Control UI For Modbus TCP

2023 by Vincent Lao

Contents

[Introduction 3](#_Toc130217592)

[Features 3](#_Toc130217593)

[What is supported 3](#_Toc130217594)

[Future development 3](#_Toc130217595)

[Controls 4](#_Toc130217596)

[Configuration file 6](#_Toc130217597)

[Configuration Tool 10](#_Toc130217598)

[Starting the Tool 10](#_Toc130217599)

[Usage 11](#_Toc130217600)

[Creating a new configuration 11](#_Toc130217601)

[To edit a line 12](#_Toc130217602)

[Delete a control 12](#_Toc130217603)

[Modify an existing config file 12](#_Toc130217604)

[Future To Dos 13](#_Toc130217605)

# Introduction

Control UI for Modbus TCP is a tool for rapid development of a basic UI for testing device communication via Modbus TCP. It is designed for easy use of the PC to control a Delta PLC or other Modbus device. The user needs to supply the device IP, Modbus address and bit position, access type and Label to be able to use the UI.

## Features

* User customizable config file. Config is just a CSV format
* Tool to create a config file
* Simple UI
* There is no need to write a single line of code in order to communicate with a Delta PLC via a Windows UI.
* Scriptable

## What is supported

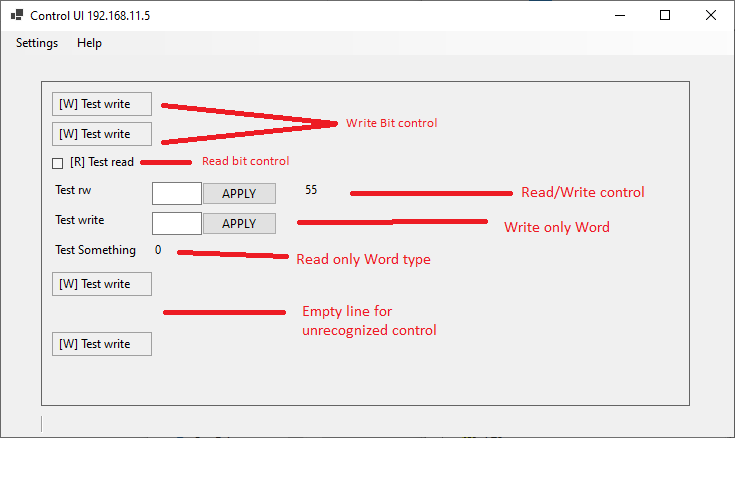
* Bit control (Boolean)
* Word control (Word)
* Reading and Write

Other data types are just derived from these 2 types, eg. 32 bit word.

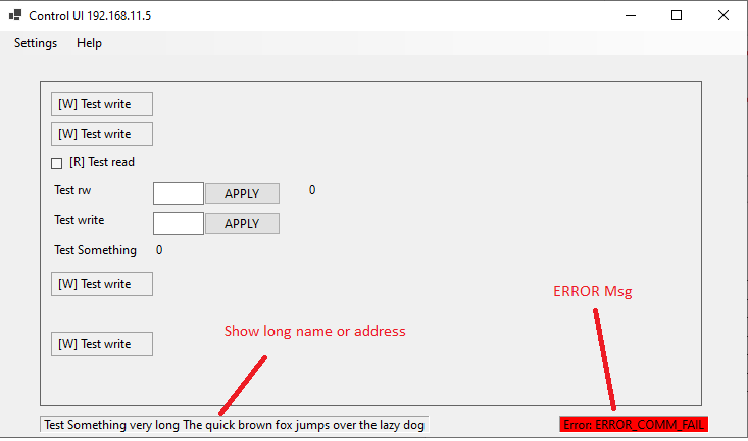
## Future development

* Scriptable 2 way communication (planned to be added later)
* Reconnection on communication issue
* Better looking interface
* More customizability
* Support for multiple configuration file (no need to restart app)

# Controls



|  |  |  |  |
| --- | --- | --- | --- |
| **Data type** | **Access** | **Control** |  |
| BIT | Read only | Checkbox will be checked |  |
|  | Write only | Toggle button by clicking |  |
| WORD | Read only | Value read from PLC |  |
|  | Write only | Type the value and click Apply |  |
|  | Read/Write | Combination of Read and Write |  |



# Configuration file

The configuration file is called *ControlUI.conf*. It is automatically loaded when the application is started. The configuration file can be written by hand, extracted from an Excel sheet as CSV or generated by the configuration maker tool. Using the Configuration tool is **recommended** to avoid errors.

The file is just a text file with the format:

**IP,***<device IP>*,*<port>*

**CTRL,***<WO/RO/RW>,<WORD/BIT>,<Address>[.<bit position>],<1=toggle,2=input>,<Label...>*

…

***Device IP*** – IP address (IPv4 only)

***Port*** – port number, usually 502

***WO/RW/RW*** – Write only, Read only, or Read-write

***WORD/BIT*** – 16 bit WORD or 1 BIT

***Address[.bit position]*** – Modbus address. For BIT control, the bit position must be specified, range is 0-15. For bit position, 0 is the most significant bit. The address will be shown on the UI by hovering the mouse over the control. For example: 4000.0 is first bit of address 4000.

***1/2/3*** – 1 is a toggle control (for BIT), 2 is an input control or output for WORD, 3 is similar to 2 but using hexadecimal input and output

***Label*** – is any text used to indicate what control it is used for. If the text is very long, it will only the first portion of it. Hovering the mouse over the label will show the entire text at the bottom. This is designed to avoid cluttering the screen with too much text.

The current maximum is 200 input/output controls. This can be easily increased. However, the current screen size of 1080p can fit 100 controls.

Any keyword that is not recognized or line with missing important data is ignored (treated as comment).

Example config file:

IP,192.168.11.5,502

CTRL,WO,BIT,100.0,1,Test read bit

CTRL,WO,BIT,100.1,1,Test read bit

CTRL,RO,BIT,101.1,1,Test read bit 2

CTRL,RW,WORD,200,2,Test rw

CTRL,WO,WORD,300,2,Test write

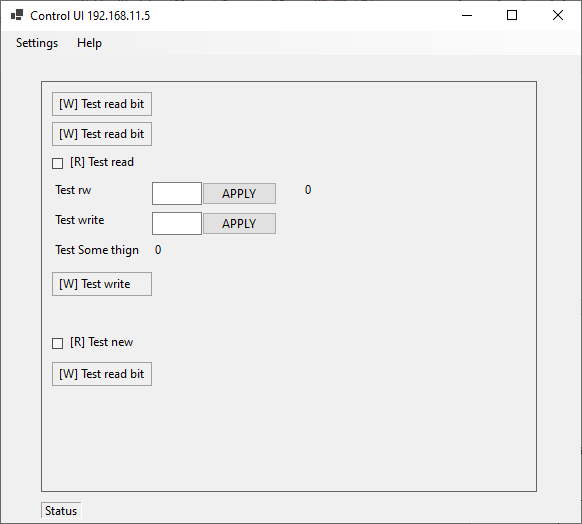
CTRL,RO,WORD,400,2,Test Some thign very long The quick brown fox jumps over the lazy dog

CTRL,WO,BIT,500.1,1,Test write

CTRL,RW,XXX,0,0,Test rw (incomplete data will be ignored)

CTRL,RO,BIT,600.7,1,Test new entry creation

CTRL,WO,BIT,4000.0,1,Test read bit



Example 2:

IP,192.168.11.5,502

LABEL,PLC Type B

CTRL,WO,WORD,4000,2,Valve %,: 1

CTRL,WO,BIT,4465.0,1,Valve On,: 1

CTRL,WO,WORD,4001,2,Valve %,: 2

CTRL,WO,BIT,4465.1,1,Valve On,: 2

CTRL,WO,WORD,4002,2,Valve %,: 3

CTRL,WO,BIT,4465.2,1,Valve On,: 3

CTRL,WO,WORD,4003,2,Valve %,: 4

CTRL,WO,BIT,4465.3,1,Valve On,: 4

CTRL,WO,WORD,4004,2,Valve %,: 5

CTRL,WO,BIT,4465.4,1,Valve On,: 5

CTRL,WO,WORD,4005,2,Valve %,: 6

CTRL,WO,BIT,4465.5,1,Valve On,: 6

CTRL,WO,WORD,4006,2,Valve %,: 7

CTRL,WO,BIT,4465.6,1,Valve On,: 7

CTRL,WO,WORD,4007,2,Valve %,: 8

CTRL,WO,BIT,4465.7,1,Valve On,: 8

CTRL,WO,WORD,4008,2,Valve %,: 9

CTRL,WO,BIT,4465.8,1,Valve On,: 9

CTRL,WO,WORD,4009,2,Valve %,: 10

CTRL,WO,BIT,4465.9,1,Valve On,: 10

CTRL,WO,WORD,4010,2,Valve %,: 11

CTRL,WO,BIT,4465.10,1,Valve On,: 11

CTRL,WO,WORD,4011,2,Valve %,: 12

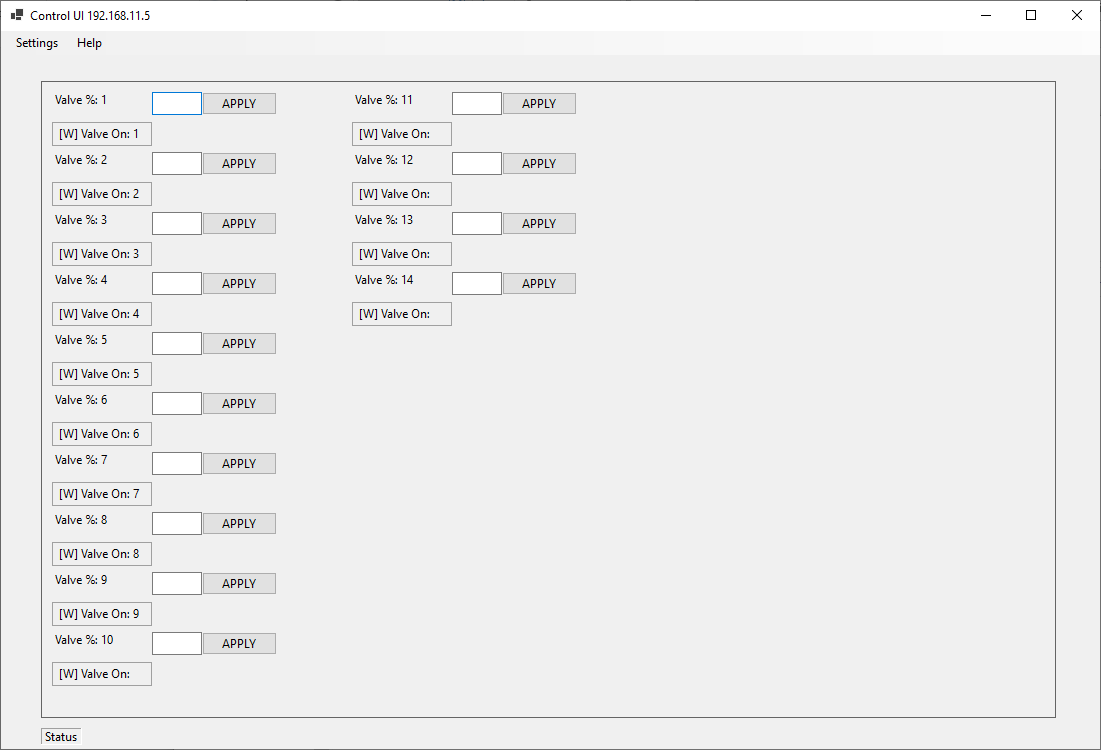
CTRL,WO,BIT,4465.11,1,Valve On,: 12

CTRL,WO,WORD,4012,2,Valve %,: 13

CTRL,WO,BIT,4465.12,1,Valve On,: 13

CTRL,WO,WORD,4013,2,Valve %,: 14

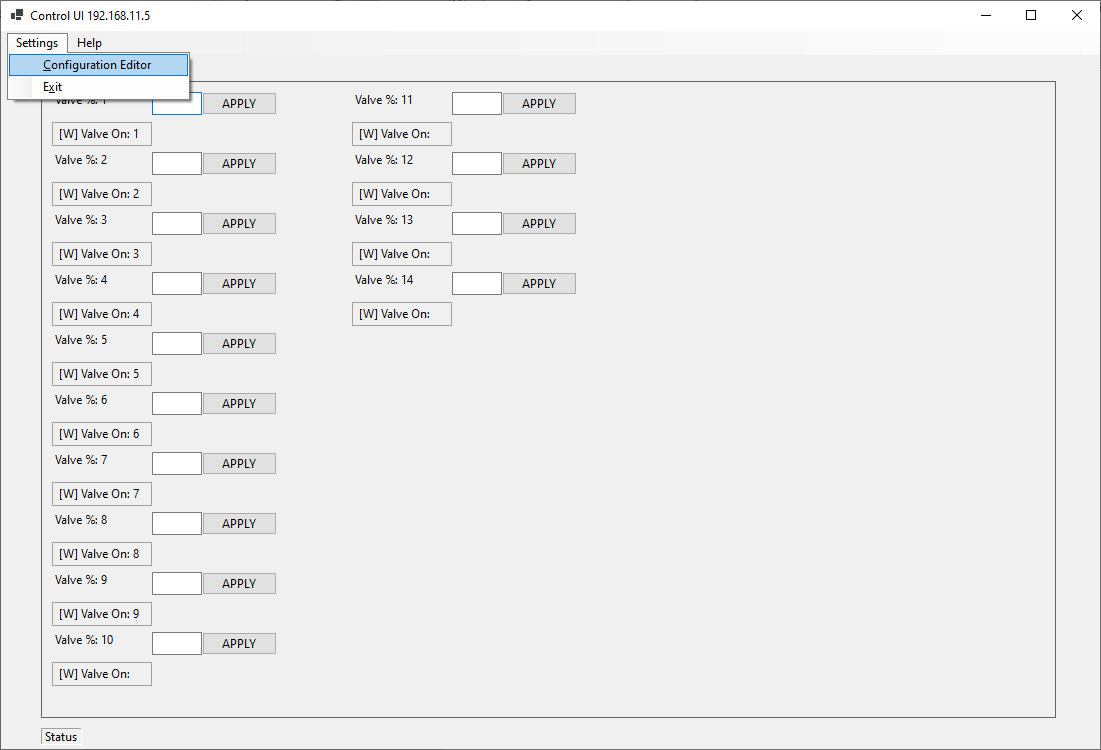
CTRL,WO,BIT,4465.13,1,Valve On,: 14



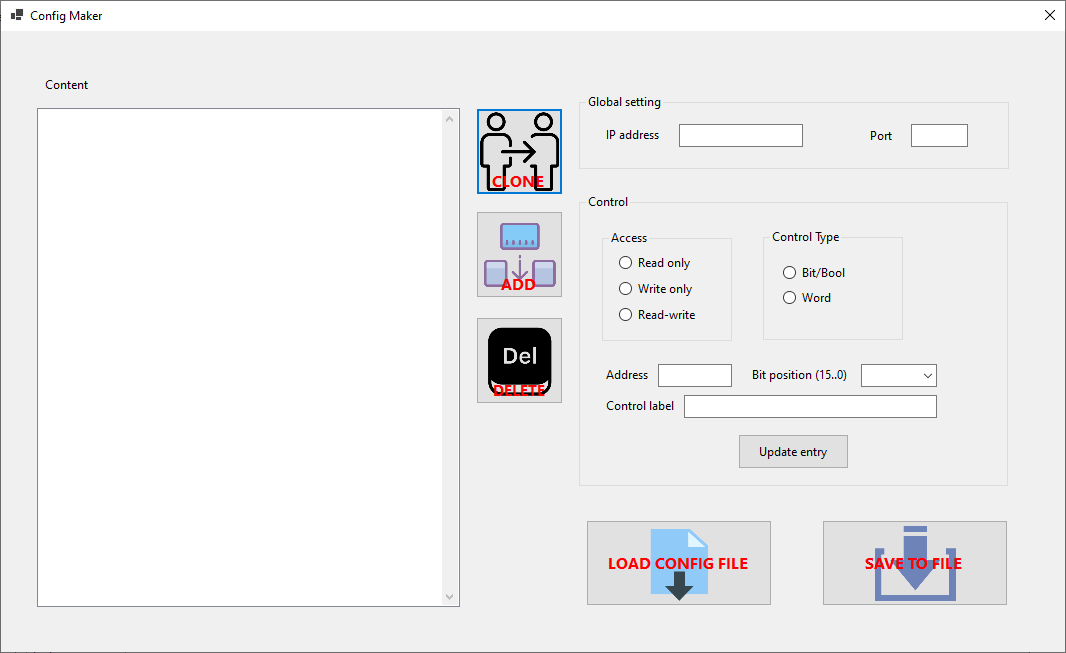
# Configuration Tool

## Starting the Tool

Go to Setting -> Configuration Editor to use the Tool for configuration maker.

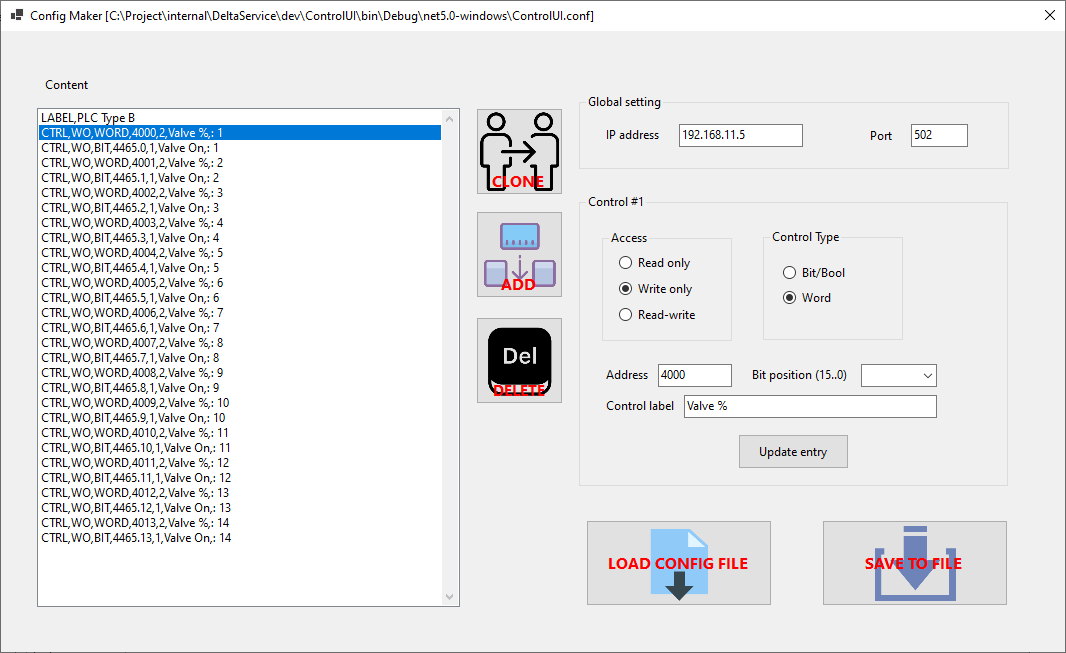


It will show another window.



By default, the configuration tool will have a blank entry. You can start a new configuration file or load an existing configuration file to edit.

## Usage



Note that the Control UI currently only starts with ***ControlUI.conf***. If you wish to use a different config file, rename the current one to something else and rename the new config file to ***ControlUI.conf***.

### Creating a new configuration

To create a new configuration, do not load an existing configuration file. Then follow these steps:

1. Specify the IP and Port
2. Click **ADD** button.
3. Make sure a line is selected in the Content box
4. Select the Access type (read only, write only, read-write)
5. Select the control type (Bit or Word)
6. Specify the Modbus address and bit position (for bit control only)
7. Specify the Label
8. Click **Update entry**
9. Repeat steps 2-7 until the desired number of controls are created
10. Click **SAVE TO FILE** and either use a different file name or overwrite the old file.

### To edit a line

1. Click the specific line
2. Change the parameters
3. Click **Update entry**
4. Repeat
5. Click **SAVE TO FILE**

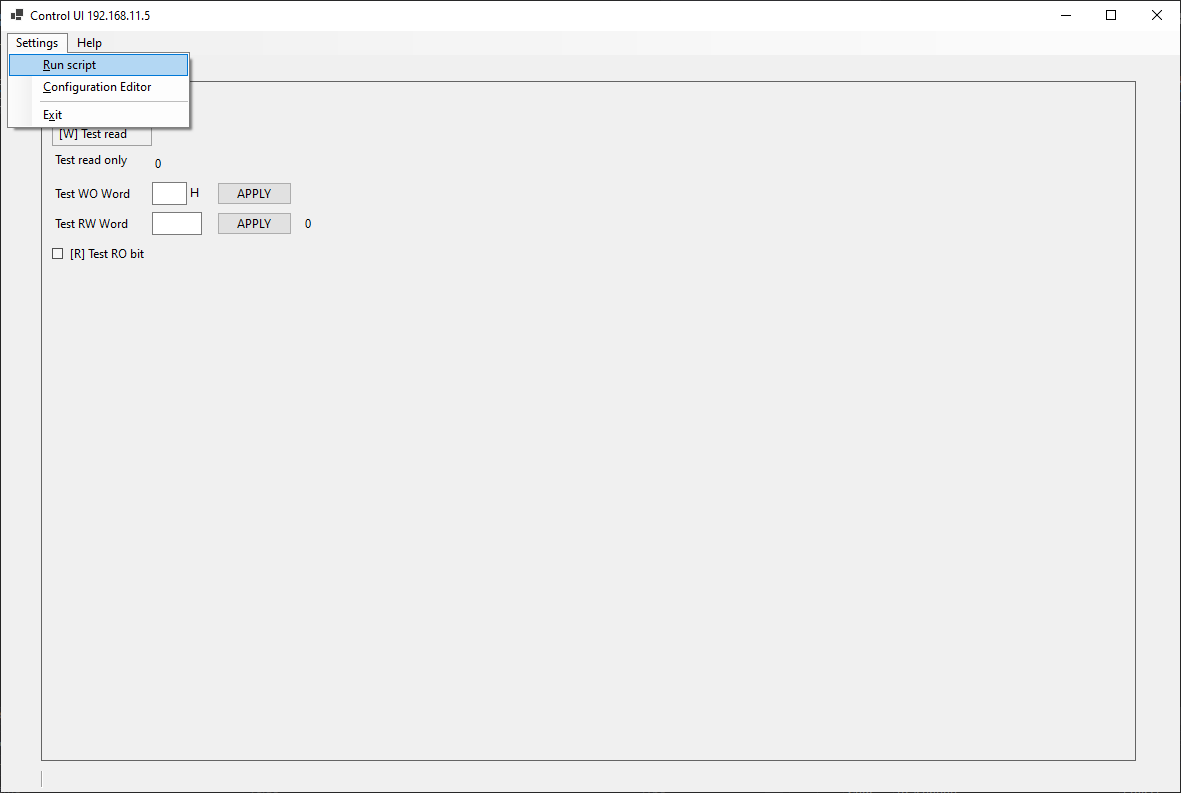
### Delete a control

1. Select the line to be deleted
2. Click the **DELETE** button
3. Click **SAVE TO FILE**

### Modify an existing config file

1. Click LOAD CONFIG FILE
2. Select the desired config file and it will be loaded.
3. Add, Modify, Delete as needed
4. Click **SAVE TO FILE**

## Scripting



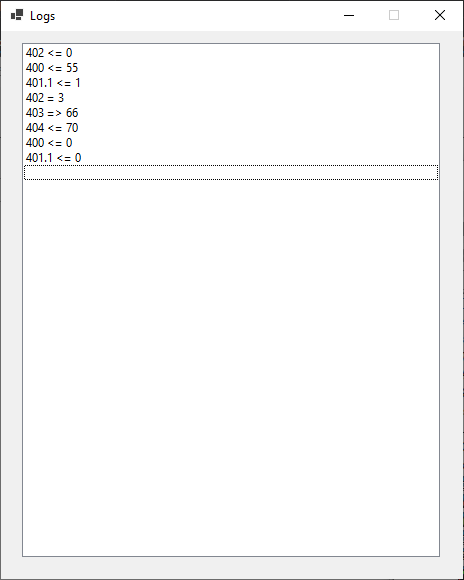
To use the scripting function, write a script file and load it via **Settings -> Run script**, then select the file to run.

The script syntax is described below.

The script file is a normal text file and is executed one line at a time. The command list is listed below:

|  |  |  |
| --- | --- | --- |
| **Keyword** | **Parameters** | **Description** |
| WAIT | * Address * Target value * Timeout (unused) | Wait (keep on reading from address) until a certain value is received. Timeout is currently not implemented |
| WAITBIT | * Address & bit position (eg 4000.1) * Target value (1 or 0) * Timeout | Wait (keep on reading from address and bit position) until a 1 “true” or 0”false” is received. Timeout is currently not implemented |
| SET | * Address * Value | Set a value at given address |
| GET | * Address | Get the value at address given |
| SETBIT | * Address & bit position (eg. 100.0) * Value (1 or 0) | Set bit at address & bit position. 1 = true, 0 = false |
| GETBIT | * Address & bit position (eg 120.2) | Reading for a particular bit at address and bit position specified. |

Note that reading something will only be displayed in the log.



This log is the result of this short script:

SET 402 0

SET 400 55

SETBIT 401.1 1

WAIT 402 3 0

GET 403

SET 404 70

SET 400 0

SETBIT 401.1 0

This sample script will do the following:

1. Clear address 402
2. Set 400 with value of 55
3. Set 2nd bit of address 401 to 1 (value of 2)
4. Wait until address 402 becomes 3
5. Read address 403
6. Set 404 to 70
7. Clear 400
8. Clear 2nd bit of 401

# Future To Dos

Currently, the module for logging Modbus data to file, database or other medium is present, but no implementation is done.