FieldTalk™ Modbus Master C++ Library

FT-MBMP-WIN-ALL Read Me Notes

Revision 2.6.8, 2012-05-02

Thank you for purchasing FieldTalk™ Modbus Master C++ Library Windows Edition!

This *Read Me* file contains last-minute product information for the *FieldTalk* Modbus Master C++ Library.

Documentation

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

To read or print the *Product Manual* please click here [doc/README.pdf].

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

Files part of the package

`README`, `README.pdf`
: These Read Me notes.

`LICENSE-LIB`, `LICENSE-LIB.pdf`
: Library License containing the Terms & Conditions of use of this software.

`doc/README.pdf`
: Documentation in PDF format.

`doc/README.chm`
: Documentation in Microsoft Help format.
: **Note:** If you cannot see the content of the help when opening his file, try the following: [1]
- Clear the **`Always ask before opening this file`** check box in the `Open File-Security Warning` dialog box that Windows XP displays when opening the CHM file.
- Right-click the CHM file, and then click **`Properties`**. Click **`Unblock`**.
- CHM files cannot be viewed on a shared drive. Download or copy this file to a local drive on your PC, then it will view properly.

`doc/html/index.html`
: Documentation in HTML format.

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

[1] http://support.microsoft.com/kb/902225/

`include`
: Fieldtalk library C/C++ header files.

`lib/wce/…/release/libmbusmaster.lib`
: Precompiled libraries for the Pocket PC, Windows CE and Windows Mobile platform.

`lib/win/win32/release/libmbusmaster.lib`
: Precompiled library for Windows and MS Visual C++.

`lib/win/x64/release/libmbusmaster.lib`
: Precompiled library for Windows 64-bit and MS Visual C++.

`lib/win/borland/libmbusmaster.lib`
: Precompiled library for Windows and Borland C++.

`src`
: *FieldTalk* library source code files. Only required if it is desired to re-compile the library with custom settings. The precompiled libraries should be sufficient for most applications.

`Visual Studio 2010`
: MS VC++ 2010 project files to re-build library from source.

`Visual Studio 2005`
: MS VC++ 2005 project files to re-build library from source. The files with prefix _win are for the Windows platform. The files with prefix _wce are for the Pocket PC, Windows CE and Windows Mobile platforms.

`Visual C++ 6.0`
: MS Visual C++ 6.0 project files to re-build library from source.

`src/MAKEFILE.win`
: Makefile suitable for MS Visual C++ nmake and Borland C++ make utility to re-build library from source.

`Visual Studio 2005`
: MS VC++ project files to build library for your target CPU.

`Embedded VC++ 4.0`
: MS eVC++ 4.0 project files to build library for your target CPU.

Distributable components

This *FieldTalk* edition is a static link library and does not contain any redistributable components.

You may distribute executable files (applications) which have been linked against *FieldTalk* in accordance with the Library License (see file `LICENSE-LIB`).

> ⚠ You must not redistribute any source (`*.h`, `*.c`, `*.hpp`, `*.cpp`), object (`*.o`, `*.obj`) or lib (`*.a`, `\*.lib`) files.

Release history

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 and non-UNICODE projects without having to recompile the library.
- Added openUseExistingConnection() for Windows platform to support Modbus over existing Modem connections.
- 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
- 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

- Windows: COMn will now automatically be substituted with \\.\COMn for more convenient access to serial ports >= 10
- Added support for MinGW compiler on Windows platform
- Slave ID of 0 is supported for Modbus/TCP
- Linux: Serial port is now put into low_latency mode
- Added support for Intel based Mac OS X
- Systems supporting POSIX.4 timers now use monotonic timers (CLOCK_MONOTONIC). This applies for example to QNX, BSD and Linux.
- Added binary compatibility with eVC++ 4.0 by adding additional entry points for functions using wchar_t as paramater type.
- Added Visual Studio 2010 project files

Bug fixes

- Windows: Hostname resolving was not working on Windows
- Solaris: Removed -compat compiler flag which is only supported by Sparc but not i386.

Other

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

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.
- `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.

Documentation

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

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 ojects 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.
- 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 intead of absolute system time for time-out control.

Bug fixes

- Modbus/TCP and RTU over TCP: Added handling of EINTR during receiption of first message part.
- VxWorks: Added -mlongcall flag and removed -nostdinc Flag to build files, corrected case of vxWorks.h include in hmserio.h.
- 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 definiton for `EINTR`; before redfinition 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-C++) to support importing the DLL version.
  1. 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