
GP4I-4O General Purpose Input Output ModBus Manual

1.

INTRODUCCIÓN A MODBUS Y AL PRODUCTO

1.1. Principio de operación

1.2. Características básicas

2. PANEL FRONTAL

2.1. Leds, power, Rx y Tx

2.2. Botón PROG

3. CONEXIONES

3.1 Conexiones eléctricas

3.2 Conexiones BUS

3.2.1. Conexiones de contactos libres de tensión

3.2.2. Estandar RS-485 y conexiones de entradas digitales

4. CARACTERÍSTICAS DEL CABLEADO

4.1 Características del cableado eléctrico

4.2 Estandar Rs-485 y entradas digitales

5. AISLAMIENTO DEL BUS Y TERMINACIÓN RESISTIVA

6. DESCRIPCIÓN MODBUS

6.1 Funciones soportadas

6.2 Códigos de Excepción

7. DESCRIPCIÓN Y CONFIGURACIÓN DEL APARATO

7.1 Descripción General

7.2 Diagrama de estados

7.3 Configurando el GPIO

7.3.1. Configuración de dirección

7.3.2. Configuración de comunicaciones

7.3.3. Configuración de fábrica a través del boton PROG

7.4 Broadcasting

7.5 Watchdog

7.5.1. Watchdog_time

7.5.2. Watchdog_config

7.5.3. WDT_relay_state

8. ENTRADAS DIGITALES

8.1 Configuración del filtro de las entradas digitales

8.2 Lectura de contactos con tensión

9. COUNTERS

9.1 Digital_Input Counters

9.2 Power Counter

9.3 WTD_Counter

10. MODOS DE OPERACIÓN

10.1 MODO BÁSICO

10.1.1. Start relay state

10.1.2. Comprobando el estado actual

10.1.3. Petición de cambio en los relés

10.1.4. Comprobando el registro de alarmas latcheadas

10.1.5. Comprobando el registro de alarmas instantaneas

11. MAPA BÁSICO DE REGISTROS MODBUS RTU

12. REVISIÓN DEL PRODUCTO

1. INTRODUCCIÓN A MODBUS Y AL PRODUCTO

Muchas gracias por haber obtenido su GP4I-4O equipado con MODBUS-RTU. Este manual está enfocado para el instalador profesional, si usted no lo es, por favor consulte con su distribuidor oficial. De ahora en adelante cuando hablemos de GPIO nos estaremos refiriendo a GP4I-4º (General Purpose Input Output).

MODBUS es un campo de estudio muy amplio que actualmente conecta dispositivos del mundo a través de un maestro. Esta es la razón por la que MODBUS se ha convertido en nuestra opción para ofrecerles a nuestros

clientes soluciones automatizadas y sencillas para integrar no solo nuestros productos, sino también una vasta colección de otros componentes y controladores de terceras empresas.

MODBUS, MODBUS-RTU y otros nombres relacionados son marcas registradas de la Organización MODBUS. Se puede encontrar información pertinente y documentación en <http://www.Modbus.org/>

1.1 PRINCIPIO DE OPERACIÓN

El GPIO tiene implementado MODBUS-RTU como protocolo de comunicaciones que permite su operación y supervisión desde el entorno de automatización de MODBUS. EL mantenimiento preventivo y la asistencia a distintos fallos son también posibles gracias a la implementación de registros internos en el GPIO donde se almacenan errores y se controlan estados

Al enviar un mensaje por MODBUS-RTU, el GPIO es capaz de gestionar los relés de salida, leer entradas discretas digitales, histórico de errores y mucho más, dando al instalador/usuario un amplio abanico de nuevas opciones basadas en la automatización.

1.2. CARACTERÍSTICAS BÁSICAS

La comunicación MODBUS involucra una disposición de red maestro/esclavo entre los dispositivos compartiendo conexiones físicas. Para el GPIO, esta conexión física se realiza a través de conexión en serie RS-485 Half Duplex, la cuál ha sido escogida como la mejor opción entre otras debido a su amplitud de implementaciones y robustez.

Para el GPIO, se ha implementado una conexión RS-485 Half Duplex y el proyectista ha decidido implementarla en una red de un solo maestro. En esta implementación, Maestro y Esclavo son dos piezas clave para entender claramente el funcionamiento de la red.

El dispositivo maestro es el que controla el intercambio de información a través del bus de datos y, si es necesario, coordina tareas entre los diferentes esclavos (por ejemplo PLC Programmable Logic Controller, SCADA, etc).

El Esclavo es aquel dispositivo conectado al bus que atiende a las peticiones del maestro, ya sea mandando información al maestro o bien ejecutando tareas que este le ha solicitado.

2. PANEL FRONTAL

El módulo del panel frontal del GPIO tiene tres LEDs y un interruptor, el propósito de estos elementos es el de interactuar y observar el estado del aparato, La Imagen 1 muestra el panel central.

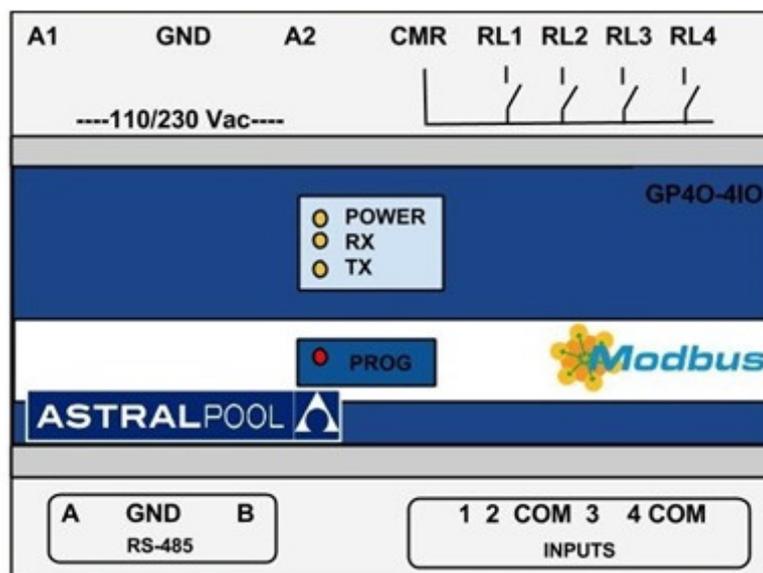


Imagen 1 Panel frontal

2.1. Leds Power, Rx y Tx

El funcionamiento de estos tres elementos es el siguiente:

- POWER, se enciende cuando el módulo está conectado, en caso contrario estará desconectado
- RX, parpadea cuando el GPIO está recibiendo datos.
- TX parpadeará cuando el GPIO esté enviando datos.

Rx y Tx son, respectivamente, las abreviaciones de "recibiendo" y "transmitiendo". Cuando el esclavo detecte una trama desde el maestro el led Rx parpadeará y si el esclavo responde a esa petición el led Tx también parpadeará.

2.2. Botón PROG

El botón PROG está ideado para resetear manualmente el dispositivo para devolverlo a la versión de fábrica y está localizado debajo de los leds, el procedimiento para resetear está explicado en el punto 7.3.3 Configuración de fábrica a través del botón PROG

3. Conexiones

3.1. Conexiones eléctricas

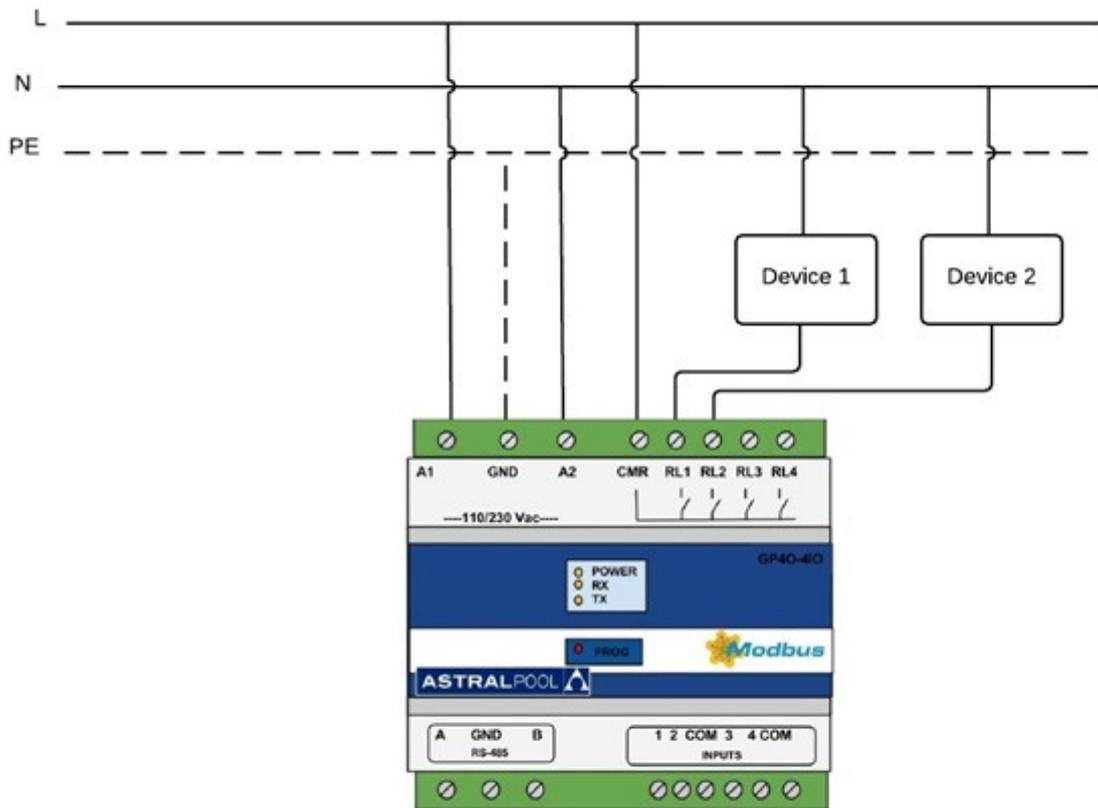


Imagen 2 Conexiones eléctricas

Los relés del GPIO pueden permitir o interrumpir la corriente de cuatro dispositivos de manera simultánea. Sin embargo algunas precauciones se han de tener en cuenta a la hora de garantizar la integridad del GPIO.

1. Todas las salidas deben estar alimentadas al mismo voltaje (terminal CMR en Imagen 2). En caso de no hacerlo la diferencia de potencial que se crearía de intentar controlar distintas líneas puede dar lugar a sobrealimentación y provocar fallos irreversibles a pesar de la protección interna del aparato, la salida de voltaje nominal es de 250V.
2. Del mismo modo, la suma de las corrientes no debe de ser mayor que la intensidad nominal del GPIO ($I_n=5A$). De lo contrario los relés no soportarán la carga.
3. Cuando el GPIO controla varios aparatos, la intensidad debe cortarse por la misma fase, en el ejemplo se ha cortado la línea. Si se llegase a un cortocircuito dentro del GPIO este puede sufrir daños.

Nota: Tómese estos tres consejos e intégreles en su metodología de trabajo lo más pronto posible con el fin de instalar correctamente el GPIO, la Imagen 2 muestra un ejemplo de cómo controlar la maniobra de dos dispositivos.

3.2. Bus Connections

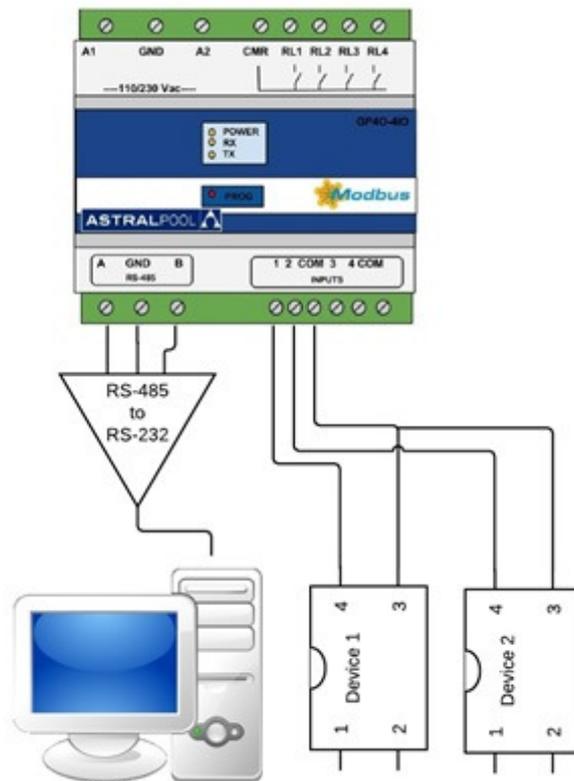


Imagen 3 Conexiones BUS

3.2.1 conexión de contactos libres de tensión

El GPIO tiene cuatro entradas discretas implementadas, las cuales no tienen voltaje asociado (libres de tensión) y pueden detectar pulsos digitales. Estos pulsos pueden ser leídos y sus flancos de bajada almacenados, también tiene implementado un filtro digital que garantiza una medida correcta. Si se quiere leer los pulsos de señales con voltaje se debe de usar un **opto aislador**. En el apartado 8 Digital_inputs se detallan sus características.

3.2.2 Estandar RS-485 y conexiones de entradas digitales

Por otra parte el bus RS-485 aporta el estándar de comunicaciones conocido como ANSI/TIA/EIA-485. Algunos fabricantes intercambian los cables “A” y “B” con respecto de este estándar, esto puede ocasionar confusiones pero no es el caso del GPIO. Téngase en cuenta este aspecto cuando conecte dispositivos de otras compañías en su red MODBUS.

4. CARACTERÍSTICAS DEL CABLEADO

4.1. Características del cableado eléctrico

Ya que el GPIO se alimenta por la red doméstica no importa si línea (L, negro marrón o gris) está conectado al conector A1 y el neutro (N, azul) al A2 o viceversa. El cable de protección eléctrica (PE, verde y amarillo) deberá conectarse al conector GND.

Este dispositivo puede ser alimentado por otros valores de tensión alterna, consúltese el folleto manual para más información BRON_GPIO_62368_v1.0_ES.

4.2. Estandar Rs-485 y entradas digitales

El cableado recomendado para las comunicaciones MODBUS-RTU está basado en una estructura lineal, bus activo, con terminación en ambos lados. Es posible conectar y desconectar dispositivos durante la operación sin afectar a otros dispositivos. El cable debe de ser trenzado y apantallado de acuerdo con la norma EN 50 170.

Los datos transmitidos en el dispositivo soportan desde de 1200m de cable sin repetidores a 10 Km con repetidores, siempre y cuando la instalación sea de acuerdo al estándar

Para los pares trenzados del cable RS-485, una característica de impedancia superior a 100 Ohmios será recomendable, especialmente para comunicaciones a 19200 baudios o más.

5. AISLAMIENTO DEL BUS Y TERMINACIÓN RESISTIVA

Si el usuario tiene acceso al bus, este debería de estar doblemente aislado. En general la accesibilidad del bus dependerá de cada instalación, el GPIO dispone de un aislamiento de seguridad en la capa física del bus del GPIO. Además, por motivos de seguridad, se recomienda asegurar que los demás dispositivos también comparten el mismo aislamiento.

Es recomendable realizar una terminación al final de cada extremo del BUS, incluso cuando se use un cable más largo o velocidad de comunicación superiores. La terminación resistiva se usa para prevenir que una señal de radio frecuencia (RF) interfiera en las comunicaciones. Debe usarse en los dos extremos de la línea y conectarse en paralelo (tal y como se puede ver en la Imagen 4, $R_t = 120 \text{ Ohm}$, 0.5 W).

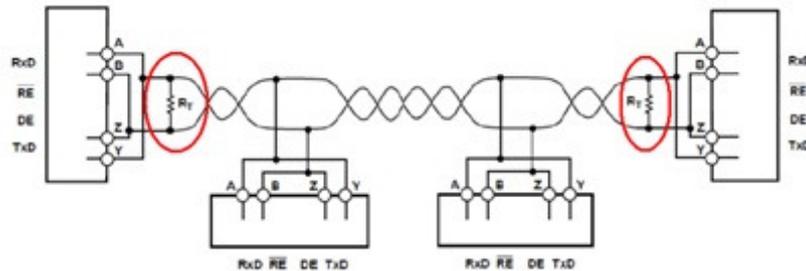


Imagen 4. Terminación resistiva

6. DESCRIPCIÓN MODBUS

6.1. Funciones soportadas

Por favor, tenga cuidado al utilizar estas funciones y asegúrese de que la función utilizada es la correcta.

MODO ACCESO A BIT

Las funciones que acceden a bits están implementadas de acuerdo con el estándar MODBUS-RTU descrito en:

http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

0x01 READ COILS

0x0F WRITE MULTIPLE COILS

0x05 WRITE SINGLE COIL.

0x02 READ DISCRETE INPUTS.

MODO ACCESO A REGISTRO

Las funciones que acceden a registros están implementadas de acuerdo con el estándar MODBUS-RTU descrito en:

http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Los registros son, en general enteros sin signo de 16 bits.

0x03 READ HOLDING REGISTERS

0x04 READ INPUT REGISTERS

0x10 WRITE MULTIPLE REGISTERS

0x06 WRITE SINGLE REGISTER

6.2. Códigos de Excepción

Los códigos de excepción están implementados de acuerdo con el estándar MODBUS-RTU descrito en:

http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Los códigos de excepción implementados son del 1 al 4

La excepción 4 aparece cuando se trata de escribir sobre power_counter o wtd_counter, descrito en el punto 11, los códigos de excepción aparecen de forma normal

7. DESCRIPCIÓN Y CONFIGURACIÓN DEL APARATO

7.1. Descripción General

Por medio de peticiones a través de MODBUS, se pueden manipular los 4 relés y el estado actual de sus 4 entradas digitales se pueden leer.

Cuando se realizan peticiones simultáneas a través de MODBUS y no hay ningún conflicto en el BUS, la última petición será la que tenga efecto.

En general, no hay una verificación en la consistencia de los valores mandados a específicos registros. Por ello es responsabilidad del operador verificar dicha consistencia.

En este manual, el número en hexadecimal es representado en formato **0xZZ**, donde ZZ es el número.

El Input Register que indica el estado del GPIO en el diagrama de estados se indica en el punto 7.2 Diagrama de estados y todas las localizaciones en memoria están en el punto 11 Mapa básico de registros Modbus RTU.

7.2. Diagrama de estados

Cuando el sistema se enciende el estado es siempre start. Si no se activa el Watchdog y no se realiza ninguna petición el estado permanecerá en Start.

Desde Start es posible ir al estado Request (si se realiza una petición) o al estado Watchdog (si este es activado). Desde el estado de Request también podemos volver a Request cuando realizamos otra petición-

Y cuando el Watchdog se activa entramos al estado Watchdog, de este estado o bien podemos salir enviando una petición y entrando en Request o bien si la configuración de Watchdog es la correcta al dispararse se volverá al estado start (indicado con una flecha con puntos) como se ve en la Imagen 5.

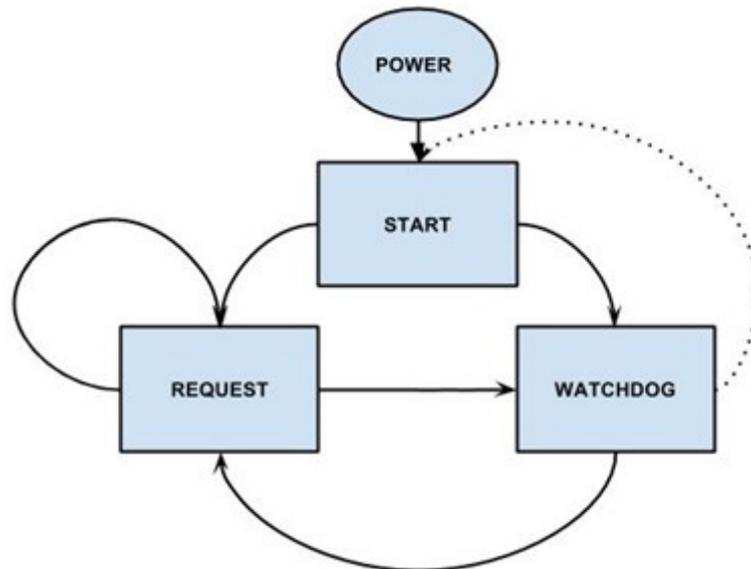


Imagen 5. Diagrama de estados

7.3. Configurando el GPIO

7.3.1 Configuración de dirección

La dirección del GPIO en el bus se realiza a través del Holding Register 0x00

ID_Address: Dirección del GPIO en el bus

Valor de fábrica: 0x1A.

Valores sugeridos: 0x1A - 0x1E.

El valor predefinido para el GPIO es 0x1A. Sin embargo, si se quiere implementar más de un GPIO entonces sus direcciones no deben de ser las mismas.

Esta situación puede solucionarse escribiendo un nuevo valor en el holding register 0x00. El siguiente ejemplo muestra cómo hacer este procedimiento.

Ejemplo: cambiar la ID_Address de 0x1A (defecto) a 0x1B.

Mensaje transmitido: 1A 10 00 00 00 01 02 00 1B 58 AB

Donde:

- 1A es la dirección del esclavo (la actual)
- 10 es la función usada. Write multiple registers.
- 00 00 es la dirección del holding register que se va a escribir.
- 00 01 es el número de registros que se van a escribir (1 en este caso).
- 02 es el número de bytes de datos que se van a enviar
- 00 1B es la nueva ID_Address
- 58 AB es el CRC.

7.3.2. Configuración de comunicaciones

La selección de los Baudios en la comunicación en serie se realiza a través del holding register 0x01. Por defecto es de 9600 bps y 8E1 (8 bits de datos, par con un bit de paro). Sin embargo, se pueden soportar otras velocidades, esto se muestra en la tabla de abajo.

La razón por la cual se debe soportar tramas de N2 es para mantener el estándar de MODBUS, ya que se mandan 11 bits por byte (1 de marcha + 8 de datos + 1 de paridad + 1 de paro). Cuando no se indica el tipo de paridad se introducirán 2 bits de paro para mantener la estructura de 11 bits requeridos por el estándar.

Por razones de compatibilidad, las tramas N1 también están soportadas. Sin embargo, téngase en cuenta que de esta manera no se está cumpliendo el estándar MODBUS ya que solo se usan 10 bits por byte.

Por esto, la selección de la velocidad de transmisión se completa definiendo el ratio de Baudios, número de bits de datos, paridad y número de bits de paro.

COM_Setup:	Configuración de comunicaciones	
Valor de fábrica:	0	9600, 8E1
Valores soportados:	0	9600, 8E1
	1	19200, 8E1
	2	9600, 8N2
	3	19200, 8N2
	4	9600, 8N1
	5	19200 8N1

7.3.3 Configuración de fábrica a través del boton PROG

El siguiente ejemplo sirve para ilustrar el procedimiento para resetear el GPIO con el fin de recuperar la ID_Address original. Sea consciente de que los registros DI_x_counter (donde "x" es un número del 1 al 4) y los latcheadas alarm registers serán borrados en el proceso (estos registros están detallados en los puntos 9.1 y 10.1.4 respectivamente). Con el fin de resetear a los valores de fábrica los pasos han de ser los siguientes

1. Desconecte el aparato
2. Presione el botón PROG
3. Mientras mantiene presionado el botón PROG, vuelva a conectar el GPIO.
El GPIO volverá al estado Start y tendrá otra vez el ID_Address original en el holding register 0x00 (0x1A). Sin embargo se restearán los registros DI_x_counter (donde "x" es un número del 1 al 4) y las alarmas latcheadas así que tenga cuidado con esto.

7.4. BROADCASTING

El broadcasting no está soportado por el GPIO.

7.5. WATCHDOG

El Watchdog es un temporizador implementado en el GPIO cuya finalidad es comprobar si la comunicación en el bus está activa.

Según se definirá a continuación, el tiempo considerado para disparar el Watchdog y las tareas que debe realizar en ese caso necesitan ser definidas.

Cuando el GPIO pierde comunicación durante un tiempo mayor que el watchdog_time, tanto el bit más significativo del registro de alarmas latcheadas del holding register 0x20 como el del input register 0x01 se pondrán a 1.

Nota: Si el watchdog_time es menor de 30 segundos la primera vez que encendamos el GPIO este se disparará a los 30 segundos por razones de seguridad.

7.5.1. WATCHDOG Time

El watchdog time se define en el holding register 0x10. Este tiempo está en segundos. 0 significa que el watchdog está deshabilitado, este es el valor por defecto.

Cada vez que el watchdog se dispare el registro WDT_counter situado en el holding register 0x31 se incrementará, explicado en el punto 9.3.

Para activar la función de watchdog, establezca el tiempo de watchdog a un valor diferente de 0.

Ejemplo: establecer el tiempo de watchdog a 30s

Mensaje transmitido: 1A 10 00 10 00 01 02 00 1E 9A 38

Donde:

- 1A es la dirección del esclavo
- 10 es la función utilizada. Frite Múltiple Registers.
- 00 10 es la dirección del holding register que se va a escribir.
- 00 01 es el número de registros que se van a escribir, 1 en este caso.
- 02 es el número de bytes que se van a recibir
- 00 1E es el valor enviado, 30 es decimal.
- 9A 38 es el CRC.

Ahora, el watchdog_time está establecido en 30d. Por lo tanto, cuando dos mensajes correctamente contruidos se reciban en el GPIO en menos de 30 segundos, el watchdog no se activará. De modo contrario, sí que se activará.

Para conocer el valor del watchdog_time, es necesario leer el holding register.

Mensaje recibido: 1A 03 02 00 1E 5C 4E

- 1A es la dirección del esclavo
- 03 es la función usada. Read holding registers.
- 02 es el número de bits de datos que se van a enviar
- 00 1E es el tiempo configurado (30s).
- 5C 4E es el CRC.

7.5.2. WATCHDOG config

Localizado en el holding register 0x11, indica que estado tendrá lugar cuando se dispare el watchdog

El Byte alto:

- Si es 0, entonces cuando se dispare el watchdog, este forzará al GPIO a entrar en el estado Watchdog.
- Si es puesto a un valor distinto de 0, entonces cuando se dispare el watchdog, este entrará en el estado de Start.

El Byte bajo: no se usa en el GPIO

Ejemplo: Configuración del watchdog para que entre en el estado Watchdog al dispararse.

Mensaje transmitido: 1A 10 00 11 00 01 02 00 00 1B E1

Donde:

- 1A es la dirección del esclavo.
- 10 es la función usada. Write multiple registers.
- 00 11 es la dirección del primer registro que se va a escribir.
- 00 01 es la cantidad de registros que se van a escribir, 1 en este caso
- 02 es el número de bytes que se van a enviar.
- 00 00

Byte alto: al dispararse el GPIO estará en el estado de watchdog.

Byte bajo. No se usa.

- 1B E1 es el CRC.

7.5.3. WDT_Relay_state

El WDT_Relay_State determina que relés estarán abiertos o cerrados cuando se active el watchdog, está localizado en el holding register 0x14.

La manera de controlarlos es mediante el valor en binario mandado a este registro, desde el bit más significativo al cuatro son los relés en orden creciente.

Ejemplo: poner el WDT_relay_state de la siguiente manera: el primero y el tercero cerrado y los demás abiertos (RL4=0; RL3=1; RL2=0; RL1=1).

Mensaje transmitido: 1A 10 00 14 00 01 02 00 05 DB B7

Donde:

- 1A es la dirección del esclavo
- 10 es la función usada. Write Multiple Registers.
- 00 14 es la dirección del primer registro que se va a escribir.
- 00 01 es el número de registros que se van a escribir.
- 02 es el número de bytes que se van a enviar.
- 00 05 en binario, es el equivalente a cerrar sólo el primer y el tercer relé.
- DB B7 es el CRC.

NOTA: Téngase en cuenta que el watchdog se ha implementado por motivos de seguridad. Sin embargo el que esté activado o no depende siempre del criterio del instalador/integrador. Siempre se tiene que tener en cuenta las implicaciones al implementarlo. Úsese esta característica bajo su propia responsabilidad.

Tanto el Watchdog_time como el Watchdog_config y el WDT_relay_state deben de definirse conjuntamente.

8. Entradas Digitales

Localizadas en el Input Register 0x20 pueden adquirir el valor de señales discretas tales como los contactos libres de tensión, o bien contactos con tensión aislados con **optoaisladores (o opto acopladores)**. Sus flancos de bajada pueden ser filtrados y almacenados (explicaciones en puntos 9.1 Digital_input Counters y 8.1 Configuración de los filtros de entrada digitales respectivamente). La Imagen 6 muestra un ejemplo del conexionado MODBUS y para leer dos dispositivos que generan señales libres de tensión (como un interruptor).

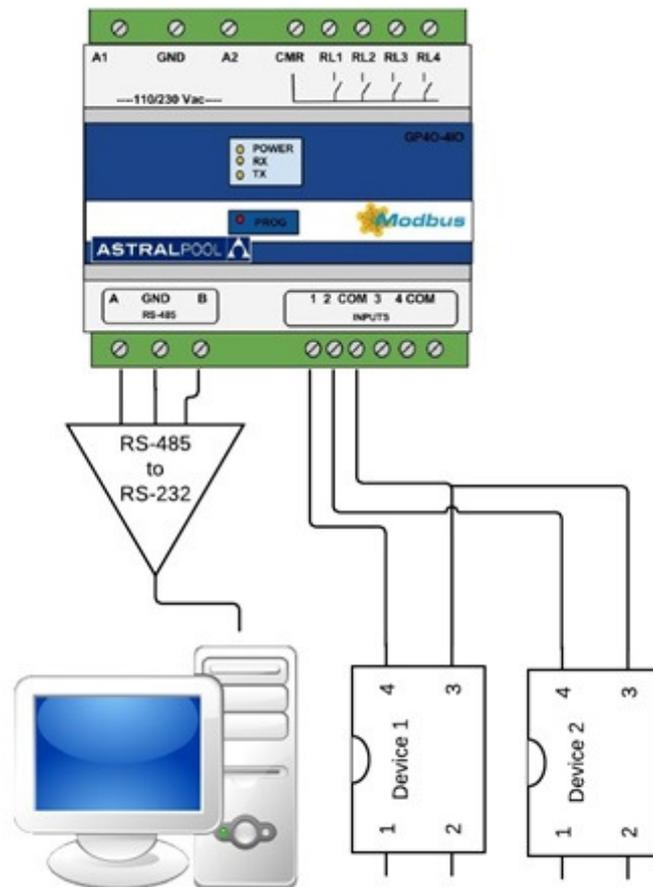


Imagen 6 Entradas digitales en el GPIO

8.1. Configuración del filtro de las entradas digitales

Algunas señales tienen comportamientos espurios y pueden ocasionar problemas o medidas no consistentes. Estas conexiones pueden tener picos aleatorios producidos por rebotes físicos y los contadores pueden almacenar más de un incremento por flanco de bajada real, ilustrado en el diagrama de arriba de la Imagen 7.

La intención de este filtro es evitar estos flancos de bajada erróneos y conseguir un solo incremento por pulso, ilustrado en el diagrama de abajo de la Imagen 7.

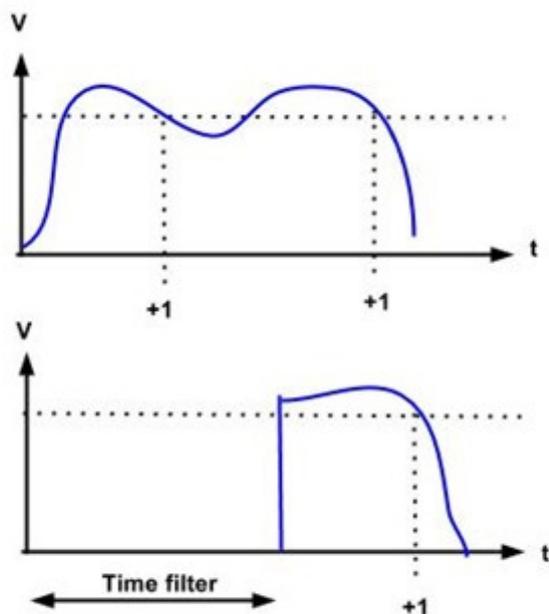


Imagen 7 Señal espuria con y sin filtro digital

Este problema puede solucionarse configurando digInputFilterConfig (localizado en el holding register 0x12) definiendo el tiempo en el que el GPIO ignorará la señal, después de este tiempo la señal será leída normalmente.

Este tiempo ha de ser mayor cuando la señal tenga más picos y menos o cero cuando la señal no tenga o tenga pocos picos. Los parámetros del filtro serán:

0x12 Byte alto Tiempo necesario para ignorar malas lecturas.
 0 Sin filtro. 2 Tiempo de 10ms 4 Tiempo de 500ms
 1. Tiempo de 1ms. 3 Tiempo de 100ms

0x12 Byte bajo Entrada o entradas que se van a filtrar.

Ejemplo: configuración de filtro para las entradas 1, 3 y 4 con 500ms de tiempo.

Mensaje transmitido: 1A 10 00 12 00 01 02 04 0D D8 D7

Donde

1A es la dirección del esclavo.
 10 es la función usada. Write Multiple Registers.
 00 12 es la dirección del primer registro que se va a escribir.
 00 01 es la cantidad de registros escritos.
 02 es el número de bytes que se van a escribir.
 04 0D

Byte alto 0000 0100 para indicar los 500ms.

Byte bajo 0000 1101 para indicar las entradas 1, 3 and 4.

D8 D7 es el CRC.

Mensaje recibido: 1A 10 00 12 00 01 A2 27

8.2. Lectura de contactos con tensión (Wet contacts)

En vez de generar un cortocircuito o un abierto como las de libre tensión (dry contacts), estas generan una diferencia de potencial entre sus terminales. Si se quiere leer un contacto con tensión (wet) se debe usar un **opto aislador** con el fin de garantizar aislamiento entre las entradas del GPIO y el contacto con tensión, como se puede ver en el ejemplo de la Imagen 8.

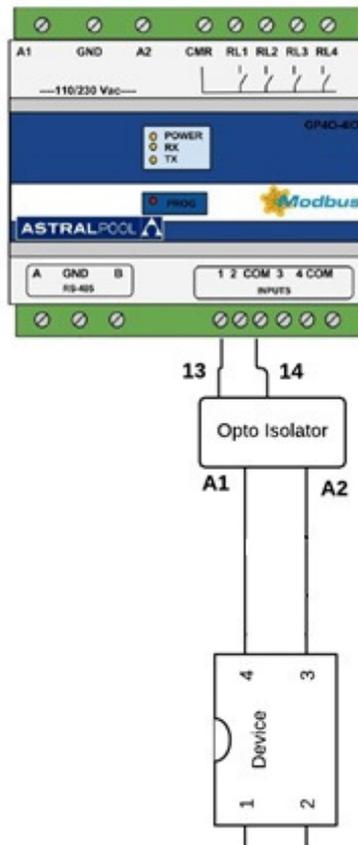


Imagen 8 Lectura de contactos con tensión en el GPIO

Si se requiere más información acerca de los **opto aisladores**, por favor consúltese cualquier manual de un **opto aislador**.

9. Contadores

El GPIO tiene un total de 3 contadores, cada uno cuenta distintos aspectos del GPIO relacionados con su comportamiento.

9.1 Digital Input Counters

Estos contadores están localizados en el input register 0x03-0x06 y sólo se incrementan cuando la entrada detecta un flanco de bajada (cuando pasa de 1 a 0). Se llaman DI_x_counter (donde "x" es un número del 1 al 4). A partir de la versión de firmware 120 (SW) estos registros son de 32 bits en vez de 16 bits.

Recuerde que si el GPIO se reprograma como se explicó en el punto 7.3.3 Configuración de fábrica a través del botón PROG estos input registers serán reseteados.

Ejemplo: lectura de digital input counters como input registers. Valores después de resetear el dispositivo a configuración de fábrica.

Mensaje transmitido: 1A 04 00 03 00 04 02 22

1A es la dirección del esclavo

04 es la función usada. Read Input Registers.

00 03 es la dirección del input register que se va a leer.

00 04 es el número de bytes que se van a leer.

02 22 es el CRC

Mensaje recibido: 1A 04 08 00 00 00 00 00 00 00 00 55 E9

9.2 Power counter

Localizado en el holding register 0x30 se incrementa cada vez que el GPIO es conectado,

Ejemplo: consultando el power_counter antes y después de forzar al dispositivo a restearse a configuración de fábrica.

Mensaje transmitido (antes de resetear): 1A 03 00 30 00 01 87 EE

Donde

1A es la dirección del esclavo

03 es la función usada. Read Holding Registers.

00 30 es la dirección del holding register que se va a leer.

00 01 es el número de bytes que se va a leer.

87 EE es el CRC

Mensaje recibido: 1A 03 02 00 3F 9C 56

Mensaje transmitido (después de resetear): 1A 03 00 30 00 01 87 EE

Mensaje recibido: 1A 03 02 00 40 DD B6

Donde:

1A es la dirección del esclavo.

03 es la función usada. Write multiple registers.
 02 son los bytes recibidos.
 00 40 son los datos bytes que se han leído.
 DD B6 es el CRC

9.3 WDT counter

Localizado en el holding register 0x31, se incrementará cada vez que el watchdog se dispare

Ejemplo: consultando WDT_counter antes de que el watchdog se dispare

Mensaje transmitido (antes) 1A 03 00 31 00 01 D6 2E

Mensaje recibido: 1A 03 02 44 D3 AE DB

Donde

1A es la dirección del esclavo
 03 es la función usada, read holding registers.
 02 son los bytes recibidos.
 44 D3 es el valor del WDT_counter
 AE DB es el CRC

La cantidad de veces en las que el GPIO ha entrado en watchdog son 0x44D3

Mensaje transmitido (después): 1A 03 00 31 00 01 D6 2E

Mensaje recibido: 1A 03 02 44 D4 EF 19

Donde

1A es la dirección del esclavo
 03 es la función usada, read holding registers.
 02 son los bytes recibidos.
 44 D4 es el valor del WDT_counter.
 EF 19 es el CRC

10 Modos de operación

10.1. Modo básico

En esta sección se asume que se ha establecido una conexión correcta con el GPIO y por lo tanto, las configuraciones y la configuración de comunicaciones son las correctas.

10.1.1. Start relay state

Es la disposición de los relés en el GPIO cuando se encienda (localizados en el holding register 0x15 y coils 0x150 a 0x153) la Imagen 9 muestra la disposición de estos relés en relación con el registro de la tabla.

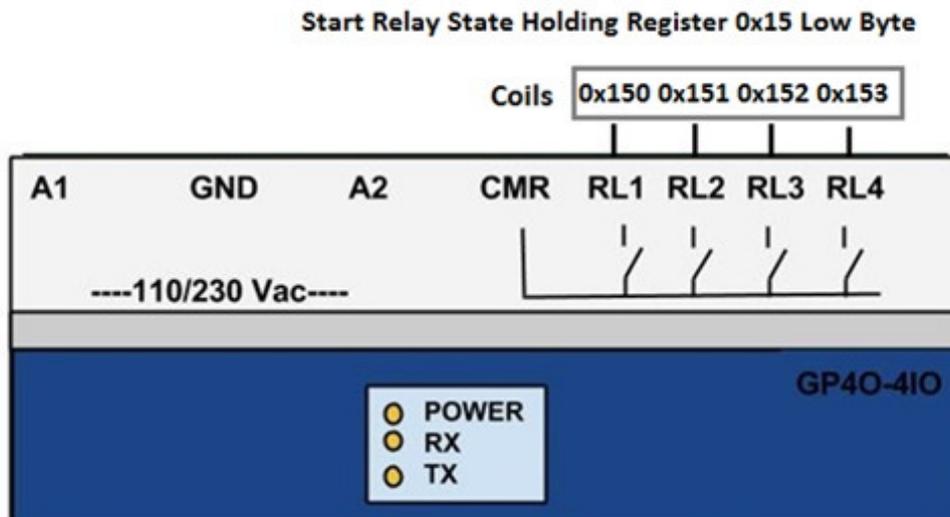


Imagen 9 Startt_relay_state

Ejemplo: configurar el start_relay_state con todos los valores a 0

Mensaje transmitido: 1A 10 00 15 00 01 02 00 00 1A 65

Donde:

1A es la dirección del esclavo.
 10 es la función usada, write multiple registers.
 00 15 es la dirección del holding register que se va a escribir.
 00 01 es el numero de registros que se van a escribir.
 02 es el numero de bytes de datos que se van a enviar.
 00 00

Byte Alto: no tiene uso

Byte Bajo: los cuatro bits menos significativos se refieren a los relés en orden creciente

1A 65 es el CRC.

Mensaje recibido: 1A 10 00 15 00 01 13 E6

10.1.2. Comprobando el estado actual

El estado actual del GPIO está disponible a través del input register status 0x00. Este registro tiene un significado diferente para el **Byte alto** y para el **Byte bajo**.

En el **Byte bajo**, el bit menos significativo se usa para mostrar si hay algún error activo, cuando exista un error este bit será 1. Se puede haber más información detallada acerca de los errores detectados a través del input register 0x01 (alarmas instantáneas) o bien a través del holding register 0x20 (explicado más adelante en los puntos 10.1.5 Comprobando el registro de alarmas instantáneas y 10.1.4 Comprobando el registro de alarmas latcheadas respectivamente).

Sólo se ha implementado la alarma de watchdog, la cual significa que, si existe un error, este es el de watchdog. Por otra parte el **Byte Alto** indica los códigos para los distintos estados, como se muestra en la siguiente tabla.

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

Ejemplos de valores del input register Status 0x00:

0x000	start state no hay errores.
0x001	start state y error, debido al watchdog.
0x100	request state y no hay error.
0x201	watchdog state y error debido al watchdog.

Ejemplo, leer el estado actual en el GPIO

Mensaje transmitido: 1A 04 00 00 00 01 32 21

Donde:

1A es la dirección del esclavo.

04 es la función usada. Read input registers.

00 00 es la dirección del primer registro que se va a leer.

00 01 es el número de input registers que se van a leer

32 21 es el CRC

Mensaje recibido: 1A 04 02 **02 01** 1D 92, lo cual indica que está en el watchdog state con la alarma activada..

Donde:

1A es la dirección del esclavo.

04 es la función usada. Read input registers.

02 es el número de bytes que se van a leer.

02 01

Byte alto: el GPIO está en el Watchdog state.

Byte bajo: El GPIO tiene una alarma activa (la del watchdog).

1D 92 es el CRC.

10.1.3. Petición de cambio de los relés

Los relés pueden cambiar de estado mediante una petición al GPIO a través del control_word_wo (localizada en el holding register 0x21 o los coils 0x210 al 0x213).

Se puede cambiar tanto por holding register como por coils, ambos métodos hacen lo mismo pero no se deben de mezclar para evitar errores, la imagen 10 muestra la disposición de estos holding registers y coils con respecto al dispositivo.

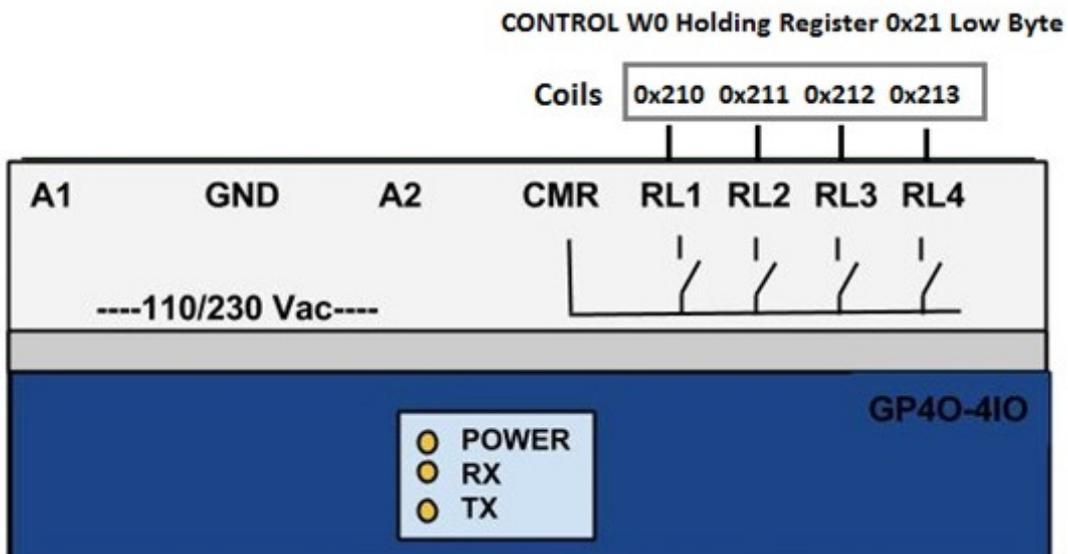


Imagen 10 localización en la tabla de los relés

Ejemplo: activando los relés 1 y 3.

Mensaje transmitido: 1A 10 00 21 00 01 02 00 05 DE 12

Donde:

1A es la dirección del esclavo.

10 es la función usada. Write multiple registers.

00 21 es la dirección de los holding registers que se van a escribir.
00 01 es el número de registros que se van a escribir.
02 es el número de bytes de datos que se van a escribir.
00 05

Byte alto: no tiene uso.

Byte bajo: es (en binario) el equivalente a que estén cerrados sólo el primero y el segundo relé

1A 65 es el CRC.

Mensaje recibido: 1A 10 00 21 00 01 52 28

10.1.4. Comprobando el registro de alarmas latcheadas

Es posible comprobar las alarmas que se han activado desde que el GPIO ha sido conectado, o reseteado. Solo se ha implementado la alarma de watchdog.

Para comprobarlo, se debe leer el holding register 0x20, desde el bit 0 al 14 no tiene uso, pero se puede leer, el bit 15 es el de la alarma de watchdog, también se pueden leer como coils (desde el 0x200 al 0x20F).

El valor 1 significa que la alarma está activada, 0 significa desactivada.

Si la alimentación falla o el equipo se desconecta, todas las alarmas volverán a 0. Estos valores también se pueden forzar.

Ejemplo: leer alarmas latcheadas como holding registers cuando la alarma de watchdog esté activada

Mensaje transmitido: 1A 03 00 20 00 01 86 2B (read holding register 0x20).

Mensaje recibido: 1A 03 02 80 00 BD 86

Donde:

1A es la dirección del esclavo.

03 es la función utilizada. Read holding registers.

02 es el número de bytes de datos que se van a leer.

80 00

Byte Alto: el bit más significativo indica error de watchdog

Byte Bajo: no se usa.

BD 86 es el CRC.

Ejemplo: leer alarmas latcheadas como coils cuando la alarma de watchdog haya sido activada.

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

Mensaje recibido es: 1A 01 01 01 96 AC

Donde:

1A es la dirección del esclavo.

01 es la función usada. Read Coils.

01 es el número de coils que se van a leer.

01 es el coil leído, indica que la alarma se ha activado.

96 AC es el CRC.

Ejemplo: resetear todas las alarmas escribiendo 00 00 en el holding register 0x20

Mensaje transmitido: 1A 10 00 20 00 01 02 00 00 1F C0

Donde:

1A es la dirección del esclavo.

10 es la función usada. Write multiple registers.

00 20 es la dirección del primer holding register que se va a leer.

00 01 es el número de holding registers que se van a escribir, 1 en este caso.

02 es el número de bytes de datos que se van a enviar.

00 00 es el dato enviado (0 para desactivar la alarma).

1F C0 es el CRC.

Mensaje recibido: 1A 10 00 20 00 01 03 E8

10.1.5. Comprobando el registro de alarmas instantáneas

Cuando exista un error el bit menos significativo de Status IR 0x00 y el más significativo de alarmas instantáneas se pondrán a 1, tal y como se explicó en 10.1.2 Comprobando el estado actual

Ejemplo: comprobando el estado del registro alarmas instantáneas

Mensaje transmitido: 1A 04 00 01 00 01 63 E1

Donde

1A es la dirección del esclavo.

04 es la función usada. Read input registers.

00 01 es la dirección del primer input register que se va a leer.

00 01 es el número de input registers que se van a poner.

63 E1 es el CRC.

Mensaje recibido: 1A 04 02 80 00 BC F2

Donde:

1A es la dirección del esclavo.

04 es la función usada. Read input registers.

02 es la cantidad de bytes que se van a leer.

80 00 es el valor de los input registers leídos, solo el valor más significativo está a 1

BC F2 es el CRC.

La localización de este input register se muestra en la tabla del punto 11 Mapa Básico de registros Modbus RTU, del mismo modo que las latcheadas, del bit 0 al 14 no se usa, mientras que el 15 indica error de watchdog cuando se activa.

11. Mapa básico de registros MODBUS RTU

La tabla que se muestra en este punto es nuestro mapa de registros exclusivo y original, con el nombre de la función y su dirección.

Además del propio mapa de registros, hay una relación directa entre los holding registers 0x21 y los input registers 0x00. Mientras que el primero sirve para pedir acciones, el segundo sirve para comprobar que la acción requerida se está realizando.

También hay una relación directa entre las alarmas latcheadas del holding register 0x20 y las alarmas instantáneas del input register 0x01. Mientras que el primero son todas las alarmas acumuladas desde que el GPIO se encendió, el segundo son las alarmas que actualmente están activadas. Como sólo se ha implementado el watchdog esa será la única alarma posible. Lo que significa que estos dos registros deben coincidir cuando la alarma se active.

Para resetear estas alarmas es necesario resetear el holding register 0x20 y no los input registers 0x01. Ya que los input registers son de solo lectura y la alarma desaparecerá cuando se desactive.

Nota: una desconexión de la corriente de alimentación también reseteará todas las alarmas latcheadas

Name	Holding Registers	Input registers	Coils	Inputs	Description
ID_Address	0x00				Este parámetro es la dirección del GPIO, por defecto 0x1A. Si la instalación tiene más de un GPIO, será necesario que los dos no tengan la misma dirección
COM_Setup	0x01				Este parámetro determina la velocidad en la comunicación MODBUS, hay 6 posibilidades Configuración de fábrica 0. 9600, E81 Valores soportados 1. 19200, 8E1 2. 9600, 8N2 3. 19200, 8N2 4. 9600, 8N1 5. 19200, 8N1
ID_Manufacturer_hi	0x02				Este parámetro indica el byte alto que representa el manufacturer code.
ID_Manufacturer_lo	0x03				Este parámetro indica el byte bajo que representa el manufacturer code
ID_Product_code_hi	0x04				Este parámetro indica el byte alto que representa el Product code.
ID_Product_code_lo	0x05				Este parámetro indica el byte bajo que representa el Product code.
Reserved	0x06				Reservado.
HW_Version	0x07				Este parámetro indica la versión de hardware del GPIO
SW_Version	0x08				Este parámetro indica la versión de Software del GPIO
MODEL_Serie_hi	0x09				Este parámetro indica el byte alto del nº de serie del GPIO.
MODEL_Serie_lo	0x0A				Este parámetro indica el byte bajo del nº de serie del GPIO.
MODEL_Production_hi	0x0B				Este parámetro indica el byte alto del nº de lote del producto
MODEL_Production_low	0x0C				Este parámetro indica el byte bajo del nº de lote del producto
Watchdog_time	0x10				Este parámetro indica el tiempo en segundos que puede pasar desde la última comunicación hasta que el GPIO se active. Si se pone 0 se desactiva, rango 0-65535
Watchdog_config	0x11 low byte high byte				Este parámetro define como actuará el watchdog cuando se active.. Byte Bajo Is the state when the Watchdog triggers (not used). Byte Alto Si es 0 entonces cuando se active irá al estado watchdog Si es diferente de 0 irá al estado start
					Este parámetro define el tiempo en el cual se

DigInputFilterConfig	0x12 bit 0...3 bit 7...10				ignoran las lecturas de las inputs del GPIO para evitar malas medidas Byte Alto: en binario, las entradas filtradas. Byte bajo 0 Sin filtro. 1 filtro con 1 milisegundo. 2 filtro con 10 milisegundos. 3 filtro con 100 milisegundos. 4 filtro con 500 milisegundos.
wdt_relay_state	0x14 bit 0...3		0x140 0x141 0x142 0x143		Este registro establece la disposición de relés cuando el GPIO entra en el estado de watchdog Byte bajo: En binario, el estado deseado de los 4 relés. RL1 RL2 RL3 RL4
Start Relay State	0x15 bit 0...3		0x150 0x151 0x152 0x153		Este registro define el estado de los 4 relés al entrar en el estado de start. Byte bajo: en binario, el estado deseado de los relés RL1 RL2 RL3 RL4
Latched Alarm Watchdog	0x20 .bit 0... 14 .bit 15		0x200 to 0x20E 0x20F		Este parámetro indica si se ha activado alguna alarma desde que el GPIO se ha encendido Bit más significativo: 1 watchdog activado, 0 watchdog no activado.
control_w0	0x21 .bit 0...3 bit 4..15		0x210 0x211 0x212 0x213 0x214 to 0x21F		Byte bajo: en binario. RL1. 0 = Relé cerrado, 1 = relé abierto RL2. 0 = Relé cerrado, 1 = relé abierto d RL3. 0 = Relé cerrado, 1 = relé abierto RL4. 0 = Relé cerrado, 1 = relé abierto d no usado
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_4_Counter_lo	0x30 0x31	0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A			Se incrementa cada vez que el GPIO se conecta a la red. Se incrementa cada vez que el GPIO entra en watchdog. Cada uno de estos contadores se incrementa cuando detectan un flanco de bajada en el input correspondiente del GPIO. En SW < 120, DI_x_counter son de 16 bits sin signo y se mapean del IR 0x03 al IR 0x06. De SW 120 en adelante, DI_x_counter son de 32 bits sin signo y se mapean en parte alta y parte baja desde el IR 0x03 al IR 0x0A.
Status Alarm ON 0: Start State 1: Request State 2: Watchdog state		0x00 low byte .bit 0 high byte .bit 8...11		0x000 0x008... 0x00B	Byte bajo: Indica la alarma, 1 = alarma de Watchdog. 0 = no hay alarma de watchdog Byte alto: Indica el estado actual del GPIO.
Instantaneous Alarm Watchdog		0x01 .bit 1...14 .bit 15		0x01F	Este parámetro indica las alarmas instantáneas. Bit más significativo: 1 watchdog activado, 0 watchdog no activado.

Digital inputs		0x02 .bit0...3		0x020 0x021 0x022 0x023	Este registro define los contactos leídos a través de las entradas. Input 1 Input 2 Input 3 Input 4
----------------	--	-------------------	--	----------------------------------	---

12. Revisión del producto

Manual v.1.05: Toda la información de este manual describe el comportamiento de la versión de Hardware 1.02 y Software 1.20
