CJ6485 Temperature and Humidity Transmitter Data Sheet

Product Overview

The CJ6485 is a high-precision temperature and humidity transmitter designed for applications such as weather stations and HVAC system closed-loop control.

Technical Specifications

Parameter	Specification
Temperature Range	-40 to 120°C
Humidity Range	0 to 99.9% RH
Temperature Accuracy	±0.3°C (at 25°C)
Humidity Accuracy	±2% RH (at 25°C)
Sampling Period	3 seconds
Supply Voltage	12 to 36V DC
Product Dimensions	200mm (L) × 15.7mm (D)
Output Signal	RS485
Communication Protocol	Standard MODBUS RTU
Baud Rate	9600 (Default)
Display Resolution	Temp: 0.1°C; Humidity: 1% RH
Net Weight	110.0g
Sensitivity Drift	Temp: <0.1°C/year; Humidity: <0.5% RH/year

Wiring Instructions

The lead wire assignments for the FG6485 (CJ6485) are shown in Table 1.

Table 1: Lead Wire Assignment

Wire Color	Name	Description
Yellow	A+	RS485 A terminal
Red	V+	Power positive input
Black	GND	Power negative input
White	B-	RS485 B terminal

RS485 Communication Protocol

1. Internal Register Mapping

Register Info	Address	Register Info	Address
Humidity	0x0000	Device Model	0x0008
Temperature	0x0001	Version (Low 8 bits)	0x0009
Reserved	0x0002	Device ID (High 16 bits)	0x000A
Reserved	0x0003	Device ID (Low 16 bits)	0x000B
Reserved	0x0004	Temp Upper Alarm Value	0x000C
Reserved	0x0005	Temp Upper Alarm Enable	0x000D
Reserved	0x0006	Temp Lower Alarm Value	0x000E
Reserved	0x0007	Temp Lower Alarm Enable	0x000F
Humidity Upper Alarm Value	0x0010	Reserved	0x0018
Humidity Upper Alarm Enable	0x0011	Reserved	0x0019
Humidity Lower Alarm Value	0x0012	Reserved	0x001A
Humidity Lower Alarm Enable	0x0013	Reserved	0x001B
Reserved	0x0014	Temp Calibration Update	0x001C
Reserved	0x0015	Humidity Calibration Update	0x001D
Reserved	0x0016	Reserved	0x001E
Reserved	0x0017	Reserved	0x001F

2. Supported Function Codes

- **0x03**: Read Multiple Registers
- **0x10**: Write Multiple Registers

Read Command Format:

- **Master Frame**: Slave Address + 0x03 + Start Address (2 bytes) + Quantity (2 bytes) + CRC Low + CRC High
- **Transmitter Response**: Slave Address + 0x03 + Byte Count (1 byte) + Data 0 + … + Data n + CRC Low + CRC High

Write Command Format:

- **Master Frame**: Slave Address + 0x10 + Start Address (2 bytes) + Quantity (2 bytes) + Byte Count (1 byte) + Data 0 + … + Data n + CRC Low + CRC High
- **Transmitter Response**: Slave Address + 0x10 + Start Address (2 bytes) + Quantity (2 bytes) + CRC Low + CRC High

Special Notes for Write Function:

1. Only addresses 0x000C to 0x001E are writable. Others are read-only.
2. For 0x000C to 0x001B, if the data is out of range or violates logic, the register will not update and will retain its original value.
3. For 0x001C, 0x001D, and 0x001E, values exceeding the range will be clamped to the boundary values.
4. When sending data, the actual value must be multiplied by 10 (e.g., 25.5 becomes 255).

3. Error Codes

- **0x81**: Illegal Function Code (Unsupported)
- **0x82**: Illegal Data Address
- **0x83**: Illegal Data Value (Writing to read-only register or prohibited write)

4. Communication Examples

Read Request Example:

Master sends: `01 03 00 00 00 02 C4 0B`

Field	Bytes	Value	Description
Slave Address	1	01	Target slave address 01
Function Code	1	03	Read registers
Start Address	2	0000	Starting at address 0x0000
Quantity	2	0002	Read 2 registers (4 bytes total)
CRC	2	C40B	CRC-16 (Low byte first)

Response Example:

Transmitter returns: `01 03 04 01 D7 00 D6 CA 69`

Field	Bytes	Value	Description
Slave Address	1	01	From slave address 01
Function Code	1	03	Read registers
Byte Count	1	04	4 bytes of data returned
Reg 0 High	1	01	Humidity high byte (Address 0x00)
Reg 0 Low	1	D7	Humidity low byte (Address 0x00)
Reg 1 High	1	00	Temperature high byte (Address 0x01)
Reg 1 Low	1	D6	Temperature low byte (Address 0x01)
CRC	2	CA69	CRC-16 (Low byte first)

Data Calculation:

- Uploaded values must be divided by 10.
- Positive range: 0x0000 to 0x7FFF.
- Negative range: 0x8000 to 0xFFFF (Two's complement).
- **Example 1**: Temp `0x00FF` = 255 → **25.5°C**
- **Example 2**: Humidity `0x0311` = 785 → **78.5% RH**
- **Example 3**: Temp `0xFF9B` = -100 (0xFFFF - 0xFF9B = 0x64) → **-10.0°C**

CRC-16 Calculation Method

1. Preset a 16-bit register to `0xFFFF` (CRC register).
2. XOR the first 8-bit byte of the message with the low 8 bits of the CRC register.
3. Shift the CRC register right by one bit, filling the MSB with 0. Check the bit shifted out.
4. If the shifted bit is 0, repeat step 3. If it is 1, XOR the CRC register with the polynomial `0xA001`.
5. Repeat steps 3 and 4 until 8 shifts are completed.
6. Repeat steps 2 to 5 for the next byte of the message.
7. After all bytes are processed, swap the high and low bytes of the resulting 16-bit CRC register.
8. The final value is the CRC code.

C Language Implementation

```c
unsigned short crc16(unsigned char *ptr, unsigned char len) {
    unsigned short crc = 0xFFFF;
    unsigned char i;
    while (len--) {
        crc ^= *ptr++;
        for (i = 0; i < 8; i++) {
            if (crc & 0x01) {
                crc >>= 1;
                crc ^= 0xA001;
            } else {
                crc >>= 1;
            }
        }
    }
    return crc;
}
```

Address Configuration

The slave address is set using an 8-bit DIP switch located on the back of the device.

DIP Switch Mapping

Switch	1	2	3	4	5	6	7	8
Value	1	2	4	8	16	32	64	128

Calculation: The address is the sum of the values of all switches set to the **ON** position.

Examples:

- **Address 1**: Switch 1 is ON (Value = 1).
- **Address 2**: Switch 2 is ON (Value = 2).
- **Address 13**: Switches 1, 3, and 4 are ON (1 + 4 + 8 = 13).

Note: Please turn off the power before changing the measurement range jumpers or address settings.