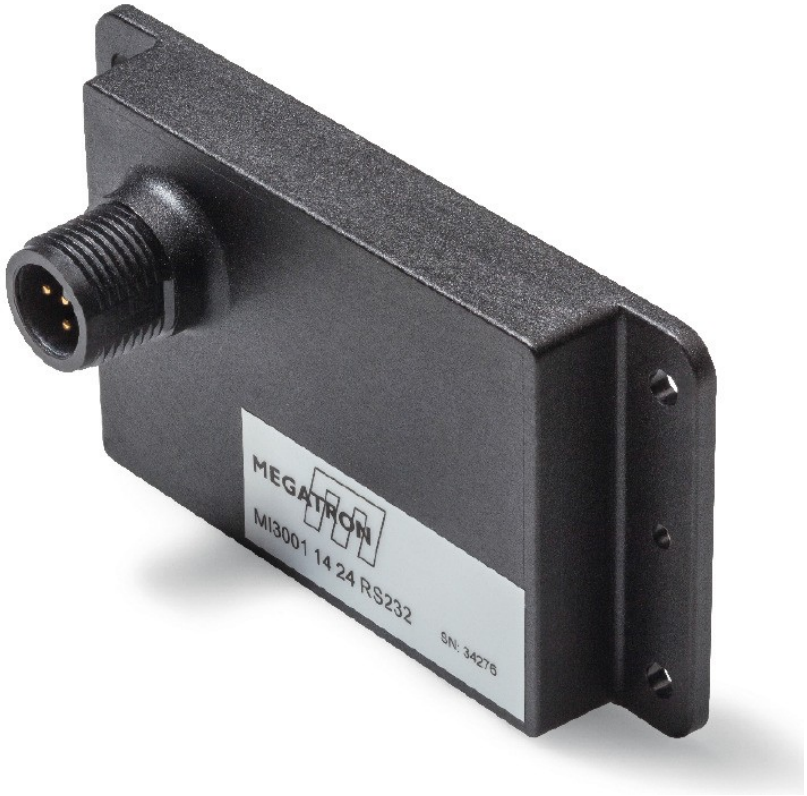
MI Modbus Communication User Guide

Version 1.4
8/8/2018



Megatron Elektronik GmbH
Hermann-Oberth-Straße 7
85640 Putzbrunn / München
www.megatron.de

Errors and specifications subject to change without notice

MI Modbus Communication User Guide

Contents

Notes:

Amendments

Date	Comment(s)
8/8/2018	Removed CAN version. Version 1.4
1/3/2012	Added Register 07 -Temperature. Firmware version v1.1 or later. Version 1.3
11/21/2011	Updated command response times. Version 1.2
9/29/2011	Corrected device address range. Version 1.1
5/16/2011	MI Modbus Communication User Guide. Version 1.0

MI Modbus Communication User Guide

1 General

The **MI** MEMS Digital Inclinometer is available with several interface options: US Digital standard serial protocol over RS232 or RS485 (**MI-1-232** or **MI-1-485**) or Modbus RTU protocol over RS232 or RS485 (**MI-1-MOD2** or **MI-1-MOD4**).

This document describes how to communicate with a **MI-1-MOD2** or **MI-1-MOD4**.

- A **MI-1-MOD2** uses RS232 and supports only a single device on the RS232 bus.
- A **MI-1-MOD4** uses a Modbus RTU compatible protocol over RS485. A USB-to-RS485 adapter is typically used to connect to a PC. Up to 32 **MI-1-MOD4** devices can be connected to the same RS485 bus.

The following section provides some basic information about the serial communication between the host computer and a **MI-1-MODx**.

- The full Modbus specification can be obtained from http://www.modbus.org. Modbus is a command/response protocol over a serial bus.
- The default Modbus serial parameters are: 9600 baud, 1start bit, 8 data bits, even parity and 1 stop bit. The 8 data bits are sent LSB first. The baud rate can be changed to 115200, 57600, 38400, 19200 or 9600 by sending the appropriate command.
- The byte order for all 16-bit values is Big Endian (most significant byte first).
- Read and write access to the **MI-1-MODx** is done using Modbus Function Code 3 (Read Holding Registers) and Modbus Function Code 6 (Write Single Register) commands. These two function codes provide the basic functionality needed by most users of the MI. A User Defined Modbus Function Code 110 is provided for less commonly used, off-line functions such as setting serial port parameters and changing the device address.
- Modbus device address must be in the range 1 to 100 or 127 (decimal). All MI's are shipped with a default address of 127 (decimal). Address 0 is the Modbus broadcast address. All devices will perform the action of the function code, but no reply will be sent.
- All Modbus commands and responses have a 16-bit CRC for error detection. C source code for the CRC-16 calculation is available from the Modbus website (http://www.modbus.org) in the *MODBUS over serial line specification and implementation guide V1.02* document.
- Modbus RTU data is in binary format rather than ASCII, so it cannot be viewed properly on a text terminal. US Digital's free ComTool application or the open source Realterm program may be used to view binary serial data on a PC. ComTool can also calculate the Modbus CRC-16.
- The MI only responds to commands from the host. It does not initiate communication and cannot be configured to respond periodically.
- At 9600 baud, the MIs response time is approximately 5.7 milliseconds for "Read" commands and about 27.3 milliseconds for "Write" commands. At 115200 baud, the response time is approximately 550 microseconds for "Read" commands and about 22.7 milliseconds for "Write" commands. These delays include the required Modbus RTU framing delay.
- A command from the host must be less than 500 milliseconds in duration (first byte to last byte) or the MI will silently discard the command.

2 MI-1-MODx Registers

The following table lists all the **MI-1-MODx** registers accessible using Function Code 3 (Read Holding Registers) and Function Code 6 (Write Single Register). All registers are 16-bits wide. On some Modbus hosts, the base address of the registers may be at a decimal address such as 30000 or 40000, but the address offset from the base is what is actually transmitted.

Register	Description	Read Holding Registers (Function code 3)	Write Single Register (Function Code 6)
0x0000	Reported Angle (lower 16-bits)	Get the lower 16-bits of the reported angle, including angle offset.	Calculates and sets and angle offset such that the reported angle equals the specified angle (lower 16-bits)
0x0001	Reported Angle (upper 16-bits)	Get the upper 16-bits of the reported angle, including angle offset.	Calculates and sets and angle offset such that the reported angle equals the specified angle (upper 16-bits)
0x0002	Angle Offset (lower 16-bits)	Get the lower 16-bits of the angle offset. Factory default is 0x0000	Set the lower 16-bits of the angle offset
0x0003	Angle Offset (upper 16-bits)	Get the upper 16-bits of the angle offset. Factory default is 0x0000	Set the upper 16-bits of the angle offset
0x0004	Damping Time (16-bits)	Get the damping time in milliseconds (2-5000). Factory default is 1000	Set the damping time in milliseconds. Must be 2-5000
0x0005	Angle Direction (16-bits)	Get Angle Direction Mode 0: Normal (factory default) 1: Reversed	Set Angle Direction Mode 0: Normal 1: Reversed
0x0006	Angle Output Mode (16-bits)	Get Angle Output Mode 0: angle output is -180000 to 179999 (factory default) 1: angle output is 0 to 359999	Set Angle Output Mode 0: angle output is -180000 to 179999 1: angle output is 0 to 359999
0x0007	Temperature (16-bits)	Get the device temperature in °C	Error. Will return Function Code 0x86 with “invalid register address”

All angles are represented as signed (2’s complement) 32-bit integers in units of .001 degrees. For example, an angle value of 0x000237ac (hex) = 145324 (decimal) corresponds to an angle of 145.324 degrees. 0x0002 is the “upper 16-bits” and 0x37ac is the “lower 16-bits”. An angle of -145.324 is represented as 0xfffdc854.

The temperature is returned as a signed (2’s complement) 16-bit integer in units of 0.01 degrees Celsius. For example, a temperature value of 0x096c (hex) = 2412 (decimal) corresponds to a temperature of 24.12C. A temperature of -5.23C is represented as 0xfdf5. The device temperature is warmer than the actual ambient temperature due to self-heating from the electronics.

Since all Modbus registers are 16-bits wide, all angles are stored in two consecutive registers. For 32-bit values such as the reported angle or angle offset, reading the lower or upper 16-bits will latch the entire 32-bit value so that the value does not change between the two reads. On a write of a 32-bit quantity, write the lower 16-bits first, followed by the upper 16-bits. The update of the 32-bit quantity happens after the upper 16-bits are written.

The range of the reported angle depends on the Angle Output Mode register:

Angle Output mode	**Reported Angle**
0 (bidirectional)	-180000 to 179999
1 (unidirectional)	0 to 359999

The rotation sense of the angle depends on the Angle Direction register:

Angle Direction	**Rotation Sense**
0	normal
1	reversed

The reported angle can have an angle offset parameter included as follows:

Reported_Angle = Absolute_Angle + Angle_Offset

The Angle Offset register can be written directly to affect the Reported Angle register. Alternately, by *writing* a desired angle to the Reported Angle register, the MI will calculate and update Angle Offset register so that the Reported Angle becomes the desired angle.

Note that the registers for Angle Offset, Damping Time, Angle Output Mode and Angle Direction are non-volatile. Writing to these registers will store the value in flash memory so the setting will be retained when power is turned off.

The next section describes the byte level contents of each Modbus RTU command frame and response frame.

3 MI Modbus RTU Command/Response Data Formats

3.1 *Read Holding Register (Function Code 3)*

The Modbus "Read Holding Register" command is used to read registers in the **MI-1-MODx**

	Parameter	# bytes	Units	Notes
Command	Address	1		Destination address of MI
	0x03	1		Modbus Function Code = 0x03
	Starting register to read (hex)	2		Big-endian order. Ex. Register 0x0004 (Damping Time) is sent 0x00 first, then 0x04
	Number of registers to read (hex)	2		There are 8 registers supported in the MI-1-MODx. All 8 can be read at once with one command.
	CRC-16	2		CRC-16 of all bytes in the command
Response	Address	1		Address of responding MI
	0x03	1		Modbus Function Code = 0x03
	Byte Count	1		2*(Number of Registers Read). Ex. 0x10 if all 8 registers are read.
	Register N	2		Read Register data. 2 bytes if one register is read. 16 bytes if all 8 registers are read
	Register N+1	2		
	Register N+2	2		
	…	2		
	CRC-16	2		CRC-16 of all bytes in the response.

If the CRC of the command is invalid, no action is taken. If the CRC is valid but there is an error in the command, the response is:

	Parameter	# bytes	Units	Notes
Error Response	address	1		Address of responding MI
	0x83	1		Modbus Error Function Code = 0x83
	Exception Code	1		0x01: invalid function code 0x02: invalid register address
	CRC-16	2		CRC-16 of all bytes in the response.

Example 1 (Read the damping time from MI with address 127):

```
            address
            |  fcode
            |  |  starting reg. to read (0x0004)
            |  |  |     number of reg. to read (0x0001)
            |  |  |     |
            |  |  |     |     CRC-16
            |  |  |     |     |
Command:    7f 03 00 04 00 01 cf d5

            address
            |  fcode
            |  |  byte count
            |  |  |  damping (0x03e8 = 1000 decimal (1000 milliseconds)
            |  |  |  |     CRC-16
            |  |  |  |     |
Response:   7f 03 02 03 e8 90 f0
```

Example 2 (Read the lower/upper 16-bits of Angle 2 from MI with address 127):

```
            address
            |  fcode
            |  |  starting reg. to read (0x0000)
            |  |  |     number of reg. to read (0x0002)
            |  |  |     |
            |  |  |     |     CRC-16
            |  |  |     |     |
Command:    7f 03 00 00 00 02 ce 15

            address
            |  fcode
            |  |  byte count
            |  |  |  angle2 (0x000197c5 = 104389 decimal (104.389 degrees)
            |  |  |  |           CRC-16
            |  |  |  |           |
Response:   7f 03 04 97 c5 00 01 99 bd
```

3.2 *Write Single Register (Function Code 6)*

The Modbus “Write Single Register” command is used to write to a register in the **MI-1-MODx**. If the command is successful, the response is identical to the command. Two “Write Single Register“commands are needed to update the Reported Angle or Angle Offset values since these are 32-bit quantities.

	Parameter	**# bytes**	**Units**	**Notes**
Command	Address	1		Destination address of MI
	0x06	1		Modbus Function Code = 0x06
	Register to Write (hex)	2		Big-endian order. Ex. Register 0x0004 (Damping Time) is sent 0x00 first, then 0x04
	Data to Write	2		16-bit data to write
	CRC-16	2		CRC-16 of all bytes in the command
Response (identical to command)	Address	1		Address of responding MI
	0x06	1		Modbus Function Code = 0x06
	Register to Write (hex)	2		Big-endian order. Ex. Register 0x0004 (Damping Time) is sent 0x00 first, then 0x04
	Data to Write	2		16-bit data to write
	CRC-16	2		CRC-16 of all bytes in the response.

If the CRC of the command is invalid, no action is taken. If the CRC is valid but there is an error in the command, the response is:

	Parameter	**# bytes**	**Units**	**Notes**
Error Response	address	1		Address of responding MI
	0x86	1		Modbus Error Function Code = 0x86
	Exception Code	1		0x01: invalid function code 0x02: invalid register address
	CRC-16	2		CRC-16 of all bytes in the response.

Example 1 (Set the damping time to 2000 milliseconds for MI with address 127):

```
              address
              |  fcode
              |  |  reg. to write (0x0004)
              |  |  |     data to write (2000 decimal = 0x07d0)
              |  |  |     |
              |  |  |     |     CRC-16
              |  |  |     |     |
   Command:   7f 06 00 04 07 d0 c1 b9

   Response:  7f 06 00 04 07 d0 c1 b9 (identical to Command)
```

Example 2 (Set the current Reported Angle to 0 degrees for MI with address 127. 2 commands needed):

```
              address
              |  fcode
              |  |  reg. to write (0x0000 – lower 16-bits of Reported Angle Register)
              |  |  |     data to write (0x0000)
              |  |  |     |
              |  |  |     |     CRC-16
              |  |  |     |     |
   Command:   7f 06 00 00 00 00 83 d4

   Response:  7f 06 00 00 00 00 83 d4 (identical to Command)


              address
              |  fcode
              |  |  reg. to write (0x0001 - upper 16-bits of Reported Angle Register)
              |  |  |     data to write (0x0000)
              |  |  |     |
              |  |  |     |     CRC-16
              |  |  |     |     |
   Command:   7f 06 00 01 00 00 d2 14


   Response:  7f 06 00 01 00 00 d2 14 (identical to Command)
```

3.3 *User Defined Function Code 110 – Additional Functions*

User defined Function Code 110 is for less commonly used, off-line functions such as setting serial port parameters and changing the device address. Some Modbus hosts cannot support this Function code, but the frames can still be sent using a serial terminal program on a PC that can send binary data on the serial port. Changes to the serial port parameters and device address are stored in flash memory so these settings are retained when power is off.

3.3.1 Set Serial Port Baud Rate

	Parameter	**# bytes**	**Units**	**Notes**
Command	Address	1		Destination address of MI
	0x6e	1		Modbus Function Code = 0x6e
	0x04	1		Command length
	0x8f	1		MI Command
	Baud Rate	1		0 = 115.2 kbps 1 = 57.6 kbps 2 = 38.4 kbps 3 = 19.2 kbps 4 = 9.6 kbps
	CRC-16	2		CRC-16 of all bytes in the command
Response	Address	1		Address of responding MI
	0x6e	1		Modbus Function Code = 0x6e
	0x04	1		Command length
	0x8f	1		MI Command
	Status	1		0 = Command successful Otherwise = Command failed
	CRC-16	2		CRC-16 of all bytes in the response.

The Response is sent using the previous baud rate. The baud rate is changed about 10 milliseconds after a successful response is sent.

Example 1 (Set Baud rate to 9.6 kbps for MI with address 127):

```
            address
            |  fcode
            |  |
            |  |          baud rate (0x04 = 9.6 kbps)
            |  |          |
            |  |          |  CRC-16
            |  |          |  |
Command:    7f 6e 04 8f 04 09 10

            address
            |  fcode
            |  |
            |  |         status (0 = command successful)
            |  |         |  CRC-16
            |  |         |  |
Response:   7f 6e 04 8f 00 08 d3
```

Example 2 (Set Baud rate to 115.2 kbps for MI with address 127):

```
            address
            |  fcode
            |  |
            |  |          baud rate (0x00 = 115.2 kbps)
            |  |          |
            |  |          |  CRC-16
            |  |          |  |
Command:    7f 6e 04 8f 00 08 d3

            address
            |  fcode
            |  |
            |  |         status (0 = command successful)
            |  |         |  CRC-16
            |  |         |  |
Response:   7f 6e 04 8f 00 08 d3
```

3.3.2 Set Serial Port Parity

	Parameter	# bytes	Units	Notes
Command	Address	1		Destination address of MI
	0x6e	1		Modbus Function Code = 0x6e
	0x04	1		Command length
	0x93	1		MI Command
	Parity	1		0 = no parity, 1 stop bit 1 = no parity, 2 stop bits 2 = even parity, 1 stop bit (default) 3 = odd parity, 1 stop bit
	CRC-16	2		CRC-16 of all bytes in the command
Response	Address	1		Address of responding MI
	0x6e	1		Modbus Function Code = 0x6e
	0x04	1		Command length
	0x93	1		MI Command
	Status	1		0 = Command successful Otherwise = Command failed
	CRC-16	2		CRC-16 of all bytes in the response.

The Response is sent using the previous parity setting. The parity is changed about 10 milliseconds after a successful response is sent.

Example 1 (Set Parity to "no parity, 1 stop bit" for MI with address 127):

```
             address
             |  fcode
             |  |
             |  |          parity (0x00 = no parity, 1 stop bit)
             |  |          |
             |  |          |  CRC-16
             |  |          |  |
Command:     7f 6e 04 93 00 00 13

             address
             |  fcode
             |  |
             |  |         status (0 = command successful)
             |  |         |  CRC-16
             |  |         |  |
Response:    7f 6e 04 93 00 00 13
```

Example 1 (Set Parity to "even, 1 stop bit" for MI with address 127):

```
             address
             |  fcode
             |  |
             |  |          parity (0x02 = even parity, 1 stop bit)
             |  |          |
             |  |          |  CRC-16
             |  |          |  |
Command:     7f 6e 04 93 02 81 d2

             address
             |  fcode
             |  |
             |  |         status (0 = command successful)
             |  |         |  CRC-16
             |  |         |  |
Response:    7f 6e 04 93 00 00 13
```

3.3.3 Get Device Information

	Parameter	# bytes	Units	Notes
Command	Address	1		Destination address of MI
	0x6e	1		Modbus Function Code = 0x6e
	0x03	1		Command length
	0x8e	1		MI Command
	CRC-16	2		CRC-16 of all bytes in the command
Response	Address	1		Address of responding MI
	0x6e	1		Modbus Function Code = 0x6e
	0x15	1		Command length
	0x8e	1		MI Command
	Serial number	4		Factory set serial number (unsigned)
	Firmware version	6		ASCII characters padded with spaces (not null terminated)
	Product type	6		ASCII characters padded with spaces (not null terminated)
	Factory info	2		Factory info (ignore)
	CRC-16	2		CRC-16 of all bytes in the response.

Example 1 (Get Device Information for MI with address 127):

```
              address
              |  fcode
              |  |
              |  |
              |  |
              |  |        CRC-16
              |  |        |
Command:      7f 6e 03 8e f8 b9

              address
              |  fcode
              |  |        serial number (0x00000001)
              |  |        |            firmware version (1.06 in ascii)
              |  |        |            |                 product (MI-1 in ascii) CRC-16
              |  |        |            |                 |                       |
Response:     7f 6e 15 8e 00 00 00 01 31 2e 30 36 20 20 54 37 2d 31 20 20 00 0f  8a d6
```

3.3.4 Set Address

This command changes the address of the MI. For extra safety, the serial number of the device to be changed is sent as well. Use the “Get Device Information” command to read the serial number.

	Parameter	# bytes	Units	Notes
Command	Address	1		Destination address of MI
	0x6e	1		Modbus Function Code = 0x6e
	0x09	1		Command length
	0x91	1		MI Command
	0x04	1		Device type
	Serial number	4		Serial number of MI to change address
	New address	1		New address: 1 to 100 or 127
	CRC-16	2		CRC-16 of all bytes in the command
Response	Address	1		Address of responding MI
	0x6e	1		Modbus Function Code = 0x6e
	0x04	1		Command length
	0x91	1		MI Command
	Status	1		0 = Command successful Otherwise = Command failed
	CRC-16	2		CRC-16 of all bytes in the response.

Example 1 (Change address of MI with address 127 and serial number 0x00000001 to address 10):

```
              address
              |  fcode
              |  |
              |  |           serial number (0x00000001)
              |  |           |           new address (0x0a)
              |  |           |           |  CRC-16
              |  |           |           |  |
Command:      7f 6e 09 91 04 00 00 00 01 0a 66 ce

              address
              |  fcode
              |  |
              |  |        status (0 = command successful)
              |  |        |  CRC-16
              |  |        |  |
Response:     7f 6e 04 91 00 01 73
```