भारते निर्मितम्

ATREYO®

SOLUTIONS THAT SIMPLIFY

AMB-8I-40

8 input 4 output expander with Modbus and IEC-60870-5-101

CONFIGURATION GUIDE

Configuration Guide

www.atreyo.in

1. Device Overview

The AMB-8I4O is a digital input/output module implementing the Modbus RTU and IEC60870-5-101 protocol. It provides:

- 8 digital inputs
- 4 digital outputs
- RS-485 communication interface
- Modbus RTU and IEC60870-5-101 protocol support

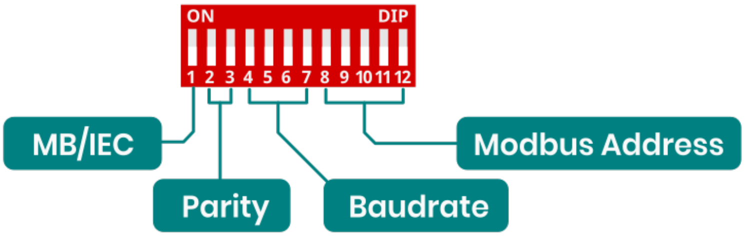
2. Basic information

AMB-8I-4O is an I/O expander operating over Modbus RTU and IEC60870-5-101 protocol. The first DIP switch is used to switch between Modbus RTU and IEC modes. If it is in position 0, then the AMB is operating in Modbus RTU mode. If it is in position 1, then it works in the IEC protocol. As the two protocols have different functions, the DIP switch is also different. In IEC mode, the last DIP switch is used to change the operating mode from balanced/unbalanced. To save the DIP switch settings in the internal memory, the SAVE button must be pressed.

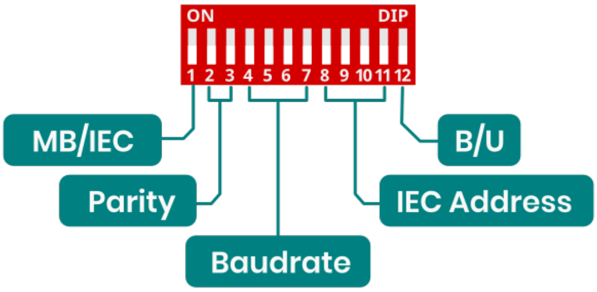
3. Configuration

3.1. DIP switch configuration

This is DIP switch functions diagram for Modbus RTU configuration. For enable Modbus RTU protocol 1 switch should be in OFF position.



This is DIP switch function diagram for IEC protocol. For enable IEC protocol 1 switch should be in ON position.



3.2. Parity configuration for both Modbus and IEC

Parity	Function
00	NONE
01	ODD

Parity	Function
10	EVEN
11	NONE

3.3. Baudrate configuration for both Modbus and IEC

Switch	Baudrate
0000	9600
0001	1200
0010	2400
0011	4800
0100	14400
0101	19200
0110	28800
0111	38400
1000	57600
1001	76800
1010	115200
1011	230400
1100	256000
1101	460800
1110	576000
1111	921600

3.4. Address configuration for Modbus RTU

Switch	Value	Modbus Address
00000	0	1
00001	1	2
00010	2	3
00011	3	4
00100	4	5
00101	5	6
00110	6	7
00111	7	10
01000	8	20
01001	9	30
01010	10	40
01011	11	50

Switch	Value	Modbus Address
01100	12	60
01101	13	70
01110	14	80
01111	15	90
10000	16	100
10001	17	110
10010	18	120
10011	19	130
10100	20	140
10101	21	150
10110	22	160
10111	23	170
11000	24	180
11001	25	190
11010	26	200
11011	27	210
11100	28	220
11101	29	230
11110	30	240
11111	31	250

3.5. Address configuration for IEC

Switch	Value	IEC Address
0000	0	1
0001	1	2
0010	2	3
0011	3	4
0100	4	5
0101	5	6
0110	6	7
0111	7	10
1000	8	20
1001	9	30
1010	10	40
1011	11	50
1100	12	60

Switch	Value	IEC Address
1101	13	70
1110	14	80
1111	15	90

4. EC60870-5-101 Implementation Guide

4.1. Protocol Overview

The device implements IEC60870-5-101 protocol with the following key functions:

4.1.1. Reading States (Interrogation)

- Function: C_IC_NA_1 (100)
- Purpose: Reading all inputs and outputs states
- IOA ranges:
  - Inputs: 1000-1007
  - Outputs: 2000-2003

4.1.2. Controlling Outputs

- Function: C_SC_NA_1 (45)
- Purpose: Setting individual output states
- IOA range: 2000-2003
- Values: ON (1) / OFF (0)

4.2. Command Structure

4.2.1. General Interrogation Command

```
Type ID: 100 (C_IC_NA_1)
Qualifier: 20
Cause of Transmission: 6 (Activation)
Common Address: 1
```

4.2.2. Single Command (Output Control)

```
Type ID: 45 (C_SC_NA_1)
Cause of Transmission: 6 (Activation)
Common Address: 1
IOA: 2000-2003
Value: 0/1
```

5. SCADA system configuration

5.1. IEC60870-5-101 Protocol Configuration

- Common Address (ASDU): 1
- Frame format: FT1.2
- Link Layer Address: according to device configuration

5.2. I/O Points Configuration:

For inputs (8 channels):

- Type: Single Point Information (M_SP_NA_1)

- IOA addresses: 1000-1007
- Direction: Monitoring (read)

For outputs (4 channels):

- Type: Single Command (C_SC_NA_1)
- IOA addresses: 2000-2003
- Direction: Control (write)

Verification:

- Check if input states are correctly read (IOA 1000-1007)
- Confirm control operation for each output (IOA 2000-2003)
- Verify device responds to General Interrogation

6. Implementation in Python

6.1. Reading Device State

```
def read_device_state():
    asdu = ASDU()
    asdu.type = 100  # C_IC_NA_1
    asdu.cause = 6   # Activation
    asdu.address = 1 # Common Address

    # Send command and wait for response
    send_command(asdu)
```

6.2. Controlling Outputs

```
def set_output(output_number, state):
    asdu = ASDU()
    asdu.type = 45   # C_SC_NA_1
    asdu.cause = 6   # Activation
    asdu.address = 1

    ioa = 2000 + output_number
    value = 1 if state else 0

    io = InformationObject(ioa, value)
    asdu.add_io(io)

    send_command(asdu)
```

7. Programming Examples

7.1. Basic Implementation

```
# Initialize connection
master = IEC101Master("/dev/ttyUSB0", 9600)
master.set_link_address(1)

# Read all states
states = master.read_all_states()

# Set output
master.set_output(0, True)  # Turn ON output 1
master.set_output(1, False) # Turn OFF output 2
```

7.2. Continuous Monitoring

```
def monitor_device():
  while True:
    states = master.read_all_states()
    process_states(states)
    time.sleep(1)
```

8. Troubleshooting

8.1. Common Issues

Communication Errors

- Verify serial port settings
- Check device address
- Confirm cable connections

Command Failures

- Verify IOA addresses
- Check command format
- Validate checksum calculations

9. Appendix

9.1. Error Codes

Code	Description	Solution
0x01	Invalid Address	Check device address
0x02	Command Timeout	Verify connection
0x03	Checksum Error	Check data integrity

For more details scan or click on QR code

Copyright

Trademarks

Atreyo and the Atreyo logo are registered trademarks of Atreyo Research and Development LLP.

All other trademarks and copyrights are the property of their respective owners.

Disclaimer

- The information contained in this guide is provided "as is" and is subject to change without notice. While every effort has been made to ensure the accuracy and completeness of the information contained herein, Atreyo Research and Development LLP assumes no responsibility or liability for any errors or omissions.
- Images used in this specification may differ in appearance from the actual product.
- All dimensions mentioned in the drawings are not to scale and are subject to change and update as may be decided by the Atreyo Research and Development LLP.
- Atreyo Research and Development LLP makes no warranties, either express or implied, regarding the contents of this document, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.
- Users of this guide should verify the applicability and suitability of the information and procedures to their particular use. Atreyo Research and Development LLP shall not be liable for any damages, including but not limited to direct, indirect, special, incidental, or consequential damages, arising from or related to the use or inability to use this guide or the products described herein.
- By using this guide, you agree to the terms and conditions stated above.

**Atreyo Research
& Development LLP**

**+91 9727741417
info@atreyo.in**

414, Sunrise Shopping Mall, Mansi
Circle, Ahmedabad, Gujarat, India