



1 Overview

Simple Modbus network should be setup easily.

This document provides more in-depth information about eWON Modbus operation to help you debug more tricky situations.

This Technical Note note describes some internal mechanisms in the eWON Modbus engine, and gives you some hints to debug the communication problems.

2 Internal eWON Modbus RTU

Modbus RTU is the communication protocol that is used on the RS485 eWON extension bus.

On this RS485 network, the eWON always acts as a master.

That means that a transaction between the eWON and one of the devices on the RS485 bus is always made on a request from the eWON, then the eWON waits for a response from the device until received or until time-out occurs. Then the transaction is considered as finished.

The eWON will never break a transaction. That means that if 2 transactions are pending, the eWON will always wait for the first to complete before starting the next one.

If the eWON is waiting for a reply from a device, it will not start a new transaction before reply is received or time-out has occurred.

There are 2 sources of transaction requests inside the eWON:

- **The internal Modbus IO server**
- **The Modbus Gateway**

Each device on the Modbus network has a modbus address between 1 and 255. Even the eWON has a Modbus address, although it is a Modbus master.

The Modbus Router will receive the transactions and will execute them in sequence.

For all eWON versions

The following picture describes the router operation. As you can see, transaction requests can come from the IO server itself or from the Modbus gateway:

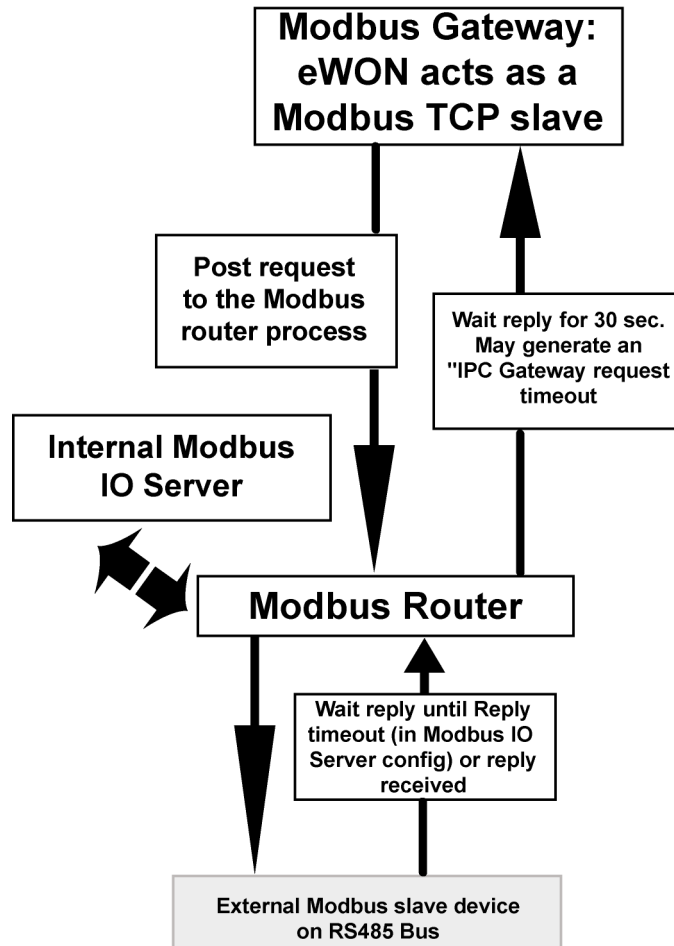


Figure 1: Description of the router operation

When a request is received in modbus TCP and the unit ID (Modbus address) is not equal to the Modbus address of the eWON, the request is passed to the Modbus gateway. This request must be executed on the 485 Modbus RTU bus and the device reply must be forwarded to the Modbus TCP network.

The gateway operation is always enabled.

When requests comes at the same time from the Modbus IO server and from the Modbus gateway, the Modbus IO server requests will be executed first. When no more requests are present in the IO server, then the router will check if requests are pending in the gateway.

This means that if the IO server is polling continuously its devices, the gateway request will never be executed, resulting in time-outs (see the 30sec. time-out). If this happens, it means that your modbus RTU network is overbooked, you need to slow down device polling. This error is logged in the eWON event log file.

3 Very special situations

- **One device on the Modbus RTU does not respond to Modbus IO server (device A), reply time-out is set to 5 secs.**

Suppose a Modbus TCP master posts a request for device B, with a reply time-out of 3secs, the eWON polls the device B, the device does not respond so there is a time-out of 5secs, but before the 5secs of device B are over, the Modbus TCP master time-out occurs (3secs); the master may close the TCP socket in that case and open a new one (this is recommended).

When the eWON receives the reply from device B, the socket is already closed and reply posting fails (a socket error will be logged in the events file).

- **A gateway request to Device A results in a time-out on the Modbus RTU.**

It may happen that the Modbus TCP request arrives, the router posts the request on the Modbus RTU bus, but the device never answers (communication problem), then the reply time-out occurs. In that case, the Modbus gateway will post a standard Modbus message to its master with error code 0x0B (*gateway target device failed to respond*). No error is logged in the events file.

4 See also

The Modbus protocol (RTU and TCP) can be downloaded from the Schneider/Modicon web site:

<http://www.modicon.com/openmbus/>

The specification is an open standard.