

Configuration manual

Features

- Configurable device address (1 ~ 255) using 8 bit DIP switch, multi devices can be cascaded on RS485 bus
- Configurable device baud rate (2400bps ~ 128000bps) using 4-bit DIP switch
- TVS (Transient Voltage Suppressor) protection, effectively suppress surge voltage and transient spike voltage in the circuit, lightning proof & anti-electrostatic
- Optocoupler isolated digital inputs, prevent interference from external high-voltage circuit
- Reverse-proof circuit, prevent the circuit from being damaged accidentally by incorrect connection
- DIN-Rail mounted plastic enclosure, easy to install, safe to use
- 3 LEDs for indicating the MCU status and signal transceiving status, also LED indication for power and inputs
- Watchdog Timer

Protocol Description

MODBUS Protocol is a messaging structure, widely used to establish master-slave communication between intelligent devices. A MODBUS message sent from a master to a slave contains the address of the slave, the 'command' (e.g. 'read register' or 'write register'), the data, and a check sum (LRC or CRC).

Since Modbus protocol is just a messaging structure, it is independent of the underlying physical layer. In AMB-4I-4O is implemented using RS485 interface.

Modbus RTU Framing

The AMB-4I-4O module supports standard RTU Modbus protocol for getting the input status and setting/getting the relays status. RTU Modbus Protocol frame is illustrated below:

Start	Address	Function	Data	RC Check	End
≥ 3.5 Char	4 Bits	8 Bits	N*8 Bits	16 Bits	≥ 3.5 Char

- **Start** is at least 3.5 char time of silent interval between master and slave to start communication
- **Address** is a slave ID Address
- **Function** is the function code of MODBUS protocol
- **Data** is the appropriate data for function code
- **CRC** is Cyclic Redundancy Check for Error Checking (here, CRC-16,Big Endian(ABCD))

- **End** is at least 3.5 char time of silent interval between master and slave to start communication

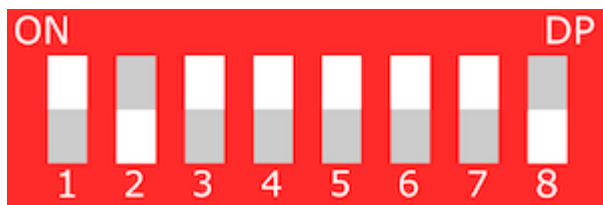
Modbus Serial Port Parameter

Parameter	Option/range
Baudrate	9600 – 128000 configurable
ata bit	8
Parity	None
Stop bit	1

ID Configuration of Modbus Card

User can set the slave ID address using 8 Bit DIP switch found on board. This allow 0 to 255 different IDs to be set. The number below the switches are added together and the result will be identifier of the device slave ID.

To set the slave ID 125, you need to turn ON switch 1,3,4,5,6,7 on DIP switch and OFF the others (In this, convert slave ID number 125 into 8bit binary form – 01111101, turn ON DIP switch 1,3,4,5,6,7 and turn OFF DIP switch 2,8) and so on....



Baudrate Configuration

User can set the baudrate using 4 Bit DIP switch found on board. To set baudrate, first turn off power supply of card. This allows to set baudrate between 2400bps to 128000bps as shown below table.

Baudrate	SW1	SW2	SW3	SW4
9600	OFF	OFF	OFF	OFF
2400	ON	OFF	OFF	OFF
4800	OFF	ON	OFF	OFF
14400	OFF	OFF	ON	OFF
19200	OFF	OFF	OFF	ON
28800	ON	ON	OFF	OFF

Baudrate	SW1	SW2	SW3	SW4
38400	OFF	ON	ON	OFF
57600	OFF	OFF	ON	ON
115200	ON	OFF	OFF	ON
128000	ON	ON	ON	ON

Modbus RTU Commands

The following commands (functions) were implemented in the Modbus RTU AMB-4I-4O.

- 01 (01 hex) Function code – Read Coils

Coils are mapped to outputs and discrete inputs are mapped to inputs. Coils read/write output status and discrete inputs read Input status. There are 4 inputs for reading-input 0 to input 3 which refer to input 1 to input 4 and coil 0 to coil 3 which refer to output 1 to output 4.

Note: For all below example, consider slave address – 0x01 will change with 8bit DIP switch.

01 (01 hex) Function Code – Read Coils

Read Coils command is requesting the ON/OFF status of coils. Read Coils command will return individual current outputs status. Also Read Coils Command will return sequentially outputs current status.

Example of how to read output status are given below:

REQUEST FRAME –

01 01 0000 0008 3DCC

01 – The Slave Address (01 hex = 01)

01 – The Function Code

0000 – The Data Address of first coil to read (0000hex = coil 0)

0008 – The total number of inputs requested (0008hex = 8 inputs)

3DCC – The CRC for error checking

RESPONSE FRAME –

01 01 01 C9 91DE

01 – The Slave Address (01 hex = 01)

01 – The Function Code

01 – The number of data bytes to follow (8 inputs / 8 bits per byte = 1 byte)

C9 – Input 0 to 7(Inputs), C9 hex = 1100 1001 (0 – ON, 1 – OFF)

Modbus Exception Codes

Exception responses from either the master or a slave can result from data processing errors. One of the following events can occur after a request from the master:

- If the slave receives the request from the master without a communication error and can handle the request correctly, it returns a normal response.
- If the slave does not receive the request from the master due to a communication error, it does not return a response. The master program eventually processes a timeout condition for the request.
- If the slave receives the request from the master but detects a communication error, it does not return a response. The master program eventually processes a timeout condition for the request.
- If the slave receives the request from the master without a communication error, but cannot handle it (for example, the request is to read a register that does not exist), the slave returns an exception response to inform the master of the nature of the error.

Exception Frame:

Sn.	Definition	Description
1	Slave Address	The Slave Address of requested slave
2	Exception Function Code	Requested Function Code + 128 (0x80)
3	Exception Code	See in next table
4	CRC check	The CRC for error checking

Exception Codes:

Exception Code	Name	Description
0X01	Illegal Function Code	The function code received in the request is not an allowable action for the slave. This may be because the function code was not implemented in the unit selected.
0x02	Illegal Data Address	The data address received in the request is not an allowable address for the slave.
0x03	Illegal Data Value	A value contained in the request data field is not an allowable value for the slave. This indicates a fault in the structure of remainder of a complex request, such as that the implied length is incorrect.

Revision #4

Created 11 March 2025 12:00:03 by Yogesh

Updated 11 March 2025 12:07:23 by Yogesh