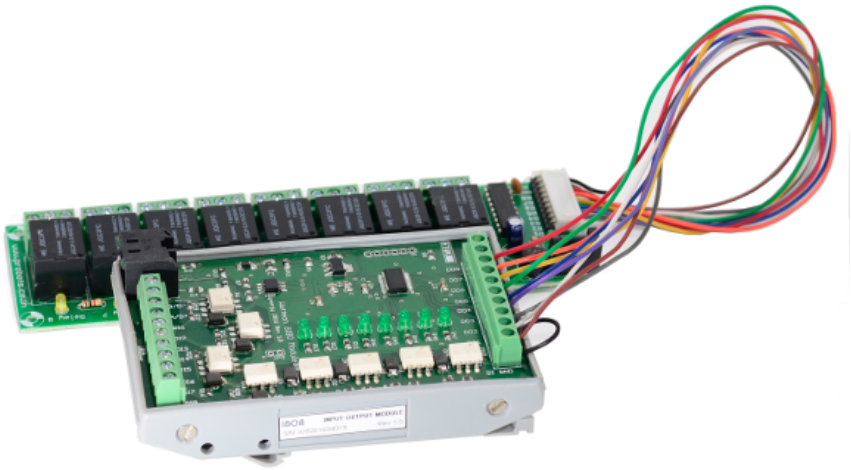
I8O8 Datasheet

8-CHANNEL DIGITAL I/O MODULE

FEATURES

- 8 Digital Inputs
- 8 Digital Outputs
- Optional Relay Module
- RS-485 MODBUS RTU Compatible



DESCRIPTION

The I8O8 module is ideal for controlling low power loads and collecting digital data from switches or float sensors. This module can power up to 10A at 250V AC through each relay output. Up to 8 digital inpus can be connected and can detect input voltages of 3 to 25V DC with an LED indication of the status. The module can be controlled by any Modbus-compliant server over RS-485.

CONNECTOR DETAILS

Pin	Name	Description
1	B/D-	RS-485 B
2	A/D+	RS-485 A
3	DI1	Digital Input 1
4	DI2	Digital Input 2
5	DI3	Digital Input 3
6	DI4	Digital Input 4
7	DI5	Digital Input 5
8	DI6	Digital Input 6
9	DI7	Digital Input 7
10	DI8	Digital Input 8

Table 1: Bottom side connector pinouts

Pin	Name	Description
1	DO8	Digital Output 8
2	DO7	Digital Output 7
3	DO6	Digital Output 6
4	DO5	Digital Output 5
5	DO4	Digital Output 4
6	DO3	Digital Output 3
7	DO2	Digital Output 2
8	DO1	Digital Output 1
9	GND	GND connector to relay board
10	DIGND	Digital input GND (connect this to GND if common)

Table 2: Top side connector pinouts (left to right)

MODBUS INTERFACE

This module supports Modbus RTU over RS-485 and operates at fixed settings as follows:

Baud rate: 9600 bps
Parity: None
Stop bits: 1

The default slave ID of this module is set to 1 but can be changed via Modbus register 10000.

The two RJ45 sockets are connected in parallel, and are used for providing power and interfacing with the RS-485 bus. Devices can be daisy chained together easily using this method with standard Ethernet patch cables. When used in conjuction with the Wattmon controller any standard Ethernet patch cable (straight through) can be used to connect with the Wattmon master. One connector has two LED indicators. The left LED indicates that the module is powered, and the right LED blinks whenever a packet is transmitted. The second LED will start blinking slowly if Modbus communication stops for over 30 seconds.

PHYSICAL CONNECTIVITY

When using this device with a third party Modbus master you will need to cut one end of a patch cable and wire it up as described in the table below. You will need to provide 5V DC to power the device through the cable.

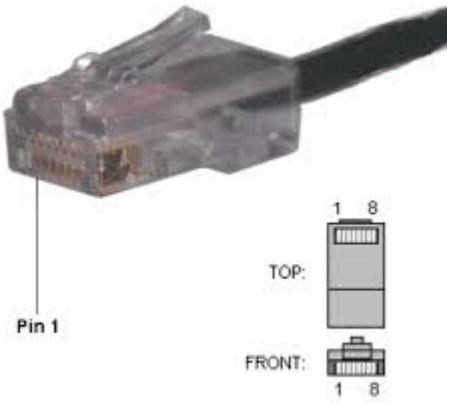


Figure 1: RJ-45 Pinouts

PIN	Description
1,2	GND (white-orange, orange)
3	Not connected
4	A (D-) (blue)
5	B (D+) (white-blue)
6	Not connected
7,8	5V DC (white-brown, brown)

Table 3: RJ-45 Connector pinout

MODBUS LOGICAL LAYER

Modbus Packet Structure

Every Modbus packet consists of four fields:
- Slave address field
- Function field
- Data field
- Error check field (checksum)

NOTE

- The values shown in the packets are in hexadecimal format.
- In the tables that show the packet structure, white background denotes the DATA field of the packet.

Address	Function Code	Data	Checksum

Table 4: Modbus Packet structure

Slave address field

The slave address field of a Modbus packet is one byte in length and uniquely identifies the slave device involved in the transaction. Valid addresses range between 1 and 255.

A slave device performs the command specified in the packet when it receives a request packet with the slave address field matching its own address.

A response packet generated by the slave has the same value in the slave address field.

Function field

The function field of a Modbus request packet is one byte in length and tells the addressed slave which function to perform. Similarly, the function field of a response packet tells the master what function the addressed slave has just performed.

Data field

The data field of a Modbus request is of variable length, and depends on the function. This field contains information required by the slave device to perform the command specified in a request packet or data being passed back by the slave device in a response packet.
Data in this field is contained in 16-bit registers. Registers are transmitted in the order of high-order byte first, low-order byte second.

Example:

A 16-bit register contains the value 0x12AB. This register is transmitted:
• High order byte = 0x12
• Low order byte = 0xAB

This register is transmitted in the order 12 AB.

Error check field (checksum)

The checksum field lets the receiving device determine if a packet is corrupted with transmission errors. In Modbus RTU mode, a 16-bit Cyclic Redundancy Check (CRC-16) is used.
The sending device calculates a 16-bit value, based on every byte in the packet, using the CRC-16 algorithm. The calculated value is inserted in the error check field.
The receiving device performs the calculation, without the error check field, on the entire packet it receives. The resulting value is compared to the error check field. Transmission errors are indicated when the calculated checksum does not equal the checksum stored in the incoming packet. The receiving device ignores a bad packet.

Modbus Functions

This device supports the following functions:

FUNCTION 1 - READ COILS
FUNCTION 2 - READ DISCRETE INPUTS
FUNCTION 3 - READ HOLDING REGISTERS
FUNCTION 4 - READ INPUT REGISTERS
FUNCTION 5 – WRITE SINGLE COIL
FUNCTION 6 - WRITE SINGLE REGISTER
FUNCTION 15 – WRITE MULTIPLE COILS
FUNCTION 16 - WRITE MULTIPLE REGISTERS
FUNCTION 17 - REPORT SLAVE ID

Function 01 : Read Coils

To read the status of the digital outputs the master must use the Read Coils request packet.

MODBUS COILS MAP

Address (Decimal)	Register Name	Description
0	DO 1	Digital Output 1
1	DO 2	Digital Output 2
2	DO 3	Digital Output 3
3	DO 4	Digital Output 4
4	DO 5	Digital Output 5
5	DO 6	Digital Output 6
6	DO 7	Digital Output 7
7	DO 8	Digital Output 8

Table 5: Digital Outputs

Function 02 : Read Discrete Inputs

To read the status of the digital inputs the master must use the Read Discrete Inputs request packet.

DISCRETE INPUTS MAP

Address (Decimal)	Register Name	Description
0	DI 1	Digital Input 1
1	DI 2	Digital Input 2

2	DI 3	Digital Input 3
3	DI 4	Digital Input 4
4	DI 5	Digital Input 5
5	DI 6	Digital Input 6
6	DI 7	Digital Input 7
7	DI 8	Digital Input 8

Table 6: Digital Inputs

Function 03: Read Holding Registers

To read the calibration parameter values, a master must send the slave device a Read Holding Registers request packet.
The Read Holding Registers request packet specifies a start register and a number of registers to read. (You can read 1 or more registers.) The start register may be from 0 to 65535 (0xFFFF).

MODBUS HOLDING REGISTER MAP

Address (Decimal)	**Name**	**Description**
4000	Restore State	Set this value to 1 to retain the state of digital outputs after a power up. Settigns this to 0 (default) will reset all outputs to 0 upon power up
10000	ADDR	Slave Address. This can be set using WRITE SINGLE REGISTER (Function 6) to set the slave address and is Write Only. To read the slave address, use the REPORT SLAVE ID function.

Table 7: Holding registers

Function 04 : Read Input Registers

To read the parameter values, a master must send the slave device a Read Input Registers request packet.
The Read Input Registers request packet specifies a start register and a number of registers to read. (You can read 1 or more registers.) The start register may be from 0 to 65535 (0xFFFF).

MODBUS INPUT REGISTER MAP

Address (Decimal)	**Register Name**	**Description**
3000	Digital Outputs	Bitmask of digital outputs

3001	Digital Inputs	Bitmask of Digital inputs

Table 8: Input registers

Function 05 : Write Single Coil

To change the status of a digital outputs the master must use the Write Single Coil request packet. Refer to table 5 for the coil map.

Function 06 : Write Single Register

To change the status of a holding register the master must use the Write Single Register request packet. Refer to table 7 for the register map.

Function 15 : Write Multiple Coils

To change the status of multiple digital outputs the master must use the Write Multiple Coils request packet. Refer to table 5 for the coil map.

Function 16 : Write Multiple Registers

To change the status of multiple holding registers the master must use the Write Multiple Register request packet. Refer to table 7 for the register map.

Function 17 : Report Slave ID

To obtain the ID of a slave (and detect the presence of a modbus device at a given address) the master can use the Report Slave ID packet type.

Configuration Settings

Setting the Restore State register (4000) to 1 will ensure that even with brief power outages or when your master module resets the output state of the relays will remain unchanged.

Input Registers 3000 and 3001 provide a bitmask of the 8 digital outputs and inputs respectively (bit 0 is output/input 1 and bit 7 is output/input 8).