

PollMB and PollWS

Table of Contents

[Overview:](#)

[Supported Modbus Functions:](#)

[Installation:](#)

[Installing Python:](#)

[Command Line Parameters:](#)

[Ethernet Address Parameters](#)

[Modbus parameters](#)

[Polling parameters](#)

[Sending Data:](#)

[Return Codes:](#)

[Using PollMB:](#)

[Using PollWS:](#)

[Polling performance measurement:](#)

Overview:

PollMB is a simple Modbus/TCP command line client with the following features:

- It can read from and write to a Modbus/TCP server.
- It supports the most commonly used Modbus client functions.
- All parameters are configurable.
- It can poll repeatedly, to allow load testing of a server.

PollWS is a simple client which is similar to PollMB, but supports a Modbus-like web service.

Supported Modbus Functions:

The following Modbus functions are supported:

- 1 - Read Coils.
 - 2 - Read Discrete Inputs.
 - 3 - Read Holding Registers.
 - 4 - Read Input Registers.
 - 5 - Write Single Coil.
 - 6 - Write Single Holding Register.
 - 15 - Write Multiple Coils.
 - 16 - Write Multiple Holding Registers.
-

Installation:

MBPoll will come as an archive file in “tar-gz” (most platforms) or “zip” (MS-Windows) formats. The MS-Windows version has the special new-line character combination (CR-LF) required by that platform, but is otherwise identical to the standard version.

Extract the archive file into the desired location. MBPoll doesn't care where you place it.

Installing Python:

This is the language support for Python which is required for MBPoll to run. MBPoll has been tested with version 2.5. Other versions may work but have not been tested. For Linux, Python is often already installed. If not, it can normally be installed from the distro's package manager. For MS-Windows, Python can be downloaded from <http://www.python.org>

Command Line Parameters:

Address, function codes, and other values are specified by means of command line parameters. If a parameter is omitted, its default value is used. The parameters are:

Ethernet Address Parameters

Parameter	Description	Default
h	Host name of the Modbus/TCP server	localhost
p	Port number of the Modbus/TCP server	502 (PollMB) 80 (PollWS)
t	Receive time-out in seconds	60 (minimum = 1)

Note: -h (host). For PollMB, this is the IP address or domain name. For PollWS, this is the URL to the web service, including IP address and any additional URL path.

Note: -t (time-out) is accepted by PollWS, but is ignored.

Modbus parameters

Parameter	Description	Default
f	Function	1
a	Address (Modbus memory)	0
q	Quantity of addresses	1
u	Unit ID	1

Polling parameters

Parameter	Description	Default
r	Repeats. Number of times to perform the poll.	1
y	Delay time between repeats (in milliseconds)	1
d	Data to send to the server.	0000
s	Silent mode. 'Y' or 'y' will suppress displaying data.	'n' (display)

Note: Actual delay time is affected by the resolution of the operating system process scheduler.

Sending Data:

- For functions 1, 2, 3, or 4, any data specified is ignored.
 - For functions 5, 6, 15, or 16, data must be specified.
 - For function 5 (write single coil), data must be either 0 or 1.
 - For function 15 (write multiple coils), data must be 0 and 1 characters, in multiples of 8 characters. E.g. 00111010.
 - For functions 6 and 16 (write single or multiple registers), data must be in hexadecimal with 4 characters per register. Hexadecimal values a to f may be in lower or upper case. E.g. 23aD
-

Return Codes:

The following command line return codes are defined:

Return Code	Description
0	No errors.
2	Bad command line.
3	Program was terminated from keyboard.

Return Code	Description
4	Invalid data for modbus function.
5	Error communicating with host.
6	Invalid modbus parameters.

Using PollMB:

PollMB is normally used from the command line. For Linux, the syntax is:

```
./pollmb.py (parameters)
```

For Microsoft Windows, the syntax is:

```
c:\python25\python pollmb.py (parameters)
```

A typical example is:

```
./pollmb.py -h 192.168.10.1 -p 8502 -t 10 -f 16 -a 6789 -q 2 -u 45 -d 0123abcd
```

The above example polls the Modbus/TCP server at IP address 192.168.10.1 using port 8502 and sets a time out of 10 seconds. It requests function 16 (write multiple registers), address 6789, and unit ID 45. It requests to write data 0123abcd. The first four data digits are written to the first register, and the next for to the second register.

Another example is:

```
./pollmb.py -h localhost -p 502 -f 1 -a 6 -q 500
```

The above polls the server at localhost (located on the same computer as MBPoll), port 502. It requests function 1 (read coils), coil address 6, quantity 500. It uses the default values for all other parameters.

Using PollWS:

PollWS uses a similar syntax. For Linux, the syntax is:

```
./pollws.py (parameters)
```

For Microsoft Windows, the syntax is:

```
c:\python25\python pollws.py (parameters)
```

A typical example is:

```
./pollws.py -h 192.168.10.1/modbus -p 8502 -t 10 -f 16 -a 6789 -q 2 -u 45 -d 0123abcd
```

The above example is similar to that shown above with MBPoll. However, note that the IP address includes the path to the address endpoint (92.168.10.1/modbus).

Polling performance measurement:

- When used to measure polling performance, silent mode should be enabled.
- When more than one repeat is specified, extra information is displayed after polling is completed. This includes elapsed time, number of data elements transferred, and the data transfer rate.
- When trying to measure maximum speed a large enough number of polls should be specified to allow measurement over several (e.g. 5 to 10) seconds in order to get an accurate reading.