DEDITEC GMBH

Modbus configuration and register assignments

2025 October

INDEX

INDEX

Introduction

1. Introduction

1.1. What is Modbus?

Modbus is one of the most widely used communication protocols in automation technology. It is easy to understand and, thanks to its open protocol, can be used across different manufacturers.

Data exchange is based on a master-slave structure, whereby the master unit initiates requests and the slave units respond to its commands.

The protocol comprises four basic data types:

1. Coils	-> write and readback - 1 bit
2. Discrete Inputs	-> read - 1 bit
3. Input Registers	-> read - 16 bit
4. Holding Registers	-> write and readback - 16 bit

The two most important transmission types are Modbus RTU (serial) and Modbus TCP (Ethernet-based).

1.2. What is Modbus used for?

Modbus is typically used in automation technology, process and plant control systems, building automation, and energy and environmental technology. The most common applications are communication between a PLC and connected sensors, actuators, or frequency converters, as well as reading energy meters or other measuring devices. This can be done using a mix of Modbus devices from different manufacturers.

Advantages of Modbus

- Open protocol
- Can be used independently of manufacturer
- Modbus TCP can be applied to an existing network infrastructure.

1.3. TCP/RTU packet structure

What does a Modbus packet do?

A Modbus packet is a message from the sender (client/master) to the receiver (server/slave) or vice versa.

Example

The sender makes a request: e.g., “Read me 2 holding registers from address 40001.”

The receiver processes this request and sends back the appropriate response: e.g., “Here are the two values: 1200 and 450.”

What does a Modbus packet generally contain?

- Address information → Which device is meant (slave ID or unit identifier).
- Function code → Which action should be performed (e.g., read, write).
- Data → Which address(es) and how many values are affected, or the values themselves.
- Error checking → with TCP, this is handled by the TCP/IP protocol.

Modbus TCP packet structure

A TCP telegram is sent via Ethernet (port 502) and consists of:

- MBAP header (7 bytes):
  a. Transaction Identifier
  b. Protocol Identifier (always 0)
  c. Length (number of bytes following)
  d. Unit Identifier (similar to the slave address, important for gateways)
- Function code – specifies what is to be done (e.g., read register).
- Data – addresses, values, or parameters.

Register allocation

2. Register allocation

2.1. General information

The Mobus register assignments determine which data is stored at which address in a device.

The DEDITEC Mobus registers always start at 0 (base = 0).

General register allocation

Modbus Register (dez)	Data type	Access	Data size (Bit)	Description
0 ... 9999	Coil	write / read	1	Digital outputs Writing and reading back
10000 ... 19999	Discrete Inputs	read	1	Read digital inputs
30000 ... 39999	Input Register	read	16	Read holder register
40000 ... 49999	Holding Register	write / read	16	Register Writing and Reading Back

With the correct register assignment, you know exactly where which information is stored in the device and can read, monitor, or change it in a targeted manner.

2.2. Status register

2.2.1. Reading status registers

Information about the module can be retrieved using the following addresses.

The parameters read back are defined in the delib_defines.h file.

After installing our software package, you will find the file at the following location:

32-bit: C:\Program Files (x86)\DEDITEC\DELIB\include

64-bit: C:\Program Files\DEDITEC\DELIB64\include

Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
39200	0x23F0	read	16	Firmware version
39168	0x23D0	read	32	Software feature
39170	0x23D2	read	32	Hardware interface
39172	0x23D4	read	32	Module-Config feature
39174	0x23D6	read	32	EC events feature
39176	0x23D8	read	32	TCP feature
39178	0x23DA	read	32	HW group
39180	0x23DC	read	32	SW class

If the value 0101 is read for the firmware version, this corresponds to version number 1.01.

2.3. I/O configuration / Number of channels

The following table shows the register address you need to read out the number of channels.

Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
38960	0x2300	read	16	Digital output channels
38961	0x2301	read	16	Digital input channels
38962	0x2302	read	16	Digital input/output channels (TTL)
38963	0x2303	read	16	Analog output channels
38964	0x2304	read	16	Analog input channels
38965	0x2305	read	16	Stepper channels
38966	0x2306	read	16	Counter for input state changes
38967	0x2307	read	16	Input counter
38968	0x2308	read	16	Temperature channels
38969	0x2309	read	16	48-bit input counter
38970	0x230A	read	16	Pulse generator output channels
38971	0x230B	read	16	PWM output channels

2.4. Digital outputs

The registers for controlling the output modules can not only be written to, but also read back. This means that the current output status of one (or more) modules can be queried at a later point in time. If the software is accidentally terminated or even crashes, the status of the output modules can be recorded. This means that a running process does not have to be reset.

The digital outputs of the module can be controlled using the following register addresses.

2.4.1. Setting and reading back digital outputs (1 channel)

Coils (0xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
00000	0x0000	r/w	1	Setting and reading back channel 1
00001	0x0001	r/w	1	Setting and reading back channel 2
...	...			
00255 *	0x00FF *	r/w	1	Setting and reading back channel 256

* the maximum address range depends on the number of channels in the module

These registers are used to set and read back individual digital outputs.

If the value 1 is transferred to the register address, the respective channel is switched on. The channel is switched off with the value 0.

When reading back the register address, the return value is 0 or 1.

Example:

The value 1 is written to register 0x0001. Channel 2 is switched on.

The value 0 is written to register 0x000F. Channel 16 is switched off.

2.4.2. Setting and reading back digital outputs (16 channels)

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
41024	0x0400	r/w	16	Setting and reading back channel 1 - 16
41025	0x0401	r/w	16	Setting and reading back channel 17 - 32
...	...			
41039 *	0x040F *	r/w	16	Setting and reading back channel 241 - 256

* the maximum address range depends on the number of channels in the module

This register sets 16 digital outputs simultaneously.

When reading the register, the current bit value is read back as a hexadecimal number.

Example:

The value 0xff00 is written to address 0x0400.

Outputs 1-8 are set to 0.

Outputs 9-16 are set to 1.

2.4.3. Bitwise setting of digital outputs (16 channels)

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
41056	0x0420	r/w	16	Setting channel 1 - 16
41057	0x0401	r/w	16	Setting channel 17 - 32
...	...			
41071 *	0x042F *	r/w	16	Setting channel 241 - 256

* the maximum address range depends on the number of channels in the module

This register can be used to switch outputs to 1 without changing the states of the adjacent outputs.

Only the bits with a value of 1 in the transfer value are taken into account by the command.

When reading the register, the current bit value is read back as a hexadecimal number.

Example:

The value 0x01 is written to address 0x0420. Output 0 is set to 1, the status of outputs 1-31 remains unchanged.

The value 0xff is written to address 0x0420. Output 0-7 is set to 1, the status of output 8-31 remains unaffected.

The value 0xff000000 is written to address 0x0420. Output 23-31 is set to 1, the status of output 0-22 remains unaffected.

2.4.4. Bitwise reset of digital outputs (16 channels)

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
41088	0x0440	r/w	16	Resetting channel 1 - 16
41089	0x0441	r/w	16	Resetting channel 17 - 32
...	...			
41103 *	0x044F *	r/w	16	Resetting channel 241 - 256

* the maximum address range depends on the number of channels in the module

This register can be used to switch outputs to 0 without changing the states of the adjacent outputs.

When reading the register, the current bit value is read back as a hexadecimal number.

Example:

The value 0x1 is written to address 0x0440. Output 0 is set to 0, the status of outputs 1-31 remains unaffected.

The value 0xff is written to address 0x0440. Output 0-7 is set to 0, the status of output 8-31 remains unaffected.

The value 0xff000000 is written to address 0x0440. Output 23-31 is set to 0, the status of output 0-22 remains unaffected.

2.5. Digital inputs

The logical states of the voltage levels at the input modules can be read out via the following registers.

2.5.1. Reading digital inputs (1 channel)

Discret Inputs (1xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
10256	0x0100	read	1	Reading channel 1
10257	0x0101	read	1	Reading channel 2
...	...			
10511 *	0x01FF *	read	1	Reading channel 256

* the maximum address range depends on the number of channels in the module

These registers are used to read a single digital input.
The return value is the status of the input (0/1).

Example:

Register 0x0101 reflects the status of channel 2.
Register 0x010F reflects the status of channel 16.

2.5.2. Reading digital inputs (16 channels)

Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
30000	0x0000	read	16	Reading channel 1 - 16
30001	0x0001	read	16	Reading channel 17 - 32
...	...			
30015 *	0x000F *	read	16	Reading channel 241 - 256

* the maximum address range depends on the number of channels in the module

These registers are used to read 16 digital inputs.

When reading the register, the current bit value is returned as a hexadecimal number.

Example:

Register 0x0001 reflects the states of channels 17 to 32.

Register 0x0004 reflects the states of channels 65 to 80.

2.5.3. Reading input status changes (16 channels)

Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
30016	0x0010	read	16	Reading the input status change of channel 1 - 16
30017	0x0011	read	16	Reading the input status change of channel 17 - 32
...	...			
30031 *	0x001F *	read	16	Reading the input status change of channel 241 - 256

* the maximum address range depends on the number of channels in the module

This register reads the input state changes of 16 channels.

Attention!

When reading the input status change, these are reset directly (to 0).

Example:

Register 0x0010 reflects the input status change flags for channels 17 to 32 and resets them.

Register 0x0014 reflects the input status change flags for channels 65 to 80 and resets them.

2.5.4. Read input counter (1 channel)

Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
30512	0x0200	read	16	Reading the input counter channel 1
30513	0x0201	read	16	Reading the input counter channel 2
...	...			
30527 *	0x02FF *	read	16	Reading the input counter channel 256

* the maximum address range depends on the number of channels in the module

These registers are used to read the input counters.
When the register is read, the current counter value is returned.

Example:
Register 0x0201 reflects the input counter for channel 2.
Register 0x0204 reflects the input counter for channel 5.

2.5.5. Read input counter and reset to 0 (1 channel)

Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
30768	0x0300	read	16	Reading and resetting channel 1
30769	0x0301	read	16	Reading and resetting channel 2
...	...			
30783 *	0x03FF *	read	16	Reading and resetting channel 256

* the maximum address range depends on the number of channels in the module

These registers are used to read and reset the input counters.
When the register is read, the current counter value is returned and then set to 0.

Example:
Register 0x0301 reads the input counter of channel 2 and resets it to 0.
Register 0x0304 reads the input counter of channel 5 and resets it to 0.

2.5.6. Interval for input filter writing and reading

Interval of the input state change filter

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
48704	0x2200	r/w	8	Setting and reading the time interval of the FF filter

Register for setting and reading back the input state change filter [ms], which specifies the time interval for sampling the input state change flip-flops.

Pulse times between 5 ms and 255 ms are supported—if not specified, the default value of 100 ms applies.

Important!

This register is only supported if the module has digital inputs.

Input filter interval

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
48705	0x2201	r/w	8	Setting and reading the time interval of the input filter

Register for setting and reading an input filter in [ms], which specifies the time interval in which interference pulses are filtered on digital input channels.

This command supports pulse times between 1 ms and 255 ms. If no time is set, the default value is 0 ms (=off).

Important!

This register is only supported if the module has digital inputs.

2.6. TTL inputs/outputs

2.6.1. Setting and reading back the TTL direction (8 channels)

Holding Register (4xxxx) - Base 0

Modbus register (dez)	register address (hex)	Access	Data size (Bit)	Description
48752	0x2230	r/w	8	Sets the direction of the TTL inputs/outputs

This register is used to set the direction of up to 64 consecutive TTL inputs/outputs (8-bit wise).

1 bit represents 8 TTL inputs/outputs. Value 1 = output / 0 = input.

When reading the register, the current 8-bit value is read back as a hexadecimal number.

Bit 0: Direction from TTL 0-7 / 1=Output, 0=Input
Bit 1: Direction from TTL 8-15 / 1=Output, 0=Input
Bit 2: Direction from TTL 16-23 / 1=Output, 0=Input
Bit 3: Direction from TTL 24-31 / 1=Output, 0=Input
Bit 4: Direction from TTL 32-39 / 1=Output, 0=Input
Bit 5: Direction from TTL 40-47 / 1=Output, 0=Input
Bit 6: Direction from TTL 48-55 / 1=Output, 0=Input
Bit 7: Direction from TTL 56-63 / 1=Output, 0=Input

Example:

The value 0x1 is written to register 0x2230. Sets channels 0-7 to output, the others to input.

The value 0xC is written to register 0x2230. Sets channels 16-31 to output, the others to input.

The value 0xFF is written to register 0x2230. Sets channels 0-63 to output.

2.7. Analog Outputs

The following registers allow you to set and read back D/A values and D/A modes.

2.7.1. Writing and reading back D/A values

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
42816	0x0B00	r/w	16	Setting and Reading Back Channel 1
42817	0x0B01	r/w	16	Setting and Reading Back Channel 2
...	...			
43071 *	0x0BFF *	r/w	16	Setting and Reading Back Channel 256

* the maximum address range depends on the number of channels in the module

Diese Register dienen dazu, einem Kanal eines 16-Bit-D/A-Wandlers einen Datenwert zuzuweisen.

Der Datenwert, der geschrieben werden kann, ist ein 16-Bit-Wert im Bereich von 0 bis 65535.

When reading back the register, the current D/A value of the channel is read out.

Example:

The value 0xFFFF is written to register 0x0B01.

This sets the first output of the D/A converter to the maximum value of the selected mode.

If the mode ADDA_MODE_UNIPOL_10V is selected, the first output of the D/A converter is set to 10V.

2.7.2. D/A mode setting and readback

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
42560	0x0A00	r/w	8	Setting and Reading Back the D/A Mode of a Channel 1
42561	0x0A01	r/w	8	Setting and Reading Back the D/A Mode of a Channel 2
42815*	0x0aFF*	r/w	8	Setting and Reading Back the D/A Mode of a Channel 256

* the maximum address range depends on the number of channels in the module

This register is used to set the mode for a D/A converter.

The available voltage and current modes depend on the module and can be found in the module's manual under "Technical Specifications."

When reading back the register, the currently set D/A mode of the channel is retrieved.

Example:

If the value 0x41 is assigned to register 0x0A01, the D/A mode of channel 2 is set to "±5V".

2.8. Analog inputs

The following registers allow you to read the A/D measurement values of the module and to configure A/D modes and filter levels.

2.8.1. Reading analog inputs

Input Register (3xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
32304	0x0900	read	16	Reading channel 1
32305	0x0901	read	16	Reading channel 2
...	...			
32559 *	0x09FF *	read	16	Reading channel 256

* the maximum address range depends on the number of channels in the module

This register is used to read a data value from a channel of the A/D converter.

Example:

Register 0x0901 shows the measured value from channel 2.

Register 0x0904 shows the measured value from channel 5.

2.8.2. A/D mode Write and read back

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
42048	0x0800	r/w	8	Sets the A/D mode for channel 1
42049	0x0801	r/w	8	Sets the A/D mode for channel 2
...	...			
42303*	0x08FF*	r/w	8	Sets the A/D mode for channel 256

* the maximum address range depends on the number of channels in the module

This register is used to set the mode for a channel of the A/D converter.

The possible voltage and current modes depend on the module and can be found in the module manual under “Technical Data.”

When reading back the register, the currently set A/D mode of the channel is retrieved.

Example:

If the value 0x41 is assigned to register 0x0801, the A/D mode of channel 2 is set to "±5V".

If the value 0x80 is assigned to register 0x0804, the A/D mode of channel 5 is set to "0–20 mA".

2.8.3. Writing and reading back A/D filter levels

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
38736	0x2220	r/w	8	Setting and reading the A/D filter level

This register allows you to set the A/D filter level of your A/D module for all channels.

For modules with multiple A/D submodules, the filter level can be set separately for each submodule.

The filtering principle is based on averaging a sum of measurements. The higher the selected filter level, the more measurements are used for averaging.

However, this also means that the sampling rate of the A/D converter decreases as the filter level increases.

Depending on the module, the range of adjustable filter levels may vary.

A value of 0 disables the A/D filter.

When reading back the register, the currently set A/D filter level is read out.

Example:

If the value 5 is assigned to register 2220, the A/D filter level is set to 5.

If the value 0 is assigned to register 2220, the A/D filter is deactivated.

2.9. Timeout for Digital and Analog Outputs

The following registers allow you to configure the timeout function for digital and analog outputs.

2.9.1. Set Timeout Mode

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
48720	0x2210	write	8	Sets the Timeout Mode

This register is used to set the timeout mode.

The following modes can be set:

Value = 0 disables the timeout.

Value = 1 enables the “normal” timeout.

Value = 3 enables the “auto reactivate” timeout.

Value = 5 enables the “secure outputs” timeout.

"normal" Timeout

This is the timeout function that has been implemented in our modules since 2009.

Procedure for the timeout command: The timeout is activated via command. If a so-called timeout event occurs (i.e., the pause between two accesses to the module exceeds the permitted timeout duration), the following happens:

– All outputs are switched off.

– The timeout status changes to "2".

– The timeout LED lights up (on modules that support this status).

Further access to the outputs remains possible, but the timeout is no longer active—until it is reactivated.

"auto reactivate" Timeout

This is a timeout mode implemented since 2021, which automatically reactivates the timeout after a timeout event occurs.

Procedure for the timeout command: The timeout is activated via command. If a so-called timeout event occurs (i.e., the pause between two accesses to the module exceeds the permitted timeout duration), the following happens:

– All outputs are switched off.

– The timeout status changes to "4".

– The timeout LED lights up (on modules that support this status).

Further access to the outputs remains possible, and the timeout remains active. If the timeout duration is exceeded again, the outputs are switched off once more

"secure outputs" Timeout

This is a timeout mode implemented since 2021 that prevents write access to the outputs after a timeout event occurs.

This ensures that the software must first restore a “safe” state of the outputs, since the module’s timeout mechanism has changed the outputs to predefined values.

Procedure for the timeout command: The timeout is activated via command. If a so-called timeout event occurs (i.e., the pause between two accesses to the module exceeds the permitted timeout duration), the following happens:

– All outputs are switched off.

– The timeout status changes to "6".

– The timeout LED lights up (on modules that support this status).

Further access to the outputs is NOT possible. Outputs can only be written again after the timeout has been reactivated or deactivated.

2.9.2. Read Timeout Status

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
48721	0x2211	read	8	Reading the Timeout Status

This register is used to read the timeout status.

A timeout event has occurred if the timeout was activated and no data traffic took place for at least the duration of the configured timeout value

Return = 0 (Timeout is deactivated)

Values for reading the "normal" timeout status:

Return = 1 (Timeout "normal" is activated)

Return = 2 (Timeout "normal" has occurred)

Values for the "auto reactivate" timeout:

Return = 3 (Timeout "auto reactivate" is activated)

Return = 4 (Timeout "auto reactivate" has occurred one or more times)

Values for the "secure" timeout:

Return = 5 (Timeout "secure outputs" is activated)

Return = 6 (Timeout "secure outputs" has occurred. In this status, writing to the outputs is prevented)

2.9.3. Set and readback Timeout Value

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
48722	0x2212	write	16	Sets the timeout value.

The timeout value is calculated as the product of a 16-bit value × 100 milliseconds (i.e., the unit is 100 ms).

The permissible range for the timeout duration lies between 0.1 seconds and 6553 seconds.

When reading back the register, the currently set timeout value is read out.

2.9.4. Bitwise setting of digital outputs in case of timeout (16 channels)

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
41248	0x04E0	r/w	16	Setting and reading outputs 1–16 in the event of a timeout.
41249	0x04E1	r/w	16	Setting and reading outputs 17–32 in the event of a timeout.
...	...			
41039 *	0x04EF *	r/w	16	Setting and reading outputs 241–256 in the event of a timeout.

* the maximum address range depends on the number of channels in the module

This register defines which outputs should be set in the event of a timeout.

When reading the register, the current bit value is read back as a hexadecimal number.

Example:

At address 0x04E0, the value 0x02 is written. Channel 2 is set, all other channels remain unchanged.

At address 0x04E0, the value 0x03 is written. Channels 1 and 2 are set, all other channels remain unchanged.

At address 0x04E0, the value 0x04 is written. Channel 3 is set, all other channels remain unchanged.

2.9.5. Bitwise reset of digital outputs in case of timeout (16 channels)

Holding Register (4xxxx) - Base 0

Modbus Register (dez)	Register address (hex)	Access	Data size (Bit)	Description
41216	0x04C0	r/w	16	Resetting outputs 1–16 in the event of a timeout.
41217	0x04C1	r/w	16	Resetting outputs 17–32 in the event of a timeout.
...	...			
41231 *	0x04CF *	r/w	16	Resetting outputs 241–256 in the event of a timeout.

* the maximum address range depends on the number of channels in the module

This register defines which outputs should be switched off in the event of a timeout.

When reading the register, the current bit value is read back as a hexadecimal number.

Example:

At address 0x04C0, the value 0x02 is written. In the event of a timeout, output 2 is reset; all other channels remain unchanged.

At address 0x04C0, the value 0x03 is written. In the event of a timeout, outputs 1 and 2 are reset; all other channels remain unchanged.

At address 0x04C0, the value 0x04 is written. In the event of a timeout, output 3 is reset; all other channels remain unchanged.

Appendix

3. Appendix

3.1. Revisions

1.0	First DEDITEC manual

3.2. Copyrights and trademarks

Modbus is a registered trademark of Schneider Electric.

Linux is a registered trademark of Linus Torvalds.

USB is a registered trademark of USB Implementers Forum Inc.

LabVIEW is a registered trademark of National Instruments.

Intel is a registered trademark of Intel Corporation.

AMD is a registered trademark of Advanced Micro Devices, Inc.

ProfiLab is a registered trademark of ABACOM Ingenieurbüro GbR.

ispVM System is a registered trademark of Lattice Semiconductor Corporation.

Windows, Visual-C/C++, -C#, -Basic, -Basic.NET and Visual-Studio are registered trademarks of Microsoft Corporation.

Delphi is a registered trademark of Borland Software Corporation.

Java is a registered trademark of Oracle Corporation.