DAE Instrument Corp.

AMR100(c11)

Auto Meter Reading Module

Modbus Reference

Revision 1.1e
Updated 2012/10/19

Introduction

The AMR100 uses the Modbus/RTU protocol. The communications interface is RS485. The baud rate can be 1200, 2400, 4800 or 9600 and set from the front panel. The data format is 8 bits, no parity, 1 stop bit. The device address can be from 0~255 and is also set from the front panel.

This document applies to the AMR100 with firmware version c11 or higher.

All numerical data in this manual is in integer form.

All numerical values are in decimal unless otherwise specified or appended with an ‘h’, in which case the data is in hexadecimal.

When a command is in error, the AMR100 will not respond and simply allow the host PC doing the reading to time out. The AMR100 should have a maximum latency of 300 milliseconds, this is the guaranteed time in which the AMR100 should respond, if this time is exceeded, the host PC should issue a time out.

A command is in error in any of these conditions:

1. The function code is not supported.
2. The data is malformed or out of range.
3. The CRC is wrong.

Register Tables

Function Code	Register Address	Modscan	Description	Range/Unit
3	0	03:0001	Read total pulses	0~999,999,999 counts
	1	03:0002		
	208	03:0209	Read pulse divider	4, 8, 10, 16, 32, 64, 100, 160, 320, 640, 1000, 2000, 10000, 20000
	210	03:0211	Read device address (high byte)	0~255
	215	03:0215	Read decimal place (high byte)	0 = "X", 1 = "X.X", 2 = "X.XX"
	220	03:0221	Read number of displayed digits (high byte)	6~9 displayed digits
16	0	---	Write total pulses	0~999,999,999 counts
	1			
	210	---	Write device address (high byte)	0~255
	214	---	Write pulse divider index (low byte)	0~14 => [4, 8, 10, 16, 32, 64, 100, 160, 320, 640, 1000, 2000, 10000, 20000]
	215	---	Set decimal place (high byte)	0 = "X", 1 = "X.X", 2 = "X.XX"
	220	---	Set number of displayed digits (high byte)	6~9 displayed digits

Message Frames

Read Total Pulses

Format

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	3	0	0	0	2	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Byte Count	Read Data Word 1		Read Data Word 2		CRC	
			high	low	high	low	low	high
0~255	3	4	PULSE-3	PULSE-4	PULSE-1	PULSE-2	CRC-L	CRC-H

Total Pulses = (PULSE-1 x 16,777,216 + PULSE-2 x 65,536 + PULSE-3 x 256 + PULSE-4) x Counts

Example

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
1	3	0	0	0	2	196	11

Reply

AMR100 Address	Function Code	Byte Count	Read Data Word 1		Read Data Word 2		CRC	
			PULSE-3	PULSE-4	PULSE-1	PULSE-2	low	high
1	3	4	226	64	0	1	12	95

Total Pulses = (0 x 16,777,216 + 1 x 65,536 + 226 x 256 + 64) x Counts = 123,456 counts

Write Total Pulses

Format

Query

PULSE-1 = Total_Pulses div 16,777,216

PULSE-2 = (Total_Pulses mod 16,777,216) div 65,536

PULSE-3 = (Total_Pulses mod 65,536) div 256

PULSE-4 = Total_Pulses mod 256

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word 1		Write Data Word 2		CRC	
		high	low	high	low		high	low	high	low	low	high
0~255	16	0	0	0	2	4	PULSE-3	PULSE-4	PULSE-1	PULSE-2	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	16	0	0	0	2	CRC-L	CRC-H

Example

Query

Total_Pulses = 123,456

PULSE-1 = 123,456 div 16,777,216 = 0

PULSE-2 = (123,456 mod 16,777,216) div 65,536 = 1

PULSE-3 = (123,456 mod 65,536) div 256 = 226

PULSE-4 = 123,456 mod 256 = 64

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word 1		Write Data Word 2		CRC	
		high	low	high	low		PULSE-3	PULSE-4	PULSE-1	PULSE-2	low	high
1	16	0	0	0	2	4	226	64	0	1	5	195

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
1	16	0	0	0	2	65	200

Read Pulse Divider

See “Write Pulse Divider Index” for a note of the difference between reading and writing.

Format

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	3	0	208	0	1	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Byte Count	Read Data		CRC	
			high	low	low	high
0~255	3	2	PL	PH	CRC-L	CRC-H

Pulse Rate = PH * 256 + PL

* Note that the high and low bytes are reversed, which is the opposite of the normal Modbus conventions.

Example

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
1	3	0	208	0	1	133	243

Reply

AMR100 Address	Function Code	Byte Count	Read Data		CRC	
			PL	PH	low	high
1	3	2	232	3	182	69

Pulse Rate = 3 x 256 + 232 = 1000

Write Pulse Divider Index

Note that writing the pulse rate parameter is different from the reading of the pulse rate parameter in two ways.

1. The register address used is different, reading uses 208, while writing uses 214. This is purposely done so that
2. The parameter format is also different, reading reads the actual value, while writing is done through an index.

Format

Query

Pulse Rate	PRI (Pulse Rate Index)
1	0
4	1
8	2
10	3
16	4
32	5
64	6
100	7
160	8
320	9
640	10
1000	11
2000	12
10000	13
20000	14

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
0~255	16	0	214	0	1	2	0	PRI	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	16	0	214	0	1	CRC-L	CRC-H

Example

Query

Pulse Rate = 2000; PRI = 12

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	PRI	low	high
1	16	0	214	0	1	2	0	12	181	163

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
1	16	0	214	0	1	224	49

Read Device Address

Note that after executing this command, the AMR100 will no longer respond to the previous address.

Subsequent commands should use the new address that has just been assigned.

Format

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	3	0	210	0	1	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Byte Count	Read Data		CRC	
			high	low	low	high
0~255	3	2	ADDR	0	CRC-L	CRC-H

Example

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
1	3	0	210	0	1	36	51

Reply

AMR100 Address	Function Code	Byte Count	Read Data		CRC	
			ADDR	low	low	high
1	3	2	29	0	177	20

Device Address = ADDR = 29

Write Device Address

This command overwrites the existing Modbus device address of the AMR100. Once you issue this command, the original address is lost. Therefore, the next time commands are issued to the same AMR100, the new device address must be used, it will no longer respond to the old address, unless of course the new address is the same as the old.

Note that this address can also be changed through the front panel.

Format

Query

ADDR = Device Address

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
0~255	16	0	210	0	1	2	ADDR	0	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	16	0	210	0	1	CRC-L	CRC-H

Example

Query

Device Address = 25; ADDR = 25

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		ADDR	low	low	high
1	16	0	210	0	1	2	25	0	191	178

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
1	16	0	210	0	1	161	240

Read Decimal Place

Format

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	3	0	215	0	1	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Byte Count	Read Data		CRC	
			high	low	low	high
0~255	3	2	DP	0	CRC-L	CRC-H

DP	Decimal Place
0	no decimal place: “X”
1	one decimal place: “X.X”
2	two decimal places: “X.XX”

Example

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
1	3	0	215	0	1	52	50

Reply

AMR100 Address	Function Code	Byte Count	Read Data		CRC	
			DP	low	low	high
1	3	2	2	0	185	36

DP = 2; two decimal places: “X.XX”

Set Decimal Place

Format

Query

DP	Decimal Place
0	no decimal place: “X”
1	one decimal place: “X.X”
2	two decimal place: “X.XX”

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
0~255	16	0	215	0	1	2	DP	0	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	16	0	215	0	1	CRC-L	CRC-H

Example

Query

Decimal Place = 2; DP = 2

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		DP	low	low	high
1	16	0	215	0	1	2	2	0	181	23

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
1	16	0	215	0	1	177	241

Read Number of Displayed Digits

The AMR100 has a 6 digit LED display. In order to display more digits, it will have to scroll the display to the left. The extends the capacity of the display to show as many as 9 digits. This command reads this parameter. See the related command “Set Number of Displayed Digits” to set this same parameter.

Format

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	3	0	220	0	1	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Byte Count	Read Data		CRC	
			high	low	low	high
0~255	3	2	NDD	0	CRC-L	CRC-H

NDD	Number of Displayed Digits
6	Display 6 digits: “XXX XXX”
7	Display 7 digits: “X XXX XXX”
8	Display 8 digits: “XX XXX XXX”
9	Display 9 digits: “XXX XXX XXX”

Example

Query

AMR100 Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
1	3	0	220	0	1	69	240

Reply

AMR100 Address	Function Code	Byte Count	Read Data		CRC	
			NDD	low	low	high
1	3	2	7	0	186	116

NDD = 7; Number of Displayed Digits = 7

Set Number of Displayed Digits

The AMR100 has a 6 digit LED display. In order to display more digits, it will have to scroll the display to the left. The extends the capacity of the display to show as many as 9 digits. This command sets this parameter. See the related command “Read Number of Displayed Digits” to read back this same parameter.

Format

Query

NDD	Number of Displayed Digits
6	Display 6 digits: “XXX XXX”
7	Display 7 digits: “X XXX XXX”
8	Display 8 digits: “XX XXX XXX”
9	Display 9 digits: “XXX XXX XXX”

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
0~255	16	0	220	0	1	2	NDD	0	CRC-L	CRC-H

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	16	0	220	0	1	CRC-L	CRC-H

Example

Query

Set Number of Displayed Digits to 8; NDD = 8

AMR100 Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		NDD	low	low	high
1	16	0	220	0	1	2	8	0	178	204

Reply

AMR100 Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
1	16	0	220	0	1	192	51

CRC Computation

The AMR100 conforms to the Modbus/RTU protocol and thus uses CRC16 for its error checking. The computed CRC is appended to the end of the message with the LSB first and then the MSB. Below is the pseudo code for computing the CRC as used by the standard Modbus/RTU. The pseudo code is written in the Ruby language and can be directly used as such.

Definition

```
def get_crc (*byte_array)
    sum = 0xFFFF
    byte_array.each do |byte|
      sum ^= byte
      8.times do
        carry = (1 == sum & 1)
        sum = 0x7FFF & (sum >> 1)
        sum ^= 0xA001 if carry
    end
  end
  return [sum & 0xFF, sum >> 8]
end
```

Usage

```
>> crc = get_crc(1,3,0,141,0,5)
=> [21, 226]        <---- [CRC low byte, CRC high byte]
```

Additional Resources

Although every effort has been taken to ensure that this document is free from errors, some may still remain. If found please send an email to: info@daeinstrument.com, in the subject line write “Errata” and please indicate the name of this document “AMR100 Modbus Reference”, revision number, page number and indicate the error with its correction. Thank you.

We have made sure that this document is as clear and useful to you as possible, but any suggestions on improving this document to serve you even better would be welcome. Send comments and suggestions to: info@daeinstrument.com, in the subject line, write “Comments” and please indicate the name of this document “AMR100 Modbus Reference”. Questions are also welcome.

This document only covers the Modbus protocol registers as used by the AMR100, for hardware interfacing and other information please refer to the AMR100 user’s manual.