

ETD-0612 Communication Protocol V1.0

1. Data frame definition

Leading	Serial number	Length	Instruction	Data segment	Check sum
---------	---------------	--------	-------------	--------------	-----------

Leading : 0xAA

Serial number: The serial number of the data packet, occupying 2 bytes. When it reaches the maximum value of 65535, it resets to 0 for calculation. When the receiving end needs to respond after receiving the data, the serial number of the response packet is the same as the received serial number.

Length: 1 byte, including the preamble + serial number + length + instruction + data segment + checksum total length.

Instruction: 1 byte, detailed instructions can be found in the instruction and data definition table.

Data Segment: Detailed information can be found in the instruction and data definition table.

Checksum: 2 bytes,

(1) Sender: Accumulates the data to send and obtains a sum. The sum is then complemented to get a 16-bit validation value. This value is then sent to the recipient along with the data to be transmitted.

(2) Recipient: Checksum Analysis

Step 1: Takes the last two bytes and combines them into a 16-bit checksum

Step 2: Accumulates the remaining data

Step 3: Adds the checksum value + checksum + 1. If the result is 0, the checksum is considered to be successful. Otherwise, it fails.

```

uint16_t uart_tx_checksum16(uint8_t* data, uint8_t len)
{
    uint16_t i, ret = 0;

    for(i = 0; i < len; i++)
    {
        ret += *(data+i);
    }
    ret = ~ret;

    return ret;
}

uint16_t uart_rx_checksum16(uint8_t* data, uint8_t len)
{
    uint16_t i, ret = 0;
    uint16_t checksum = ((*data+len-2) << 8) | (*(data+len-1));

    for(i = 0; i < len-2; i++)
    {
        ret += *(data+i);
    }
    ret += checksum;

    return (ret+1);
}
    
```

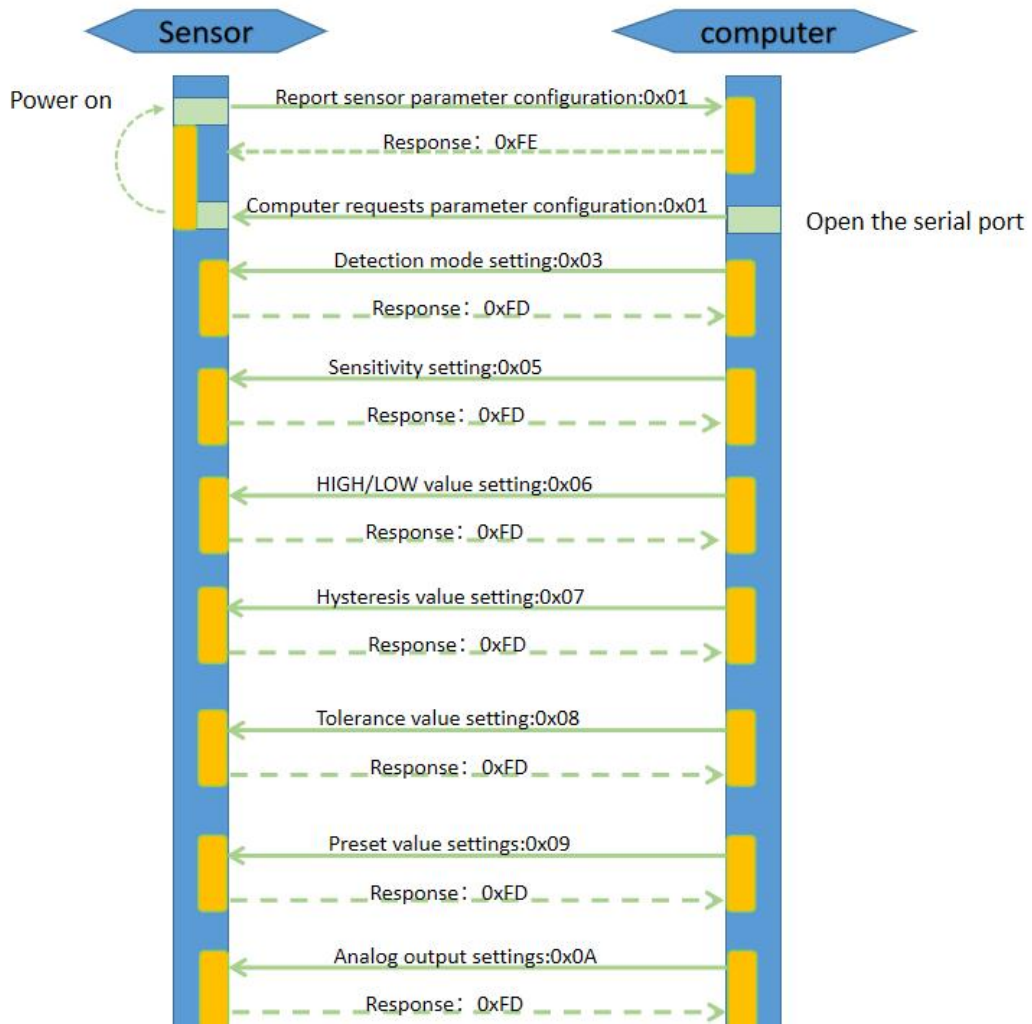
2. Directives and data definitions

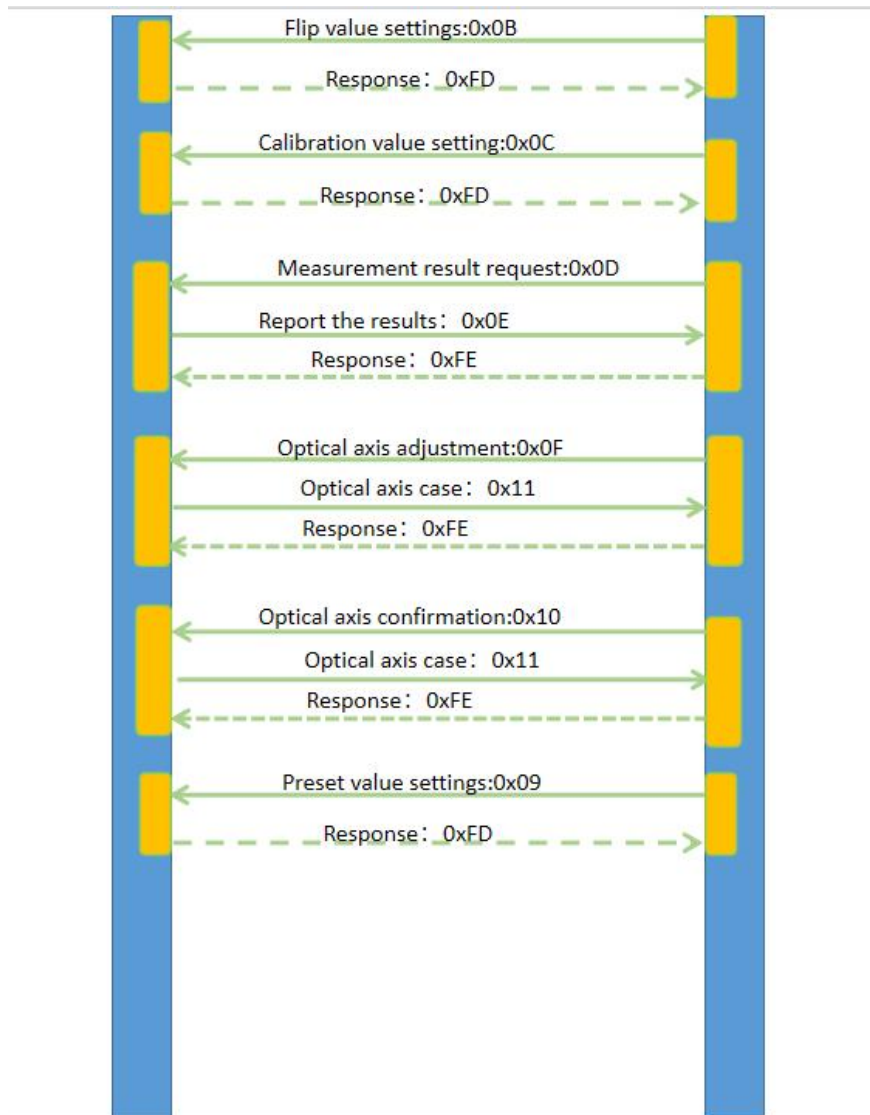
Number	Instruct	Data segment	Direction	Explain
1	0xFD		Sensor-->Host computer	Sensors respond to instructions and process requests from the host computer.
2	0xFE		Host computer- -> Sensor	The host computer responds to instructions and processes requests from sensors.
3	0x01		Sensor- -> Host computer	Sensor parameter reporting instruction: After the sensor initializes and powers on, it reports the sensor parameters.
4	0x02		Host computer- -> Sensor	Sensor parameter request instruction: The

				host computer opens the serial port and requests the sensor parameters for synchronization and display.
5	0x03	Occupies 4 bytes; 1st byte: Detection mode 0----Automatic edge detection mode 1----Edge detection mode 2----Width detection mode 3----Gap detection mode 2nd byte: Detection direction 0----Top direction 1----Bottom direction 3rd and 4th byte: specify the edge	Host computer- -> Sensor	Detection setting instruction: Sets the detection mode and direction. Currently, only automatic edge mode, edge mode, width mode, and gap mode are supported. Note that the detection direction only takes effect in edge mode. Reserving the temporary edge in bytes 3 and 4.
6	0x04	Occupies 2 bytes Short data, the value range is 1~500	Host computerr--> Sensor	Average count setting instruction.
7	0x05	Occupies 2 bytes: Byte 1: Determination level Byte 2: Decision filter	Host computer- -> Sensor	Sensitivity setting (not implemented) .
8	0x06	Occupies 8 bytes Bytes 1 to 4: High floating-point number Bytes 5 to 8: Low floating-point number	Host computer- -> Sensor	HIGH/LOW setting instruction.
9	0x07	Occupies 1 byte	Host computer- -> Sensor	Hysteresis value setting (not implemented) .
10	0x08	Occupies 4 bytes, floating-point type	Host computer- -> Sensor	Tolerance value setting.
11	0x09	Occupies 4 bytes, floating-point type	Host computerr--> sensor	Preset value setting.
12	0x0A	Occupies 1 byte 0---Voltage output 1-1---Current output	Host computer- -> Sensor	Analog output setting.
13	0x0B	Occupies 1 byte 0----No flip 1---Flip	Host computer- -> Sensor	Inversion value setting (not implemented) .
14	0x0C	Occupies 16 bytes Bytes 1 to 3: Measurement data 1, floating-point type Bytes 5 to 8: Adjustment value 1, floating-point type Bytes 9 to 12: Measurement data 2, floating-point type Bytes 13 to 16: Adjustment data 2, floating-point type	Host computer- -> Sensor	Calibration value setting (not implemented) .
14	0x0D	Occupies 4 bytes Measurement result, floating-point type	Host computer--> Sensor	Host computer request measurement result instruction: The sensor does not automatically upload digital results and requires a request from the host computer.

	0x0E	<p>Occupies 8 bytes</p> <p>Byte 1: Detection type</p> <p>1---No object being measured</p> <p>2---Top</p> <p>3---Bottom</p> <p>4---Gap</p> <p>5----Width (diameter)</p> <p>6----Complete obstruction</p> <p>Bytes 2 to 4: Alignment bytes</p> <p>Bytes 5 to 8: Measurement result, floating-point type</p>	Sensor-->Host computer	<p>Sensor report measurement result instruction. The structure is as follows:</p> <pre>typedef struct { uint8_t type; float value; }DetectResultStruct;</pre>
15	0x0F	<p>Occupies 1 byte</p> <p>0----Cancel optical axis adjustment</p> <p>1----Start optical axis adjustment</p>	Host computer- -> Sensor	Optical axis adjustment instruction.
16	0x10		Host computer- -> Sensor	Optical axis confirmation instruction: When moving the sensor to align the optical axis at the transmitting and receiving ends, perform the optical axis confirmation action.
17	0x11	<p>Occupies 1 byte</p> <p>1--TOP end offset</p> <p>2--BOTTOM end offset</p> <p>3--Aligned</p> <p>5--Beam spot too small</p>	Sensor--> Host computer	Optical axis status reporting instruction: During the optical axis calibration process, the sensor will continuously report the optical axis status in real-time. Users can adjust the sensor position based on indicator lights or information displayed on the host computer.
18	0x12	4 Bytes	Sensor-->Host computer	Spot boundary reporting instruction: The host computer reports the confirmed spot boundaries used for measurement.
19	0x13		Host computer- -> Sensor	Factory reset instruction.
20	0x14	Version information	Host computer- -> Sensor	Sensor software version request instruction.
21	0x15	<p>4 Bytes:</p> <p>"V1.0"</p> <p>Corresponding to 0x56,0x31,0x23,0x33</p>	Sensor- ->Host computer	Version information reporting instruction.

3. Data flow diagram:





4. Serial Port Settings

Baud rate: 115200; Data bits: 8 bits; Stop bits: 1 bit; Parity bit: none

5. Example Message

(1) Request Result: The request result instruction is 0x0D, and the response instruction is 0x0E

Sent message: 0xAA 0x00 0x01 0x07 **0x0D 0xFF 0x40**

0xAA: Frame Header

0x00 0x01: Sequence number (can be fixed)

0x07: Frame length

0x0D: Instruction

0xFF 0x40: Parity bit (16-bit checksum verification algorithm)

Response message: 0xAA 0x00 0x0D 0x0F **0x0E** 0x06 0x00 0x00 0x00 0x00 0x00 0x48 0x41 **0xFE 0x9C**

0xAA: Frame Header

0x00 0x0D: Sequence number (the sequence number can vary in the data packet reported by the sensor)

0x0F: Frame length

0x0E: Instruction

0x06 0x00 0x00 0x00 0x00 0x00 0x48 0x41: Data segment

0xFE 0x9C: Parity bit (16-bit checksum verification algorithm)