

# Dual Vacuum Measurement System - Modbus RTU Slave Documentation

## Configuration Settings

### Modbus Communication Parameters

Parameter	Configuration Method	Valid Range	Default
Slave ID	Web interface: <code>/system</code>	1 - 247	Configurable
Baudrate	Web interface: <code>/system</code>	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	Configurable
Data Format	Fixed	8 data bits, No parity, 1 stop bit (8N1)	-
CRC	Automatic	Modbus CRC-16	-

### How to Configure via Web Interface

#### 1. Connect to Device:

- Connect to the device's WiFi network (SoftAP mode) or via configured WiFi network
- Open browser and navigate to device IP address

#### 2. Access System Configuration:

- Navigate to: `http://<device-ip>/system`
- Enter desired Slave ID (1-247)
- Select Baudrate from dropdown menu
- Click "Save & Restart"

#### 3. Device will restart with new settings applied

---

# Modbus Register Map

## Read-Only Registers (Function Code 03)

Register Address	Data Type	Description	Unit	Access
3000-3001	Float32 (IEEE)	Vacuum Sensor 1 Reading	Selected Unit*	Read Only
3002-3003	Float32 (IEEE)	Vacuum Sensor 2 Reading	Selected Unit*	Read Only

\*Unit is configured in the device (mBar, Torr, Pa, etc.)

## Write-Only Registers (Function Code 16)

Register Address	Data Type	Description	Unit	Access
3010-3011	Float32 (IEEE)	High Limit Setpoint - Sensor 1	mBar (vacuum units)	Write Only
3012-3013	Float32 (IEEE)	Low Limit Setpoint - Sensor 1	mBar (vacuum units)	Write Only
3014-3015	Float32 (IEEE)	High Limit Setpoint - Sensor 2	mBar (vacuum units)	Write Only
3016-3017	Float32 (IEEE)	Low Limit Setpoint - Sensor 2	mBar (vacuum units)	Write Only

### Important Notes:

- Each Float32 value occupies **2 consecutive 16-bit registers**
- Float values are transmitted in **Big-Endian format** (MSB first)
- Written setpoint values are automatically **snapped to the closest valid entry** in the vacuum lookup table
- All setpoint changes are **permanently saved** to flash memory

## Supported Function Codes

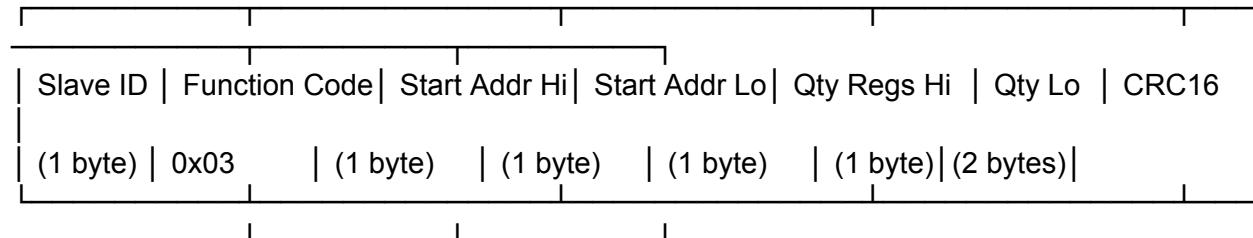
Function Code	Name	Description	Supported
03 (0x03)	Read Holding Registers	Read vacuum sensor values	Yes
16 (0x10)	Write Multiple Registers	Write setpoint values	Yes
Other codes	-	Any other function code	Returns Exception 0x01

## Read Operations (FC03)

### Purpose

Read current vacuum values from Sensor 1 and/or Sensor 2.

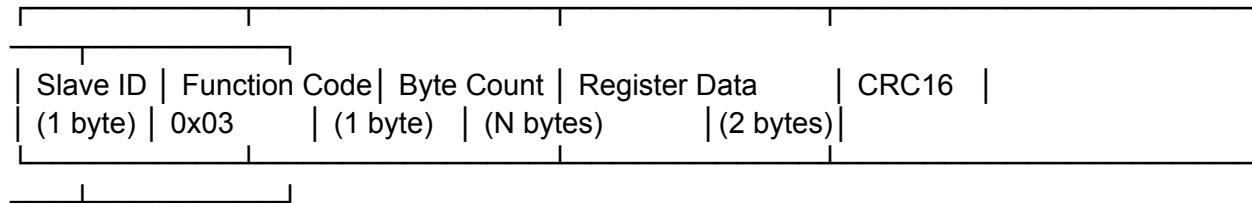
### Request Frame Format



### Request Parameters

Field	Description	Valid Values
Slave ID	Configured slave address	1-247
Function Code	Always 0x03 for read	0x03
Start Address	Starting register address	3000 or 3002
Quantity of Registers	Number of registers to read (must be even)	2 or 4

## Response Frame Format



## Float32 Data Format

Each Float32 value is transmitted as 4 bytes in **Big-Endian order**:

Register N: [Byte3 (MSB)] [Byte2]  
Register N+1: [Byte1] [Byte0 (LSB)]

## Validation Rules

### Valid Conditions:

- Start address must be 3000 or 3002
- Quantity must be 2 or 4 (for 1 or 2 floats)
- Quantity must be even (2 registers per float)
- End address must not exceed 3004

### Invalid Conditions (will return exception):

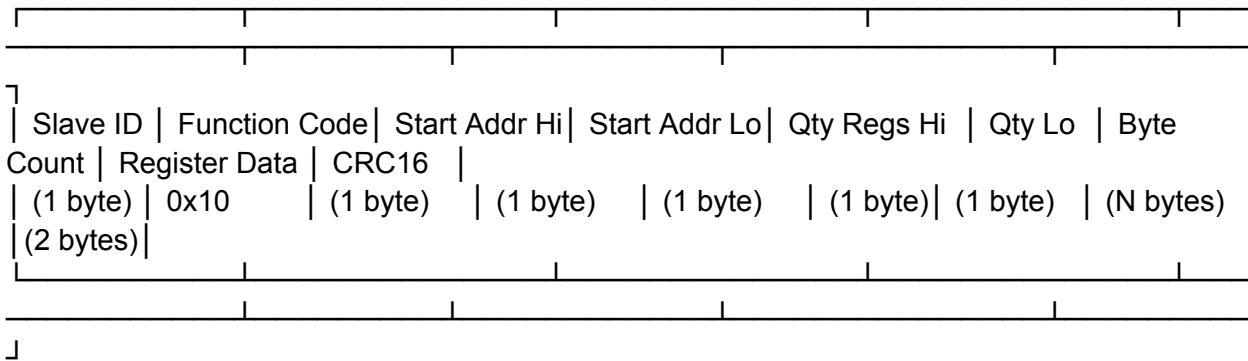
- Start address < 3000 or > 3002
- Quantity = 0 or odd number
- Quantity > 125
- Reading beyond address 3004

## Write Operations (FC16)

### Purpose

Write setpoint values (High Limit / Low Limit) for vacuum sensors.

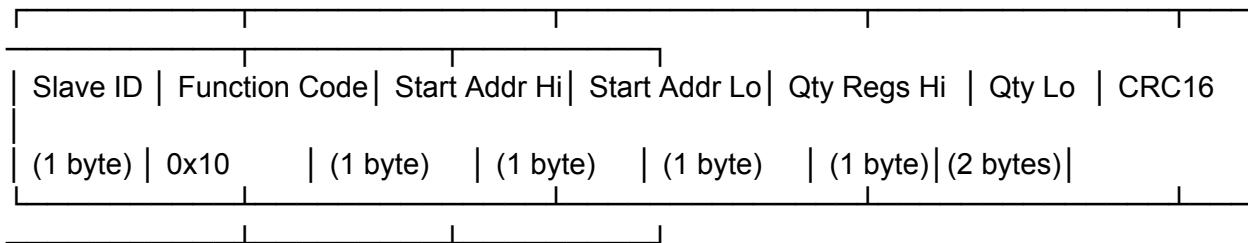
## Request Frame Format



## Request Parameters

Field	Description	Valid Values
<b>Slave ID</b>	Configured slave address	1-247
<b>Function Code</b>	Always 0x10 for write	0x10 (16 decimal)
<b>Start Address</b>	Starting register address	3010, 3012, 3014, 3016
<b>Quantity of Registers</b>	Must be even (2 per float)	2, 4, 6, or 8
<b>Byte Count</b>	Quantity × 2	4, 8, 12, or 16
<b>Register Data</b>	Float32 values in Big-Endian	See format below

## Response Frame Format (Success)



## Setpoint Value Processing

**Important:** Written values are automatically adjusted:

1. **Snapping:** The device finds the **closest valid value** in its internal vacuum lookup table
2. **Validation:** Only table-validated vacuum values are accepted
3. **Persistence:** All changes are **saved to flash memory** immediately
4. **Confirmation:** Success response is sent only after save is complete

## Validation Rules

### Valid Conditions:

- Start address must be 3010, 3012, 3014, or 3016
- Quantity must be even (2, 4, 6, or 8)
- Byte count = Quantity × 2
- Quantity ≤ 8

### Invalid Conditions (will return exception):

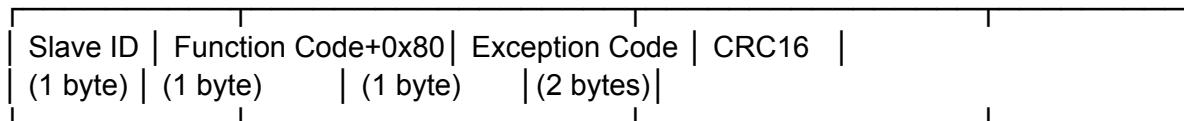
- Quantity = 0 or odd
- Quantity > 8
- Invalid start address
- Byte count doesn't match quantity

---

## Exception Responses

When an error occurs, the device returns an exception response:

### Exception Frame Format



## Exception Codes

<b>Code</b>	<b>Name</b>	<b>Description</b>	<b>Common Causes</b>
<b>0x01</b>	Illegal Function	Function code not supported	Using FC other than 03 or 16
<b>0x02</b>	Illegal Data Address	Invalid register address	Reading/writing outside valid range
<b>0x03</b>	Illegal Data Value	Invalid data in request	Odd quantity, byte count mismatch, quantity out of range

---

## Frame Examples

### Example 1: Read Vacuum Sensor 1 (Address 3000-3001)

#### Master Request:

Slave ID: 0x01 (Example: Slave 1)  
 Function Code: 0x03 (Read Holding Registers)  
 Start Addr Hi: 0x0B (3000 = 0xBB8)  
 Start Addr Lo: 0xB8  
 Quantity Hi: 0x00  
 Quantity Lo: 0x02 (2 registers = 1 float)  
 CRC Lo: 0xXX (Calculated)  
 CRC Hi: 0xXX

Complete Frame: 01 03 0B B8 00 02 [CRC-Lo] [CRC-Hi]

#### Slave Response (Example: Vacuum = 250.5 mBar):

Slave ID: 0x01  
 Function Code: 0x03  
 Byte Count: 0x04 (4 bytes for 1 float)  
 Data Byte 3: 0x43 (MSB of float)  
 Data Byte 2: 0x7A  
 Data Byte 1: 0x40  
 Data Byte 0: 0x00 (LSB of float)  
 CRC Lo: 0xXX  
 CRC Hi: 0xXX

Complete Frame: 01 03 04 43 7A 40 00 [CRC-Lo] [CRC-Hi]

Float Value: 0x437A4000 = 250.5 (IEEE 754 format)

---

## **Example 2: Read Both Sensors (Address 3000-3003)**

### **Master Request:**

Slave ID: 0x01  
Function Code: 0x03  
Start Addr Hi: 0x0B (3000 = 0x0BB8)  
Start Addr Lo: 0xB8  
Quantity Hi: 0x00  
Quantity Lo: 0x04 (4 registers = 2 floats)  
CRC Lo: 0xXX  
CRC Hi: 0xXX

Complete Frame: 01 03 0B B8 00 04 [CRC-Lo] [CRC-Hi]

### **Slave Response:**

Slave ID: 0x01  
Function Code: 0x03  
Byte Count: 0x08 (8 bytes for 2 floats)  
Sensor 1 Byte 3: 0x43 (Vacuum1 MSB)  
Sensor 1 Byte 2: 0x7A  
Sensor 1 Byte 1: 0x40  
Sensor 1 Byte 0: 0x00 (Vacuum1 LSB)  
Sensor 2 Byte 3: 0x44 (Vacuum2 MSB)  
Sensor 2 Byte 2: 0x48  
Sensor 2 Byte 1: 0x00  
Sensor 2 Byte 0: 0x00 (Vacuum2 LSB)  
CRC Lo: 0xXX  
CRC Hi: 0xXX

Complete Frame: 01 03 08 43 7A 40 00 44 48 00 00 [CRC-Lo] [CRC-Hi]

Values: Vacuum1 = 250.5, Vacuum2 = 800.0

---

### **Example 3: Write High Limit for Sensor 1 (Address 3010-3011)**

#### **Master Request (Set HL1 = 500.0 mBar):**

Slave ID: 0x01  
Function Code: 0x10 (Write Multiple Registers)  
Start Addr Hi: 0x0B (3010 = 0x0BC2)  
Start Addr Lo: 0xC2  
Quantity Hi: 0x00  
Quantity Lo: 0x02 (2 registers)  
Byte Count: 0x04 (4 bytes)  
Data Byte 3: 0x43 (500.0 MSB)  
Data Byte 2: 0xFA  
Data Byte 1: 0x00  
Data Byte 0: 0x00 (500.0 LSB)  
CRC Lo: 0xXX  
CRC Hi: 0xXX

Complete Frame: 01 10 0B C2 00 02 04 43 FA 00 00 [CRC-Lo] [CRC-Hi]

Float Value: 0x43FA0000 = 500.0

#### **Slave Response (After snapping and saving):**

Slave ID: 0x01  
Function Code: 0x10  
Start Addr Hi: 0x0B  
Start Addr Lo: 0xC2  
Quantity Hi: 0x00  
Quantity Lo: 0x02  
CRC Lo: 0xXX  
CRC Hi: 0xXX

Complete Frame: 01 10 0B C2 00 02 [CRC-Lo] [CRC-Hi]

---

### **Example 4: Write All Four Setpoints (Address 3010-3017)**

#### **Master Request:**

Slave ID: 0x01  
Function Code: 0x10  
Start Addr Hi: 0x0B (3010 = 0x0BC2)

Start Addr Lo: 0xC2  
Quantity Hi: 0x00  
Quantity Lo: 0x08 (8 registers = 4 floats)  
Byte Count: 0x10 (16 bytes)

HL1 Data: 43 FA 00 00 (500.0)  
LL1 Data: 41 20 00 00 (10.0)  
HL2 Data: 44 34 00 00 (720.0)  
LL2 Data: 40 A0 00 00 (5.0)

CRC Lo: 0xXX  
CRC Hi: 0xXX

Complete Frame:  
01 10 0B C2 00 08 10  
43 FA 00 00 41 20 00 00 44 34 00 00 40 A0 00 00  
[CRC-Lo] [CRC-Hi]

---

#### **Slave Response:**

Complete Frame: 01 10 0B C2 00 08 [CRC-Lo] [CRC-Hi]

---

### **Example 5: Exception Response (Invalid Address)**

#### **Master Request (Invalid address 2000):**

Complete Frame: 01 03 07 D0 00 02 [CRC-Lo] [CRC-Hi]

#### **Slave Exception Response:**

Slave ID: 0x01  
Function Code: 0x83 (0x03 + 0x80 = Exception)  
Exception Code: 0x02 (Illegal Data Address)  
CRC Lo: 0xXX  
CRC Hi: 0xXX

Complete Frame: 01 83 02 [CRC-Lo] [CRC-Hi]

---

## Example 6: Exception Response (Odd Quantity)

### Master Request (Invalid quantity = 3):

Complete Frame: 01 03 0B B8 00 03 [CRC-Lo] [CRC-Hi]

### Slave Exception Response:

Slave ID: 0x01  
Function Code: 0x83 (0x03 + 0x80 = Exception)  
Exception Code: 0x03 (Illegal Data Value)  
CRC Lo: 0xXX  
CRC Hi: 0xXX

Complete Frame: 01 83 03 [CRC-Lo] [CRC-Hi]

---

## Web Configuration Interface

### Accessing Configuration Pages

URL Path	Function	Description
/	Calibration Page	Main page with live sensor readings
/system	System Configuration	Configure Modbus Slave ID, Baudrate, SoftAP SSID
/wifi	WiFi Configuration	Configure WiFi credentials and network settings
/update	OTA Firmware Update	Upload new firmware (requires authentication)

### System Configuration Steps

#### Navigate to System Config Page:

1. http://<device-ip>/system
2. **Configure Modbus Parameters:**

- **Slave ID:** Enter value between 1-247
- **Baudrate:** Select from dropdown:
  - 1200 bps
  - 2400 bps

- 4800 bps
- 9600 bps (common default)
- 19200 bps
- 38400 bps
- 57600 bps
- 115200 bps

### 3. Configure SoftAP Settings (Optional):

- **SoftAP SSID:** Custom access point name
- **Password:** Not required (open network by default)

### 4. Save and Restart:

- Click "Save & Restart" button
  - Device will restart with new settings
  - Allow 10-15 seconds for restart
- 

## CRC-16 Calculation

The device uses **Modbus CRC-16** algorithm for error checking.

### CRC Properties

- **Polynomial:** 0xA001 (reversed 0x8005)
- **Initial Value:** 0xFFFF
- **Byte Order:** CRC Low byte first, then CRC High byte
- **Applied To:** All bytes except the CRC itself

### CRC Calculation Process

1. Initialize CRC register to 0xFFFF
2. XOR first byte with CRC low byte
3. Look up result in CRC tables
4. Update CRC with table values
5. Repeat for all bytes in frame
6. Append CRC Low byte, then CRC High byte

**Note:** The device automatically calculates and validates CRC for all frames.

---

# Troubleshooting Guide

## Common Issues and Solutions

Issue	Possible Cause	Solution
No response from device	Wrong Slave ID	Verify Slave ID matches configuration
	Wrong baudrate	Check baudrate setting in <a href="#">/system</a>
	Wiring issue	Check RS485 A/B connections
CRC error	Electrical noise	Add termination resistors (120Ω)
	Timing issues	Increase frame delay
Exception 0x02	Wrong register address	Use addresses 3000-3003 (read) or 3010-3017 (write)
Exception 0x03	Odd register quantity	Use even numbers (2, 4, 6, 8)
Write not saved	Device busy	Wait for response before next write
Float value incorrect	Byte order	Ensure Big-Endian format (MSB first)

---