DAE Instrument Corp.

Auto Meter Reading Module

Modbus Reference

Revision 1.3e
Updated 2014/6/18

Table of contents

General Information

About the AMR

The AMR uses the Modbus/RTU protocol. The communications interface is RS485. The baud rate can be 1200, 2400, 4800 or 9600 and can be set from the front panel or written with a Modbus command. The factory default is 9600. If this is changed then the current setting can be viewed from the touch panel. The data format is 8 bits, no parity and 1 stop bit.

The device address can be from 0~254 and can be viewed and from the front panel, or by issuing a Modbus command.

Although it is possible to set the device address as 0 (zero) or 255, it is not recommended as it may conflict with other Modbus devices on the same bus that may recognize either as being the broadcast address.

The word pulse when used plainly or with the qualifier scaled, will be used to refer to the pulses that has been scaled with the pulse rate divider, whereas the pulses from the metering devices will be referred to as raw pulses.

The displayed value as well as the accumulated pulse counts both refer to the scaled pulse.

The Modbus broadcast address recognized by the AMR is 255 and not zero. Note that differs from standard Modbus.

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

A command is an error in any of these conditions are met:

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

About this Document

All numerical data returned by the AMR through the Modbus protocol in this manual are integers.

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

This Modbus reference document applies to AMR firmware version 061 and higher. For AMRs with older firmware, please refer to the 1.1 or older revision of this reference.

Register Table

Register Address	Modscan	Description	Word	Format	Range	Range
0	03:0001	Ch 1 pulse accumulator	low	F1	0~999,999 counts	read & write
1	03:0002		high			
2	03:0003	Ch 2 pulse accumulator	low			
3	03:0004		high			
4	03:0005	Ch 3 pulse accumulator	low			
5	03:0006		high			
6	03:0007	Ch 4 pulse accumulator	low			
7	03:0008		high			
8	03:0009	Ch 5 pulse accumulator	low			
9	03:0010		high			
10	03:0011	Ch 6 pulse accumulator	low			
11	03:0012		high			
12	03:0013	Ch 7 pulse accumulator	low			
13	03:0014		high			
14	03:0015	Ch 8 pulse accumulator	low			
15	03:0016		high			
16	03:0017	Ch 9 pulse accumulator	low			
17	03:0018		high			
18	03:0019	Ch 10 pulse accumulator	low			
19	03:0020		high			
20	03:0021	Ch 11 pulse accumulator	low			
21	03:0022		high			
22	03:0023	Ch 12 pulse accumulator	low			
23	03:0024		high			
24	03:0025	Ch 13 pulse accumulator	low			
25	03:0026		high			
26	03:0027	Ch 14 pulse accumulator	low			
27	03:0028		high			
28	03:0029	Ch 15 pulse accumulator	low			
29	03:0030		high			
30	03:0031	Ch 16 pulse accumulator	low			
31	03:0032		high			
208	03:0209	Pulse Rate Divider	data	F2	0~65535	read only
210	03:0211	Device Address	data	F3	high byte = 0~254, low byte = not used	read only
215	03:0216	Decimal Point	data	F4	high byte = 0~2, low byte = not used	read only

Notes:

- The pulse accumulator for all channels is after being scaled by the **Pulse Rate Divider**.
- The maximum count of 999,999 for all channels is after being scaled by the **Pulse Rate Divider**.
- R/W = read or write
- Although the **Device Address** can accept 0 (zero) as its value, it is not recommended using this address, even though the AMR recognizes this as a valid address, other Modbus devices on the same bus may recognize 0 as being the broadcast address.
- Note that although registers 208, 210 and 215 are read only, the parameters represented by these registers can also be written but not through these same address. Writing to them is a more involved process, this is to protect these parameters from inadvertent changes. See their individual message frames and examples for details of the actual writing process.

Conversion Formats

Format	Description	Definition
F1	Channel Pulses (P)	P = {Word(H) x 65536 + Word(L)}
F2	Pulse Rate Divider	Pulse Rate Divider = data
F3	Device Address	Device Address = data % 256 (modulus division)

F4	Decimal Point (DP)	DP = data % 256 (modulus division)
		DP = 0: no decimal position, “X”
		DP = 1: one decimal position, “X.X”
		DP = 2: two decimal position, “X.XX”

Message Frames

Read Channel Pulse Accumulator

The accumulated pulses read through this Modbus command is the scaled pulse count after being divided by the pulse rate divider.

Format

Query

CRA = channel register address (refer to register table)

AMR Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
0~254	3	0	CRA	0	2	CRC-L	CRC-H

Reply

AMR Address	Function Code	Byte Count	Read Data Word 1		Read Data Word 2		CRC	
			high	low	high	low	low	high
0~254	3	4	P3	P4	P1	P2	CRC-L	CRC-H

Total Pulses = (P1 x 16,777,216 + P2 x 65,536 + P3 x 256 + P4)

Example

Query

Channel 10 => CRA = 18

AMR Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
1	3	0	18	0	2	100	14

Reply

AMR Address	Function Code	Byte Count	Data Word 1		Data Word 2		CRC	
			high	low	high	low	low	high
1	3	4	230	231	0	5	189	79

Total Pulses = 0 x 16,777,216 + 5 x 65,536 + 230 x 256 + 231 = 386,791

Write to Channel Pulse Accumulator

Writing to the channel pulse accumulator would overwrite its existing value and is equivalent to initializing the channel with a new value.

The new value written through this Modbus command will be the value shown on the front panel display.

Format

Query

CRA = channel register address (refer to register table)

P1 = Total_Pulses div 16,777,216

P2 = (Total_Pulses mod 16,777,216) div 65,536

P3 = (Total_Pulses mod 65,536) div 256

P4 = Total_Pulses mod 256

AMR Address	Function Code	Register Address		Number of Registers		Byte Count	Data Word 1		Data Word 2		CRC	
		high	low	high	low		high	low	high	low	low	high
0~254	16	0	CRA	0	2	4	P3	P4	P1	P2	CRC-L	CRC-H

Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~254	16	0	CRA	0	2	CRC-L	CRC-H

Example

Query

Channel 10 => CRA = 18

Total_Pulses = 338,856

P1 = 338,856 div 16,777,216 = 0

P2 = (338,856 mod 16,777,216) div 65,536 = 5

P3 = (338,856 mod 65,536) div 256 = 43

P4 = 338,856 mod 256 = 168

AMR Address	Function Code	Register Address		Number of Registers		Byte Count	Data Word 1		Data Word 2		CRC	
		high	CRA	high	low		high	low	high	low	low	high
1	16	0	18	0	2	4	43	168	0	5	59	125

Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
1	16	0	18	0	2	225	205

Read Device Address

The register location for reading the device address does not apply to writing the device address. There is a separate register location and a more involved process to writing the device address that is explained in the section on **Write the Device Address**.

Even though it may seem that having a command to read the device address may make no sense, since issuing the command by itself requires that the device address be already known, thus resulting in a catch-22. One can find out the device address of a device with an unknown address by using the broadcast address. Since broadcasting this command to a bus will cause all the devices on the same bus to respond and thus resulting in collision. It is necessary to isolate the device so that it is connected to the host all alone by itself on the bus.

Although the device address can also be viewed from the front panel display. There is one scenario wherein this command may come and in handy, and that is when the display is unreadable or damaged.

Format

Note that 255 is the broadcast address

Query

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

Reply

AMR Address	Function Code	Byte Count	Read Data Word		CRC	
			high	low	low	high
255	3	2	Device Address	0	CRC-L	CRC-H

Example

Query

AMR Address	Function Code	Starting Register		Number of Registers		CRC	
		high	low	high	low	low	high
255	3	0	208	0	1	144	45

Reply

AMR Address	Function Code	Byte Count	Read Data Word		CRC	
			high	low	low	high
255	3	2	47	0	141	160

Device Address = 47

Write Device Address

Writing the device address is a three step process, this is to protect the device address from being written inadvertently.

As long as the last step of disabling is not performed, writing to the device address remains enabled. It will only be disabled through an explicit command or when the device is reset.

Although it is possible to set the 0 (zero) and 255 as the device address, but it is recommended that these two be avoided as to prevent potential conflicts and errors with other devices that might recognize these as being a broadcast address.

Note that since this command affects the communication itself, after the second step, the new address must be used instead of the old address for the third step. See the example below for details of how this works.

The device address can also be changed via the front panel.

The three step process is illustrated in figure 1 as shown below.

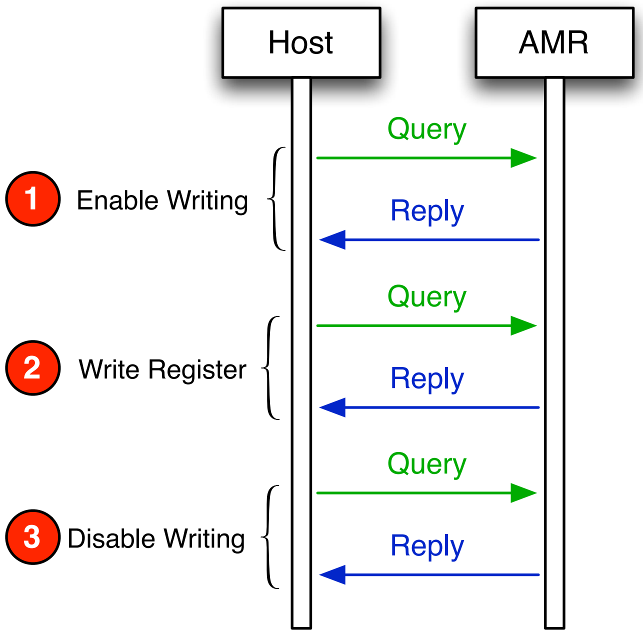

Figure 1
Three Step Parameter Changing Sequence

Format

Enable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~255	5	0	48	0	0	CRC-L	CRC-H

Write Register - Query

AMR 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	48	0	1	2	new device address	0	CRC-L	CRC-H

Write Register - Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	16	0	48	0	1	CRC-L	CRC-H

Disable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
255 or new device address	5	0	48	255	0	CRC-L	CRC-H

Example

This example sets up a one to one connection between the AMR and the PC host with no other devices on the RS485 bus. The address used is the broadcast address of 255. This scenario is the set up that is typically used when a PC (often a notebook) is used as an AMR address setting tool. By using 255 as the address, whatever existing address on the AMR can be ignored.

Although the address can also be changed from the front panel. Sometimes it is preferred that this is done programmatically or when the display on the front panel is unreadable or preferred not to be used.

Enable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
255	5	0	48	0	0	216	27

Write Register - Query

New Device Address = 135

AMR Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
255	16	0	48	0	1	2	**135**	0	136	52

Write Register - Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
255	16	0	48	0	1	20	24

✳ Note that at this point the AMR address is now 135. But since we are using the broadcast address, we can still communicate with the AMR regardless The new device address will take effect only after the entire sequence is completed.

Disable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
255	5	0	48	255	0	153	235

Change Baud Rate

Changing the baud rate is a three step process, this is to protect this parameter from inadvertent changes.

As long as the last step of disabling is not performed, writing to the device address remains enabled. It will only be disabled through an explicit command or when the device is reset.

It is also possible to change the baud rate via the front panel.

Note that since this command affects the communication itself, after the second step, the new address must be used instead of the old address for the third step. See the example below for details of how this works.

The three step process is illustrated in figure 1 as shown in previous pages.

There is no equivalent command to read the baud rate. As this would result in a catch-22, since to read the baud rate, you would already have known the baud rate beforehand. If the display is damaged or unreadable and the baud rate is unknown. One can only indirectly find out the baud rate by trying each of the valid baud rates until the AMR responds correctly.

Format

Enable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~255	5	0	55	0	0	CRC-L	CRC-H

Write Register - Query

Baud Rate	BR Index
9600	0
4800	1
2400	2
1200	3

AMR 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	55	0	1	2	BR Index	0	CRC-L	CRC-H

Write Register - Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~255	16	0	55	0	1	CRC-L	CRC-H

Disable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~255	5	0	55	255	0	CRC-L	CRC-H

Example

This example sets up a one to one connection between the AMR and the PC host with no other devices on the RS485 bus. The address used is the broadcast address of 255. This scenario is the set up that is typically used when a PC (often a notebook) is used as an AMR baud rate setting tool. By fixing the address at 255, whatever existing address on the AMR can be ignored and only one variable needs to be varied for the auto baud process.

** The auto baud process is an algorithm that tries to find the correct baud rate through trial and error.*

Although the baud rate can also be changed from the front panel. Sometimes it is preferred that this is done programmatically or when the display on the front panel is unreadable or preferred not to be used.

Enable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
255	5	0	55	0	0	105	218

Write Register - Query

New Baud Rate = 1200 => BR Index = 3

AMR Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
255	16	0	55	0	1	2	**3**	0	234	131

Write Register - Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
255	16	0	55	0	1	165	217

✳ Note that at this point the AMR has switched to the new baud rate for communication. The PC software must switch as well or subsequent packets will time out.

Disable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
255	5	0	55	255	0	40	42

Read Pulse Rate

The pulse rate is a divider for the raw pulse counts. It affects both the LED displayed value on the front panel as well as the pulse counts obtained from the **Channel Pulse Accumulator** command.

The register address for reading the pulse rate does not apply to writing the pulse rate. There is a separate location and a more involved process to writing the pulse rate that is explained in the section on **Write Pulse Rate**.

Format

Query

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~254	3	0	208	0	1	CRC-L	CRC-H

Reply

AMR Address	Function Code	Byte Count	Read Data Word		CRC	
			high	low	low	high
0~254	3	2	P1	P2	CRC-L	CRC-H

Pulse Rate = P1 x 256 + P2

Predefined Pulse Rate values:

P1 (high byte)	P2 (low byte)	Pulse Rate
0	1	1
0	4	4
0	8	8
0	10	10
0	16	16
0	32	32
0	64	64
0	100	100

P1 (high byte)	P2 (low byte)	Pulse Rate
0	160	160
1	64	320
2	128	640
3	232	1000
7	208	2000
39	16	10000
78	32	20000

Example

Query

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

Reply

AMR Address	Function Code	Byte Count	Read Data Word		CRC	
			high	low	low	high
1	3	2	2	128	184	132

Pulse Rate = 2 x 256 + 128 = 640

Write Pulse Rate

Changing the baud rate is a three step process, this is to protect this parameter from accidental changes. These steps must be carried out completely in the proper sequence in one go without any intervening commands. Skipping or interrupting the sequence will invalidate the writing.

This command cannot be performed using the Modscan software because Modscan will continuously poll the device, interrupting any attempts to do an atomic sequential write.

Note that it is also possible to change the pulse rate via the front panel.

The three step process is illustrated in figure 1 as shown in previous pages.

Format

Enable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~254	5	0	56	0	0	CRC-L	CRC-H

Write Register - Query

Pulse Rate	PR Index
1	0
4	1
8	2
10	3
16	4
32	5
64	6
100	7

Pulse Rate	PR Index
160	8
320	9
640	10
1000	11
2000	12
10000	13
20000	14

AMR Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
0~254	16	0	56	0	1	2	PR Index	0	CRC-L	CRC-H

Write Register - Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~254	16	0	56	0	1	CRC-L	CRC-H

Disable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~254	5	0	56	255	0	CRC-L	CRC-H

Example

Pulse Rate = 320 => PR Index = 9

Enable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
1	5	0	56	0	0	76	7

Write Register - Query

AMR Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
1	16	0	56	0	1	2	9	0	164	184

Write Register - Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
1	16	0	56	0	1	128	4

Disable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
1	5	0	56	255	0	13	247

Read Decimal Point

The decimal point is the the position of the dot shown on the display of the front panel that represents the decimal position of the displayed value. It is used in conjunction with the pulse rate divisor to scale the raw pulse counts to make it more meaningful to the user or operator by making it represent the displayed unit as close as possible. It is cosmetic and only affects the value shown on the LED display, it does not in any way affect the accumulated pulse counts as obtained from the **Read Channel Pulse Accumulator** command.

Format

Query

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~254	3	0	215	0	1	CRC-L	CRC-H

Reply

AMR Address	Function Code	Byte Count	Read Data Word		CRC	
			high	low	low	high
0~254	3	2	DP Index	0	CRC-L	CRC-H

Decimal Point

DP Index	Decimal Point
0	X
1	X.X
2	X.XX

Example

Query

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

Reply

AMR Address	Function Code	Byte Count	Read Data Word		CRC	
			high	low	low	high
1	3	1	2	0	73	36

DP Index = 2 => Decimal Point = “X.XX”

Change Decimal Point

Changing the decimal point is a three step process, this is to protect the unit from unintentional or unauthorized changes. These steps must be carried out completely in the proper sequence in one go without any intervening commands. Skipping or interrupting the sequence will invalidate the writing.

This command cannot be performed using the Modscan software because Modscan will continuously poll the device, interrupting any attempts to do an atomic sequential write.

Note that it is also possible to change the decimal point via the front panel.

The three step process is illustrated in figure 1 as shown in previous pages.

Format

Enable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~254	5	0	57	0	0	CRC-L	CRC-H

Write Register - Query

Decimal Point	DP Index
X	0
X.X	1
X.XX	2

AMR Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
0~254	16	0	57	0	1	2	DP Index	0	CRC-L	CRC-H

Write Register - Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
0~254	16	0	57	0	1	CRC-L	CRC-H

Disable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~254	5	0	57	255	0	CRC-L	CRC-H

Example

Enable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
1	5	0	57	0	0	29	199

Write Register - Query

Decimal Point Position = “X.XX” => DP Index = 2

AMR Address	Function Code	Register Address		Number of Registers		Byte Count	Write Data Word		CRC	
		high	low	high	low		high	low	low	high
1	16	0	57	0	1	2	2	0	162	89

Write Register - Reply

AMR Address	Function Code	Register Address		Number of Registers		CRC	
		high	low	high	low	low	high
1	16	0	57	0	1	209	196

Disable Writing - Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
1	5	0	57	255	0	92	55

Change the Display Cycling

The display can either statically display one selected channel or automatically cycle through all the channels.

This feature can also be changed through the front panel.

Format

Cycle	Action	Value
Yes	Cycle display through all channels automatically	0
No	Fix the display on the last selected channel	255

Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~254	5	0	58	Value	0	CRC-L	CRC-H

Example 1 - Cycle Display

Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
1	5	0	58	0	0	237	199

Example 2 - Fix Display

Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
1	5	0	58	255	0	172	55

Change Supported Meter Type

The AMR can support either electricity meters or water meters. Due to the characteristic of the pulses generated by each type of meter, they are sufficiently different that each needs to be treated in its own way by the AMR for optimum efficacy.

This feature can also be changed through the front panel.

Format

Meter Type	Value
Electricity	0
Water	255

Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
0~254	5	0	59	Value	0	CRC-L	CRC-H

Example 1 - Support Electricity Meter

Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
1	5	0	59	0	0	188	7

Example 2 - Support Water Meter

Query and Reply

AMR Address	Function Code	Register Address		Force Control		CRC	
		high	low	high	low	low	high
1	5	0	59	255	0	253	247

CRC Computation

The AMR 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 a 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 “AMR 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 “AMR Modbus Reference”. Questions are also welcome.

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