

FieldTalk™ Modbus Slave C++ Library

FT-MBSV-LX-EVAL Read Me Notes

Revision 2.11.0, 2026-04-03

Welcome to the evaluation of FieldTalk™ Modbus Slave C++ Library !

This evaluation package allows you to:

- Read the product manual
- Browse the Modbus Slave library API documentation
- Study the source code of the Modbus examples provided
- Modify and compile the Modbus examples (for selected targets processors only)
- Create your own Modbus programs and test them using the library (for selected targets processors only)

Documentation

To browse the *API Documentation* please click here [[doc/html/index.html](#)].

To read or print the *Product Manual* please click here [[doc/libmbusslave.pdf](#)].

Please refer to the additional *Application Notes* published on our web site: <http://www.modbusdriver.com/doc>

Limitations and Restrictions

The evaluation version is functional limited to 100 modbus operations. Once 100 operations are reached, all library functions will return error code 3 `FTALK_EVALUATION_EXPIRED`.

In addition it does not contain any redistributable components. You must not redistribute any source, object or library files.

This package is provided for the purpose of evaluating whether to purchase an ongoing license to use the Package.

THE EVALUATION PERIOD IS LIMITED TO 30 DAYS and does not include the right to reproduce and distribute Applications using the Package. At the end of the evaluation period, if You do not purchase a license, You must uninstall the Package from the computers or devices You installed it on.

THE EVALUATION VERSION IS FUNCTIONAL LIMITED TO 100 MODBUS OPERATIONS!

Files part of the package

README.txt, README.pdf
These Read Me notes.

`LICENSE-LIB.txt, LICENSE-LIB.pdf`

Library License containing the Terms & Conditions of use of this software.

`doc/libmbusslave.pdf`

Documentation in PDF format.

`doc/html/index.html`

Documentation in HTML format.

`samples`

Directory with example programs which can be linked against the evaluation library as a starting point for your application.

`include`

Library C/C++ header files.

`lib/i686-linux-gnu/libmbusslave.a`

Evaluation library for Linux x86 32-bit.

`lib/x86_64-linux-gnu/libmbusslave.a`

Evaluation library for Linux x86 64-bit. For other target processors you have to compile the library from source.

`lib/arm-linux-gnueabi/libmbusslave.a`

Evaluation library for Linux ARMv7 32-bit. This library would be suitable for the popular Raspberry PI and BeagleBone boards.

`lib/aarch64-linux-gnu/libmbusslave.a`

Evaluation library for Linux ARMv8 64-bit (AArch64). For other target processors you have to compile the library from source.

Distributable components

This is an evaluation edition and does not contain any redistributable components.



You must not redistribute any executables, source, object or library files which are based on this evaluation edition.

Release history

2.11.0 (2026-04-02)

API changes

- Added additional overloads for `installIpAddrValidationCallBack()`, `installMasterPollNotifyCallBack`, `installMasterDisconnectCallBack` which allow passing of a user data pointer.

2.10.0 (2025-02-26)

API changes

- Master poll notification callback changed the signature from `void` to `int`. It now returns 1 or 0 and if 0 the connection is dropped and the poll rejected.

2.9.1 (2024-05-14)

Enhancements

- C wrapper library: Added UDP protocol `mbusSlave_createUdpProtocol()`

API changes

- C wrapper library: renamed `mbusSlave_setTcpPort()` to `mbusSlave_setPort()`, renamed `mbusSlave_createTcpOverRtuProtocol()` to `mbusSlave_createRtuOverTcpProtocol()`, renamed `mbusSlave_startupTcpServer` to `mbusSlave_startupIpServer()`, added `mbusSlave_configureTcpNotifications()` and `mbusSlave_configureUdpNotifications`. This change only affects users using the C wrapper API.

Bug fixes

- Linux: The `serverLoop()` returned wrongly an error `FTALK_FILEDES_EXCEEDED` return code if it was interrupted by a signal. This only would occur running under Mono.

2.9.0 (2024-01-15)

API changes

- C wrapper library callbacks pass `MasterInfo` structure as additional argument. This change only affects users using the C wrapper API.

2.8.3 (2023-03-29)

Enhancements

- Updated VxWorks support for SDK 7 and VxWorks 21 and 22
- Increased length limit for Modbus function 21 Write File Record

API changes

- Added `setConnectionTimeOut` to exports

Other

- Fixed compilation error with QNX 6.6 (invalid conversion from '`SOCK_LEN_TYPE*`...')

2.8.2 (2021-03-31)

Enhancements

- Added multi-homing on non-aliased adapter to UDP protocol
- Remove limit of 2000 addressable coils
- New makefiles for Linux platforms following the triplet naming convention

Other

- Added pre-compiled binaries for Linux ARM
- Removed man pages from documentation

2.8.1 (2019-08-13)

API changes

- Added `mbusSlave_setFrameTolerance` export and init

Bug fixes

- Fixed null-pointer de-reference when receiving broadcasts for serial RTU and ASCII protocols. This would cause an access violation (Windows) or a segmentation fault (Linux).
- Fixed data length validation in Modbus/TCP header
- Fixed Modbus/TCP disconnection after FC43 *Read Device Identification* query
- Fixed incorrect number of objects reported for *Read Device ID* code 2 in Function code 43 *Read Device Identification*

Other

- Support compilation for Windows UWP

2.8.0 (2018-11-23)

Enhancements

- Added Modbus/UDP protocol variant

Other

- Removed support for the discontinued DEC True 64 and SGI IRIX platforms.

API changes

- Renamed `MbusSerialSlaveProtocol` class to `MbusSerialServerBase`
- Introduced new `MbusIpServerBase` base class for IP based protocols

Bug fixes

- Fixed bug converting COM-style port names on WinCE

2.7.0 (2018-07-09)

Enhancements

- RTU over TCP protocol added, which is also known as encapsulated RTU.

2.6.2 (2018-04-06)

Enhancements

- MODBUS/TCP: MAX_CONNECTIONS increased to 32

API changes

- getSlaveId(), getRunIndicatorStatus(), getDeviceIdObject() functions renamed to reportSlaveId(), reportRunIndicatorStatus(), readDeviceIdentification() and are now officially documented. Either a deprecation error or a compile-time (conflicting return type) error is generated if one of those functions is used, to make the user aware of the name change.

Bug fixes

- In multi-drop operation if TRACELOG logging was enabled a CRC error was logged when processing a reply from another device. This wrong indication did not affect operation and affected only the output of the logging.

2.6.1 (2018-02-28)

Enhancements

- Non-Windows platforms: Changed build system from shell scripts to makefiles to better support cross-compiling

2.6.0 (2018-02-23)

Enhancements

- Updated Visual Studio project files to native VS2017 format and removed obsolete VS2005, eVC++4.0 and VC++6.0 project files (Note: Library does still compile with VS2005, eVC++4.0 and VC++6.0)
- Removed dependency from sprintf/swprintf/sscanf/ftol for better binary compatibility between different VC++ versions

API changes

- Removed DLL export declarations from all classes

2.5.10 (2016-01-29)

Bug fixes

- Made serial `wchar_t` `startupServer` virtual.

2.5.9 (2015-03-25)

Enhancements

- Fixed some warnings in samples, removed all but ARMv6 and x86 platforms for samples.

Bug fixes

- Daniel/ENRON protocol fixed word order for Modbus function code 3 and 4

2.5.8 (2014-09-17)

Bug fixes

- Linux: Fixed timer constructor bug introduced in version 2.5.1 which could lead under certain circumstances to lock-ups

2.5.7 (2014-01-29)

Enhancements

- Fixed wrong `getConnectionStatus()` indication after TCP disconnection.
- Improved time-out handling of `timeOutHandler()` callbacks

2.5.6 (2013-05-16)

Bug fixes

- `diagslave`: Fix bug in `diagslave` for slave ID of 255.

2.5.5 (2013-03-28)

API changes

- Windows: Replaced `startupServer(const TCHAR*...)` functions with two separate functions for `char*` and `wchar_t*` types to allow compilation for UNICODE and non-UNICODE projects without having to recompile the library.

Enhancements

- Windows: `COMn` will now automatically be substituted with `\\.\\COMn` for more convenient access to serial ports ≥ 10

Bug fixes

- MbusTcpSlaveProtocol::setConnectionTimeout(long masterTimeout) function did not check if passed masterTimeout parameter is within correct range.

2.5.4 (2012-03-07)

Enhancements

- Added support for MinGW compiler on Windows platform

2.5.3 (2012-02-17)

Bug fixes

- Fixed double datatable destructor call
- Only check for POSIX.4 monotonic timers if compiler macro CLOCK_MONOTONIC set

2.5.2 (2010-09-11)

Enhancements

- Linux: Set serial port to low latency mode (ASYNC_LOW_LATENCY)
- Added enronslave example

Bug fixes

- Fixed wrong getConnectionStatus() indication for foreign slave IDs

2.5.1 (2010-08-27)

API changes

- Changed string constants from TCHAR* to const TCHAR*
- Added setConnectionTimeout/getConnectionTimeout functions to MbusTcpSlaveProtocol
- Added getMasterInfo() function to data table interface.
- Removed deprecated 2.2 API constructors which had a datatable parameter. Use addDataTable instead.
- Removed redundant and rarely used overloaded method startupServer(string portName, Int32 baudRate). Use `startupServer(portName, baudRate, dataBits, stopBits, parity) instead.

Enhancements

- Additional master meta data is made available via the MasterInfo structure

- Systems supporting POSIX.4 timers now use monotonic timers (CLOCK_MONOTONIC). This applies for example to QNX, BSD and Linux.
- Added Visual Studio 2010 project files
- Windows: COMn will now automatically be substituted with \\.\COMn for more convenient access to serial ports ≥ 10

Bug fixes

- Fixed wrong getConnectionStatus() indication after TCP disconnection
- Solaris: Removed -compat compiler flag which is only supported by Sparc but not i386.
- Modbus RTU: Reduced delay when replying.

Other

- Support for Visual Studio 2003 has been discontinued and the relevant project file removed.

2.5.0 (2009-10-16)

API changes

- Defaults for serial protocols changed to 19200 and even parity as per Modbus specification.
- Modbus function 43/14 Read Device Identification implemented
- Modbus functions 17 Report Slave ID, 20 Read File Record and 21 Write File Record implemented
- Added optional suppression of exception replies. The new functions `disableExceptionReplies()` and `enableExceptionReplies()` configure this behaviour.

Enhancements

- C wrapper library added (MbusSlaveCexports.cpp, MbusSlaveCexports.h)
- Modbus/TCP: Support for multiple Master requests on the same TCP connection added.
- Made sockets non-blocking, so we cannot get send blocked in case of a rogue master is flooding us with requests.
- Added x64 configuration for Windows platform
- Removed default libraries from library build for Windows platform
- Isolated SerialPort and Timer objects from library interface to allow master and slave to be combined in one project.
- Prefixed class names of internal classes to avoid potential namespace conflicts with other class libraries.
- Improved RS-485 support for Windows and Linux platforms, delay not required anymore, `enableRs485Mode` can now be used reliably with a delay of 0.

- Added Makefile.qnx6 to support host compilation for QNX 6 platform.

Bug fixes

- Fixed bug that a register count > 62 would trigger ENRON 32-bit mode.
- Added MSG_NOSIGNAL to send() calls to stop a broken pipe signal terminating the application on Linux and BSD.
- VxWorks: Added -mlongcall flag and removed -nostdinc flag to build files, corrected case of vxWorks.h include in hmserio.h.

2.4.3 (2008-08-25)

API changes

- Modbus/TCP: Master poll and disconnect notification callbacks and hooks added.

Enhancements

- mastersupervision example added.

Bug fixes

- Modbus/TCP: Total and success counter added.
- Fixed word order of 32-bit ENRON variables.

2.4.2 (2008-07-14)

API changes

- Added Daniel/Enron 32-bit mode support.

Enhancements

- RTS is set when opening port and not in RS-485 mode for better support of handshake line powered devices.

2.4.1 (2007-05-10)

API changes

- UNICODE string support introduced for `startupServer` and `getBusProtocolErrorText` functions.

2.4.0 (2006-10-20)

API changes

- `startupServer()` changed default parity to even to be conform with the Modbus standard.

- Method `getConnectionStatus()` added.

Enhancements

- Changed error values to decimal for easier reference.

Bug fixes

- Fixed masking bug for function code 22
- Fixed lock-up in the rare case that `serialPort.receive` returns an API error.
- Introduced new definition for `EINTR`; before redefinition, `EINTR` could cause a warning under VC++.

2.3 (2006-02-15)

API changes

- Function code 8 (Diagnostic), subcode 0 (Return query data) added.
- `getConnectionStatus` method added.
- Support of multiple slave addresses added. This required a change in the way how data tables are associated with the protocol. Instead of:

```
MbusTcpSlaveProtocol mbusServer(&dataProvider);  
mbusServer.startupServer(1);
```

one should use now:

```
MbusTcpSlaveProtocol mbusServer;  
mbusServer.addDataTable(1, &dataProvider); // More tables can be added!  
mbusServer.startupServer();
```

The v2.2 API is still kept as deprecated methods for backward compatibility with exiting source code.

Bug fixes

- ASCII protocol's `serverLoop` does return in the time-out case.

2.2 (2004-02-16)

API changes

- Added method to accept or reject a master's connection based on it's IP address.

Enhancements

- Allow 0 as slave address (for TCP).

Bug fixes

- Fixed broadcast bug (broadcasts where not processed),

2.1 (2003-09-21)

API changes

- Added ASCII protocol.
- RTU/ASCII: Added RS485 mode for Win32, QNX and Linux platforms.

Bug fixes

- RTU: Fixed checksum bug which was introduced in v2.0.

2.0 (2003-06-18)

API changes

- Added an overloaded version of `startupServer` to allow for multihomed servers.

Bug fixes

- QNX Edition: Fixed `IO_ERROR` bug for RTU protocol.

1.3 (2002-11-27)

- First release