

PGVA Manual

Version 5.

(2021.03.15)



Festo SE & Co. KG

Ruiter Straße 82
72734 Esslingen
Germany
Internet: www.festo.com

Table of contents

PGVA Manual	1
Table of contents.....	2
Specifications overview.....	3
1 Complete System.....	3
1.1 Brief Description	3
1.2.8 Using Pressure over Liquid.....	4
1.2.9 Using as Open Loop Pipette	4
1.3.1 Customer interface	5
2 Communications.....	6
2.1 Getting Started using the GUI	6
2.2 Modbus Settings.....	8
2.3 Modbus Register Map	8
2.4 Modbus Communication	20
2.5 PLC Settings.....	20
2.4.1 basic settings.....	20

Specifications overview

1 Complete System

The decentralized Pressure & Vacuum Box "PVGA-..." is designed to supply Festo pipetting system with compressed air and vacuum.

- Communication can be accomplished via Modbus TCP (Ethernet), or Modbus ASCII (serial RS232). This makes it possible to use I / O signals to coordinate pipetting actions. What enables the suction and delivery of liquids as well as the programming of various setpoints.
- P&V Box may also be paired with common LifeTech products such as pressure-driven dispense heads (VTOE, VTOI, VTOF)

1.1 Brief Description

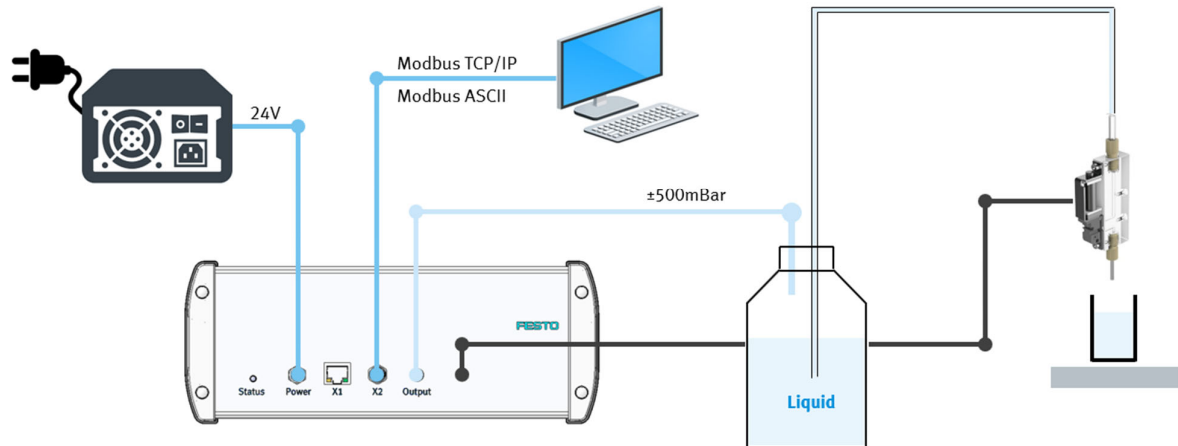
Description:

The Pressure Vacuum Generator PGVA-CS can create compressed air and vacuum (not at the same time). The intended use is in liquid handling applications in the life sciences, but the usage can be extended to other areas as well. In combination with a pipette module or a dispense head as a front end, PGVA-CS may be used to aspirate and dispense liquids by applying the principle of "pressure over liquid". The amount of liquid dispensed/aspirated is a function of the pressure/vacuum level generated by PGVA-CS as well as the time, these values are applied.

The Pressure Vacuum Generator needs a 24VDC power supply and may be operated either with a computer by using a graphical user interface or over a PLC using Modbus protocol. Since the proportional pressure regulator (Festo product, including piezo-valves) is integrated into the PGVA-CS, appropriate measures have been taken into account to avoid the over-aspiration of liquids and damage to the system.

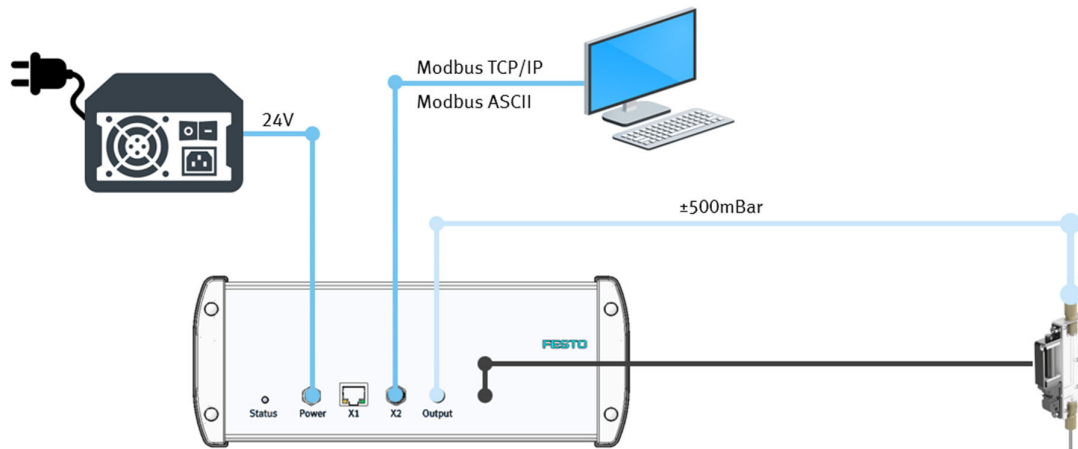
1.2.8 Using Pressure over Liquid

Description *(Draft)*:



1.2.9 Using as Open Loop Pipette

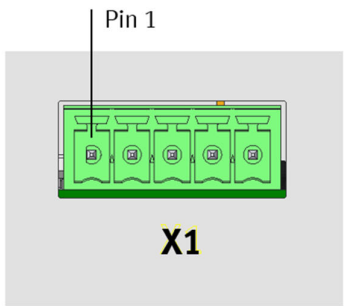
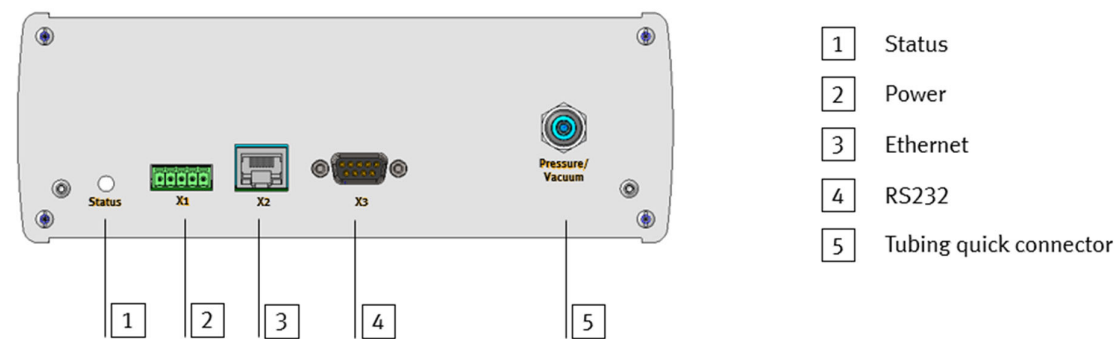
Description *(Draft)*:



NOTE: The VTOE in the picture should have a Tip adapter and Disposable tip connected

1.3.1 Customer interface

Description:

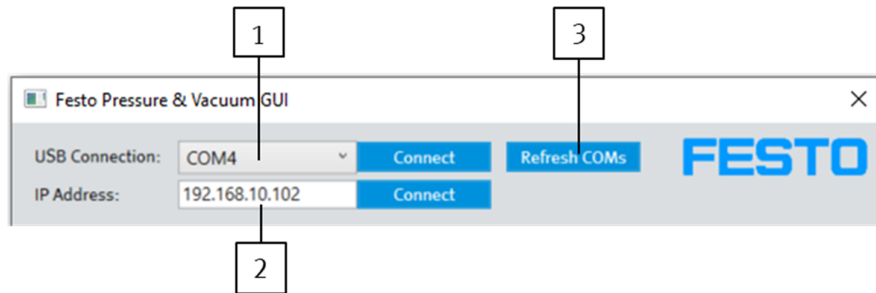


X1 - Power	
Pin 1	+24V DC
Pin 2	GND
Pin 3	FE
Pin 4	Output
Pin 5	GND

2 Communications

2.1 Getting Started using the GUI

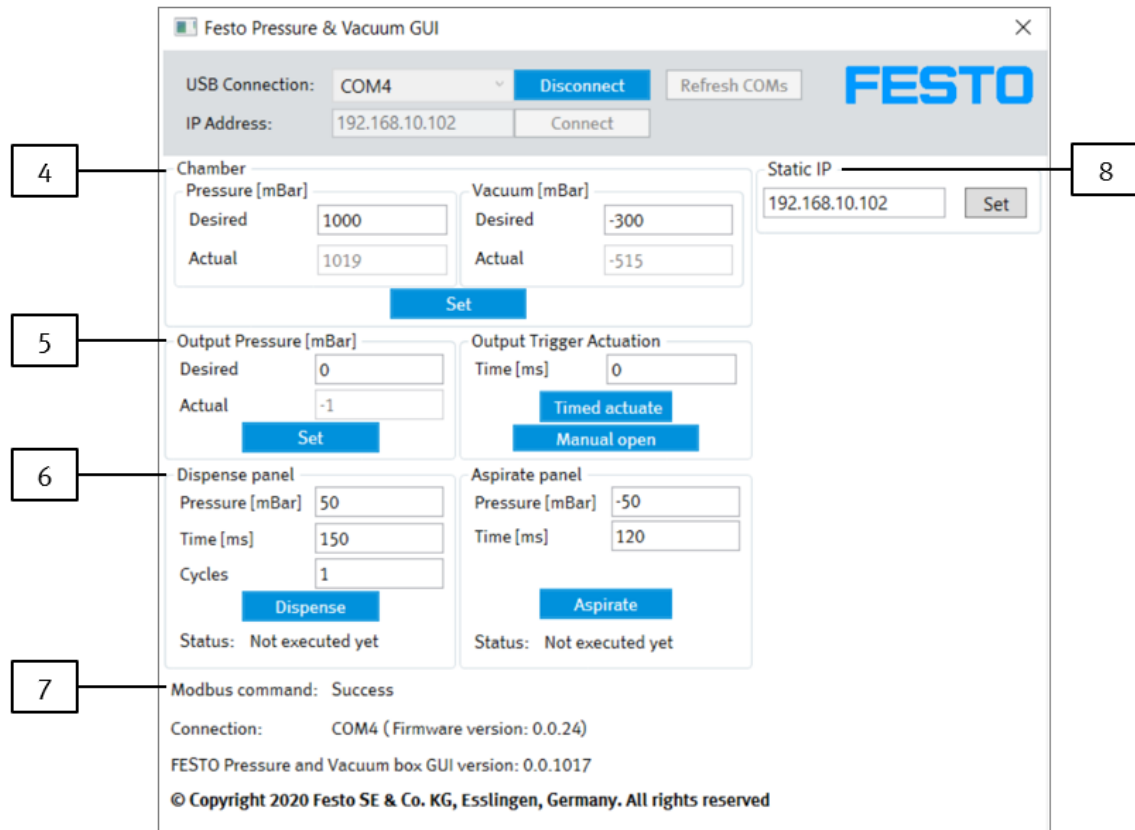
There are two ways to establish communication with the GUI. Both are shown in the following table below. If a connection hasn't been established, many of the GUI elements will remain grayed out.



1. Select the COM port that will be used for ASCII Modbus communication with the PGVA
2. Select the IP address of the PGVA
3. Refreshes the COM ports. Can be used when the serial cable is plugged into the PC after the GUI application is started.
4. Chamber panel.
 1. "Desired" textbox - user enters pressure in millibars and presses the "Set" button. Once the pressure for the selected chamber (pressure or vacuum) is reached.
 1. The individual pressure range is between 200 and 1000 [mbar].
 2. The individual vacuum range is between -160 and -640 [mbar].
 2. pump **stops** pressurizing that chamber. Note: Firmware implements hysteresis:
 1. The pump will start pressurizing the pressure chamber if the pressure in it falls below "Desired" - 100 mbar.
 2. The pump will start pressurizing the vacuum chamber if the pressure in it goes above "Desired" + 40 mbar.
 3. "Actual" textbox - displays the current pressure in the corresponding chamber (pressure or vacuum) in millibars
5. Output pressure panel
 1. "Desired" textbox - user enters the desired pressure in millibars and presses the "Set" button. The device will command via DAC the VEAB to regulate the pressure set by the user.
 2. "Actual" textbox - displays the current pressure regulated by the VEAB via ADC in millibars
 3. The additional output trigger can either be used as a time-dependent actuation or as a manual on / off switch.

Note:

In case the connection will be made over ethernet, please consider the following note. The PC must be on the same network as the Box. **The Ethernet cable must be plugged in before starting the box.** If the Ethernet cable is plugged in after the PGVA has started Ethernet connection will **not** be possible.



6. Dispense and aspirate panels

1. "Pressure" - user enters in millibars the desired pressure to regulated.
2. "Time" - user enters in milliseconds the duration for which the PGVA output will be activated (output valve opened)
3. "Cycles" - user enters the number of cycles for which the operation will be repeated
4. "Dispense" / "Aspirate" buttons - after users presses that button the following actions are taken:
 1. Regulation of the "Desired" pressure.
 2. PGVA output is activated for the "Time".
 3. Repeat from step 1. "Cycles" number of times.
 - 4.

7. Displays whether the last Modbus command was successful or not.

8. "Static IP" textbox displays the current IP address to be used as a fallback if DHCP fails. Users can alter that address and press the "Set" button. The IP address will be persistently stored on the device and the user will be notified to reset the device in order for the new setting to effect. In case of individual adjustments:

- the IP address should not be 0.x.x.x, 127.x.x.x and neither a value $\geq 240.x.x.x$.
- MAC address should be in Festo range (00:0E:F0:00:00:00 - 00:0E:F0:FF:FF:FF).

Modbus unit ID should be in the range [1, 247].

2.2 Modbus Settings

Description:

1	Modbus ASCII over RS232 (USB Serial)	Baudrate: 115200 bps Databits: 8 Parity: NONE Stop bits: 1
	Modbus TCP over Ethernet	Modbus TCP Media: Ethernet Node-address: 0x10 Port: 8502 Default static IP address: 192.168.10.102 (address can be changed if needed)

2.3 Modbus Register Map

his version Ethernet TCP/IP is configured with default coded address: 192.168.10.102, which can be changed if needed.

Implemented Modbus ASCII commands:

- Modbus Function Code 03
- Maximum number of registers to read 4
- Modbus Function Code 04
- Maximum number of registers to read 4
- Modbus Function Code 06
- Modbus Function Code 10
- Maximum number of registers to write 4

Limitations: none

Modbus registers address spaces

Register name and address	Type	Offset	Scaling	Action when read	Action when written
Vacuum actual ADDR: 0x100	Input register Dimension: mbar	- 1498.334	1.332	Acquires actual pressure in the vacuum tank and returns it as Modbus slave FC04 reply	N/A
Pressure actual ADDR: 0x101	Input register Dimension: mbar	- 1498.334	1.332	Acquire actual pressure in the pressure tank	N/A
Output pressure actual ADDR: 0x102	Input register Dimension: mbar	-500	0.244	Acquire actual pressure and vacuum box output pressure	N/A
Firmware version.version ADDR: 0x103	Input register Dimension: <version>. <sub_version>. <build>	0	1	Returns firmware version (hardcoded value in code flash)	N/A
Firmware version.sub_version ADDR: 0x104	Input register Dimension: <version>. <sub_version>. <build>	0	1	Returns firmware sub_version (hardcoded value in code flash)	N/A
Firmware version.build ADDR: 0x105	Input register Dimension: <version>. <sub_version>. < build>	0	1	Returns firmware build (hardcoded value in code flash)	N/A
PGVA status word ADDR: 0x106	Input register Bitfield. Please refer to Table 1. below.	0	1	Reads the status of the PGVA	N/A
Dispense valve opening times (MSH): ADDR: 0x107	Input register Dimension: Upper 16 bits of a 32-bit integer	0	1	Reads the most-significant halfword of the 32-bit counter for the total number	N/A

				of dispense valve openings.	
Dispense valve opening times (LSH): ADDR: 0x108	Input register Dimension: Lower 16 bits of a 32-bit integer	0	1	Reads the least-significant halfword of the 32-bit counter for the total number of dispense valve openings	N/A
Pump life counter (MSH): ADDR: 0x109	Input register Dimension: Seconds. Upper 16 bits of a 32-bit integer.	0	1	Reads the most-significant halfword of the 32-bit pump life counter (resolution is 1 sec).	N/A
Pump life counter (LSH): ADDR: 0x10A	Input register Dimension: Seconds. Lower 16 bits of a 32-bit integer.	0	1	Reads the least-significant halfword of the 32-bit pump life counter (resolution is 1 sec).	N/A
PGVA life counter (MSH): ADDR: 0x10B	Input register Dimension: Minutes. Upper 16 bits of a 32-bit integer.	0	1	Reads the most-significant halfword of the 32-bit PGVA life counter (resolution is 1 min).	N/A
PGVA life counter (LSH): ADDR: 0x10C	Input register Dimension: Minutes. Upper 16 bits of a 32-bit integer.	0	1	Reads the least-significant halfword of the 32-bit PGVA life counter (resolution is 1 min).	N/A
Vacuum actual (in mbar) ADDR: 0x10D	Input register Dimension: mbar	0	1	Same reading as register 0x100, however, the conversion between	

				increments and millibar is done in the firmware.	
Pressure actual (in mbar) ADDR: 0x10E	Input register Dimension: mbar	0	1	Same reading as register 0x101, however, the conversion between increments and millibar is done in the firmware.	
Output pressure actual (in mbar) ADDR: 0x10F	Input register Dimension: mbar	0	1	Same reading as register 0x102, however, the conversion between increments and millibar is done in the firmware.	
Min DAC increments ADDR: 0x110	Input register	0	1	Reads the DAC increments at min. calibration setpoint (see register 0x1004)	
Zero DAC increments ADDR: 0x111	Input register	0	1	Reads the DAC increments at zero calibration setpoint (see register 0x1005)	
Max DAC increments ADDR: 0x112	Input register	0	1	Reads the DAC increments at max. calibration setpoint (see register 0x1006)	
Last Modbus error ADDR: 0x113	Input register	0	1	Reads the last Modbus command error. See table 2. below	
Seed ADDR: 0x114	Input register	0	1	Reads the seed generated by the device	

Man mode ADDR: 0x115	Input register	0	1	0: Device is not in manufacturing mode 1: Device is in manufacturing mode	
Valve actuation time ADDR: 0x1000	Holding register Dimension: ms	0	1	Reading this register will always return: 0xFFFF.	Actuate output electromagnetic valve with a period matching value to write in Modbus frame Note: output electromagnetic valve is an option for the product - it might be installed on devices used to demonstrations
Vacuum threshold* ADDR: 0x1001 (lowest pressure at which vacuum pump is deactivated after regulating vacuum tank pressure)	Holding register Dimension: mbar	When writing: 1124.5 When reading: - 1498.334	When writing: 0.7505 When reading: 1.332	Returns vacuum threshold	Sets vacuum threshold. Note: the SW implements hysteresis in vacuum regulator - the value from Modbus command is applied immediately but, in some cases, if the change is in lower than hysteresis the device will react at the moment pressure reached the vacuum threshold
Pressure threshold* ADDR: 0x1002 (highest pressure at which air pump is deactivated after filling up pressure tank)	Holding register Dimension: mbar	When writing: 1124.5 When reading: - 1498.334	When writing: 0.7505 When reading: 1.332	Returns pressure threshold	Sets pressure threshold. Note: the SW implements hysteresis in pressure regulator - the value from Modbus command is applied immediately but, in some cases, if the change is in lower than hysteresis the device will react at the moment pressure reached the threshold
Output pressure ADDR: 0x1003 (output pressure set point value)	Holding register Dimension: mbar	When writing: 2047.5 When reading: - 500	When writing: 4.095 When reading: 0.244	Returns output pressure set point value	Sets output pressure set point value. Note: The set point value is applied for less than 2.5ms after the Modbus command is received. However, the output pressure regulator will need some time to regulate the output value. This time depends on the difference between new and

					old set point values, the output flow rate and how full/empty are pressure tank and vacuum tank
Min calib. setpoint* ADDR: 0x1004	Holding register Dimension: LSB	0	1	Read externally measured pressure for minimum calibration setpoint.	Set externally measured pressure for minimum calibration setpoint. Also stores the increments corresponding to that pressure. Writes the value to internal memory
Zero calib. setpoint* ADDR: 0x1005	Holding register Dimension: LSB	0	1	Read externally measured pressure for zero calibration setpoint	Set externally measured pressure for zero calibration setpoint. Also stores the increments corresponding to that pressure. Writes the value to internal memory
Max calib. setpoint* ADDR: 0x1006	Holding register Dimension: LSB	0	1	Read externally measured pressure for zero calibration setpoint	Set externally measured pressure for maximum calibration setpoint. Also stores the increments corresponding to that pressure. Writes the value to internal memory
Reserved ADDR: 0x1007	Holding register Dimension: LSB	N/A	N/A		
Reserved ADDR: 0x1008	Holding register Dimension: LSB	N/A	N/A	.	
Reserved* ADDR: 0x1009	Holding register	N/A	N/A		
Jump to first bootloader ADDR: 0x100A	Holding register Dimension: LSB	0	1	N/A	Value: 9999 ->Reboots the device in bootloader mode.

Jump to second bootloader ADDR: 0x100B	Holding register Dimension: LSB	0	1	N/A	Value: number of the downloaded firmware records ->Reboots the device to second bootloader. Automatic jump back to the application if the firmware update is successful.
Modbus TCP Port** ADDR: 0x100C	Holding register Dimension: LSB	0	1	Reads the TCP port for Modbus communication	Sets the TCP port for Modbus communication.
Modbus unit ID ADDR: 0x100D	Holding register Dimension: LSB	0	1	Read the Modbus unit ID of the PGVA	Sets the Modbus unit ID of the PGVA
Vacuum threshold (in mbar)* ADDR: 0x100E (lowest pressure at which vacuum pump is deactivated after regulating vacuum tank pressure)	Holding register Dimension: mbar	0	1	Reading the register is the same as register 0x1001, however, conversion between increments and millibars is done in the firmware.	Writing to this register has the same effect as writing to register 0x1001, however, value written will be interpreted as millibars instead of increments.
Pressure threshold (in mbar)* ADDR: 0x100F (highest pressure at which air pump is deactivated after filling up pressure tank)	Holding register Dimension: mbar	0	1	Reading the register is the same as register 0x1002, however, conversion between increments and millibars is done in the firmware.	Writing to this register has the same effect as writing to register 0x1002, however, value written will be interpreted as millibars instead of increments.
Output pressure (in mbar) ADDR: 0x1010 (output pressure set point value)	Holding register Dimension: mbar	0	1	Reading the register is the same as register 0x1003, however, conversion between increments and millibars is done in the firmware.	Writing to this register has the same effect as writing to register 0x1003, however, value written will be interpreted as millibars instead of increments.

Manual trigger ADDR: 0x1011	Holding register Dimension: 0 or 1	0	1	Reads the current state of the output.	Writing 1 to this register opens the output valve, 0 – closes it
Store to EEPROM ADDR: 0x1064	Holding register Dimension: don't care	N/A	N/A	N/A	Stores the configuration registers (marked with * and **) to the EEPROM. Value written to this register is meaningless.
Reset pump life counter ADDR: 0x1065	Holding register Dimension: don't care	N/A	N/A	N/A	Resets the life counter for the pump.
Start pre-calibration ADDR: 0x1066	Holding register Dimension: don't care	N/A	N/A	N/A	Starts device's pre-calibration procedure in order to precisely determine the voltage range.
Adjust calibration setpoints ADDR: 0x1067	Holding register Dimension: LSB	0	1	N/A	Sets the output pressure depending on the value written: 0: close to the minimum setpoint 1: close to the zero setpoint 2: close to the maximum setpoint
Authorization start ADDR: 0x10C8	Holding register Dimension: 9999	0	1	N/A	Starts authentication procedure for Manufacturing mode
Key write ADDR: 0x10C9	Holding register Dimension: LSB	0	1	N/A	Writes the generated key from the seed
Config ongoing ADDR: 0x10CA	Holding register Dimension: 9999	0	1	N/A	Keeps the manufacturing mode ongoing

Firmware download address ADDR: 0x2000 ... 0x2001	Multiple registers Dimension: LSB	0	1	N/A	Value: Record address of the new firmware data on device (UINT32)
Firmware download data ADDR: 0x2002 ... 0x2009	Multiple registers Dimension: LSB	0	1	N/A	Value: Record data of the new firmware to download (16 bytes)
Byte size of the new firmware ADDR: 0x200A ... 0x200B	Multiple registers Dimension: LSB	0	1	N/A	Value: byte quantity of the new firmware (UINT32)
IP Address (MSH)**,* ADDR: 0x3000	Multiple registers Dimension: LSB	0	1	Read the most-significant 16-bits if the PGVA static IP address	Set the most-significant 16-bits if the PGVA static IP address.
IP Address (LSH)**,* ADDR: 0x3001	Multiple registers Dimension: LSB	0	1	Read the least-significant 16-bits if the PGVA static IP address	Set the least -significant 16-bits if the PGVA static IP address.
GW Address (MSH)**,* ADDR: 0x3002	Multiple registers Dimension: LSB	0	1	Read the most-significant 16-bits if the PGVA static gateway address	Set the most-significant 16-bits if the PGVA static gateway address.
GW Address (LSH)**,* ADDR: 0x3003	Multiple registers Dimension: LSB	0	1	Read the least-significant 16-bits if the PGVA static gateway address	Set the least -significant 16-bits if the PGVA static gateway address.
Netmask (MSH)**,* ADDR: 0x3004	Multiple registers Dimension: LSB	0	1	Read the most-significant 16-bits if the PGVA static netmask	Set the most-significant 16-bits if the PGVA static netmask.
Netmask (LSH)**,* ADDR: 0x3005	Multiple registers Dimension: LSB	0	1	Read the least-significant 16-bits if the PGVA static netmask	Set the least -significant 16-bits if the PGVA static netmask.
MAC Address (A)**,* ADDR: 0x3006	Multiple registers Dimension: LSB	0	1	Read the most-significant 16-bits if the PGVA MAC address	Set the most-significant 16-bits if the PGVA MAC address

	AA:AA:xx:xx:xx:xx				
MAC Address(B)** , *** ADDR: 0x3007	Multiple registers Dimension: LSB xx:xx:BB:BB:xx:xx	0	1	Read the middle 16-bits if the PGVA MAC address	Set the middle 16-bits if the PGVA MAC address
MAC Address(C)** , *** ADDR: 0x3008	Multiple registers Dimension: LSB xx:xx:xx:xx:CC:CC	0	1	Read the least-significant 16-bits if the PGVA MAC	Set the least-significant 16-bits if the PGVA MAC address
Production year* ADDR: 0x3009	Multiple registers Dimension: year	0	1	Read the production year of the device	Set the production year of the device
Production date/month* ADDR: 0x300A	Multiple registers Dimension: year MSB – month LSB - day	0	1	Read the production month and date of the device	Set the production month and date of the device
Production number (MSH)* ADDR: 0x300B	Multiple registers Dimension: LSB	0	1	Read the most-significant 16 bits of the device's production number	Set the most-significant 16 bits of the device's production number
Production number (LSH)* ADDR: 0x300C	Multiple registers Dimension: LSB	0	1	Read the least-significant 16 bits of the device's production number	Set the least-significant 16 bits of the device's production number
Serial number (MSH)* ADDR: 0x300D	Multiple registers Dimension: LSB	0	1	Read the most-significant 16 bits of the device's serial number	Set the most-significant 16 bits of the device's serial number

Serial number (LSH)* ADDR: 0x300E	Multiple registers Dimension: LSB	0	1	Read the least-significant 16 bits of the device's serial number	Set the least -significant 16 bits of the device's serial number
--------------------------------------	--------------------------------------	---	---	--	--

* Register value can be stored persistently to the EEPROM with by writing to “Store to EEPROM” register.

** To have an effect, register value must be stored to the EEPROM, wait for ~1 second and perform a power reset.

*** Register values must be written at once only with function code 0x10 (Write multiple registers). For example, write_multiple_registers(0x3006, [0xAAAA, 0BBBB, 0CCCC]). In case register value does not match the number size of parameter represented by the register (for example, write_multiple_registers(0x3006, [0xAAAA, 0BBBB])), a ILLEGAL_DATA_ADDRESS Modbus exception is returned.

Table 1. PGVA status word

PGVA status word (16 bits)	
Bits	Meaning
16:9	Reserved
8	0: No EEPROM write pending 1: EEPROM write pending
7	0: No overtemperature 1: Overtemperature
6:5	0: Supply voltage is nominal 1: Supply voltage low 2: Supply voltage high
4	0: Vacuum in the vacuum tank is nominal 1: Vacuum in the vacuum tank is above than the higher threshold
3	0: Pressure in the pressure tank is nominal 1: Pressure in the pressure tank is below the lower threshold
2:1	0: Pump is off 1: Pump is building up pressure 2: Pump is building up vacuum 3: Reserved
0	0: Idle 1: Busy

Table 2. Modbus command error

Modbus command error	
Error number	Meaning
0	No error
1	OutputActuationTime is out of range
2	PressureThreshold is out of range
3	VacuumThreshold is out of range
4	OutputPressureSet is out of range
5	ModbusUnitId is out of range
6	IP address does not comply with the restrictions
7	ManualTrigger is invalid
8	Incorrect number of registers written with function code 0x10
9	Register is cannot be written to

2.4 Modbus Communication

2.5 PLC Settings

The following example is based on Codesys.

2.4.1 basic settings

It's important to change the Unit-ID and the Port.

2.4.2 I/O Mapping and Channels

Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length	Comment
0 Holding	Read Holding Registers (Function Code 03)	Cyclic, t#100ms	16#1000	4	Keep last Value			
1 input	Read Input Registers (Function Code 04)	Cyclic, t#100ms	16#0100	4	Keep last Value			
2 write	Write Single Register (Function Code 06)	Cyclic, t#100ms				16#1003	1	

To get the system running it's important to set up an holding, input, and write register.

Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
		Holding	%IW24	ARRAY [0..3] OF WORD			Read Holding Registers
		input	%IW28	ARRAY [0..3] OF WORD			Read Input Registers
		write	%QW14	ARRAY [0..0] OF WORD			Write Single Register

To get live data from the PGVA box it is important to enabling the Modbus each cycle.