

# FieldTalk™ Modbus Master C++ Library

## FT-MBMP-LX-EVAL Read Me Notes

Revision 2.13.3, 2025-04-03

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

This evaluation package allows you to:

- Read the product manual
- Browse the Modbus Master 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/libmbusmaster.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/libmbusmaster.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/libmbusmaster.a`

Evaluation library for Linux x86 32-bit.

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

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

`lib/arm-linux-gnueabi/libmbusmaster.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/libmbusmaster.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

### 2.13.3 (2025-04-03)

#### Bug fixes

- Corrected fix in `readDeviceIdentification()` for a slave conformity level of 1 triggered wrongly an `FTALK_INVALID_REPLY_ERROR`

### 2.13.2 (2025-02-26)

#### Bug fixes

- readDeviceIdentification() for a slave conformity level of 1 triggered wrongly an FTALK\_INVALID\_REPLY\_ERROR

### 2.13.1 (2024-07-27)

#### Enhancements

- C wrapper library: Added support for 64-bit data types (int64\_t , double) for FC3, FC4 and FC16

### 2.13.0 (2024-07-26)

#### Enhancements

- Added support for 64-bit data types (int64\_t , double) for FC3, FC4 and FC16

### 2.12.1 (2024-03-09)

#### Enhancements

- C wrapper library: Added mbusMaster\_readDeviceIdentification() function

#### Bug fixes

- readFileRecord() with a record count of 2 or less triggered FTALK\_ILLEGAL\_ARGUMENT\_ERROR
- libmbusmaster.vcxproj project file failed to load

### 2.12.0 (2023-12-19)

#### Enhancements

- Added an additional openProtocol() overload with a second argument to optionally configure the network interface adapter to be used for the connection.

### 2.11.0 (2023-10-16)

#### Enhancements

- Added Lufkin ELAM protocol variant encapsulated in TCP (ELAM over TCP)
- Lufkin ELAM protocol now supports variable length replies

#### Bug fixes

- Fix bug in serial ELAM protocol which reports wrong exception code
- Fix bug in serial ELAM protocol which configures only the first 255 slaves when using configure...() functions (like configureCountFromZero())

## 2.10.4 (2023-04-04)

### Bug fixes

- Modbus/TCP: Fix wrong invalid MBAP ID/invalid frame indication for the following frame if extraneous characters are sent in the TCP stream

### Enhancements

- Added validation for implied reply length against actual reply length
- Modbus/TCP: An FTALK\_INVALID\_MBAP\_ID error will force close Modbus/TCP connection
- Modbus/TCP: Added check for extraneous reply bytes in TCP data stream
- Updated VxWorks support for SDK 7 and VxWorks 21 and 22

## 2.10.3 (2021-11-17)

### Bug fixes

- Linux/QNX/Unix: Fix linker error when using diagnostic methods like `readDeviceIdentification`

## 2.10.2 (2021-09-23)

### Bug fixes

- Fixed MbusUdpMasterProtocol regression introduced in version 2.10.0
- Fix compilation error with QNX 6.6 (invalid conversion from SOCK\_LEN\_TYPE)

## 2.10.1 (2021-06-16)

### Bug fixes

- Added missing header file MbusDiagnosticFunctions.hpp
- Fixed MbusRtuMasterProtocol regression introduced in version 2.10.0

## 2.10.0 (2021-06-14)

### Enhancements

- Added Variable length replies to `customFunction()` for RTU and TCP to support ENRON log functions

### API changes

- Added Modbus function 43 (hex 2B) subfunction 14 (hex 0E), Read Device Identification

- Added Modbus function 20 (hex 14) Read File Record
- Added Modbus function 21 (hex 15) Write File Record

### 2.9.3 (2021-03-31)

#### API changes

- isOpen() method is now marked deprecated. This method is often misunderstood to report the connection status which it does not.

#### Enhancements

- Moved buffer clearing after poll delay period expires
- New makefiles for Linux platforms following the triplet naming convention

#### Bug fixes

- Modbus/TCP protocol only: readExceptionStatus() always returned FTALK\_REPLY\_TIMEOUT\_ERROR
- Fixed compilation issue of tcpsimple.c due to renamed exports
- UDP protocol only: Fix length detection when transaction ID is set to 0
- Fixed compilation issue of tcpsimple.c due to renamed exports
- Fixed Winsock lib paths for WinCE 80

#### Other

- Added pre-compiled binaries for Linux ARM
- Removed man pages from documentation
- Support compilation for Windows UWP

### 2.9.2 (2019-02-25)

#### Bug fixes

- Fixed bug converting COM-style port names on WinCE
- Fixed closing t/o regression under Linux introduced in v2.9.1.

### 2.9.1 (2018-11-20)

#### Enhancements

- TCP/IP connection closing can be made synchronous by configuring a closing time-out. This then prevents the master from opening a new socket before the slave has acknowledged that the old socket has been closed.

## Other

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

## 2.9.0 (2018-04-06)

### Enhancements

- Added Modbus/UDP protocol variant

### API changes

- Renamed MbusSerialMasterProtocol class to MbusSerialClientBase
- Introduced new MbusIpClientBase base class for IP based protocols

## 2.8.4 (2018-02-28)

### Enhancements

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

### Bug fixes

- v2.8.0 introduced a bug which did not initialise correctly the slave configuration modes like 32-bit endianness, count start and ENRON mode in the constructor of protocol objects.

## 2.8.3 (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/sscanf/ftol for better binary compatibility between different VC++ versions

### API changes

- Removed DLL export declarations from all classes

## 2.8.2 (2017-03-24)

### Bug fixes

modpoll: Fixed bug which prevented start register 0 in PDU mode

## 2.8.1 (2016-06-16)

### API changes

- Changed prototype of `mbusMaster_customFunction` to not depend on `size_t` which has different size on 32 and 64-bit systems

### Bug fixes

- Fixed reply array size mismatch in RTU and ASCII implementation of `customFunction`

## 2.8.0 (2015-12-03)

### Enhancements

- Added Lufkin ELAM protocol variant

## 2.7.0 (2015-06-02)

### Enhancements

- Added ASCII over TCP protocol

### Bug fixes

- VxWorks: Fixed compilation issue in `hmtimer.cpp/hpp` with VxWorks 6.9

## 2.6.10 (2014-02-12)

### Bug fixes

- Linux: Fixed timer constructor bug introduced in version 2.6.4 which could lead under certain circumstances to lock-ups

## 2.6.9 (2013-03-20)

### Enhancements

- Updated modpoll example

## 2.6.8 (2012-05-02)

### API changes

- Windows: Replaced `openProtocol(const TCHAR*...)` functions with two separate functions for `char*` and `wchar_t*` types to allow compilation for UNICODE

### Enhancements

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

- Added support for MinGW compiler on Windows platform

## 2.6.7 (2012-02-16)

### Bug fixes

- Windows: Hostname resolving for `openProtocol()` was not working on Windows

## 2.6.6 (2011-05-27)

### Enhancements

- Slave ID of 0 is supported for Modbus/TCP
- Linux: Serial port is now put into `low_latency` mode

## 2.6.5 (2011-04-05)

### API changes

- Added `openUseExistingConnection()` for Windows platform to support Modbus over existing Modem connections.

### Enhancements

- Added support for Intel based Mac OS X
- Added Visual Studio 2010 project files

### Other

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

## 2.6.4 (2010-09-24)

### Bug fixes

- Monotonic timers were not effective when changing system date into future

## 2.6.3 (2010-08-25)

### Enhancements

- Systems supporting POSIX.4 timers now use monotonic timers (`CLOCK_MONOTONIC`). This applies for example to QNX, BSD and Linux.

## 2.6.2 (2010-06-17)

### API changes

- Changed 32-bit integer data types from `long` to `int` so package can be used with compiler using 64-bit longs (eg 64-bit Linux).



- Changed string constants from `TCHAR*` to `const TCHAR*` to fix gcc 4 warnings

## 2.6.1 (2010-01-29)

### API changes

- Defaults for serial protocols changed to 19200 and even parity as per Modbus specification.
- Removed redundant and rarely used overloaded method `openProtocol(portName, baudRate)`. Use `openProtocol(portName, baudRate, dataBits, stopBits, parity)` instead.

### Enhancements

- Added binary compatibility with eVC++ 4.0 by adding additional entry points for functions using `wchar_t` as parameter type.
- Added 64-bit solution files for examples

### Bug fixes

- Linux: Fixed bug introduced on in 2.6.0 which caused lock-ups on Linux due to improper handling if `ioctl`.
- Solaris: Removed `-compat` compiler flag which is only supported by Sparc but not i386.

## 2.6.0 (2009-10-17)

### API changes

- Added support for Daniel/Enron style 32-bit mode.
- Added support for PDU based register addressing model.
- Endianess can now be configured on a per slave basis.
- DEF file removed (MS VC++)

### Enhancements

- Buffer security option disabled (`/GS-`) for Win32 static library build to maintain binary compatibility with VC++ 6.0
- Added x64 configuration 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

- Modbus/TCP and RTU over TCP: Added handling of `EINTR` during reception of first message part.
- VxWorks: Added `-mlongcall` flag and removed `-nostdinc` Flag to build files, corrected case of `vxWorks.h` include in `hmserio.h`.

## 2.5 (2007-11-06)

#### Documentation

- Fixed error in documentation of Modbus function 22 / `maskWriteRegister`.

#### API changes

- `customFunction()` method added for Modbus/TCP. This allows implementation of custom Modbus function codes.
- Modbus function 8, sub-function 0 `returnQueryData()` method added.
- Modbus function 8, sub-function 01 `restartCommunicationsOption()` method added.

#### Enhancements

- Updated maximum registers and coils to match latest Modbus specifications: Function code 16 supports now 123 registers, was 100 before. Function code 15 supports now 1968 coils, was 800 before. Also updated max values for function code 23.
- Using relative timers on Windows instead of absolute system time for time-out control.

#### Bug fixes

- Fixed double time-out time for Modbus/TCP in case of a broken network connection. The time-out is now applied only once.
- Destructor call did not close the socket.
- TCP connections are now closed gracefully using a FIN/ACK sequence rather a RST.

## 2.4.1 (2008-08-24)

#### Bug fixes

- Linux: Added `size_t` patch for `hmserio.hpp`

## 2.4.0 (2006-10-20)

#### API changes

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

- Removed DLL export declaration from `getBusProtocolErrorText()`.

#### Enhancements

- New solution files for better support of Visual Studio.
- Changed error values to decimal for easier reference.

#### Bug fixes

- Introduced new definition for `EINTR`; before redefinition of `EINTR` could cause a warning under VC++.
- Fixed compilation error for `socklen_t` on QNX 6.3

## 2.3 (2005-11-11)

#### API changes

- Unicode string support introduced for `openProtocol`, `openPort`, `getBusProtocolErrorText` functions.
- DEF file added (MS VC++) to support importing the DLL version.

#### Enhancements

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

#### Bug fixes

- Fixed compilation issue on HP-UX relating `select.h`.
- Fixed typing errors for some functions in `MbusMasterCwrapper.h` (renamed to `MbusMasterCexports.h`).
- `MbusTcpMasterProtocol`: If setting retry count  $> 0$  a broken connection could cause a core dump when retrying because the socket has been closed and set to `-1` but was still used in `deliverMessage`.
- `MbusRtuOverTcpMasterProtocol`: Fixed intermittent checksum errors caused by using a wrong array index for checking the exception code.

## 2.2 (2004-05-26)

#### API changes

- RTU over TCP protocol added, which is also known as encapsulated RTU. This protocol flavour is for example used by ISaGraf Soft-PLCs.

- Added new function `adamSendReceiveAsciiCmd` for MODBUS/TCP to support ADAM 5000/6000 ASCII commands.

#### Enhancements

- Added `MbusMasterCwrapper` C wrapper code to support plain C projects and support for DLLs.

#### Bug fixes

- Fixed bug when a reply was expected from broadcasts.
- In case of send time-out the `FTALK_SEND_TIMEOUT_ERROR` error code is now returned (was `FTALK_REPLY_TIMEOUT_ERROR` before).

## 2.1 (2003-09-21)

#### Enhancements

- Improved build scripts with configuration options.

#### Bug fixes

- Fixed TCP/IP connection bug which was introduced into the v2.0 Linux release.
- Compiles again under Solaris.
- ASCII: Fixed `INVALID_FRAME_ERROR` when master was continuously polling, caused by the fact that the LF char was not flushed from buffer quick enough.

## 2.0 (2003-06-18)

#### API changes

- Lib file renamed to `libmbusmaster`.
- `configureLittleEndianInts()` and `configureIeeeFloats()` methods introduced.
- RTU/ASCII: Added RS485 mode for Win32, QNX and Linux platforms.

#### Enhancements

- Reply validation modified, this fixes problems to write single values to Quantum PLCs.
- Support for VxWorks, Irix, Solaris, DEC Alpha True 64/Digital UNIX, HP-UX and IBM AIX added.
- TRACELOG facility added.
- MODBUS/TCP: Time-out applies now also when connecting to a server.

#### Bug fixes

- ASCII: Fixed casting bug which caused protocol error when transmitting FF.

- MODBUS/TCP: Tolerate a zero address field in an exception reply.
- MODBUS/TCP: fixed auto-retry.
- Fixed compile error when compiling with MS Visual C++.

## 1.3 (2002-11-27)

### API changes

- Renamed `closeConnection` and `closePort` methods to `closeProtocol`.
- Renamed `openConnection` and `openPort` methods to `openProtocol`.
- `openProtocol` (formerly `openPort`) method opens serial port now with `NO_PARITY`.
- Methods for 32-bit types renamed. 32-bit module 10000 data type introduced.
- `getBusProtocolErrorText()` function introduced.

### Enhancements

- Serial communication functions are now signal safe.

### Bug fixes

- Fixed bug concerning big endian 32-bit word swapping.
- Fixed bug with automatic retries.
- Fixed bug in write methods for 32-bit types.

## 1.2 (2002-07-22)

### Enhancements

- Improved bus silence period for multi-drop interfaces (RS485)

### Bug fixes

- Fixed issue with time-out monitoring

## 1.1 (2002-03-03)

- First release