Overview

Modbus RTU is an open serial protocol derived from the Master/Slave architecture that was originally developed by Modicon (now Schneider Electric). It is a widely accepted serial-level protocol due to its ease of use and reliability. Modbus RTU is commonly used within Building Management Systems (BMS) and Industrial Automation Systems (IAS).

Modbus RTU messages are a simple 16-bit structure with a CRC (Cyclic-Redundancy Check). The simplicity of these messages is to guarantee reliability. Due to this simplicity, the basic 16-bit Modbus RTU register structure can be used to pack in floating-point, tables, ASCII text, queues, and other unrelated data. This protocol primarily uses an RS-232 or RS-485 serial interface for communications.

EasyEdge Modbus RTU Engine allows you to connect devices using the Modbus protocol. EasyEdge Modbus RTU Engine operates as a Modbus master and implements two-way communication. It can connect to multiple devices even with different communication parameters.

Features

- Supports multiple equipment via RTU;
- Supports files descriptor (.pdf or .csv) parser (JiT Connector);
- Supports adjustable address base (0 or 1);
- Supports full address range (0-65535);
- Supports HEX addressing (0-FFFF);
- Supports word and byte swapping (byte order):
  - MSW: Most significant word first;
  - LSW: Least significant word first;
  - MSB: Most significant byte first;
  - LSB: Least significant byte first.
- Supports equipment slave id full range (1-247);
- Supports broadcast messages using slave id 0;
- Supported functions:
  - read coil status (01);
  - read input status (02);
  - read holding registers (03);
  - read input registers (04);
  - force single coil (05);
  - preset single register (06);
  - force multiple coils (15);
  - preset multiple registers (16);
  - Exceptions;
- Supported data types:
  - Boolean;
  - Integer8;
  - Integer16;
  - Integer32;
  - Integer64;
  - Unsigned8;
  - Unsigned16;
  - Unsigned32;
  - Unsigned64;
  - Floating Point 32;
  - Floating Point 64;
  - String;
- Support for reading/writing data that spans multiple contiguous registers with different sizes and byte order (for example interpreting the value of four contiguous registers as a 64-bit Floating Point);
- Adjustable polling request time per equipment;
- Adjustable minimum request interval per register;
- Adjustable pooling request timeout;
- Supports automatic black list, avoiding requests for disconnected equipment;
- Supports minimum channel silence, forcing a time between every request on the serial bus;
- Support for least significant word (LSW) and most significant word (MSW);
- Besides LSW/MSW, it allows big-endian/little-endian swapping on all scalar data types;
- Allows different communication options (baud rate, byte size, parity and stop bits) on the same serial bus;
- Inconsistent data canceling, suppressing Modbus vulnerability for out-of-time and out-of-order replies;
- Allows Ethernet encapsulation providing communication with serial devices over Modbus TCP/IP servers.