

ModWeigh MODBUS

Table of contents

[Wiring](#)

[Using MODBUS on the COM2 RS485 ports](#)

[MODBUS RTU considerations](#)

[Register addressing](#)

[Coil and Register address base](#)

[Real \(float\) number byte order](#)

[Testing MODBUS](#)

[MODBUS RTU Coil interface](#)

[Inputs](#)

[Outputs](#)

[Examples](#)

[Setting Coil 2 \(run\)](#)

[Request](#)

[Response](#)

[Clearing Coil 2 \(run\)](#)

[Request](#)

[Response](#)

[MODBUS RTU Register Interface](#)

[Inputs](#)

[Outputs](#)

[Examples](#)

[Writing Register 8036 \(control1C\) to set bit 2 \(run\)](#)

[Request](#)

[Response](#)

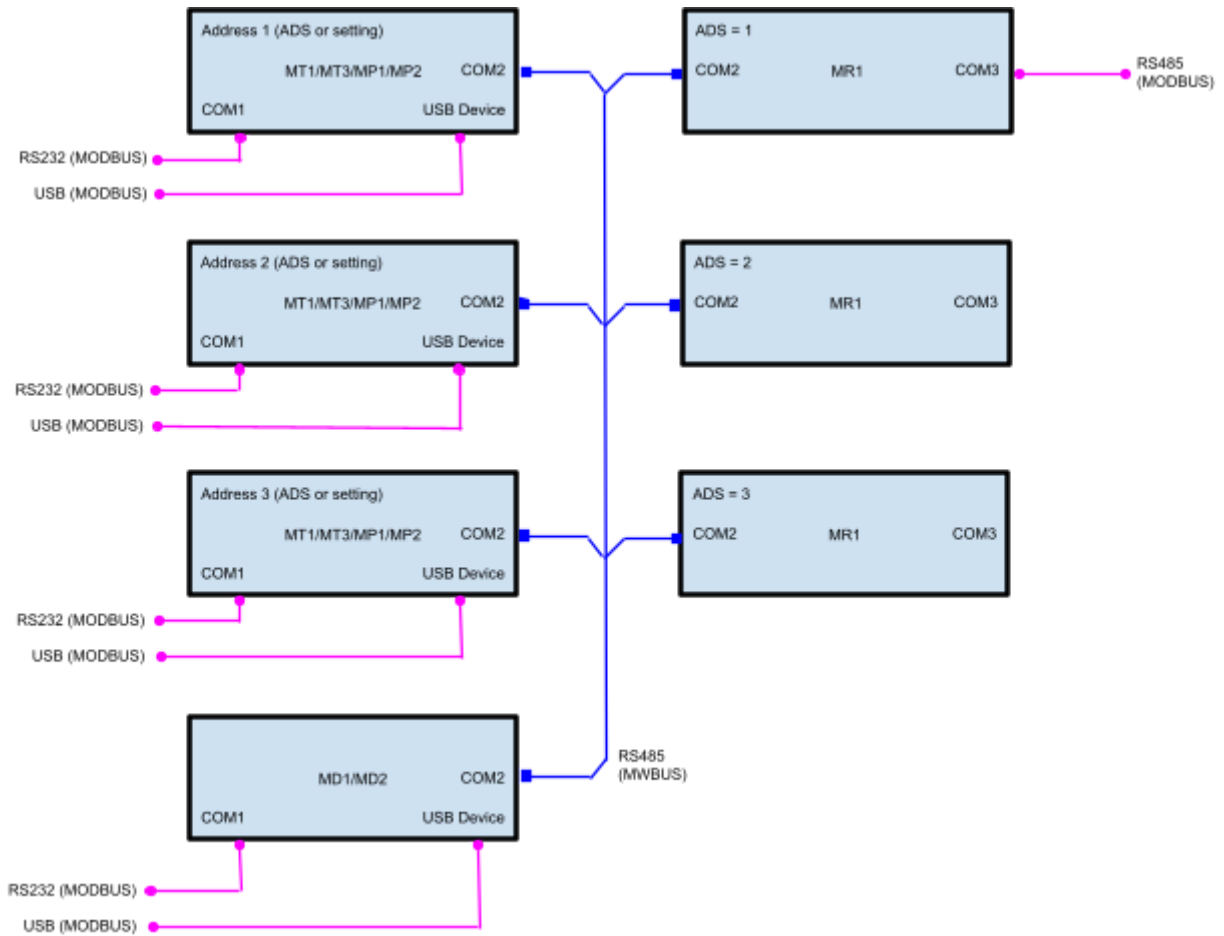
[Reading 12 registers at 8000 \(outputs\)](#)

[Request](#)

[Response](#)

[MODBUS register address ranges](#)

Wiring



This is the recommended wiring for ModWeigh units. All units are bussed together using their COM2 RS485 ports.

The modbus address of each unit can be selected using the ADS (address select) pin. For transmitters (MT1, MT3) and processors (MP1, MP2) the modbus address can be changed using a setting.

Up to 10 transmitters or processors can be connected this way, with an associated MR1. Up to 10 displays can also be connected to the MWBUS as well but they must have a unique address set with ADS (1 to 10) or in the setting.

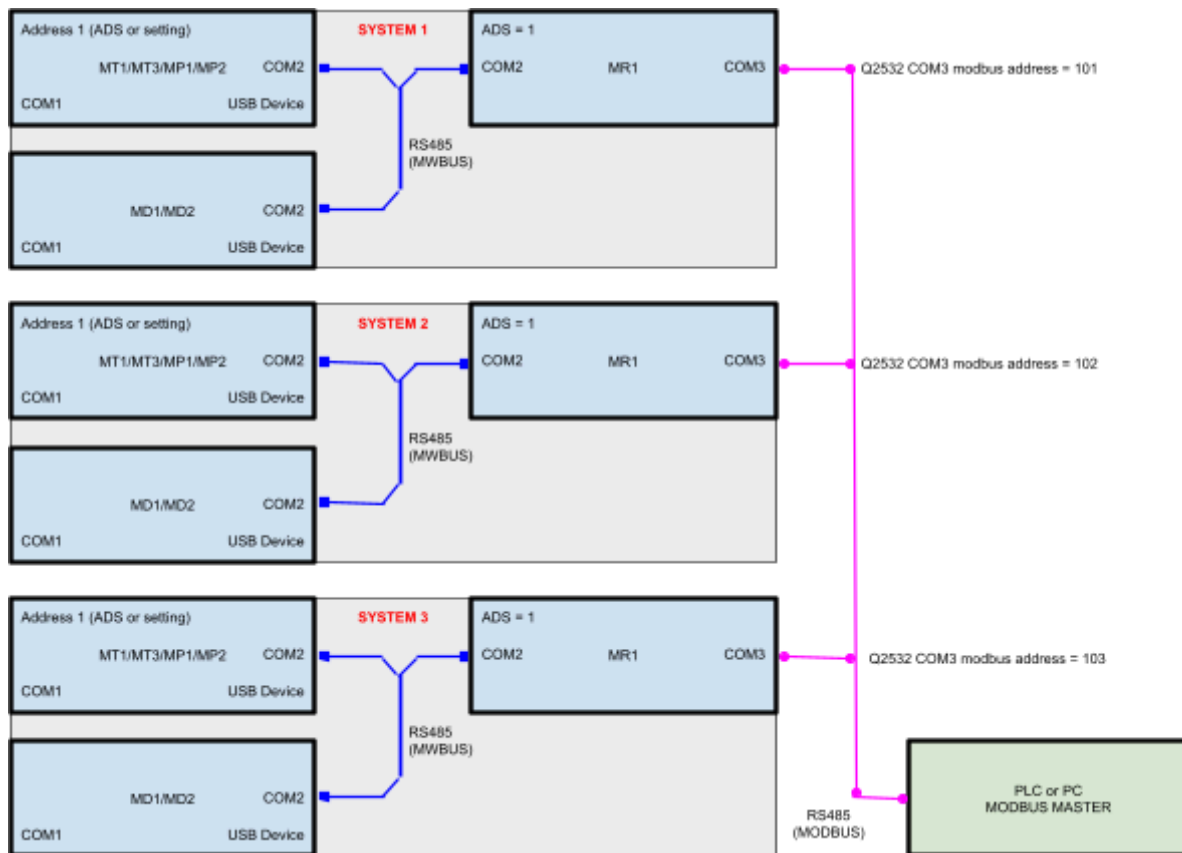
A MODBUS master can be connected to:

- The COM1 RS232 port of any unit.
- The USB device port of any unit.
- The COM3 RS485 port of **one** MR3 unit.

The MODBUS master can then access all of the units on the MWBUS using their unique addresses (1 to 10). Multiple MODBUS masters can be connected to different ports if desired.

If more than 10 transmitters/processors are required, or you wish to access multiple remote modweigh systems that are separated physically, then each transmitter/processor is wired separately with its MR1 and display (if needed). **The default modbus addresses are not changed in this configuration.** The COM3 RS485 port of each MR1 can then be bussed together to a MODBUS

master. Each MT1/MR1 pair must be assigned a unique modbus address on this bus using the setting **Q2532 COM3 modbus address** of the transmitter (available in v6.10 onwards).



In this example the MR1 COM3 modbus address is set to 101, 102 and 103. The address can be set up to a maximum of 239 which allows many more than 10 systems to be accessed remotely. Each system requires a unique address in Q2532.

To address the first system modbus slave address 101 is used. To address the second system modbus address 102 is used. To address the third system modbus address 103 is used.

When a modbus request is sent by the modbus master to modbus slave address 101 it will be answered by the first system. When a modbus request is sent by the modbus master to modbus slave address 102 it will be answered by the second system. When a modbus request is sent by the modbus master to modbus slave address 103 it will be answered by the third system.

Using MODBUS on the COM2 RS485 ports

The COM2 RS485 port can be changed to MODBUS instead of MWBUS but this has the following limitations:

- Only a single MODBUS master must be present on the RS485 bus. This can be either a display (MD1, MD2) or an external MODBUS master.
- MR1 units cannot be used.
- Connecting to the COM1 RS232 port or USB device port of a unit only allows that unit to be used with MODBUS (accessing other units is a feature only supported by MWBUS).

To change COM2 to MODBUS the following settings should be changed.

On displays (MD1/MD2):

- Q92521 COM2 baud rate. Set this to the correct value for the external converters or radio modems being used.
- Q92523 COM2 stop bits. The default of 0 works in most cases but it can be changed to 1 or 2 to match whatever external converters or radio modems are being used.
- Q92524 COM2 mode = 2 (master). The display becomes the modbus master of the RS485 bus.
- Q925694 comms latency. This can be increased when using high latency devices like radio modems.
- Q925695 modbus master options = 1 (length framing). This should be used with radio modems to prevent packet fragmentation. It uses a non-standard modbus header to achieve this so it does not work through modbus aware gateways.
- Q925699 select baudrate = 1 (fixed). This locks the display to the baud rate setting (normally the display searches through the available baud rates when scanning the bus).

On transmitters (MT1/MT3/MP1/MP2):

- Q2521 COM2 baud rate. Set this to the correct value for the external converters or radio modems being used.
- Q2523 COM2 stop bits. The default of 0 works in most cases but it can be changed to 1 or 2 to match whatever external converters or radio modems are being used.
- Q2524 COM2 mode = 1 (slave). The transmitter becomes a modbus slave on the RS485 bus.

Testing

The setting Q925690 connection test can be used to test the connection. Set this to 1 to test the connection. Information is shown about the status of the connection. There should be no errors.

MODBUS RTU considerations

Register addressing

A register in MODBUS is two bytes (a word). Therefore a register address is addressing 2 bytes or 1 word. A real (float) number is 4 bytes or 2 words so it occupies two register addresses, for example 8000 and 8001.

Coil and Register address base

Base 0	Default for v6.10 units up to v6.10r31. NB: Units upgraded from older versions will not change.	Coil and Register addresses are put directly onto the wire with no change. The new setting Q25043 modbus address base is used to control this.
Base 1	Default up to v6.09 units and v6.10r32 onwards.	Coil and Register addresses are put onto the wire with an offset of 1. So address 1 becomes 0 on the wire. <i>This is the modicon default.</i>

Real (float) number byte order

High Endian (4321)	Default for v6.10 units up to v6.10r31. NB: Units upgraded from older versions will not change.	The order of bytes in a 32 bit real (float) number is 4321. The new setting Q25042 modbus word order is used to control this.
High endian, Word swap (2143)	Default up to v6.09 units and v6.10r32 onwards.	The order of bytes in a 32 bit real (float) number is 2143. The words (2 bytes each) are swapped. <i>This is the modicon default.</i>

Testing MODBUS

From ModWeigh v6.10r32 onwards the new setting **Q25040 modbus status** is available.

Modbus
Q25040 modbus status

0.000

Use modbus function 16 (0x10) to write a non zero 32 bit real (-1000 to 1000) into register 8840
Enter 0 to clear errors

8840

MODBUS RTU Coil interface

The MODBUS RTU coil interface is defined as the following function codes.

Name	Hex code	Notes
Read Coils	01	v6.09 onwards
Read Inputs	02	v6.09 onwards
Force Coil	05	v6.10 onwards
Force Multiple Coils	0F	v6.10 onwards

These function codes can be used to read and write the **I/O Function Table** (described in the ModWeigh Instruction Manual for each product).

Inputs

Coils 1 to 47 are inputs that can be written to with function codes 0x05 and 0x0F (both function codes are available in v6.10 onwards). The coils depend on the product type (batch weigher, belt weigher etc) but the following coils are usually present in all products.

Coil	Name
1	stop
2	run
3	pause

The Input Coils can also be accessed using the control1C (8036), control2C (8034) and control3C (8035) registers (register addresses in brackets). These registers group 16 coils together into a single 16 bit register. For v6.09 and earlier this is the only way to control these inputs.

Outputs

Coils 48 onwards are outputs that can be read using function codes 0x01 and 0x02. There is no difference between these two function codes in ModWeigh. The coils depend on the product type (batch weigher, belt weigher etc) but the following coils are usually present in all products.

Coil	Name
57	running
58	paused

The Output Coils can also be accessed using the status1, status2 and status3 registers at register addresses 8015, 8016 and 8017. These registers group 16 coils together into a single 16 bit register.

Examples

NB: These examples are using base 1 addressing where the coil address on the wire is one less than the actual coil address. If the unit is set to base 0 addressing (available in v6.10 onwards) then the address on the wire will be the same as the actual coil address.

Setting Coil 2 (run)

Request

01 05 00 01 FF 00 DD FA

01	1 Byte	Slave Address
05	1 Byte	Function code
00 01	2 Bytes	Output Address (base 1: Coil 2 - 1 = 00 01)
FF 00	2 Bytes	Output value (set = FF 00)
DD FA	2 Bytes	CRC

Response

01 05 00 01 FF 00 DD FA

01	1 Byte	Slave Address
05	1 Byte	Function code
00 01	2 Bytes	Output Address
FF 00	2 Bytes	Output value
DD FA	2 Bytes	CRC

Clearing Coil 2 (run)

Request

01 05 00 01 00 00 9C 0A

01	1 Byte	Slave Address
05	1 Byte	Function code
00 01	2 Bytes	Output Address (base 1: Coil 2 - 1 = 00 01)
00 00	2 Bytes	Output value (clear = 00 00)
9C 0A	2 Bytes	CRC

Response

01 05 00 01 00 00 9C 0A

01	1 Byte	Slave Address
05	1 Byte	Function code
00 01	2 Bytes	Output Address
00 00	2 Bytes	Output value
9C 0A	2 Bytes	CRC

MODBUS RTU Register Interface

The MODBUS RTU register interface is defined as the following function codes.

Name	Hex code	Notes
Read Holding Registers	03	
Read Input Registers	04	
Preset Single Register	06	
Preset Multiple Registers	10	
ReadWrite Multiple Registers	17	

There is no difference between function codes 03 and 04 in ModWeigh.

All settings have a register address which is displayed at the bottom right of the screen when editing the setting.

The **Interface Registers** provide access to the basic inputs and outputs of the controller. The register contents depend on the product type (batch weigher, belt weigher etc). The multi-register commands can be used to efficiently read and write these registers.

Inputs

Register	Bytes	Type	Name
8027	2	bits	control4C
8028	4	real (float)	speedC
8030	4	real (float)	ratioC
8032	4	real (float)	setpointC
8034	2	bits	control2C
8035	2	bits	control3C
8036	2	bits	control1C

Outputs

Register	Bytes	Type	Name
8000	4	real (float)	platform weight
8002	4	real (float)	belt speed

8004	4	real (float)	speed demand
8006	4	real (float)	current setpoint
8008	4	real (float)	total weight
8010	4	real (float)	flow rate
8012	2	bits	control1
8013	2	bits	control2
8014	2	bits	control3
8015	2	bits	status1
8016	2	bits	status2
8017	2	bits	status3

Examples

NB: These examples are using base 1 addressing where the register address on the wire is one less than the actual register address. If the unit is set to base 0 addressing (available in v6.10 onwards) then the address on the wire will be the same as the actual register address.

NB: These examples are using High endian, Word swap (2143) for real (float) numbers. If the unit is set to High Endian (4321) word ordering (available in v6.10 onwards) then the words are not swapped.

Writing Register 8036 (control1C) to set bit 2 (run)

Request

01 06 1F 63 00 04 7F C3

01	1 Byte	Slave Address
06	1 Byte	Function code
1F 63	2 Bytes	Register Address (base 1: Register 8036 - 1 = 1F 63)
00 04	2 Bytes	Register Value
7F C3	2 Bytes	CRC

Response

01 06 1F 63 00 04 7F C3

01	1 Byte	Slave Address
----	--------	---------------

06	1 Byte	Function code
1F 63	2 Bytes	Register Address
00 04	2 Bytes	Register Value
7F C3	2 Bytes	CRC

Reading 12 registers at 8000 (outputs)

Request

01 03 1F 3F 00 0C 72 17

01	1 Byte	Slave Address
03	1 Byte	Function code
1F 3F	2 Bytes	Starting Address (base 1: Register 8000 - 1 = 1F 3F)
00 0C	2 Bytes	Quantity of Registers (N = 12)
72 17	2 Bytes	CRC

Response

01 03 18 F5 C3 42 49 6D 0F 3E AB 49 3A 42 05 00 00 42 C8 4B D7 45 9A DB CC 42 CA 82 24

01	1 Byte	Slave Address
03	1 Byte	Function code
18	1 Byte	Byte count (2 x N = 24)
F5 C3 42 49 6D 0F 3E AB 49 3A 42 05 00 00 42 C8 4B D7 45 9A DB CC 42 CA	2 x N Bytes	platform weight (2143 = 50.49) belt speed (2143 = 0.334816) speed demand (2143 = 33.3215) current setpoint (2143 = 100) total weight (2143 = 4937.48) flow rate (2143 = 101.429)
82 24	2 Bytes	CRC

MODBUS register address ranges

All settings and values in ModWeigh can be accessed using register addresses in the range of 8000 to 8999. Most of the data is 32 bit IEEE floats (real numbers) with exceptions for the control and status words which are 16 bit binary values.

These values can also be accessed as 16 bit and 32 bit integers by using the address ranges 6000 to 6499 (16 bit integer) and 7000 to 7999 (32 bit integer). The following table shows this:

Register addresses	Type	Mapping from 8000 range
6000 to 6499	16 bit integer	$6000 + (\text{address} - 8000) / 2$
7000 to 7999	32 bit integer	$\text{address} - 1000$
8000 to 8999	default	

When a value is converted from the 32 bit IEEE float (real) to an integer it is controlled by display settings like divisions.