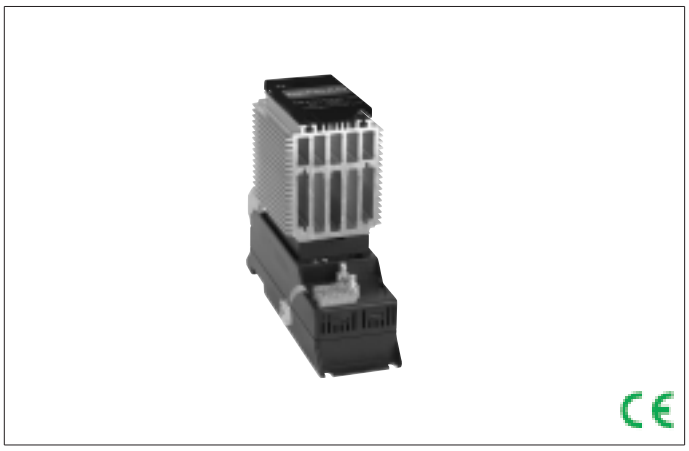
GEFRAN

# GEFLEX-ETHERNET



## INSTALLATION AND OPERATION MANUAL

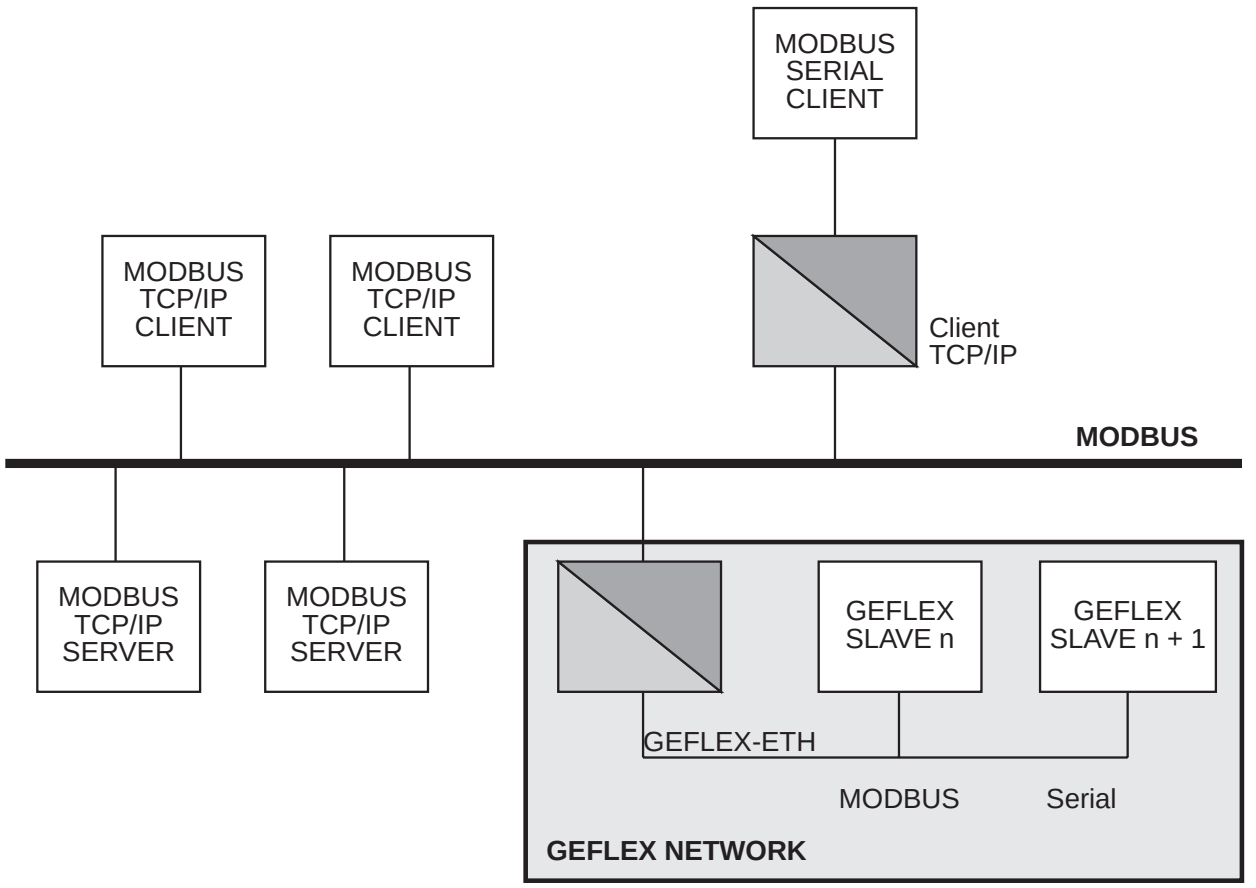
Software Version: **1.x**

Code: **80367** / Edition **01** - **0708** **ENGLISH**

## INDEX

## 1 • Introduction

The Geflex GEFLEX-ETH is IEEE 802.3 compliant supporting Ethernet operation for MODBUS TCP/IP communication architecture




## 2 • Features

- IEEE 802.3 compliant
- Full-duplex and half-duplex modes
- Capable of supporting both 10 Mbps and 100 Mbps data rates
- Single RJ45 connection with leds indicator (yellow = link , green = activity)
- One socket available
- All MODBUS/TCP ADU are sent via TCP on registered port 502
- Default address: 192.168.1.100
- Subnet mask: 255.255.255.0
- Default gateway: 192.168.1.1

## 3 • Dip-switch Function

Resetting Parameters to Factory Defaults

At any moment, all of the GTR ETH board parameters can be restored to the factory default settings. To do so:

1. Set Dip Switch S1 dip 1 = ON (located near RJ45 connector ) ;
2. Restart GTR ETH board;
3. Set Dip Switch S1 dip 1 = OFF ;

S1 dip 2 off = 100Mb (DEFAULT)
S1 dip 2 on = 10Mb

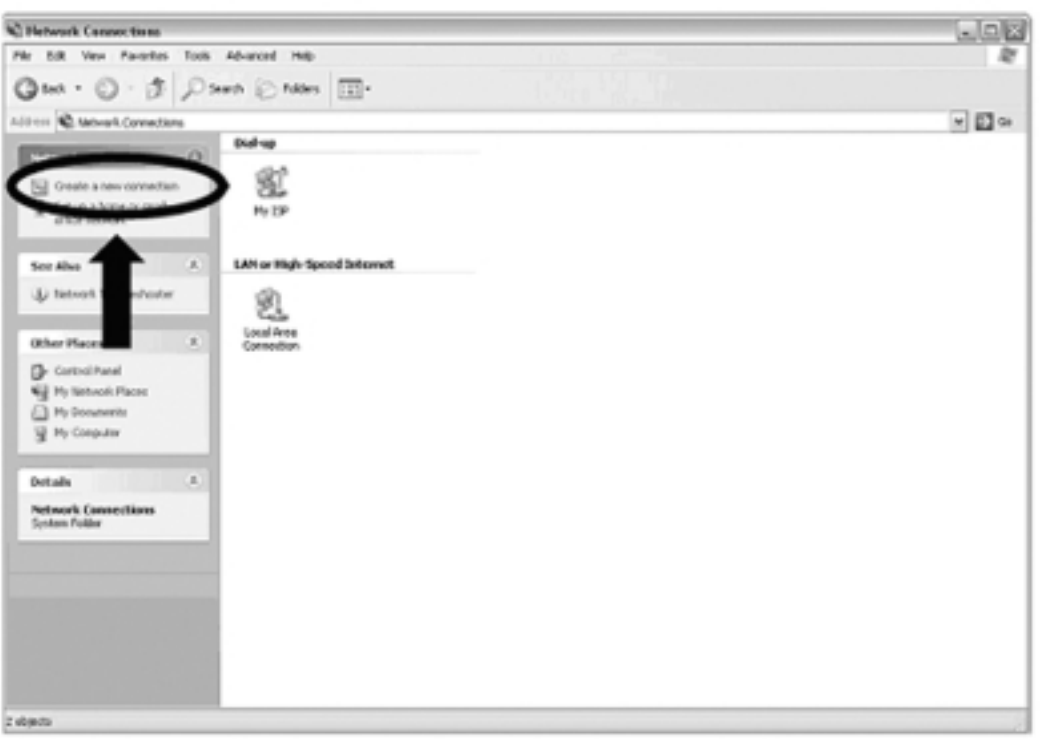
## 4 • Setup PC Connection

All the information in this quick guide require that the GEFLEX-ETH communicates with the host PC through a LAN connection.
In this pages you'll learn how to configure the PC to communicate with the GEFLEX-ETH using an Ethernet cross cable or hub/switch. The following steps are based on a Windows XP system.

**1.** Open the Windows Control Panel and open the Network. Connections item:

Windows Control Panel

**2.** If a Local Area Connection is already available, go to step 8.
Otherwise, run the "New Connection" wizard.



Network Connections

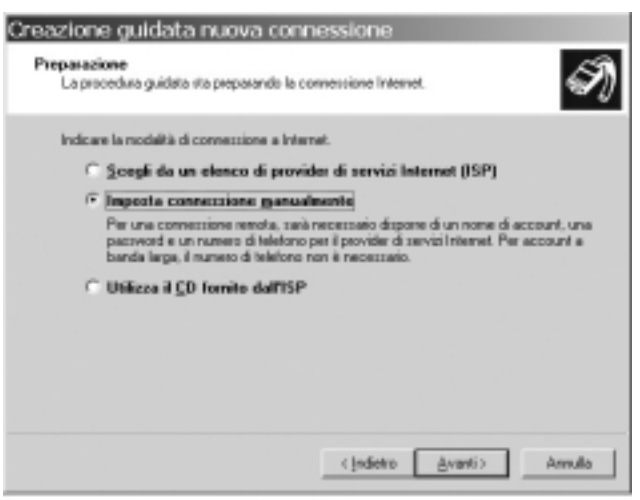
**3.** The following dialog box will appear. Click Next.


New Connection Wizard, Step 1

**4.** On the next dialog box, select the “Connect to the Internet” option and click the “Next” button.

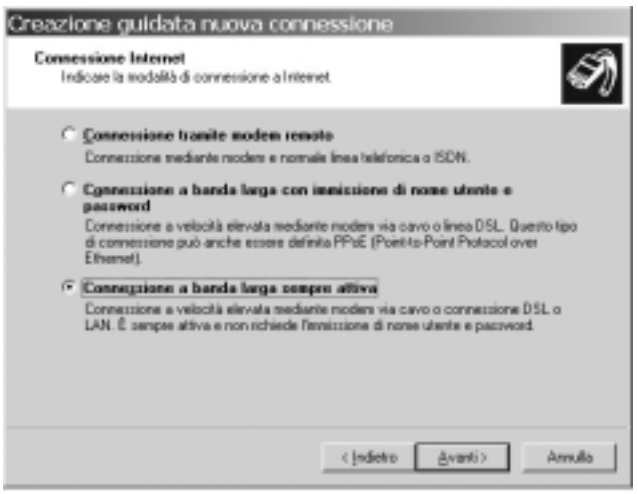

New Connection Wizard, Step 2

**5.** On the next dialog box, select the “Set up my connection manually option and click the “Next” button.




**New Connection Wizard,Step 3**

**6.** On the next dialog box, select the “Connect using  a broadband connection that is always on” option and click the “Next” button.




**New Connection Wizard,Step 4**

**7.** When the last dialog box appears, click the “Finish” button.


**New Connection Wizard,Step 5**

**8.** Open the newly created connection (or the existing connection if you are coming from step 2) and click the “Properties” button.


**Local Area Connection Status**

**9.** Select the “Internet Protocol (TCP/IP)” item and click the “Properties” button.


**Local Area Connection Status**

**10.** If you have started from an existing connection, write down all of the TCP/IP parameters. You’ll need them later to restore your LAN settings to the original parameters.

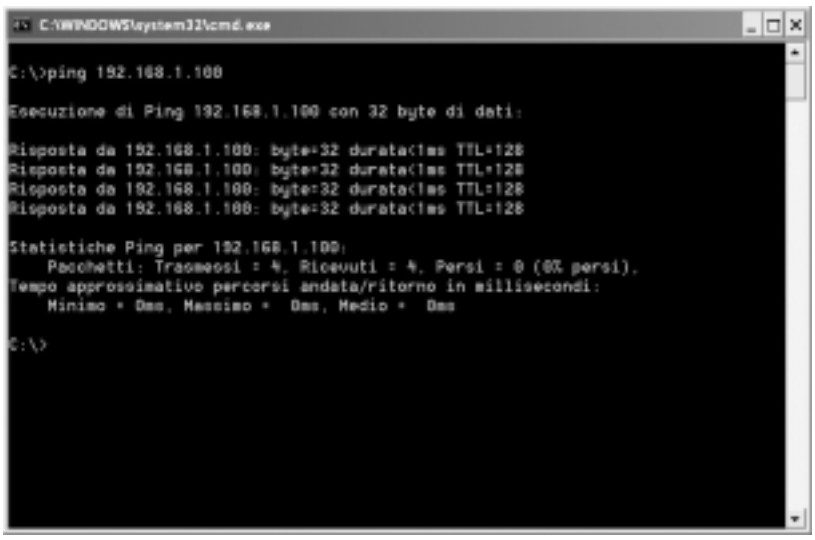
**11.** Select the manual settings and type in the following values:
IP address: 192.168.1.101
Subnet mask: 255.255.255.0


**TCP/IP properties**

Click the “OK” button on all LAN setup dialog boxes.

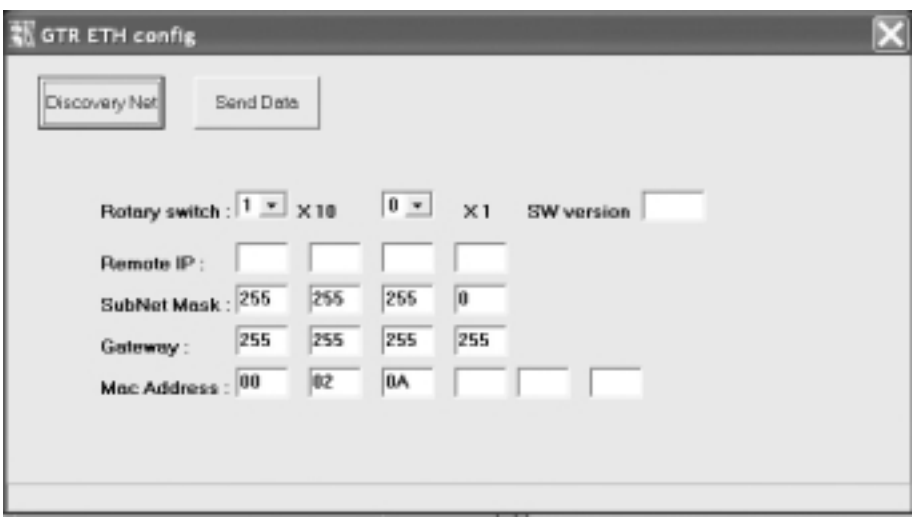
## 5 • Test Board Connection Using PING Command

1. Power up the board and verify the POWER led turn on.
2. Connect the GEFLEX-ETH to the PC through an Ethernet cross cable or hub / switch.
3. Open a Command Prompt window (on Windows XP, from the Windows Start menu, select Run, enter CMD and click the “OK” button).
4. At the command prompt, type: PING 192.168.1.100
5. If the connection has been properly setup, the PING command will return a positive feedback.




## 6 • Connect the Board to Your LAN

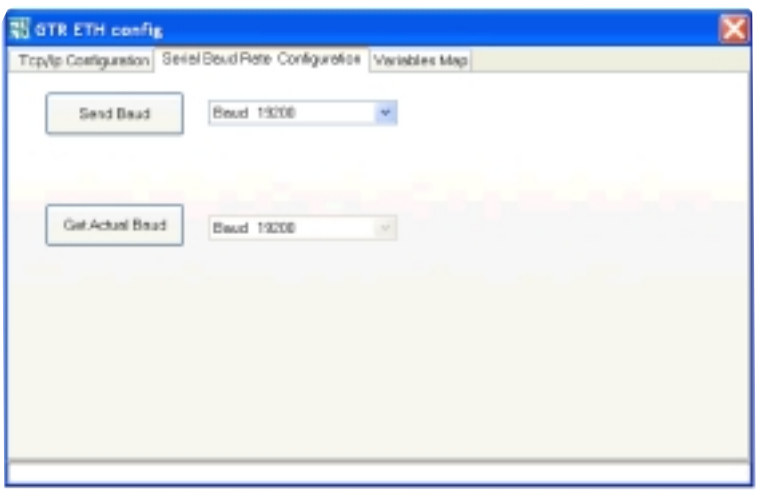
Before connecting the GEFLEX-ETH board to your LAN, the IP address must be changed to match the LAN settings.
To change the GEFLEX-ETH’s IP address, do the following.




1. With the GEFLEX-ETH still connected to the PC via the Ethernet cross cable or hub /switch, launch the GtrEthConfig.exe utility that is located in the CD-ROM.
2. Set the Rotary Switch combobox to match your actual GEFLEX-ETH’s Rotary value.
3. Press the **Discovery Net** button. You can see actual IP address , subnet mask , default gateway and MAC adress.
4. Set your parameter and after click on **Send Data** button.
5. Restart the GEFLEX-ETH Board.
6. Goto to step 2 if you want to see new parameter.

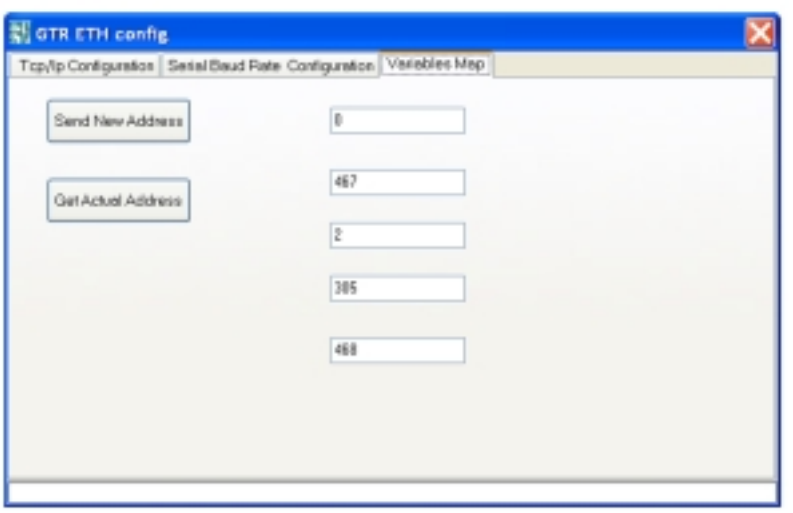
## 7 • Getting Serial baud Information

Using this option to get serial baud information from serial network. Send Baud has no effect




## 8 • Getting Process Variables Map Information

Using this option to get / set Process Variables  map



## 9 • GEFLEX-ETH's Modbus TCP/IP implemented command

- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs

**PDU Example**: Read bit 0 from zone 10

| Request | | Starting Address | | Quantity of Inputs | |
|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo |
| 10 | 2 | 0 | 0 | 0 | 1 |

Resp.

| Rotary switch | Function code | Byte count | Inputs |
|---|---|---|---|
| 10 | 2 | 1 | 0 |

- 03 (0x03) Read Holding Registers / 04 (0x04) Read Input Registers

**PDU Example**: Read word 138 from zone 10

| Request | | Starting Address | | Quantity of words | |
|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo |
| 10 | 2 | 0 | 138 | 0 | 1 |

| Resp. | | | Data Value = 400 (hex 190) | |
|---|---|---|---|---|
| Rotary switch | Function code | Byte count | Hi | Lo |
| 10 | 4 | 2 | 1 | 144 |

- 05 (0x05) Write Single Coil

**PDU Example**: Writing to zone 10 single bit address = 1 data value =0xff00

| Request | | Starting Address | | Output Value | |
|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo |
| 10 | 5 | 0 | 1 | 0xFF | 0 |

| Resp. | | Starting Address | | Output Value | |
|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo |
| 10 | 5 | 0 | 1 | 0xFF | 0 |

- 06 (0x06) Write Single Register

**PDU Example**: Writing to zone 10 single word address = 400 data value = 1

| Request | | Word Address | | Word Value | |
|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo |
| 10 | 6 | 1 | 144 | 0 | 1 |

| Resp. | | Word Address | | Output Value | |
|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo |
| 10 | 6 | 1 | 144 | 0 | 1 |

- 15 (0x0F) Write Multiple Coils

**PDU Example**: Writing to zone 10 Multiple Bits address = 10, quantity of bits =2, bits value=3

| Request | | Starting Address | | Quantity of bits | | Byte count | Bits Value |
|---|---|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo | | |
| 10 | 15 | 0 | 10 | 0 | 2 | 1 | 3 |

| Resp. | | Starting Address | | Quantity of bits | |
|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo |
| 10 | 15 | 0 | 10 | 0 | 2 |

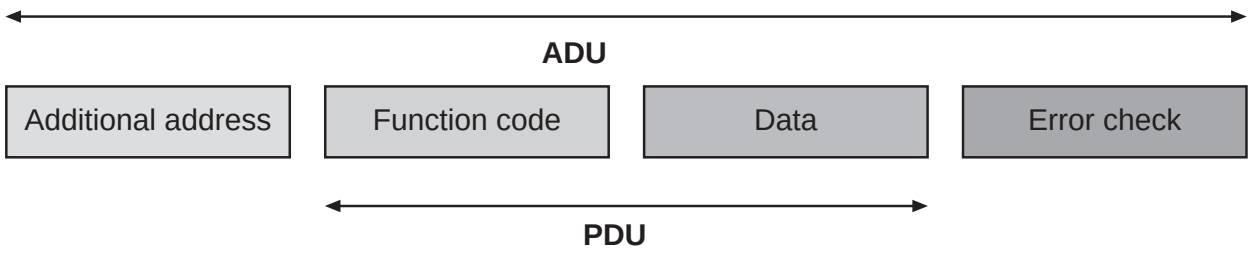
- 16 (0x10) Write Multiple registers

**PDU Example**:
Writing to zone 6 Multiple Words address = 20, quantity of words =2, word 1 value =0 , word 2 value =1000

| Request | | Starting Address | | Quantity of words | | Byte count | Word 1 Value | | Word 2 Value | |
|---|---|---|---|---|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo | | Hi | Lo | Hi | Lo |
| 10 | 16 | 0 | 20 | 0 | 2 | 4 | 0 | 0 | 3 | 232 |

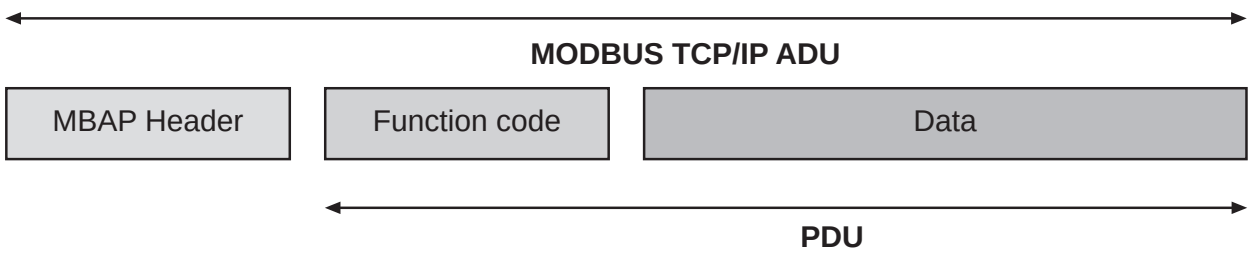
| Resp. | | Starting Address | | Quantity of Words | |
|---|---|---|---|---|---|
| Rotary switch | Function code | Hi | Lo | Hi | Lo |
| 10 | 16 | 0 | 20 | 0 | 2 |

The MODBUS protocol defines a simple Protocol Data Unit (PDU) independent of the underlying communication layers. The mapping of MODBUS protocol on specific buses or networks can introduce some additional fields on the Application Data Unit (ADU).




**General Modbus Frame**

The client that initiates a MODBUS transaction builds the MODBUS Application Data Unit.
The function code indicates to the server which kind of action to perform.




**MODBUS request/response over TCP/IP**

A dedicated header is used on TCP/IP to identify the MODBUS Application Data Unit. It is called the MBAP header (MODBUS Application Protocol header).
This header provides some differences compared to the MODBUS RTU application data unit used on serial line: the MODBUS 'slave address' field usually used on MODBUS Serial Line is replaced by a single byte 'Unit Identifier' within the MBAP Header.
The 'Unit Identifier' is used to communicate via devices such as bridges, routers and gateways that use a single IP address to support multiple independent MODBUS end units.
All MODBUS requests and responses are designed in such a way that the recipient can verify that a message is finished.
For function codes where the MODBUS PDU has a fixed length, the function code alone is sufficient.
For function codes carrying a variable amount of data in the request or response, the data field includes a byte count.
When MODBUS is carried over TCP, additional length information is carried in the MBAP header to allow the recipient to recognize message boundaries even if the message has been split into multiple packets for transmission.
The existence of explicit and implicit length rules, and use of a CRC-32 error check code (on Ethernet) results in an infinitesimal chance of undetected corruption to a request or response message.

The MBAP Header contains the following fields:

| Fields | Length | Description | Client | Server |
|---|---|---|---|---|
| **Transaction Identifier** | 2 Bytes | Identification of a MODBUS Request / Response transaction | Initialized by the client | Recopied by the server from the received request |
| **Protocol Identifier** | 2 Bytes | 0 = MODBUS protocol | Initialized by the client | Recopied by the server from the received request |
| **Length** | 2 Bytes | Number of following bytes | Initialized by the client (request) | Initialized by the server (Response) |
| **Unit Identifier** | 1 Byte | Identification of a remote slave connected on a serial line or on other buses | Initialized by the client | Recopied by the server from the received request |

The header is 7 bytes long:

1. **Transaction Identifier** - It is used for transaction pairingthe MODBUS server copies in the response the transaction identifier of the request.
2. **Protocol Identifier** - It is used for intra-system multiplexing. The MODBUS protocol is identified by the value 0.
3. **Length** - The length field is a byte count of the following fields, including the Unit Identifier and data fields.
4. **Unit Identifier (Rotary switch value see Geflex Software Manual)** - This field is used for intra-system routing purpose. It is typically used to communicate to a MODBUS serial line slave through a gateway between an Ethernet TCP-IP network and a MODBUS serial line. This field is set by the MODBUS Client in the request and must be returned with the same value in the response by the server.

## 10 • MEMORY MAP

| Address hex | Description | R/W | Default Value | Note | Access |
|---|---|---|---|---|---|
| 8000 | Device ID | R | (3) | - | Single word |
| 8001 | SW Version | R | - | | Single word |
| 8002 | Mac Id , bytes 0-1 | R | 0002 hex | Gefran OUI Byte 0-1 | Single word |
| 8003 | Mac Id , bytes 2-3 | R | 0A-- hex | Gefran OUI Byte 2 , xx byte 3 | Single word |
| 8004 | Mac Id , bytes 4-5 | R | ---- | Gefran internal | Single word |
| 8005 | IP bytes 0-1 | R/W | C0A8 hex192.168dotted | LAN | Single word |
| 8006 | IP bytes 2-3 | R/W | 0164 hex 1.100 dotted | LAN | Single word |
| 8007 | SubNet Mask bytes 0-1 | R/W | FFFF hex 255.255 dotted | Class C | Single word |
| 8008 | SubNet Mask bytes 2-3 | R/W | FF00 hex 255.0 dotted | Class C | Single word |
| 8009 | Default gateway bytes 0-1 | R/W | C0A8 hex 192.168 dotted | | Single word |
| 800a | Default gateway bytes 2-3 | R/W | 0101 hex 1.1 dotted | | Single word |
| 800b | Slave Online bytes 0-1 | R | Bit field | (1) | Single word |
| 800c | Serial baud | R | 4 (19200 baud) | | Single word |
| 800d | Modbus Address Map for var Process1 | R/W | 0 PV | Geflex PV | Single word |
| 800e | Modbus Address Map for var Process 2 | R/W | 01d3 hex | Geflex STATUS_STRUMENTO | Single word |
| 800f | Modbus Address Map for var Process 3 | R/W | 0002 hex Ou.P | Geflex Control output value | Single word |
| 8010 | Modbus Address Map for var Process 4 | R/W | 0131 hex | Geflex STATUS_W | Single word |
| 8011 | Modbus Address Map for var Process 5 | R/W | 01d4 hex I.1on | Geflex VALAUX_ON | Single word |
| 8012 | Quick Buffer for var process 1 | R | - | Zone 1 | Multiple word access |
| 8013 | Quick Buffer for var process 2 | R | - | Zone 1 | Multiple word access |
| 8014 | Quick Buffer for var process 3 | R | - | Zone 1 | Multiple word access |
| 8015 | Quick Buffer for var process 4 | R | - | Zone 1 | Multiple word access |
| 8016 | Quick Buffer for var process 5 | R | - | Zone 1 | Multiple word access |
| 8017 | Quick Buffer for var process 1 | R | - | Zone 2 | Multiple word access |
| 8018 | Quick Buffer for var process 2 | R | - | Zone 2 | Multiple word access |
| 8019 | Quick Buffer for var process 3 | R | - | Zone 2 | Multiple word access |
| 801a | Quick Buffer for var process 4 | R | - | Zone 2 | Multiple word access |
| 801b | Quick Buffer for var process 5 | R | - | Zone 2 | Multiple word access |

| Address hex | Description | R/W | Default Value | Note | Access |
|---|---|---|---|---|---|
| 801c | Quick Buffer for var process 1 | R | - | Zone 3 | Multiple word access |
| 801d | Quick Buffer for var process 2 | R | - | Zone 3 | Multiple word access |
| 801e | Quick Buffer for var process 3 | R | - | Zone 3 | Multiple word access |
| 801f | Quick Buffer for var process 4 | R | - | Zone 3 | Multiple word access |
| 8020 | Quick Buffer for var process 5 | R | - | Zone 3 | Multiple word access |
| 8021 | Quick Buffer for var process 1 | R | - | Zone 4 | Multiple word access |
| 8022 | Quick Buffer for var process 2 | R | - | Zone 4 | Multiple word access |
| 8023 | Quick Buffer for var process 3 | R | - | Zone 4 | Multiple word access |
| 8024 | Quick Buffer for var process 4 | R | - | Zone 4 | Multiple word access |
| 8025 | Quick Buffer for var process 5 | R | - | Zone 4 | Multiple word access |
| 8026 | Quick Buffer for var process 1 | R | - | Zone 5 | Multiple word access |
| 8027 | Quick Buffer for var process 2 | R | - | Zone 5 | Multiple word access |
| 8028 | Quick Buffer for var process 3 | R | - | Zone 5 | Multiple word access |
| 8029 | Quick Buffer for var process 4 | R | - | Zone 5 | Multiple word access |
| 802a | Quick Buffer for var process 5 | R | - | Zone 5 | Multiple word access |
| 802b | Quick Buffer for var process 1 | R | - | Zone 6 | Multiple word access |
| 802c | Quick Buffer for var process 2 | R | - | Zone 6 | Multiple word access |
| 802d | Quick Buffer for var process 3 | R | - | Zone 6 | Multiple word access |
| 802e | Quick Buffer for var process 4 | R | - | Zone 6 | Multiple word access |
| 802f | Quick Buffer for var process 5 | R | - | Zone 6 | Multiple word access |
| 8030 | Quick Buffer for var process 1 | R | - | Zone 7 | Multiple word access |
| 8031 | Quick Buffer for var process 2 | R | - | Zone 7 | Multiple word access |
| 8032 | Quick Buffer for var process 3 | R | - | Zone 7 | Multiple word access |
| 8033 | Quick Buffer for var process 4 | R | - | Zone 7 | Multiple word access |
| 8034 | Quick Buffer for var process 5 | R | - | Zone 7 | Multiple word access |

| Address hex | Description | R/W | Default Value | Note | Access |
|---|---|---|---|---|---|
| 8035 | Quick Buffer for var process 1 | R | - | Zone 8 | Multiple word access |
| 8036 | Quick Buffer for var process 2 | R | - | Zone8 | Multiple word access |
| 8037 | Quick Buffer for var process 3 | R | - | Zone 8 | Multiple word access |
| 8038 | Quick Buffer for var process 4 | R | - | Zone 8 | Multiple word access |
| 8039 | Quick Buffer for var process 5 | R | - | Zone 8 | Multiple word access |
| 803a | Quick Buffer for var process 1 | R | - | Zone 9 | Multiple word access |
| 803b | Quick Buffer for var process 2 | R | - | Zone9 | Multiple word access |
| 803c | Quick Buffer for var process 3 | R | - | Zone 9 | Multiple word access |
| 803d | Quick Buffer for var process 4 | R | - | Zone 9 | Multiple word access |
| 803e | Quick Buffer for var process 5 | R | - | Zone 9 | Multiple word access |
| 803f | Quick Buffer for var process 1 | R | - | Zone 10 | Multiple word access |
| 8040 | Quick Buffer for var process 2 | R | - | Zone 10 | Multiple word access |
| 8041 | Quick Buffer for var process 3 | R | - | Zone 10 | Multiple word access |
| 8042 | Quick Buffer for var process 4 | R | - | Zone 10 | Multiple word access |
| 8043 | Quick Buffer for var process 5 | R | - | Zone 10 | Multiple word access |
| 8062 | Serial link error 1 | R | | Slave 1 | Multiple word access |
| 8063 | Serial link error 2 | R | | Slave 2 | Multiple word access |
| 8064 | Serial link error 3 | R | | Slave 3 | Multiple word access |
| 8065 | Serial link error 4 | R | | Slave 4 | Multiple word access |
| 8066 | Serial link error 5 | R | | Slave 5 | Multiple word access |
| 8067 | Serial link error 6 | R | | Slave 6 | Multiple word access |
| 8068 | Serial link error 7 | R | | Slave 7 | Multiple word access |
| 8069 | Serial link error 8 | R | | Slave 8 | Multiple word access |
| 806A | Serial link error 9 | R | | Slave 9 | Multiple word access |
| 806B | Serial link error 10 | R | | Slave 10 | Multiple word access |

(1): **Note 1** Value Bit = 1 slave On Line , Value Bit = 0 slave Not On Line

(2): **Note 2** GFX4 only

(3): **Note 3** GEFLEX=199 ;GFX4 =200

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Offset |
|---|---|---|---|---|---|---|---|---|
| Slave 16 (2) | Slave 15 (2) | Slave 14 (2) | Slave 13 (2) | Slave 12 (2) | Slave 11 (2) | Slave 10 | Slave 9 | High |

| Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Offset |
|---|---|---|---|---|---|---|---|---|
| Slave 8 | Slave 7 | Slave 6 | Slave 5 | Slave 4 | Slave 3 | Slave 2 | Slave 1 | Low |

**GEFRAN spa**
via Sebina, 74 - 25050 Provaglio d'Iseo (BS) - ITALY
Tel. +39 0309888.1 - Fax +39 0309839063 - www.gefran.com