	03/04	通道 5 测量值	读	压电流信号的量程成比例。举例如下：当前通道量程选择为 0-5V，读出数据为 2000，则((2000-0)/(4000-0))*(5V-0V)=2.5V。当前通道量程选择为 0-20mA，读出数据为 1500，则((1500-0)/(4000-0))*(20mA-0mA)=7.5mA。 当通道不连接传感器，或者超出测量范围时，读出的数据为 0x8000,即-32768。
0x0006	03/04	通道 6 测量值	读	
0x0007	03/04	通道 7 测量值	读	
0x0040	03/06/16	通道 0 校准值	读	校准值*10。 此寄存器用于调整实际测量到的温度与用户期望的温度的偏差，比如当前读到数据为 24.7,用户期望数值为 25.0，偏差为 25.0-24.7=0.3，写入校准值时需要将偏差扩大 10 倍，即 0.3*10=3。负数以补码表示。 **通道 N 测量值= 通道 N 测量值 + 通道 N 校准值。**如第一测量通道：寄存器 0x0020=0x0020+0x0040
0x0041	03/06/16	通道 1 校准值	读	
0x0042	03/06/16	通道 2 校准值	读	
0x0043	03/06/16	通道 3 校准值	读	
0x0044	03/06/16	通道 4 校准值	读	
0x0045	03/06/16	通道 5 校准值	读	
0x0046	03/06/16	通道 6 校准值	读	
0x0047	03/06/16	通道 7 校准值	读	
0x0060	03/06/16	通道 0 传感器类型	读/写	0=NTC R(25℃)=10K,B25/50=3600(热敏电阻)(默认值) 1=NTC R(25℃)=5K,B25/50=3950(热敏电阻) 2=NTC R(25℃)=10K,B25/50=3892(热敏电阻) 3=模拟量输入0-1V 4=模拟量输入0-5V 5=模拟量输入0-10V 6=模拟量输入0-10mA 7=模拟量输入0-20mA/4-20mA 8=NTC YSI-B-Mix-10K(热敏电阻) 9=NTC YSI-H-Mix-10K(热敏电阻) 10=NTC R(25℃)=10K,B25/50=3455 11=NTC R(25℃)=10K,B25/50=3435 12=NTC R(25℃)=10K,B25/50=3977 13=NTC R(25℃)=5K,B25/50=3470 14=NTC R(25℃)=10K,B25/50=3470 15=NTC R(25℃)=10K,B25/50=3950 16=RES_65000HM=0~32000 欧姆
0x0061	03/06/16	通道 1 传感器类型	读/写	
0x0062	03/06/16	通道 2 传感器类型	读/写	
0x0063	03/06/16	通道 3 传感器类型	读/写	

<table>
<tr><td>0x0064</td><td>03/06/16</td><td>通道 4 传感器类型</td><td>读/写</td><td rowspan="4"></td></tr>
<tr><td>0x0065</td><td>03/06/16</td><td>通道 5 传感器类型</td><td>读/写</td></tr>
<tr><td>0x0066</td><td>03/06/16</td><td>通道 6 传感器类型</td><td>读/写</td></tr>
<tr><td>0x0067</td><td>03/06/16</td><td>通道 7 传感器类型</td><td>读/写</td></tr>
<tr><td>0x0080</td><td>03/06/16</td><td>通道 0 滤波次数</td><td>读/写</td><td rowspan="8">1-16
1=1 次滤波
2=2 次滤波
3=3 次滤波(默认值)
…
16=16 次滤波</td></tr>
<tr><td>0x0081</td><td>03/06/16</td><td>通道 1 滤波次数</td><td>读/写</td></tr>
<tr><td>0x0082</td><td>03/06/16</td><td>通道 2 滤波次数</td><td>读/写</td></tr>
<tr><td>0x0083</td><td>03/06/16</td><td>通道 3 滤波次数</td><td>读/写</td></tr>
<tr><td>0x0084</td><td>03/06/16</td><td>通道 4 滤波次数</td><td>读/写</td></tr>
<tr><td>0x0085</td><td>03/06/16</td><td>通道 5 滤波次数</td><td>读/写</td></tr>
<tr><td>0x0086</td><td>03/06/16</td><td>通道 6 滤波次数</td><td>读/写</td></tr>
<tr><td>0x0087</td><td>03/06/16</td><td>通道 7 滤波次数</td><td>读/写</td></tr>
<tr><td>0x00D0</td><td>03/06/16</td><td>通信超时安全输出时间</td><td>读/写</td><td>0-255 秒（默认值为 0，为禁用）</td></tr>
<tr><td>0x00E0</td><td>03/06/16</td><td>继电器 0 配置</td><td>读/写</td><td rowspan="9">高 8 位定义：
<table><tr><td>H</td><td>.7</td><td>.6</td><td>.5</td><td>.4</td><td>.3</td><td>.2</td><td>.1</td><td>.0</td></tr></table>H.7:未定义
H.6:未定义
H.5:未定义
H.4:未定义
H.3:未定义
H.2:未定义
H.1:未定义
H.0:未定义</td></tr>
<tr><td>0x00E1</td><td>03/06/16</td><td>继电器 1 配置</td><td>读/写</td></tr>
<tr><td>0x00E2</td><td>03/06/16</td><td>继电器 2 配置</td><td>读/写</td></tr>
<tr><td>0x00E3</td><td>03/06/16</td><td>继电器 3 配置</td><td>读/写</td></tr>
<tr><td>0x00E4</td><td>03/06/16</td><td>继电器 4 配置</td><td>读/写</td></tr>
<tr><td>0x00E5</td><td>03/06/16</td><td>继电器 5 配置</td><td>读/写</td></tr>
<tr><td>0x00E6</td><td>03/06/16</td><td>继电器 6 配置</td><td>读/写</td></tr>
<tr><td>0x00E7</td><td>03/06/16</td><td>继电器 7 配置</td><td>读/写</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td>低 8 位定义：
<table><tr><td>L</td><td>.7</td><td>.6</td><td>.5</td><td>.4</td><td>.3</td><td>.2</td><td>.1</td><td>.0</td></tr></table>L.7：继电器上电输出值
1=吸合
0=断开(默认)
L.6：无有效通信超过一段时间后，继电器的安全输出值
1=吸合
0=断开(默认)
L.5:未定义
L.4, L.3, L.2, L.1， L.0：关联的温度通道
00000:通用 I/O，由通信设置。
00001:加热控制，温度低于设定值时输出
00010:制冷控制，温度高于设定值时输出
00011:上限报警，温度高于设定值时输出
00100:下限报警，温度低于设定值时输出
00101:上下限内报警，温度在上下限内时输出
00110:上下限外报警，温度在上下限外时输出</td></tr>
<tr><td>0x00F0</td><td>03/06/16</td><td>继电器 0 输出配置</td><td>读/写</td><td rowspan="8">高 8 位定义：
<table><tr><td>H</td><td>.7</td><td>.6</td><td>.5</td><td>.4</td><td>.3</td><td>.2</td><td>.1</td><td>.0</td></tr></table>H.7:未定义
H.6:未定义
H.5:未定义
H.4:未定义
H.3:未定义
H.2:未定义
H.1:未定义
H.0:未定义

低 8 位定义：
<table><tr><td>L</td><td>.7</td><td>.6</td><td>.5</td><td>.4</td><td>.3</td><td>.2</td><td>.1</td><td>.0</td></tr></table>L.7：未定义
L.6：未定义
L.5：未定义
L.4：未定义
L.3, L.2, L.1， L.0：关联的温度通道
0000：关联温度通道 0
0001：关联温度通道 1
0010：关联温度通道 2
0011：关联温度通道 3
0100：关联温度通道 4
0101：关联温度通道 5
0110：关联温度通道 6
0111：关联温度通道 7</td></tr>
<tr><td>0x00F1</td><td>03/06/16</td><td>继电器 1 输出配置</td><td>读/写</td></tr>
<tr><td>0x00F2</td><td>03/06/16</td><td>继电器 2 输出配置</td><td>读/写</td></tr>
<tr><td>0x00F3</td><td>03/06/16</td><td>继电器 3 输出配置</td><td>读/写</td></tr>
<tr><td>0x00F4</td><td>03/06/16</td><td>继电器 4 输出配置</td><td>读/写</td></tr>
<tr><td>0x00F5</td><td>03/06/16</td><td>继电器 5 输出配置</td><td>读/写</td></tr>
<tr><td>0x00F6</td><td>03/06/16</td><td>继电器 6 输出配置</td><td>读/写</td></tr>
<tr><td>0x00F7</td><td>03/06/16</td><td>继电器 7 输出配置</td><td>读/写</td></tr>
<tr><td>0x0100</td><td>03/06/16</td><td>继电器 0 报警上限值</td><td>读/写</td><td>读写值为温度实际值*10。</td></tr>
</table>

0x0101	03/06/16	继电器 1 报警上限值	读/写	
0x0102	03/06/16	继电器 2 报警上限值	读/写	
0x0103	03/06/16	继电器 3 报警上限值	读/写	
0x0104	03/06/16	继电器 4 报警上限值	读/写	
0x0105	03/06/16	继电器 5 报警上限值	读/写	
0x0106	03/06/16	继电器 6 报警上限值	读/写	
0x0107	03/06/16	继电器 7 报警上限值	读/写	
0x0110	03/06/16	继电器 0 报警上限值回差	读/写	读写值为温度实际值*10。
0x0111	03/06/16	继电器 1 报警上限值回差	读/写	
0x0112	03/06/16	继电器 2 报警上限值回差	读/写	
0x0113	03/06/16	继电器 3 报警上限值回差	读/写	
0x0114	03/06/16	继电器 4 报警上限值回差	读/写	
0x0115	03/06/16	继电器 5 报警上限值回差	读/写	
0x0116	03/06/16	继电器 6 报警上限值回差	读/写	
0x0117	03/06/16	继电器 7 报警上限值回差	读/写	
0x0120	03/06/16	继电器 0 报警下限值	读/写	读写值为温度实际值*10。
0x0121	03/06/16	继电器 1 报警下限值	读/写	
0x0122	03/06/16	继电器 2 报警下限值	读/写	
0x0123	03/06/16	继电器 3 报警下限值	读/写	
0x0124	03/06/16	继电器 4 报警下限值	读/写	
0x0125	03/06/16	继电器 5 报警下限值	读/写	
0x0126	03/06/16	继电器 6 报警下限值	读/写	
0x0127	03/06/16	继电器 7 报警下限值	读/写	
0x0130	03/06/16	继电器 0 报警下限值回差	读/写	读写值为温度实际值*10。
0x0131	03/06/16	继电器 1 报警下限值回差	读/写	
0x0132	03/06/16	继电器 2 报警下限值回差	读/写	
0x0133	03/06/16	继电器 3 报警下限值回差	读/写	
0x0134	03/06/16	继电器 4 报警下限值回差	读/写	
0x0135	03/06/16	继电器 5 报警下限值回差	读/写	

0x0136	03/06/16	继电器 6 报警下限值回差	读/写	
0x0137	03/06/16	继电器 7 报警下限值回差	读/写	
0x0140	03/06/16	继电器 0 报警设定值	读/写	读写值为温度实际值*10。
0x0141	03/06/16	继电器 1 报警设定值	读/写	
0x0142	03/06/16	继电器 2 报警设定值	读/写	
0x0143	03/06/16	继电器 3 报警设定值	读/写	
0x0144	03/06/16	继电器 4 报警设定值	读/写	
0x0145	03/06/16	继电器 5 报警设定值	读/写	
0x0146	03/06/16	继电器 6 报警设定值	读/写	
0x0147	03/06/16	继电器 7 报警设定值	读/写	
0x0150	03/06/16	继电器 0 报警设定值回差	读/写	读写值为温度实际值*10。
0x0151	03/06/16	继电器 1 报警设定值回差	读/写	
0x0152	03/06/16	继电器 2 报警设定值回差	读/写	
0x0153	03/06/16	继电器 3 报警设定值回差	读/写	
0x0154	03/06/16	继电器 4 报警设定值回差	读/写	
0x0155	03/06/16	继电器 5 报警设定值回差	读/写	
0x0156	03/06/16	继电器 6 报警设定值回差	读/写	
0x0157	03/06/16	继电器 7 报警设定值回差	读/写	
0x0200	03/06/16	Modbus 从机地址	读/写	1-255 （默认值为 1） 当拨码开关全部为 OFF 时，模块的 Modbus 地址由此寄存器设置。
0x0201	03/06/16	串行通讯波特率	读/写	0=1200 bps 1=2400 bps 2=4800 bps 3=9600 bps （默认值） 4=19200 bps 5=38400 bps 6=57600 bps 7=115200 bps
0x0202	03/06/16	串行通讯协议	读/写	**高 8 位定义：** H | .7 | .6 | .5 | .4 | .3 | .2 | .1 | .0 H.7:未定义 H.6:未定义 H.5:未定义 H.4:未定义

<table>
<tr><td></td><td></td><td></td><td></td><td>H.3:未定义
H.2:未定义
H.1:未定义
H.0:未定义

低 8 位定义:
<table><tr><td>L</td><td>.7</td><td>.6</td><td>.5</td><td>.4</td><td>.3</td><td>.2</td><td>.1</td><td>.0</td></tr></table>L.7: 串行通讯是否主动输出数据,
1=主动输出
0=请求响应(默认)
L.6: 研华协议是否校验
1=有校验
0=无校验(默认)
L.5:未定义
L.4:未定义
L.3:未定义
L.2:未定义
L.1, L.0: 通信协议
00= Modbus RTU (默认值)
01= Modbus ASCII
10= Adam 研华协议
11=未定义</td></tr>
<tr><td>0x0203</td><td>03/06/16</td><td>串行通讯校验位</td><td>读/写</td><td>0=无校验 (默认值)
1=偶校验
2=奇校验</td></tr>
<tr><td>0x0204</td><td>03/06/16</td><td>串行通讯数据位</td><td>读/写</td><td>读出写入均无效。默认为8个数据位。</td></tr>
<tr><td>0x0205</td><td>03/06/16</td><td>串行通讯停止位</td><td>读/写</td><td>读出写入均无效。默认为1个停止位。</td></tr>
<tr><td>0x0206</td><td>03/06/16</td><td>串行通讯延迟响应时间</td><td>读/写</td><td>0-255，以10ms为单位。
比如设置为10时，模块收到请求命令后，延时10*10ms=100ms 然后响应命令请求。
当设置为0时表示不延时，立刻响应请求命令。</td></tr>
<tr><td>0x0207</td><td>03/06/16</td><td>串行通讯主动输出数据时间间隔</td><td>读/写</td><td>0-255，以1s为单位。
比如设置为10时，模块会以10*1s=10s为时间间隔，以当前设置的通信协议进行数据输出。</td></tr>
<tr><td>0x0000</td><td>1/5/15</td><td>开关量输出通道0</td><td>读/写</td><td>0-1
0:输出关闭
1:输出开启</td></tr>
<tr><td>0x0001</td><td>1/5/15</td><td>开关量输出通道1</td><td>读/写</td><td>0-1
0:输出关闭
1:输出开启</td></tr>
<tr><td>0x0002</td><td>1/5/15</td><td>开关量输出通道2</td><td>读/写</td><td>0-1
0:输出关闭
1:输出开启</td></tr>
<tr><td>0x0003</td><td>1/5/15</td><td>开关量输出通道3</td><td>读/写</td><td>0-1</td></tr>
</table>

				0:输出关闭 1:输出开启
0x0004	1/5/15	开关量输出通道4	读/写	0-1 0:输出关闭 1:输出开启
0x0005	1/5/15	开关量输出通道5	读/写	0-1 0:输出关闭 1:输出开启
0x0006	1/5/15	开关量输出通道6	读/写	0-1 0:输出关闭 1:输出开启
0x0007	1/5/15	开关量输出通道7	读/写	0-1 0:输出关闭 1:输出开启

通信例 1：以读取 8 通道数据（下表中红色粗体）为例。通信前请先确认模块的 Modbus 地址，通讯配置默认为：9600,N,8,1（9600bps，无校验位，8 个数据位，一个停止位）

举例：读寄存器 0x0000H-0x0007H，即 8 通道测量值（负值按补码表示）

测量值寄存器：

寄存器	功能号	功能说明	读/写
0x0000	04	通道0测量值	读
0x0001	04	通道1测量值	读
0x0002	04	通道2测量值	读
0x0003	04	通道3测量值	读
0x0004	04	通道4测量值	读
0x0005	04	通道5测量值	读
0x0006	04	通道6测量值	读
0x0007	04	通道7测量值	读

Modbus-RTU 数据格式

请求：01 04 00 00 00 08 F1 CC（8 个字节）

从机地址	1 字节	0x01
功能号	1 字节	0x04
起始地址	2 字节	0x0000
寄存器数量	2 字节	0x0008
校验	2 字节	0xF1CC

响应：01 04 10 07 CF 80 00 80 00 80 00 80 00 80 00 80 00 80 00 43 3C（21个字节）

从机地址	1字节	0x01
功能号	1字节	0x04
有效字节数	1字节	0x10
数据	16字节	0x07（第0路高字节）

		0xCF（第0路低字节）
		0x80（第1路高字节）
		0x00（第1路低字节）
		0x80（第2路高字节）
		0x00（第2路低字节）
		0x80（第3路高字节）
		0x00（第3路低字节）
		0x80（第4路高字节）
		0x00（第4路低字节）
		0x80（第5路高字节）
		0x00（第5路低字节）
		0x80（第6路高字节）
		0x00（第6路低字节）
		0x80（第7路高字节）
		0x00（第7路低字节）
校验	2字节	0x433C

在 RTU 模式中采用 CRC(循环冗余检测)校验。具体的方法是将信息域中地址域、功能码、数据域的所有字节按规定的方式进行位移并进行 XOR(异或)计算，得到 2 字节的 CRC 码。校验码在信息帧作为一连续的流进行传输。从站在收到该信息帧时按同样的方式进行计算．并将结果同收到双字节校验码比较，如果一致就认为通信正确，否则认为通信有误，从站将发送错误应答。以下为校验的示意程序。

当接收到设备返回的 21 个字节数据后，进行以下 crc 计算操作，其中 num（输入参数 2）= 21

```
//-------------------------------------------------------------------------------------
//CRC 计算 C51 语言函数如下
//输入参数 1: snd，待校验的字节数组名
//输入参数 2: num，待校验的字节总数（包括 CRC 校验的 2 个字节）
//函数返回值: 校验失败时返回非 0 值。校验成功返回 0。
//-------------------------------------------------------------------------------------
unsigned int calc_crc16 (unsigned char *snd, unsigned char num)
{
  unsigned char i, j;
  unsigned int c,crc=0xFFFF;//crc 初始化为 0xFFFF
  for(i = 0; i < num; i ++)
  {
    c = snd[i] & 0x00FF;//待发送的字节和 0x00FF 进行“与”操作
    crc ^= c; //crc 与 c 做“异或”操作，结果存储于 crc 中
    for(j = 0;j < 8; j ++)
    {
      if (crc & 0x0001) //检查 crc 最低位是否为 1
      {
        crc>>=1; //crc 右移一位
        crc^=0xA001; //crc 与 0xA001 做“异或”操作，结果存储于 crc 中
```

```
            }
            else
            {
               crc>>=1; //crc 右移一位
            }
          }
      }
      return(crc);//返回 crc 校验结果
  }
```

得到返回结果为 0 时那么校验成功，如果校验失败返回为非零值。

校验成功后，使用以下公式计算（负值以补码表示）：

第 0 路数据= （0x07*256+0xCF） = 1999

第 1 路数据= （(0xFF*256+0x00) -0xFFFF-0x01） = -32768（未连接传感器的值或者异常值）

第 2 路数据= （(0xFF*256+0x00) -0xFFFF-0x01） = -32768（未连接传感器的值或者异常值）

第 3 路数据= （(0xFF*256+0x00) -0xFFFF-0x01） = -32768（未连接传感器的值或者异常值）

第 4 路数据= （(0xFF*256+0x00) -0xFFFF-0x01） = -32768（未连接传感器的值或者异常值）

第 5 路数据= （(0xFF*256+0x00) -0xFFFF-0x01） = -32768（未连接传感器的值或者异常值）

第 6 路数据= （(0xFF*256+0x00) -0xFFFF-0x01） = -32768（未连接传感器的值或者异常值）

第 7 路数据= （(0xFF*256+0x00) -0xFFFF-0x01） = -32768（未连接传感器的值或者异常值）

负值的判断与处理：如果返回值的二进制最高位为 1，那么表明返回的数据是负数，假设返回的值是 0xFF05（16 进制，补码），那么其二进制表示为：0b 1111111100000101，其最高位为 1，那么表明这个返回值是负数。处理数值时第一字节高字节为 0xFF，第二字节低字节为 0x 05，那么测量值为((0xFF*256+0x05) -0xFFFF-0x01) = (0xFF05-0xFFFF-0x01) =-251。

如果校验不成功，说明传输过程发生错误，应放弃此次采集到的数据，重新采集。

通信例 2：以读取 8 通道继电器输出为例。通信前请先确认模块的 Modbus 地址，通讯配置默认为：9600，N，8，1（9600bps，无校验位，8 个数据位，一个停止位）

举例：读开关量输出寄存器 0x0000H-0x0007H，即 8 通道继电器输出

Modbus-RTU 数据格式

请求：01 01 00 00 00 08 3D CC（8 个字节）

从机地址	1 字节	0x01
功能号	1 字节	0x01
起始地址	2 字节	0x0000
寄存器数量	2 字节	0x0008
校验	2 字节	0x3DCC

响应：01 01 01 01 90 48（5个字节）

从机地址	1字节	0x01
功能号	1字节	0x01
返回字节个数	1字节	0x01
数据	1字节	0x01,转换为二进制表示8个继电器输出的状态，即0b00000001（二进制），即第0通道的继电器输出为ON，其余1-7通道的状态为OFF。
校验	2字节	0x9048

Modbus-ASCII 数据格式

呼叫与应答的信息用 16 进制的字符 0~9、A~F 表示，每两个 ASCII 字符组成一个信息字节，字符冒号<:>表示待传递信息的起始处，字符<CR>（回车）、<LF>（换行）表示此信息传送结束。MODBUS 中的 ASCII 码方式多用于实时性要求不高的场合。

请求： :010400000008F3(CR)(LF)

定界符	:	命令起始字符
从机地址	01	
功能号	04	
起始地址	0000	
寄存器数量	0008	
校验和	F3	为一个字节的累加和转换成的两位 16 进制数 ASCII，除定界符“:”外，所有二进制字节和的相反数。
回车	CR	命令结束字符
换行	LF	命令结束字符

响应：:01041007CF8000800080008000800080008000800095(CR)(LF)

定界符	:	命令起始字符
从机地址	01	
功能号	04	
有效字节数	10	
数据		0x07（第 0 路高字节）
		0xCF（第 0 路低字节）
		0x80（第 1 路高字节）
		0x00（第 1 路低字节）
		0x80（第 2 路高字节）
		0x00（第 2 路低字节）
		0x80（第 3 路高字节）
		0x00（第 3 路低字节）
		0x80（第 4 路高字节）
		0x00（第 4 路低字节）
		0x80（第 5 路高字节）
		0x00（第 5 路低字节）
		0x80（第 6 路高字节）

		0x00（第 6 路低字节）
		0x80（第 7 路高字节）
		0x00（第 7 路低字节）
校验和	95	为一个字节的累加和转换成的两位 16 进制数 ASCII，除定界符“:”外，所有二进制字节和的相反数。
回车	CR	命令结束字符
换行	LF	命令结束字符

ASCII 方式采用 LRC(纵向冗余检测)校验，LRC 域是一个包含一个 8 位二进制值的字节。LRC 值由传输设备来计算并放到消息帧中，接收设备在接收消息的过程中计算 LRC，并将它和接收到消息中 LRC 域中的值比较，如果两值不等，说明有错误。具体的方法是将消息域中的地址域、功能码、数据域的所有 8 位的字节数据连续累加，即排出协议中的起始字符冒号以及结束字符回车换行．不考虑进位。得到 2 字节的 LRC 码。它仅仅是把每一个需要传输的数据按字节叠加后取反加 1 即可。以下为发送校验的示意程序，接收校验程序可按此原理得到。

```
BYTE GetCheckCode(const char * pSendBuf, int nEnd)//获得校验码
{
  BYTE byLrc = 0;
  char pBuf[4];
  int nData = 0;
  for(i=1; i<end; i+=2)     //i 初始为 1，避开“开始标记”冒号
  {
        //每两个需要发送的 ASCII 码转化为一个十六进制数
         pBuf [0] = pSendBuf [i];
         pBuf [1] = pSendBuf [i+1];
         pBuf [2] = '\0';
         sscanf(pBuf,"%x",& nData);
         byLrc += nData;
  }
  byLrc = ~ byLrc;
  byLrc ++;
  return byLrc;
}
```

3.2 ADAM 研华通信协议

研华协议是研华公司为其设备定义的一种通信协议，协议为 ASCII 码传输，简单直观。 模块支持带有校验字节的 ADAM 研华数据读取命令，可以读取单通道数据或者全部通道数据。如果希望使用研华协议作为模块的默认数据读取协议，可先使用设置软件将模块的通信协议设置为“研华 ADAM”通信协议，具体请参照设置软件使用章节。设置后重新上电以使协议生效。当模块通信协议为“研华 ADAM”通信协议时，仅支持数据读取命令，其他设置相关的命令不支持。

3.2.1 研华通信协议命令

请求	响应	功能	其他
#AAN(checksum) (CR)	>(data) (checksum) (CR)	读取单通道的数据	读取指定通道的数据
#AA (checksum) (CR)	>(data0) (data1) (data2) (data3) (data4) (data5) (data6) (data7) (checksum) (CR)	读取所有通道的数据	读取所有通道的数据

注：当选择研华协议无校验时，命令中的 checksum 字段将不出现。

3.2.1.1 读取单通道的数据命令

读取地址为 AA 的模块的第 N 通道的测量值。发送与接收全部用 ASCII 码表示。

请求： #AAN (checksum) (CR)

定界符	#	1 字节	命令起始字符
从机地址	AA	2 字节	AA（范围 00~FF）表示模块的两位十六进制地址
通道号	N	1 字节	N 为将要读出的通道号，N 值为 0~7
校验和	checksum	2 字节	校验和。为定界符，从机地址以及通道号数据的 ASCII 码 16 进制的累加和，然后对 100H(16 进制)求余数。并将余数转换为 ASC 码作为 checksum 发送。
回车	CR	1 字节	命令结束字符，即回车(0DH)

响应：>(data) (checksum) (CR)

定界符	>	1 字节	命令起始字符
通道数据	data	7 字节	通道测量值，以“+”或“-”开头，后面是 4 位十进制整数，一位小数点以及一位小数，测量值小于四位整数是前面以 0 补位， 比如：-0035.7，+0125.6，+1200.0。 传感器异常时输出-3276.8。
校验和	checksum	2 字节	校验和。为定界符，通道数据 data 的 ASCII 码 16 进制的累加和，然后对 100H(16 进制)求余数。然后将余数转换为两个 ASC 码并与接收到的 checksum 比较，相等说明传输无误。
回车	CR	1 字节	命令结束字符，即回车(0DH)

举例如下：读取 01 号模块的第 0 通道数据。

请求：#010B4(CR)

定界符	#	1 字节	命令起始字符
从机地址	01	2 字节	模块地址为 01
通道号	0	1 字节	通道 0
校验和	B4	2 字节	校验值 B4 计算如下：定界符，从机地址以及通道号数据的 ASCII 码 16 进制的累加和，然后对 100H(16 进制)求余数。即（’#’+’0’+’1’+’0’）=(23H+30H+31H+30H) =B4H，B4H%100H=B4H。然后将 B4H 转换为两个 ASC 码‘B’，‘4’发送出去。
回车	CR	1 字节	命令结束字符，即回车(0DH)

响应：>+0265.99D(CR)

定界符	>	1 字节	命令起始字符

通道数据	+0265.9	7 字节	通道 0 测量值为 265.9 度
校验和	9D	2 字节	校验值 9D 计算如下：定界符，通道数据的 ASCII 码 16 进制的累加和，然后对 100H(16 进制)求余数。即（‘>’ + ‘+’ + ‘0’ + ‘2’ + ‘6’ + ‘5’ + ‘.’ + ‘9’）=(3EH+2BH+30H+32H +36 H +35 H +2E H +39 H)= 19DH，19DH %100H=9DH。然后将 9DH 转换为两个 ASC 码 ‘9’，‘D’ 并与接收到的校验和比较，相等说明传输无误。
回车	CR	1 字节	命令结束字符，即回车(0DH)

3.2.1.2 读取所有通道的数据命令

读取地址为 AA 的模块的所有通道的测量值。发送与接收全部用 ASCII 码表示。

请求： #AA (checksum) (CR)

定界符	#	命令起始字符
从机地址	AA	AA（范围 00~FF）表示模块的两位十六进制地址
校验和	checksum	校验和。为定界符，从机地址的 ASCII 码 16 进制的累加和，然后对 100H(16 进制)求余数。并将余数转换为 ASC 码作为 checksum 发送。
回车	CR	命令结束字符，即回车(0DH)

响应：>(data) (checksum) (CR)

定界符	>	命令起始字符
通道 0 数据	Data0	通道 0 测量值，以“+”或“-”开头，后面是 4 位十进制整数，一位小数点以及一位小数，测量值小于四位整数是前面以 0 补位， 比如：-0035.7，+0125.6，+1200.0。 传感器异常时输出-3276.8。
通道 1 数据	Data1	通道 1 测量值，同上所述
通道 2 数据	Data2	通道 2 测量值，同上所述
通道 3 数据	Data3	通道 3 测量值，同上所述
通道 4 数据	Data4	通道 4 测量值，同上所述
通道 5 数据	Data5	通道 5 测量值，同上所述
通道 6 数据	Data6	通道 6 测量值，同上所述
通道 7 数据	Data7	通道 7 测量值，同上所述
校验和	checksum	校验和。为定界符，通道数据 Data0，Data1，Data2，Data3，Data4，Data5，Data6，Data7 的 ASCII 码 16 进制的累加和，然后对 100H(16 进制)求余数。然后将余数转换为两个 ASC 码并与接收到的 checksum 比较，相等说明传输无误。
回车	CR	命令结束字符，即回车(0DH)

举例如下：读取 01 号模块的所有数据。

请求：#0184(CR)

定界符	#	1 字节	命令起始字符
从机地址	01	2 字节	模块地址为 01
校验和	84	2 字节	校验值 84 计算方法如上节所述
回车	CR	1 字节	命令结束字符，即回车(0DH)

响应：>+0199.7-3276.8-3276.8-3276.8-3276.8-3276.8-3276.8-3276.864(CR)

定界符	>	1 字节	命令起始字符
通道 0 数据	+0199.7	7 字节	通道 0 测量值为 199.7 度
通道 1 数据	-3276.8	7 字节	通道 1 测量值为传感器异常值-3276.8
通道 2 数据	-3276.8	7 字节	通道 2 测量值为传感器异常值-3276.8
通道 3 数据	-3276.8	7 字节	通道 3 测量值为传感器异常值-3276.8
通道 4 数据	-3276.8	7 字节	通道 4 测量值为传感器异常值-3276.8
通道 5 数据	-3276.8	7 字节	通道 5 测量值为传感器异常值-3276.8
校验和	64	2 字节	校验值 64 计算方法如上节所述
回车	CR	1 字节	命令结束字符，即回车(0DH)

3.3 ASCII 码对照表

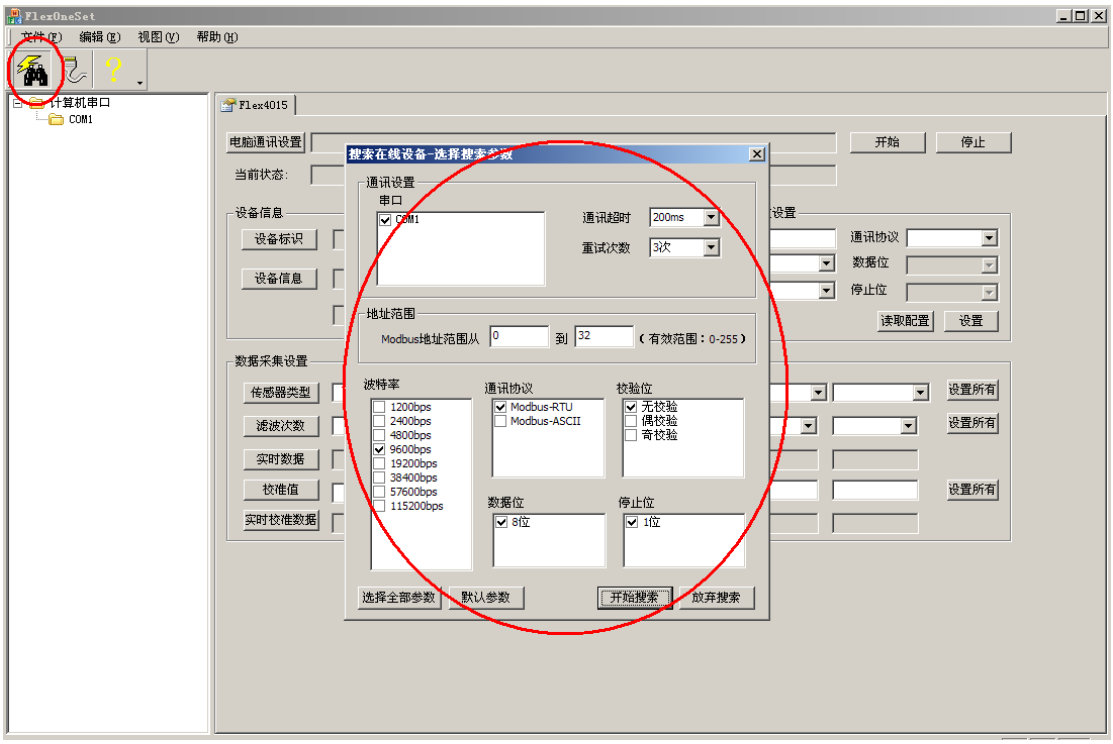
HEX	ASCII	HEX	ASCII	HEX	ASCII	HEX	ASCII
21	!	40	@	5F	_	7E	~
22	""	41	A	60	'		
23	#	42	B	61	a		
24	$	43	C	62	b		
25	%	44	D	63	c		
26	&	45	E	64	d		
27	'	46	F	65	e		
28	(	47	G	66	f		
29	)	48	H	67	g		
2A	*	49	I	68	h		
2B	+	4A	J	69	i		
2C	,	4B	K	6A	j		
2D	-	4C	L	6B	k		
2E	.	4D	M	6C	l		
2F	/	4E	N	6D	m		
30	0	4F	O	6E	n		
31	1	50	P	6F	o		
32	2	51	Q	70	p		
33	3	52	R	71	q		
34	4	53	S	72	r		
35	5	54	T	73	s		
36	6	55	U	74	t		
37	7	56	V	75	u		
38	8	57	W	76	v		
39	9	58	X	77	w		
3A	:	59	Y	78	x		
3B	;	5A	Z	79	y		
3C	<	5B	[	7A	z		
3D	=	5C	\	7B	{		
3E	>	5D	]	7C	|		
3F	?	5E	^	7D	}		

4 设置软件使用说明

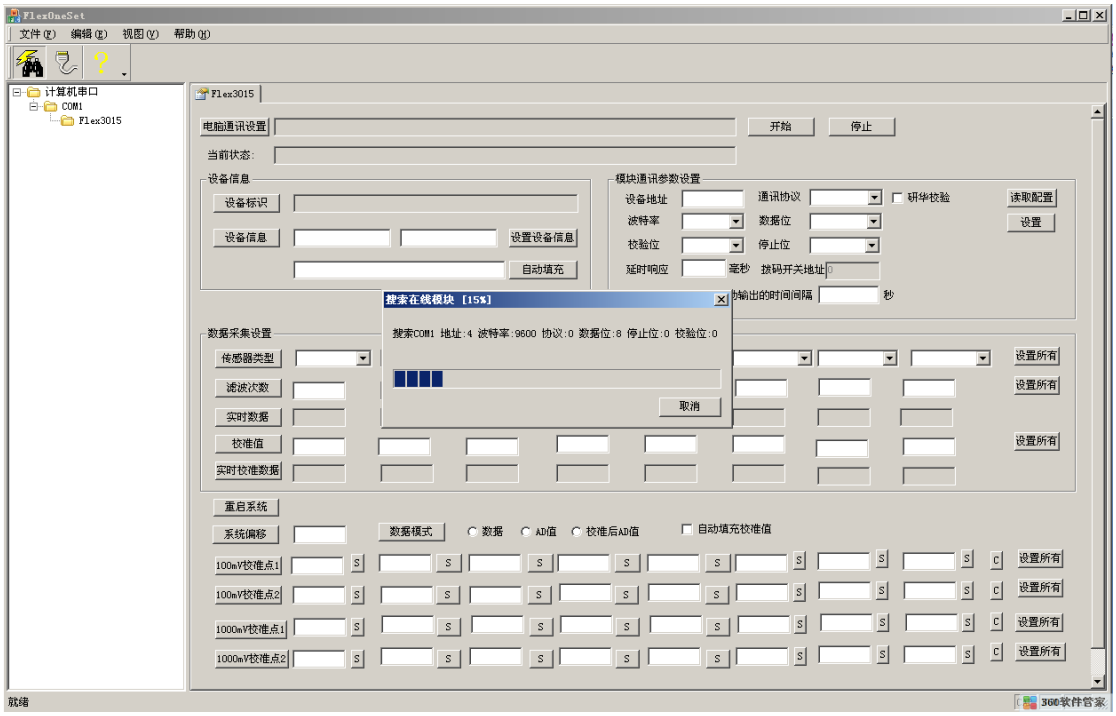
4.1 设置软件与处于设置状态的模块通信

先将模块的拨码开关第 9 位拨为 OFF，第 10 位拨为 ON，然后重新上电模块，此时模块进入设置模式，模块的 Modbus 地址为固定为 0，通信参数固定为：9600，N,8,1（9600bps，无校验，8 个数据位，一个停止位）。打开设置软件。如果模块的通信地址，通信参数已知，也可不必设置拨码开关第 9 位拨为 OFF，第 10 位拨为 ON，在以下步骤的串口搜索参数中选择适当的参数即可。

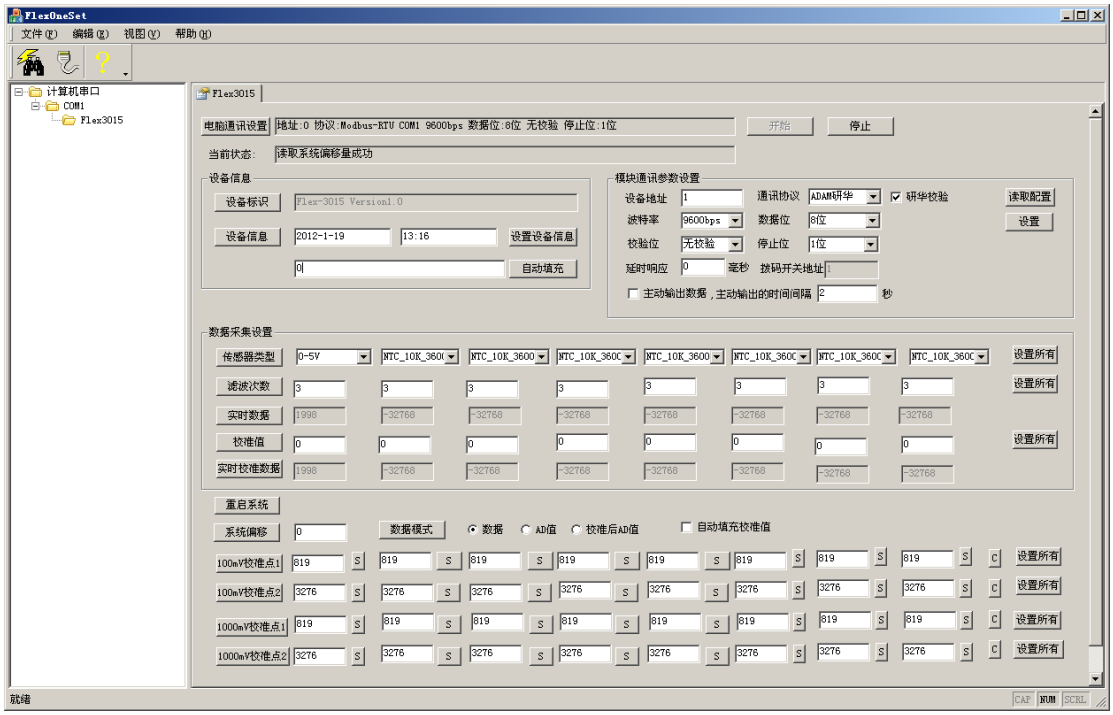
点选工具条中的搜索模块按钮，并确认通信设置对话框中的参数设置为 9600bps，无校验，8 个数据位， 一个停止位。地址范围从 0-32。选择串口号（确保此串口未被其他程序占用），然后点击“开始搜索”按钮。



搜索过程中，所有检测到的模块均会添加到左侧设备栏中，如图：

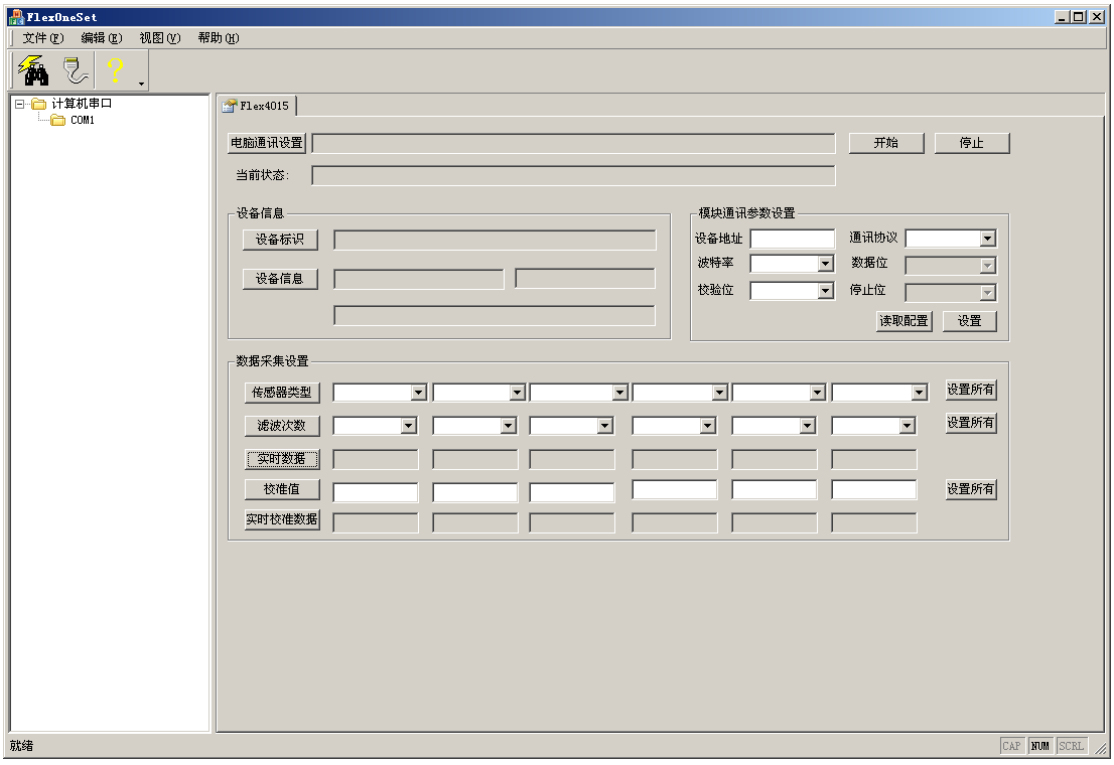


双击左侧计算机串口树形目录下的“Flex3015”设备，通信参数将自动复制到右侧“电脑通讯设置”一栏中，然后点击右侧“开始”按钮，即可与此设备通信，并开始设置模块的参数。

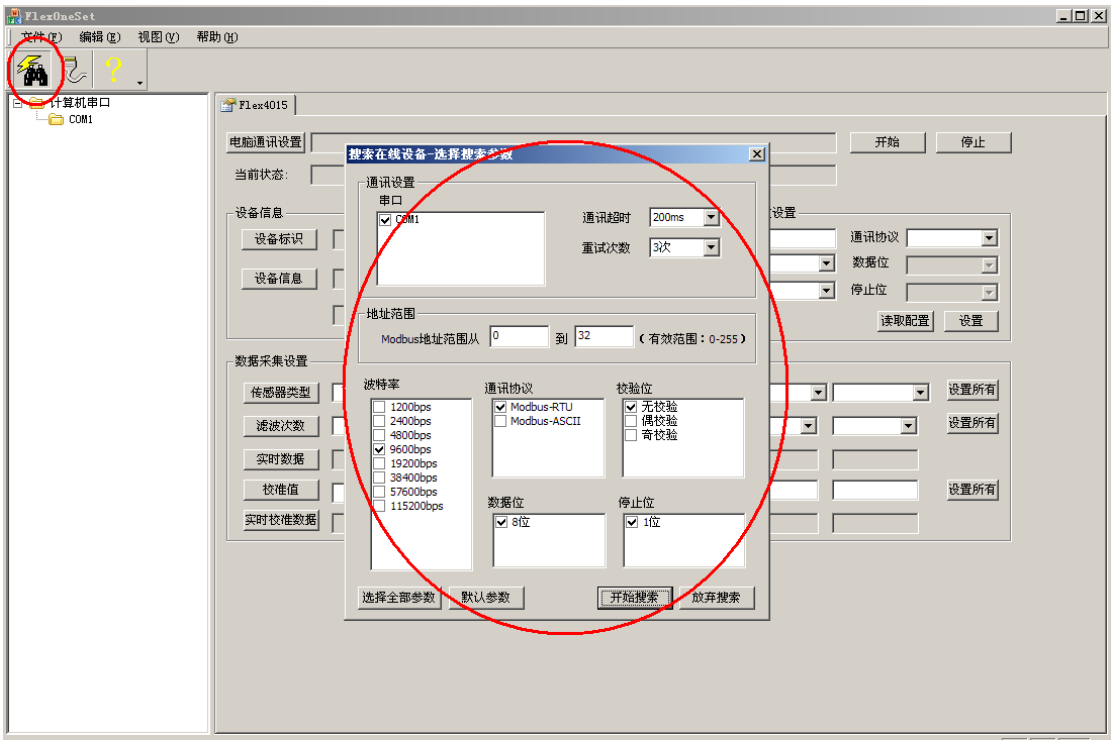


4.2 设置软件与处于运行状态的模块通信

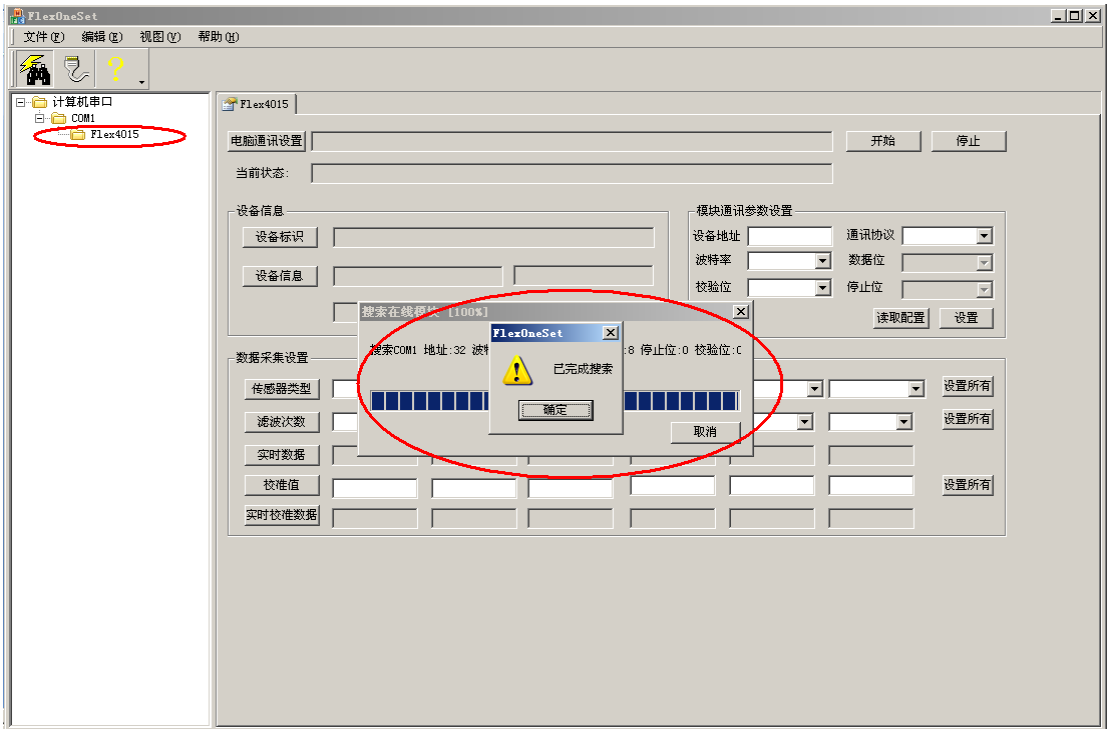
在不知道模块地址，波特率等串口通信参数时，启动设置软件，如下：



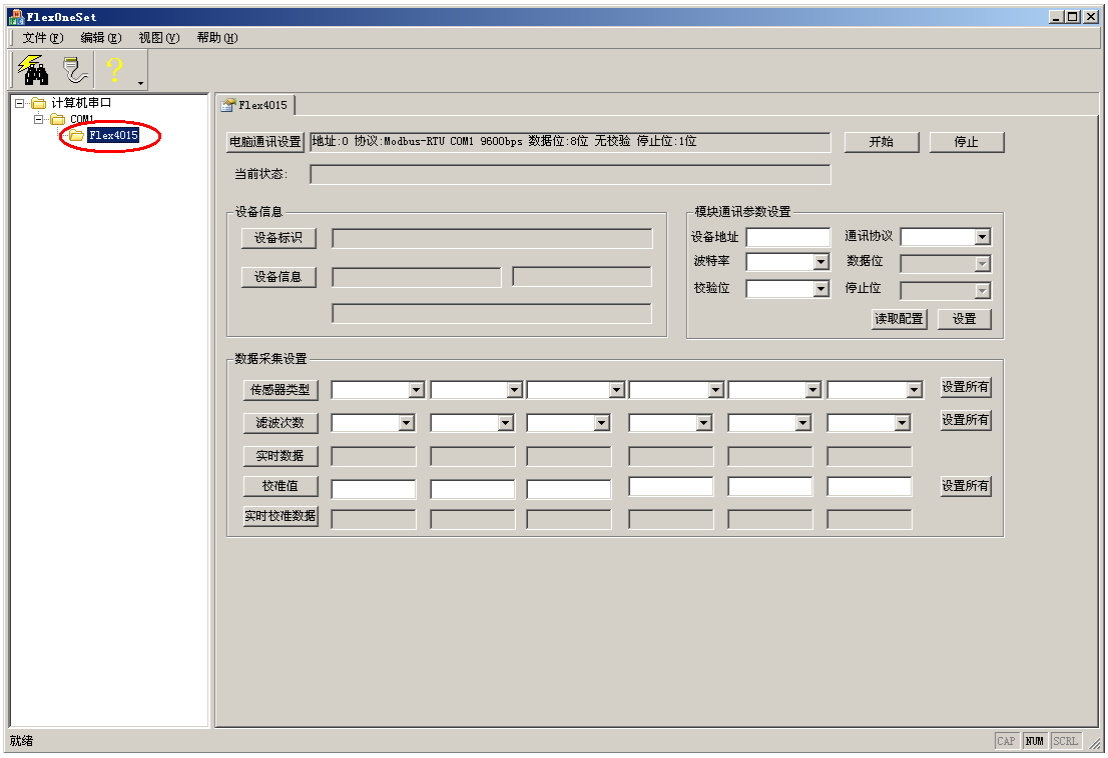
点选工具条中的搜索模块按钮，并选择通信设置对话框中可能的通信参数，地址范围。然后点击“开始搜索”按钮。



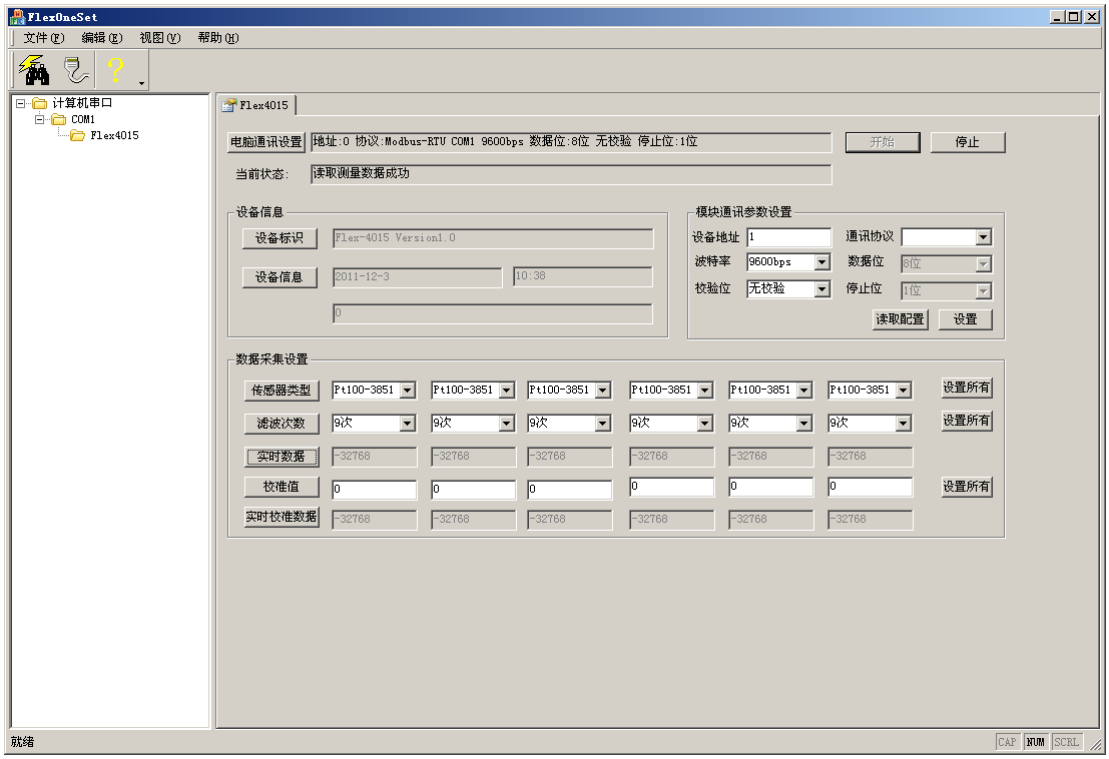
搜索过程中，所有检测到的模块均会添加到左侧设备栏中，如图：



双击左侧计算机串口树形目录下的“Flex3015”设备，通信参数将复制到右侧“电脑通讯设置”一栏中：

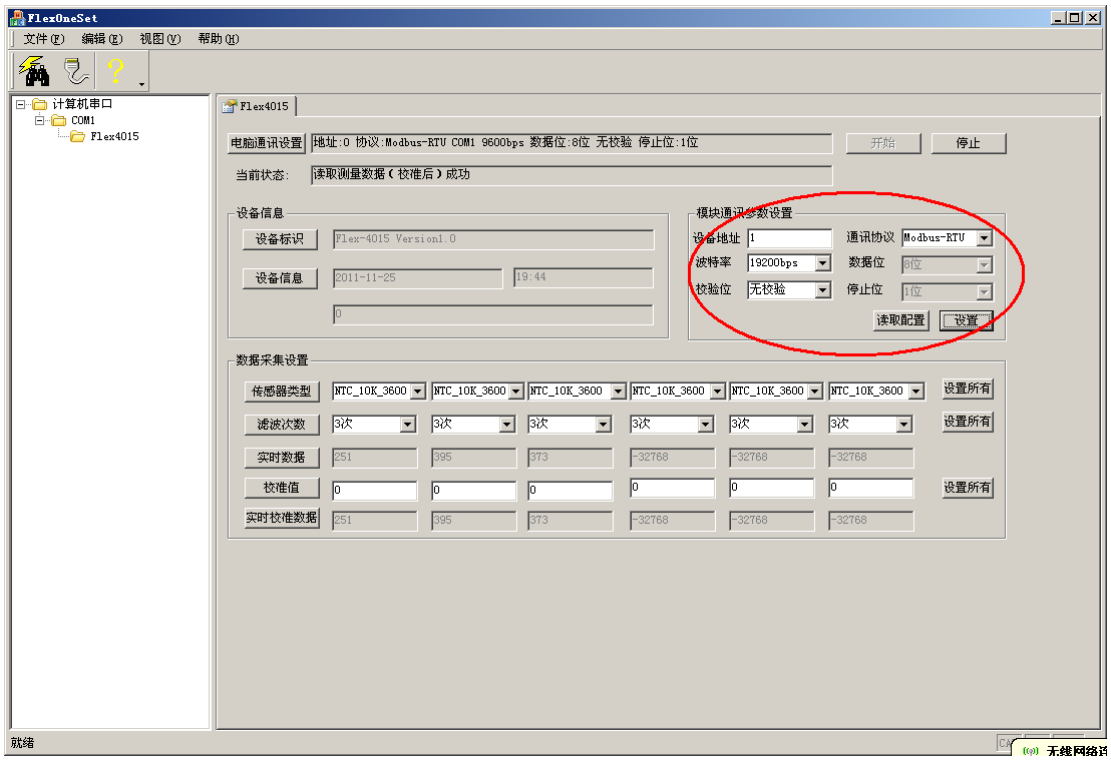


然后点击右侧“开始”按钮，即可与此设备通信。



4.3 串口通信参数如何设置

开始通讯后，在“模块通讯参数设置”一栏中，选择期望的通讯参数，如图，选择后，点击“设置”按钮，稍后更新成功后会弹出设置成功对话框。设置通讯参数后，**确保拨码开关不在全部 OFF 的状态，然后将模块重新上电，以使得通讯参数生效**。此时，模块即可按照设置的通信参数进行通信。



5 使用串口调试软件读取数据

串口调试软件以 SSCOM32 为例。注意软件的波特率，校验位，数据位，以及停止位必须与模块的设置一致方可通信。

5.1 Modbus-RTU 通信协议

发送：01 04 00 00 00 08 F1 CC

接收：01 04 10 07 CF 80 00 80 00 80 00 80 00 80 00 80 00 80 00 43 3C

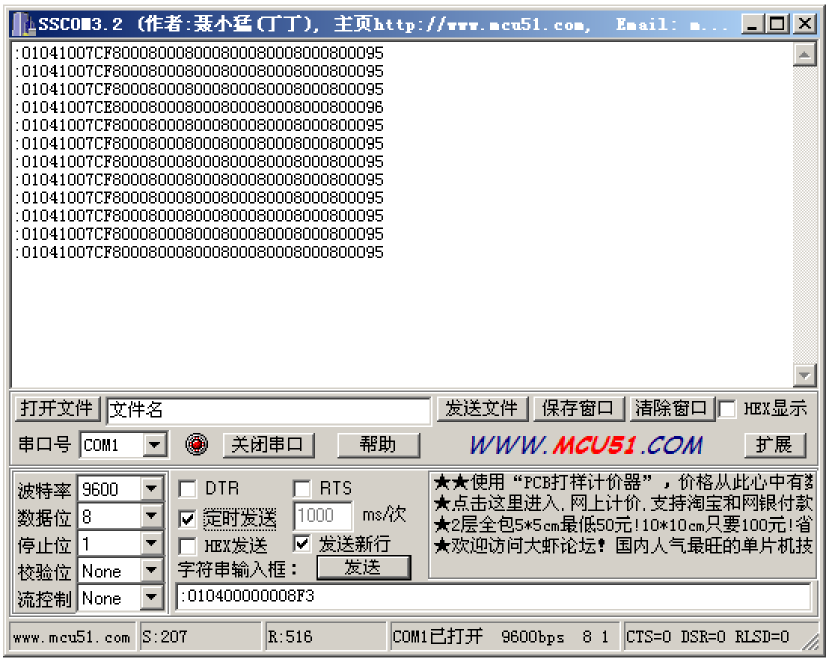
注意软件中要点选“HEX 发送”，“HEX 显示”

5.2 Modbus-ASCII 通信协议

发送：:010400000008F3

接收：:01041007CF80008000800080008000800080000095

注意软件中要点选“发送新行”，并取消选择“HEX 发送”，“HEX 显示”



5.3 ADAM 研华通信协议

发送：#0184

接收：>+1999.0-3276.8-3276.8-3276.8-3276.8-3276.8-3276.8-3276.866

注意软件中要点选“发送新行”，并取消选择“HEX 发送”，“HEX 显示”

修订说明

修订时间	修订说明
2012-8-31	增加 RS232 接口连接说明以及通讯接口选型