WN69LP Modbus RTU V1.0.3

Revision History

Version	Date	Revision
V1.0.0	2025/12/25	- Initial release
V1.0.1	2026/03/12	- Corrected UVI calculation
V1.0.2	2026/06/05	- Added low-power control register - Added firmware version register
V1.0.3	2026/09/24	- Add an option to disable automatic data reporting by default - Update the documentation for the latest rainfall registers.

1. Modbus

1.1 Parameters

Item	Bit Description
Data Bits	8
Parity Bit	None
Stop Bit	1
Checksum	CRC (polynomial 0x8005)
Baud Rate	User configurable (default 9600 bps)

1.2 Data Frame Definition

1.2.1 Standard Modbus RTU Frame Format

Host Query Frame:

Address	Function Code	Start Register Address	Register Quantity	CRC Low Byte	CRC High Byte
1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	1 Byte

● In this mode, the host can send the standard frame above directly, and the slave responds normally.

● The host may also prepend the wake-up byte 0xFF before the frame for redundant compatibility, and the slave can still process it normally.

Slave Response Frame:

Address	Function Code	Data Length	Data Set 1	Data Set 2	Data Set N	CRC Low Byte	CRC High Byte
1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte

1.2.2 Low-Power Mode Modbus RTU Frame Format

● When the slave is in low-power mode, the host must add a wake-up byte 0xFF before the standard Modbus RTU frame to wake the slave. The complete frame format is:

Wake-up Byte	Address	Function Code	Start Register Address	Register Quantity	CRC Low Byte	CRC High Byte
1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	1 Byte

● 0xFF: Wake-up byte, used to trigger the slave to exit the low-power state.

● The wake-up byte is not a valid part of the Modbus frame and is not included in CRC verification. The CRC is calculated only from the slave address through the data field.

● When sending, the host should send 0xFF and the following Modbus frame continuously. The interval between bytes must not exceed 3 character times to ensure correct recognition by the slave.

Slave Response Frame:

Address	Function	Data	Data Set 1	Data Set 2	Data Set N	CRC Low	CRC High

	Code	Length				Byte	Byte
1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte

1.3 Registers

Register Address	Function	Write/Read	Description
0160 H	Slave Address	Write/Read	- Data type: UINT16 - Range: 1 -> 252
0161 H	Baud Rate	Write/Read	- Data type: UINT16 - Range: 1 -> 4 - Example: 1: 4800 bps 2: 9600 bps 3: 19200 bps 4: 115200 bps
0162 H	Automatic Reporting Interval	Write/Read	- Data type: UINT16 - Unit: 250 ms. - Formula: Time = value * 250 ms - Range: 2 -> 16383.75 seconds - Example: 3CH = 60 * 0.25 = 15 seconds If the automatic reporting time is less than 2 seconds, automatic reporting is disabled. ~~- Default setting: Automatic reporting time is 16 seconds.~~ - Default setting: Automatic reporting is disabled (starting from firmware version V1.0.5). - Note: After automatic reporting is enabled, all weather data is measured 1 second before reporting.
0163 H	Sampling Period		- Data type: UINT16 - Unit: 250 ms. - Range: 2 -> 16383.75 seconds - Example: F0H = 240 * 0.25 = 60 seconds. **- Disable condition: If the configured value is less than 2 seconds, periodic sampling is disabled. In this state, weather data is not measured automatically.**

			- Normal operating range: If the configured value is between 2 and 16383.75 seconds, weather data is measured periodically at the specified interval. **- Default setting: The sampling period is 16 seconds.**
0164 H	Clear Rainfall	Write Only	- Data type: UINT16 - Write 0xA55A: Clear the rainfall register and latest rainfall register.
0165 H	Software Reset	Write Only	- Data type: UINT16 - Write 0xA55A: Trigger software reset. After confirmation, the slave immediately enters a 5-second reset/no-response state.
0166 H	Low-Power Control	Write/Read	- Data type: UINT16 - Enable low-power mode: Write 0x0001. - Disable low-power mode: Write 0x0000. **- Default setting: Low-power mode disabled. Switching between low-power and non-low-power modes is supported only by firmware V1.0.3 and later.**
0167 H	Firmware Version	Read Only	- Data type: UINT16 - Example: Ver1.0.0 = 64H
0168 H ... 017F H	Reserved	/	/
0180 H	Illuminance	Read Only	- Data type: UINT16 - Unit: lux - Formula: Illuminance = value * 10 lux Range: 0 lux -> 300,000 lux Invalid value: FFFFH
0181 H	UV Index	Read Only	- Data type: UINT16 - Formula: UV index = value Range: 0 -> 15 Invalid value: FFFFH
0182 H	Temperature	Read Only	- Data type: UINT16 - Unit: deg C - Formula: Temperature = (value - 400) * 0.1 deg C - Range: -40.0 deg C -> 60.0 deg C

			Invalid value: FFFFH - Example: 10.5 deg C = 1F9H -10.5 deg C = 127H
0183 H	Humidity	Read Only	- Data type: UINT16 - Unit: % - Formula: Humidity = value - Range: 1% -> 99% - Invalid value: FFFFH - Example: 68% = 44H
0184 H	Average Wind Speed	Read Only	- Data type: UINT16 - Unit: m/s - Formula: Average wind speed = value * 0.1 m/s - Range: 0.0 -> 50.0 m/s - Invalid value: FFFFH - Example: 1.2 m/s = 0CH
0185 H	Gust	Read Only	- Data type: UINT16 - Unit: m/s - Formula: Average wind speed = value * 0.1 m/s - Range: 0.0 -> 50.0 m/s - Invalid value: FFFFH - Example: 1.2 m/s = 0CH
0186 H	Wind Direction	Read Only	- Data type: UINT16 - Unit: deg - Formula: Wind direction = register value - Range: 0 deg -> 359 deg - Invalid value: FFFFH - Example: 125 deg = 7DH
0187 H	Rainfall	Read Only	- Data type: UINT16 - Unit: mm - Formula: Rainfall = value * 0.254 mm - Example: 4.572 mm = 12H
0188 H	Barometric Pressure	Read Only	- Data type: UINT16 - Unit: hPa - Formula: Barometric pressure = value * 0.1 hPa - Range: 300.0 -> 1100.0 hPa - Invalid value: FFFFH - Example: 1002.6 hPa = 272AH
0189 H	Battery Voltage	Read Only	- Data type: UINT16 - Unit: V

			- Formula: Battery voltage = value * 0.01 V - Range: 0.00 V -> 3.30 V - Battery voltage is automatically measured and updated every 60 seconds. - Invalid value: FFFFH - Example: 2.40V = F0H 3.00V = 12CH
018A H	VCC (External Power Supply Voltage)	Read Only	- Data type: UINT16 - Unit: V - Formula: VCC = value * 0.1 V - Range: 6.0 V -> 24.0 V - VCC supply voltage is automatically measured and updated every 60 seconds. - Invalid value: FFFFH - Example: 12.0V = 78H 24.0V = F0H
018B H	Latest Rainfall	Read Only	- Data type: UINT16 - Unit: mm - Formula: Latest rainfall = value * 0.254 mm - Example: 1.524 mm = 06H - The value is automatically reset to zero after being read. Each read returns the incremental rainfall since the previous read. For continuous monitoring, the register should be read periodically.

1.4 Examples

1.4.1 Normal Operations

Example 1: Read illuminance, UV index, temperature, humidity, average wind speed, gust, wind direction, rainfall, barometric pressure, battery voltage, external power supply voltage, and latest rainfall.

Query Frame:

Slave Address	Function	Start Register	Register	CRC Low Byte	CRC High Byte

	Code	Address	Quantity		
0x24	0x03	0x0180	0x000C	0x42	0xEE

Response Frame:

Slave Address	Function Code	Valid Data Length	Valid Data	CRC Low Byte	CRC High Byte
0x24	0x03	0x18	0x06E7 0x0001 0x0296 0x003C 0x000C 0x001C 0x0096 0x008B 0x271F 0x0138 0x0078 0x0005	0xBA	0xB9

Read Data:

Illuminance = 0x06E7 * 10 = 17670 lux

UV index = 0x0001 = 1

Temperature = (0x0296 - 400) * 0.1 = 26.2 deg C

Humidity = 0x003C = 60%

Average wind speed = 0x000C * 0.1 = 1.2 m/s

Gust = 0x001C * 0.1 = 2.8 m/s

Wind direction = 0x0096 = 150

Rainfall = 0x008B * 0.254 = 35.306 mm

Barometric pressure = 0x271F * 0.1 = 1001.5 hPa

Battery voltage = 0x0138 * 0.01 = 3.12 V

External power supply voltage = 0x0078 * 0.1 = 12.0 V

Latest rainfall = 0x0005 * 0.254 = 1.270 mm

Example 2: Read temperature

Query Frame:

Slave Address	Function Code	Register Address	Register Quantity	CRC Low Byte	CRC High Byte
0x24	0x03	0x0182	0x0001	0x22	0xEB

Response Frame:

Slave Address	Function Code	Valid Data	Valid Data	CRC Low Byte	CRC High Byte

		Length			
0x24	0x03	0x02	0x028A	0x75	0x44

Temperature = (0x028A - 400) * 0.1 = 25.0 deg C

Example 3: Set the baud rate to 4800 bps.

Query Frame:

Slave Address	Function Code	Register Address	Register Quantity	CRC Low Byte	CRC High Byte
0x24	0x06	0x0161	0x0001	0x1F	0x1D

Response Frame:

Slave Address	Function Code	Valid Data Length	Valid Data	CRC Low Byte	CRC High Byte
0x24	0x06	0x02	0x0001	0x34	0x8F

Example 4: Set the slave device address to 0x26.

Query Frame:

Slave Address	Function Code	Register Address	Register Quantity	CRC Low Byte	CRC High Byte
0x24	0x06	0x0160	0x0026	0x0E	0xC7

Response Frame:

Slave Address	Function Code	Valid Data Length	Valid Data	CRC Low Byte	CRC High Byte
0x24	0x06	0x02	0x0026	0x74	0x95

Example 5: Software reset the slave device.

Query Frame:

Slave Address	Function Code	Register Address	Valid Data	CRC Low Byte	CRC High Byte
0x24	0x06	0x0165	0xA55A	0x64	0x77

Response Frame:

Slave Address	Function Code	Valid Data Length	Valid Data	CRC Low Byte	CRC High Byte
0x24	0x06	0x02	0xA55A	0x0E	0x24

Example 6: Enable low-power mode. Firmware V1.0.3 and later only.

Query Frame:

Slave Address	Function Code	Register Address	Valid Data	CRC Low Byte	CRC High Byte
0x24	0x06	0x0166	0x0001	0xAE	0xDC

Response Frame:

Slave Address	Function Code	Valid Data Length	Valid Data	CRC Low Byte	CRC High Byte
0x24	0x06	0x02	0x0001	0x34	0x8F

Example 7: Disable low-power mode. Firmware V1.0.3 and later only.

Query Frame:

Wake-up Byte	Slave Address	Function Code	Register Address	Valid Data	CRC Low Byte	CRC High Byte
0xFF	0x24	0x06	0x0166	0x0000	0x6F	0x1C

Response Frame:

Slave Address	Function Code	Valid Data Length	Valid Data	CRC Low Byte	CRC High Byte
0x24	0x06	0x02	0x0000	0xF5	0x4F

1.4.2 Special Operations

When the configuration is abnormal or unclear, use this command to check the status.

Host Query Frame:

Wake-up Byte	Prefix	Read/Write Baud Rate	Slave Device Address	CRC Low Byte	CRC High Byte
1 Byte Fixed value: 0xFF	3 Bytes Fixed value: 0xFDFDFD	1 Byte 0: Read baud rate 1: Set baud rate to 4800 2: Set baud rate to 9600 3: Set baud rate to 19200 4: Set baud rate to 115200	1 Byte 0: Read slave device address 1~252: Valid range of slave device address	1 Byte	1 Byte

Slave Response Frame:

Prefix	Baud Rate	Slave Device Address	CRC Low Byte	CRC High Byte
3 Bytes Fixed	1 Byte 1: Baud rate 4800 2: Baud rate 9600	1 Byte	1 Byte	1 Byte

value: 0xFDFDFD	3: Baud rate 19200 4: Baud rate 115200			

Example 8: Read the baud rate and slave device address.

Query Frame:

Wake-up Byte	Prefix	Read Baud Rate	Read Slave Device Address	CRC Low Byte	CRC High Byte
0xFF	0xFDFDFD	0x00	0x00	0xE9	0x88

Response Frame:

Prefix	Baud Rate	Slave Device Address	CRC Low Byte	CRC High Byte
0xFDFDFD	0x04	0x24	0xEB	0x53

Baud rate 115200, slave device address 0x24.

Example 9: Set the baud rate to 9600 bps.

Query Frame:

Wake-up Byte	Prefix	Set Baud Rate	Read Slave Device Address	CRC Low Byte	CRC High Byte
0xFF	0xFDFDFD	0x02	0x00	0xE8	0xE8

Response Frame:

Prefix	Baud Rate	Slave Device Address	CRC Low Byte	CRC High Byte
0xFDFDFD	0x02	0x24	0xE8	0xF3

Set the baud rate to 9600 bps and read slave device address 0x24.

Example 10: Set the slave device address to 0x5A.

Query Frame:

Wake-up Byte	Prefix	Read Baud Rate	Set Slave Device Address	CRC Low Byte	CRC High Byte
0xFF	0xFDFDFD	0x00	0x5A	0x69	0xB3

Response Frame:

Prefix	Baud Rate	Slave Device Address	CRC Low Byte	CRC High Byte
0xFDFDFD	0x02	0x5A	0x68	0xD3

Read baud rate 9600 bps and set the slave device address to 0x5A.

1.5 Error Codes

Error Code	Meaning	Description
01	Illegal Function Code	The function code is not 0x03 or 0x06.
02	Illegal Data Address	The data address is out of

		range.
03	Illegal Data	The data length is out of range.
08	CRC Check Failed	CRC check failed.

When replying with an error code, add 0x80 to the original function code. Example:

Example 8: Error-code response frame indicating CRC check failure in the query frame.

Slave Address	Function Code	Error Code	CRC Low Byte	CRC High Byte
0x24	0x83	0x08	0x51	0x3D

2. Wiring Definition

Color	Description	Remarks
Red	VCC	6.0 -> 24 V DC power supply
Black	GND	Ground
Green	485_A	RS485A line
White	485_B	RS485B line

3. Default Settings

Default slave device address is 24H.

4. CRC Calculation

The wake-up byte (0xFF) in the host query frame is not included in CRC calculation.

Example 9:

```
const uint8_t CRCHi_Table[256] =  {
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, 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,
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
};
```

```
const uint8_t CRCLo_Table[256] = {
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
};

uint16_t F_GetCRC16Bit(uint8_t * lpBuff, uint8_t ucLen)
{
    /*
     * CRC_Width: CRC16
     * CRC_Polynomial: 0x8005(x16+x15+x2+1)
     * CRC_Input_Reflected: LSB First
     * CRC_Initial_Value: 0xFFFF
     * CRC_Final_XOR: 0x0000
     */
    uint8_t Index;
    uint8_t CRCHi = 0xFF;
    uint8_t CRCLo = 0xFF;

    while(ucLen)
    {
        ucLen--;
        Index = *lpBuff ^ CRCLo;
        CRCLo = CRCHi ^ CRCHi_Table[Index];
        CRCHi = CRCLo_Table[Index];
        lpBuff++;
    }
    return (uint16_t)((CRCHi << 8) | CRCLo);
}
```