we make green energy work

Embion Modbus interface

Embion B.V.

Reference date: 20 June 2025
Author: Rik Baeten
Status: Accepted

www.embion.nl

Embion Modbus interface

Document change log

Revision	Release date	Remark
V0.1.0	9-12-2024	First unofficial release
V0.2.0	21-1-2025	Minor changes and add registers to definitions
V1.0.0	3-2-2025	First official release
V1.0.1	28-2-2025	Update Embion logo in document
V1.1.0	19-5-2025	Add external load register (supported from V5.0.0)

A.1. Compatibility

The Modbus client function is available in the SolarGateway from software version 4.2.0. In the Setup TCP section of the Modbus menu activate Client & server option to also enable the client function.
Currently only Modbus TCP is supported, the port and unit id are configurable in the Setup TCP section of the Modbus menu.

A.2. Contents

Introduction

This document outlines the Modbus protocol specification designed for interaction with the SolarGateway Energy Management System (EMS). The SolarGateway EMS enables advanced energy monitoring and control in high-end residential and business environments. By using Modbus, the SolarGateway facilitates real-time data acquisition and control over connected devices, such as inverters, meters, EV chargers, and batteries.

The protocol supports a wide range of PlantControl features. These features empower users to optimize energy consumption, balance grid interactions, and control connected assets dynamically. Leveraging Modbus communication, the SolarGateway allows for robust and reliable integration with third-party systems, allowing for full customization and scalability.

1.1. Clarification of Modbus Registers and Usage

The Modbus registers in this document are structured to simplify system implementation and operation. The following categories are used:

1. **Input Registers (Read-Only)**
   Input registers can be accessed using function code 0x04. These registers provide real-time monitoring data, such as grid measurements, device statuses, and energy statistics. They cannot be modified.
2. **Holding Registers (Read/Write)**
   Holding registers can be read using function code 0x03 and written to using function codes 0x06 or 0x10. These registers are used for setting control parameters, such as power limits, system modes, and asset-specific configurations.
3. **Address Assignment**
   Each register is assigned a unique address within its category. The addresses are sequential and grouped logically based on functionality, ensuring ease of implementation.
4. **Data Types and Units**
   Data types, such as Uint16, Float32, and STRING, are used to represent values in the registers. Units (e.g., kW, kWh, %, etc.) are specified. Where applicable, ENUM values are defined to map numeric codes to operational states.
5. **Endianness**
   The data retrieved or written follows the Modbus standard, using big-endian format for both word order and byte order.
6. **Hex and decimal values**
   Values starting with 0x represent the value as Hexadecimal
   Values not starting with 0x represent Decimal values.

2. Content

The content section provides a detailed overview of all Modbus registers used to control and monitor the SolarGateway EMS.

2.1. Input Registers (Read-only registers)

5

Address	Number of (16bit) registers	Description	Type	Unit
0x100	10	PN of SolarGateway	STRING (UTF8)	-
0x10A	10	SN of SolarGateway	STRING (UTF8)	-
0x114	3	Actual software version	vx.x.x = Uint16.Uint16.Uint16	-
0x117	3	Product hardware version	vx.x.x = Uint16.Uint16.Uint16	-
System status				
0x1000	1	# of configured assets	Uint16	-
0x1001	1	# of online assets	Uint16	-
0x1002	1	Device status	Uint16	ENUM (0=Unconfigured 1=ERROR 2= Running)
Plant information				
0x1100	2	Fixed grid import power limit	Float32	kW
0x1102	2	Fixed grid export power limit	Float32	kW
0x1104	2	Fixed grid import/export current limit	Float32	A
0x1106	2	Grid active power	Float32	kW
0x1108	2	Grid phase 1 current	Float32	A
0x110A	2	Grid phase 2 current	Float32	A
0x110C	2	Grid phase 3 current	Float32	A
0x110E	2	Grid phase 1 voltage	Float32	V
0x1110	2	Grid phase 2 voltage	Float32	V
0x1112	2	Grid phase 3 voltage	Float32	V
0x1114	2	Grid import energy (total)	Float32	kWh
0x1116	2	Grid export energy (total)	Float32	kWh
0x1118	2	Total configured PV power	Float32	kW
0x111A	2	Total configured battery power	Float32	kW
0x111C	2	Actual PV power	Float32	kW
0x111D	2	Daily PV yield	Float32	kWh
0x1120	2	Total PV yield	Float32	kWh
0x1122	2	Actual reduction PV	Float32	%
0x1124	2	Actual battery power	Float32	kW (-discharge +charge)
0x1126	2	Average battery SoC	Float32	%
0x1128	2	Reserved	-	-
0x112A	2	Reserved	-	-
0x112C	2	Actual EV power	Float32	kW
0x112E	2	Total EV charge energy	Float32	kWh
0x1130	1	GP input 1 state	Uint16	Bool (0=Low, 1=High)
0x1131	1	GP input 2 state	Uint16	Bool (0=Low, 1=High)
0x1132	1	GP output 1 state	Uint16	Bool (0=False, 1=True)
0x1133	1	GP output 2 state	Uint16	Bool (0=False, 1=True)
0x1140	2	External load setpoint	Float32	%

2.2. Holding Registers (Read/write)

Address	Number of registers	Description	Type	Default	Unit
0x1000	2	Grid import limit	Float32	NaN[2]	kW
0x1002	2	Grid export limit	Float32	NaN[2]	kW
0x1004	2	Relative grid import limit	Float32	NaN[2]	%
0x1006	2	Relative grid export limit	Float32	NaN[2]	%
0x1008	1	Control plant generation	Uint16	0	ENUM (0=NOM 1=MIN 2=MAX)
0x1009	1	Control plant consumption	Uint16	0	ENUM (0=NOM 1=MIN 2=MAX)
0x100A	2	Control plant PV limit	Float32	NaN[2]	%
0x100C	2	Control plant battery power setpoint	Float32	NaN[2]	% (-discharge +charge)
0x100E	2	Control plant EV limit	Float32	NaN[2]	%
0x1010	1	Watchdog timer[1]	Uint16	60	S (min 1, max 300s)
0x1011	1	Clear PlantControl (reset all Holding registers to default)	Uint16	0	Write 1 to reset

Note[1]: When the timer reaches 0 all settings written by Modbus are reset. The user application needs to write at least one of the holding registers (0x1000 to 0x100E) to reset the watchdog timer to the preset value.

Note[2]: For all Float32 values, NaN is defined as 0xFFFFFFFF.

Note: PlantControl actions given through the Modbus interface will always override the PlantControl settings done via the public API.

Note: If conflicting setpoints are provided via Modbus, the setpoint resulting in the lowest power output will take precedence.