Features

- **Modbus-Interface**
  - Widely used protocol for measuring and automation
- **gpio.NET *Core* Interface**
  - Uniform set of commands and registers
  - Supports Modbus RTU and Modbus ASCII via RS232/RS485
  - Supports Modbus ASCII via USB
  - Gateway with protocol conversion between RS232/RS485 (RTU) and USB (ASCII)
  - Configuration stored in EEPROM
- **RS232/RS485**
  - Selectable baud rate from 1200 Baud to 1MBit
  - Free configurable serial device (data width, parity, stop bits)
  - Selectable operation mode (RS232 or RS485)
- **USB**
  - USB 2.0 CDC-Device
- **Unique serial number**
  - Modbus-Address changeable through Modbus command via serial line
- **1 KByte EEPROM**
  - Seperated Config-Section and User-Section
  - Both sections can be write-protected seperately
- **4 relays**
  - max. 220 V
  - max. 5 A
- **4 digital inputs**
  - opto-decoupled inputs
- **4 frequency counters**
  - simultaneously counting on all digital inputs
  - up to 32 kHz
- **filtered 32bit impuls counter**
  - selectable pulse width between 32µs and 65ms

gpio.NET

Relais

Preliminary

Short Description

gpio.NET
Relay card

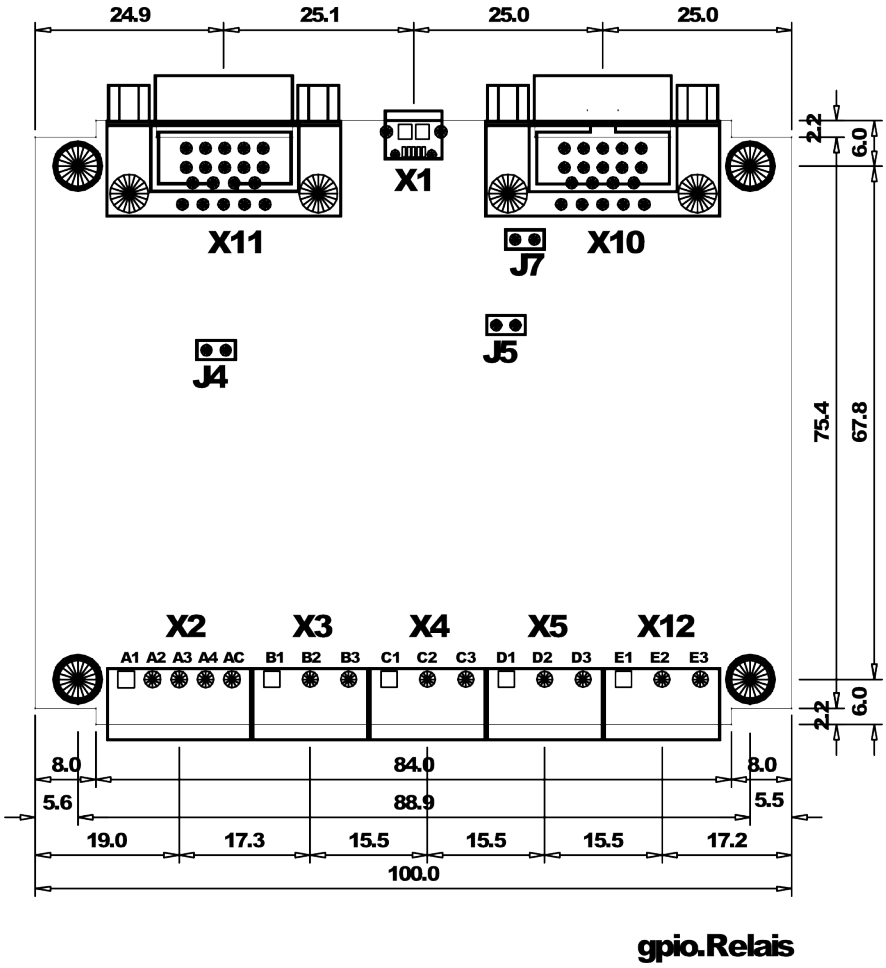
gpio.Relais
U-Series

gpio.Relais
R-Series

Overview

The gpio.Relais modules offer access to four relays. Additionally, they extend the gpio.NET *Core* interface by four digital, opto-decoupled inputs offering one puls counter as well as four frequency counters. The modules easily integrate into existing Modbus systems and can be identified and configured by a unique serial number even in a running system. Beyond that, they offer a uniform set of registers over the hole gpio.NET family.

Draft



Drawing 1: Dimensions

Interfaces

For communication, gpio.NET modules feature up to two Modbus interfaces. R-Series provide routing between those. The DSUB connectors (`X10` and `X11`) are only available in R-Series.

USB

The USB device port (`X1`) belongs to the standard equipment of gpio.NET cards. Because of this, the module is accessible from prevalent PC or laptops. To deal

with issues resulting of this, Modbus ASCII is the only available protocol used over USB.[1] A gpio.NET card is detected as a virtual serial port by the host system and so does not differ from a real one from a programs point of view.

USB device port is used as power source if connected to a host. As an alternative, PIN9 from either X10 or X11 can used instead.[2] When using PIN9 care has to be taken to use a well-regulated 5 volts source.

RS232

A serial interface with RS232 levels is available at gpio.NET module's X11 DSUB connector. Except the both data lines and GND no additional control lines are used.[3] The RS232 interface is capable of de- and encoding Modbus-RTU as well as Modbus-ASCII. The highest possible baud rate is 250k baud. By contemporaneous utilisation of USB and RS323, it is possible to control one additional card connected via RS232 using the hosts USB connection.

Pin	X11
1	
2	RXD
3	TXD
4	
5	GND
6	
7	
8	
9	VBUS

Table 1: Assignment RS232

RS485

The R-Series offer possibility to link via a RS485 bus. In that case, both potential data lines are available on X10 as well as on X11.[4] In this manner, a RS485 bus is easily realised with RS232 cables. As if using RS232, Modbus-RTU and Modbus-ASCII are both usable. The maximum bit rate is 1 Mbit. A 110 ohms terminating resistor is selectable by jumper J4.

Pin	X10	X11
1		
2		
3		
4	RS485+	RS485+
5	GND	GND
6	RS485-	RS485-
7		
8		
9	VBUS	VBUS

Table 2: Assignment RS485

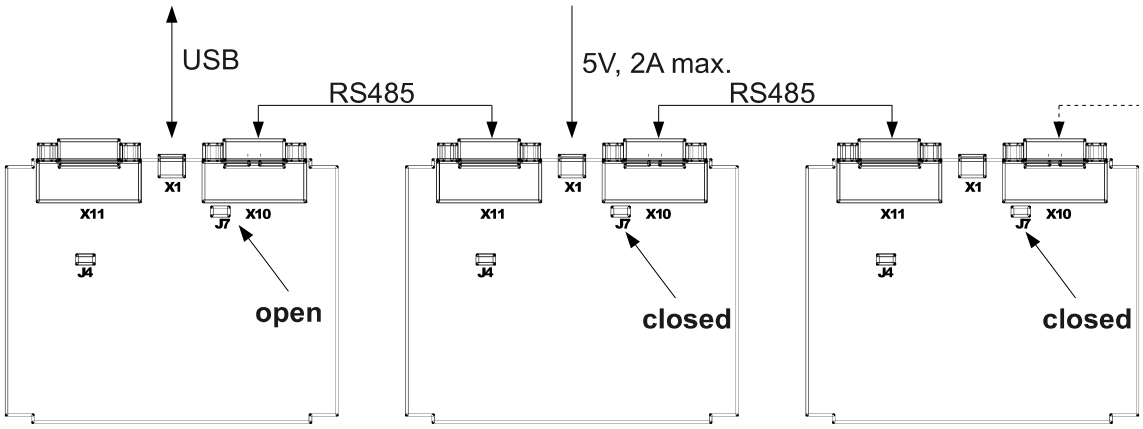
Via RS485 connected cards are routed over USB to the host, allowing access to the RS485 bus is without special PC hardware.

1 Modbus RTU – the standard mode – requires the exact temporal detection of a characters reception. This criterion cannot be guaranteed on standard PCs what causes RTU mode to become instable. Modbus ASCII is not affected by this constrain.

2 Ring indicators (PIN9) of both DSUB are connected with USB 5V power source (VBUS) (valid for X10 only with connected jumper J7). This way, power can be transmitted to the next or all gpio.NET modules in one RS485 bus system.

3 When plugging into a PC's RS232 interface, consider the RS485 assignment that is using control lines of a standard RS232 interface. These pins (DTR, DSR) should **not** be connected to a PC directly.

4 The potential data lines use standard RS232 control lines DTR and DSR. Connecting those signals to a PC directly is to be **avoided**.



Drawing 2: RS485 bus on USB with additional power supply

Individual parts of a bus constructed as shown in drawing 2 can be supplied at arbitrary points by an isolated power supply. Mind to really seperate `VBUS` of these groups. The example demonstrates this by means of the first card connected to the host PC via USB and the rest of the bus being supplied by a seperate power source. Opened jumper `J7` disrupts the `VBUS` connection on DSUB's `PIN9`.

Register description

The common set of registers simplifies working with different gpio.NET modules. Device identification, configuration of interfaces and general behaviour remain consitent throughout the family.

The gpio.Net *Core* Interface defines four sections in *Holding Register*'s address space.[5] Section `0x0000-0x00FF` contains general configuration registers – called *Core*-Register. Registers that are gpio.Relais specific occupy the *Application*-Register section `0x0100-0x0FFF` which is followed by the EEPROM. Registers `0x1000-0x107F` serve as persistent configuration used at start-up. This group collects default values of *Core*- and *Application*-Registers. Section `0x2000-0x237F` is non-volatile memory available for the user.

EEPROM sections `0x1000-0x107F` and `0x2000-0x237F` can be protected against unaware writes separately.

All registers are – as common by Modbus – 16bit wide.

5 The Modbus protocol distinguishes four address ranges – *Input registers*, *Holding registers*, *Inputs* und *Coils*. *Inputs* and *Coils* offer bitwise access to ressources. While *Holding registers* and *Coils* are read/writeable, the other two are read-only. Each type uses a 16bit address space that can overlap each other.

Holding registers

0x0000	HREG_CTRL	0x1000	reserved
0x0001	HREG_PM	0x1001	HREG_PM
0x0002	HREG_BAUDSEL	0x1002	HREG_BAUDSEL
0x0003	HREG_DBITS	0x1003	HREG_DBITS
0x0004	HREG_PARITY	0x1004	HREG_PARITY
0x0005	HREG_STOP	0x1005	HREG_STOP
0x0006	HREG_SERIAL_MODE	0x1006	HREG_SERIAL_MODE
0x0007	HREG_MODBUS_MODE	0x1007	HREG_MODBUS_MODE
...	reserved	...	reserved
0x0100	OUT_VAL	0x1012	CNT_CTRL
0x0101	IN_VAL	0x1013	PULSE_WIDTH
0x0102	CNT_CTRL	0x1014	PULSE_CH
0x0103	PULSE_WIDTH	...	reserved
0x0104	PULSE_CH		

Table 3: Holding registers

HREG_CTRL

15	14	13	12	11	10	9	8
CTRL_ACCESS_KEY (0x8E)							
7	6	5	4	3	2	1	0
reserved			CONN_ INIT	USER_ WREN	CFG_ WREN	DEFAULT	RESET

CTRL_ACCESS_KEY

Any changes to `HREG_CTRL` take only effect when the correct access key is written to the upper eight bits each time `HREG_CTRL` is accessed. This key (`0x8E`) needs to be set every time this register is about to be written to.

CONN_INIT

A logic `1` initialises the serial connection (RS232/RS485) according to the related holding registers.

USER_WREN

Enables write access to the user EEPROM section.

CFG_WREN

Enables write access to the persitent configuration section.

DEFAULT

Restores default factory settings.

HREG_PM

This register is currently unused.

HREG_BAUDSEL

Specifies the baud rate that is used on RS232/RS485. The register contains one of those selectors described below.

Selector	Baud rate	Selector	Baud rate
0	1200	6	57600
1	2400	7	115200
2	4800	8	230400
3	9600	9	250000
4	19200	10	500000
5	38400	11	1000000

HREG_DBITS

Contains the number of data bits used on serial line – selectable between seven and eight bits.

HREG_PARITY

Configures which type of parity checking is used. Valid selectors are listed below.

Selector	Parity
0	none
1	odd
2	even

HREG_STOP

The number of additional stopbits – default is 0.

SERIAL_MODE

Selects the mode the serial line operates on. Choose 0 for RS232 or 1 for RS485.

MODBUS_MODE

To switch between MODBUS RTU and ASCII modes this register is used. Changes will not affect the USB connection that always operates in MODBUS ASCII. Choose 0 for ASCII or 1 for RTU.

OUT_VAL

15	14	13	12	11	10	9	8
reserved							
7	6	5	4	3	2	1	0
E_{state}	D_{state}	C_{state}	B_{state}	reserved			

Defines the state of relays B (X3), C (X4), D (X5), E (X12). This register can also be accessed bitwise via write coils command.

IN_VAL

15	14	13	12	11	10	9	8
reserved							
7	6	5	4	3	2	1	0
E_{state}	D_{state}	C_{state}	B_{state}	A4	A3	A2	A1

This register contains the input value of inputs A1 to A4 (X2) as well as the actual state of the four relays. If the state of a relay differs from the desired value in `OUT_VAL` this indicates the corresponding relay to be malicious.

CNT_CTRL

15	14	13	12	11	10	9	8
reserved							
7	6	5	4	3	2	1	0
reserved							START

Setting `START` to `1` enables frequency measurement on all four inputs of X2. Detected frequencies can be read from the correstponding input registers which are updated every second.

PULSE_WIDTH

The minimal width for pulse counting is specified in µs . It describes the minimum duration of an event that occurs on one of the four inputs. Any noise that appears during this time span is supressed and not counted. Valid configurations for `PULSE_WIDTH` are 32 µs to 65534 µs.

PULSE_CH

This register selects the input on which events are counted. Setting a value other than `0` will start the impuls counter, immedeately.

Channel	Events counted on
0	none
1	A1
2	A2
3	A3
4	A4

Read-only input registers

0x0000	FREQ_CH1	0x0004	COUNTER_H
0x0001	FREQ_CH2	0x0005	COUNTER_L
0x0002	FREQ_CH3		
0x0003	FREQ_CH4		

Table 4: Input registers

FREQ_CH1/2/3/4

These input registers hold frequencies detected on A1/2/3/4. Data is updated secondly as long as `START` in holding register `CNT_CTRL` is set.

COUNTER_H/L

This pair of input registers stores the number of events counted by the impuls counter that occurred since enabling the selected channel in `PULSE_CH`. The counter value is calculates as:

$$n_{event} = COUNTER\_H * 65536 + COUNTER\_L$$

Changing the selected channel will reset the counter.

Bit-wise inputs (Inputs)

0x0000	A1	0x0004	B_{STATE}
0x0001	A2	0x0005	C_{STATE}
0x0002	A3	0x0006	D_{STATE}
0x0003	A4	0x0007	E_{STATE}

Table 5: Bit-wise inputs

The bit-wise inputs are directly mapped to holding register `IN_VAL`.

Bit-wise outputs (Coils)

	0x0004	B_{STATE}
	0x0005	C_{STATE}
	0x0006	D_{STATE}
	0x0007	E_{STATE}

Table 6: Bit-wise outputs (Coils)

Coils represent the bits in holding register `OUT_VAL`.

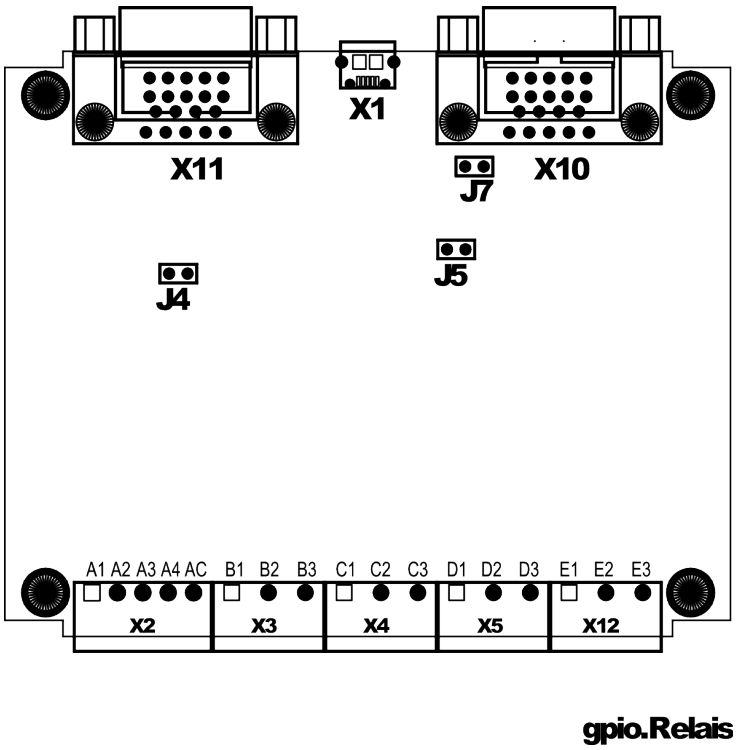
Connector description

Communication:

X1: Micro-USB Device

DSub-9 (R-Version only)

Pin	X11 (female)	X12 (male)
1	Future Appl.	Future Appl.
2	TXD	
3	RXD	
4	RS485+	RS485+
5	GND	GND
6	RS485-	RS485-
7	(RTS)	
8	(CTS)	
9	VBUS	VBUS



Opto-decoupled Inputs:

A1	+5V / 8mA
A2	+5V / 8mA
A3	+5V / 8mA
A4	+5V / 8mA
AC	Common (-)

Relay pinning:

Relay B

B1	Close
B2	Common
B3	Open

Relay C

C1	Close
C2	Common
C3	Open

Relay D

D1	Close
D2	Common
D3	Open

Relay E

E1	Close
E2	Common
E3	Open

Technical data

Operating voltage via USB, VBUS: 4,5 Volt – 5,5 Volt

Maximal current draw: ca. 200mA*

*) all Relays active, all Inputs active

Input voltage/current of opto-decoupled inputs:

Parameter	Min.	Typ.	Max.	Unit
low/passive	-5	0	2	Volt
low/passive	-	0	2.5	mA
high/active	2.5	5	28	Volt
high/active	3	8	60	mA

Limitations of relay contacting:

Maximal continuous current	6A
Maximal switch-on current	15A
Nominal voltage	250 V AC
Maximal contact voltage	400 V AC
Maximal switching power AC1	1.500 VA
Maximal switching power AC15	250 VA
1-phasen motor load, AC3-usage (230V AC)	0,185 W
Maximal switched current DC1: 30/110/220V	3/0,35/0,2A
Minimal switching load	500mW (10V / 5mA)
Contact material	AgCdO

Caution! When operating the relay contacts with voltages higher than 42V (AC/DC) strongly obey not to touch the board's surface on the bottom side. Pin voltages there can be health damaging.

Voltages larger than 100V (AC/DC) imply mortal danger!

Achtung! Wenn über die Relaiskontakte Spannungen von mehr als 42V (AC/DC) geschaltet werden, muss streng darauf geachtet werden, dass die Baugruppen-Unterseite ("Lötseite") nicht berührt werden kann, da solche Spannungen bereits gesundheitsgefährdend sein können.

Bei Spannungen über 100V (AC/DC) besteht Lebensgefahr!

Compliance declaration

The **taskit GmbH** confirms EMV conformity to the EU directive for products **gpio.Relais-R** and **gpio.Relais-U** (2004/108/EG).

The **taskit GmbH** declares the products **gpio.Relais-R** and **gpio.Relais-U** to be manufactured with respect to EU directive 2002/95/EG (RoHS) and to be free of the listed substances:

1. Lead [Pb] and its compounds
2. Mercury (quicksilver) [Hg] and its compounds
3. Polybrominated biphenyles (PBB)
4. Polybrominated diphenyl ether (PBDE)
5. Chrom-VI compounds

taskit GmbH, Groß-Berliner Damm 37, 12847 Berlin, Germany
Tel.:+49(0)30 6112950, Fax: +49(0)30 61129510