

# CCS 系列单通道光谱共焦 传感器

MODBUS 通信协议

Version 1.0.0.8

2024 年 05 月 31 日

## 变更记录

日期	版本	变更内容
2022 年 3 月 2 日	V1.0.0.0	发版
2022 年 3 月 9 日	V1.0.0.1	增加“附录 2 Modbus 通信参数配置”内容
2022 年 3 月 17 日	V1.0.0.2	修改 4.2 数据读取类寄存器对应的地址
2022 年 4 月 8 日	V1.0.0.3	新增数据滑动平均滤波次数:1、2
2022 年 6 月 7 日	V1.0.0.4	更新 3.2.2 功能码 04 的内容
2022 年 8 月 1 日	V1.0.0.5	在 V1.4 硬件版本上增加 2 项功能： （1）增加触发计数器复位功能； （2）数据读取类增加通道 1 的峰 1、峰 2 高度值。
2023 年 8 月 22 日	V1.0.0.6	新增 TCP 格式帧与数据缓存功能
2023 年 11 月 10 日	V1.0.0.7	修改了部分报文示例说明
2024 年 05 月 31 日	V1.0.0.8	添加多层测厚相关寄存器

## 目录

1. 引言 .....	4
2. 通信规格 .....	5
3. Modbus 通信规约 .....	6
3.1. RTU 模式信息帧格式 .....	6
3.2. TCP 模式信息帧格式 .....	7
3.3. 传感器支持的功能码 .....	9
4. Modbus 寄存器分配 .....	14
4.1. 参数设置类寄存器 .....	14
4.2. 数据读取类寄存器 .....	19
5. 应用示例 .....	25
5.1. 读取参数示例（功能码 03） .....	25
5.2. 设置参数示例（功能码 06） .....	27
5.3. 读取数据示例（功能码 04） .....	29
5.4. 读取缓存数据示例 .....	32
6. 附录 .....	34
附录 1 Modbus RTU CRC 算法代码 .....	34
附录 2 Modbus 通信参数设置 .....	37

# 1. 引言

本文档介绍了如何使用 Modbus RTU/TCP 协议,实现对 CCS 系列单通道光谱共焦传感器的参数配置与数据读取功能。

**使用前, 注意事项:**

- (1) 使用 Modbus RTU 协议时, 传感器只能作为从机使用。
- (2) 使用前, 请确保从机地址、串口波特率设置正确, 从机地址取值范围: 1 ~ 247。
- (3) 从机地址、串口波特率可通过配套上位机软件进行设置, 串口波特率设置最大值为 115200 bits/s。(默认设置: 从机地址为 1, 波特率为 115200 bits/s。)
- (4) 传感器从上电到完成配置, 大约需要 6s。在配置完成后, 才可通过 Modbus RTU/TCP 协议读写寄存器数据, 否则读写操作会超时。
- (5) 使用 Modbus TCP 协议时, 不同设备通常可用 IP 地址进行区分, 从机地址默认 1

Modbus 通信参数设置操作, 见“附录 2 Modbus 通信参数设置”。

## 2. 通信规格

按照表 2-1 规格，进行 MODBUS RTU 串行通信，波特率通过配套的上位机软件进行设置。

表 2-1 通信规格

通信规格	通信接口	RS-485，半双工 2 线制
	波特率	19200/38400/57600/115200
	数据位	8 位
	奇偶校验位	无
	停止位	1 位

表 2- 1 MODBUS RTU 通信规格

按照表 2-2 规格，进行 MODBUS TCP 通信，IP 地址通过配套的上位机软件进行设置。

通信规格	通信接口	TCP/IP
	监听端口	502
	IP 地址	192.168.0.10（默认，可通过上位机软件更改）

表 2- 2 MODBUS TCP 通信规格

### 3. Modbus 通信规约

#### 3.1. RTU 模式信息帧格式

表 3-1 RTU 模式信息帧格式

开始	地址字段	功能代码	数据	错误校验 (CRC)	结束（起始）
3.5 个字符时间以上的间隔	1 个字节	1 个字节	0~252 个字节	2 字节	3.5 个字符时间以上的间隔

(1) 地址字段

地址字段为从设备地址，地址取值范围：0~247（十进制）。单个设备的实际地址范围是 1~247，地址 0 作为广播地址。主设备发送消息时，将从设备地址放到信息帧中，以便从设备识别此消息是否是发给本机的，如是本机地址，则从设备做出响应。从设备回复主设备时，在回应消息的地址域中提供本机地址，以便主设备识别是哪个从设备返回的数据。

(2) 功能码

功能码用于表示消息帧的功能，取值范围为 1~255（十进制）。从设备根据功能码，执行相应的功能，执行完成后在响应消息帧中返回同样的功能码。如果出现异常，返回的消息帧中将功能码最高位（MSB）设置为 1。

(3) 数据域

数据域存放功能码需要操作的具体数据。数据域以字节为单位，长度可变。

(4) 错误校验（CRC）

参与 CRC 计算的数据区域：包含一帧内从地址字段至数据区段内容，即计算 CRC 校验前的所有字节。

在 Modbus RTU 串行通信中，错误校验码是 16 位（2 个字节）的二进制值，采用 CRC-16 校验方法，多项式为 CRC-16/MODBUS:  $x^{16} + x^{15} + x^2 + 1$ 。

错误校验码在数据帧中的传输顺序为：低字节在前，高字节在后。

CRC 值由发送设备计算，并添加到报文中。接收设备在报文接收过程中重新计算 CRC，并和接收的实际值进行比较,进行比较的值如果不同则为出错。

CRC 具体算法,见附录 1：Modbus RTU CRC 算法代码

3.2. TCP 模式信息帧格式

Modbus TCP 数据帧与 Modbus RTU 协议数据帧的区别：

Modbus TCP 协议是在 RTU 协议前面添加 MBAP 报文头，由于 TCP 是基于可靠连接的服务，RTU 协议中的 CRC 校验码在此不需要，因此在 Modbus TCP 协议中没有 CRC 校验码；同时 RTU 报文中的“地址字段”被“单元标识符”替代，放在 Modbus TCP 应用协议报文头中。

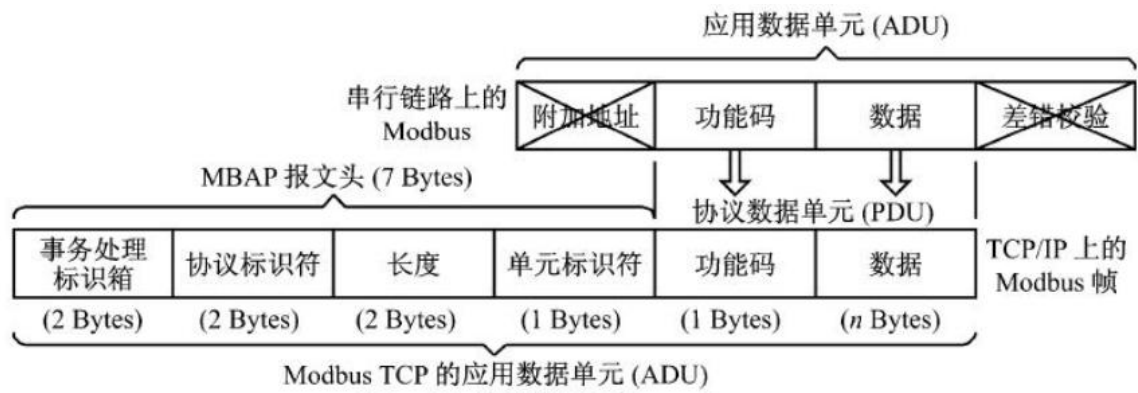


表 3-2 TCP 模式信息帧格式

事务处理标识	协议标识	长度	单元标识符	功能码	数据
2 个字节	2 个字节	2 个字节	1 个字节	1 个字节	0~253 个字节

Modbus TCP 协议数据帧由 MBAP（Modbus Application Protocol Head）和 PDU（Protocol Data Unit）两部分构成。

MBAP 报文头：事务处理标识 + 协议标识 + 长度 + 单元标识符；

PDU：功能码 + 数据；

表 3-3 MBAP 报文头说明

MBAP 报文头	事务处理标识	协议标识	长度	单元标识符
	2 个字节	2 个字节	2 个字节	1 个字节

(1) 事务处理标识

该标识可以理解为报文的序列号，一般每次通信之后就要加 1，用于区分不同的通信数据报文。

从站（服务器端）应答时从接收的请求中复制该值。

(2) 协议标识符

Modbus TCP 协议时，该值固定为 0x0000;

### (3) 长度

用于表示后续字节数，即从单元标识符开始后面数据的长度（单元标识符+功能码+数据的总字节数），后续字节必须在 256 字节以内。

### (4) 单元标识符

串口链路或其它总线上的远程终端标识，可以理解为设备地址。

从站（服务器端）应答时，从接收的请求中复制该值。

**注：**Modbus 设备可分为主站和从站。主站只有一个，从站有多个，主站向各从站发送请求帧，从站给予响应。在使用 TCP 通信时，主站为 client 端，主动建立连接；从站为 server 端，等待连接。



### 3.3. 传感器支持的功能码

本传感器支持的功能码，如表 3-4 所示：

表 3-4 支持的功能码列表

功能代码	功能名	操作
03H	读保持寄存器	读取 1 个或多个保持寄存器的内容
04H	读输入寄存器	读取 1 个或多个输入寄存器的内容
06H	写单个寄存器	向 1 个保持寄存器写入值

在 Modbus TCP 中，单元标识符号即为从机地址，在不同设备之间通过 IP 地址进行区别时，可默认为 1 且 Modbus TCP 无 CRC 校验。

#### 3.3.1. 功能码 03(0x03)

功能码 03 的请求帧数据格式（主机→从机）如表 3-5 所示：

表 3-5 请求帧数据格式（功能码 03）

字节序号	0	1	2	3	4	5	6	7
数据内容	从机地址	功能码	起始地址高字节	起始地址低字节	寄存器数量高字节	寄存器数量低字节	CRC 校验低字节	CRC 校验高字节
取值范围	1~247	03	0 ~ 65535		1~125		(Modbus TCP 无 CRC 校验)	(Modbus TCP 无 CRC 校验)

功能码 03 的响应帧数据格式（从机→主机）如表 3-6 所示：

表 3-6 响应帧数据格式（功能码 03）

字节序号	0	1	2	3	4	.....
------	---	---	---	---	---	-------

数据内容	从机地址	功能码	字节数	寄存器值 1 低字节	寄存器值 1 高字节	.....
取值范围	1~247	03	2*N (N 为读取的寄存器个数)			

字节序号	2N+1	2N+2	2N+3	2N+4
数据内容	寄存器值 N 低字节	寄存器值 N 高字节	CRC 校验 低字节	CRC 校验 高字节
取值范围			(Modbus TCP 无 CRC 校验)	(Modbus TCP 无 CRC 校验)

### 3.3.2. 功能码 04(0x04)

功能码 04 的请求帧数据格式（主机→从机）如表 3-7 所示：

表 3-7 请求帧数据格式（功能码 04）

字节序号	0	1	2	3	4	5	6	7
数据内容	从机地址	功能码	起始地址 高字节	起始地址 低字节	寄存器数量 高字节	寄存器数量 低字节	CRC 校验 低字节	CRC 校验 高字节
取值范围	1~247	04	0 ~ 65535		1~125		(Modbus TCP 无 CRC 校验)	(Modbus TCP 无 CRC 校验)

功能码 04 的请求帧数据格式（从机→主机）如表 3-8 所示：

表 3-8 响应帧数据格式（功能码 04）

字节序号	0	1	2	3	4	.....
数据内容	从机地址	功能码	字节数	寄存器值 1 低字节	寄存器值 1 高字节	.....
取值范围	1~247	04	2*N (N 为读取的寄存器个数)			

字节序号	2N+1	2N+2	2N+3	2N+4
数据内容	寄存器值 N 低字节	寄存器值 N 高字节	CRC 校验 低字节	CRC 校验 高字节
取值范围			(Modbus TCP 无 CRC 校验)	(Modbus TCP 无 CRC 校验)

### 3.3.3. 功能码 06(0x06)

功能码 06 的请求帧数据格式（主机→从机）如表 3-9 所示：

表 3-9 请求帧数据格式（功能码 06）

序号	0	1	2	3	4	5	6	7
数据内容	从机地址	功能码	寄存器地址 高字节	寄存器地址 低字节	寄存器值 高字节	寄存器值 低字节	CRC 校验 低字节	CRC 校验 高字节
取值	1~247	06					(Modbus TCP 无 CRC 校验)	(Modbus TCP 无 CRC 校验)

功能码 06 的响应帧数据格式（从机→主机）如表 3-10 所示：

表 3-10 响应帧数据格式（功能码 06）

序号	0	1	2	3	4	5	6	7
数据内容	从机地址	功能码	寄存器地址高字节	寄存器地址低字节	寄存器值高字节	寄存器值低字节	CRC 校验低字节	CRC 校验高字节
取值	1~247	06					（Modbus TCP 无 CRC 校验）	（Modbus TCP 无 CRC 校验）

3.4. 错误响应帧格式

当从机收到主机发送的信息帧时，先进行 CRC 校验，如 CRC 校验错误，则忽略该帧；

如从机检测到了 CRC 校验以外的错误，则向主机返回错误信息帧。错误信息帧中返回的功能码，是将主机发送的功能码最高位置 1 而来，相当于在主机发送的功能码基础上加 0x80。

**举例：**主机发送的功能码为 0x03，则对应的错误功能码为 0x83；

表 3-11 错误响应帧格式

字节序号	0	1	2	3	4
数据内容	地址码	功能码	错误码	CRC 校验低字节	CRC 校验高字节
				（Modbus TCP 无 CRC 校验）	（Modbus TCP 无 CRC 校验）

表 3-12 错误码说明

错误码	错误码含义	详细说明
01	非法的功能码	接收到的功能码不支持
02	非法的数据地址	指定的地址超出从机的范围
03	非法的数据值	接收到主机发送的数据值超出相应地址的数据范围

## 4. Modbus 寄存器分配

### 4.1. 参数设置类寄存器

参数存放在保持寄存器（Holding Register）中。

参数设置与读取，通过功能码“03”、功能码“06”操作保持寄存器实现。

#### 4.1.1. 设置寄存器列表

表 4-1 保持寄存器列表（对应功能码 03H、06H）

寄存器地址 (10 进制)	功能名称	数据类型	寄存器个数
0000	光源开关	2 字节 无符号	1
0001	曝光模式	2 字节 无符号	1
0002	曝光时间	2 字节 无符号	1
0003	位置清零	2 字节 无符号	1
0004-0007	预留		4
0008	采样间隔	2 字节 无符号	1
0009	数据滑动平均 滤波次数	2 字节 无符号	1
0010	无效数据保持次 数设置	2 字节 无符号	1
0011	编码器 1 计数使能	2 字节 无符号	1

0012	编码器 2 计数使能	2 字节 无符号	1
0013	保存参数	2 字节 无符号	1
0014	触发计数复位	2 字节 无符号	1
0015-0016	厚度 1 修正系数	2 字节 无符号	1
0017	缓存数据上报数 量	2 字节 无符号	1
0018	缓存上锁标志	2 字节 无符号	1
0019	缓存数据类型	2 字节 无符号	1
0020-0021	厚度 2 修正系数	2 字节 无符号	1
0022-0023	厚度 3 修正系数	2 字节 无符号	1
0024-0025	厚度 4 修正系数	2 字节 无符号	1
0026-0027	厚度 5 修正系数	2 字节 无符号	1
0028	厚度 1 对应折射 率表编号	2 字节 无符号	1
0029	厚度 2 对应折射 率表编号	2 字节 无符号	1
0030	厚度 3 对应折射 率表编号	2 字节 无符号	1

0031	厚度 4 对应折射率表编号	2 字节 无符号	1
0032	厚度 5 对应折射率表编号	2 字节 无符号	1

注：

- (1) 传感器定义的保持寄存器最大数量为 60 个；
- (2) 寄存器写入无效值时，原寄存器的值不发生改变；

#### 4.1.2. 寄存器说明

表 4-2 寄存器说明列表

寄存器地址 (10 进制)	功能名称	取值范围	数据类型	读/写	寄存器 个数
0000	光源开关	0: 关闭 1: 打开	2 字节 无符号	读/写	1
0001	曝光模式	0: 手动 1: 自动	2 字节 无符号	读/写	1
0002	曝光时间	0~50000 单位: 0.1us	2 字节 无符号	读/写	1
0003	位置清零	0: 取消清零; 1: 位置清零, 将距离 1 此刻 位置置为零 点。	2 字节 无符号	读/写	1
0004-0007	预留				4
0008	采样间隔	0: 250us 1: 500us 2: 1ms 3: 2ms 4: 5ms 5: 10ms 6: 100us 7: 125us	2 字节 无符号	读/写	1



		8: 160us 9: 200us			
0009	数据滑动平均滤波次数	0x00: 4 次 0x01: 16 次 0x02: 64 次 0x03: 256 次 0x04: 1024 次 0x05: 4096 次 0x06: 1 次 0x07: 2 次	2 字节 无符号	读/写	1
0010	无效数据保持次数设置	0 ~ 1000	2 字节 无符号	读/写	1
0011	编码器 1 计数使能	0: 关闭 1: 打开	2 字节 无符号	读/写	1
0012	编码器 2 计数使能	0: 关闭 1: 打开	2 字节 无符号	读/写	1
0013	保存参数	1: 当前配置参数写入 flash, 写完后该标志自动清 0; 其它值无效; 注: 写 flash 耗时大约 400ms, 该过程中读写数据会造成超时。	2 字节 无符号	读/写	1
0014	触发计数复位	1: 对当前触发计数进行复位, 复位后该值会自动清 0; 其它值无效。	2 字节 无符号	读/写	1
0015-0016	厚度修正系数	厚度 1 的修正系数	2 字节 无符号	读/写	1
0017	缓存数据上	缓存数据类型	2 字节	读/写	1

	报数量	的整数倍	无符号		
0018	缓存上锁标志	给缓存上锁, 0: 解锁 1: 锁定	2 字节 无符号	读/写	1
0019	缓存数据类型	数据类型选择为一个 16 位的值, 其中 bit0 为 MATH, bit1-bit3 预留、 bit4-bit5 为通道 1 距离 1、 通道 1 距离 2、 bit6 为通道 1 厚度 1、bit7-bit10 为通道 1 距离 3、通道 1 距离 4、通道 1 距离 5、通道 1 距离 6、 bit11-bit14 为通道 1 厚度 2、 通道 1 厚度 3、 通道 1 厚度 4、 通道 1 厚度 5)	2 字节 无符号	读/写	1
0020-0021	厚度 2 修正系数	厚度 2 修正系数	2 字节 无符号	读/写	1
0022-0023	厚度 3 修正系数	厚度 3 修正系数	2 字节 无符号	读/写	1
0024-0025	厚度 4 修正系数	厚度 4 修正系数	2 字节 无符号	读/写	1
0026-0027	厚度 5 修正系数	厚度 5 修正系数	2 字节 无符号	读/写	1
0028	厚度 1 对应折射率编号	厚度 1 的对应折射率编号	2 字节 无符号	读/写	1
0029	厚度 2 对应	厚度 2 的对应	2 字节	读/写	1

	折射率编号	折射率编号	无符号		
0030	厚度 3 对应 折射率编号	厚度 3 的对应 折射率编号	2 字节 无符号	读/写	1
0031	厚度 4 对应 折射率编号	厚度 4 的对应 折射率编号	2 字节 无符号	读/写	1
0032	厚度 5 对应 折射率编号	厚度 5 的对应 折射率编号	2 字节 无符号	读/写	1

注：保存参数时，位置清零的参数不存储。

## 4.2. 数据读取类寄存器

测量数据存放在输入寄存器中（Input Register）。

通过功能码“04”操作输入寄存器，实现测量数据的读取操作。

### 4.2.1. 寄存器列表

表 4-3 输入寄存器列表（对应功能码 04H）

寄存器地址 (10 进制)	功能名称	单位	数据类型	寄存器数	备注
0000-0001	距离 1	mm	4 字节 有符号	2	数值扩大 1000000 倍
0002-0003	距离 2	mm	4 字节 有符号	2	数值扩大 1000000 倍
0004-0005	厚度	mm	4 字节 有符号	2	数值扩大 1000000 倍
0006-0007	编码器 1 脉冲计数	个	4 字节 无符号	2	
0008-0009	编码器 2 脉冲计数	个	4 字节 无符号	2	
0010	峰 1 高度	无量纲	2 字节 无符号	1	

0011	峰 2 高度	无量纲	2 字节 无符号	1	
0012	单层测厚缓存 数据已读个数	个	2 字节 无符号	1	
0013	本次读取缓存 数据寄存器有 效数据个数	个	2 字节 无符号	1	
0014-0015	缓存数据存放 首个数据	无量纲	2 字节 无符号	2	
0016-0017	距离 3	mm	4 字节 有符号	2	数值扩大 1000000 倍
0018-0019	距离 4	mm	4 字节 有符号	2	数值扩大 1000000 倍
0020-0021	距离 5	mm	4 字节 有符号	2	数值扩大 1000000 倍
0022-0023	距离 6	mm	4 字节 有符号	2	数值扩大 1000000 倍
0024-0025	厚度 2	mm	4 字节 有符号	2	数值扩大 1000000 倍
0026-0027	厚度 3	mm	4 字节 有符号	2	数值扩大 1000000 倍
0028-0029	厚度 4	mm	4 字节 有符号	2	数值扩大 1000000 倍
0030-0031	厚度 5	mm	4 字节 有符号	2	数值扩大 1000000 倍
0032	峰 3 高度	无量纲	2 字节 无符号	1	

0033	峰 4 高度	无量纲	2 字节 无符号	1	
0034	峰 5 高度	无量纲	2 字节 无符号	1	
0035	峰 6 高度	无量纲	2 字节 无符号	1	
0090-0091	控制器温度	2 字节 无符号	只读	1	数值扩大 1000000 倍
0120	多层测厚缓存 数据已读个数	个	2 字节 无符号	1	
0121	本次读取缓存 数据寄存器有 效数据个数	个	2 字节 无符号	1	
0122-0123	缓存数据存放 首个数据	无量纲	2 字节 无符号	2	

#### 4.2.2. 寄存器说明

表 4-4 输入寄存器说明

寄存器地址 (10 进制)	功能名称	数据类型	读/写	寄存器个数
0000	距离 1 高 16 位	2 字节 有符号	只读	1
0001	距离 1 低 16 位	2 字节 有符号	只读	1
0002	距离 2 高 16 位	2 字节 有符号	只读	1

0003	距离 2 低 16 位	2 字节 有符号	只读	1
0004	厚度 高 16 位	2 字节 有符号	只读	1
0005	厚度 低 16 位	2 字节 有符号	只读	1
0006	编码器 1 脉冲计数 高 16 位	2 字节 无符号	只读	1
0007	编码器 1 脉冲计数 低 16 位	2 字节 无符号	只读	1
0008	编码器 2 脉冲计数 高 16 位	2 字节 无符号	只读	1
0009	编码器 2 脉冲计数 低 16 位	2 字节 无符号	只读	1
0010	峰 1 高度	2 字节 无符号	只读	1
0011	峰 2 高度	2 字节 无符号	只读	1
0012	单层测厚缓存数据 已读个数	2 字节 无符号	只读	1
0013	本次读取缓存数据 寄存器有效数据个 数	2 字节 无符号	只读	1
0014-0015	缓存数据存放首个 数据	2 字节 无符号	只读	1
0016	距离 3 高 16 位	2 字节 有符号	只读	1

0017	距离 3 低 16 位	2 字节 有符号	只读	1
0018	距离 4 高 16 位	2 字节 有符号	只读	1
0019	距离 4 低 16 位	2 字节 有符号	只读	1
0020	距离 5 高 16 位	2 字节 有符号	只读	1
0021	距离 5 低 16 位	2 字节 有符号	只读	1
0022	距离 6 高 16 位	2 字节 有符号	只读	1
0023	距离 6 低 16 位	2 字节 有符号	只读	1
0024	厚度 2 高 16 位	2 字节 有符号	只读	1
0025	厚度 2 低 16 位	2 字节 有符号	只读	1
0026	厚度 3 高 16 位	2 字节 有符号	只读	1
0027	厚度 3 低 16 位	2 字节 有符号	只读	1
0028	厚度 4 高 16 位	2 字节 有符号	只读	1
0029	厚度 4 低 16 位	2 字节 有符号	只读	1
0030	厚度 5 高 16 位	2 字节 有符号	只读	1

0031	厚度 5 低 16 位	2 字节 有符号	只读	1
0032	峰 3 高度	2 字节 有符号	只读	1
0033	峰 4 高度	2 字节 有符号	只读	1
0034	峰 5 高度	2 字节 有符号	只读	1
0035	峰 6 高度	2 字节 有符号	只读	1
0090-0091	控制器温度	2 字节 无符号	只读	1
0120	多层测厚缓存数据 已读个数	2 字节 无符号	只读	1
0121	本次读取缓存数据 寄存器有效数据个 数	2 字节 无符号	只读	1
0122-0123	缓存数据存放首个 数据	2 字节 无符号	只读	1

注：传感器定义的输入寄存器数量为 40 个，超出范围将会报错。



## 5. 应用示例

### 5.1. 读取参数示例（功能码 03）

#### 5.1.1. ModbusRTU 格式数据解析

**举例：**传感器的从机地址为“1”,读取传感器通道 1 的光源开关、曝光模式、曝光时间等参数。

**分析：**读取传感器参数，需使用“功能码 03”，读取保持寄存器的值。通道 1 光源开关、通道 1 曝光模式、通道 1 曝光时间对应的寄存器地址分别为 0000、0001、0002，因此可通过一次性读取 3 个寄存器的方式读取参数值。即用功能码 03，读保持寄存器起始地址为 0000，读取寄存器个数为 3 个。

**主机发送请求报文：**01 03 00 00 00 03 05 CB

表 5-1 读取参数请求报文解析（RTU）

主机发送报文	占字节数	数据内容（16 进制）
从机地址	1	01
功能码	1	03
寄存器起始地址	2	00 00
读取寄存器个数	2	00 03
CRC 校验	2	05 CB

**从机发送响应报文：**01 03 06 00 01 00 00 0F A0 19 3D

表 5-2 读取参数响应报文解析（RTU）

从机响应报文	占字节数	数据内容（16 进制）
从机地址	1	01
功能码	1	03
字节数	1	06
寄存器 0000 的数据	2	00

		01
寄存器 0001 的数据	2	00 00
寄存器 0002 的数据	2	0F A0
CRC 校验	2	19 3D

数值分析：光源开关打开、曝光模式为手动、曝光时间为 400us（曝光时间单位为 0.1us）。

### 5.1.2. ModbusTCP 格式数据解析

主机发送请求报文：00 00 00 00 00 06 01 03 00 00 00 03

表 5-3 读取参数请求报文解析（TCP）

主机发送报文	占字节数	数据内容（16 进制）
事务处理标识	2	00 00
协议标识	2	00 00
后续字节长度	2	00 06
单元标识符	1	01
功能码	1	03
寄存器起始地址	2	00 00
读取寄存器个数	2	00 03

传感器发送的响应报文：00 00 00 00 00 09 01 03 06 00 01 00 00 0F A0

表 5-4 读取参数响应报文解析（TCP）

从机响应报文	占字节数	数据内容（16 进制）
事务处理标识	2	00 00

协议标识	2	00 00
后续字节长度	2	00 09
单元标识符	1	01
功能码	1	03
数据长度	1	06
寄存器 0000 的数据	2	00 01
寄存器 0001 的数据	2	00 00
寄存器 0002 的数据	2	0F A0

**数值分析：**光源开关打开、曝光模式为手动、曝光时间为 400us。（曝光时间单位为 0.1us）。

## 5.2. 设置参数示例（功能码 06）

**举例：**传感器的从机地址为“1”，设置曝光时间为 204.7us。

**分析：**设置曝光时间，需使用“功能码 06”，往地址为 0002 的保持寄存器写入值，204.7us 对应的寄存器值为 2047（0x07ff），即向地址为“0002”的保持寄存器，写入 0x07ff 数值。

### 5.2.1. ModbusRTU 格式数据解析

主机发送请求报文：01 06 00 02 07 FF 6A 7A

表 5-5 设置参数请求报文解析 (RTU)

主机发送报文	占字节数	数据内容（16 进制）
从机地址	1	01
功能码	1	06
寄存器地址	2	00 02

寄存器值	2	07 FF
CRC 校验	2	6A 7A

从机发送响应报文：01 06 00 02 07 FF 6A 7A

表 5-6 设置参数响应报文解析 (RTU)

从机响应报文	占字节数	数据内容 (16 进制)
从机地址	1	01
功能码	1	06
寄存器地址	2	00 02
寄存器值	2	07 FF
CRC 校验	2	6A 7A

### 5.2.2. ModbusTCP 格式数据解析

主机发送请求报文：00 00 00 00 00 06 01 06 00 02 07 FF

表 5-7 设置参数请求报文解析 (TCP)

主机发送报文	占字节数	数据内容 (16 进制)
事务处理标识	2	00 00
协议标识	2	00 00
后续字节长度	2	00 06
单元标识符	1	01
功能码	1	06
寄存器起始地址	2	00 02
寄存器值	2	07 FF

传感器发送的响应报文：00 00 00 00 00 06 01 06 00 02 07 FF

表 5-8 设置参数响应报文解析(TCP)

从机响应报文	占字节数	数据内容（16 进制）
事务处理标识	2	00 00
协议标识	2	00 00
后续字节长度	2	00 06
单元标识符	1	01
功能码	1	06
寄存器起始地址	2	00 02
寄存器值	2	07 FF

### 5.3. 读取数据示例（功能码 04）

**举例：**传感器的从机地址为“1”,读取该传感器通道 1 测量数据距离 1。

**分析：**读取通道 1 的距离 1，需使用“功能码 04”，读取输入寄存器地址 0x0000-0x0001 的数值，读取的寄存器个数为 2 个。

#### 5.3.1. RTU 格式数据解析

主机发送的请求报文：01 04 00 00 00 02 71 CB

表 5-9 读取数据请求报文解析

主机发送报文	占字节数	数据内容（16 进制）
从机地址	1	01
功能码	1	04
寄存器起始地址	2	00 00
读取寄存器个数	2	00 02

CRC 校验	2	71 CB
--------	---	----------

传感器发送的响应报文：01 04 04 00 03 E9 4C D2 44

表 5-10 读取数据响应报文解析

从机响应报文	占字节数	数据内容（16 进制）
从机地址	1	01
功能码	1	04
字节数	1	04
寄存器 0000 的数据	2	00 03
寄存器 0001 的数据	2	E9 4C
CRC 校验	2	45 1E

### 5.3.2. TCP 格式数据解析

主机发送请求报文：00 00 00 00 00 06 01 04 00 00 00 02

表 5-11 读取参数请求报文解析（TCP）

主机发送报文	占字节数	数据内容（16 进制）
事务处理标识	2	00 00
协议标识	2	00 00
后续字节长度	2	00 06
单元标识符	1	01
功能码	1	04
寄存器起始地址	2	00 00
读取寄存器个数	2	00 02

传感器发送的响应报文：00 00 00 00 00 07 01 04 04 00 03 E9 4C

表 5-12 读取参数响应报文解析(TCP)

从机响应报文	占字节数	数据内容（16 进制）
事务处理标识	2	00 00
协议标识	2	00 00
后续字节长度	2	00 07
单元标识符	1	01
功能码	1	04
数据长度	1	04
寄存器 0000 的数据	2	00 03
寄存器 0001 的数据	2	E9 4C

**数值分析：**

距离 1 的数值为 0x0003E94C，转换为有符号整数为 256332，由于实际值放大了 1000000 倍，所以距离 1 的测量值为 0.256332mm。

注：当读取的数据值为 0x80000000（16 进制）或-2147483648（有符号整数）时，则表示当前的测量值无效。

## 5.4. 读取缓存数据示例



图 5-4 开启缓存数据功能与读缓存数据操作

### 举例：

使用外部触发功能，单次触发采集 2000 组探头通道 1 的距离 1、距离 2 和厚度数据，距离 1、距离 2 和厚度按顺序输出，单组数据包含数据类型为 3，因此数据采集个数为 6000 个，数据采集结束后，通过 Modbus 协议将 6000 个数据读出。

### 分析：

- (1) 读取距离 1 的缓存数据，需将“缓存数据类型”寄存器（19）值设置为 112（即对应 bit4-6 位为 1）；
- (2) 缓存数据上报个数为 6000 个，需将“缓存数据上报个数”寄存器（17）值设置为 6000；
- (3) 读缓存数据前，需将“缓存区上锁标志”寄存器（18）值设置为 0；此时如果是通过外部触发进行采样，外部电路应进行触发操作
- (4) 稍等一段时间，将寄存器（18）值设置为 1，将已缓存的数据锁住，供后续读取
- (5) 由于 modbus 一帧最大只能读取 60 个缓存数据，即 120 个寄存器。为防止漏读数据，直接从“缓存数据已读个数”寄存器开始读取，连续读取 122 个寄存器：已读个数（1 个）+有效个数（1 个）+缓存数据（60\*2



个)。

(6) 6000 个缓存数据需要读取 100 次才能读完，每次固定读取 60 个缓存数据。

(7) 读完 6000 个缓存数据后，将“缓存区上锁标志”寄存器（18）置 0。

**操作步骤：**

➤ 开启缓存数据功能操作：

- ① “缓存数据类型”设置为探头通道 1 的距离 1、距离 2 和厚度；
- ② “缓存数据上报个数”设置为 6000；
- ③ “缓存区上锁标志”设置为 0。

➤ 缓存数据读取操作：

- ① “缓存区上锁标志”设置为 1；
- ② 从“缓存数据已读个数”寄存器开始读取 122 个寄存器，连续读 100 次。

➤ 缓存数据读完后操作：

“缓存区上锁标志”设置为 0。

## 6. 附录

### 附录 1 Modbus RTU CRC 算法代码

```
static const unsigned char aucCRCHi[] = {  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,  
    0x00, 0xC1, 0x81, 0x40  
};
```

```

static const unsigned char aucCRCLo[] = {
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,
    0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E,
    0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9,
    0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
    0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
    0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D,
    0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
    0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF,
    0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
    0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
    0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB,
    0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA,
    0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
    0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97,
    0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E,
    0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89,
    0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
    0x41, 0x81, 0x80, 0x40
};

```

```

unsigned short MB_CRC16(unsigned char * pucFrame, unsigned short usLen )
{
    unsigned char    crc_h = 0xFF;
    unsigned char    crc_l = 0xFF;
    unsigned short   index, crc_value;

    while( usLen-- )
    {
        index = crc_l ^ *( pucFrame++ );

        crc_l = (unsigned char)(crc_h ^ aucCRCHi[index]);

        crc_h = aucCRCLo[index];
    }

    crc_value = crc_h;

    crc_value <<= 8;

    crc_value |= crc_l;

    return crc_value;
}

```

## 附录 2 Modbus 通信参数设置

Modbus 通信参数设置包括从机地址、波特率设置。参数设置步骤如下所示：

**步骤 1：** 打开上位机软件，与传感器成功连接；

**步骤 2：** 进入 Modbus 配置界面

在上位机软件主窗口，点击菜单栏参数配置——>测量配置——>通信配置——>Modbus 配置，如图 6-1 所示。

**步骤 3：** 设置从机地址

在从机地址设置框里选择输入设置的地址，取值范围 1-247；

**步骤 4：** 设置 Modbus 通信波特率

在波特率配置设置下拉框里选择要设置的波特率。

图 6-1 Modbus 通信参数配置