ADDITEL CORPORATION

PRODUCT COMMUNICATION PROTOCOL · TECHNICAL DOCUMENT

A20 Series Pressure Transmitter Modbus-RTU Communication Protocol and Command Set

RS-485 Interface · Modbus-RTU

Document No.	A20-485-ModbusRTU-EN
Version	V00
Release Date	2026-09-30
Applicable Product	A20 Series Pressure Transmitter

Additel Corporation

ICS Schneider Messtechnik GmbH
Briesestraße 59
D-16562 Hohen Neuendorf / OT Bergfelde
Tel.: 03303 / 50 40 66
Fax.: 03303 / 50 40 68
info@ics-schneider.de
www.ics-schneider.de

Table of Contents

1 Introduction

2 Serial Port Settings

3 Message Format

3.1 Master Request Frame

3.2 Slave Response Frame

3.3 Error Handling

4 Modbus-RTU Communication Functions

4.1 Function Code 03H: Read Registers

4.2 Function Code 06H: Write Single Register

4.3 Function Code 08H: Echo Test (Diagnostics)

4.4 Function Code 10H: Write Multiple Registers

4.5 Modbus Register Map

5 Examples

6 Revision History

1 Introduction

This document specifies the Modbus-RTU communication protocol of the A20 series pressure transmitter over the RS-485 interface, including the serial port settings, message formats, communication function codes (03H / 06H / 08H / 10H), and the Modbus register map. It is intended for the development of host (master) devices and for system integration and commissioning.

2 Serial Port Settings

Parameter	Default	Description
Data bits	8	—
Start bit	1	—
Stop bits	1	1 by default; 2 selectable
Parity	None	None by default; odd or even selectable
Baud rate	9600	9600 bps by default; switchable to 115200 bps by command
Frame length	10 bits	11 bits if 2 stop bits or parity is selected; 12 bits if both are selected

3 Message Format

Communication between the master and slave devices follows the Modbus-RTU frame structure: the master sends a request frame, and the slave returns a response frame.

3.1 Master Request Frame

DevAddr	Function Code	N bytes of parameters (optional)	CRC16_L	CRC16_H
Device address	Function code	n = 0 … 6	CRC low byte	CRC high byte

- **DevAddr (device address)**: Address 0 is reserved for broadcast; addresses 1 (default) through 247 are available in bus mode; address 250 is the transparent address, reserved for non-bus mode, through which any device may communicate; addresses 251–255 are reserved for future development.
- **Function Code**: The device selects and executes a function according to the function code, which is a 7-bit code whose MSB (bit 7) is always 0. The functions are described in Section 4.
- **N bytes of parameters**: Parameters required by the function (n = 0…6, depending on the function).
- **CRC16**: A 16-bit checksum used to verify the integrity of the received data. If an error is detected, the entire message is discarded.

> **Note**: A message sent by the master is at least 4 bytes long.

3.2 Slave Response Frame

DevAddr	Function Code	N bytes of parameters (optional)	CRC16_L	CRC16_H
Device address	Function code (MSB = X)	Returned data	CRC low byte	CRC high byte

- **DevAddr**: Corresponds to the address of the responding device.
- **Function Code**: Identical to the function code sent by the master. If the MSB X = 0, the function was executed correctly; if X = 1, an exception error occurred.
- **N bytes of parameters**: Any data requested through the function.
- **CRC16**: As described above.

> **Note**: A response message from the slave is at least 5 bytes and at most 250 bytes long.

3.3 Error Handling

Two types of errors may occur during message exchange between the master and slave devices: transmission errors and exception errors.

3.3.1 Transmission Errors

Transmission errors are mainly caused by line faults, in which case the message format is incorrect. Possible problems include:

- The received message is too short, e.g., the time interval between bytes within a frame is excessive;
- The message length exceeds the capacity of the internal transmission buffer;
- The byte length cannot be parsed correctly;
- The CRC16 checksum is wrong.

In these cases, the slave rejects the request and does not reply, which causes a timeout at the master; the master must resend the request. When a transmission error occurs, all received data is ignored, the slave remains in receive mode, and the master must initiate a new data exchange.

3.3.2 Exception Errors

The message has been received correctly (no transmission error), but the function code and/or the parameters are invalid. The slave returns an exception response unless the message was received in broadcast mode. The exception response frame has the following format (the MSB of the function code is set to 1):

DevAddr	1	Function Code	Exception Code	CRC16_L	CRC16_H
Device address	MSB = 1	Original function code	Exception code	CRC low byte	CRC high byte

The exception code defines 5 exception errors:

Exception Code	Name	Typical Cause
1	Illegal function (not implemented)	The requested function code is not supported
2	Illegal data address	Wrong or undefined start address, wrong register quantity/range
3	Illegal data value	A parameter value is outside the permitted range (e.g., more than 120 registers, incorrect frame length, etc.)
4	Slave device failure	Write to a protected register, written data out of the defined range, etc.
32	Initialization	The device has restarted but has not yet been initialized; this occurs every time the device is re-powered after power-off

4 Modbus-RTU Communication Functions

4.1 Function Code 03H: Read Registers

Reads a series of consecutive registers starting at StAdd in the Modbus address space. The returned data must be interpreted as defined in the "Modbus Register Map" (Section 4.5).

- One register contains 2 bytes (16 bits);
- A maximum of 120 registers can be read in a single cycle;
- The number of returned data bytes must be even (1 register = 2 bytes);
- Process values (pressure and temperature) of double-precision float size must be read within a single request;
- Reading a float value always requires 2 data registers;
- If a channel error occurs, NaN (float) or ovl (integer) is returned.

Request frame:

DevAddr	0x03	StAdd_H	StAdd_L	#Reg_H	#Reg_L	CRC16_L	CRC16_H
Device address	Function code	Start address, high byte	Start address, low byte	No. of registers, high byte	No. of registers, low byte	CRC low byte	CRC high byte

Normal response frame:

DevAddr	0x03	#Bytes	Data_H	Data_L	…	Data_H	Data_L	CRC16_L	CRC16_H
Device address	Function code	Byte count	Data	Data	…	Data	Data	CRC low byte	CRC high byte

Error response frame:

DevAddr	0x83	Exception Code	CRC16_L	CRC16_H
Device address	Function code + 0x80	Exception code	CRC low byte	CRC high byte

Exception code descriptions:

- **2 — Illegal data address**: Wrong start address or wrong number of registers; the start address is undefined; the registers are outside the defined range.
- **3 — Illegal data value**: The requested number of registers is out of range; the value must be ≤ 120.
- **32—**The device is not yet initialized.

4.2 Function Code 06H: Write Single Register

This function is the same as function 10H, but supports writing only 1 register. One register contains 2 bytes (16 bits). This function is suitable for a single configuration step, and its returned errors can be easily distinguished from other register write operations.

> **Note**: Single-register operations are not permitted at any address involving floating-point values.

Request frame:

DevAddr	0x06	StAdd_H	StAdd_L	Data_H	Data_L	CRC16_L	CRC16_H
Device address	Function code	Start address, high byte	Start address, low byte	Data, high byte	Data, low byte	CRC low byte	CRC high byte

Normal response frame (echo of the request):

DevAddr	0x06	StAdd_H	StAdd_L	Data_H	Data_L	CRC16_L	CRC16_H
Device address	Function code	Start address, high byte	Start address, low byte	Data, high byte	Data, low byte	CRC low byte	CRC high byte

Error response frame:

DevAddr	0x86	Exception Code	CRC16_L	CRC16_H
Device address	Function code + 0x80	Exception code	CRC low byte	CRC high byte

Exception code descriptions:

- **1 —** Writing is not supported;
- **2 — Illegal data address**: The address cannot be accessed with function code 06; the start address (for writing) is undefined;
- **3 — Illegal data value**: Incorrect frame length;
- **4 — Slave device failure**: An attempt was made to write to a protected register; the written data is outside the defined data range;
- **32—**The device is not yet initialized.

4.3 Function Code 08H: Echo Test (Diagnostics)

This function performs a quick line check. It simply returns the received data; the data may be any 2-byte value (the sub-function code must be 0x0000).

Request frame:

DevAddr	0x08	0x00	0x00	Data_H	Data_L	CRC16_L	CRC16_H
Device address	Function code	Sub-function code, high byte	Sub-function code, low byte	Data, high byte	Data, low byte	CRC low byte	CRC high byte

Normal response frame (echo):

DevAddr	0x08	0x00	0x00	Data_H	Data_L	CRC16_L	CRC16_H
Device address	Function code	Sub-function code, high byte	Sub-function code, low byte	Data, high byte	Data, low byte	CRC low byte	CRC high byte

Error response frame:

DevAddr	0x88	Exception Code	CRC16_L	CRC16_H
Device address	Function code + 0x80	Exception code	CRC low byte	CRC high byte

Error code descriptions:

- **1 — Illegal sub-function code**: The sub-function code is not supported (it is not 0x0000);
- **32—**The device is not yet initialized.

4.4 Function Code 10H: Write Multiple Registers

Writes a series of consecutive registers starting at StAdd into the Modbus address space. One register contains 2 bytes (16 bits), and the number of returned bytes is even (1 register = 2 bytes). Writing a float value always requires 2 data registers. The #Reg field in the response declares the number of registers actually written (if an error occurs, this number differs from that in the request).

Request frame:

DevAddr	0x10	StAdd_H	StAdd_L	#Reg_H	#Reg_L	#Bytes	Data_H	Data_L	…	CRC16_L	CRC16_H
Device address	Function code	Start address, high byte	Start address, low byte	No. of registers, high byte	No. of registers, low byte	Byte count	Data	Data	…	CRC low byte	CRC high byte

Normal response frame:

DevAddr	0x10	StAdd_H	StAdd_L	#Reg_H	#Reg_L	CRC16_L	CRC16_H
Device address	Function code	Start address, high byte	Start address, low byte	No. of registers, high byte	No. of registers, low byte	CRC low byte	CRC high byte

Error response frame:

DevAddr	0x90	Exception Code	CRC16_L	CRC16_H
Device address	Function code + 0x80	Exception code	CRC low byte	CRC high byte

Exception code descriptions:

- **1—**Writing is not supported;
- **2 — Illegal data address**: The start address is undefined or the number of registers is wrong; the start address (for writing) is undefined;

- **3 — Illegal data value**: The amount of data is out of range; the number of bytes is not twice the number of registers;
- **4 — Slave device failure**: An attempt was made to write to a protected register; the written data is outside the defined data range;
- **32—**The device is not yet initialized.

4.5 Modbus Register Map

4.5.1 Process Value Read Range (0x000x) — 32-bit Float (Big-Endian)

Modbus Address	Channel	R/W	Unit	Description
0x0002	Pressure	R	kPa	Sensor pressure value (32-bit float: high word register first, low word register second)

4.5.2 Process Value Read Range (0x002x) — 32-bit Integer

Modbus Address	Channel	R/W	Unit	Description
0x0022	Pressure	R	Pa	Sensor pressure value (32-bit integer: high word register first, low word register second)

4.5.3 Device Configuration Range (0x02xx)

Modbus Address	R/W	Register Name	Description
0x0200	R/W	UART	UART settings: Bit3–0 baud rate (0: 9600; 1: 115200); Bit4 parity enable (0: no parity; 1: parity enabled); Bit5 parity mode (0: odd; 1: even); Bit6 stop bits (0: 1 stop bit; 1: 2 stop bits)
0x0202	R	S/N-H	Serial number, high 16 bits
0x0203	R	S/N-L	Serial number, low 16 bits
0x0204	R	CFG_P	Active pressure channel: Bit1: pressure
0x0208	R/W	LP_FILT	Bit7–4: filter setting value
0x0209	R	P-Mode	Sensor and calibration type, Bit3–0 (pressure): 0: G (gauge); 2: A (absolute); 3: D (differential); 15: not defined
0x020C	R	STATUS	Device status byte (see Section 4.5.4)
0x020D	R/W	DEV_ADDR	Device address (default 1)
0x020E	R	VER_H	Software version (16 bits)
0x020F	R	VER_L	Software version (16 bits)
0x0215	R/W	ModbusInterFrameTime9k6	Minimum inter-frame time at 9600 bps: value × 100 µs = inter-frame timeout (default: 35 → 3.5 ms)
0x0216	R/W	ModbusInterFrameTime115k2	Minimum inter-frame time at 115200 bps: value ×

			100 µs = inter-frame timeout (default: 18 → 1.8 ms)
`0x0223`	R/W	SPS	Sampling rate in samples per second: 0x00: 2.5 Sps; 0x01: 5 Sps; 0x02: 10 Sps; 0x03: 25 Sps; 0x04: 60 Sps; 0x05: 122 Sps; 0x06: 244 Sps; 0x07: 488 Sps
`0x0250–0x0257`	R	ASCII serial number	16-byte ASCII serial number

4.5.4 Status Byte STATUS (0x020C)

The STATUS byte contains the current device status:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	**ERR2**	Reserved	Reserved	Reserved	Reserved	**P1**	Reserved

- **ERR2 (Bit6)**: Set when a computation error occurs during the calculation of the analog output. This happens when the analog signal is in saturation (depending on the scaling settings);
- **P1 (Bit1)**: Set when a measurement or computation error occurs in the associated channel.

4.5.5 Device Command Range (0xFFxx)

Modbus Address	R/W	Command Name	Description
`0xFF00`	W	`Set Zero P1`	Zero adjustment of the pressure sensor
`0xFF02`	W	`Reset Zero P1`	Cancel the zero adjustment of the pressure sensor

Constraints:

- Writing is allowed with function code 10H only (a write cycle using 2 registers);
- Only one command may be sent per request;
- This address range is not readable.

> **Note**: Absolute pressure modules do not support the zero adjustment; the unit of the zero-point data is kPa.

5 Examples

No.	DevAddr	Function Code	Register Address	Operation	Message (Request / Response)	Result
1	1	03H	0x0002	Read pressure value	**Request:** 01 03 00 02 00 02 65 CB **Response:** 01 03 04 42 C9 08 58 38 4F	100.5162963 kPa
2	1	03H	0x020D	Read device address	**Request:** 01 03 02 0D 00 01 14 71 **Response:** 01 03 02 00 01 79 84	1
3	250	03H	0x020D	Read device address (transparent address)	**Request:** FA 03 02 0D 00 01 01 FA **Response:** FA 03 02 00 01 9C 50	1
4	250	06H	0x020D	Change device address to 2	**Request:** FA 06 02 0D 00 02 8D FB **Response:** FA 06 02 0D 00 02 8D FB	2
5	250	10H	0x020D	Change device address to 2	**Request:** FA 10 02 0D 00 01 02 00 02 72 78 **Response:** FA 10 02 0D 00 01 84 39	—

6 Revision History

Version	Date	Description	Prepared by
V00	2026-09-30	Initial release	Additel Corporation