1. Modbus Slave/Master Device

Revision History			
Revision	Date	Author	Comments
01A	Nov 26, 2024	G Wright	Initial release

1.1 Overview

The Modbus Master and Modbus Slave Devices are configurable input/output devices that use the Modbus communications protocol. This protocol supports reading and writing of coils and reading and writing of registers. Coils are binary on/off settings, while holding registers can hold various data types.

The Modbus master initiates communication with one or more other Modbus slaves, specifically it initiates a request for data, and the slaves respond with the requested data. The master then checks the response from each slave, and if it is valid, the data is accepted.

The NavView devices support the reading and writing of data to and from registers.

1.2 Terminology

When using Modbus devices, it is important to understand the terminology.

Slave Id: The Slave Id field is used to identify the Modbus Slave device. There can be up to 247 slave devices connected, and the slave Id is used by the master to differentiate them. Valid values are between 1 and 247

Bit: Smallest unit of digital data contains one of only two values, 0 and 1.

Byte: A byte consists of 8 bits.

Word: A word consists of 2 or more bytes.

Endian: Refers to the order of bits in a byte and bytes in a word.

Big Endian: Decreasing numeric significance with increasing memory addresses. In the case of a word, the most significant byte of the word is at the smallest memory address and the least significant byte at the largest. Similarly, in the case of a byte, the most significant bit is at the smallest memory address and the least significant bit at the largest.

Little Endian: Increasing numeric significance with increasing memory addresses. In the case of a word, the most significant byte of the word is at the largest memory address and the least significant byte at the smallest. Similarly, in the case of a byte, the most significant bit is at the largest memory address and the least significant bit at the smallest.

Data Addresses: Modbus uses data addresses from 0000 to 270E or 0 to 9998 with an offset based on the register use.

Register: The location a value is within the respective data addresses. Each register is 1 word = 2 bytes – 16 bits. The data field is made up of two hexadecimal digits, in the range of 00 to FF. These can be made from a pair of ASCII characters (8 bits each) or one RTU character.

Holding Register: A holding register can be used for read and write operations, i.e., a master and slave can exchange data using holding registers. These have an address offset of 40001 and therefore a range of 40001 for register 0 to 49999 for register 270E.

Input Register: An input register can be used for read operations only, i.e., only a master can read data from a slave. These have an address offset of 30001 and therefore a range of 30001 for register 0 to 39999 for register 270E.

Value Type: The value type controls binary format of the data and therefore how many registers it will occupy and what kind of data can be passed.

Int16: 16 bits = 1 register. A signed integer between -32,768 and 32,767

Int32: 32 bits = 2 registers. A signed integer between -2,147,483,648 and 2,147,483,647

Int64: 64 bits = 4 registers. A signed integer between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807

Uint16: 16 bits = 1 register. An unsigned integer between 0 and 65,535

Uint32: 32 bits = 2 registers. An unsigned integer between 0 and 4,294,967,295

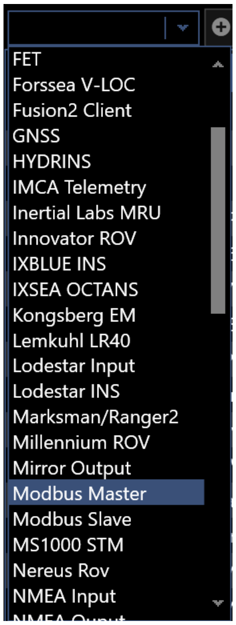
Uint64: 64 bits = 4 registers. An unsigned integer between 0 and 18,446,744,073,709,551,615

Float: 32 bits – 2 registers. A decimal value between 1.175494351 E-38 and 3.402823466 E+38

Double: 64 bits = 4 registers. A decimal value between 2.2250738585072014 E-308 and 1.7976931348623158 E+308

1.3 Add Modbus Device

1. From the Explorer view or Setup ribbon, select Devices to display the IO Devices
2. From the dropdown list, select Modbus Master or Modbus Slave and click the **+** button



FET
Forssea V-LOC
Fusion2 Client
GNSS
HYDRINS
IMCA Telemetry
Inertial Labs MRU
Innovator ROV
IXBLUE INS
IXSEA OCTANS
Kongsberg EM
Lemkuhl LR40
Lodestar Input
Lodestar INS
Marksman/Ranger2
Millennium ROV
Mirror Output
Modbus Master
Modbus Slave
MS1000 STM
Nereus Rov
NMEA Input

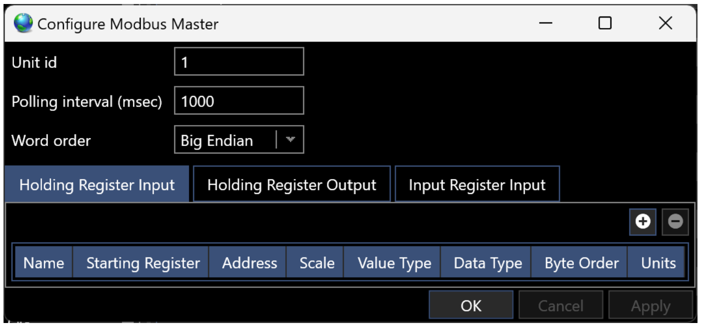
Figure 1 Add Modbus Device

3. Configure the Device IO parameters and apply them accordingly (refer to the Devices section of the User Guide)

Note: A Modbus Master device defaults to TCP Client to match the master role of initiating data requests. Similarly, a Modbus Slave defaults to TCP server to match its role of responding to queries.

1.4 Configure Modbus Master Device

1. Access the Configure Modbus view from the IO Device view by either right mouse clicking on the Modbus device in the list and selecting Device Settings or selecting it in the list and clicking the Configure device icon (in the Device view tool bar.



Configure Modbus Master

Unit id: 1
Polling interval (msec): 1000
Word order: Big Endian

Name	Starting Register	Address	Scale	Value Type	Data Type	Byte Order	Units

Figure 2 Modbus Master Configuration - Holding Register Input Tab

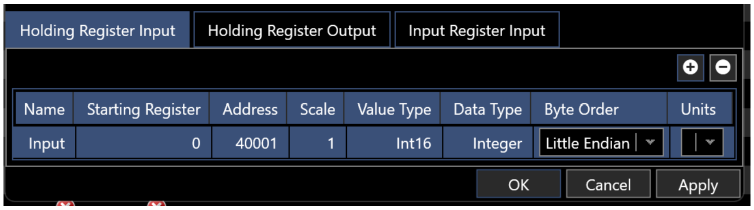
2. Configure general device parameters

   Unit Id: Enter the ID of the Modbus Slave that is to be communicated with

   Polling interval (msec): Enter the interval between the master's queries of the slaves

Word order: From the dropdown list, select if the word order is Big Endian or Little Endian.

3. Configure the Registers
   a. Holding Register Input
      i. Click on the Holding Register Input tab
      ii. Add an input item by clicking the ⊕ button



Holding Register Input | Holding Register Output | Input Register Input

Name	Starting Register	Address	Scale	Value Type	Data Type	Byte Order	Units
Input	0	40001	1	Int16	Integer	Little Endian	

Figure 3 Modbus Master Holding Register Input Tab

      iii. Remove an item by selecting it and clicking the ⊖ button.
      iv. Configure the item

Name: Enter the name the associated data will be listed as.

Starting Register: Enter the starting register for the data, the default is the first available starting with 0 with subsequent values calculated from the preceding data Value Types

Note: It is important that the Holding Registers used for Output and Input do not conflict given they use the same address offset and range.

Address: This displays the address of the data calculated from the offset address of 40001 and the Starting Register, it is not configurable

Scale: Enter the scalar that was applied to the value when it was placed in this register, e.g., if a value was multiplied by 10 before being put in this register by the slave, enter 10 here and the master will divide the read value by 10 to get the original value

Value Type: From the dropdown list, select the value type to use

Data Type: From the dropdown list, select the NavView data type the input value represents

Byte Order: From the dropdown list, select Big Endian or Little Endian

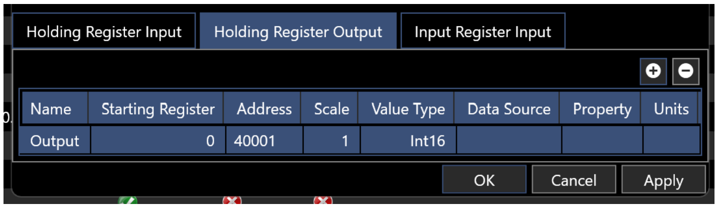
Units: From the dropdown list, populated based on the Data Type selected, select the units of the input value.

      v. Click Apply to apply the settings without closing the dialog, or OK to apply the settings and close the dialog

b. Holding Register Output

   i. Click on the Holding Register Output tab

   ii. Add an item by clicking the ⊕ button



Name	Starting Register	Address	Scale	Value Type	Data Source	Property	Units
Output	0	40001	1	Int16			

Figure 4 Modbus Master Holding Register Output Tab

   iii. Remove an item by selecting it and clicking the ⊖ button.

   iv. Configure the item

Name: Enter the name the associated data will be listed as when published.

Starting Register: Enter the starting register for the data, the default is the first available starting with 0 with subsequent values calculated from the preceding data Value Types

Note: It is important that the Holding Registers used for Output and Input do not conflict given they use the same address offset and range.

Address: This displays the address of the data calculated from the offset address of 40001 and the Starting Register, it is not configurable

Scale: Enter the scalar to apply to the value before placing it in the register

Value Type: From the dropdown list, select the value type to use

Data Source: From the dropdown list, select the NavView data source of the output value

Property: From the dropdown list, select the Property of the selected Data Source to output

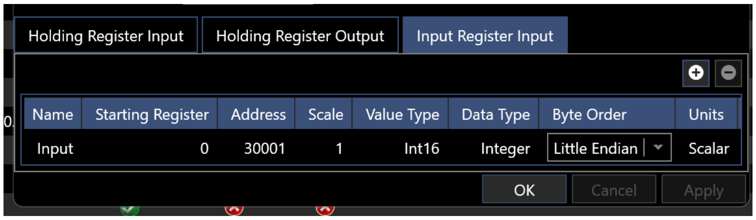
Units: From the dropdown list, populated based on the Data Type selected, select the units to output the value in

   v. Click Apply to apply the settings without closing the dialog, or OK to apply the settings and close the dialog

c. Input Register Input

   i. Click on the Input Register Input tab

   ii. Add an item by clicking the ⊕ button



Name	Starting Register	Address	Scale	Value Type	Data Type	Byte Order	Units
Input	0	30001	1	Int16	Integer	Little Endian	Scalar

Figure 5 Modbus Master Input Register Input Tab

iii. Remove an item by selecting it and clicking the ⊖ button.

iv. Configure the item

Name: Enter the name the associated data will be listed as.

Starting Register: Enter the starting register for the data, the default is the first available starting with 0 with subsequent values calculated from the preceding data Value Types

Address: This displays the address of the data calculated from the offset address of 30001 and the Starting Register, it is not configurable

Scale: Enter the scalar that was applied to the value when it was placed in this register, e.g., if a value was multiplied by 10 before being put in this register by the slave, enter 10 here and the master will divide the read value by 10 to get the original value

Value Type: From the dropdown list, select the value type to use

Data Type: From the dropdown list, select the NavView data type the input value represents

Byte Order: From the dropdown list, select Big Endian or Little Endian

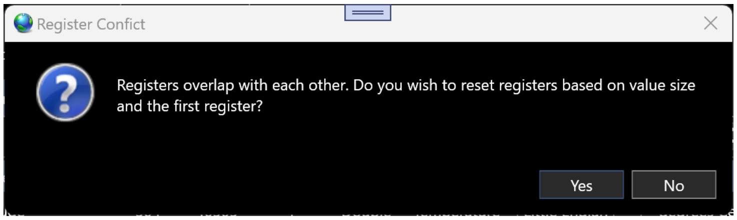
Units: From the dropdown list, populated based on the Data Type selected, select the units of the input value.

v. Click Apply to apply the settings without closing the dialog, or OK to apply the settings and close the dialog

4. Close the configuration dialog

Note: Upon applying the settings, if NavView detects that one or more registers in one of the tabs overlap based on their Starting Register and Value Type the user is alerted with the message in Figure 4. If yes is selected NavView will use the first item's Starting Register and the Value Types to calculate the starting Register for all items. If No is selected, it is recommended that the user review the settings to address any conflicts to prevent issues reading or writing to the respective registers.

Note: This check is applied to the Holding Register Input and Holding Register Output independently. NavView does NOT check if there is a conflict between these 2 configurations, it is the user's responsibility to check this.



Register Confict

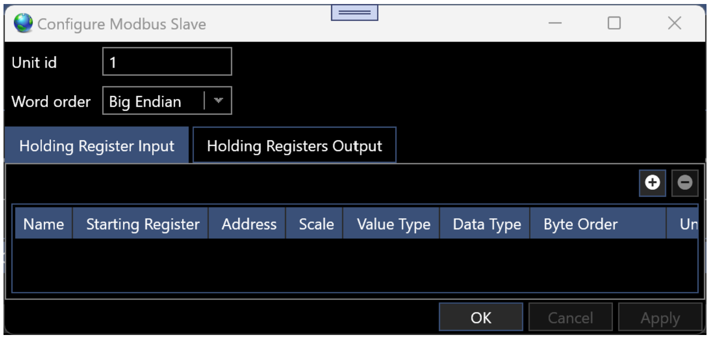
Registers overlap with each other. Do you wish to reset registers based on value size and the first register?

Figure 6 Modbus Register Conflict Message

Note: When the settings are applied, NavView automatically sorts the items based on Starting Register.

1.5 Configure Modbus Slave Device

1. Access the Configure Modbus view from the IO Device view by either right mouse clicking on the Modbus device in the list and selecting Device Settings or selecting it in the list and clicking the Configure device icon (in the Device view tool bar.



Configure Modbus Slave

Unit id: 1

Word order: Big Endian

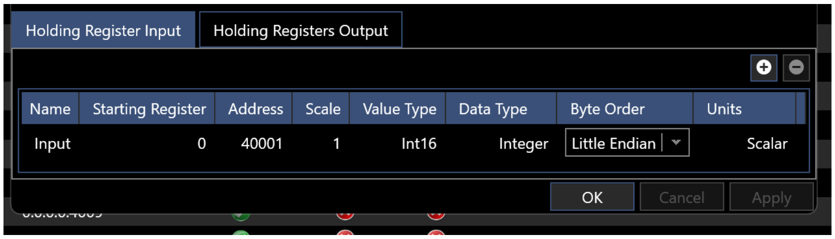
Figure 7 Modbus Slave Configuration - Holding Register Input Tab

2. Configure general device parameters

   Unit Id: Enter the ID of this Modbus Slave

   Word order: From the dropdown list, select if the word order is Big Endian or Little Endian.

3. Configure the Registers
   a. Holding Register Input
      i. Click on the Holding Register Input tab
      ii. Add an input item by clicking the ⊕ button



Name	Starting Register	Address	Scale	Value Type	Data Type	Byte Order	Units
Input	0	40001	1	Int16	Integer	Little Endian	Scalar

Figure 8 Modbus Slave Holding Register Input Tab

iii. Remove an item by selecting it and clicking the ⊖ button.

iv. Configure the item

Name: Enter the name the associated data will be listed as.

Starting Register: Enter the starting register for the data, the default is the first available starting with 0 with subsequent values calculated from the preceding data Value Types

Note: It is important that the Holding Registers used for Output and Input do not conflict given they use the same address offset and range.

Address: This displays the address of the data calculated from the offset address of 40001 and the Starting Register, it is not configurable

Scale: Enter the scalar that was applied to the value when it was placed in this register, e.g., if a value was multiplied by 10 before being put in this register by the slave, enter 10 here and the master will divide the read value by 10 to get the original value

Value Type: From the dropdown list, select the value type to use

Data Type: From the dropdown list, select the NavView data type the input value represents

Byte Order: From the dropdown list, select Big Endian or Little Endian

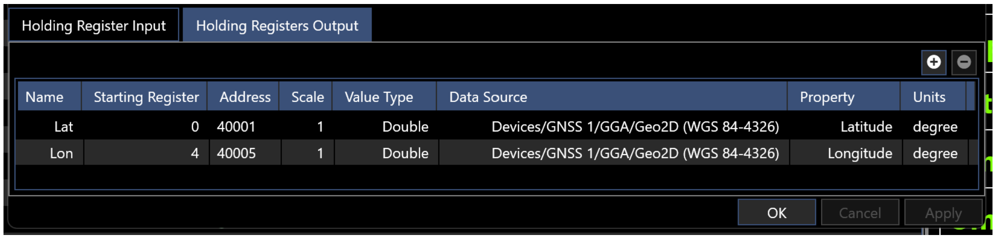
Units: From the dropdown list, populated based on the Data Type selected, select the units of the input value.

v. Click Apply to apply the settings without closing the dialog, or OK to apply the settings and close the dialog

b. Holding Register Output

i. Click on the Holding Register Output tab

ii. Add an item by clicking the ⊕ button



Name	Starting Register	Address	Scale	Value Type	Data Source	Property	Units
Lat	0	40001	1	Double	Devices/GNSS 1/GGA/Geo2D (WGS 84-4326)	Latitude	degree
Lon	4	40005	1	Double	Devices/GNSS 1/GGA/Geo2D (WGS 84-4326)	Longitude	degree

Figure 9 Modbus Slave Holding Register Output Tab

iii. Remove an item by selecting it and clicking the ⊖ button.

iv. Configure the item

Name: Enter the name the associated data will be listed as when published.

Starting Register: Enter the starting register for the data, the default is the first available starting with 0 with subsequent values calculated from the preceding data Value Types

Note: It is important that the Holding Registers used for Output and Input do not conflict given they use the same address offset and range.

Address: This displays the address of the data calculated from the offset address of 40001 and the Starting Register, it is not configurable

Scale: Enter the scalar to apply to the value before placing it in the register

Value Type: From the dropdown list, select the value type to use

Data Source: From the dropdown list, select the NavView data source of the output value

Property: From the dropdown list, select the Property of the selected Data Source to output

Units: From the dropdown list, populated based on the Data Type selected, select the units to output the value in

v. Click Apply to apply the settings without closing the dialog, or OK to apply the settings and close the dialog

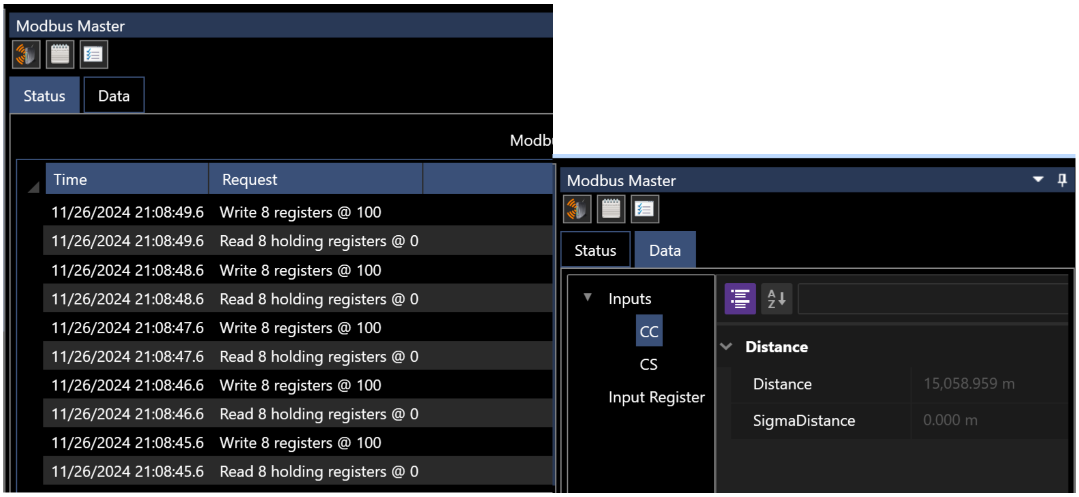
4. Close the configuration dialog

1.6 Monitoring the Modbus Device

The operation of the Modbus device can be monitored using the IO Device view and the Device Status view. The following addresses the Device Status view.

1. Modbus Master (see Figure 10)
    a. Open a Device Status view (see the Devices section in the User Guide)
    b. Status Tab: Modbus History
        i. Timestamp and registers read when slave was queried and it responded
        ii. Timestamp and registers written to for output to slave
    c. Data Tab
        i. Tree of input items
        ii. Expand and select an item to display the published data for it
2. Modbus Slave (see Figure 11)
    a. Status tab: Modbus request History
        i. Timestamp of request and registers written to by the master
        ii. Timestamp and registers read by the master

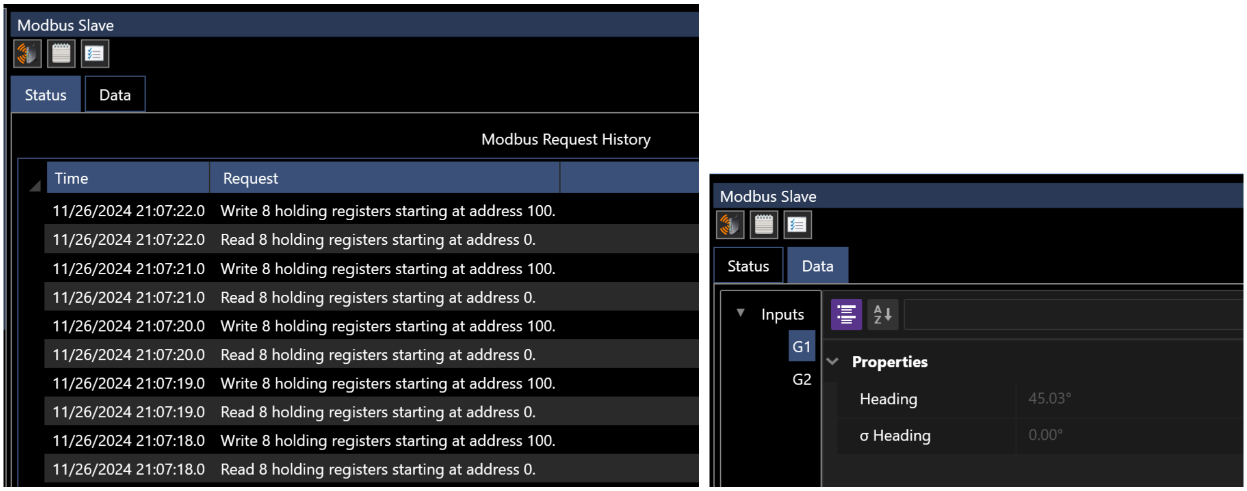
b. Data Tab
   i. Tree of input items
   ii. Expand and select an item to display the published data for it



Time	Request
11/26/2024 21:08:49.6	Write 8 registers @ 100
11/26/2024 21:08:49.6	Read 8 holding registers @ 0
11/26/2024 21:08:48.6	Write 8 registers @ 100
11/26/2024 21:08:48.6	Read 8 holding registers @ 0
11/26/2024 21:08:47.6	Write 8 registers @ 100
11/26/2024 21:08:47.6	Read 8 holding registers @ 0
11/26/2024 21:08:46.6	Write 8 registers @ 100
11/26/2024 21:08:46.6	Read 8 holding registers @ 0
11/26/2024 21:08:45.6	Write 8 registers @ 100
11/26/2024 21:08:45.6	Read 8 holding registers @ 0

Distance	15,058.959 m
SigmaDistance	0.000 m

Figure 10 Modbus Master Input/Output Monitoring



Time	Request
11/26/2024 21:07:22.0	Write 8 holding registers starting at address 100.
11/26/2024 21:07:22.0	Read 8 holding registers starting at address 0.
11/26/2024 21:07:21.0	Write 8 holding registers starting at address 100.
11/26/2024 21:07:21.0	Read 8 holding registers starting at address 0.
11/26/2024 21:07:20.0	Write 8 holding registers starting at address 100.
11/26/2024 21:07:20.0	Read 8 holding registers starting at address 0.
11/26/2024 21:07:19.0	Write 8 holding registers starting at address 100.
11/26/2024 21:07:19.0	Read 8 holding registers starting at address 0.
11/26/2024 21:07:18.0	Write 8 holding registers starting at address 100.
11/26/2024 21:07:18.0	Read 8 holding registers starting at address 0.

Heading	45.03°
σ Heading	0.00°

Figure 11 Modbus Slave Input/Output Monitoring