

MODBUS TCP Client Simulator

Introduction

The MODBUS TCP Client Simulator communicates to any Modbus TCP Server.

You can read from or write to multiple registers of your MODBUS device. Moreover, this can be done in a sequence simulating a small PLC program.

Supported P+F Hardware

The program can be used with a Pepperl+Fuchs or any other device that supports MODBUS TCP. The device must support one of two commands. Command 3, Read Holding Registers, and Command 16, Write Multiple Registers.

ICE2 and ICE3 IO-link masters

- ICE2-8IOL-G65L-V1D
- ICE2-8IOL-K45P-RJ45
- ICE2-8IOL-K45S-RJ45
- ICE3-8IOL-G65L-V1D
- ICE3-8IOL-K45P-RJ45
- ICE3-8IOL-K45S-RJ45
- ICE3-8IOL-G65L-V1D-Y

MODBUS gateway (DeviceMaster)

- ICDM-RX/MOD-2DB9/RJ45-DIN
- ICDM-RX/MOD-2ST/RJ45-DIN
- ICDM-RX/MOD-4DB9/2RJ45-DIN
- ICDM-RX/MOD-DB9/RJ45-DIN
- ICDM-RX/MOD-DB9/RJ45-PM
- ICDM-RX/MOD-ST/RJ45-DIN

RFID control interfaces:

- IC-KP-B17-AIDA1
- IC-KP2-1HB17-2V1D
- IC-KP2-2HB17-2V1D

Vision sensors:

- VOS1000-F226R-8MM-S
- VOS1000C-F226W-8MM-S
- VOS2000-F226-C-S
- VOS2000-F226R-8MM-S
- VOS5000-F227-C-S

AS-Interface gateways:

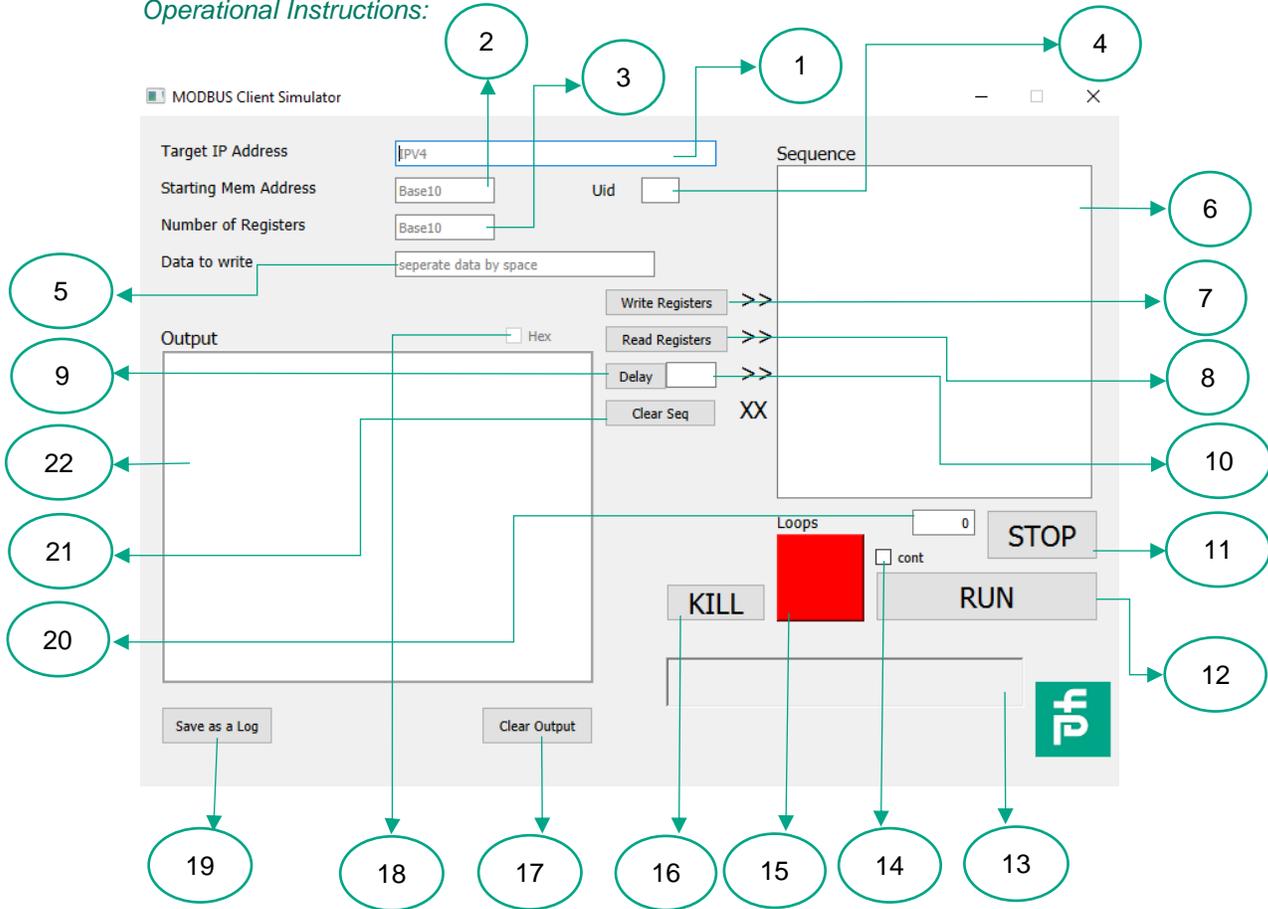
- VBG-ENX-K20-D-EV24
- VBG-ENX-K20-DMD-EV
- VBG-ENX-K30-DMD-S16-EV
- VBG-ENX-K20-DMD
- VBG-ENX-K30-DMD-S16
- VBG-ENX-K20-D

Installation

The program is an executable file that can be run directly after extracting the downloaded folder. No installation is required.

Platform tested on: Windows 10

Operational Instructions:



1. The IP Address input field takes a valid IP address in the format x.x.x.x (x = 0-255). If an invalid IP address is entered, the message "Invalid IP address" appears in the status bar.
2. The starting address of the intended read/write registers is entered in decimal format in the "Starting Memory Address" field. MODBUS addressing can have a value from 0 to 9999.
3. The Number of Registers field specifies the number of MODBUS registers to be read.
4. Uid specifies the unit ID for the server, generally it is kept at 1.
5. The Data to write field contains the data that will be used to perform the MODBUS write operation. The data should be in decimal format from 0 to 255 and separated by a space. Each number that is separated by a space specifies the data to be written to a single register with respect to the starting address.
6. The Sequence text box displays the current sequence of instructions being executed. The instructions include read/write register(s) and the delay function.
7. The Write Register button must be clicked to include a write command in the sequence. MODBUS Command 16 is used.
8. The Read Register button must be clicked to add a read instruction to the sequence. MODBUS Command 3 is used.
9. The Delay input field contains an integer decimal value specified in milliseconds.
10. The Delay button must be clicked to include the delay command in the sequence.
11. The STOP button is used to disconnect the running MODBUS connection. (Note: In v1.0, a process running in a continuous loop or with a high loop number cannot be stopped with the STOP button, and the STOP button is grayed out. In this case, the KILL button must be used.
12. The RUN button is used to start a MODBUS connection and the instructions in the sequence.
13. The status bar shows the status and errors if an invalid value is used.

14. The cont checkbox lets the sequence run indefinitely.
15. The status indicator lights green when the program is connected to the server and red when it is not.
16. The KILL button is used when a running sequence of commands or a long running sequence must be stopped.
17. The Clear Output button clears the read values from the output field.
18. The Hex checkbox converts decimal data in the output field to hex. This can also be done during read and write operations.
19. The Save as a Log button saves the output data in a txt log file in the same application directory. The file is named with the date of the running program. If several files with the same date are created, the file names must be changed, otherwise the existing log file will be overwritten.
20. The Loops field specifies how often the program will be executed. The default value is 0, which means that the program is executed once.
21. The Clear Seq button deletes all entered instructions in the sequence field.
22. The Output field shows the response of the server when a read instruction is executed. This is only input data coming from the MODBUS TCP server.

Simulation Example: Flashing IO-link Stack Lights

Requirements:

- ICE2/3 master
- Signal tower connection element IO-Link VAZ-CLAMP-40MM-IO
- Two VAZ-LED-40MM-CLD-XX (any color of preference)
- MODBUS Client Simulator

Setup:

Make sure your ICE2/3 is powered and that the stack lights are connected to one of the available ports.

The ICE2/3 is on the same network as your laptop/PC. To verify this, you should be able to open the ICE2/3 block's web interface from your web browser.

You need to enable MODBUS TCP/IP in your ICE2/3 web configuration. If this option is disabled, the device will not be able to communicate via MODBUS.

CONTROL Home Diagnostics Configuration Advanced Attached Devices Support ICE2-300-449-030 Logout

IO-LINK ETHERNET/IP MODBUS/TCP OPC UA MQTT NETWORK MISC LOAD/SAVE CLEAR SETTINGS

Modbus/TCP Settings

MODBUS/TCP PORT CONFIG	PORT 1	PORT 2	PORT 3	PORT 4	PORT 5	PORT 6	PORT 7	PORT 8
ISDU Data Settings:								
ISDU Response Timeout (1 - 10000)	20 sec							
Process Data Settings:								
PDI Data Block Size (To PLC)	36 bytes	8 bytes	36 bytes	36 bytes	36 bytes	36 bytes	36 bytes	36 bytes
PDI Byte-Swap Method	no byte-swap							
PDO Data Block Size (From PLC)	32-bytes							
PDO Byte-Swap Method	no byte-swap							
Append PDO to PDI Data	false							
Clear Event Code In PDO Block	false							
Clear Event Code After Hold Time	true							
Active Event Hold Time (1 - 65535)	1000	1000	1000	1000	1000	1000	1000	1000
Event Hold Time Units	ms							
Clear Event Hold Time (1 - 65535)	500	500	500	500	500	500	500	500
Event Clear Time Units	ms							
Transfer Mode Settings:								
Slave Mode Device ID (1 - 247)	1	1	1	1	1	1	1	1
PDI Receive Mode(s) (To PLC)	Slave							
MODBUS/TCP CONFIGURATION	enable							
Modbus Enable	enable							

Load the IODD file for your stack light and verify in attached devices to see if the device is displayed on its corresponding port.

CONTROL Home Diagnostics Configuration Advanced Attached Devices Support ICE2-300-449-030 Logout

IODD FILES SUMMARY PORT 1 PORT 2 PORT 3 PORT 4 PORT 5 **PORT 6** PORT 7 PORT 8

IO-Link Device - Port 6

Parameter Name	Index	Subindex	Value	Description	R/W	Unit	Min	Max	Comments	Gradient	Offset	Data Type	Simple Datatype	Bit Length	Fixed Length
- Identification															
- Device Information															
Vendor Name	16		Pepperl+Fuchs		RO							StringT			64
Vendor Text	17		www.pepperl-fuchs.com/io-link		RO							StringT			64
Product Name	18		VAZ-CLAMP-40MM-IO		RO							StringT			64
Product Text	20		Signal Lamp		RO							StringT			64
Product ID	19		324822		RO							StringT			64
Serial Number	21		0000015748070022		RO							StringT			16
- Revision Information															
Hardware Version	22		HW01.00		RO							StringT			64
Firmware Version	23		FW01.42		RO							StringT			64
- Application Specific Information															
Application Specific Tag	24		Your automation. Our passion.		RW							StringT			32
+ Parameter															
+ Observation															
+ Diagnosis															

IO-Link Device ISDU Interface - Port 6 Port Status: Operational, PDO Invalid

Now open the MODBUS Client Simulator and you can start flashing some stack lights.

The Target IP Address will be the IP of your ICE2/3 module.

The stack light in this demo is connected to port6 and the corresponding PDO (write) address can be found in the MODBUS Address Definitions in the ICE2/3 manual.

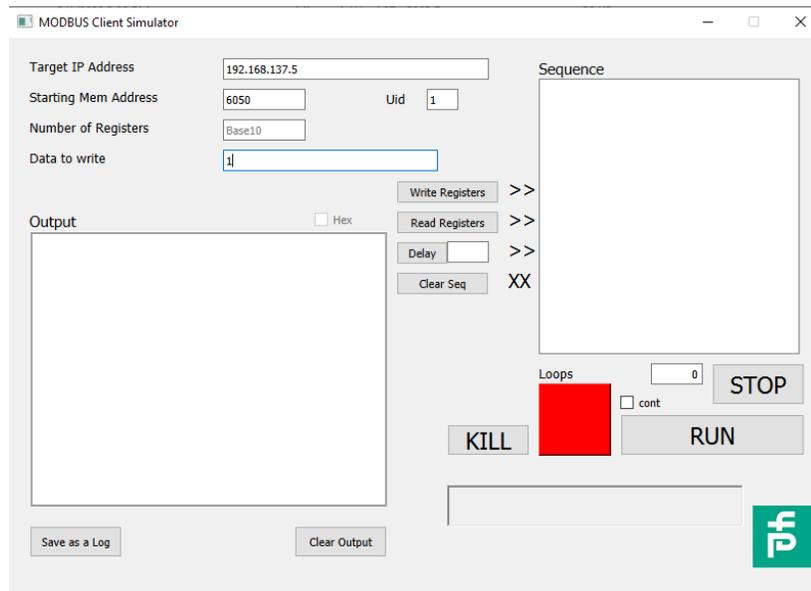
	IO-Link Port 5	IO-Link Port 6	IO-Link Port 7	IO-Link Port 8	Access	Length
Multiple Port PDI Data Block(s)	4999 (Base 0) 5000 (Base 1)	5999 (Base 0) 6000 (Base 1)	6999 (Base 0) 7000 (Base 1)	7999 (Base 0) 8000 (Base 1)	Read-Only	Configurable per port (s)
Port Specific PDI Data Block	5000 (Base 0) 5001 (Base 1)	6000 (Base 0) 6001 (Base 1)	7000 (Base 0) 7001 (Base 1)	8000 (Base 0) 8001 (Base 1)	Read-Only	Configurable per port
Multiple Port PDO Data Block(s)	5049 (Base 0) 5050 (Base 1)	6049 (Base 0) 6050 (Base 1)	7049 (Base 0) 7050 (Base 1)	8049 (Base 0) 8050 (Base 1)	Read/Write	Configurable per port(s)
Port Specific PDO Data Block	5050 (Base 0) 5051 (Base 1)	6050 (Base 0) 6051 (Base 1)	7050 (Base 0) 7051 (Base 1)	8050 (Base 0) 8051 (Base 1)	Read/Write	Configurable per port

The data that is written for the stack light can be more easily imagined if you keep the binary system in mind.

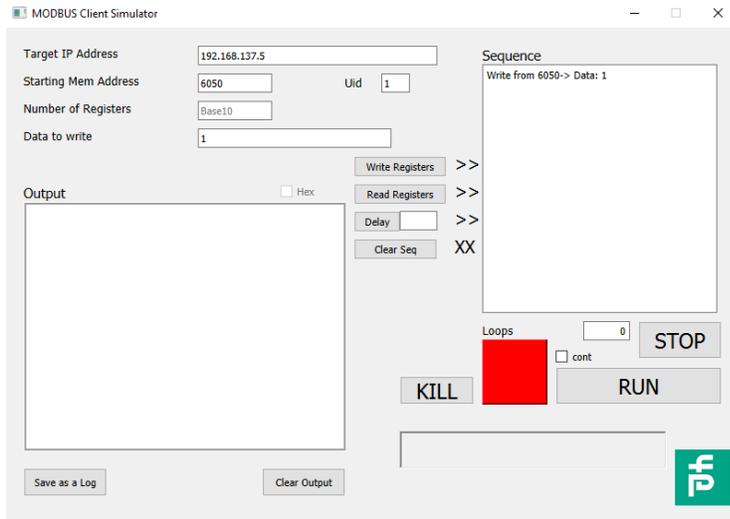
1(base10) = 01(base2) lights up the first stack light

2(base10) = 10(base2) lights up the second stack light

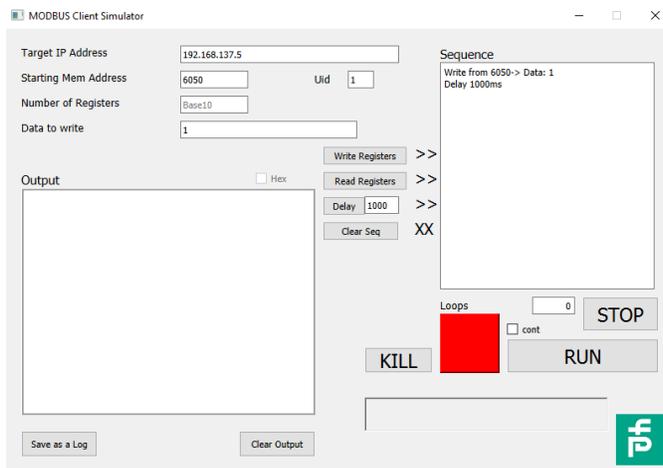
3(base10) = 11(base2) lights up both the first and the second stack light



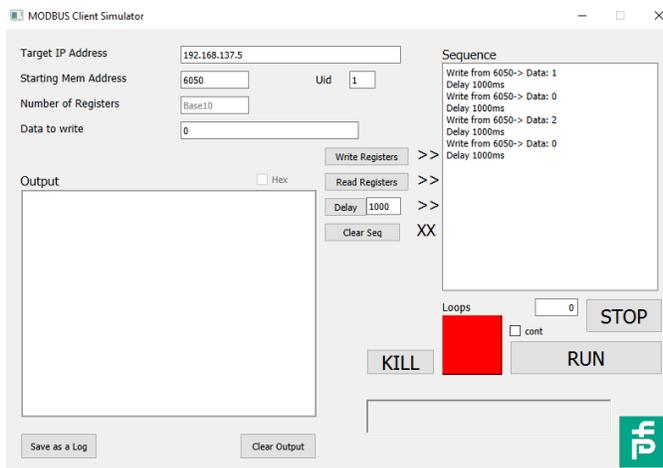
Once decided that you want to write data register 6050 first, click Write Registers to push it into Sequence.



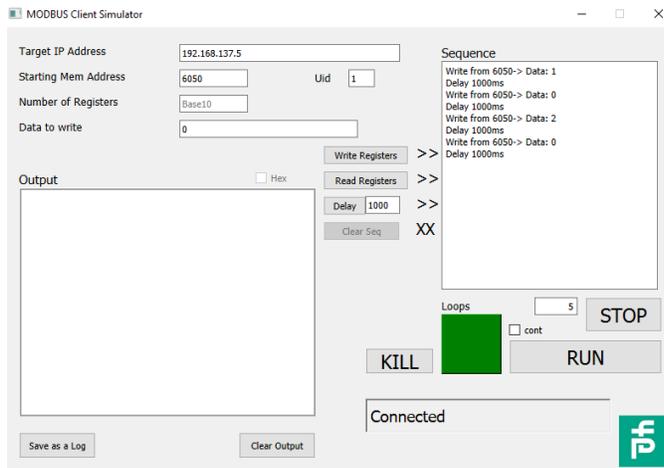
To visualize the flashing effect, a delay between the writes is required. Therefore, add a delay of 1 second by writing 1000 (1000 millisecond = 1second) and press the delay button to push the delay instruction into the Sequence.



Similarly, create more write and delay instructions to make the flashing more interesting.



To run the sequence once, simply click the RUN button. However, to run the sequence multiple times, enter a loop number. In the example 5 was entered, so that the sequence repeats itself five times, and then click RUN.



You should see your stack lights flashing now.

Simulation Example: Reading Sensor Data

Requirements:

- ICE2/3 master
- Any IO-Link sensor connected to the ICE2/3 master
- MODBUS Client Simulator

Setup:

Make sure that your ICE2/3 is powered and that the sensor is connected to one of the available ports.

The ICE2/3 is on the same network as your laptop/PC. To verify this, you should be able to open the web interface of the ICE2/3 block via your web browser.

You need to enable MODBUS TCP/IP in your ICE2/3 web configuration. If this option is disabled, the device cannot communicate via MODBUS.

(A screenshot of the configuration can be seen in the previous example)

Load the IODD file for your sensor and verify in the connected devices that the device is shown in its corresponding port.

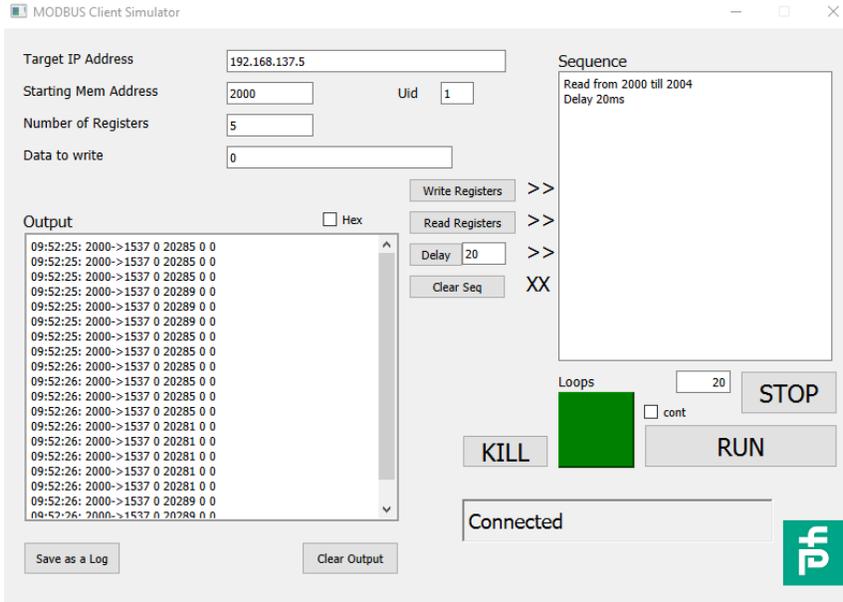
(A screenshot can be seen in the previous example).

Open the MODBUS Client Simulator and you are ready to flash some stack lights.

The Target IP Address will be the IP of your ICE2/3 module.

The Sensor in this demo is connected to port2 and the corresponding PDI (Read) address can be found in the MODBUS Address Definitions in the ICE2/3 manual.

When you enter 5 in the Number of Registers field and click the Read Registers button, the read instructions are inserted into the sequence in the order after which a delay of 20 ms is provided. The simulator will read the registers from 2000 to 2004 (5 in total). In below example, it will do this 20 times. When you click RUN the corresponding data is populated in the output screen in decimal format. In the following example, the third register shows the distance measurement of your ultrasonic sensor.



If the Hex checkbox is checked, the data will be converted to hex format.

