

User manual for liblightmodbus v1.4

Jacek Wieczorek
and all the contributors

July 1, 2018

Contents

1	About liblightmodbus	4
2	Building liblightmodbus	5
3	Library core	5
3.1	Data types	5
3.1.1	Error type - <code>ModbusError</code>	5
3.1.2	Modbus exception type - <code>ModbusExceptionCode</code>	5
3.1.3	Modbus data type type - <code>ModbusDataType</code>	5
3.2	Functions	5
3.2.1	Bit masks operation functions - <code>modbusMaskRead</code> , <code>modbusMaskWrite</code>	5
3.2.2	Modbus 16-bit CRC calculation function - <code>modbusCRC</code>	5
4	Slave device functionality	5
4.1	Slave-related data types	5
4.1.1	Slave device data container - <code>ModbusSlave</code>	5
4.1.2	Register query type - <code>ModbusRegisterQuery</code>	5
4.1.3	User-defined Modbus function type - <code>ModbusUserFunction</code>	5
4.1.4	Register callback function type - <code>ModbusRegisterCallbackFunction</code>	5
4.2	Setup and cleanup	5
4.2.1	Slave initialization function - <code>modbusSlaveInit</code>	5
4.2.2	Slave destructor function - <code>modbusSlaveEnd</code>	5
4.3	Request processing	5
4.3.1	Universal request parser function - <code>modbusParseRequest</code>	5
4.3.2	Modbus exception builder function - <code>modbusBuildException</code>	5
4.3.3	User-defined Modbus functions	5
4.3.4	User-defined register/coil callback functions	5
5	Master device functionality	5
5.1	Master-related data types	5
5.1.1	Master device data container - <code>ModbusMaster</code>	5
5.2	Setup and cleanup	5
5.2.1	Master initialization function - <code>modbusMasterInit</code>	5
5.2.2	Master destructor function - <code>modbusMasterEnd</code>	5
5.3	Making requests	5
5.3.1	<code>modbusBuildRequest0102</code>	5
5.3.2	<code>modbusBuildRequest01</code>	5
5.3.3	<code>modbusBuildRequest02</code>	5
5.3.4	<code>modbusBuildRequest0304</code>	5
5.3.5	<code>modbusBuildRequest03</code>	5
5.3.6	<code>modbusBuildRequest04</code>	5

5.3.7	modbusBuildRequest05	5
5.3.8	modbusBuildRequest06	5
5.3.9	modbusBuildRequest16	5
5.3.10	modbusBuildRequest15	5
5.3.11	modbusBuildRequest22	5
5.4	Processing slaves' responses	5
5.4.1	Universal response parser function - modbusParseResponse	5
6	Addons	5

1 About liblightmodbus

Liblightmodbus is a lightweight cross-platform Modbus RTU library.

2 Building liblightmodbus

3 Library core

3.1 Data types

3.1.1 Error type - ModbusError

3.1.2 Modbus exception type - ModbusExceptionCode

3.1.3 Modbus data type type - ModbusDataType

3.2 Functions

3.2.1 Bit masks operation functions - modbusMaskRead, modbusMaskWrite

3.2.2 Modbus 16-bit CRC calculation function - modbusCRC

4 Slave device functionality

4.1 Slave-related data types

4.1.1 Slave device data container - ModbusSlave

4.1.2 Register query type - ModbusRegisterQuery

4.1.3 User-defined Modbus function type - ModbusUserFunction

4.1.4 Register callback function type - ModbusRegisterCallbackFunction

4.2 Setup and cleanup

4.2.1 Slave initialization function - modbusSlaveInit

4.2.2 Slave destructor function - modbusSlaveEnd

4.3 Request processing

4.3.1 Universal request parser function - modbusParseRequest

4.3.2 Modbus exception builder function - modbusBuildException

4.3.3 User-defined Modbus functions

4.3.4 User-defined register/coil callback functions

5 Master device functionality

5.1 Master-related data types

5.1.1 Master device data container - ModbusMaster

5.2 Setup and cleanup

5.2.1 Master initialization function - modbusMasterInit

5.2.2 Master destructor function - modbusMasterEnd

5.3 Making requests

5.3.1 modbusBuildRequest0102

5.3.2 modbusBuildRequest01

5.3.3 modbusBuildRequest02

5.3.4 modbusBuildRequest0304