GP4I-4O General Purpose Input Output ModBus Manual

1. INTRODUCTION TO MODBUS AND PRODUCT

Thank you very much for purchasing our GP4I-4O with MODBUS-RTU features. This manual is intended for professional installer, if you are not, please consult to your official distributor. From now on when speaking about the GPIO we are referring to the GP4I-4O.

MODBUS is an open field bus successfully used through the world to connect field devices to a main controller. This is the reason why MODBUS has been our choice to offer to our customers and partners an automated solution easy to integrate not only with our brand products but also with a vast collection of third party components and controllers

MODBUS, MODBUS-RTU and other related names are registered trademarks of MODBUS Organization. Further information and documentation can be found at http://www.Modbus.org/

1.1 PRINCIPLE OF OPERATION

The GPIO implements MODBUS-RTU as a control-communications feature that allows its operation and supervision tasks from a MODBUS automation environment. Preventive maintenance and fault analysis is also possible thanks to the implementation of internal registers in the GPIO with the more relevant operational and error events.

Using a MODBUS-RTU message, the GPIO can manage output relays, read digital discrete inputs, report errors, historical data and so on, giving to the user/installer a wide range of new features based in the automation.

1.2. BASIC CHARACTERISTICS

The MODBUS communication system provides a Master/Slave implementation among devices sharing a physical connection. For the GPIO, the physical connection is a RS485 half duplex serial layer, which has been chosen among other options due to its wide implementation and roughness.

For the GPIO, a RS-485 half duplex wired connection has been implemented and the projector is designed to run in a single-master system. In this implementation, Master and Slave figures has a clear role that is crucial to clear understand for a proper system implementation.

Master Device: Device that controls the data exchange in the bus and, if necessary, implements co-ordination tasks among different slaves (i.e. PLC Programmable Logic Controller, SCADA, etc).

Slave Device: Devices connected to the bus that attends to the requests from the master, either reporting information or executing tasks as per Master request.

2. FRONT PANEL INDICATOR

The GPIO module´s panel indicator has three LEDS and one switch, the purpose of these elements is to interact and see the appliance condition, the Image 1 shows the panel indicator.

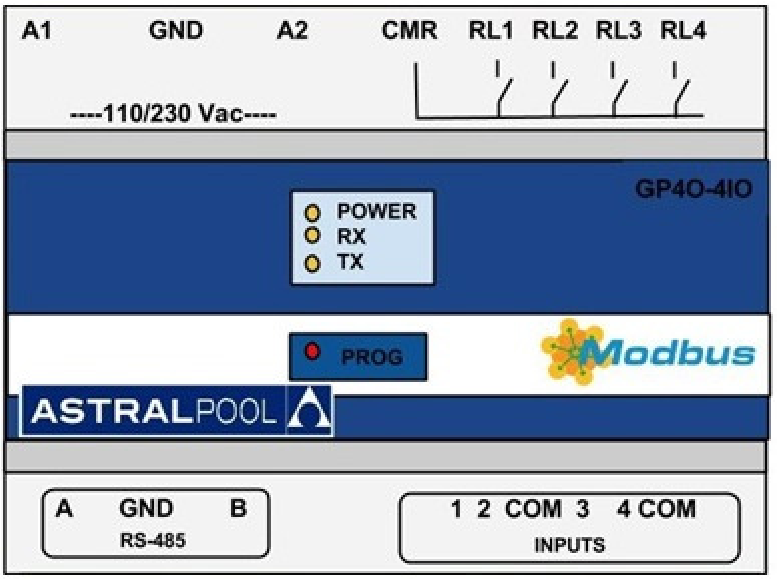

Image 1 Panel indicator

2.1. Power, rx and tx leds

The operation of these tree elements is as follows:

- POWER LED is ON when the module is powered, otherwise will be OFF.
- The RX led will be blinking when the GPIO module has receiving data.
- The TX led will be blinking when the GPIO module has sending data.

Rx and Tx are, respectively, the abbreviations of “receiving” and “transmitting”. Whenever the slave detects a frame from the master the Rx led will blink and if the slave respond to this request the Tx will blink.

2.2. Prog switch

The Prog button is intended to reset the device manually to factory configuration and it is located at the bottom of the leds, the procedure of resetting is explained in the point 7.3.3 Factory configuration through PROG switch

3. Connections

3.1. Electrical connections

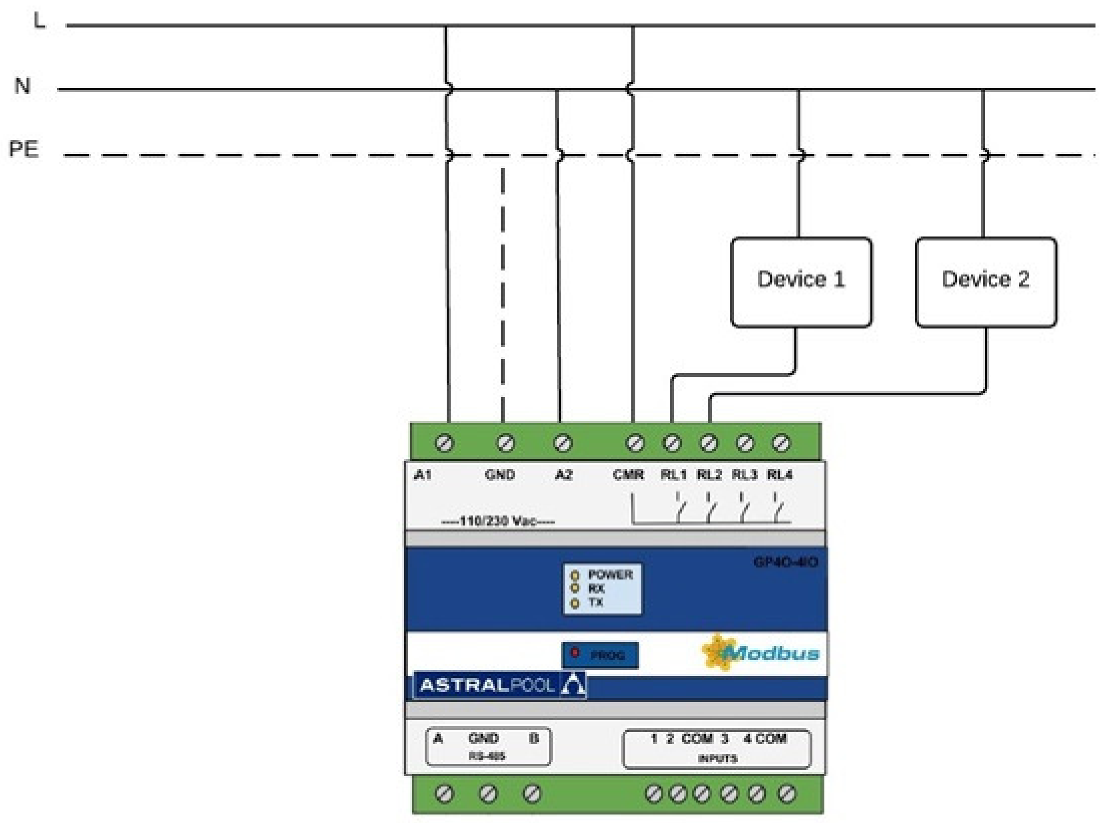

Image 2 Electrical Connection

The GPIO relays can allow or prevent the four devices intensity at the same time. But some precautions need to be done in order to guaranty the integrity of the GPIO.

1. All the outputs must be fed at the same voltage (CMR terminal in Image 2), never mix devices at different voltage supplies, this error may cause a potential difference between various outputs and can oversupply or undersupply them, provoking irremediable failures regardless of the device´s internal protection. The output nominal voltage is Vn= 250 V.
2. Likewise, the addition of the relay intensities must not be greater than the nominal intensity (In=5 A). Otherwise the relays will crash.
3. When the GPIO controls various appliances, the intensity must be cut by the same phase side **in all the devices.** If not, phase and neutral will meet **inside the GPIO**, causing serious damage to the relays.

Note: Taking this three tips and integrate it to your working methodology as soon as is required in order to install a GPIO, the Image 2 shows an example of two devices controlled.

3.2. Bus Connections

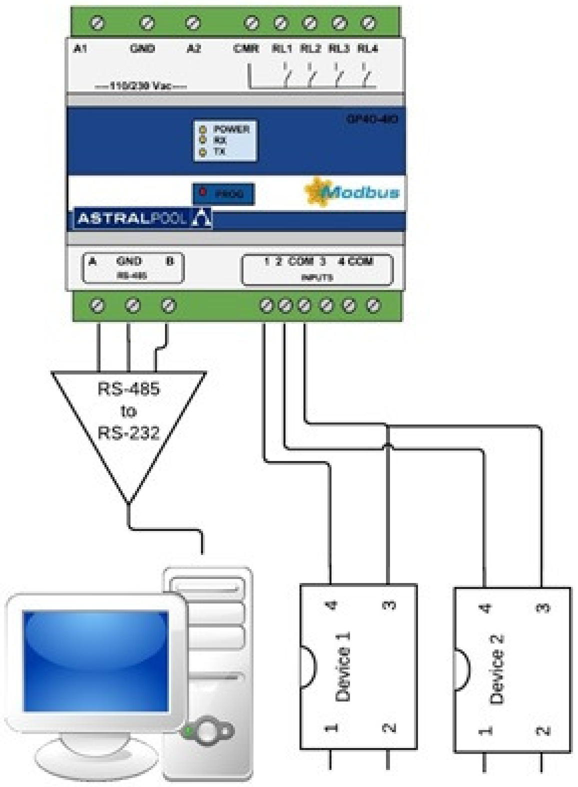

Image 3 Bus Connection

3.2.1 Dry contact input connections

The GPIO have four digital discrete inputs implemented, which doesn´t have voltage associated (dry contacts) and can detect digital pulses. These pulses can be read and his falling edges stored and also implement a digital filter in order to guarantee a correct measure. If some signals with voltage (wet contacts) want to be read, it can be used a **opto-isolator**. The point 8 Digital_inputs explain all these features..

3.2.2 RS-485 Standard and digital _inputs connections

On the other hand the RS-485 provides ANSI/TIA/EIA-485 standard communications.
Some manufacturers shifts the "A" and "B" connections and this may cause confusion but it is not the GPIO´s case. Mind this aspect when connecting the GPIO to MODBUS appliances of other companies.

4. Cable characteristics

4.1. Electrical cable characteristics

Since the GPIO supplies with the mains electricity it doesn´t matter if line (L, black, brown or grey) goes to the A1 and the neutral (N, blue) to the A2 or vice versa. The protective earth/ground (PE, yellow and green) should connect in the GND terminal
This appliance can be supplied with any main electricity, check the manual brochure BRON_GPIO_62368_v1.0_EN for more technical details.

4.2. RS-Standard and digital inputs

The recommended wiring for a MODBUS-RTU communication is based in a linear structure, active bus with termination at both ends. It is possible coupling and uncoupling devices during operation without affecting other devices. The wire shall be twisted and shielded according to EN 50 170.
The values of transmission rate supported for the device, allow maximum cable length of 1,200 m without repeaters, or up to 10 km using repeaters, when installation is according to the standard.
For the balanced pairs used in an RS485-system, a Characteristic Impedance with a value higher than 100 Ohms may be preferred, especially for 19200 and higher baud rates.

5. BUS ISOLATION AND TERMINATION RESISTORS

If the communication bus is accessible for the user, it shall be double insulated. As far as in general the accessibility of the bus to users will depend on each single installation, safety isolation has been implemented in the GPIO physical bus layer. Moreover, for safety purposes, it is recommended to ensure that other devices sharing this bus also implements this insulation. Additionally, the use of bus insulated devices not only enhances the security level, furthermore increases the equipment reliability, larger immunity to electromagnetic interference, longer life, higher reliability, more stability over the range of temperatures. Whenever single or multiple devices are connected sharing a bus physical connection, it is recommended to use terminating resistors at the ends of the bus, even more when use large cable length or high speed data rates. The terminating resistor, is used to prevent an RF signal from being reflected back from the end, causing interference. The terminating resistor must be in both ends of the bus, connected in parallel (as shown in the image below). A typical value of this resistance is 120?, 0.5W. The value of the resistor must be the same in both ends. The terminating resistors are the resistors R_T of the Image 2. Interconnection of devices.

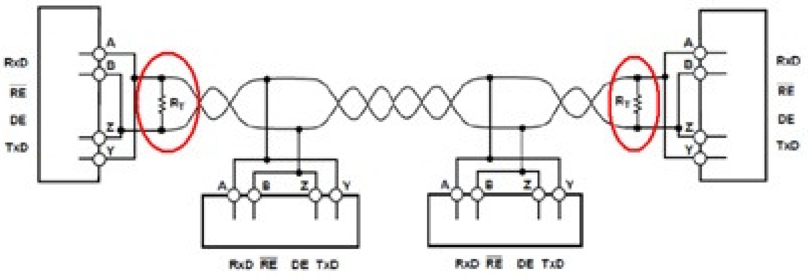

Image 4. Interconnection of devices

6. MODBUS FUNCTIONS

6.1. FUNCTIONS SUPPORTED

Please, be careful at the possible actuations, and make sure that the function used is the correct.

BIT ACCESS MODE

Functions in bit access mode are implemented according to the MODBUS-RTU standard described in http://www.Modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

0x01 READ COILS.
0x0F WRITE MULTIPLE COILS.
0x05 WRITE SINGLE COIL.
0x02 READ DISCRETE INPUTS.

REGISTER ACCESS MODE

Functions in register access mode are implemented according to the MODBUS-RTU standard described in http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

In general registers are unsigned 16 bit coded.

0x03 READ HOLDING REGISTERS
0x04 READ INPUT REGISTERS
0x10 WRITE MULTIPLE REGISTERS
0x06 WRITE HOLDING REGISTER

6.2. EXCEPTION RESPONSES

Exception responses are implemented according to the MODBUS-RTU standard described in the chapter MODBUS exception responses:
http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf
The exceptions implemented are from 1 to 4.
The exception 4 appears either when a frame tries to write power_counter or wtd_counter, described in the point 11 Basic ModBus RTU Register Map. The other exception codes (one to three) normally appear.

7. DEVICE DESCRIPTION AND CONFIGURATION

7.1. GENERAL DESCRIPTION

Through request in MODBUS, is possible to manipulate the four relays of the GPIO and the actual state of his four digital inputs can be read.
When a simultaneous request is made through Modbus and remote control, and there is no conflict, the last request will take effect.
In general, there is not check on the constancy of the values sent to specific registers. Therefore is the operator responsibility to check that consistency.
In this manual, the numbers in hexadecimal have been represented with the format **0xZZ**, where ZZ is the number.
The input register that governs the State Machine Diagram are in the point 7.2, and all memory locations quoted are in the point 11 Basic ModBus RTU Register Map.

7.2. STATE MACHINE DIAGRAM

When the system Powers ON, the state is always Start. If the watchdogs doesn´t triggers and no request is made since powers ON, the system will remain in start State.
From Start is possible to go to Request state (if some request is made) or the Satchdog state (if it is triggered). From the request state we can also go to the request state again by doing another request. And when the watchdog state is trigged it only can be change the state by sending a request, or if watchdog setting is suitable the start state will be achieved (ilustrated in with a dotted arrow). The Image 5 shows this flow.

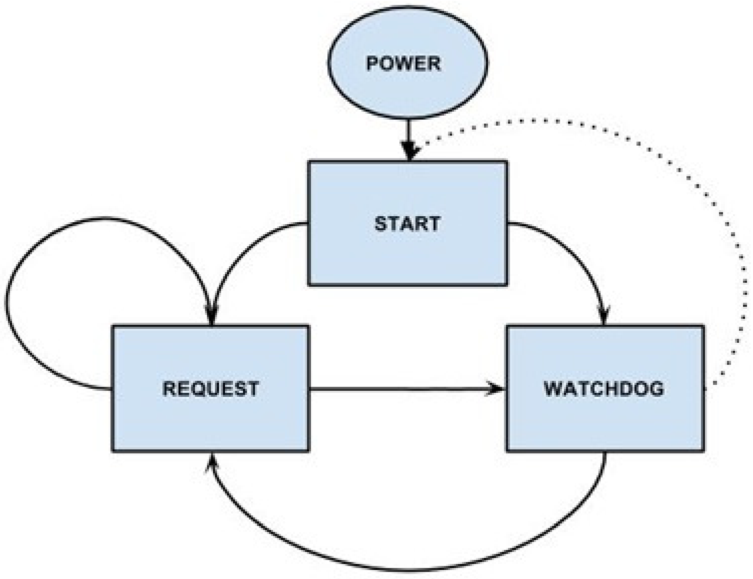

Image 5. State Diagram

7.3. Setting the GPIO

7.3.1 Addressing setting

The address of the GPIO in the bus is set through the 0x00 holding register.

ID_Address:	Address of the GPIO in the bus.
Factory setting:	0x1A.
Suggested range:	0x1A - 0x1E.

The factory default for the GPIO is (0x1A). However if more than one GPIO wants to be implemented both ID_Address values must not match.
This situation can be avoided by writing in holding register 0x00. The following example shows how to do such procedure.
Example: changing the ID_Address from 0x1A (default) to 0x1B.
Transmit Message: 1A 10 00 00 00 01 02 00 1B 58 AB
Where:

1A	is the slave address. (the actual ID_Address).
10	is the function used. Write multiple registers.
00 00	is the address of the holding register to be written.
00 01	is the number of registers to be written (1 in this case).
02	is the number of bytes of data to be sent.
00 1B	is the new ID_Address.
58 AB	is the CRC.

7.3.2. Communications setup

The baud rate selection of the serial communications is set through the 0x01 holding register. By default 9600 bps and 8E1 (8 data bits, even parity, 1 stop bit). However, other settings are also supported, showed in the next table GPIO supported communication settings.
The reason for supporting N2 frames is to keep the MODBUS standard requirement of sending eleven bits per byte (1 start + 8 data + 1 parity + 1 stop). Whenever a no parity configuration is chosen, then 2 stop bits are introduced to keep the eleven bits per byte required by the standard.
For compatibility reasons, N1 frames are also supported. However, mind that using this selection you are not fulfilling the MODBUS standard requirements as far as only ten bits per byte are used.
According to this, the baud rate and frame selection is completed defining the baud rate (in bauds), number of data bits, parity and number of stop bits.

COM_Setup:	Communication setup	
Factory setting:	0	9600, 8E1
Supported values:	0	9600, 8E1
	1	19200, 8E1
	2	9600, 8N2
	3	19200, 8N2
	4	9600, 8N1
	5	19200 8N1

7.3.3 Factory configuration through prog switch

The following example serves to illustrate the procedure to reset the GPIO in order to retrieve the original ID_Address. Be aware that the DI_x_counter (where “x” is a number between 1 and 4) and the latched alarm registers will be erased in the process (these registers are explained in points 9.1 and 10.1.4 respectively).
In order to reset to factory configuration the steps will be as follows.

1. Unplug the device from the mains electricity.
2. Hold the PROG button.
3. While the PROG button is pushed, plug the GPIO again.

The GPIO will return to “start” state and will have the predefined ID_Address in the holding register 0x00 (0x1A).
The DI_x_counter (where “x” is a number between 1 and 4) and the latched alarm will be set to 0, so please be careful.

7.4. BROADCASTING

Broadcasting is not supported by the GPIO.

7.5. WATCHDOG

The watchdog is a timer implemented in the GPIO aimed to check if the communications in the bus keeps alive.
According to these descriptions, the time considered for triggering the watchdog and the task performed in this case needs to be defined.
When the GPIO lost communication for a time greater than the watchdog_time, the most significant bit located in the latched alarms register in holding register 0x20 and also located in the nstantaneous alarms in input register 0x01 will be at 1.
Note: If the watchdog_time is less than 30 seconds the first time the GPIO turns ON it triggers at 30 seconds for security reasons.

7.5.1. WATCHDOG Time

The watchdog triggering time is defined in the holding register 0x10. This time is set in seconds. 0 means watchdog is disabled and this is the default value. Every time the watchdog is triggered his WDT_counter increases, explained in point 9.3.
To enable the watchdog feature, set the watchdog_time to a value different from 0.
Example: Set watchdog triggering time to 30s:
Transmit Message: 1A 10 00 10 00 01 02 00 1E 9A 38
Where:

1A	is the slave address.
10	is the function used. Write Multiple Registers.
00 10	is the address of the holding register to be written.
00 01	is the number of registers to be written, 1 in this case.
02	is the number of bytes to be sent
00 1E	is the value to be sent, 30 in decimal.
9A 38	is the CRC.

Now, the watchdog_time is set to 30s. Therefore, whenever two properly constructed messages are read in less than 30s, even not addressed to the GPIO, the watchdog is not triggered. Otherwise, it is triggered.
To know the watchdog_time, is necessary read the holding registers.
Message received: 1A 03 02 00 1E 5C 4E

1A	is the slave address.
03	is the function used. Read holding registers.
02	is the number of bytes of data to be read.
00 1E	is the time configured (30 in decimal).
5C 4E	is the CRC.

7.5.2. WATCHDOG config

Located in holding register 0x11, it sets what state will occur when the watchdog triggers.
The High Byte:
IF IS 0, THEN THE WATCHDOG TRIGGERS IT FORCES THE GPIO TO ENTER IN THE WATCHDOG STATE.
IF IS DIFFERENT FROM 0, WHEN THE WATCHDOG IS TRIGGERED, THE DEVICE WILL RESET AND RETURN TO THE START STATE.
The Low Byte: is not used in the GPIO.
Example: Configuring the watchdog to enter into the watchdog state when it triggers.
Transmit Message: 1A 10 00 11 00 01 02 00 00 1B E1
Where:

1A	is the slave address.
10	is the function used. Write multiple registers.
00 11	is the address of the first register to be written.
00 01	is the quantity of registers to be written, 1 in this case.
02	is the number of bytes to be sent.
00 00	**High Byte**. the watchdog will enter into the watchdog state when it triggers **Low Byte**. has no use.
1B E1	is the CRC.

7.5.3. WDT_Relay_state

The watchdog predefined relay state, in holding register 0x14, determines whether the four relays are opened or closed (respectively) when the watchdog triggers.
The four relays will open or close according to the binary value sended to this register, from the less significant bit to the fourth bit relays in crescent order.
Example: set WDT_relay_state as it follows: the first and the third relay ON and the others will be turned OFF (RL4=0; RL3=1; RL2=0; RL1=1).
Transmit Message: 1A 10 00 14 00 01 02 00 05 DB B7
Where:

1A	is the slave address.
10	is the function used. Write Multiple Registers.

00 14	is the address of the first register to be written.
00 01	is the number is bytes written.
02	is the number of bytes to be sent.
00 05	is the binary number equivalent to open only the first and the third relay.
DB B7	is the CRC.

NOTE: Mind that watchdog function is implemented for safety reasons. However, the reasons for activating it, or not, depends on the installer/integrator criteria. Always, mind of implications on what is being implemented. Use this feature under your responsibility.
Watchdog_time, Watchdog_config and WDT_relay_state should be configured in conjunction.

8. Digital Inputs

Located in the Input Register 0x20 can acquire the value of a discrete signal such as a dry contact, or wet contacts optocoupled. These falling edges can be filtered and stored (explained in point 9.1 Digital_input Counters and 8.1 Digital Input Filter Config respectively). The Image 6 shows an example of use to read the pulses of 2 devices.

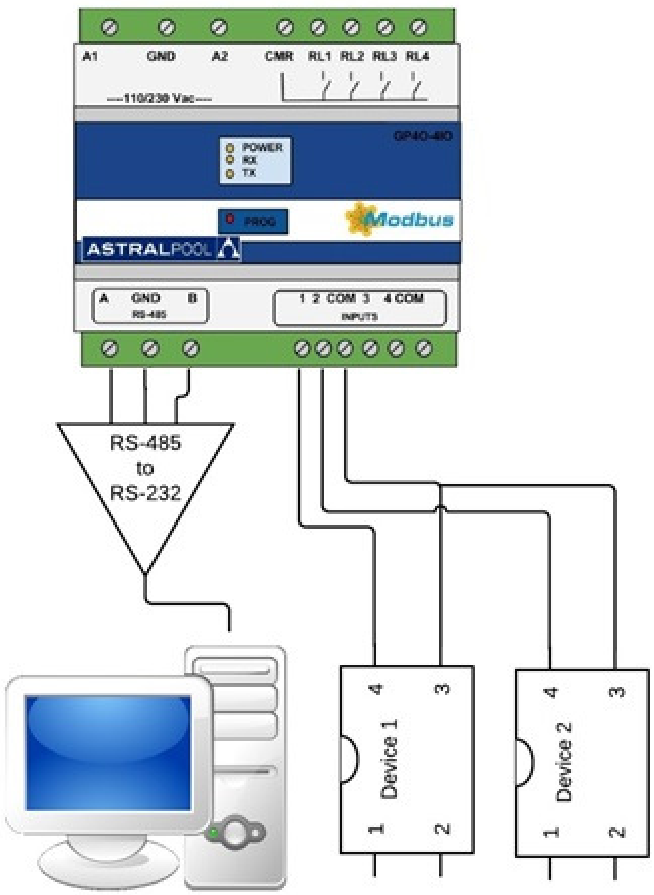

Image 6 Digital Inputs in the GPIO

8.1. Digital inputs filter config

Some signals may have spurious behaviors and give different problems or non consistent measures. Those connections may have random peaks produced by physical rebounds and the counters may store more than one increase per falling edge, illustrated at the top diagram of the Image 7.
The intention of this filter is to avoid these extra falling edges and achieve one increase per pulse, illustrated at the bottom of the Image 7.

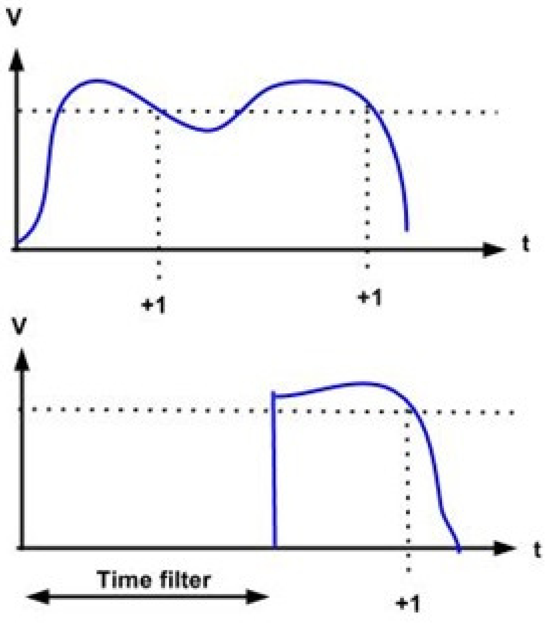

Image 7 Spurious signal with and without digiital filter

This issue can be solved by setting digInputFilterConfig (located at 0x12 holding register) defining a time in which the input ignores the signal, after this time the signal will be read correctly.

This time should be greater when the signal has more peaks and lower or zero when the signal has a few or no peaks. The filter parameters will be:

0x12 **High Byte** Time needed to avoid bad measures.

0 Without Filter
1. Time filter 1ms.
2 Time filter 10ms
3 Time filter 100ms
4 Time filter 500ms

0x12 **Low Byte** Input or inputs to be filtered.

Example: setting the inputs 1, 3 and 4 with a 500ms filter.

Transmit Message: 1A 10 00 12 00 01 02 04 0D D8 D7

Where

1A	is the slave address.
10	is the function used. Write Multiple Registers.
00 12	is the address of the first register to be written.
00 01	is the quantity of registers written.
02	is the number of bytes to be written.
04 0D	

High Byte is the filter Config 0000 0100 related to 500ms.
Low Byte is the filter Config 0000 1101 related to inputs 1, 3 and 4.

D8 D7 is the CRC.

Received Message: 1A 10 00 12 00 01 A2 27

8.2. Reading Wet contacts

Wet contacts have a power source included with the equipment; if a contact like this wants to be read a **opto isolator** needs to be used in order to guarantee the isolation between the inputs and the wet contacts, the contacts must be as follows in the Image 9.

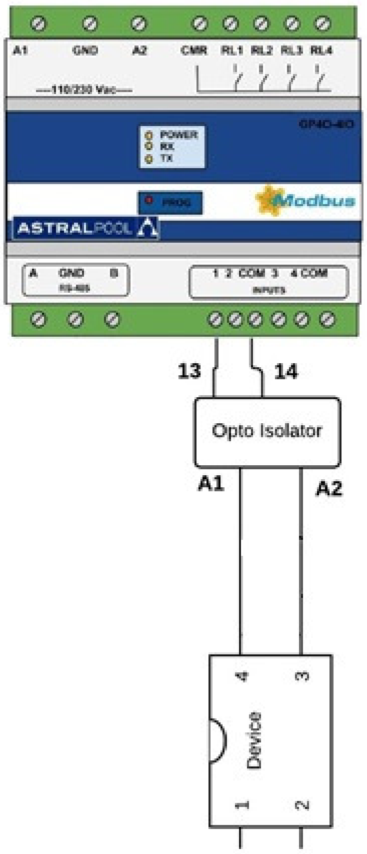

Image 8 Reading wet contacts through GPIO inputs

If more info about this type of connections is needed, please refer to any optoisolator datasheet.

9. Counters

The GPIO boasts three counters, each one are aimed to count a different aspect related to the GPIO behavior.

9.1 Digital Input Counters

These counters are located in the input registers (from 0x03 to 0x06) and only increase whenever the input detects a falling edge (from 1 to 0). They are named DI_x_counter (where "x" is a number between 1 and 4). From the firmware version 120 forward, these registers are 32 bits format instead of 16 bits.
Remember that if the GPIO is reprogrammed like we explained in the point 7.3.3 Factory configuration through PROG switch these four input registers will be reseted to 0.
Example: read digital input counters as input registers. Values resulting after resetting the device to factory configuration.
Transmit: 1A 04 00 03 00 04 02 22

1A	is the slave address.
04	is the function used. Read Input Registers.
00 03	is the address of the holding register to be read.
00 04	is the number of bytes to be read.
02 22	is the CRC

Received message: 1A 04 08 00 00 00 00 00 00 00 00 55 E9

9.2 Power counter

Located in holding register 0x30 is incremented every time the GPIO is powered.
Example: checking the power_counter value before and after device to factory defaults.
Transmit Message (before resetting): 1A 03 00 30 00 01 87 EE
Where

1A	is the slave address.
03	is the function used. Read Holding Registers.
00 30	is the address of the holding register to be read.
00 01	is the number of bytes to be read.
87 EE	is the CRC

Received Message: 1A 03 02 **00 3F** 9C 56
Transmit Message (after resetting): 1A 03 00 30 00 01 87 EE
Received Message: 1A 03 02 **00 40** DD B6
Where:

1A	is the slave address.
03	is the function used. Write multiple registers.
02	are the bytes received.
00 40	is the number of bytes to be read.
DD B6	is the CRC

9.3 WDT counter

Located in the holding register 0x31, will be increased every time the watchdog is triggered
Example: checking the WDT_counter before and after the watchdog triggers.
Transmit Message (before watchdog triggers): 1A 03 00 31 00 01 D6 2E
Received Message: 1A 03 02 **44 D3** AE DB
Where

1A	is the slave address
03	is the function used, read holding registers.
02	are the bytes received.
44 D3	is the WDT_counter value.
AE DB	is the CRC

The amount of times that the GPIO have entered watchdog State is 44D3 in hexadecimal.
Transmit Message (after watchdog triggers): 1A 03 00 31 00 01 D6 2E
Received Message: 1A 03 02 **44 D4** EF 19
Where

1A	is the slave address
03	is the function used, read holding registers.
02	are the bytes received.
44 D4	is the WDT_counter value.
EF 19	is the CRC

10 Operation Modes

10.1. Basic mode

In this section it is assumed that a successful connection has been established with the GPIO and therefore, address, communication settings and other configurations has been already set.

10.1.1. Start predefined relay state
Is the relay´s disposition in the GPIO when it Powers ON (located in the input register 0x15 and coils 150 to 153) the Image 9 shows the disposition of these holding registers and coils in the appliance.

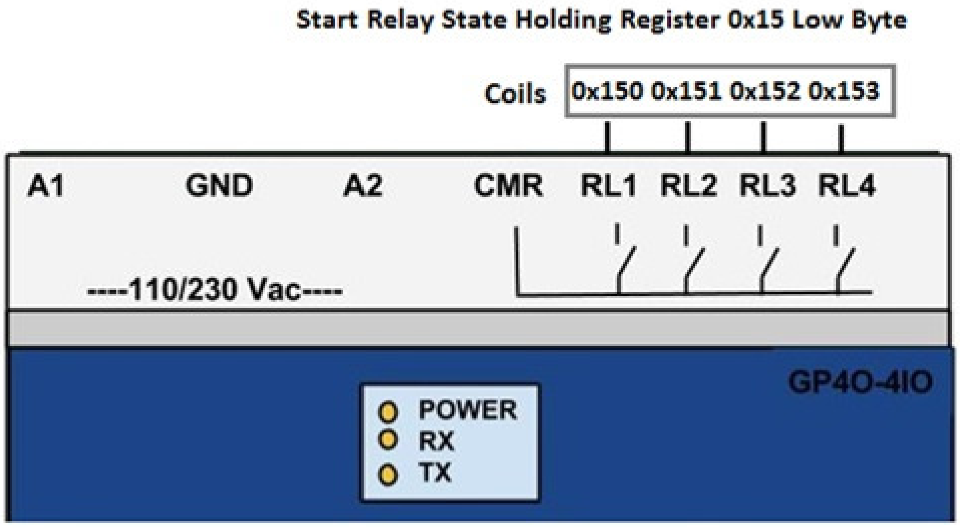

Image 9 Startt_relay_state

Example: setting the start_relay_state with all values to 0.
Transmit Message: 1A 10 00 15 00 01 02 00 00 1A 65
Where:

1A	is the slave address (the actual ID_Address).
10	is the function used, write multiple registers.
00 15	is the address of the holding register to be written.
00 01	is the number of registers to be written.
02	is the number of bytes of data to be sent.
00 00	**High byte:** have no use. **Low byte:** the four less significant bits refer to the four relays in crescent order
1A 65	is the CRC.

Received Message: 1A 10 00 15 00 01 13 E6

10.1.2. Checking current state
The state in which the GPIO is currently set is available through status input register 0x00. This register has a different meaning for the **Low byte** and for the **High byte**.

The less significant bit is used to show if an error has occurred and is set to 1 whenever an error exist.
Detailed information of the error/errors detected can be requested to the alarms input register 0x01 or the latched alarms holding register0x20 (explained with detail in points 10.1.5 Checking Instantaneous Alarm Register and 10.1.4 Checking Latched Alarms Register respectively).
Only the watchdog alarm has been implemented, which means that if an error exist, is because the watchdog is triggered).
The codes for the different states implemented in the high byte are shown in the following table:

High byte	Low Byte
00 start.	00 no error
01 watchdog.	01 error
02 request.	

Coding Examples (High byte first):

0x000	start state and no error.
0x001	start state and error due to watchdog.
0x100	request state and no error.
0x201	watchdog state and error due to watchdog.

Example: Reading the current state in the GPIO.
Transmit Message: 1A 04 00 00 00 01 32 21
Where:

1A is the slave address.
04 is the function used. Read input registers.
00 00 is the address of the first input register to be read.
00 01 is the number of input registers to be read
32 21 is the CRC

Received Message: 1A 04 02 **02 01** 1D 92, which indicates watchdog state with watchdog alarm.
Where:

1A	is the slave address.
04	is the function used. Read input registers.
02	is the number of bytes to be read.
02 01	

High byte: GPIO is in watchdog state.
Low byte: GPIO has some alarm (watchdog).
1D 92 is the CRC.

10.1.3. Relays change request

Relays change request can be sent to the GPIO through control_w0 (located in holding register 0x21).
However, these requests can also be sent using a bit address mode starting on coil 0x210. Both ways to change the relays should not be mixed to avoid misunderstandings, the Image 10 shows the disposition of these holding registers and coils in the appliance.

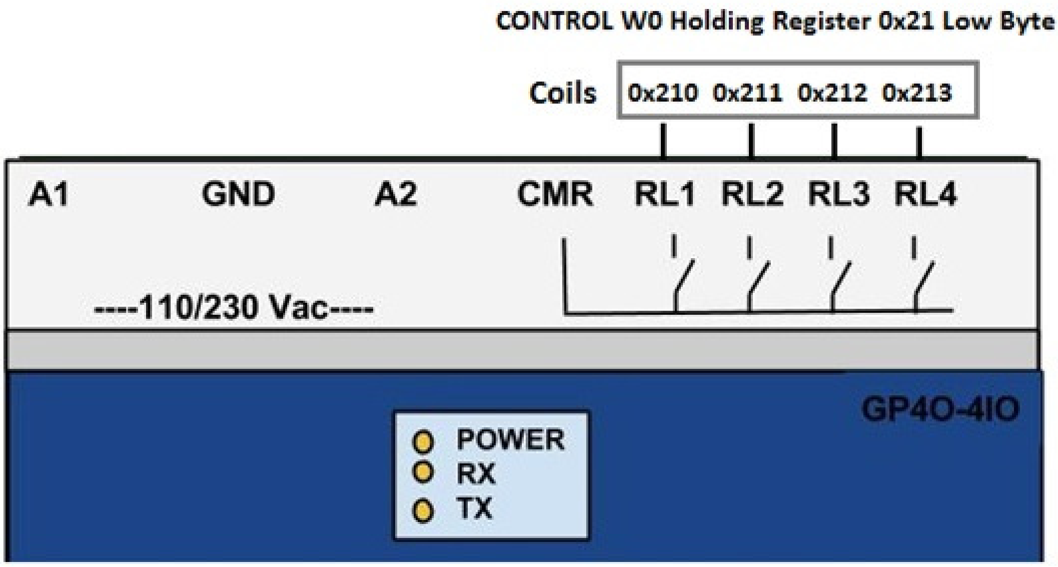

Image 10 Table location of the output relays

Example: activating relays 1 and 3
Transmit Message: 1A 10 00 21 00 01 02 00 05 DE 12

Where:

1A	is the slave address.
10	is the function used. Write multiple registers.
00 21	is the address of the holding register to be written.
00 01	is the number of registers to be written.

02 is the number of bytes of data to be sent.
00 05
High byte: have no use.
Low byte: is the binary number equivalent to open only the first and the third relay 1A 65 is the CRC.

Received Message: 1A 10 00 21 00 01 52 28

10.1.4. Checking latched alarms register

It is possible to check which alarm has been activated since last alimentation or last alarm reset. Only the watchdog alarm has been implemented.
To check it, holding register 0x20 must be read, from the bit 0 to the bit 14 are not used, but can be read, and the bit 15 is the watchdog alarm, it also can be read as coils (located from the coil 0x200 to 0x20F)
The value 1 indicates that the alarm has been activated and 0 means deactivated.
If the alimentation fails, all the alarms will be at 0.These alarm values can be forced too.
Example: reading latched alarms with holding registers when watchdog alarm has been activated
Transmit Message: 1A 03 00 20 00 01 86 2B (read holding register 0x20).
Received message: 1A 03 02 80 00 BD 86

Where:

1A is the slave address.
03 is the function used. Read holding registers.
02 is the number of bytes of data to be read.
80 00
High Byte: The most significant bit indicates watchdog error
Low byte: has no use.
BD 86 is the CRC.

Example: reading latched alarms with coils when watchdog alarm has been activated:

Transmitted message: 1A 01 02 0F 00 01 CF 9A (read coil 0x20F).

Received message is: 1A 01 01 01 96 AC

Where:

1A is the slave address.
01 is the function used. Read Coils.
01 is the number of coils to be read.
01 indicates that the alarm has been activated.
96 AC is the CRC.

Example: reset all the alarms by writing a 00 00 in the holding register 0x20.
Transmitting message: 1A 10 00 20 00 01 02 00 00 1F C0

Where:

1A is the slave address.
10 is the function used. Write multiple registers.
00 20 is the address of the first holding register to be set.
00 01 is the number of holding register to be set, 1 in this case.
02 is the number of bytes of data sent.
00 00 is the data send (0 to disable the alarm).
1F C0 is the CRC.

Received Message: 1A 10 00 20 00 01 03 E8

10.1.5. CHecking instantaneous alarms register

Whenever errors exist, the less significant bit in the status input register 0x00 (explained in point 10.1.2 Checking Current State) and the most significant bit of the instantaneous alarm Digital Input 0x01 is set to 1.
Example: Checking the instantaneous alarm register.
Transmit message: 1A 04 00 01 00 01 63 E1
Where
1A is the slave address.
04 is the function used. Read input registers.
00 01 is the address of the first input register to be read.
00 01 is the number of input register to be set.
63 E1 is the CRC.
Received message: 1A 04 02 80 00 BC F2
Where:

```
1A       is the slave address.
04       is the function used. Read input registers.
02       is the quantity of bytes to be read.
80 00    is the value of the input registers read, only the most significant bit is set to 1.
BC F2    is the CRC.
```

Location of this input register 0x01 error register is shown in 11 Basic ModBus RTU Register Map, the bits 0 to 14 are not used, but it can be read without limitation, while bit 15 shows the watchdog MODBUS error.

11. BASIC MODBUS-RTU REGISTER MAP

The table shown in this chapter is our exclusive and original register map with the name of the function and their address. Furthermore to the register map itself, there is a direct relationship between holding register 0x21 and input register 0x00. While the first one stands for a request action, the second one states for the current action in progress.
There is also a direct relationship between latched alarms holding register 0x20 and instantaneous alarms input register 0x01. While holding register 0x20 is the total alarms latched, the input register 0x01 is the current alarm. This relationship also applies to the bit-to-bit relation between registers. Only the watchdog alarm has been implemented, which means that the latched alarms and instantaneous alarms, always match.
To reset the alarm errors, it is necessary to reset it from the holding Register 0x20 and not from Input Register 0x01, due to, the Input Register 0x01 will reset when the current error alarm disappears. To reset all the alarms, it is necessary to set to 0 from Coil 0x200 to 0x20F.
Note: a disconnection of the power supply will reset all the latched alarms.

Name	**Holding Registers**	**Input registers**	**Coils**	**Inputs**	**Description**
ID_Address	0x00				This parameter is the address of the GPIO, by default it is set to 0x1A. If the installation has more than one device connected in the bus, it is necessary to change de ID_Address of the device before connecting another one.
COM_Setup	0x01				This parameter determinates the Baud rate selection. There are 6 possible configurations to choose: Factory settings: 0. 9600, 8E1 Supported values: 1. 19200, 8E1 2. 9600, 8N2 3. 19200, 8N2 4. 9600, 8N1 5. 19200, 8N1
ID_Manufacturer_hi	0x02				This parameter indicates the high byte that represents the Manufacturer code
ID_Manufacturer_lo	0x03				This parameter indicates the low byte that represents the Manufacturer code
ID_Product_code_hi	0x04				This parameter indicates the high byte that represents the Product code.
ID_Product_code_lo	0x05				This parameter indicates the low byte that represents the Product code.
Reserved	0x06				Reserved.
HW_Version	0x07				This parameter indicates the Hardware Version of the GPIO.
SW_Version	0x08				This parameter indicates the Software Version of the GPIO.
MODEL_Serie_hi	0x09				This parameter indicates the high byte of the Serial Number.
MODEL_Serie_lo	0x0A				This parameter indicates the low byte of the Serial Number.
MODEL_Production_hi	0x0B				This parameter indicates the high byte of the production batch.
MODEL_Production_low	0x0C				This parameter indicates the low byte of the production batch.
Watchdog_time	0x10				This parameter defines the time in seconds that can elapse between the last communication and the actuation of the watchdog. If it is set to 0, the Watchdog is disabled. Range: 0-65535
					This parameter defines how will actuate the watchdog if the communications fail.

Watchdog_config	0x11 low byte high byte				The Low Byte Is the state when the Watchdog triggers (not used). The High Byte If is 0 when Watchdog state achieves it goes to the Watchdog state. If is diferent from 0 when the Watchdog state achieves it reset the GPIO.
DigInputFilterConfig	0x12 bit 0…3 bit7…10				This parameter defines the filter time to ignore the rebounds and the inputs filtered. The Low Byte in binary the input or inputs filtered. The High Byte 0 No filter. 1 Filter with 1 millisecond. 2 Filter with 10 milliseconds. 3 Filter with 100 milliseconds. 4 Filter with 500 milliseconds.
wdt_relay_state	0x14 bit0...3		0x140 0x141 0x142 0x143		This register sets the relay disposition when the GPIO enters the Watchdog State. Low byte: In binary, the state desired for the four relays. RL1 RL2 RL3 RL4
Start Relay State	0x15 bit0...3		0x150 0x151 0x152 0x153		This register sets the relay disposition when the GPIO enters the Watchdog State. Low byte: In binary, 0 = Relay closed, 1 = relay opened. RL1 RL2 RL3 RL4
Latched Alarm Watchdog	0x20 .bit0…14 .bit 15		0x200 to 0x20E 0x20F		This parameter indicates which alarm has been activated since the GPIO was ON. Most significant bit: 1 watchdog triggered, 0 watchdog not triggered.
control_w0	0x21 .bit0…3 .bit 4..15		0x210 0x211 0x212 0x213 0x214 to 0x21F		Low byte: In binary. RL1. 0 = Relay closed, 1 = relay opened RL2. 0 = Relay closed, 1 = relay opened RL3. 0 = Relay closed, 1 = relay opened RL4. 0 = Relay closed, 1 = relay opened not used.
Counters power_counter WDT_counter DI_1_counter_hi DI_1_counter_lo DI_2_counter_hi DI_2_counter_lo DI_3_counter_hi DI_3_counter_lo DI_4_counter_hi DI_5_counter_lo	0x30 0x31	0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A			Counter increased when the GPIO lost electrical supply (writte disabled). Counter increased when the GPIO enters into the Watchdog state (writte disabled). This counters increase in "1" when the input experiments a falling edge. In SW < 120, DI_x_counter are unsigned 16 bits and they are mapped from IR 0x03 to IR 0x06. From SW 120 forward, DI_x_counter are signed 32 bits and they are mapped in high part and low part from IR 0x03 to IR 0x0A.
Status Alarm ON 0: Start State		0x00 low byte .bit 0 high byte		0x000	The Low Byte: Indicates the alarm, 1 = Watchdog alarm. 0 = no Watchdog alarm.

1: Request State 2: Watchdog state		.bit 8…11		0x008… 0x00B	The High Byte: Indicates the actual state.
Instantaneous Alarm Watchdog		0x01 .bit 1…14 .bit 15		0x01F	This parameter indicates the instantaneous alarms when the read is done. Most significant bit: 1 watchdog triggered, 0 watchdog not triggered.
Digital inputs		0x02 .bit0…3		0x020 0x021 0x022 0x023	This register defines the GPIO input that can read pulses. Input 1 Input 2 Input 3 Input 4

12. PRODUCT REVISION

Manual v.1.05 : All the information of this manual, describes the behavior of the Hardware Version 1.02, and Software Version 1.20.