

Overview



Modbus TCP is a variant of the Modbus family of vendor-neutral communication protocols intended for supervision and control of automation equipment. Specifically, it covers the use of Modbus messaging in an 'Intranet' or 'Internet' environment using the TCP/IP protocols. The most common use of the protocols at this time is for Ethernet attachment of PLCs, I/O modules, and 'gateways' to other simple field buses or I/O networks.

Modbus data transactions are traditionally stateless, making them highly resistant to disruption from noise and yet requiring minimal recovery information to be maintained at either end. Programming operations, on the other hand, expect a connection-oriented approach. Modbus TCP handles both situations. A connection is easily recognized at the protocol level, and a single connection may carry multiple independent transactions. In addition, TCP/IP allows a very large number of concurrent connections, so in most cases, it is the choice of the initiator whether to reconnect as required or re-use a long-lived connection.

EasyEdge Modbus TCP Engine allows you to connect devices using the Modbus protocol. EasyEdge Modbus TCP Engine operates as a Modbus master and implements two-way communication and can connect to multiple devices simultaneously and support Modbus RTU over TCP/IP.

Features

- Supports multiple equipment via IP addressing;
- Supports Modbus RTU over TCP/IP;
- 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);
- 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.