

<https://www.halvorsen.blog>



Modbus with MATLAB

Hans-Petter Halvorsen

Contents

- Modbus
- Modbus Demo Software
 - Modbus Server and Modbus Client used for Demo and Testing. We will use this when testing the MATLAB scripts.
- Modbus with MATLAB
 - Industrial Communication Toolbox
 - Modbus Explorer App
 - MATLAB Examples
 - “Coils” Examples
 - “Discrete Input Registers” Examples
 - “Input Registers” Examples
 - “Holding Registers” Examples



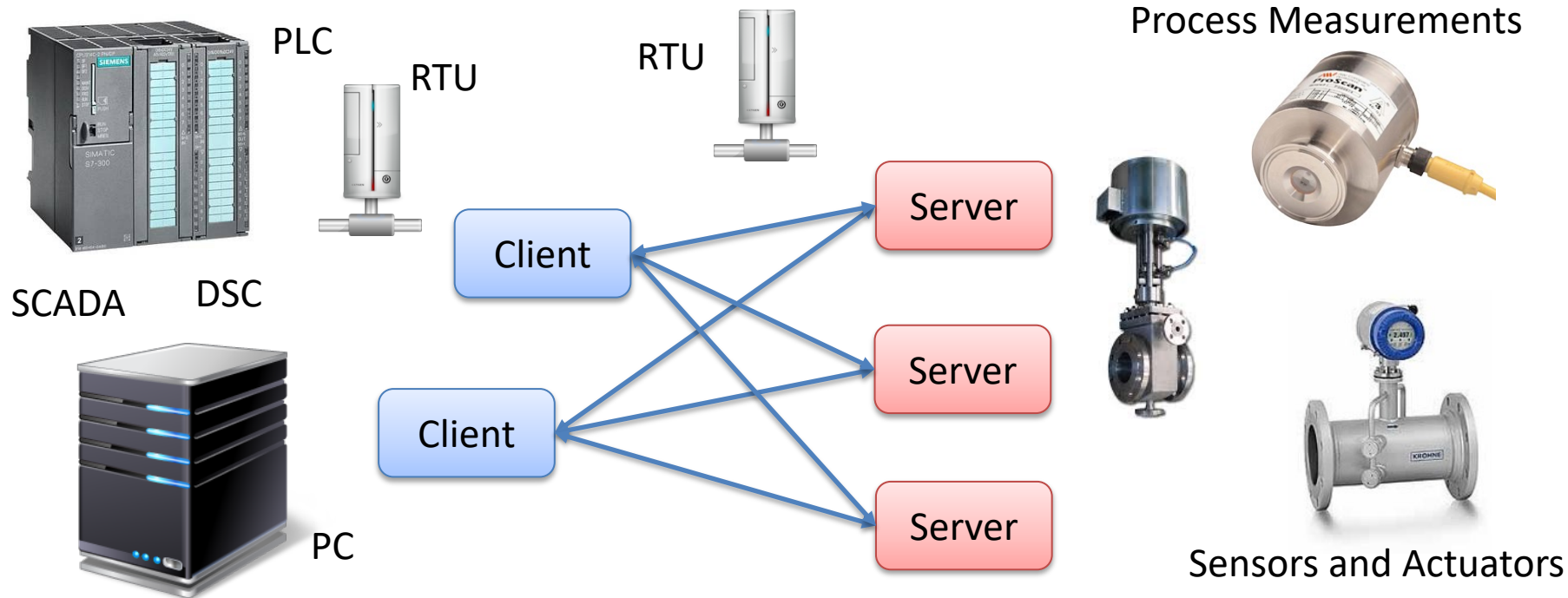
Modbus

What is Modbus?

- **Modbus is a Serial Communications Protocol** originally published by Modicon (now Schneider Electric) in 1979 for use with its programmable logic controllers (PLCs).
- **Simple and robust**, it has since become a de facto standard communication protocol, and it is now a commonly available means of connecting industrial electronic devices
- The development and update of Modbus protocols has been managed by the Modbus Organization since April 2004, when Schneider Electric transferred rights to that organization (<https://modbus.org>)
- Modbus became the first widely accepted fieldbus standard.
- Very popular in the industry and supported by PLCs, SCADA systems, DCS Systems, Process Equipment like Valves, PID controllers, etc.

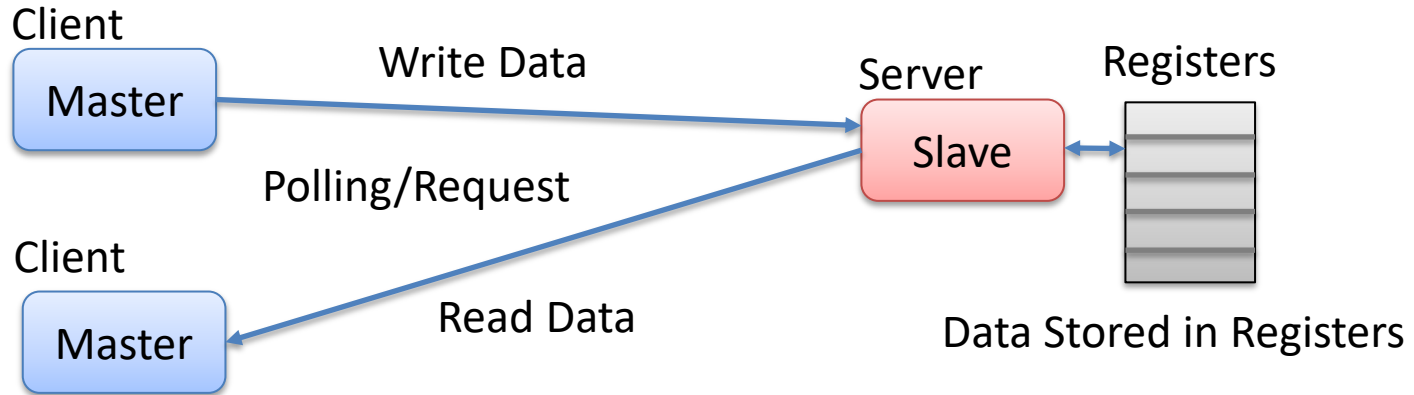
Modbus

The Server typically is a PLC (Programmable Logic Controller), PC or DCS (Distributed Control System)



A remote terminal unit (RTU) is a microprocessor-controlled electronic device that interfaces objects in the physical world to a DCS or SCADA System

Client/Server



Note! The terms “Master” and “Slave” used in Modbus has been replaced with the terms “Client” and “Server”. It takes time to establish these new terms, so in different literature and different software you may still see both the old and the new terms.

Modbus Register Types

- **Coil** (Discrete Output)
 - Coils are 1-bit registers, used to control discrete outputs, Read or Write
- **Discrete Input** (Read Only)
 - 1-bit registers
- **Input Register** (Read Only)
- **Holding Register** (Read/Write)

Access Levels

Register Type	Data Type	Client Access	Server Access
Coils	Bit (Boolean)	Read/Write	Read/Write
Discrete Input	Bit (Boolean)	Read-only	Read/Write
Input Register	Unsigned Word	Read-only	Read/Write
Holding Register	Unsigned Word	Read/Write	Read/Write

Note! An **Unsigned Word** is a 16-bit nonnegative Integer Value between 0 – 65535 (2^{16})
For decimal values, you typically need to use 2 registers to store a decimal number



Modbus Demo Software

Hans-Petter Halvorsen

[Table of Contents](#)

Modbus Demo Software


- The Modbus Demo Software consists of
 - A Modbus Demo Server
 - A Modbus Demo Client
- The Modbus Demo Software is created with LabVIEW
- It supports all 4 Modbus Registers
 - Coils, Discrete Input, Input Register and Holding Register
- Source Code is also available
- In order to use it you need to
 - Download the Software from <https://www.halvorsen.blog/>
 - <https://www.halvorsen.blog/documents/technology/modbus/>

Modbus Demo Software

Modbus Demo Server

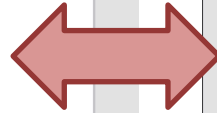
Coils Discrete Input Input Register Holding Register Settings Help

Coils Register (Bit, Boolean, Read/Write Access)



Starting Address: 0 Number of Coils: 10

Start Stop Exit




Modbus Demo Client

Modbus Server Address: localhost Port: 502

Start Stop

Coils Discrete Input Input Register Holding Register Help

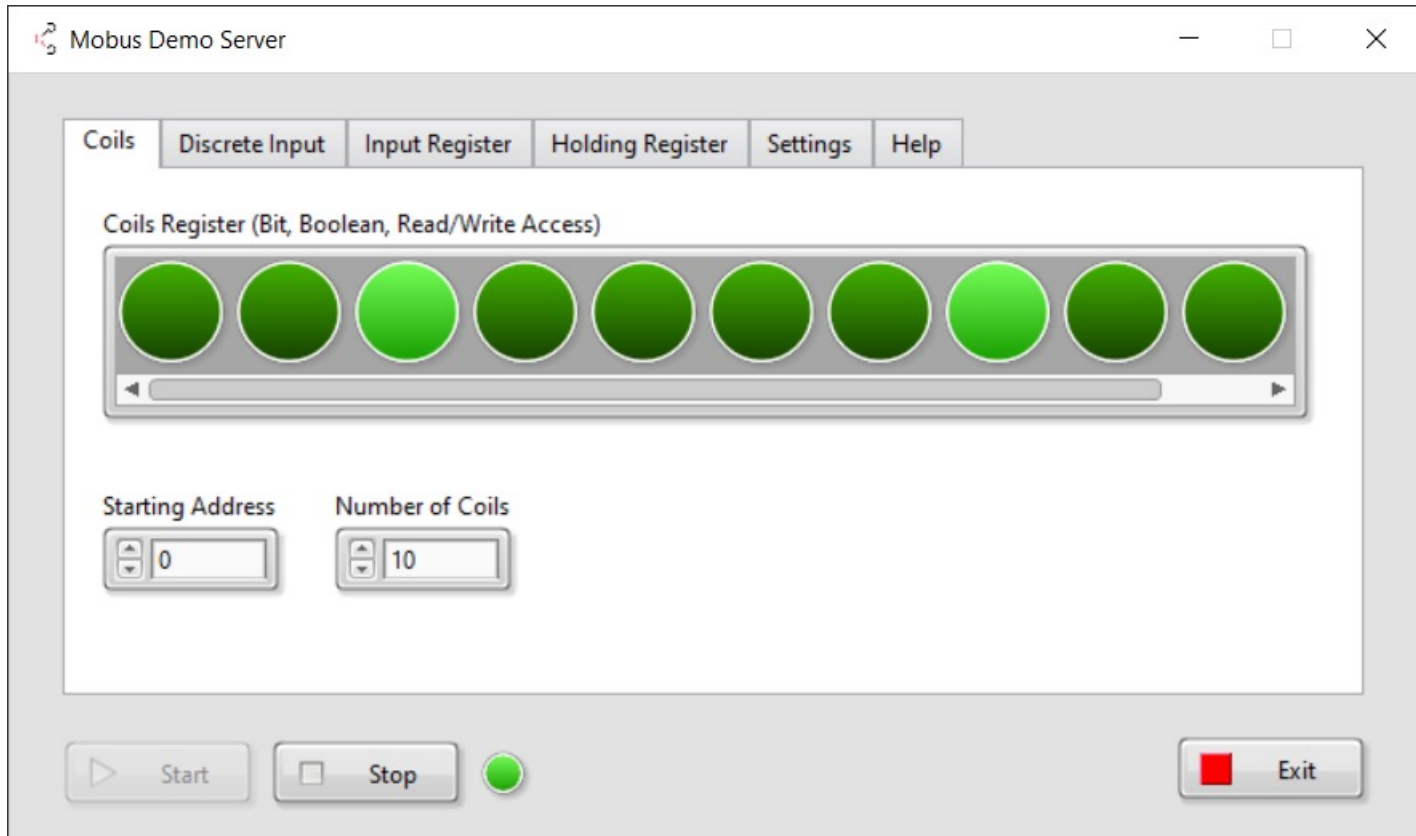
Coils Register (Bit, Boolean, Read/Write Access)



Starting Address: 0 Number of Coils: 10

Start Stop Exit

Modbus Demo Server




Modbus Demo Client

Modbus Demo Client

Modbus Server Address: Port:

Coils Discrete Input Input Register Holding Register Help

Coils Register (Bit, Boolean, Read/Write Access)



Starting Address: Number of Coils:



Modbus with MATLAB

Hans-Petter Halvorsen

[Table of Contents](#)

Modbus with MATLAB

- In order to use Modbus with MATLAB you can use the “**Industrial Communication Toolbox**”.
- The “Industrial Communication Toolbox” supports the following Protocols:
 - OPC, both OPC DA and OPC UA (previously OPC Toolbox)
 - MQTT
 - Modbus (which is the focus in this Tutorial)
- **Note!** “Industrial Communication Toolbox” is a new Toolbox that is included in “MATLAB R2022a” and newer versions

<https://mathworks.com/help/icomm/modbus.html>



Industrial Communication Toolbox

Hans-Petter Halvorsen

[Table of Contents](#)

Industrial Communication Toolbox

- You can use it to communicate with Modbus Servers, such as a PLC, etc.
- It supports Modbus interface over TCP/IP or Serial RTU
- You can use the Modbus Explorer App or ordinary MATLAB programming

MATLAB Functions

Main MATLAB Functions:

- **modbus** – Establish Connection to a Modbus Server
- **read** - Read data from a Modbus Server
- **write** - Write data to a Modbus Server

modbus

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;
```

```
m = modbus(modbustype, modbusserver, modbusport);
```

- Choose between the Modbus TCP/IP Protocol “tcpip” or the Modbus RTU Protocol “serialrtu”
- Then you need to specify the address to the Modbus Server, typically in form of an IP Address, or you can use “localhost” if the server is located on the same computer
- The default Port used by Modbus is “502”.

<https://mathworks.com/help/icomm/ug/create-a-modbus-connection.html>

read

`read(m, target, address, count)`

- Reads data from the Modbus object `m`
- “target” is the type of Modbus Register (“coils”, “inputs”, “inputregs” or “holdingregs”) to read data from
- “address” is the starting address for the Modbus register
- “count” is number of values to read

Basic Read Example

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'coils';  
startaddress = 1;  
count = 10;  
  
data = read(m,registertype, startaddress, count)
```

Mobus Demo Server

Coils Discrete Input Input Register Holding Register Settings Help

Coils Register (Bit, Boolean, Read/Write Access)

Starting Address: 0 Number of Coils: 10

Start Stop

Mobus Demo Client

Modbus Server Address: localhost Port: 502

Start Stop

Coils Discrete Input Input Register Holding Register Help

Coils Register (Bit, Boolean, Read/Write Access)

Starting Address: 0 Number of Coils: 10

Exit

MATLAB R2022a - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Compare Print Go To Find Refactor Profiler Analyze Run Section Run and Advance Run Step Stop

FILE NAVIGATE CODE ANALYZE SECTION RUN

C:\Users\hansp\OneDrive\Dokumente\MATLAB

Current Folder

Editor - C:\Users\hansp\OneDrive\Documents\Industrial IT and Automation\Mobus\Mobus Examples\MATL...

```

modbus_client_read_coil.m
1 clear, clc
2
3 modbustype = 'tcpip';
4 modbusserver = 'localhost';
5 modbusport = 502;
6
7 m = modbus(modbustype, modbusserver, modbusport);
8
9
10 registertype = 'coils';
11 startaddress = 1;
12 count = 10;
13
14 data = read(m,registertype, startaddress, count)

```

Workspace

Name	Value
count	10
data	[0,0,1,0,0,0,0,1,0,0]
m	1x1 Modbus
modbusport	502
modbusserver	'localhost'
modbustype	'tcpip'
registertype	'coils'
startaddress	1

Command Window

```
data =
0 0 1 0 0 0 0 1 0 0
```

Zoom: 100% UTF-8 CRLF script Ln 3 Col 22

write

`write(m, target, address, values)`

- Writes data to the Modbus object m
- “target” is the type of Modbus Register (“coils” or “holdingregs”)
- “address” is the starting address for the Modbus register
- “values” is the actual coil values, which is binary values 1 (True) or 0 (False)
- You can write to coils or holding registers.


Basic Write Example

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'coils';  
startaddress = 1;  
values = [1 1 0 1];  
  
write(m, registertype, startaddress, values)
```


Modbus Demo Server

Coils Discrete Input Input Register Holding Register Settings Help

Coils Register (Bit, Boolean, Read/Write Access)



Starting Address: 0 Number of Coils: 10


Modbus Demo Client

Modbus Server Address: localhost Port: 502

Start Stop

Coils Discrete Input Input Register Holding Register Help

Coils Register (Bit, Boolean, Read/Write Access)



Starting Address: 0 Number of Coils: 10

Exit

MATLAB R2022a - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

File Edit View Tools Window Help

Current Folder: C:\Users\hansp\OneDrive\Documents\Industrial IT and Automation\Modbus\Modbus Examples\MATLAB

Editor - C:\Users\hansp\OneDrive\Documents\Industrial IT and Automation\Modbus\Modbus Examples\MATLAB\modbus_client_read_coil.m

```

1 clear, clc;
2
3 modbustype = 'tcpip';
4 modbusserver = 'localhost';
5 modbusport = 502;
6
7 m = modbus(modbustype, modbusserver, modbusport);
8
9 registertype = 'coils';
10 startaddress = 1;
11 values = [1 1 0 1];
12
13 write(m, registertype, startaddress, values)

```

Command Window

fx >>

Select a file to view details

Workspace

Name	Value
m	1x1 Modbus
modbusport	502
modbusserver	'localhost'
modbustype	'tcpip'
registertype	'coils'
startaddress	1
values	[1,1,0,1]

Zoom: 100% UTF-8 CRLF script Ln 9 Col 24

<https://www.halvorsen.blog>



Modbus Explorer App

Hans-Petter Halvorsen

[Table of Contents](#)

Modbus Explorer App

The screenshot displays the Modbus Explorer application interface. The main window is titled "Modbus Explorer" and features a "Getting Started" section with two buttons: "Configure Modbus Serial" and "Configure Modbus TCP/IP". Below these, a "Device List" section shows "All Hardware" with a refresh icon and a message "No hardware detected". A "Modbus TCP/IP" configuration dialog is open, showing "Device Address: '1...", "Port: 502", and "Server ID: 1".

The "MODBUS EXPLORER" toolbar includes buttons for Import, Export, Insert, Delete, Move Up, Move Down, Sort, Generate Script, Help, Close Session, and Close. Below the toolbar, the "Read Registers" section is active, displaying a table with the following data:

Se...	Name	Address	Register Type	Precision	Read Value
<input type="checkbox"/>	Reg_1	1	Coil	bit	1

The "Write Register" section on the right allows for writing data to a single register, with fields for Address (1), Register Type (Coil), Precision (bit), and Write Value (1), and a "Write" button.

The "Plot Tools" section at the bottom right includes a "Show Legend" checkbox, a table for selecting registers to plot, and a "Duration" field set to 10. The plot area shows a graph of "Read Value" vs "Time (sec)" for "Reg_1", with a value of 1 from 156 to 157 seconds and 159 to 160 seconds, and 0 in between.


<https://mathworks.com/videos/modbus-explorer-app-overview-1549974072928.html>

Modbus Explorer App

Modbus Demo Server

Coils Discrete Input Input Register Holding Register Settings Help

Coils Register (Bit, Boolean, Read/Write Access)



Starting Address: 0 Number of Coils: 10

Start Stop Exit

MODBUS EXPLORER

Import Export Insert Delete Move Up Move Down Sort Generate Script CODE Help Close Session CLOSE

FILE TABLE ROWS RESOURCES

Device List

All Hardware

Modbus TCP/IP
Device Address: 1...
Port: 502
Server ID: 1

Read Registers Enter register data in the table to read live data. **LIVE** Resume Reads

Se...	Name	Address	Register Type	Precision	Read Value
<input type="checkbox"/>	Reg_1	1	Coil	bit	1

Write Register Write data to a single re...

Address: 1

Register Type: Coil

Precision: bit

Write Value: 1

Write

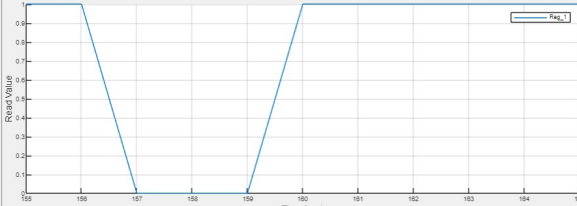
Plot Tools

Show Legend

Se...	Name	Y-Axis
<input checked="" type="checkbox"/>	Reg_1	Autosc... <input checked="" type="checkbox"/>

Max: 100
Min: 0

X-Axis: Duration 10



Read Value

Time (sec)



MATLAB Examples

Modbus MATLAB Examples

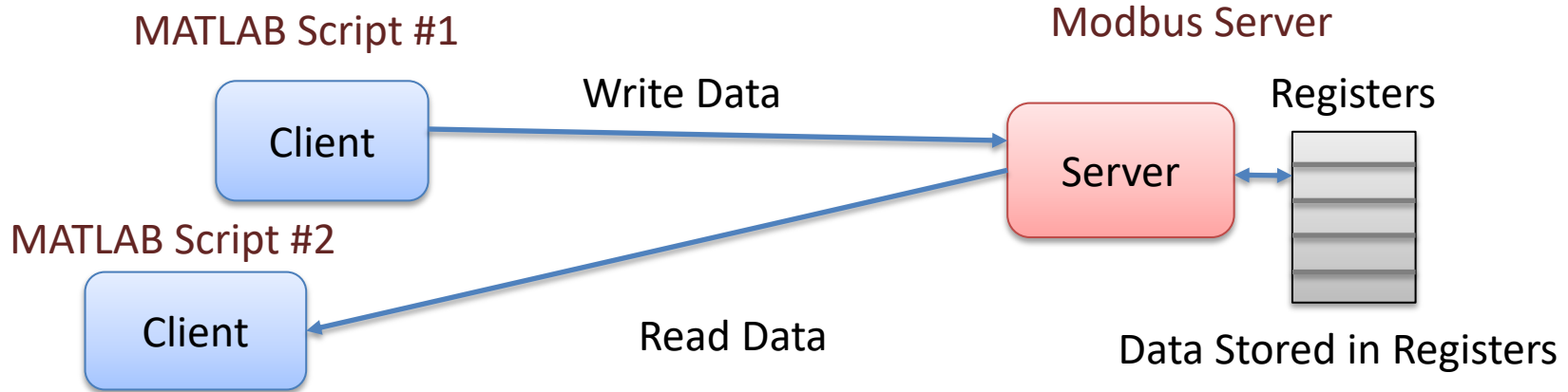
- Coils Examples
- Discrete Input Registers Examples
- Input Registers Examples
- Holding Registers Examples

Register Type	Data Type	Client Access	Server Access
Coils	Bit (Boolean)	Read/Write	Read/Write
Discrete Input	Bit (Boolean)	Read-only	Read/Write
Input Register	Unsigned Word	Read-only	Read/Write
Holding Register	Unsigned Word	Read/Write	Read/Write



Coils Examples

Coils Examples



Memory Type	Data Type	Server Access	Client Access
Coils	Bit (Boolean)	Read/Write	Read/Write

Write Coils

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'coils';  
startaddress = 1;  
values = [1 1 0 1];  
  
write(m, registertype, startaddress, values)
```

Read Coils

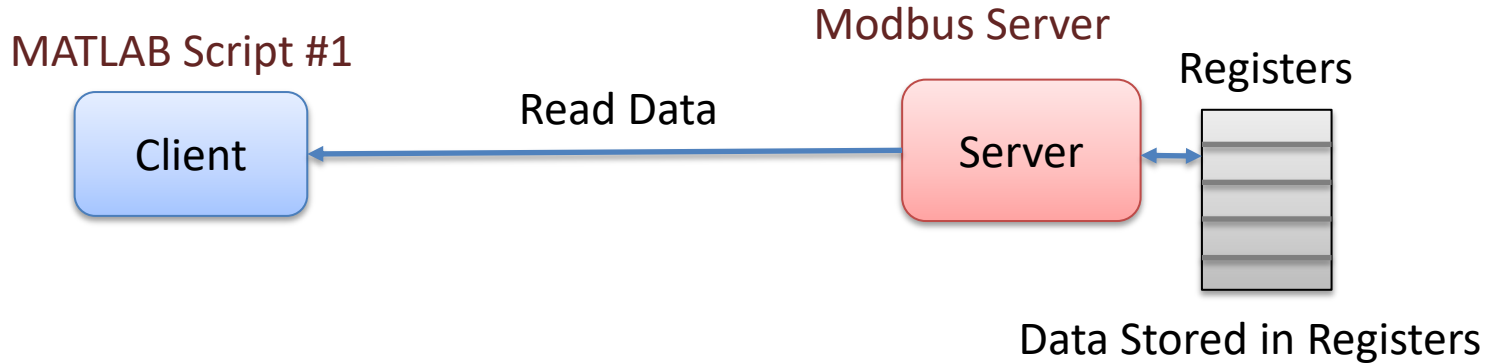
```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'coils';  
startaddress = 1;  
count = 10;  
  
data = read(m,registertype, startaddress, count)
```



Discrete Input Registers

Examples

Discrete Input Examples



Memory Type	Data Type	Client Access	Server Access
Discrete Input	Bit (Boolean)	Read-only	Read/Write

Read Discrete Input

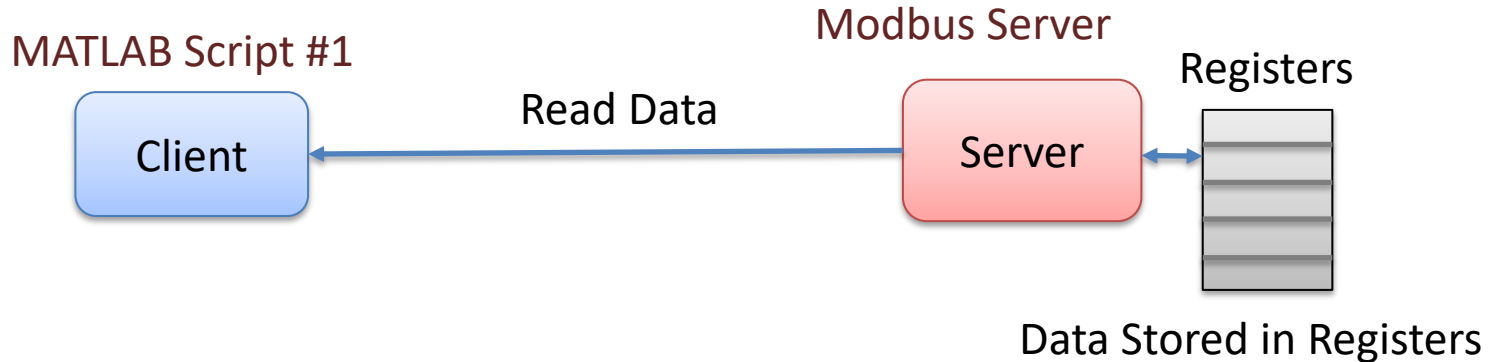
```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'inputs';  
startaddress = 1;  
count = 10;  
  
data = read(m,registertype, startaddress, count)
```



Input Registers

Examples

Input Registers Examples



Memory Type	Data Type	Client Access	Server Access
Input Register	Unsigned Word	Read-only	Read/Write

Read Input Registers

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'inputregs';  
startaddress = 1;  
count = 10;  
  
data = read(m,registertype, startaddress, count)
```

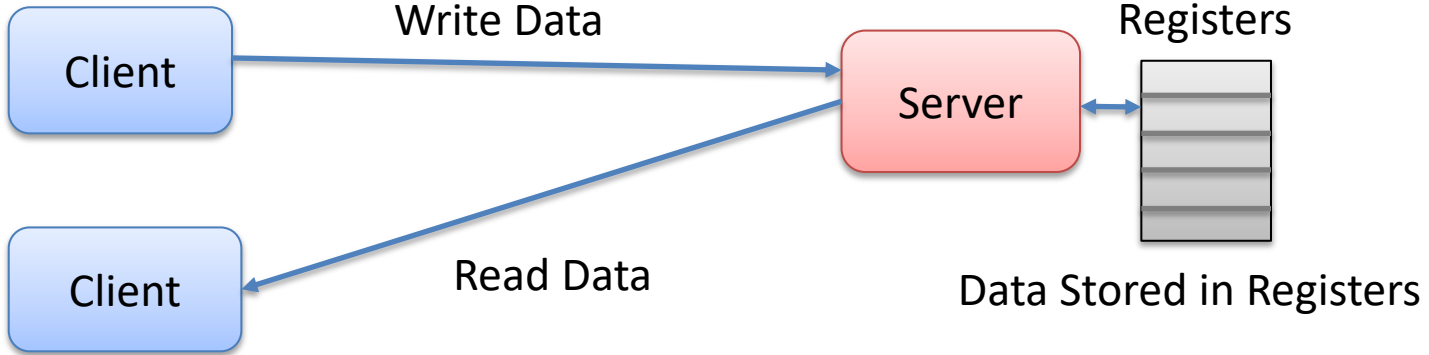



Holding Registers

Examples

Holding Registers Examples

MATLAB Script #1



MATLAB Script #2

Memory Type	Data Type	Client Access	Server Access
Holding Register	Unsigned Word	Read/Write	Read/Write

Write Holding Registers

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'holdingregs';  
startaddress = 1;  
values = [189 67 450 121];  
  
write(m, registertype, startaddress, values)
```

Read Holding Registers

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'holdingregs';  
startaddress = 1;  
count = 10;  
  
data = read(m,registertype, startaddress, count)
```



Decimal Values

Decimal Values

- In Modbus, the Input Registers and the Holding Registers are Unsigned 16bit Word values
- An Unsigned Word is a 16-bit nonnegative Integer Value between 0 – 65535 (2^{16})
- In Modbus, the default practice is to split a 32-bit floating point value across two 16-bit registers.
- We then need to use 2 Modbus register for representing one number

Data Type “Word” – 16bit Unsigned Integer

Data Type “Single” – 32bit Single Precision (Decimal Values)

Write Decimal Values to Holding Registers

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'holdingregs';  
startaddress = 1;  
values = [19.2, 21.7, 22.4, 23.1];  
  
write(m, registertype, startaddress, values, 'single')
```

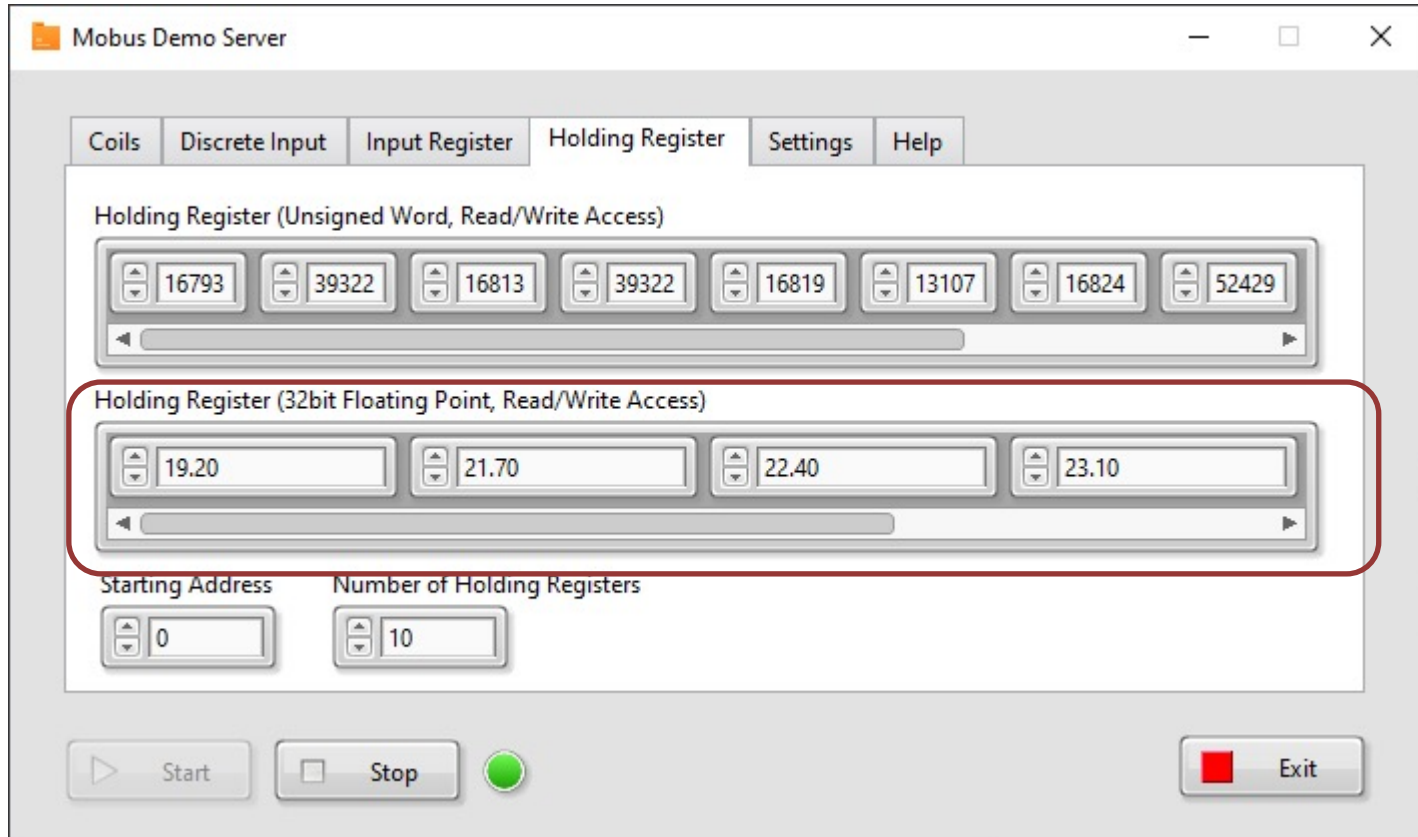
Read Decimal Values from Holding Registers

```
modbustype = 'tcpip';  
modbusserver = 'localhost';  
modbusport = 502;  
  
m = modbus(modbustype, modbusserver, modbusport);  
  
registertype = 'holdingregs';  
startaddress = 1;  
count = 4;  
  
data = read(m,registertype, startaddress, count, 'single')
```

```
data =
```

```
19.2000    21.7000    22.4000    23.1000
```


Modbus Demo Server



References

- Modbus Organization: <http://www.modbus.org>
- Modbus (Wikipedia): <https://en.wikipedia.org/wiki/Modbus>
- Introduction to Modbus (National Instruments):
<http://www.ni.com/white-paper/7675/en/>
- Connect LabVIEW to Any PLC With Modbus (National Instruments):
<http://www.ni.com/tutorial/13911/en/>
- Modbus 101 - Introduction to Modbus:
http://www.csimn.com/CSI_pages/Modbus101.html
- Modbus TCP/IP: <http://www.rtaautomation.com/technologies/modbus-tcpip/>
- Modbus RTU: <http://www.rtaautomation.com/technologies/modbus-rtu/>
- Using Modbus for Process Control and Automation (PDF):
http://www.miinet.com/Portals/0/articles/Using_MODBUS_for_Process_Control_and_Automation.pdf

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

