

# FieldTalk™ modpoll

## Linux Edition Read Me Notes

Revision 3.15, 2024-09-27

This *Read Me* file contains last-minute product information for the *FieldTalk™* modpoll utility.

modpoll is a command line based Modbus master simulator and test utility.

## Files part of the package

README.txt, README.pdf

These Read Me notes.

LICENSE-FREE.txt, LICENSE-FREE.pdf

License containing the Terms & Conditions of use for this free software.

arm-linux-gnueabi/modpoll

ARMv7 binary for 32-bit ARM Linux systems (Raspberry Pi, BeagleBoard etc)

aarch64-linux-gnu/modpoll

ARMv8 binary for 64-bit AArch64 Linux systems

armv6-rpi-linux-gnueabi/modpoll

ARMv6 binary for 32-bit ARM Linux systems (Raspberry Pi Zero)

i686-linux-gnu/modpoll

x86 binary for 32-bit x86 Linux systems

x86\_64-linux-gnu/modpoll

x86\_64 binary for 64-bit x86 Linux systems

## Usage

Usage: modpoll [OPTIONS] SERIALPORT|HOST [WRITEVALUES...]

Arguments:

SERIALPORT      Serial port when using Modbus ASCII or Modbus RTU protocol  
                   COM1, COM2 ...                                   on Windows  
                   /dev/ttyS0, /dev/ttyS1 ...                       on Linux

HOST             Host name or dotted IP address when using MODBUS/TCP protocol

WRITEVALUES     List of values to be written. If none specified (default) modpoll reads data

General options:

-m ascii         Modbus ASCII protocol  
 -m rtu           Modbus RTU protocol (default if SERIALPORT contains \ or COM)  
 -m tcp           MODBUS/TCP protocol (default otherwise)  
 -m udp           MODBUS UDP  
 -m enc           Encapsulated Modbus RTU over TCP  
 -a #             Slave address (1-247 for serial, 0-255 for TCP, 1 is default)  
 -r #             Start reference (1-65536, 1 is default). Use -0 for 0-based references.  
 -c #             Number of values to read (1-125, 1 is default), optional for writing (use -o)  
 -t 0             Discrete output (coil) data type (FC 1)

```
-t 1          Discrete input data type (FC 2)
-t 3          16-bit input register data type (FC 4)
-t 3:hex      16-bit input register data type with hex display
-t 3:i32      32-bit integer data type in input register table
-t 3:i64      64-bit integer data type in input register table
-t 3:mod      32-bit module 10000 data type in input register table
-t 3:f32      32-bit float data type in input register table
-t 3:f64      64-bit double data type in input register table
-t 4          16-bit holding register data type (FC3, default)
-t 4:hex      16-bit holding register data type with hex display
-t 4:i32      32-bit integer data type in holding register table
-t 4:i64      64-bit integer data type in holding register table
-t 4:mod      32-bit module 10000 type in holding register table
-t 4:f32      32-bit float data type in holding register table
-t 4:f64      64-bit double data type in holding register table
-t id         Read device identification objects (FC 43/14)
-t file       File record reference type 6 (FC 20/21)
-n #         File number for file record (default is 4)
-u           Display data as unsigned integers
-i           Slave operates on big-endian 32-bit/64-bit integers
-f           Slave operates on big-endian 32-bit/64-bit floats
-e           Use Daniel/Enron single register 32-bit mode (implies -i and -f)
-x           Use Lufkin ELAM extensions (RTU and RTU over TCP only)
-0           First reference is 0 (PDU addressing) instead 1
-1           Poll only once only, otherwise every poll rate interval
-l #         Poll rate in ms, (1000 is default)
-o #         Time-out in seconds (0.01 - 10.0, 1.0 s is default)
Options for MODBUS/TCP, UDP and RTU over TCP:
-p #         IP protocol port number (502 is default)
Options for Modbus ASCII and Modbus RTU:
-b #         Baudrate (e.g. 9600, 19200, ...) (19200 is default)
-d #         Databits (7 or 8 for ASCII protocol, 8 for RTU)
-s #         Stopbits (1 or 2, 1 is default)
-p none      No parity
-p even      Even parity (default)
-p odd       Odd parity
-4 #         RS-485 mode, RTS on while transmitting and another # ms after
```

## Release history

### Version 3.15 (2024-09-27)

- Add support to display integer values as unsigned values

### Version 3.14 (2024-09-07)

- Read device identification for a slave conformity level of 1 triggered wrongly an *Invalid reply error*

### Version 3.13 (2024-08-09)

- Fixed bug regards display of 64-bit input registers

### Version 3.12 (2024-07-26)

- Added support for 64-bit data types for FC3, FC4 and FC16
- Fixed issue of Read Device Identification (-t id) being limited to slaved address 1

### Version 3.11 (2024-01-22)

- Added Read Device Identification for FC 43 subfunction 14 (-t id)
- Added Read File Record FC 20 (-t file)
- Added Write File Record FC 21 (-t file)
- Added Lufkin ELAM protocol variant (-x)
- Modbus UDP: Fix length detection when transaction ID is set to 0
- Modbus/TCP: Fix wrong invalid MBAP ID/invalid frame indication for the following frame if extraneous characters are sent in the TCP stream
- Added Linux ARMv6 RPI (32-bit) platform for Pi Zero

### Version 3.10 (2021-03-26)

- Added Linux ARMv8 AArch64 (64-bit) platform

### Version 3.9 (2020-07-14)

- Added support for single register and single coil writes
- Display function code used in protocol configuration
- Removed automatic fallback to FC6 added in 3.7.

### Version 3.8 (2020-03-24)

- Writing negative values was causing *Unrecognized option or missing option parameter* error under Linux

### Version 3.7 (2019-07-21)

- Write functions with a count of 1 use now the following scheme: Registers use FC16 first, and if an illegal function exception is received will try FC6 as fallback. Coils always use FC5 for a count of 1. This helps with slave devices which do not implement mandatory FC16.

### Version 3.6 (2018-04-05)

- MODBUS UDP protocol added (-m udp)

### Version 3.5 (2017-03-24)

- Fixed argument validation bug which prevented using PDU mode with a start register of 0 (-r0 -0)

### Version 3.4 (2013-01-30)

- Increased reference count to 2000 for discretes/coils

### Version 3.3 (2012-10-25)

- Fixed error message when passing negative float values on the command line

### Version 3.2 (2012-03-28)

- COMn syntax can now also be used for COM port number  $\geq 10$

### Version 3.1 (2011-05-27)

- Slave ID of 0 is supported for Modbus/TCP

### Version 3.0 (2011-03-05)

- Write function added
- protocol is now auto-detected as RTU or TCP depending on value of first parameter
- -l pollDelay parameter added — Added "--" separator before values are printed to make parsing of result easier

### Version 2.10 (2010-08-26)

- -c parameter now accepts a value of 125.
- Changed default start reference (-r) to 1

### Version 2.9 (2010-01-29)

- Fixed lock-up issue on some Linux platforms which was introduced in 2.7.

### Version 2.8 (2009-11-16)

- Default baudrate is now 19200 as per Modbus standard.

### Version 2.7 (2009-06-04)

- Corrected help and range check for -a parameter

### Version 2.6 (2008-10-30)

- Added option -0 for PDU addressing and option -e for Enron/Daniel 32-bit mode.

### Version 2.5 (2008-04-03)

- A return code of 1 is returned if operation was not successful otherwise 0
- -c parameter now accepts a value of 100.
- Added time-out command line parameter.
- Retry count is now 0 for serial protocols (was 2 before).

### Version 2.4.0 (2006-10-20)

- Default parity changed to even as per Modbus standard.

### Revision 1.17 (2005-06-07)

- Using the -i command line parameters returned an error message in earlier releases.

### Version 2.2.1 / Revision 1.16 (2004-09-22)

- Using the -d and -s command line parameters returned an error message in earlier releases.

### Version 2.2 / Revision 1.15 (2004-04-25)

- RTU over TCP protocol added, which is also known as encapsulated RTU.
- Recompiled against 2.2 release of libmbusmaster.

### Version 2003-05-20

- Recompiled against 2.0 release of libmbusmaster.
- RTU/ASCII: Added RS-485 mode for Win32, QNX and Linux platforms.
- ASCII: Fixed casting bug which caused protocol error when transmitting FF.
- MODBUS/TCP: Time-out applies now also when connecting to a server, tolerate a zero address field in an exception reply, fixed auto-retry.

### Version 1.2 (2002-11-19)

- Terminates in case of a closed TCP/IP connection.
- Some error messages changed.
- Changed command line options for holding and input registers. -t4 is now holding register, -t3 input register.
- Retry option is now working.
- --version parameter introduced.

- Retries fixed.
- -p parameter for MODBUS/TCP introduced.
- Default parity changed to NONE.
- Based on *FieldTalk* v1.3.

### Version 1.1 (2002-07-15)

- Reference index print-out for 32-bit values corrected.
- Based on updated *FieldTalk* library which fixed issue with time-out monitoring

### Version 1.0 (2002-03-03)

- First release

---

Copyright (c) 2002-2021 *proconX* Pty Ltd. All rights reserved.

Please refer to file LICENSE-FREE for license and distribution terms.

THIS SOFTWARE IS PROVIDED BY PROCONX AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL PROCONX OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.