The Robur GEN10 Modbus Interface, f/w 1.003

Abstract

This document provides information specific to Robur GEN10 Modbus interface.
GEN10 f/w 1.003 Modbus interface is backward compatible with previous firmware versions.

Modbus Communication

The Robur GEN10 implements the Modbus RTU communication standard, as a Modbus slave agent.
It can be accessed through a galvanically isolated RS485 port (intended for permanent bus connection) or, alternatively, through an RS232 port (intended for temporary connection, e.g. for service purposes). If a Modbus master agent is connected via the RS232 port while communication is active over the RS485 port, such communication is interrupted and communication over the RS232 port is established, provided that the DTR (Data Terminal Ready) GEN10 input is active. Communication is resumed over the RS485 port when disconnecting the RS232 port, or deactivating the DTR signal.

Communication parameters can be independently set for the RS485 and the RS232 ports by means of parameters P82 and P83 respectively.
The supported settings are:

- 9600 8N1
- 9600 8N2
- 9600 8E1
- 19200 8N1 (default)
- 19200 8N2
- 19200 8E1
- 38400 8N1
- 38400 8N2
- 38400 8E1

The Modbus address can be set with parameter P40. Default value is 1.

Communication timeout:
Modbus timeout can be set with parameter P41. Default is 60 seconds.
If timeout elapses without any Modbus activity, the Modbus controller command request (MODBUS CONTROLLER COMMANDS area) resets to OFF.

Modbus Map

The GEN10 Modbus map uses the Input Registers and the Holding Registers area.

The following Modbus function codes are supported:

- (03) Read Holding Register
- (04) Read Input Register
- (06) Write Single Register
- (16) Write Multiple Register

The GEN10 supports Modbus broadcast messages.

Modbus addresses are listed using two different notations: the first column shows addresses using the “Type Prefix” Modbus notation, while the second column shows addresses using the “Real” Modbus address notation.
In the “Type Prefix” Modbus address notation:

- the first digit is a type prefix, so it's not actually part of the address;
- the addresses themselves start from 1;

E.g. the first holding register address will be shown as "40001", with "4" being the prefix for ‘holding register type’.
In the “Real” Modbus address notation, the address shown is exactly the same being sent in the actual Modbus communication, so:

- there's no prefix;
- the addresses start from 0;

Please note that “Real” Modbus addresses are given in hexadecimal form.
The Modbus function code will specify the register type.

Italic addresses mark out new registers.

INPUT REGISTERS (R/O)

"Type prefix" address (dec)	"Real" address (hex)	Type	category	Description
30001	0x0000	Bit-map	**PLATFORM TYPE**	Platform ID (*) = [Platform type (bits 15-10)] [network id (bits 9-0)] T10 DDC Platform type = 0 T10 CCI Platform type = 1 GHP10/11 Platform type = 2 **GEN10 Platform type = 3** (this device) **NOTE: Network id not used on GEN10 (reads as 0)**
30002 - 30100	0x0001 - 0x0063			FREE (Not mapped)

(*) Every Modbus enabled platform can be identified using this Input Register.

HOLDING REGISTERS (R/W)

"Type prefix" address (dec)	"Real" address (hex)	Type	category	Description	Read only	Write
40001	0x0000	Enum	**OEM IDENTIFICATION**	OEM Code - Robur (0)	R.O.	
40002	0x0001	Unsigned Word		OEM Compatibility Code Major - for this GEN10 version is **0**	R.O.	
40003	0x0002	Unsigned Word		OEM Compatibility Code minor - for this GEN10 version is **1**	R.O.	
40004 - 40100	0x0003 – 0x0063			FREE (Not mapped)		
40101	0x0064	Unsigned Word	**BOARD PARAMETERS**	P000 - Board Serial Number (MSB)	R.O.	
40102	0x0065	Unsigned Word		P000 - Board Serial Number (LSB)	R.O.	
40103	0x0066	Unsigned Word		P002 - Firmware Version Major	R.O.	
40104	0x0067	Unsigned Word		P003 - Firmware Version Minor	R.O.	
40105	0x0068	Unsigned Word		P004 - Hardware Version	R.O.	
40106	0x0069	Unsigned Word		P005 - Bootloader Version	R.O.	
40107	0x006A	Unsigned Word		P006 - Firmware Version Sub-minor (Internal)	R.O.	
40108	0x006B	Enum		P007 - Compile Option - STANDARD (0) - LAB (3)	R.O.	
40109	0x006C	Enum		P008 – GEN10 Serial Number / HW Key check result - OK (0) - Missing serial number (1) - Missing HW Key (2) - Missing serial number and HW key (3)	R.O.	
40110	0x006D	Enum		P009 - Calibration Check Result - Calibration data not used (3)	R.O.	
40111	0x006E	Enum	**APPLIANCE TYPE CONFIGURATION PARAMETERS**	P010 - Appliance Type		C.W.O.
40112 - 40140	0x006F – 0x008B			Reserved		
40141	0x008C	Enum		Reserved		C.W.O.
40142	0x008D	Unsigned Word		Reserved		
40143	0x008E	Unsigned Word		Reserved		
40144	0x008F	Enum		Reserved		C.W.O.
40145	0x0090	Unsigned Word		Reserved		
40146	0x0091	Unsigned Word		Reserved		
40147	0x0092	Unsigned Word		Reserved		C.W.O.
40148	0x0093	Unsigned Word		Reserved		C.W.O.
40149 - 40170	0x0094 – 0x00A9			Reserved		

40171	0x00AA	Enum		P030 - Heating Module Type Major (16 bit)		C.W.O.
40172	0x00AB	Unsigned Word		Reserved		
40173	0x00AC	Unsigned Word		Reserved		
40174	0x00AD	Enum		P031 - Heating Module Type Minor (8 bit)		C.W.O.
40175	0x00AE	Unsigned Word		Reserved		
40176	0x00AF	Unsigned Word		Reserved		
40177	0x00B0	Unsigned Word		P032 - Heating Module Serial Number (MSB)		C.W.O.
40178	0x00B1	Unsigned Word		P032 - Heating Module Serial Number (LSB)		C.W.O.
40179 - 40202	0x00B2 – 0x00C9			Reserved		
40203	0x00CA	Enum		Appliance Configuration Phase Status – Succeeded (0x0000) – In Progress (0x0001) – Interface Locked (0x0002) – Error (0x0003)	R.O.	
40204	0x00CB	Boolean		Appliance Configuration Phase Start/Stop START (1) / STOP (0)		W.O.
40205	0x00CC	Unsigned Word	**APPLIANCE FUNCTIONAL PARAMETERS**	Parameter Index (Parameter Index Range: 40 ... 240)		
40206	0x00CD	Boolean		Parameter(Index) – Exists TRUE (1) / FALSE (0)	R.O.	
40207	0x00CE	Signed Word		Parameter(Index) - Min value	R.O.	
40208	0x00CF	Signed Word		Parameter(Index) - Max value	R.O.	
40209	0x00D0	Enum		Parameter(Index) - Access Level - User level (2) - System installer level (0) - Service level (1)	R.O.	
40210	0x00D1	Enum		Parameter(Index) - Type	R.O.	
40211	0x00D2	Boolean		Parameter(Index) - Write Access Allowed TRUE (1) / FALSE (0)	R.O.	
40212	0x00D3	Unsigned Word		Parameter(Index) - String Code	R.O.	
40213	0x00D4	Signed Word		Parameter(Index) - Value		
40214	0x00D5	Enum		Functional Parameter Write Status – Succeeded (0x0000) – In Progress (0x0001) – Failed (0x0002)	R.O.	
40215 - 40500	0x00D6 – 0x01F3			FREE (Not Mapped)		
40501	0x01F4	Unsigned Word	**APPLIANCE ANALOG DATA**	Analog Data Index (Analog Data Index Range: 0 ... <*Number of Analog Data*> - 1)		
40502	0x01F5	Boolean		Analog Data(Index) – Exists TRUE (1) / FALSE (0)	R.O.	
40503	0x01F6	Enum		Analog Data(Index) - Type	R.O.	
40504	0x01F7	Unsigned Word		Analog Data(Index) - String Code	R.O.	
40505	0x01F8	Analog Data Type Dependant		Analog Data(Index) - Value	R.O.	
40506	0x01F9	Unsigned Word		*Number of Analog Data*	R.O.	
40507 - 40700	0x01FA – 0x02BB			FREE (Not mapped)		

40701	0x02BC	Bitmap	**APPLIANCE STATUS**	Appliance Operating Status - bit 0: Appliance can Operate TRUE (1) / FALSE (0) - bit 1: Error Condition Active TRUE (1) / FALSE (0) - bit 2: Warning Condition Active TRUE (1) / FALSE (0) - bit 3: Flame ON - bit 4: Post ventilation Fan ON - bit 5: Thermostatation Fan ON - bit 6: Chimney sweep function active - bits 7 - 15: Unused (Read as 0)	R.O.	
40702	0x02BD	Bitmap		Appliance Digital Inputs Status (0: Open; 1: Closed) - bit 0: Limit Thermostat (error if open) - bit 1: Flue Thermostat (unused) (error if open) - bit 2: Unused - bit 3: Heating Request - bit 4: Ventilation Request / Heating Power Level Select - bit 5: Alarms Reset Request - bit 6 - 15: Unused ((Read as 0)	R.O.	
40703	0x02BE	Bitmap		Appliance Digital Outputs Status (0: off; 1: On) - bit 0: Gas Valve - bit 1: Combustion Blower Fan (rotating) - bit 2: Ignition Transformer (spark) - bit 3: Alarm relay - bit 4: Ventilation Speed 1 - bit 5: Ventilation Speed 2 - bit 6: Ventilation Speed 3 - bit 7: Ventilation Speed 4 - bits 8 - 15: Unused (Read as 0)	R.O.	
40704	0x02BF	enum		Appliance functional status - Off (0) - Ventilation (1) - Heating (2)	R.O.	
40705	0x02C0	enum		Burner functional status - Stand-by (0) - Pre-purge phase (1) - Ignition phase (2) - Flame Stabilization phase (3) - Full operation phase (4) - Post-ventilation phase (5) (6) - Forced post-ventilation phase (7)	R.O.	
40706	0x02C1	enum		Fan functional status - Fan Standby (0) - Fan Pre-ventilation Wait (1) - Fan Pre-ventilation (2) - Fan Full Operation (3) - Fan Normal Post Ventilation Start (4) - Fan Forced Post Ventilation (5) - Fan Summer Ventilation (6) - Fan Destratification (7)	R.O.	
40707 - 40800	0x02C2 – 0x031F			FREE (Not Mapped)		

40801	0x0320	enum	**MODBUS CONTROLLER COMMANDS**	Operation Request - Off (0) - Ventilation (Summer) (1) - Heating - Discrete Power Levels (2) - Heating - Modulating (Setpoint) (3) - Heating - Modulating (Power) (4) *Only relevant if P42= 1, i.e. remote controller installed.* *On board controller will reset this register to value 0 (Off) in case of Modbus timeout.*		
40802	0x0321	enum		Ventilation Speed in Ventilation (Summer) Mode - Ventilation Speed 1 (1) - Ventilation Speed 2 (2) - Ventilation Speed 3 (3) - Ventilation Speed 4 (4) *Only relevant if P42 = 1 and Operation Request (40801) = Ventilation (Summer)* *See Functional Parameter having the name "Standalone Ventilation Speed in Ventilation (Summer) Mode" for Ventilation Speed selection for stand-alone operation.*		
40803	0x0322	Signed Word		Heating Ambient Setpoint [°C/10] (range: -20.0 ÷ 50.0) *Only relevant if P 42 = 1 and Operation Request (40801) = Heating Discrete Power Levels or Heating Modulating (Setpoint)*		
40804	0x0323	enum		Heating Discrete Power Level - Discrete Power Level 1 (1) - Discrete Power Level 2 (2) - Discrete Power Level 3 (3) *Only relevant if P42 = 1 and Operation Request (40801) = Heating Discrete Power Levels.*		
40805	0x0324	Signed Word		Ambient Regulation Temperature [°C/10] (range: -40.0 ÷ 300.0) (0x7FFF if not available) (0x7FFE if fault probe) *Only relevant if P42 = 1, Operation Request (40801) = Heating Discrete Power Levels or Heating Modulating (Setpoint), and local Ambient Regulation Temperature Probe is not installed on the Appliance.*		
40806	0x0325	Unsigned Word		Heating Power Modulation Request [%] (range: 0 ÷ 100) *Only relevant if P42 = 1 and Operation Request (40801) = Heating Modulating (Power).* *Not scaled based on Power Modulation Min & Max Bounds, simply saturated (e.g. if value <= Power Modulation Min Bound, actual power will be Power Modulation Min Bound).*		
40807	0x0326	Unsigned Word		Alarms Reset Request Request (0x55AA) / Ignored (any other value) *Modbus master sends Request by writing Request value, then reads register 40808 to check command result.*		W.O.
40808	0x0327	enum		Alarms Reset Acknowledgment - Request Accepted (0x0000) - Request Not Supported (0x0001) - Request Refused (0x0002) *i.e. UI not idle* - Function currently disabled (0x0003) *due to EN14459, i.e. exceeded max number of flame lockout reset.* *May be retried after 15 minutes.* *Other alarms (if present) will be reset.*	R.O.	

40809	0x0328	enum		Destratification Request Off (0) On (1) *Only relevant if P42=1, P46 =1, and Operation Request (40801) = Heating Discrete Power Levels or Heating Modulating (Setpoint)*		
40810 - 40928	0x0329 – 0x039F			FREE (Not mapped)		
40929	0x03A0	Unsigned Word	**MODBUS CONTROLLER DATA**	Unix / POSIX Real Time Clock (MSW) *Used by Modbus master to provide RTC information to the Appliance; should be sent ASAP at each Power On, plus periodically.*		
40930	0x03A1	Unsigned Word		Unix / POSIX Real Time Clock (LSW) *Writing LSW commits new value.*		
40931	0x03A2	enum		0: RTC not updated since Power On 1: RTC updated	R.O.	
40932	0x03A3	enum		0: RTC never written 1: RTC written at least once since commissioning	R.O.	
40933 - 41000	0x03A4 – 0x03E7			FREE (Not mapped)		
41001	0x03E8	Unsigned Word	**CURRENTLY ACTIVE LEADING EVENT**	Leading Event - Event String Code *0 = no event*	R.O.	
41002	0x03E9	Unsigned Word		Leading Event - Event Code *0 = no event*	R.O.	
41003	0x03EA	enum		Leading Event – Event category (Information = 1, Warning = 2, Error = 3) *0 = no event*	R.O.	
41004 - 41010	0x03EB – 0x03F1			FREE (Not mapped)		
41011	0x03F2	Enum	**CURRENTLY ACTIVE EVENTS**	General Status - No Active Event (0) - Active Events (1) *If 0 is returned, Modbus master can suspend scanning the Event Slots*	R.O.	
41012	0x03F3	Unsigned Word		Event Slot 1 - Event String Code	R.O.	
41013	0x03F4	Unsigned Word		Event Slot 1 - Event Code and Flags	R.O.	
41014	0x03F5	Unsigned Word		Event Slot 2 - Event String Code	R.O.	
41015	0x03F6	Unsigned Word		Event Slot 2 - Event Code and Flags	R.O.	
41016	0x03F7	Unsigned Word		Event Slot 3 - Event String Code	R.O.	
41017	0x03F8	Unsigned Word		Event Slot 3 - Event Code and Flags	R.O.	
41018	0x03F9	Unsigned Word		Event Slot 4 - Event String Code	R.O.	
41019	0x03FA	Unsigned Word		Event Slot 4 - Event Code and Flags	R.O.	
41020	0x03FB	Unsigned Word		Event Slot 5 - Event String Code	R.O.	
41021	0x03FC	Unsigned Word		Event Slot 5 - Event Code and Flags	R.O.	
41022 - 41100	0x03FD – 0x044B			FREE (Not mapped)		

41101	0x044C	Boolean	**EVENTS LOGGER Permanent installation (on RS-485 Port)**	New Event Available TRUE (1) / FALSE (0) *Read to understand whether new Events are available for download.*	R.O.	
41102	0x044D	Unsigned Word		New Event - Timestamp MSW *Unix time format*	R.O.	
41103	0x044E	Unsigned Word		New Event - Timestamp LSW *Unix time format*	R.O.	
41104	0x044F	Unsigned Word		New Event - Event String Code	R.O.	
41105	0x0450	Unsigned Word		New Event - Event Code and Flags	R.O.	
41106	0x0451	Unsigned Word	Permanent Logger Commands	Counter Reset *Write to reset the New Events Counter so that all Events in Logger can be read again.*		W.O.
41107	0x0452	Unsigned Word		Read New Event Acknowledge *Acknowledge having read one New Event.*		W.O.
41108	0x0453	Unsigned Word	**EVENTS LOGGER Temporary installation (on RS-232 Port)**	Number of New Events to Read *Total Number of New Events that can be read.*	R.O.	
41109	0x0454	Unsigned Word		New Event - Timestamp MSW *Unix time format*	R.O.	
41110	0x0455	Unsigned Word		New Event - Timestamp LSW *Unix time format*	R.O.	
41111	0x0456	Unsigned Word		New Event - Event String Code	R.O.	
41112	0x0457	Unsigned Word		New Event - Event Code and Flags	R.O.	
41113	0x0458	Unsigned Word	Temporary Logger Commands	Counter Reset *Write to reset the New Events Counter so that all Events in Logger can be read again*		W.O.
41114	0x0459	Unsigned Word		Read New Event Acknowledge *Acknowledge having read one New Event.*		W.O.
41115 - 41200	0x045A – 0x04AF			FREE (Not mapped)		
41201	0x04B0	Unsigned Word	**APPLIANCE STATISTICS**	Total Electrical Power On Time (MSW) [s]	R.O.	
41202	0x04B1	Unsigned Word		Total Electrical Power On Time (LSW) [s]	R.O.	
41203	0x04B2	Unsigned Word		Total Flame On Time (MSW) [s]	R.O.	
41204	0x04B3	Unsigned Word		Total Flame On Time (LSW) [s]	R.O.	
41205	0x04B4	Unsigned Word		Total Ignitions Number (MSW)	R.O.	
41206	0x04B5	Unsigned Word		Total Ignitions Number (LSW)	R.O.	
41207	0x04B6	Unsigned Word		Total Missed Ignitions Number	R.O.	
41208 - 41500	0x04B7 – 0x05DB			FREE (Not mapped)		

GENERAL NOTES

Reading undefined registers (i.e. free, not mapped) is allowed, undefined value is returned.
Writing Undefined registers is illegal (Modbus will return "Illegal Data Address").

Reading Write-Only registers is allowed, undefined value is returned.
Writing Read-Only registers is illegal (Modbus will return "Illegal Data Address").

Writing Conditional-Write-Only (C.W.O.) register is allowed only when writing is enabled by specific Modbus command, otherwise write value is discarded without errors.

Signed Word registers use two's complement representation.

PLATFORM, OEM AND APPLIANCE IDENTIFICATION AND COMPATIBILITY CHECKS

Platform Type: read by the Modbus master to identify the general platform type it is communicating to and ensure it is the one supported (or one of those supported).
NOTE: Modbus Map compatibility among different Robur platforms is guaranteed only for this specific field.

OEM Code: read by the Modbus master to identify OEM and ensure it is the one expected for correct communication. This field is included to allow implementation of Modbus maps variants dedicated to specific partners.

OEM Compatibility Code Major: it must be included in a range of the Modbus master known code values to ensure compatibility. The Modbus master maintains such range and checks at start up if the read value is included in the range; if it is not included the Modbus master does not start normal operation as this implies it is not compatible with the appliance.
In a new GEN10 f/w release this value is incremented only when the new release breaks backward compatibility with the corresponding OEM Code Modbus master; it implies that the Modbus master needs to be updated.

OEM Compatibility Code Minor: if the Modbus master knows it then it can perform full operation, otherwise some features will not be supported; however, acceptable functionality is ensured.
Some possible missing features are new parameter descriptions, new analogs descriptions, new events descriptions, new status information and/or new control functionalities.
In a new GEN10 f/w release this value is incremented every time the new release adds and/or modifies features which do not break backward compatibility with the corresponding OEM Code Modbus master.

The Modbus master must also check “**Appliance type**” and “**Module type major**” to determine if it knows them. If either one, or both, are not known, the Modbus Master cannot perform normal operation as this also implies it is not compatible with the appliance.
On the other hand, “**Module type minor**” does not need to be checked for compatibility, as different values do not imply **any** variation in the appliance functionality offered to the Modbus

Master. However, if known by the Modbus master, it can be used to convey a more complete and accurate textual or graphical description of the appliance to the user.

BOARD PARAMETERS

These registers allow access to Read Only parameters that provide detailed information about the GEN10 On Board Controller HW and FW.
A priori Modbus master knowledge about location, format and meaning of these parameters is required.

APPLIANCE TYPE CONFIGURATION PARAMETERS

These registers allow access to Read / Conditional Write parameters that can be read to detect the exact unit type and Serial Number; they can also be conditionally written to perform a new Unit Type Configuration.

In order to do this the configuration phase must be enabled writing value 1 to register "Appliance Configuration Phase Start/Stop"; then, configuration parameters can be written (see tables below) and finally the configuration phase must be completed by writing value 0 to register "Appliance Configuration Phase Start/Stop".

After writing 0 to such register, the "Appliance Configuration Phase Status" register must be read in order to check whether the configuration succeeded; for more information, please refer to "Appliance Configuration Phase Status" register notes provided in the table above.

No clipping rule apply when writing configuration parameters and validity check is applied only after commit (i.e. Stop command); the only exception are the module serial number parameters where write value is clipped to max 999.999.999.
If an invalid set of parameters is written, the operation will fail and will need to be performed again.

A priori Modbus master knowledge about location, format and meaning of these parameters is required.

General notes for the Undefined Registers apply to the Reserved Registers.
Please note that void parameters (i.e. parameters never written, brand new GEN10) have value 0x7FFF.

Appliance type table:

P10 Appliance Type	Description
0	NEXT-G 20
1	NEXT-G 30
2	NEXT-G 35
3	NEXT-G 45
4	NEXT-G 60
5	NEXT-G 75
6	NEXT-G 90

Module type (major / minor) table:

P30 **Module type major**	**P31** **Module type minor**	**Description**	**Configuration details**
0	0	NEXT-G 20	Axial fan multi speed
0	1	NEXT-G 20	Centrifugal/Axial fan single speed
0	2	NEXT-G 20	Brushless fan
1	0	NEXT-G 30	Axial fan multi speed
1	1	NEXT-G 30	Centrifugal/Axial fan single speed
1	2	NEXT-G 30	Brushless fan
2	0	NEXT-G 35	Axial fan multi speed
2	1	NEXT-G 35	Centrifugal/Axial fan single speed
2	2	NEXT-G 35	Brushless fan
3	0	NEXT-G 45	Axial fan multi speed
3	1	NEXT-G 45	Centrifugal/Axial fan single speed
3	2	NEXT-G 45	Brushless fan
4	0	NEXT-G 60	Axial fan multi speed
4	1	NEXT-G 60	Centrifugal/Axial fan single speed
4	2	NEXT-G 60	Brushless fan
5	0	NEXT-G 75	Axial fan multi speed
5	1	NEXT-G 75	Centrifugal/Axial fan single speed
5	2	NEXT-G 75	Brushless fan
6	0	NEXT-G 90	Axial fan multi speed
6	1	NEXT-G 90	Centrifugal/Axial fan single speed
6	2	NEXT-G 90	Brushless fan

APPLIANCE FUNCTIONAL PARAMETERS

These registers allow access to the functional parameters that can be read or written to check or modify several appliance functions settings.
A parameter is accessed by first writing its Index value to register "Parameter Index", then accessing the relevant information provided in the following registers.
Out of range parameter index implies Exists = false.
If register “Parameter(Index) - Exists" returns FALSE, parameter does not exist and following information is meaningless.
If the parameter exists, the following information is provided in the following registers:

- Range (Min. and Max. values)
- Parameter access level
- Parameter Type, which specifies format and measurement unit (see Modbus data type table below)
- Parameter is Read Only or Read / Write
- String Code, which identifies a unique description string (association between string code and description will never change) (see Parameters strings table below)
- Parameter value, which can also be modified by writing to this register, provided the parameter is not Read Only.

Writing register “Parameter(Index) – Value” with a value which is not within allowable range is clipped to the relevant range bound.
After writing a parameter, the “Functional parameter write status” register must be read in order to check whether writing succeeded. See register notes in the table above for details.

Because of the information provided, Modbus master a priori knowledge of existence, location, format and meaning of these parameters is NOT required.

If a parameter does not exists, reading and writing Parameter Value register is still allowed: read value is undefined, written value is discarded.
If a parameter is Read-Only, writing Parameter Value register is still allowed and written value is discarded.
For all register in this category GENERAL NOTES apply.
Void parameters, i.e. parameters never written, have value 0x7FFF.

Modbus data types table:

Parameter Type	Description
0	Unsigned integer – Non Dimensional
1	Signed integer – Non Dimensional
2	Unsigned integer – Fixed Point (1 decimal) – Non Dimensional
3	Signed integer – Fixed Point (1 decimal) – Non Dimensional
4	Signed integer – Fixed Point (1 decimal) – Celsius Degree
5	Signed integer – Fixed Point (1 decimal) - Volt
6	Unsigned integer – Litres Per Hour
7	Boolean
8	Signed integer – Fixed Point (1 decimal) - Kelvin
9	Unsigned integer – Fixed Point (1 decimal) - Kilowatt
10	Unsigned integer – Fixed Point (2 decimal) - Percentage
11	Unsigned integer - Second
12	Unsigned integer - Minute
13	Unsigned integer - Hour
14	Signed integer – Fixed Point (1 decimal) – Micro Ampere

15	Unsigned integer - RPM,
16	Unsigned integer - RPMx10
17	Signed integer - RPMx10
18	Unsigned integer – Litres Per Hour X100
19	Unsigned integer - Percentage
20	Unsigned integer – Celsius Degrees By Minutes
21	Unsigned integer - Days

Parameters strings table

"001">Appliance modbus address
"002">Modbus timeout
"003">Remote controller installed
"004">Gas type
"005">Blower speed fine adjustment
"006">Internal ambient temperature probe installed
"007">Destratification temperature probe installed
"008">Outdoor temperature probe installed
"009">Analog input function
"010">Ambient temperature setpoint at 0V
"011">Ambient temperature setpoint at 10V
"012">Ambient temperature setpoint differential
"013">Building thermal inertia
"014">Ambient temperature setpoint
"030">Ventilation speed in cooling mode
"031">Ventilation management in heating mode
"032">Standalone heating mode
"033">Power modulation low bound percentage
"034">Power modulation high bound percentage
"053">heating PID KP
"054">heating PID KI
"055">heating PID KD
"056">Destratification differential temperature threshold
"057">Destratification max time
"058">RS485 communication parameters
"059">RS232 communication parameters

APPLIANCE ANALOG DATA

These Read Only Data deliver useful information about the Appliance operation.
Analog Data access is very similar to Functional Parameters access; after writing an Index value to register "Analog Data Index", existence of the specific Analog Data can be checked by reading register "Analog Data(Index) - Exists".

If Analog Data exists, the following information is provided:

- Analog Data Type, which specifies format and measurement unit (see Modbus data type table above, the same used for parameters)
- String Code, which identifies a unique description string (association between string code and description will never change) (see Analog data strings table below)
- Analog Data Value

Because of the information provided, Modbus master a priori knowledge of existence, location, format and meaning of these Analog Data is NOT required.
This also implies that these Appliance Analog Data must NOT be used to affect the Modbus master functionality, i.e. binding an analog data to a specific meaning and use.

If a given Analog Data does not exist, reading is still allowed and read value is undefined.
Writing Analog Data is always illegal (Modbus returns "Illegal Data Adrress").
If a given Analog Data has a fault condition, read value is:

- 0x7FFF for signed analogs
- 0xFFFF for unsigned analogs.

For all other register in this category GENERAL NOTES apply.

Analog data strings table

```
"001">Heat exchanger temperature</string>
"002">Flue temperature</string>
"003">Ambient temperature</string>
"004">Destratification temperature</string>
"005">Outdoor temperature</string>
"006">External voltage 1</string>
"007">Power percentage</string>
"008">Combustion blower speed</string>
"009">Ventilation speed</string>
"010">Current setpoint</string>
"011">Ionisation current</string>
```

APPLIANCE STATUS

These bitmapped and enum registers provide important Appliance Status information.
Description of the register contents is provided in registers table.
Modbus master a priori knowledge of location and meaning of these data is required.
The information provided can be used to affect the Modbus master functionality.

MODBUS CONTROLLER COMMANDS

These registers allow the Modbus master to perform actual Appliance Control.
To do this parameter P42 should be set to 1, i.e. Remote controller installed, except for “Alarm reset request” that is always available.

If timeout (P41) elapses without any Modbus activity, the Modbus controller command “Operation request” reset to value 0 (Off).

Setpoint. Ambient regulation temperature and Heating Power Percentage values which are not within the allowable ranges are clipped to the relevant range bound.
This values are expressed as Celsius Degrees Tenths. (e.g. value 10 means 1.0 °C).

Modbus master a priori knowledge of location and meaning of these data is required.

MODBUS CONTROLLER DATA

Presently this category contains only registers to get and set Real time clock information.
Real time clock uses the Unix/Posix format, i.e. number of seconds elapsed from 01.01.1970 00.00.00.

More generally registers in this category contain data which are exchanged between the Modbus master and the Appliance to support conjunct Regulation and Control.

Modbus master a priori knowledge of location and meaning of these data is required.

Please note that the Appliance does not have an internal RTC. However, the Modbus master can provide Real Time data at any Power ON and then periodically in order to resynchronize the Appliance internal software time counter.
The Appliance Event Logger timestamps always use the software timer: if it's updated since last power-on the event flag will be set to show that proper timestamp is recorded, otherwise the flag is 0 and timestamp recorded is obsolete.
In both cases the Event Logger timestamp uses the Unix/Posix format.

At each Power On, timestamps are qualified as "RTC not updated since Power On" till the Appliance receives the first Real Time data update from the Modbus master.

CURRENTLY ACTIVE LEADING EVENT

Used to get simplified information about active events.

- When one or more events are active, these registers report only the most important event.
- See *Event string code table* below for event description string binding with string code.

CURRENTLY ACTIVE EVENTS

Used to get full information about active events.

- Up to five alarm conditions can be simultaneously reported in five event slots.
- Every slot has two registers: “Event string code” and “Event code and flags”.
- See *Event string code table* below for event description string binding with string code.
- See table below for “Event code and flag” register bit fields.
- Events have a 3 digit “Event code” that is showed on the appliance display.
- Events have three (currently two) increasing severity levels: Info (currently not used), Warning, Error.
- Mask event flag is used to specify events not intended for end user.

Event Code and Flags	
Bit	Description
15 -- 6	Event Code
5 -- 4	Always read as 0
3 -- 2	00 No Event 01 Information 10 Warning 11 Error
1	Always reads as 0
0	Mask Event To End User - MASK (1) - DO NOT MASK (0)

EVENTS LOGGER

Used to access event logger information managed by GEN10.

- The internal Events (alarms) Logger, storing up to 128 event records, can be accessed through the Modbus Interface.
- There are two registers group, intended for two independent Modbus masters acting independently, one for permanent installation and one for temporary installation.
- The Appliance internal Event Logger records can be retrieved by the Modbus master through the "Permanent Logger" interface.
- The Modbus master periodically polls the "New Event Available" register; if it’s TRUE the Modbus master reads the "Timestamp", "Event String Code" and "Event Code and flags" registers; then, it writes to the "Read New Event Acknowledge" register.
- If the Modbus master does not store events record information in non volatile memory, it can write (e.g. “1”) to the "Counter Reset" register to force the Appliance to consider all records as "Not Read".
- See *Event string code table* below for event description string binding with string code.
- See table below for “Event code and flag” register bit fields.
- The “Temporary Logger" interface provides a similar mean to access the Appliance Events Logger from a Diagnostic Tool connected to the Appliance RS232 Port. Downloading event records from this interface does not affect the Unread event records counter of the other interface (and viceversa). However, connection of the Diagnostic Tool to the RS232 Port does interrupt communication between the Modbus master and the Appliance.
- For “Temporary Logger", the “Number of New Events to Read” register contains the number of unread events, i.e. events that have not been read through the Modbus Interface since they were generated. For "Permanent Logger" the "New Event Available" is true if at least one event is unread.

Event Code and Flags	
Bit	**Description**
15 -- 6	Event Code
5 -- 4	00 Event Termination 01 Event Occurence 10 Instantaneous Event 11 Reserved
3 -- 2	00 No Event 01 Information 10 Warning 11 Error
1	0: RTC not updated since Power On / 1: RTC updated
0	Always reads as 0

Events string codes table

"01">Appliance powered on
"02">New appliance configuration
"06">Limit thermostat trip
"07">Flue gas thermostat trip
"14">Flame controller lockout
"15">Flame controller communication error
"21">Flue gas temperature probe fault
"26">Blower fault
"30">Parasitic flame lockout
"31">Heat exchanger overtemperature
"32">Low flue temperature
"33">Ambient probe required
"34">Heat exchanger probe fault
"35">Ambient probe fault
"36">Modbus communication lost
"37">Invalid parameter set - corrupted memory
"38">Parameter memory fault
"41">Incomplete functional parameters
"42">Invalid configuration parameters
"46">Invalid module type configuration parameters
"53">Flame controller not compatible
"54">Flame controller parameters error
"55">Internal flame controller error
"56">Flame loss
"57">Flame controller communication error
"58">Destratification probe fault
"59">Missing board serial or HWKey

APPLIANCE STATISTICS

These registers convey information about Appliance statistics.
Modbus master a priori knowledge of location and meaning of these data is required.

All 32 bit counters uses two Modbus registers:

- MSW i.e. most-significant word
- LSW i.e. least-significant word

To get a counter value Modbus master shall read both registers and apply formula:
Counter = (MSW * 65536) + LSW

All time counters are expressed in seconds.

Revision history

- Modbus interface f/w 1.002
  - V.0 - 23.03.2022
    - First Release
  - V.1 – 30/03/2022
    - Modified “Modbus Communication” section
    - Renamed and modified “OEM Identification….” subsection
    - Added description for parameters string codes 58 and 59
    - Applied a few other minor changes/fixes

- Modbus interface f/w 1.003
  - V.1.003.MB0 - 22.04.2022
    - Modified references to GEN10 FW Release
    - Contents identical to V.1