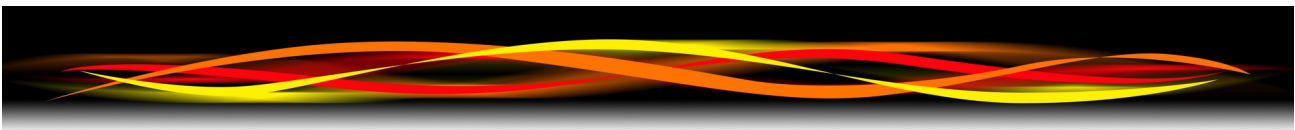


Newflow

Modbus Master Simulator

User Manual





Newflow Ltd
Unit 4B, 21 Cherry Farm Close
Malton Enterprise Park
Malton, North Yorkshire
YO17 6AW
UK

Tel: +44 1653 310 000
Email: sales@newflow.co.uk

Document Information

Document Name:	Modbus Master Simulator User Manual
Document Identifier	MMS-UM

This document, submitted in confidence contains proprietary information, which shall not be reproduced or transferred to others without prior written permission of Newflow Limited

This document shall not form part of any contract. Specifications are subject to change without notice, and Newflow Ltd accepts no liability of any kind for errors or omissions.

© Newflow Ltd 2025

Full contractual terms are available on the website at www.newflow.co.uk/pdf/tandc.pdf

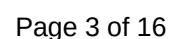


Table of Contents

1 Introduction.....	<u>5</u>
2 Installation.....	<u>5</u>
3 Launching the Program.....	<u>6</u>
4 The Main Window.....	<u>7</u>
4.1 Loading/Saving Polls.....	<u>8</u>
4.2 Logging.....	<u>8</u>
4.3 Disabling Poll Values and/or Debug Messages Windows.....	<u>9</u>
4.4 Debug Messages Window.....	<u>9</u>
4.5 Poll Values Window.....	<u>10</u>
4.6 Creating a New Poll.....	<u>12</u>
4.7 Sequence Control.....	<u>14</u>
4.8 Frame Counters.....	<u>15</u>
5 Revision History.....	<u>16</u>

1 Introduction

The Newflow Modbus Master Simulator program can be used for exercising or validating a Modbus Slave device, such as a Coriolis meter, a Gas Chromatograph or a flow computer slave port. The program runs on a Windows PC and supports both serial and Ethernet Modbus/TCP communications.

The simulator can issue a sequence of polls. Each poll can be a read or write to one or multiple registers and coils. Function codes 01, 02, 03, 04, 05, 06, 15 & 16 are supported, along with the Function Code 08 loopback.

All the items in any one poll will be of the same data type, which can be Int16, UInt16, Int32, UInt32, Int64, UInt64, Float32, Double64, or a String. Each item in a poll can be given a description for the convenience of the operator.

Live values are displayed in decimal or hexadecimal format in the Poll Values window (selectable per poll). The raw hexadecimal message data is displayed in the Debug Messages window. Both the Poll Values and Debug Messages windows can be shown or hidden. If hidden, this allows faster polls to be set when stress testing the attached device.

The simulator can handle up to 99 individual polls. These polls can be saved to the PC and loaded again for subsequent use. The files are in XML format and can be edited in a text editor if required.

A sequence engine is included which allows multiple polls to be linked together, and control the rate that the polls are transmitted.

2 Installation

The Modbus Master Simulator program is a single executable program and does not require an installer as it only uses a single (auto generating) initialization file to operate.

The file can be placed in any folder, including the desktop, a USB memory stick or a network drive.

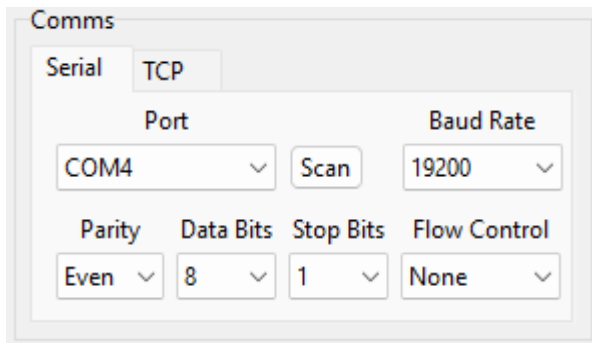
The first time the program is closed, it will create the ModbusMasterSimulator.ini file in the same folder as the simulator was run from. This file stores information such as the serial port settings and is used to pre-load the simulator next time it is launched.

3 Launching the Program

When the simulator loads, the main window is presented. Depending on whether the program has been run before will determine if the Comms settings are default ones or the ones remembered from the last time the program was run (as these values are stored in the local ModbusMasterSimulator.ini file).

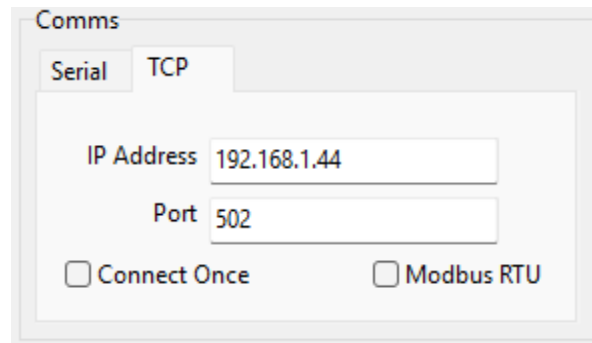
To select the transport mechanism required, choose the required tab between Serial (for Serial Communications) or TCP (for Ethernet TCP communications). Screenshots are shown for both options below;

SERIAL



The SERIAL Comms settings window shows the 'Serial' tab selected. It includes a 'Port' dropdown menu set to 'COM4', a 'Scan' button, and a 'Baud Rate' dropdown menu set to '19200'. Below these are four more dropdown menus: 'Parity' set to 'Even', 'Data Bits' set to '8', 'Stop Bits' set to '1', and 'Flow Control' set to 'None'.

TCP



The TCP Comms settings window shows the 'TCP' tab selected. It includes an 'IP Address' text field with '192.168.1.44' entered, and a 'Port' text field with '502' entered. At the bottom, there are two checkboxes: 'Connect Once' and 'Modbus RTU', both of which are currently unchecked.

The settings are as follows;

In Serial mode;

Select the Serial Port of the PC from the pulldown list given, then choose the Baud Rate, number of Data Bits, Parity, number of Stop Bits and Flow Control as required by the Modbus Slave device.

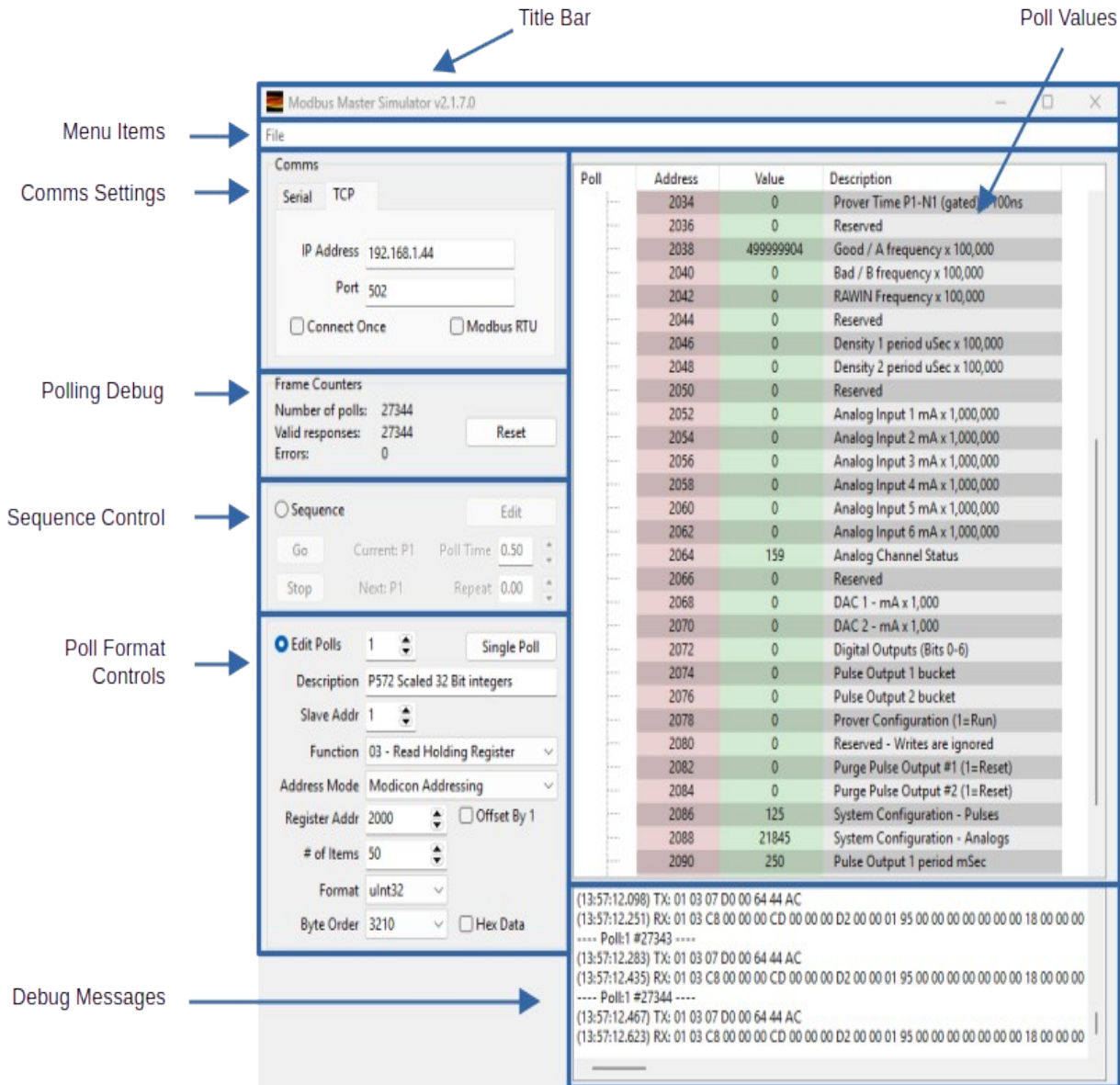
In TCP mode;

Enter the IP Address and Port of the Modbus Slave device. Previous settings are remembered however the port number defaults to 502 if no previous value is available. This is the commonly used Modbus TCP port number.

The Connect Once tick box configures the simulator to open the port on the Modbus Slave device and keep it open until the simulator is closed or the IP Address or Port are changed. If the tick is removed, the simulator will open and close the port on the Modbus Slave device for each poll.

The Modbus RTU tick box configures the simulator to pack/unpack the messages as though they were coming from a serial port. This is often termed 'Modbus over Ethernet' or 'Modbus/RTU' and is used when using Serial to Ethernet convertors. If the tick is removed, the simulator will transmit and decode messages using Modbus/TCP.

4 The Main Window



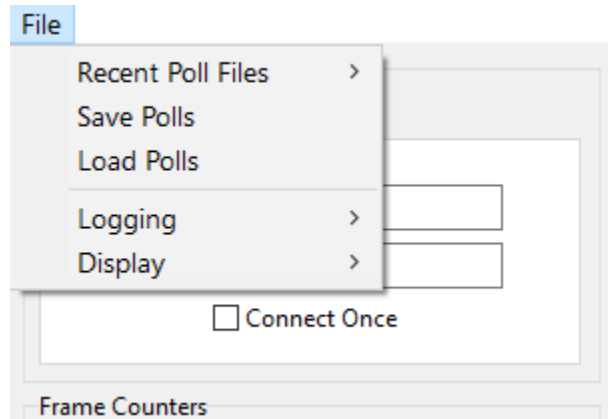
The title bar shows the name of the program and the version running. Below there is the File menu item.

The File menu has five options, these allow the saving and loading of poll files, sets options for logging and sets which windows are visible. See [4.1 Loading/Saving Polls](#), [4.2 Logging](#) and [4.3 Disabling Poll Values and/or Debug Messages Windows](#).

The Polls are constructed for each poll message in the Edit Polls (Poll Format Controls) section. Full details including the format are described in [4.6 Creating a New Poll](#).

The Sequence Control area is used to configure multiple polls, and is explained in detail in [4.7 Sequence Control](#).

4.1 Loading/Saving Polls



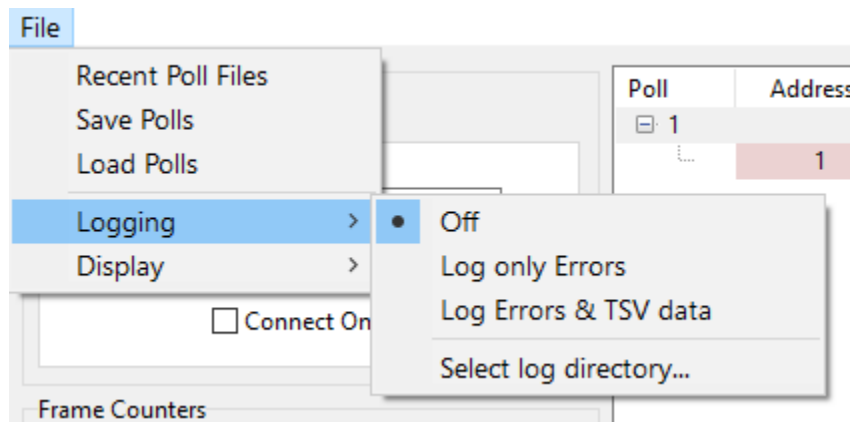
To save the work in building these polls each time the program is run, in the File menu, choose either Load Polls or Save Polls.

Select the folder that the polls are to be saved to or loaded from. If loading, select the file and if saving type the name of the file.

A shortcut is also provided under the File menu to quickly load Recent Poll Files.

NOTE: Files are stored with a .xml suffix.

4.2 Logging

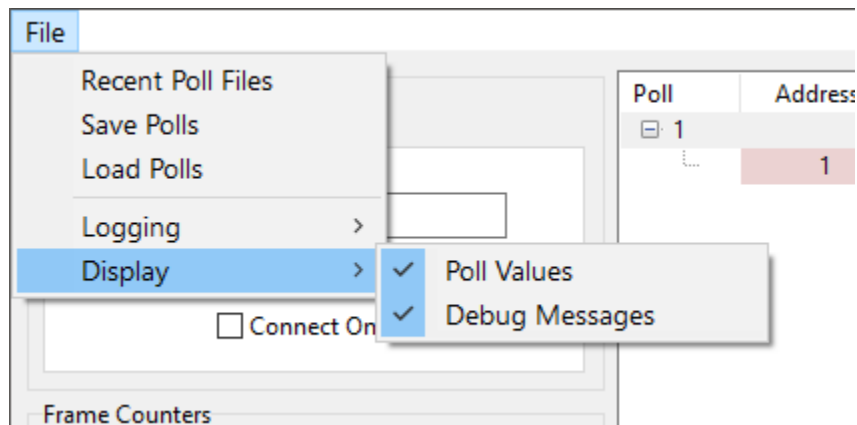


In the File menu, choose Logging and select the level of logging required by clicking on the relevant option, then choose the folder to store the log.

The Debug Messages window shows the last 1000 lines, whereas logging to a file will keep all messages.

4.3 Disabling Poll Values and/or Debug Messages Windows

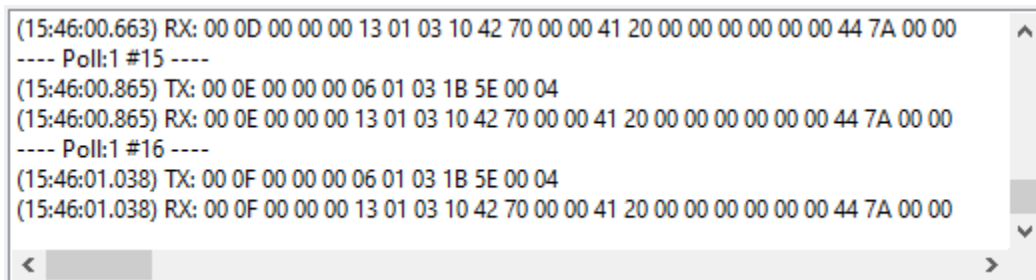
To hide the Poll Values or Debug Messages windows of the application, in the File menu, choose Display and tick/untick the windows required by clicking on the relevant option.



Disabling both of these two features will decrease the CPU load of the application, however little diagnostics are then available.

Decreasing the CPU load also allows polls to be performed quicker, with poll rates in excess of 100 per second achievable.

4.4 Debug Messages Window



This displays the sent and received Modbus Messages in their Hexadecimal form with timestamps. The Poll and number is displayed as a header above the message and response.

Individual lines can be selected by left clicking on them with the mouse/trackpad.

Multiple lines can be selected by left clicking and dragging the cursor, or using shift+click to select a range of lines.

Right clicking in the message window allows the user to copy all of the lines to the clipboard, copy just the selected lines to the clipboard, or to clear the window (useful when debugging to remove irrelevant entries).

4.5 Poll Values Window

The Display Table has two modes;

- Edit Polls mode
- Sequence mode

In either mode, the top row shows the headings, Poll, Address, Value and Description.

On the second row, under Polls it shows both the Poll number and an expand/collapse box. Clicking the box toggles this between + and – to hide or show the registers/addresses in the poll.

When collapsed, under the Value heading, the data type is shown and the Modbus function code (under the Description heading).

In Edit Polls mode, only the poll selected is shown. When in Sequence mode, all polls in the sequence are shown (as below):

Poll	Address	Value	Description
+ 1		uint32	Function: 03 - Read Holding Register
+ 2		uint32	Function: 16 - Preset Multiple Registers
+ 3		uint32	Function: 03 - Read Holding Register

In either Edit Polls mode or Sequence mode, if a poll is expanded, the next rows show the address/register information.

Poll	Address	Value	Description
- 1		uint32	Function: 03 - Read Holding Register
	2000	205	P513 Hardware Version
	2002	210	P513 Software Version
	2004	405	P572 Firmware Version

The Addresses will start at the Register Address selected and will increment as defined by the Address Mode for each subsequent address/register.

If the Number of items (in each poll) is increased, addition addresses will be shown.

On any new addresses/registers the Value fields will show as zero and the Descriptions will be blank.

On new or existing address/register information, double clicking in the Description field allows the user to add or edit the description text for each item.

If the poll is a read poll, the value field will show the value retrieved from the Modbus Slave device. If however, the poll is a write type poll, the value can be double clicked and the value set to the value required to be sent to the slave device.

The value field can be shown in decimal or hexadecimal notation. This is defined in the Edit Polls section for each poll. See [4.6 Creating a New Poll](#).

Untick/Tick the option to alternate the values. See examples below;

Decimal notation

Poll	Address	Value	Description
.....	2010	0	Digital Inputs (Bits: 0-8)
.....	2012	2048	Prover Status
.....	2014	323846	Message Id (2Hz)
.....	2016	0	Reserved

Hexadecimal notation

Poll	Address	Value	Description
.....	2010	0x00000000	Digital Inputs (Bits: 0-8)
.....	2012	0x00000800	Prover Status
.....	2014	0x0004F106	Message Id (2Hz)
.....	2016	0x00000000	Reserved

4.6 Creating a New Poll

Each poll can be a minimum of one datapoint up to a maximum of 99 datapoints.

Checks are made on the validity of the message being sent, with regard to the message length. Should this exceed the maximum length set out in the Modicon Modbus specification, the Single Poll button will grey out and the wording will change to 'Poll Too Big'.

The options vary depending on the format chosen as shown and detailed below.

Edit Polls 1 **Single Poll**

Description: P572 Scaled 32 Bit integers

Slave Addr: 1

Function: 03 - Read Holding Register

Address Mode: Modicon Addressing

Register Addr: 2000 ☐ Offset By 1

of Items: 50

Format: uInt32

Byte Order: 3210 ☐ Hex Data

Edit Polls 4 **Single Poll**

Description:

Slave Addr: 1

Function: 03 - Read Holding Register

Address Mode: Modicon Addressing

Register Addr: 4000 ☐ Offset By 1

of Items: 3

Format: String String Length: 16

Byte Order: 10 ☐ Hex Data

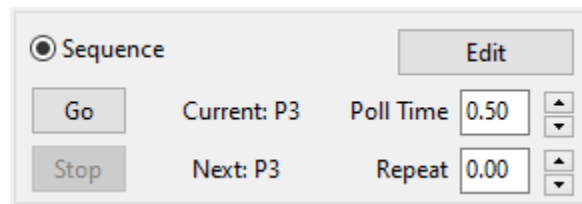
Edit Polls	This is a radio button. Selecting Edit Polls deselects the Sequence editing function and vice-versa.
POLL	Up to 99 different poll messages can be handled for any one instance. All of these poll messages can be saved, for subsequent use. Each poll message is given a number and a description can also be added for each poll to identify each poll if a polling sequence is generated.
Single Poll	Clicking this button causes the simulator to request the currently selected poll once. If enabled, the Poll Values and/or Debug Messages windows will also be populated. If the message defined is too big to be valid, this button will be greyed out so it cannot be pressed and the text will change to 'Poll Too Big'.
Description	This Description field allows a name to be associated with each poll. Keeping the name to less than 20 characters will ensure the whole name can be displayed in the Sequence Editor.
Slave Addr	This can be 1 to 99. There are up and down arrows to change the value or it can be typed in directly.
Function	This the Modbus function code. The simulator supports Functions 01, 02, 03, 04, 05, 06, 08, 15 & 16.
Address Mode	The Address Mode field is used to select Modicon or Logical Addressing. If the format below is set to uInt16 this will always be one address. However if the Format is set to uInt32, this would be a single address in Logical Addressing mode but two addresses in Modicon Addressing mode.

Register Addr	This is the Modbus Slave register address which data is being read from or written to.
Offset by 1	In Modicon Addressing mode, it is normal that the addresses are offset by 1, so if you are looking at 7001, the poll actually requests address 7000. This is not guaranteed though. A tick box is therefore provided to easily enable this or not, without the need to repopulate the map with new register address details.
# of Coils or # of Items	This is the number of items requested by each poll.
Format	This is a pull down box, which defaults to uInt16, but can be chosen between uInt16, uInt32, Int64, Float32, Double64 or String.
String Length	If the Format is set to String, this is the number of characters to be requested or sent. The addresses shown in the Poll Values window will then be updated accordingly.
Byte Order	The Byte Order options depend on the Format selected. For example uInt16 only has two bytes of data, so there are only two options, 10 (Big-Endian) or 01 (Little-Endian), whereas Float32 has four bytes of data therefore options of 3210, 0123, 1032 or 2301 are provided.
Hex Data	The value fields can be shown in decimal or hexadecimal notation. With the box ticked, hexadecimal notation is shown, unticked will show decimal notation.

NOTE: The Modbus Function, Data Type and Format will be the same for all registers within any one poll.

To add additional polls, increment the **POLL** number, and repeat the process above.

4.7 Sequence Control

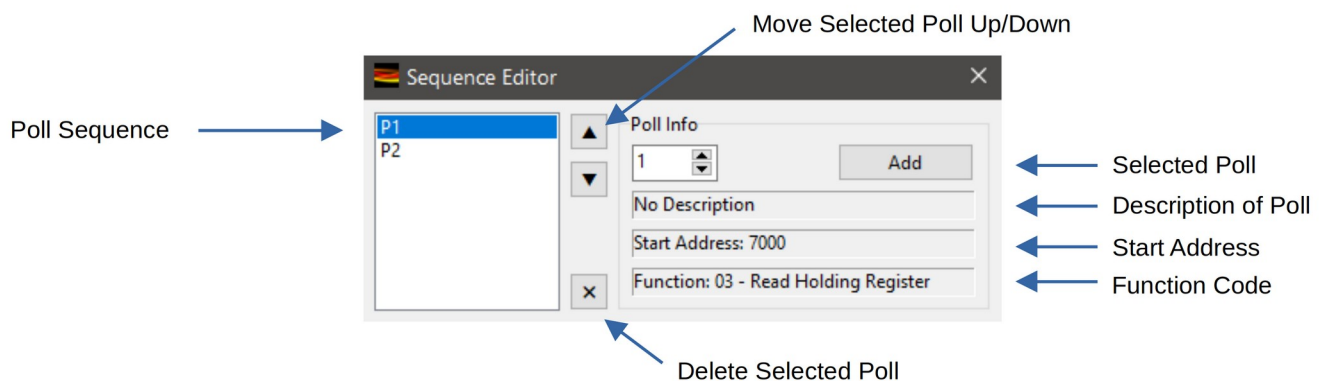


The Sequence Control dialog box features a 'Sequence' radio button and an 'Edit' button. Below these are 'Go' and 'Stop' buttons. The 'Current' field is set to 'P3', and the 'Poll Time' is '0.50'. The 'Next' field is also 'P3', and the 'Repeat' value is '0.00'. Up and down arrows are provided for the Poll Time and Repeat fields.

The sequence section allows the user to poll a number of register addresses in sequence.

To enter the sequence mode the “Sequence” radio button must be pressed, this deselects the Edit Polls section. If no current sequence is defined, the Sequence Editor will be automatically opened.

If a sequence has already been defined or the user wants to change the sequence, the Edit button should be pressed, this will also open the Sequence Editor.



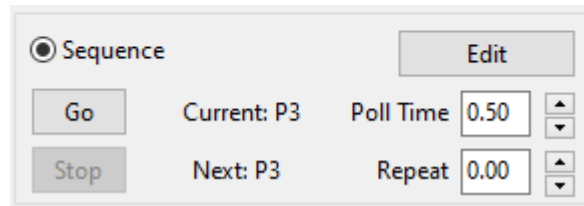
The left hand pane shows the poll sequence that is already configured. An empty list signifies that no sequence is currently defined.

To add polls to the sequence use the up and down arrows to increment the Poll value (or the value can be typed in directly to the number field). The poll description, start address and function code is shown to assist identification. Press the Add button to add the poll into the sequence.

If the poll sequence is in the incorrect order, each poll can be selected, then by using the up and down arrow buttons, can be moved up or down the polling order. Duplicate polls are also allowed so the user could define, as an example; P1, P2, P1, P3.

Polls can be removed from the sequence by selecting them in the list and pressing the X button (Delete).

Once the polling sequence is as required or the user wants to exit the editor, the X (Close) button in the top right hand corner of the Sequence Editor should be pressed.



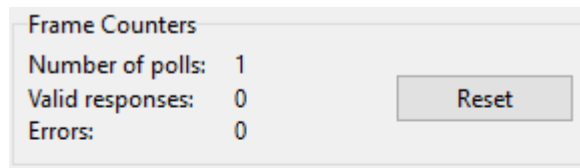
A configuration dialog box for a sequence. It features a radio button labeled "Sequence" which is selected. To its right is an "Edit" button. Below the radio button are two rows of controls. The first row contains a "Go" button, the text "Current: P3", the text "Poll Time", a text box containing "0.50", and a vertical spinner control. The second row contains a "Stop" button, the text "Next: P3", the text "Repeat", a text box containing "0.00", and a vertical spinner control.

In the main window, when a current sequence is defined, the polling can be started/stopped with the “Go” and “Stop” buttons. The “Go” will be greyed out and unable to press if no sequence is defined.

The “Poll Time” value is the time between polls. The “Repeat” value is the time after a sequence has completed till it starts again. These values can either be edited using the relevant up and down arrows or by typing a value in directly.

Between the “Go” and “Stop” button and the poll timers, a diagnostic is provided to show the current and next polls that are due to be performed.

4.8 Frame Counters



A dialog box titled "Frame Counters". It displays three rows of data: "Number of polls: 1", "Valid responses: 0", and "Errors: 0". To the right of these values is a "Reset" button.

The Frame Counters show the number of polls that have been performed, how many valid responses have been received and how many errors have been recorded.

A Reset button is also provided so that the values can be reset to aid diagnostics.

5 Revision History

Rev	Date	Changes	Prepared	Checked	Authorized
0	19 Aug 2014	Initial	GPL	Josh	GPL
1	9 Mar 2023	Program Version: 1v9	MOB	GPL	MOB
2	29 Jul 2024	Program Version: 2.1.1.0	GPL	MPFJ	GPL
3	12 Nov 2024	Program Version: 2.1.2.0 Bugfix to Fn6 & Fn16 – no interface or operational changes	GPL	GPL	GPL
4	22 Jul 2025	Program Version: 2.1.4.0 Various Bugfixes, added Hex/Decimal display format	GPL	MOB	MOB
5	26 Sep 2025	Program Version: 2.1.7.0 Various Bugfixes Add Modbus/RTU mode in TCP Allow Hex/Decimal display format per poll Enhance error reporting	GPL	MPFJ	MPFJ
6					

End of Document