PRONEM-midi
REGISTER ADDRESS FOR MODBUS RTU

EMKO ELEKTRONIK A.S.
Demirtas Org. San. Bolg.
Karanfil Sk. No: 6 TR
16369 Bursa / TURKEY

Phone	: +90 224 261 1900
Fax	: +90 224 261 1912
Url	: www.emkoelektronik.com.tr
e-mail	: emko@emkoelektronik.com.tr
	exposales@emkoelektronik.com.tr

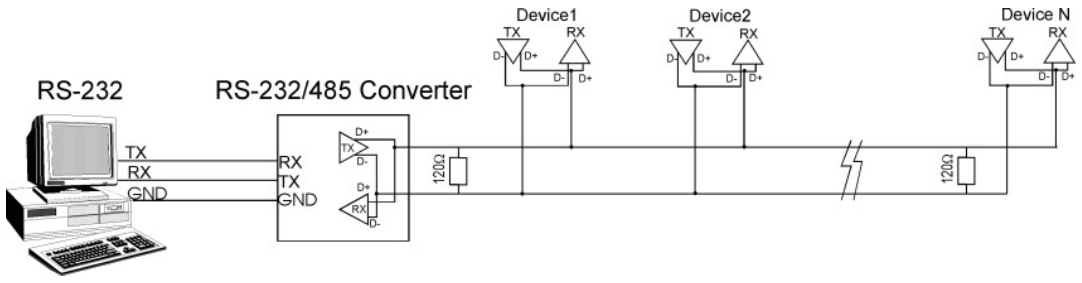
1. Introduction

This manual describes the RS-232 and RS-485 communication using Modbus® protocol in PRONEM-midi instruments.
In PRONEM-midi instruments Modbus® RTU protocol is used.

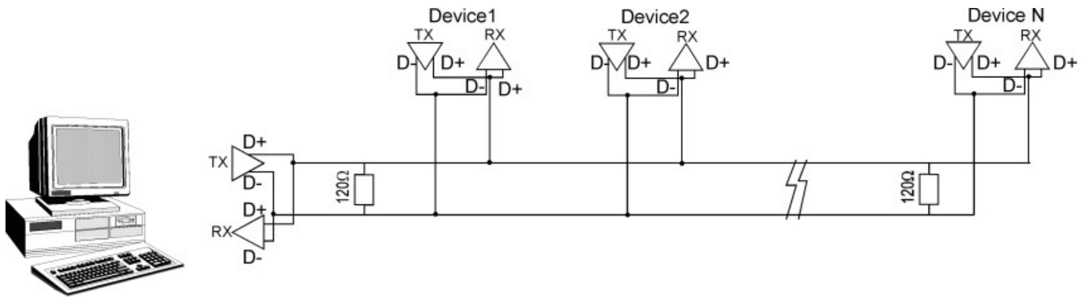
2. Connecting Devices To The Bus

Note : Each instrument connected to the bus must have a unique address.
All instruments must be adjusted to the same mode (RTU), baudrate, parity and stop bit.

It is recommended to use twisted-pair cable for RS-485 connection in order to minimaze signal errors due to the noise. To reduce cable reflections over long distances, RS-485 systems require line termination. This is achieved by putting two120 Ω terminating resistors. One resistor must be put PC's input / output buffer and the other buffer of the last device.

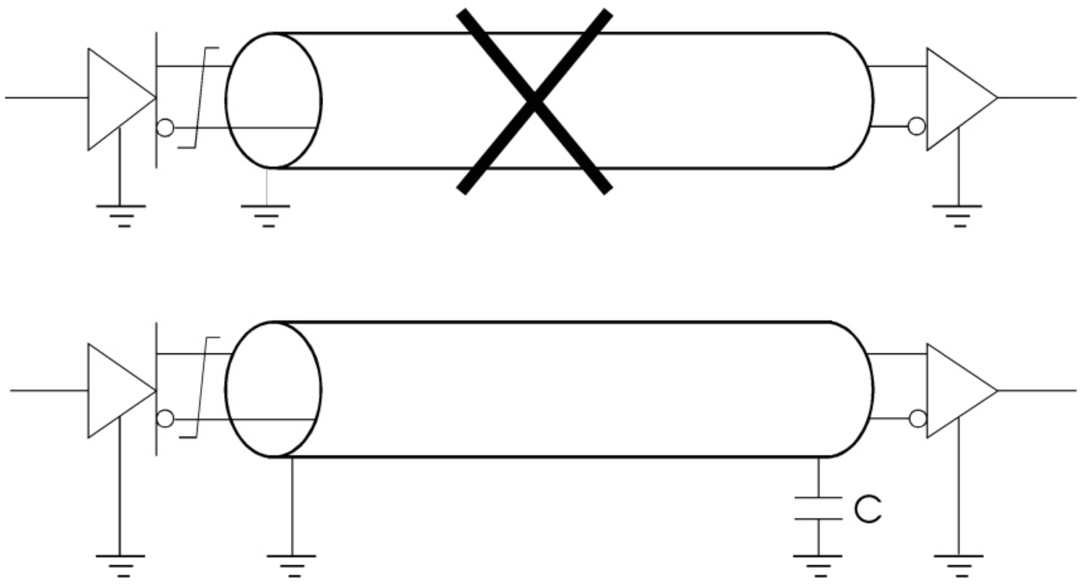


RS-485 interface with seperate converter.



RS-485 interface with plug in board.

SHIELDING of the CABLE



When the shield is connected to ground at both ends, also high frequency interference is avoided. Circulating currents are avoided by a capacitor C in series with one of the ground terminals.

3. Modbus® Protocol

A communication protocol defines commands and data formats that will be known by all instruments on the system. Modbus® is a master-slave protocol with all transaction initiated by a single host (e.g. PC). Message packets contains device address, a command, data and a checksum for error detection. Each slave device continually monitors the bus looking for the beginning of the message. Message packets are detected by all slaves, but only one ,whose address matches that transmitted, answers it, others ignore the message.

3.1.Transmission Modes in Modbus®

3.1.1.Transmission Spesification

Interface	: RS-232 and RS-485
Communication System	: Half Dublex
Synchronizing System	: Start-stop synchronizing
Data Length	: 8 bits
Parity	: None, odd, even
Stop Bit	: 1, 2 stop bits
Transmission Rate	: 9600, 19200, 38400, 57600 and 115200
Transmission Cable	: Twisted pair cable with shield.
Error Detection Techniques	:1. Parity Checks: None / Odd / Even parity 2. Cyclic redundancy checks (CRC)

3.1.2.Function Codes

Function Code 03 : Read Holding Register
Function Code 04 : Read Input Registers
Function Code 06 : Write Single Register

3.2. Modbus® Message Framing

Modbus® messages transmit with frames. Beginning and ending of the frame is known by the slave devices.
If the slave devices receive a character that is beginning of the frame, they read the address field and determine the owner of the device. Also they know when the message is completed. If the message isn't completed failures can be occurred.

3.2.1. RTU Framing

In this mode messages start with a silent interval of at least 3.5 character times. After this interval device address is sent. The devices on the network waits for the silent intervals.When the first field is received each device decodes it to know if it is the owner of the message. After the last character is sent 3.5 character silent interval marks the end of the message. A new message can begin after this interval.

The whole message must be continuous. If a 1.5 character silent interval occur before completing the message, the device eliminates the received message and assumes that the next byte is the address field of a new message. If a new message begins earlier than 3.5 characters times, message will be considered the continuation of the message, then CRC field will not be okey for the message.

3.2.2. Address Field

In RTU mode address field has eight bits. Slave device address can be between 1-247.

3.2.3. Function Field

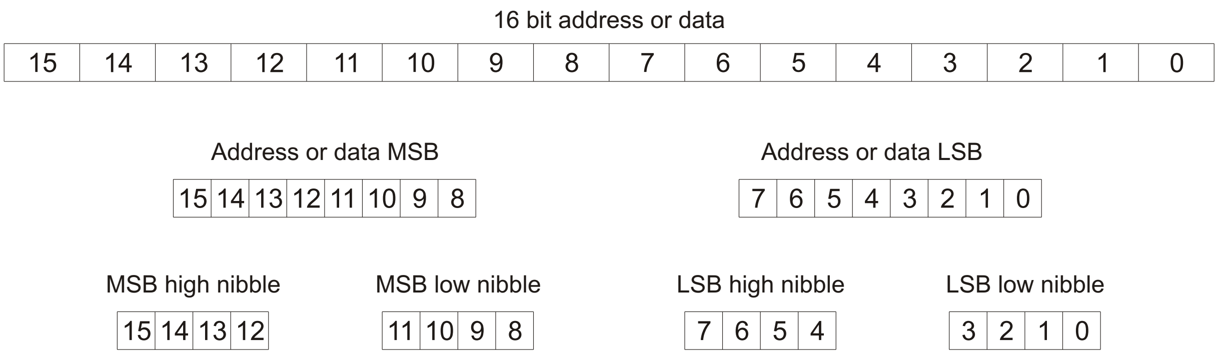
In RTU mode function field has eight bits. Function field tells the slave device which action will be performed. If there is no failure slave returns the same function code but if there is a failure to indicate the error slave device returns the function code with its most significant bit set to a logic 1. Error codes will be explained.

For example if master sends a message to read a group of holding registers and the function code will be:
0000 0011 (Hex 03)
If the slave takes the message without error, it returns back the same function code, but if there is an exception the function code will be:
1000 0011 (Hex 83)

3.2.4. Data Field

This field can include register addresses, how many byte will be read and the count of the read bytes. For example if the master wants to read a group of holding registers, the data field includes the register address where to start to read, and how many registers are to be read.

3.3. RTU Mode



Note : 1 must be substracted from register address value when data is sent to the device or data is read from the device. E.g. If you want to read register address 15, 14 is sent for register address.

When controllers are set up to communicate on a Modbus® network using RTU (Remote Terminal Unit) mode, each 8-bit byte in a message contains two 4-bit hexadecimal characters. Each message must be transmitted in a continuous stream.

The format in RTU mode is:
Coding System : 8 bit binary,hexadecimal 0-9,A-F
Error Check Field: Cyclical Redundancy Check (CRC)

3.3.1 CRC Calculation

A 16-bit CRC field is added to the end of the message. CRC is a calculation of a message contents. The slave device recalculates the CRC and compares with the CRC contained in the message. If two values aren't equal a failure occurs, slave device ignores the message.

A simpler method involves swapping the low and high order bytes of the CRC integer at the end of the calculation. This is shown in the following routine.

1- Load a 16-bit register (CRC Register) with FFFF Hex. (all1's).
2 - Exclusive-OR the first eight bits of the message with the low-order byte of the CRC register. Put the result in the CRC register.
3 - Shift the CRC register one bit to the right (divide by two), filling the MSB with a zero.
4 - If the bit shifted out in three is a one, Exclusive-OR the CRC register with the value A001 Hex.
5 - Repeat steps 3 and 4 until eight shifts have been performed and the bits tested. A single byte has thus been processed.
6 - Repeat steps 2 to 5 using the next eight-bit byte of the message until all bytes have been processed.
7 - The final contents of the CRC register are tagged on to the end of the message with the most significant byte first.
8 - Swap the low and high order bytes of the integer result.

An implementation of the CRC calculation in C code is show below.

```
unsigned int check_sum(unsigned char *buff, char start, char bytes)
{
        Char byte_cnt,bit_cnt; /* loop counters */
        unsigned int crc_reg; /* Result register */
        unsigned int CRCHi, CRCLO;/*Low and high order bytes of the crc*/
        /* Set the CRC register to all 1's */ crc_reg = 0xFFFF;
        /* Repeat for each byte of sub string */
        for(byte_cnt=start; byte_cnt<(bytes+start); byte_cnt++)
        {
                crc_reg = crc_reg ^ (Unsigned int)buff[byte_cnt]; /*EXOR CRC &Next Byte*/
                /* Test each bit of the CRC */
                for(bit_cnt=0; bit_cnt<8; bit_cnt++)
                {
                        if(crc_reg & 0x0001)
                        {
                                crc_reg = crc_reg >>1; /* IF LSB=1 EXOR
                                CRC with A001H*/
                                crc_reg = crc_reg ^ 0Xa001; /* Then shift
                                CRC toward LSB */
                        }
                        else crc_reg = crc_reg>>1; /* ELSE Shift CRC towards LSB */
                }
        }
        CRCLo=crc_reg>>8; /*Swap the low and high order bytes of the crc result*/
        CRCHi=crc_reg<<8;
        crc_reg = CRCLo+CRCHi;
        return crc_reg; /*Final CRC register Result */
}
```

3.4 Exception Responses

* If the slave does not receive the query because of a communication error , no response is returned. (Timeout Error)

* If the slave receives the query, but detects a communication error (parity, LRC or CRC) , no response is returned. (Timeout Error)

* If the slave receives the query without communication error, but can not handle it (e.g.if Master wants to read non-existent register), the slave will return an exception

The Error Codes are below:

01 : ILLEGAL FUNCTION : The function code received in the query is not an allowable action for the slave.

02 : ILLEGAL DATA ADDRESS : The data address received in the query is not an allowable address for the slave.

03 : ILLEGAL DATA VALUE : A value contained in the query data field is not an allowable value for the slave.

05 : ACKNOWLEDGE : The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occuring in the master. The master can next issue a Poll Program Complete message to determine if processing is completed.

If there is a failure to indicate the error slave device returns the function code with its most significant bit set to a logic 1.
Command* = Command's most significant bit set to a logic 1.

4. PRONEM-midi REGISTER ADDRESS FOR MODBUS RTU

4.1. HOLDING REGISTERS (4xxxx)

Parameter	Parameter Description	Address	Multiplier
Slave address	Slave Address (1 - 247) (Default = 1)	40001	1
Baud rate	Baud Rate 0 - 9600 baud (Default) 1 - 19200 baud 2 - 38400 baud 3 - 57600 baud 4 - 115200 baud	40002	1
Parity	Parity 0 - NONE (Default) 1 - ODD 2 - EVEN	40003	1
Stop bit	Stop Bit 0 - 1 Stop Bit (Default) 1 - 2 Stop Bit	40004	1

Example-1 : To read the "Baud rate” parameter's value;

Slave ID	: 1
Command	: 3 (Read Holding Register)
Register Address	: 1
Register Count	: 1

Slave ID	1
Command	3
Register Address MSB	0
Register Address LSB	1
Register Count MSB	0
Register Count LSB	1
CRC MSB	213 (decimal)
CRC LSB	202 (decimal)

If the value of the Baud rate parameter is 9600kbps, the device sends the bytes below:

Slave ID	1
Command	3
Byte Count	2
Data MSB	0
Data LSB	0
CRC MSB	184 (decimal)
CRC LSB	68 (decimal)

Byte Count = 2. (The number of bytes of Received Data)

Example-2 : To write the "Baud rate” parameter's value;

Slave ID	**:** 1
Command	**:** 6 (Write Single Register)
Register Address	**:** 1
Data	**:** 4

Slave ID	1
Command	6
Register Address MSB	0
Register Address LSB	1
Data MSB	0
Data LSB	4
CRC MSB	217 (decimal)
CRC LSB	201 (decimal)

The normal response is an echo of the request, returned after the register contents have been written. The device sends the bytes below:

Slave ID	1
Command	6
Register Address MSB	0
Register Address LSB	1
Data MSB	0
Data LSB	4
CRC MSB	217 (decimal)
CRC LSB	201 (decimal)

Byte Count = 2. (The number of bytes of Received Data)

The device must be power off and on if you want to use changed parameters.

4.2. INPUT REGISTERS (3xxxx)

Description	Address	Format	Multiplier
Temperature	30001	INT 16	0.01
Relative Humidity	30002	INT 16	0.1
Device Revision	30003	INT 16	1

INT 16 : signed int, 16 bit data

To read data from the instrument;
Example-1 : Reading the value of the **Temperature**

Slave ID	: 1
Command	: 4 (Read Input Register)
Register Address	: 0
Register Count	: 1

Slave ID	1
Command	4
Register Address MSB	0
Register Address LSB	0
Register Count MSB	0
Register Count LSB	1
CRC MSB	49 (decimal)
CRC LSB	202 (decimal)

If the value of the temperature parameter is 2764, the device sends the bytes below:

Slave ID	1
Command	4
Byte Count	2
Data MSB	10 (decimal)
Data LSB	204 (decimal)
CRC MSB	191 (decimal)
CRC LSB	197 (decimal)

Byte Count = 2. (The number of bytes of Received Data)

The temperature parameter must muptiply with 0.01
For example;
2764*0.01 = 27.64°C

Example-2 : Reading the value of the **Relative Humidity**

Slave ID	**:** 1
Command	**:** 4 (Read Input Register)
Register Address	**:** 1
Register Count	**:** 1

Slave ID	1
Command	4
Register Address MSB	0
Register Address LSB	1
Register Count MSB	0
Register Count LSB	1
CRC MSB	96 (decimal)
CRC LSB	10 (decimal)

If the value of the relative humidity parameter is 417, the device sends the bytes below:

Slave ID	1
Command	4
Byte Count	2
Data MSB	1
Data LSB	161 (decimal)
CRC MSB	121 (decimal)
CRC LSB	24 (decimal)

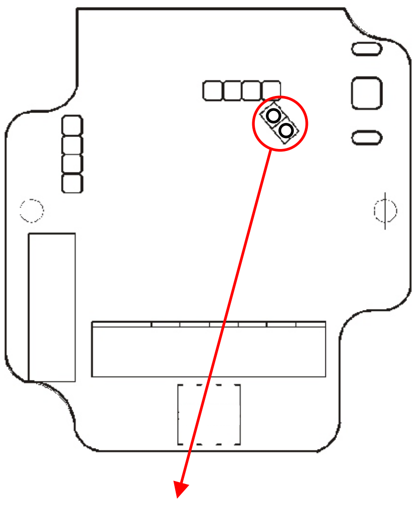
Byte Count = 2. (The number of bytes of Received Data)

The relative humidity parameter must muptiply with 0.1
For example;
417*0.1 = 41.7%

5. Default Parameters

Default parameters;

Slave address	: 1
Baud rate	: 9600 kbps
Parity	: None
Stop bit	: 1 bit



If you want to return default parameters, please short circuit pads that showing you above at least 1 second.