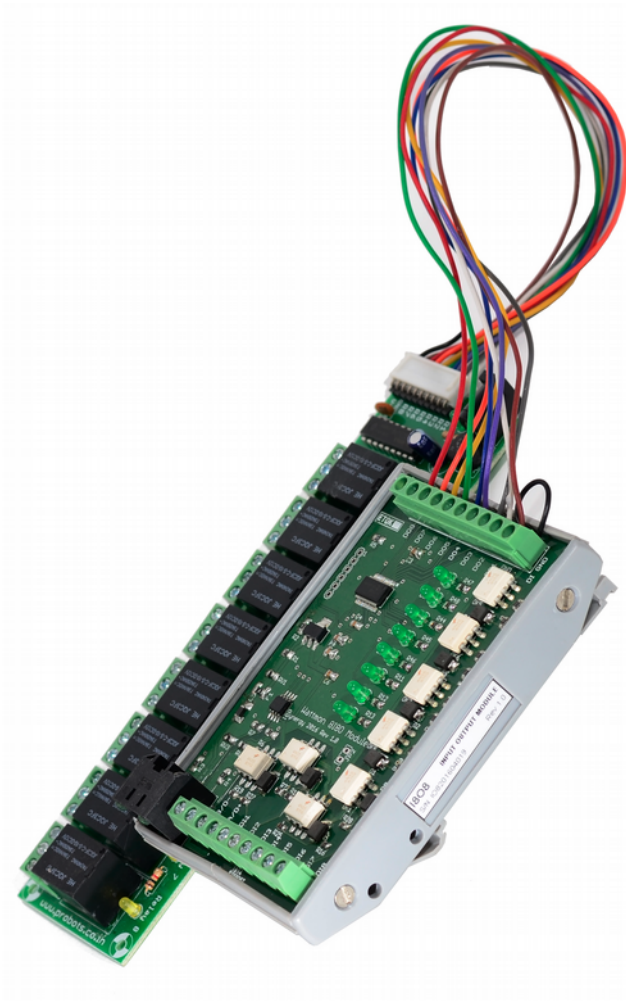




# I808

## 8-Channel Digital I/O Module + Relay Board



### Description

The I808 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 inputs can be connected and can detect input voltages ranging from 3 to 25V DC with an LED indication of the status.

It can be configured and controlled by any Modbus RTU master device over RS-485.

### Material

Base Nylon (Light Gray)

### Dimensions & Weight

Length	112 mm
Width	40 mm
Height	38 mm
Weight	123 g
Weight (with Relay)	240 g

### Bottom Side 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

### Features

- **8 Digital Inputs**  
Detect 8 individual DC Voltages with a LED Indication of Status
- **8 Digital Outputs**  
Output 8 individual DC Voltages ranging from 8-24V
- **RS-485 Modbus RTU Compatible**  
Interface with any Modbus RTU Master Device over RS-485



## I808 Module Specifications & Modbus Registers

### Top Side Connector Details

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 to GND if Common)

### Modbus Interface

The I808 module supports Modbus RTU over RS-485 and operates at fixed settings as follows:

- Baud Rate: 9600 bps
- Parity: None
- Stop Bits: 1
- Default ID: 1

While the default slave ID of this module is set to 1, it can be changed via Modbus register 10000.

There are two RJ45 sockets connected on opposite ends, which are used for providing power to the I808 module, and for interfacing with the RS-485 bus. Devices can be daisy-chained together easily using this method with standard CAT5e/CAT6 networking cables. When used in conjunction with a Wattmon Data Logger, any standard Ethernet cable (straight through) can be used to connect with the I808 to the Wattmon.

One of the RJ45 connectors has two LED indicators. The left LED indicates that the module is powered, and the right LED blinks when a data packet is transmitted. The second

LED will blink slowly if the Modbus communication stops for a period over 30 seconds.

### Physical Connectivity

When using the I808 module with a third-party Modbus master, a custom patch cable may be required. To make such a cable, cut one end of a networking cable and wire it up as described in the table below. A power supply of 5V DC will be needed to power the module through the cable.

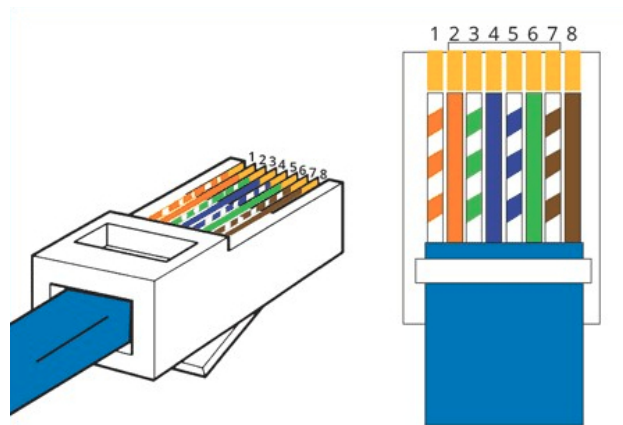


Figure 1: RJ45 Pinout

Pin	Description
1,2	GND (WHITE-orange, ORANGE-white)
3	Not Connected
4	A (D+) (BLUE-white)
5	B (D-) (WHITE-blue)
6	Not Connected
7,8	5V DC (WHITE-brown, BROWN-white)



## 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, the white background denotes the *Data* field of the packet.

Address	Function Code	Data	Checksum
---------	---------------	------	----------

#### 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 a 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 module 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, a Modbus master must use the Read Coils request packet.

### MODBUS COILS REGISTERS 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

## Function 02 : Read Discrete Inputs

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

### DISCRETE INPUTS REGISTERS 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

## Function 03 : Read Holding Register

To read the calibration parameter values, a Modbus master must send the slave device (I808) a Read Holding Registers request packet.



## I808 Module Specifications & Modbus Registers

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 REGISTERS MAP

Address (Decimal)	Register Name	Description
4000	Restore State	Set this value to 1 to retain the state of digital outputs after a power up. Setting this to 0 (default) will reset all outputs to 0 upon power up
10000	ADDR	Slave Address. This can be set using the 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.

### Function 04 : Read Input Register

To read the parameter values, a Modbus master must send the slave device (I808) 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 REGISTERS MAP

Address (Decimal)	Register Name	Description
3000	Digital Outputs	Bitmask of Digital Outputs
3001	Digital Inputs	Bitmask of Digital Inputs

### Function 05 : Write Single Coil

To change the status of a digital outputs, a Modbus 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, a Modbus 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, a Modbus 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, a Modbus master must use the Write Multiple Register request packet. Refer to table 7 for the register map.

### Function 06 : Report Slave ID

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

### Configuration Options

Setting the Restore State register (4000) to 1 will ensure that even with brief power outages or when the Modbus master device 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).

Copyright © 2018

Cynergy  
Maitreye, Auroville  
605 101 Tamil Nadu, India  
Phone: (+91) 413 262 2059  
Website: <http://www.wattmon.com>