PMC - 3601F

Distributed Optical Fiber Temperature Sensor

Modbus Communication Protocol

(V1.1)

Table of Contents

Chapter 1 Introduction

This protocol describes in detail the input and output commands, information, and data of the PMC-3601F device in Modbus communication mode, to facilitate development and use by third parties.

This protocol enables the effective transmission of information and data between the host computer master station and the PMC-3601F slave station (hereinafter referred to as the slave station), including:

- Allowing the master station to access all measurement data collected by the PMC-3601F.

Chapter 2 Detailed Description of Modbus Serial Communication Protocol

2.1 Basic Protocol Rules

The Modbus protocol is widely used in RS-485 communication networks, which can support multiple slave stations:

a) All communication follows the master/slave mode, where information and data are transmitted between a single Modbus master station and up to 32 slave monitoring devices;
b) The master station initiates and controls all information transmitted on the RS485 communication loop;
c) Slave stations cannot initiate communication;
d) All information is transmitted in the form of "data packets," consisting of a string of bytes (8 bits per byte), with a maximum of 255 bytes per packet;
e) A data packet sent by the master station is called a request, and one sent by the slave station is called a response;
f) Only one slave station can respond to a master station request at any time.

2.2 Transmission Modes

The Modbus protocol supports ASCII and RTU modes. This protocol uses Modbus-RTU mode with:

- Parity check mode: 1 start bit + 8 data bits + 1 parity bit + 1 stop bit.
- No parity check mode: 1 start bit + 8 data bits + 2 stop bits.

The device supports all three transmission modes.

2.3 Modbus Data Packet Structure Description

Each Modbus packet consists of:

- Address field
- Function code field
- Data field
- Checksum field

2.3.1 Address Field

The slave address field (1 byte) ranges from 1 to 247. Address 0 is reserved for broadcast. A slave executes a packet's command if the address matches its own, and responds with its address in the address field.

2.3.2 Function Code Field

The function code field (1 byte) instructs the slave on the operation to perform. The slave's response includes the same function code. Supported function codes:

Function Code	Description	Function
0x03	Read registers	Tele-signaling, Telemetering, SOE, etc.
0x10	Set one or more registers	Time synchronization
0x05	Set coil	Tele-control relay

2.3.3 Data Field

The data field length varies by function and uses "BIG ENDIAN" format (high-byte first). For example, a register value of 0x12AB is sent as 0x12 (high) followed by 0xAB (low).

This protocol supports uploading data types such as char (character type), uint16 (unsigned 16-bit short integer), int16 (signed 16-bit short integer), uint32 (unsigned 32-bit long integer), int32 (signed 32-bit long integer), float (32-bit floating-point number), int64 (signed 64-bit long integer), and status words. The specific formats for uploading data are as follows:

- char: 1 byte (byte0). If 1 char occupies 1 register: 0x00, byte0; if 2 chars share 1 register: char1, char2.
- uint16 and int16: A 16-bit short integer consists of two bytes: byte0 (bit0–bit7) and byte1 (bit8–bit15). The order of transmission for register data is: byte1, byte0.
- uint32 and int32: A 32-bit long integer consists of four bytes: byte0 (bit0–bit7), byte1 (bit8–bit15), byte2 (bit16–bit23), and byte3 (bit24–bit31). The order of transmission for register data is: byte3, byte2, byte1, byte0.
- float: Floating-point numbers are uploaded in the IEEE754 4-byte format, occupying two registers. For example, 2.66 is represented as 0x71, 0x3D, 0x2A, 0x40. These four bytes are uploaded using two registers, and the transmission order is: 0x40, 0x2A, 0x3D, 0x71.

- int64: A 64-bit long integer consists of eight bytes: byte0 (bit0–bit7), byte1 (bit8–bit15), byte2 (bit16–bit23), byte3 (bit24–bit31), byte4 (bit32–bit39), byte5 (bit40–bit47), byte6 (bit48–bit55), and byte7 (bit56–bit63). The order of transmission for register data is: byte7, byte6, byte5, byte4, byte3, byte2, byte1, byte0.
- Status word: Each bit has a defined meaning, sent high-byte first.

2.3.4 Checksum Field

Modbus-RTU uses 16-bit CRC with polynomial ($X^{16} + X^{15} + X^2 + 1$). The sending device should perform a CRC16 calculation on every Byte of the data packet and store the result in the Checksum Field. The receiving device should also perform a CRC16 calculation on every Byte of the data packet (except the Checksum Field) and compare the result with the Checksum Field. Specific data packets with matching results can be accepted. The specific CRC error checking algorithm is described in the appendix.

2.4 Network Time Settings

RS-485 transmission time rules:

- Master request to slave response: 20~250 ms (typical 60 ms).
- Slave response to next master request: 100 ms (typical).
- Inter-byte delay: ≤3 x byte transmission time (e.g., 3 ms @ 9600 baud, 6 ms @ 4800 baud).
- Data processing delay: ≥3.5 character lengths after the last byte, with subsequent data treated as a new frame.

2.5 Abnormal Responses

If the master station sends an illegal data packet to PMC-3601F or requests an invalid data register, an abnormal response will be generated. This abnormal response includes the slave address, function code, error code, and Checksum Field. When the high bit of the function code is set to 1, it indicates that the packet is an abnormal response. The following table explains the meaning of the error codes:

Error Code	Function Code	Description
0x01 (Illegal Function Code)	0x80+Original Function Code	Indicates that the slave station received an unsupported function code (excluding the function codes listed above).
0x02 (Illegal Data Address)	0x80+Original Function Code	Indicates that the device received an invalid data address or the requested register is outside the valid register range
0x03 (Illegal Data Value)	0x80+Original Function Code	The number of registers read or written exceeds the allowable range; The data written is out of range
0x04 (Operation Register Failed)	0x80+Original Function Code	Read or write operation failed.

2.6 Broadcast Commands

A broadcast command uses address 00H (valid only for function code 0x10). When the master station sends a broadcast command, the slave station only receive the data packet and do not return a response packet. This prevents all slave stations in the network from responding simultaneously, which could cause network congestion.

Chapter 3 Communication Data Packets

3.1 Read Registers (0x03)

This is a data packet request sent by the host and this device responds with all valid registers (between the starting register and the ending register). The PMC-3601F supports up to 125 registers per frame:

Read Register Data Packet Mode (Host → PMC-3601F)		Response Mode (PMC-3601F → Host)	
Slave Address	1 Byte	Slave Address	1 Byte
Function code	1 Byte	Function code	1 Byte
Register Starting Address High-order	1 Byte	No. of Byte (n)	1 Byte
Register Starting Address Low-order	1 Byte	Data1 High-order	1 Byte
Register Count High (Byte)	1 Byte	Data1 Low-order	1 Byte
Register Count Low (Byte)	1 Byte	……	
		Datan/2 High-order	1 Byte
		Datan/2 Low-order	1 Byte
CRC Checking Code Low-order	1 Byte	CRC Checking Code Low-order	1 Byte
CRC Checking Code High-order	1 Byte	CRC Checking Code Low -order	1 Byte

3.2 Write Registers (0x10)

This command allows the master station to set the operating parameters of this device. The data format is as follows:

Write Register Data Packet Mode (Host → PMC-3601F)		Response Mode (PMC-3601F → Host)	
Slave Address	1 Byte	Slave Address	1 Byte
Function code	1 Byte	Function code	1 Byte
Register Starting Address High-order	1 Byte	Register Starting Address High-order	1 Byte
Register Starting Address Low-order	1 Byte	Register Starting Address Low-order	1 Byte
Register Count High (Byte)	1 Byte	Register Count High (Byte)	1 Byte
Register Count Low (Byte)	1 Byte	Register Count Low (Byte)	1 Byte
No. of Byte (n)	1 Byte	CRC Checking Code Low-order	1 Byte
Data1 High	1 Byte	CRC Checking Code High-order	1 Byte
Data1 Low	1 Byte		
……			
Data(n/2) High	1 Byte		
Data(n/2) Low	1 Byte		
CRC Checking Code Low-order	1 Byte		
CRC Checking Code High-order	1 Byte		

Chapter 4 Ethernet Communication Instructions

The Ethernet port uses Modbus TCP protocol. The rules for Modbus TCP communication are as follows:

- Default port: 502 (PMC-3601F as server), supporting 5 simultaneous connections.
- Command format: MBAP header + Function code + Data field (The function code and data field are identical to those used in Modbus serial communication; Modbus CRC is not included).

MBAP header format:

Field	Length	Description
Transaction Identifier	2 Bytes	Identifies Modbus requests/responses.
Protocol Identifier	2 Bytes	0 indicates the Modbus protocol
Length	2 Bytes	Bytes from unit identifier to command end.
Unit identifier	1 Byte	Remote slave identifier (recommended 0xff in practice).

Note:

1. The unit identifier indicates the channel number (value range 1–4) in the real-time data registers, channel basic parameter registers, channel insertion point parameter registers, and channel connection point parameter registers. In other registers, the unit identifier has no practical significance. It is recommended that the host computer use a value of 0xff when communicating with the device. For example:
   - To read the number of valid temperature data points for Channel 1 in the real-time data, the request message is: 0000 0000 0006 01 03 0000 0001.
   - To read the alarm status of Zone 1_Segment 1 in the zone-segment alarm parameters, the request message is: 0000 0000 0006 FF 03 7918 0001.
2. Modbus TCP allows reading up to 4096 registers per request.

Chapter 5 Serial Port Communication Instructions

Serial port uses Modbus RTU. When reading fixed forwarding tables, note the ID format:

- High 5 bits: Actual device ID
- Low 3 bits: Number of Channel to read

Chapter 6 PMC-3601F Registers Description

In this protocol, the register address in the table corresponds directly to the address used by the master in the request. For example, register address 10 maps to address 10 in the data packet sent by the master. See the detailed register table below.
When communicating via TCP, the ID range in the request message is 1–4, corresponding to fiber channels 1–4. When communicating via serial port, the upper 5 bits of the ID in the request message represent the actual device ID (range 1–31), and the lower 3 bits indicate the fiber channel to be read, representing fiber channels 1–4. For serial port 1, the default IDs are 9–12, corresponding to fiber channels 1–4; for serial port 2, the default IDs are 17–21, corresponding to fiber channels 1–4.

6.1 Device Data Registers

6.1.1 Real-time Data Registers

Register Address	Type	Description	Format	Unit	Note
0	RO	Valid Temperature Data Count [1]	INT16		
1	RO	Temperature Point 1 [2]	INT16	℃/×0.1	
2	RO	Temperature Point 2	INT16		
……					
30000	RO	Temperature Point 30000	INT16		

Notes:

1. The real-time data registers display 30,000 data points for a single channel; different data can be retrieved by setting different IDs. The number of valid temperature data points indicates the count of valid data entries for that fiber temperature measurement channel.
2. The temperature points are ordered by their distance from the fiber temperature measurement host. Each point represents a physical distance of 0.5 m. For example, temperature point 1 is the first point from the host, at a physical distance of 0.5 m, reflecting the average temperature over the 0–0.5 m fiber segment; temperature point 2 is at 1.0 m, reflecting the average temperature over the 0.5–1.0 m fiber segment, and so on.

6.1.2 Zone/Segment Alarm Registers

Register Address	Type	Description	Format	Note
31000	RO	Zone 1_Segment 1 Alarm Status [1]	UINT16_t	
31001	RO	Zone 1_Segment 2 Alarm Status	UINT16_t	
...	...	...	...	
31099	RO	Zone 1_Segment 100 Alarm Status	UINT16_t	
31100~31199	RO	Zone 2_Segment 1 ~ 100 Alarm Status	UINT16_t	
31200~31299	RO	Zone 3_Segment 1 ~ 100 Alarm Status	UINT16_t	
...	...	...	...	
35900~35999	RO	Zone 50_Segment 1 ~ 100 Alarm Status	UINT16_t	

Notes:

1. The Zone/Segment Alarm Status Bit Register records 5,000 segments. The system consists of 50 zones, each zone containing 100 segments, arranged sequentially. Each segment is represented by a uint16 value indicating its alarm status. The reading of zone/segment alarm status is independent of the request message ID — it refers to the configuration of the entire system. One uint16_t contains the status of 4 alarm models. The status of each alarm model is represented by 2 bits, indicating 4 states: Normal, Warning, Alarm, and Pre-Alarm. The status of each alarm model is indicated as follows:

Bit1	Bit0	Status
0	0	Normal
0	1	Warning
1	0	Alarm
1	1	Pre-Alarm

Bit mapping in an alarm status register:

<table>
<tr><th colspan="8">Bit8~Bit15</th></tr>
<tr><td colspan="8">(Unused)</td></tr>
<tr><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td colspan="2">Over Temperature Difference Setpoint</td><td colspan="2">Over Temperature Rise Rate Setpoint</td><td colspan="2">Under Temperature Setpoint</td><td colspan="2">Over Temperature Setpoint</td></tr>
</table>

6.1.3 Channel Abnormal Point Alarm Registers

Register Address	Type	Description	Format	Note
37000	RO	Channel 1 Disconnection Status	UINT16_t	
37001	RO	Channel 1 Total Abnormal Temperature Status [1]	UINT16_t	
37002	RO	Channel 3 Disconnection Status	UINT16_t	
37003	RO	Channel 2 Total Abnormal Temperature Status	UINT16_t	
37004	RO	Channel 3 Disconnection Status	UINT16_t	
37005	RO	Channel 3 Total Abnormal Temperature Status	UINT16_t	
37006	RO	Channel 4 Disconnection Status	UINT16_t	
37007	RO	Channel 4 Total Abnormal Temperature Status	UINT16_t	

Notes:

1. The Channel Abnormal Point Alarm Registers are independent of the request message ID. Disconnection status: 0 = Normal, 1 = Disconnected. Abnormal temperature status (2 bits):

Bit1	Bit0	Status
0	0	Normal
0	1	Warning
1	0	Alarm

Bit mapping in an alarm status register:

<table>
<tr><th colspan="3">Bit8~Bit15</th></tr>
<tr><td colspan="3">Reserved</td></tr>
<tr><th>Bit2~Bit7</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>Reserved</td><td colspan="2">Disconnection/Abnormal Temperature Point</td></tr>
</table>

6.1.4 Real-time Optical Path Operation Registers

Register Address	Type	Description	Format	Note
38000	RO	1550nm Laser Current	Float	mA
38002	RO	1550nm Laser Cooling Current	Float	mA
38004	RO	1550nm Laser Operating Temperature	Float	℃
38006	RO	980nm Laser Current	Float	mA
38008	RO	980nm Laser Cooling Current	Float	mA
38010	RO	980nm Laser Operating Temperature	Float	℃
38012	RO	APD Voltage	Float	V
38014	RO	APD Cooling Current	Float	mA
38016	RO	APD Operating Temperature	Float	℃
38018	RO	Channel 1 Reference Temperature	INT16	×0.1℃
38019	RO	Channel 2 Reference Temperature	INT16	×0.1℃
38020	RO	Channel 3 Reference Temperature	INT16	×0.1℃
38021	RO	Channel 4 Reference Temperature	INT16	×0.1℃

6.2 SOE Data Registers

6.2.1 SOE Pointer Count Register

Register Address	Type	Description	Format	Note
40000	RO	SOE Pointer Count [1]	UINT32	

6.2.2 SOE Operation Register

Register Address	Type	Description	Format	Note
40002	WO	Set Read Start Index	UINT32	

Note:

1. Range: 1–5000. Writing index 1 loads the latest 100 entries (≤100 if fewer). Only the latest 5000 entries are stored.

6.2.3 SOE Information Registers

Register Address	Type	Description	Format	Note
40010~40024	RO	SOE1 [1]	SOE_LOG	
40025~40039	RO	SOE2	SOE_LOG	
...	...	...		
41495~41509	RO	SOE100	SOE_LOG	

Note:

1. SOE_LOG structure:

Offset	Meaning
+0	Hi: Classification (1=Alarm, 2=Clear SOE) Lo: Sub-Classification (0=Over Temperature Setpoint, 1=Under Temperature Setpoint, 2= Over Temperature Rise Rate Setpoint, 3=Over Temperature Difference Setpoint, 4=Over Abnormal Temperature Point Setpoint, 5=Disconnection)
+1	Hi: Year (-2000), Lo: Month
+2	Hi: Day, Lo: Hour
+3	Hi: Minute, Lo: Second
+4	Millisecond
+5	Zone Index
+6	Segment Index
+7	Channel Number
+8	Fiber Channel Offset (Data Point Count, 1=0.5 m, 2=1 m, etc.)
+9	Hi: Alarm Status (0=Unmonitored, 1=Normal, 2=Warning, 3=Alarm, 4=Pre-Alarm) Lo: Double Position (0=None, 1=Return, 2=Active)
+10~11	Recorded Value (Float)
+12~14	Reserved

A single SOE record contains 15 registers and must be read consecutively. The fiber channel offset represents the data count offset within the corresponding channel. For example, a fiber channel offset of 1 indicates the first data point on that channel, corresponding to an actual distance of 1 × 0.5 = 0.5 m; an offset of 2 corresponds to 1.0 m, and so on.

6.3 Data Recorder Registers

6.3.1 Data Recorder Pointer Registers

Register Address	Type	Description	Format	Note
42000	RO	Fiber Channel 1 Total Pointer	UINT32	
42002	RO	Fiber Channel 2 Total Pointer	UINT32	
42004	RO	Fiber Channel 3 Total Pointer	UINT32	
42006	RO	Fiber Channel 4 Total Pointer	UINT32	

6.3.2 Data Recorder Operation Registers

Register Address	Type	Description	Format	Note
42010	WO	Fiber Channel Index [1]	UINT32	1–4
42012	WO	Data Recorder Pointer Index	UINT32	1–5760
42014	RW	Read Recorder Offset [2]	UINT32	0–29999

Notes:

1. Registers 42010–42013 must be written consecutively. The Data recorder pointer index is used to specify which recorder to read, with a maximum of one recorder per read. For example, writing a pointer index of 1 retrieves the most recent recorder, 2 retrieves the second most recent, and 5760 retrieves the oldest available recorder. After writing to register 42012, the read recorder offset (42014) is automatically restored to its default value of 0. Only the most recent 5,760 data recorders can be read.
2. The read recorder offset is used to set the starting offset within the data recorder file for the current read operation. If not set, reading starts from the beginning of the data recorder.

6.3.3 Data Recorder Buffer Registers

Register Address	Type	Description	Format	Note
42020	RO	Recorder Size [1]	UINT32	
42022	RO	Unix Timestamp	UINT32	
42024	RO	Valid Bytes in Frame [2]	UINT32	
42025~42124	RO	Data Buffer [3]	INT16	

Notes:

1. 42020~42022 registers must be read consecutively.
2. The number of valid bytes at the beginning of the data buffer.
3. The data buffer can hold up to 100 registers. When reading data, it should start from the valid byte count register of 42024 and read all data at once (until the end of the data buffer).

Reading process of Data Recorder:

a. Get the total pointer count for the channel.
b. Specify fiber channel index and data recorder pointer index.
c. If necessary, set read recorder offset (defaults to 0).
d. Read recorder size and Unix timestamp.

e. Read from 42024 to buffer end; offset increments automatically. Stop when valid bytes = 0.

6.4 Configuration File Information Data Register

Register Address	Type	Description	Format	Note
44000~44019	RW	Filename Path [1]	UINT16	
44020	RO	File Size	UINT32	
44022	RW	File Offset [2, 3, 4]	UINT32	
44024	RO	Valid Bytes in Frame	UINT16	
44025~44148	RO	File Data Buffer	UINT16	Read up to 244 bytes

Notes:

1. The configuration file information data registers are independent of the request message ID. When performing a read filename operation, all 20 registers must be read in a single read.
2. Reading this register individually returns the offset of the first byte in the file within the current file data buffer; reading it together with registers 44024–44148 will automatically adjust the file offset.
3. Read Flow Requirements:
   (1) Write the relative pathname of the file to be read (e.g., StationList.json) into the filename register. If the file does not exist or fails to open, the device shall respond with exception code 03.
   (2) If the file needs to be read starting from a specific offset, write the specific offset into the file offset register; if not written, the file is read from the beginning by default. During consecutive file reading, as long as the file offset register is read together with the data buffer each time, the device will automatically adjust the file offset as needed — there is no need to adjust the file read position by writing to the file offset register. Writing an illegal offset shall result in a response with exception code 03.
   (3) Read multiple registers from the file offset through the file data buffer in a single read. Each time the file offset register is read, the file offset increments automatically and the file data buffer updates automatically. If a packet of data needs to be re-read, simply skip reading the file offset register — the file data buffer will retain the original data.
   (4) Repeat Step 3 until the number of valid data bytes in the current frame is 0, or the file offset equals the file size, at which point the file transfer is complete.
   (5) Writing to the filename register in any state indicates the start of a new read flow.
4. There are only two ways to read file data: (1) start a consecutive read from register 44022, or (2) start a consecutive read from register 44024. The first method is recommended.

6.5 Channel Basic Parameter Registers

Register Address	Type	Description	Format	Note
50000	RW	Channel Enable [1]	UINT16	1=enable, 0=disable
50001	RW	Channel Fiber Temp. Measurement Length	UINT16	0~15000m (default: 0m)
50002	RW	Temperature Coefficient	Float	0~0.01 (default: 0.0018)
50004	RW	Meter Mark Coefficient	Float	0.8~1.2 (default: 1.0)
50006	RW	Meter Mark Offset	UINT16	0~100m (default: 0m)

6.6 Channel Insertion Point Parameter Registers

Register Address	Type	Description	Format	Note
50100	RO	Insertion Point Count [2]	UINT 16	
50101	RW	Insertion Point 1 Position	Float	Unit: m
50103	RW	Insertion Point 2 Position	Float	
...				
50499	RW	Insertion Point 200 Position	Float	

6.7 Channel Connection Point Parameter Registers

Register Address	Type	Description	Format	Note
51100	RO	Connection Point Count [2]	UINT 16	
51101	RW	Connection Point 1 Position	Float	Unit: m
51103	RW	Connection Point 2 Position	Float	
...				
51499	RW	Connection Point 200 Position	Float	

Notes:

1. The registers in sections 6.5 to 6.7 are related to the request message ID. They read the channel-related parameters at the specified index of the given fiber channel.
2. The parameters for channel insertion points and channel connection points must be written in a consecutive arrangement. When consecutive writing is ensured, all values can be written as 0 consecutively, which will clear the position information of all channel connection points / insertion points.

6.8 Time Parameter Registers

Register Address	Type	Description	Format	Note
60000	RW	Hi: Year (-2000), Lo: Month	UINT16	Year: 0~37(-2000), Month: 1~12
60001	RW	Hi: Day, Lo: Hour	UINT16	Day: 1~31, Hour: 0~23
60002	RW	Hi: Minute, Lo: Second	UINT16	Minute: 0~59, Second: 0~59
60003	RW	Millisecond	UINT16	0~999
60004~60005	RW	UNIX seconds (UTC) [1, 2]	UINT32	0x386D4380–0x7FE8177F (2000–2037 UTC)

Notes:

1. UNIX time and calendar date-time cannot be set simultaneously.
2. Registers 60000–60003 must be written consecutively, and registers 60004–60005 must be written consecutively; however, the complete time information (60000–60005) cannot be written in a single operation.

6.9 Device Information Registers

Register Address	Type	Description	Format	Note
60200~60219	RO	Device Type	UINT16	
60220	RO	Program Version	UINT16	
60221	RO	Modbus Protocol Version	UINT16	
60222	RO	Program Version Date (Year-2000)	UINT16	
60223	RO	Program Version Date (Month)	UINT16	
60224	RO	Program Version Date (Day)	UINT16	
60225~60226	RO	Reserved	UINT16	

Appendix A: 16-bit CRC Check Method

```
unsigned int formcrc16(unsigned char * startaddress, unsigned char bytecount)
{   unsigned char i, j ;
    unsigned int crc16value = 0xffff ;
    unsigned int polynomial = 0xa001 ;
    for ( i = 0 ; i < bytecount ; i ++ )
    {   crc16value = crc16value ^ ( * ( startaddress + i ) ) ;
        for ( j = 0 ; j < 8 ; j ++ )
        {   if ( ( crc16value & 1 ) == 1 )
                crc16value = ( crc16value >> 1 ) ^ polynomial ;
            else
                crc16value = crc16value >> 1 ;
        }
    }
    return   crc16value ;
}
```

Revision History Log

Version	Date	Description of Changes
1.0	2021/10/27	First version
1.1	2025/07/04	1. Updated serial ID description 2. Added "Pre-Alarm" to Zone/Segment alarms 3. Added Disconnection/Pre-Alarm to SOE events