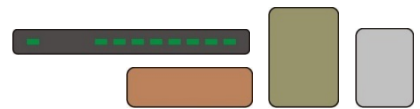




The ModbusTCP Driver



The ModbusTCP driver allows North to interface with a wide range of equipment supporting the Modbus over TCP/IP protocol. The driver can both request values from Modbus devices, and provide values to a Modbus front-end when requested. Available for Commander and ObSys.

This document relates to ModbusTCP driver version 3.0

Please read the *Commander Manual* or *ObSys Manual* alongside this document, available from www.northbt.com

Contents

Compatibility with the Modbus TCP/IP System	3
Equipment	3
Values.....	4
Prerequisites.....	5
Using the Driver	6
Starting the Interface	6
Setting up the Driver.....	6
Checking Communications	6
Understanding Modbus	7
Data Model.....	7
Function Codes	7
Value Decoding.....	8
Object Specifications.....	11
Example Object Reference	11
Device Top-Level Objects	11
ModbusTCP Setup	12
Database Privilege Levels.....	13
Unit Details	14
Formula Setup	15
Modbus System	16
Default Modbus Device.....	17
Appendix A: Modbus Register List.....	22
Essential Data	22
Extra Data	29
Driver Versions	30

Compatibility with the Modbus TCP/IP System

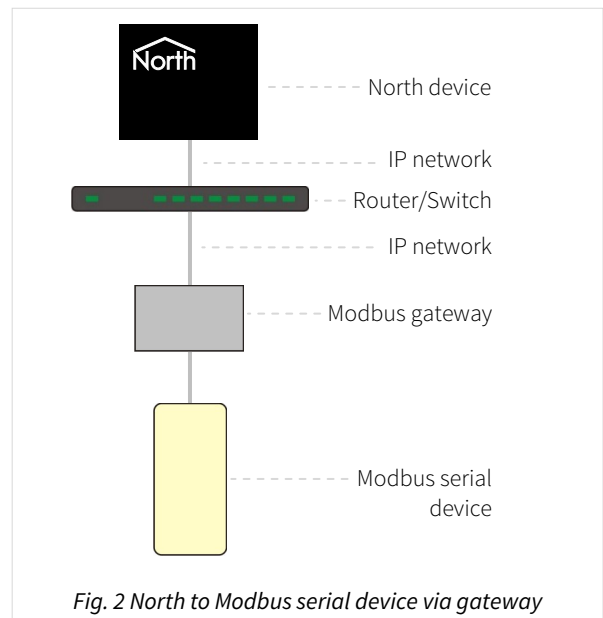
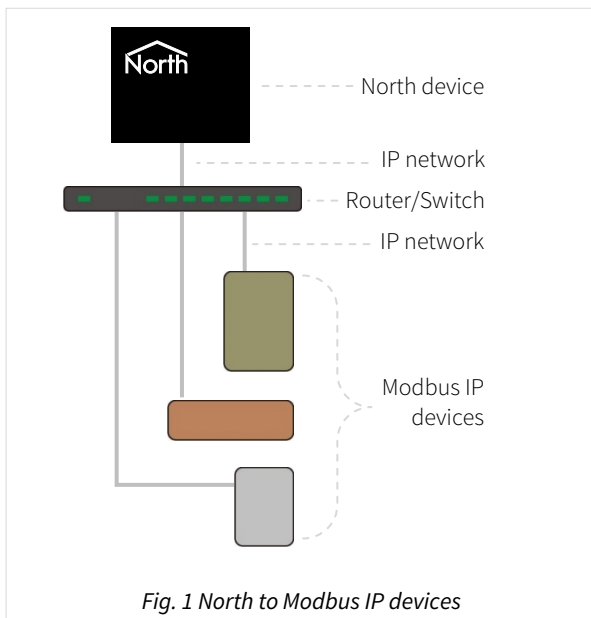
The ModbusTCP driver allows North to interface with a wide range of equipment supporting the Modbus over TCP/IP protocol. The driver can both request values from Modbus devices, and provide values to a Modbus front-end when requested.

Modbus over TCP/IP uses a client-server model. The ModbusTCP driver is capable of both requesting values from server devices (i.e. power meters, controllers, etc.), and providing values from the Essential Data and Extra Data within the North device when requested by a client device (i.e. display or front-end).

The driver connects to an IP network, and is capable of accessing up to 30 Modbus TCP/IP server devices on the network (Fig. 1). The driver is also capable of being accessed by five client devices simultaneously.

Modbus over serial line and JBus equipment are also compatible with the driver, when connected to the IP network via a Modbus gateway (Fig. 2).

Two alternative Modbus over serial line drivers are also available, which use a master-slave model. The Modbus driver provides a master device interface, capable of requesting values from a slave device. The ModbusSlave driver provides a slave device interface, capable of responding to the master device.



Equipment

Many different types of Modbus equipment are compatible with the driver, including:

- Energy meters
- Generators
- Variable speed drives
- PLCs
- AHUs
- Control systems

Equipment is available from many different manufacturers, including ABB, Autometers, Carel, Carlo Gavazzi, Carrier, Daikin, Danfoss, Mitsubishi Electric, Schneider Electric, Siemens, Socomec, Stulz, Swegon, Tyco Electronics, plus many more.

Application Notes are available for some of these devices, search North product documentation.

Values

The ModbusTCP driver is capable of both requesting values from a Modbus controller, and providing values when requested by a Modbus front-end.

Values available from a Modbus Controller

Depending on the type of Modbus equipment, each device can typically have the following primary types available:

- **Coils** – digital output, e.g. enable command
- **Discrete Inputs** – digital input, e.g. off/on, or alarm states
- **Holding registers** – analogue output, e.g. setpoint values
- **Input registers** – analogue input, e.g. meter readings

Read the Modbus register list, available from the equipment manufacturer, for the values available from a specific device.

Values available to a Modbus Front-end

The driver presents values from the North device's Essential Data and Extra Data as Modbus values, accessible to any client device on the Modbus network. Essential Data contains 640 values on Commander, and 1280 values on ObSys. If necessary, start the Extra Data driver (which requires an interface licence) for an additional 1024 values. Access to these values can be controlled by configuring privilege levels within the driver.

Values may be accessed using any of the following Modbus function codes:

Function code	Action
01	Read value as a Modbus coil (digital output) state
02	Read value as a Modbus discrete input (digital input) state
03	Read value as a Modbus holding register (analogue output) value
04	Read value as a Modbus input register (analogue input) value
05	Write value as a Modbus coil (digital output) state
06	Write value as a Modbus holding register (analogue output) value

The values from Essential Data may be accessed using any of the supported Modbus function codes. A single address range is used for all functions – 0...639 on Commander, and 0...1279 on ObSys. Refer to [Appendix A](#) for a list of Modbus register addresses and their corresponding Essential Data page/object reference.

If ExtraData is used, the extra 1024 values appear in registers that follow on from the existing Essential Value registers – i.e. Registers 640...1663 on Commander, or 1280...2303 on ObSys.

The value is translated to a Modbus register or state depending which Modbus function is used and how the value is configured in Essential Data:

	Essential Data Type			
	Number	Float	NoYes or OffOn	ENum
Input Register Holding Register	16-bit unsigned register value in the range 0...65535. When reading and the value is negative, the register is converted to a 16-bit signed value in the range -32768...32767	Value scaled 10^{dp} (dp is the number of decimal places configured in Essential Data). 16-bit unsigned register value in the range 0...65535. When reading and the value is negative, the register is converted to a 16-bit signed value in the range -32768...32767	'No' and 'Off' states are converted to the value 0 'Yes' and 'On' states are converted to the value 1 16-bit unsigned register value in the range 0...1.	16-bit unsigned register value in the range 0...65535
Discrete Input Coil	Binary on/off state. If the value is zero (0) then an 'off' state is returned, any other value returns an 'on' state			

Prerequisites

If an *Application Note* is not available for your Modbus controller, you will need a Modbus register list from the equipment manufacturer. This should describe the function codes or commands supported, register addresses available, and how register values are stored (16-bit integer, 32-bit integer, IEEE float, multipliers, etc.)

All Modbus devices must be configured with a unique address on the TCP/IP network. If the TCP port can be configured, then this should be set to 502. If you are connecting to Modbus over serial devices via a gateway, each device must be assigned a unique serial address.

If you are connecting to the Modbus IP network via a firewall, then the driver will require outbound access to controllers on TCP port 502. For inbound access from a Modbus front-end, TCP port 502 is also used.

Enable values to a Modbus front-end by setting the Server TCP Port object to 502). The driver will respond to requests with a unit identifier of 255, 0, or 1.

The ModbusTCP driver requires Essential Data v3.0 (build 01/09/2015) or later.

Using the Driver

On ObSys and Commander, the ModbusTCP driver is pre-installed. Using all of these North devices, you can use the driver to create an interface to a ModbusTCP system. Once started, you will need to set up the driver before it can communicate with the ModbusTCP system.

The ModbusTCP driver uses zero licence units.

Starting the Interface

- 📖 To start an interface using the ModbusTCP driver, follow these steps:
 - **Start Engineering** your North device using ObSys
 - Navigate to **Configuration, Interfaces**, and set an unused **Interface** to 'ModbusTCP' to start the particular interface
 - Navigate to the top-level of your North device and re-scan it

The driver setup object (Mc), labelled **ModbusTCP Setup**, should now be available.

Setting up the Driver

- 📖 To set up the driver, follow these steps:
 - Navigate to the **ModbusTCP Setup** object (Mc). For example, if you started interface 1 with the driver earlier, then the object reference will be 'M1'
 - Navigate to **Unit 1 Details** object and set the **IP Address** of a Modbus controller that you wish to communicate. If the controller is connected via a Modbus gateway, then set the **Serial Address**
 - Follow additional steps on the *Application Note*, if available for the device
 - Repeat for each Modbus controller that your wish to communicate
 - To enable the Modbus server, so the driver is accessible to a Modbus front-end, set the **Server TCP Port** to 502

Checking Communications

Scanning the Modbus System will respond with all units configured with an IP address in Modbus Setup. You can check the interface is communicating by and viewing values within a unit.

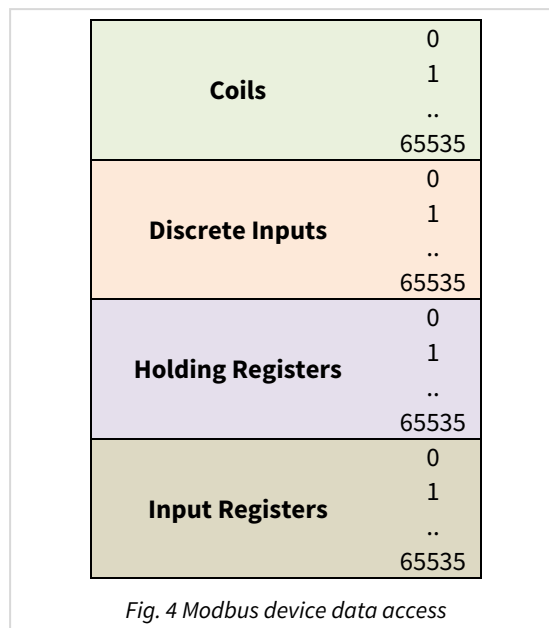
Understanding Modbus

Data Model

A Modbus controller stores data using four primary tables.

Table	Data type	Adjustable	Used for
Coils	1-bit data values	Read-Write	Digital outputs
Discrete Inputs	1-bit data values	Read-only	Digital inputs
Holding Registers	16-bit data values	Read-Write	Analogue outputs: setpoints, internally calculates values
Input Registers	16-bit data values	Read-only	Analogue inputs: sensor readings, meter values

Each table can contain up to 65536 entries, addressed in the range 0 to 65535 (Fig. 4).



Implementations of the Modbus protocol in a device can vary:

- It is common that these four tables may overlap in a device, so a single address range is used
- Distinctions between inputs and outputs, or 1-bit and 16-bit data values may be blurred
- Multiple consecutive table entries are combined to store a larger data value, e.g. 32-bit data values
- Register addresses are often documented in the range 1 to 65536. To rescale these in the 0 to 65535 range of the Modbus protocol, you will need to subtract 1. E.g. Register 100 becomes 99.

Function Codes

Values from a particular table are read and adjusted using different Modbus function codes. A device may only support some of these codes. The driver function is used as part of the object reference described later.

Table	Action	Function Code (read)	Function Code (write)	Driver Function
Coils	Read/Write digital output	01	05	C
Discrete Inputs	Read digital input	02		A
Holding Registers	Read/Write single output register	03	06	D
Holding Registers	Read/Write multiple output registers	03	16	E to L
Input Registers	Read single input register	04		B
Input Registers	Read multiple input registers	04		M to T

Value Decoding

The Modbus protocol only describes storing values as either a 1-bit digital or 16-bit register value. All implementations of Modbus have variations from this, including:

- 32-bit and 64-bit integer values
- IEEE floating point values
- Bit array in register
- Multipliers to change register data from integer
- ASCII string
- Byte-order changed

The driver has several decode types available that are used to translate a raw Modbus register, from a controller, into a value. The decode is used as part of the object reference described later.

Digital State (Decode A)

The value stored within a discrete input or coil entry is always 0 or 1, and always decodes to 0 or 1.

Unsigned Integer (Decode B)

The value stored within the register entry decodes to an unsigned number. For a single register, this will be in the range 0 to 65535.

Either one, two, or four registers can be decoded to a 16-bit, 32-bit, or 64-bit value respectively.

Examples

The single register value of 0x4849 (hex) will decode to 18505 (decimal). The multi-register values of 0x0012 and 0x3456 will decode to 1193046.

Signed Integer (Decode C)

The value stored within a register entry decodes to a signed integer number. For a single register, this will be in the range -32768 to 32767.

Either one, two, or four registers can be decoded to a 16-bit, 32-bit, or 64-bit value respectively.

Examples

The single register value of 0x4849 (hex) decodes to 18505 (decimal), and the single register value of 0xB7B7 decodes to -18505.

BCD in Lower Nibbles (Decode D)

The value stored within the lower nibbles of the register(s) decodes to a binary decoded decimal value.

One, two, three, or four register values can be decoded.

Example

The two lower nibbles of a single register decodes as Sum(V): $80+8+1 = 89$

Bit value (V)	Unused nibble				Lower nibble				Unused nibble				Lower nibble			
					80	40	20	10					8	4	2	1
Register value	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1

BCD in Register (Decode E)

The value stored within the register(s) decodes to a binary coded decimal value.

One, two, three, or four register values can be decoded.

Example

A single register value of 0x4849 (hex) decodes as Sum(V): $4000+800+40+8+1 = 4849$

Bit value (V)	8000	4000	2000	1000	800	400	200	100	80	40	20	10	8	4	2	1
Register value	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1

ASCII value in LSB (Decode F)

The value stored within the least significant byte of the register(s) decodes to a single ASCII character. Up to 16 registers can be accessed at once, i.e. 16 characters.

Example

The LSB of a single register decodes as a single character: $64+8+1 =$ ASCII code 73 = 'I'

Bit value									ASCII character							
									128	64	32	16	8	4	2	1
Register value	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1

ASCII String (Decode G)

The value stored within each register decodes to two ASCII characters. Up to 16 registers can be accessed at once, i.e. 32 characters.

Example

A single register decodes as two characters: $64+8 =$ ASCII code 72, and $64+8+1 =$ ASCII code 73. The full string is 'HI'.

Bit value	First ASCII character								Second ASCII character							
	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1
Register value	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1

Unsigned Integer in LSB (Decode H)

The value stored within the least significant byte (LSB) of a single register decodes to an unsigned number, a byte in the range 0 to 255. When writing, the register is first read to preserve the MSB.

Example

The LSB of a single register decodes as Sum(V): $64+8+1 = 73$

Bit value (V)	Unused MSB								LSB							
									128	64	32	16	8	4	2	1
Register value	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1

Unsigned Integer in MSB (Decode I)

The value stored within the most significant byte (MSB) of a single register decodes to an unsigned number, a byte in the range 0 to 255. When writing, the register is first read to preserve the LSB.

Example

The MSB of a single register decodes using $\text{Sum}(V)$: $64+8 = 72$

Bit value (V)	MSB								Unused LSB							
	128	64	32	16	8	4	2	1								
Register value	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1

IEEE Float (Decode J and M)

The value stored within two registers decodes to an IEEE floating-point number. A four register value can be decoded to a double-precision floating-point number.

Use decode type J for big-endian values (MSW in first register and LSW in second), or decode type M for little-endian values (LSW in first register and MSW in second).

Decode type J can also decode a single register as a half-precision floating-point number.

Single Bit of Register (Decode K)

Returns the specified bit number from a register value. Bits are numbered starting with the most significant byte of the first register, as shown in the example. Multiple register values are supported. When writing, the register is first read to preserve remaining bits.

Example

If a register has the value 0x4849 (hex). Bits 8, 11, 14, 3, and 6 have the value 1. All other bits have the value 0.

Bit number	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
Register value	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1

Bit Mask (Decode L)

Performs a bitwise AND operation on a register value and mask value. The result indicates which bits of the mask value are also set in the register value.

Example

If a register has the value 18505 (decimal). To find the value of bits 3 to 6, we first calculate the mask value: $\text{Sum}(V) = 64+32+16+8 = 120$. So, $\text{Register AND Mask} = 18505 \text{ AND } 120 = 72$.

We could additionally use a formula to rescale the value, i.e. $72 \times 0.125 = 9$

Bit value (V)	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Register value	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1
Mask value	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0

Object Specifications

Once an interface is started, one or more extra objects become available within the top-level object of the device. As with all North objects, each of these extra objects may contain sub-objects, (and each of these may contain sub-objects, and so on) – the whole object structure being a multi-layer hierarchy. It is possible to navigate around the objects using the ObSys Engineering Software.

Each object is specified below, along with its sub-objects.

Example Object Reference

An example of a reference to an object in the same device: the Modbus System (S1) contains a Unit (U1), which contains an Input register (B2.B). Therefore, the object reference will be 'S1.U1.B2.B'.

An example of a reference to an object in a different device: the IP network object (IP) contains Default Commander object (CDIP), which contains the object above (S1.U1.B2.B) – therefore the complete object reference is 'IP.CDIP.S1.U1.B2.B'.

Device Top-Level Objects

When an interface is started using the Modbus driver, the objects below become available within the top-level object of the device. For example, if Interface 1 is started, then the object with references 'M1' and 'S1' become available.

Description	Reference	Type
ModbusTCP Setup Set up the ModbusTCP driver, started on interface <i>c</i> (<i>c</i> is the interface number)	Mc	Fixed Container: On the Commander platform this will be <i>[CDM v20\ModbusTCP v30]</i> On the ObSys platform this will be <i>[OSM v20\ModbusTCP v30]</i>
Modbus System Access Modbus system connected to interface <i>c</i> (<i>c</i> is the interface number)	Sc	Variable Container: <i>[ModbusTCP]</i>

ModbusTCP Setup

Object Type: [OSM v20\ModbusTCP v30]

Object Type: [CDM v20\ModbusTCP v30]

The ModbusTCP Module contains the following objects:

Description	Reference	Type
System Label Label displayed when scanning the system	DL	Obj\Text: 20 chars max; Adjustable
Maximum Requests Maximum number of client devices that can be connected to simultaneously	MR	Obj\Num: 1...3; Adjustable
Server TCP Port Enables Modbus server function, so that a Modbus front-end can request values from the driver. If the Modbus front-end requires a unit identifier or serial address, the driver will respond to 255, 0, or 1.	SP	Obj\Num: 0...65535; Adjustable Value 0: disables Modbus server Standard Modbus TCP port is 502
Database Objects Available Count of maximum objects available from Essential Data and Extra Data	EDC	Obj\Num
Database Privilege Levels Configure privilege levels to control read and adjust access to Essential Data and Extra Data from a Modbus front-end	P	Fixed Container: On the Commander platform this will be [CDM v20\ModbusTCP v30\Privs] On the ObSys platform this will be [OSM v20\ModbusTCP v30\Privs]
Unit x Details Configure Modbus client device address. Unit number, x, is in the range 1...30	Ux	Fixed Container: On the Commander platform this will be [CDM v20\ModbusTCP v30\Unit] On the ObSys platform this will be [OSM v20\ModbusTCP v30\Unit]
Formula y User defined mathematical formula, used in decoding values from device. The formula number, y, is in the range 1...20	Fy	Fixed Container: [Standard\AMFormula]

Database Privilege Levels

Object Type: [OSM v20\ModbusTCP v30\Privs]

Object Type: [CDM v20\ModbusTCP v30\Privs]

Security Areas and Levels

Within the North security model, there are eight security areas. Security areas could be actual areas in a building, but are normally functional areas – for example, ‘environmental control’ and ‘North engineering’ areas would allow a user to have different privileges in controlling set points and engineering Commanders.

Typically, a user is assigned a privilege level in each of the eight areas. The level is in the range zero to seven, seven being the most powerful. When a user wishes to pass a door, his/her privilege level in the door’s area is checked against the minimum required for that area – and then either allowed to pass, or rejected.

The engineer must decide the use of the eight areas. The engineer must also decide the power of the privilege levels. Most systems use only a few levels per area: 0=None, 1=Guest, 2=User, 7=Administrator.

As an example, imagine a page of values in Essential Data. The page needs a user to have a minimum privilege level of 2 in area 1 before it can be viewed. The page is available in a Web browser that checks users with a security database. User A has privilege level 7 in area 1 – she can view the page. User B has privilege level 5 in area 1 – he can also view the page. User C has privilege level 1 in area 1 – she cannot view the page.

The example continues: within this page of values in Essential Data is a temperature set point object. Users need a minimum privilege level of 6 in area 1 to adjust it – therefore User A can adjust the set point, but User B cannot.

Specifying Access Security

Essential Data and Extra Data have Access Security objects to control who can view a page, and who can adjust an adjustable object.

Each Access Security object has a two-digit value. Each controls the access to a particular feature - such as viewing the page, or adjusting the value. The two-digit value is made up of the area digit (1-8), followed by the minimum privilege level (1-7) – for example, if the minimum privilege level is 6 in area 2, then the two digit value is 26. If the value is 00, then no security checks are made.

ModbusTCP Driver

The Database Privilege Levels object contains a privilege level for each of the eight security areas, representing a virtual user. The ModbusTCP driver uses these to control access to Essential Data and Extra Data when reading or adjusting a value.

Description	Reference	Type
Privilege Level in Area x The area, x, can be in the range 1...8	Px	Obj\Num; Adjustable; Range: 0...7

Unit Details

Object Type: [OSM v20\ModbusTCP v30\Unit]

Object Type: [CDM v20\ModbusTCP v30\Unit]

The Unit Details object is used to configure the address of a Modbus device on the network.

Description	Reference	Type
IP Address	IA	Obj\IP; Adjustable
Serial Address If connecting via a gateway, set to the Modbus over serial device address. If connecting to a single Modbus TCP device, set to the address 255.	A	Obj\Num: 1...255; Adjustable Value 1...247: Modbus serial address Value 255: Modbus TCP device
Device Type Leave blank to view default Modbus objects for a device. Refer to <i>Application Note</i> for other types available	T	Obj\Text: 20 chars; Adjustable

Formula Setup

Object Type: [Standard\AMFormula]

A standard formula setup is a maths module that allows values to be converted into engineering units.

This module allows a simple formula to be applied to a Modbus register value so that the resulting object value contains a meaningful value.

The module can convert the number into an object value using the formula:

$$\text{real-value} = (\text{M} \times \text{raw-value}) + \text{A}$$

The formula values M and A are engineer-defined. When writing, the module applies the formula below:

$$\text{raw-value} = (\text{real-value} - \text{A}) / \text{M}$$

Example

A register value contains a temperature value, where value 0 = -50°C and value 65535 = +50°C.

If A is set to -50 and M=0.0015259, then the formula would be:

$$\text{temperature} = (0.0015259 \times \text{register-value}) + -50$$

So, the following temperatures can be calculated from the register values:

Register value	Temperature
0	-50°C
32767	0°C
49150	25°C
65535	50°C

The module contains the following objects:

Description	Reference	Type
Addition value	A	Obj\Float; Adjustable
Multiplication value	M	Obj\Float; Adjustable

Modbus System

Object Type: *[ModbusTCP]*

The Modbus system contains objects to access the Modbus TCP client devices available.

Description	Reference	Type
Unit <i>x</i> The unit address, <i>x</i> , can be in the range 1...30	U <i>x</i>	Fixed container, one of the following: Default Modbus Device <i>[Modbus\Default]</i> If the <i>Device Type</i> is configured then the container will be of the type <i>[Modbus\Device Type]</i> . Refer to <i>Application Note</i> for more information

Default Modbus Device

Object Type: [Modbus\Default]

A default Modbus device contains a generic list of objects that enable you to access the values in a device. Use this with the device manufacturer's register list. For a full description of Modbus values and how to decode them, refer to the *Understanding Modbus* section earlier in this document.

Frequently Used Objects

Here we list a summary of the most frequently used decode objects. Refer to the *Notes* section below for additional information, and *Full Object List* below for the complete list of objects available.

Description	Reference	Type
Coil r – State The digital output, r , is in the range 0...65535 (see note 1)	Cr.A	Obj\OffOn; Adjustable
Discrete Input r – State The digital input, r , is in the range 0...65535 (see note 1)	Ar.A	Obj\OffOn
Holding Register r – Unsigned 16-bit Integer The register, r , is in the range 0...65535 (see note 1).	Dr.B	Obj\Num; Range: 0...65535; Adjustable
Holding Register r – Unsigned 16-bit (x10)	Dr.B26	Obj\Float; Range: 0...6553.5; Adjustable
Holding Register r – Unsigned 16-bit (x100)	Dr.B27	Obj\Float; Range: 0...655.35; Adjustable
Holding Register r – Unsigned 16-bit (x1000)	Dr.B28	Obj\Float; Range: 0...65.535; Adjustable
Holding Register r – Signed 16-bit Integer	Dr.C	Obj\Num; Range: -32768...32767; Adjustable
Holding Register r – Signed 16-bit (x10)	Dr.C26	Obj\Float; Range: -3276.8...3276.7; Adjustable
Holding Register r – Signed 16-bit (x100)	Dr.C27	Obj\Float; Range: -327.68...327.67; Adjustable
Holding Register r – Signed 16-bit (x1000)	Dr.C28	Obj\Float; Range: -32.768...32.767; Adjustable
Holding Register r – IEEE Float Single precision, stored in two registers (MSW, LSW)	Fr.J	Obj\Float; Adjustable
Holding Register r – Unsigned 32-bit Integer Stored in two registers (see note 2)	Fr.B	Obj\Num; Range: 0...4294967295; Adjustable
Holding Register r – Signed 32-bit Integer Stored in two registers (see note 2)	Fr.C	Obj\Num; Range: -2147483648...2147483647; Adjustable
Holding Register r – Unsigned 64-bit Integer Stored in four registers (see note 2)	Hr.B	Obj\Text; Range: 0...18446744073709551615; Adjustable
Holding Register r – Signed 64-bit Integer Stored in four registers (see note 2)	Hr.C	Obj\Text; Range: -923372036854775808...923372036854775807; Adjustable
Holding Register r – IEEE Double Float Double precision, stored in four registers (MSW... LSW) (see note 2)	Hr.J	Obj\Text; Adjustable

Description	Reference	Type
Holding Register r – Bit b The bit number, b , can be in the range 0...15. Refer to <i>Single Bit of Register</i> section earlier in this manual	Dr.Kb	Obj\OffOn; Adjustable
Input Register r – Unsigned 16-bit Integer The register, r , is in the range 0...65535 (see note 1)	Br.B	Obj\Num; Range: 0...65535
Input Register r – Unsigned 16-bit (x10)	Br.B26	Obj\Float; Range: 0...6553.5
Input Register r – Unsigned 16-bit (x100)	Br.B27	Obj\Float; Range: 0...655.35
Input Register r – Unsigned 16-bit (x1000)	Br.B28	Obj\Float; Range: 0...65.535
Input Register r – Signed 16-bit Integer	Br.C	Obj\Num; Range: -32768...32767
Input Register r – Signed 16-bit (x10)	Br.C26	Obj\Float; Range: -3276.8...3276.7
Input Register r – Signed 16-bit (x100)	Br.C27	Obj\Float; Range: -327.68...327.67
Input Register r – Signed 16-bit (x1000)	Br.C28	Obj\Num; Range: -32.768...32.767
Input Register r – IEEE Float Single precision, stored in two registers (MSW,LSW)	Nr.J	Obj\Float
Input Register r – Unsigned 32-bit Integer Stored in two registers (see note 2)	Nr.B	Obj\Num; Range: 0...4294967295
Input Register r – Signed 32-bit Integer Stored in two registers (see note 2)	Nr.C	Obj\Num; Range: -2147483648...2147483647
Input Register r – Unsigned 64-bit Integer Stored in four registers (see note 2)	Pr.B	Obj\Text; Range: 0...18446744073709551615
Input Register r – Signed 64-bit Integer Stored in four registers (see note 2)	Pr.C	Obj\Text; Range: -923372036854775808...923372036854775807
Input Register r – IEEE Double Float Double precision, stored in four registers (MSW... LSW) (see note 2)	Pr.J	Obj\Text
Input Register r – Bit b The bit number, b , can be in the range 0...15. Refer to <i>Single Bit of Register</i> section earlier in this manual	Br.Kb	Obj\OffOn

Notes

1. The discrete input, coil, or register number, r , is in the range 0...65535. Manufacturers sometimes document the register number in the range 1...65536. To rescale these in the 0 to 65535 range from the Modbus protocol, you will need to subtract 1. For example, register 100 becomes 99.
2. Large numbers: 64-bit and 32-bit values can contain up to 20 significant figures. Numbers this size are ok for displaying to a user, but may be too large to perform accurate maths functions. These values can be read in blocks of six significant figures by appending the object reference with a block number. Block 1 reads the six least significant figures, block 2 the next six significant figures, etc.
For example, if object 'H1.B' reads the 64-bit value '6744073709551615', then object 'H1.B.1' will read the least six significant figures '551615', object 'H1.B.2' the value '073709', and object 'H1.B.3' the value '6744'.
You can also use a formula with a block number. The object has the format 'H1.B28.1'. This will apply the formula first then access the requested block of six significant figures.

If the formula uses a divisor, then the value will be formatted to three decimal places. For example, object 'H1.B28.1' will read the value '551.615'.

Full Object List

The Modbus driver contains the following objects:

Description	Reference	Type
Function f, Entry r – Decode d The function, f, is in the range A...T (see note 3) The entry, r, is in the range 0...65535 (see note 1) The decode, d, is in the range A...M (see note 4) (For 2 and 4 register values, see note 5)	fr.d	The value type is dependent on the function, f, and decode, d. Adjustable for Coils and Holding Registers
Function f, Entry r – Decode d, Formula z As above, but with formula, z, applied (see note 2)	fr.dz	The value type is dependent on the function, f, decode, d, and formula, z. Adjustable for Coils and Holding Registers
Function f, Entry r – Bit b The function, f, is in the range B,D...T (see note 3) The entry, r, is in the range 0...65535 (see note 1) The bit number, b, can be in the range 0...15. Refer to Single Bit of Register section earlier in this manual	fr.Kb	Obj\OffOn; Adjustable for Holding Registers
Function f, Entry r – Bit Mask m The function, f, is in the range B,D...T (see note 3) The entry, r, is in the range 0...65535 (see note 1) The bit mask, m, is a number in the range 0...65535. Refer to Bit Mask section earlier in this manual	fr.Lm	Obj\Num
Function f, Entry r – Bit Mask m, Formula z As above, but with formula, z, applied (see note 2)	fr.Lm.z	Obj\Float

Notes

1. The discrete input, coil, or register number, *r*, is in the range 0...65535. Manufacturers sometimes document the register number in the range 1...65536. To rescale these in the 0 to 65535 range from the Modbus protocol, you will need to subtract 1. E.g. Register 100 becomes 99.
2. An optional formula number, *z*, may be applied where indicated above. The formula number is in the range 1...40, where 1...20 refer to user-defined formula, and 21...40 are fixed as follows:

Formula	Multiply	Add
21	10	0
22	100	0
23	1000	0

Formula	Multiply	Add
31	2	0
32	5	0
33	0.2	0

Formula	Multiply	Add
24	10000	0
25	100000	0
26	0.1	0
27	0.01	0
28	0.001	0
29	0.0001	0
30	0.00001	0

Formula	Multiply	Add
34	0.5	0
35	0.05	0
36	0.005	0
37	0.000001	0
38	1	0
39	1	0
40	Four quadrant power factor (Float decode only)	

3. The function, f , is the Modbus function or command and can be in the range A...T. Refer to the [Function Codes](#) section earlier in this document.

Driver Function	Table	Action	Function Code (read)	Function Code (write)
C	Coils	Read/Write digital output	01	05
A	Discrete Inputs	Read digital input	02	
D	Holding Registers	Read/Write 1 output register	03	06
E	Holding Registers	Read/Write 1 output multi-registers	03	16
F	Holding Registers	Read/Write 2 output multi-registers	03	16
G	Holding Registers	Read/Write 3 output multi-registers	03	16
H	Holding Registers	Read/Write 4 output multi-registers	03	16
I	Holding Registers	Read/Write 5 output multi-registers	03	16
J	Holding Registers	Read/Write 6 output multi-registers	03	16
K	Holding Registers	Read/Write 7 output multi-registers	03	16
L	Holding Registers	Read/Write 8 output multi-registers	03	16
B	Input Registers	Read 1 input register	04	
N	Input Registers	Read 2 input multi-registers	04	
O	Input Registers	Read 3 input multi-registers	04	
P	Input Registers	Read 4 input multi-registers	04	
Q	Input Registers	Read 5 input multi-registers	04	
R	Input Registers	Read 6 input multi-registers	04	
S	Input Registers	Read 7 input multi-registers	04	
T	Input Registers	Read 8 input multi-registers	04	

4. The decode, d , is in the range A...M. Refer to the [Value Decoding](#) section earlier in this document.

Decode	Use	Object Type
A	Digital State	Obj\OffOn
B	Unsigned Integer	Obj\Num
C	Signed Integer	Obj\Num
D	BCD in lower nibbles only	Obj\Num
E	BCD in register	Obj\Num
F	ASCII in LSB	Obj\Text
G	ASCII string	Obj\Text
H	Unsigned Integer in LSB	Obj\Num
I	Unsigned Integer in MSB	Obj\Num
J	IEEE Float (MSW, LSW order)	Obj\Float
K	Single bit of register	Obj\OffOn
L	Bit Mask	Obj\Num
M	IEEE Float (LSW, MSW order)	Obj\Float

5. Large numbers: 64-bit and 32-bit values can contain up to 20 significant figures. Numbers this size are ok for displaying to a user, but may be too large to perform accurate maths functions. These values can be read in blocks of six significant figures by appending the object reference with a block number. Block 1 reads the six least significant figures, block 2 the next six significant figures, etc.

For example, if object 'H1.B' reads the 64-bit value '6744073709551615', then object 'H1.B.1' will read the least six significant figures '551615', object 'H1.B.2' the value '073709', and object 'H1.B.3' the value '6744'.

You can also use a formula with a block number. The object has the format 'H1.B28.1'. This will apply the formula first then access the requested block of six significant figures.

If the formula uses a divisor, then the value will be formatted to three decimal places. For example, object 'H1.B28.1' will read the value '551.615'.

Appendix A: Modbus Register List

When the Server TCP Port object (Mc.SP) is set, the driver presents values from the North device's Essential Data and Extra Data as Modbus values. For further information on Modbus function codes supported, refer to the section *Values available to a Modbus front-end*.

Essential Data

Essential Data contains 640 values on Commander, and 1280 values on ObSys. These values are arranged in pages of 10, 16, 32, or 48 objects.

The values may be accessed using any of the supported Modbus function codes. A single register address range is used for all functions – 0...639 on Commander, and 0...1279 on ObSys.

This address maps to the Essential Data page/object reference as shown below. Download this list as a spreadsheet from http://resource.northbt.com/driver/Modbus_RegisterList.xlsx

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
0	P1.01	P1.01	P1.01	P1.01
1	P1.02	P1.02	P1.02	P1.02
2	P1.03	P1.03	P1.03	P1.03
3	P1.04	P1.04	P1.04	P1.04
4	P1.05	P1.05	P1.05	P1.05
5	P1.06	P1.06	P1.06	P1.06
6	P1.07	P1.07	P1.07	P1.07
7	P1.08	P1.08	P1.08	P1.08
8	P1.09	P1.09	P1.09	P1.09
9	P1.010	P1.010	P1.010	P1.010
10	P2.01	P1.011	P1.011	P1.011
11	P2.02	P1.012	P1.012	P1.012
12	P2.03	P1.013	P1.013	P1.013
13	P2.04	P1.014	P1.014	P1.014
14	P2.05	P1.015	P1.015	P1.015
15	P2.06	P1.016	P1.016	P1.016
16	P2.07	P2.01	P1.017	P1.017
17	P2.08	P2.02	P1.018	P1.018
18	P2.09	P2.03	P1.019	P1.019
19	P2.010	P2.04	P1.020	P1.020
20	P3.01	P2.05	P1.021	P1.021
21	P3.02	P2.06	P1.022	P1.022
22	P3.03	P2.07	P1.023	P1.023
23	P3.04	P2.08	P1.024	P1.024
24	P3.05	P2.09	P1.025	P1.025
25	P3.06	P2.010	P1.026	P1.026
26	P3.07	P2.011	P1.027	P1.027
27	P3.08	P2.012	P1.028	P1.028
28	P3.09	P2.013	P1.029	P1.029
29	P3.010	P2.014	P1.030	P1.030
30	P4.01	P2.015	P1.031	P1.031

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
31	P4.02	P2.016	P1.032	P1.032
32	P4.03	P3.01	P2.01	P1.033
33	P4.04	P3.02	P2.02	P1.034
34	P4.05	P3.03	P2.03	P1.035
35	P4.06	P3.04	P2.04	P1.036
36	P4.07	P3.05	P2.05	P1.037
37	P4.08	P3.06	P2.06	P1.038
38	P4.09	P3.07	P2.07	P1.039
39	P4.010	P3.08	P2.08	P1.040
40	P5.01	P3.09	P2.09	P1.041
41	P5.02	P3.010	P2.010	P1.042
42	P5.03	P3.011	P2.011	P1.043
43	P5.04	P3.012	P2.012	P1.044
44	P5.05	P3.013	P2.013	P1.045
45	P5.06	P3.014	P2.014	P1.046
46	P5.07	P3.015	P2.015	P1.047
47	P5.08	P3.016	P2.016	P1.048
48	P5.09	P4.01	P2.017	P1.049
49	P5.010	P4.02	P2.018	P1.050
50	P6.01	P4.03	P2.019	P1.051
51	P6.02	P4.04	P2.020	P1.052
52	P6.03	P4.05	P2.021	P1.053
53	P6.04	P4.06	P2.022	P1.054
54	P6.05	P4.07	P2.023	P1.055
55	P6.06	P4.08	P2.024	P1.056
56	P6.07	P4.09	P2.025	P1.057
57	P6.08	P4.010	P2.026	P1.058
58	P6.09	P4.011	P2.027	P1.059
59	P6.010	P4.012	P2.028	P1.060
60	P7.01	P4.013	P2.029	P1.061
61	P7.02	P4.014	P2.030	P1.062

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
62	P7.03	P4.015	P2.031	P1.063
63	P7.04	P4.016	P2.032	P1.064
64	P7.05	P5.01	P3.01	P2.01
65	P7.06	P5.02	P3.02	P2.02
66	P7.07	P5.03	P3.03	P2.03
67	P7.08	P5.04	P3.04	P2.04
68	P7.09	P5.05	P3.05	P2.05
69	P7.010	P5.06	P3.06	P2.06
70	P8.01	P5.07	P3.07	P2.07
71	P8.02	P5.08	P3.08	P2.08
72	P8.03	P5.09	P3.09	P2.09
73	P8.04	P5.010	P3.010	P2.010
74	P8.05	P5.011	P3.011	P2.011
75	P8.06	P5.012	P3.012	P2.012
76	P8.07	P5.013	P3.013	P2.013
77	P8.08	P5.014	P3.014	P2.014
78	P8.09	P5.015	P3.015	P2.015
79	P8.010	P5.016	P3.016	P2.016
80	P9.01	P6.01	P3.017	P2.017
81	P9.02	P6.02	P3.018	P2.018
82	P9.03	P6.03	P3.019	P2.019
83	P9.04	P6.04	P3.020	P2.020
84	P9.05	P6.05	P3.021	P2.021
85	P9.06	P6.06	P3.022	P2.022
86	P9.07	P6.07	P3.023	P2.023
87	P9.08	P6.08	P3.024	P2.024
88	P9.09	P6.09	P3.025	P2.025
89	P9.010	P6.010	P3.026	P2.026
90	P10.01	P6.011	P3.027	P2.027
91	P10.02	P6.012	P3.028	P2.028
92	P10.03	P6.013	P3.029	P2.029
93	P10.04	P6.014	P3.030	P2.030
94	P10.05	P6.015	P3.031	P2.031
95	P10.06	P6.016	P3.032	P2.032
96	P10.07	P7.01	P4.01	P2.033
97	P10.08	P7.02	P4.02	P2.034
98	P10.09	P7.03	P4.03	P2.035
99	P10.010	P7.04	P4.04	P2.036
100	P11.01	P7.05	P4.05	P2.037
101	P11.02	P7.06	P4.06	P2.038
102	P11.03	P7.07	P4.07	P2.039
103	P11.04	P7.08	P4.08	P2.040
104	P11.05	P7.09	P4.09	P2.041
105	P11.06	P7.010	P4.010	P2.042
106	P11.07	P7.011	P4.011	P2.043

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
107	P11.08	P7.012	P4.012	P2.044
108	P11.09	P7.013	P4.013	P2.045
109	P11.010	P7.014	P4.014	P2.046
110	P12.01	P7.015	P4.015	P2.047
111	P12.02	P7.016	P4.016	P2.048
112	P12.03	P8.01	P4.017	P2.049
113	P12.04	P8.02	P4.018	P2.050
114	P12.05	P8.03	P4.019	P2.051
115	P12.06	P8.04	P4.020	P2.052
116	P12.07	P8.05	P4.021	P2.053
117	P12.08	P8.06	P4.022	P2.054
118	P12.09	P8.07	P4.023	P2.055
119	P12.010	P8.08	P4.024	P2.056
120	P13.01	P8.09	P4.025	P2.057
121	P13.02	P8.010	P4.026	P2.058
122	P13.03	P8.011	P4.027	P2.059
123	P13.04	P8.012	P4.028	P2.060
124	P13.05	P8.013	P4.029	P2.061
125	P13.06	P8.014	P4.030	P2.062
126	P13.07	P8.015	P4.031	P2.063
127	P13.08	P8.016	P4.032	P2.064
128	P13.09	P9.01	P5.01	P3.01
129	P13.010	P9.02	P5.02	P3.02
130	P14.01	P9.03	P5.03	P3.03
131	P14.02	P9.04	P5.04	P3.04
132	P14.03	P9.05	P5.05	P3.05
133	P14.04	P9.06	P5.06	P3.06
134	P14.05	P9.07	P5.07	P3.07
135	P14.06	P9.08	P5.08	P3.08
136	P14.07	P9.09	P5.09	P3.09
137	P14.08	P9.010	P5.010	P3.010
138	P14.09	P9.011	P5.011	P3.011
139	P14.010	P9.012	P5.012	P3.012
140	P15.01	P9.013	P5.013	P3.013
141	P15.02	P9.014	P5.014	P3.014
142	P15.03	P9.015	P5.015	P3.015
143	P15.04	P9.016	P5.016	P3.016
144	P15.05	P10.01	P5.017	P3.017
145	P15.06	P10.02	P5.018	P3.018
146	P15.07	P10.03	P5.019	P3.019
147	P15.08	P10.04	P5.020	P3.020
148	P15.09	P10.05	P5.021	P3.021
149	P15.010	P10.06	P5.022	P3.022
150	P16.01	P10.07	P5.023	P3.023
151	P16.02	P10.08	P5.024	P3.024

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
152	P16.03	P10.09	P5.025	P3.025
153	P16.04	P10.010	P5.026	P3.026
154	P16.05	P10.011	P5.027	P3.027
155	P16.06	P10.012	P5.028	P3.028
156	P16.07	P10.013	P5.029	P3.029
157	P16.08	P10.014	P5.030	P3.030
158	P16.09	P10.015	P5.031	P3.031
159	P16.010	P10.016	P5.032	P3.032
160	P17.01	P11.01	P6.01	P3.033
161	P17.02	P11.02	P6.02	P3.034
162	P17.03	P11.03	P6.03	P3.035
163	P17.04	P11.04	P6.04	P3.036
164	P17.05	P11.05	P6.05	P3.037
165	P17.06	P11.06	P6.06	P3.038
166	P17.07	P11.07	P6.07	P3.039
167	P17.08	P11.08	P6.08	P3.040
168	P17.09	P11.09	P6.09	P3.041
169	P17.010	P11.010	P6.010	P3.042
170	P18.01	P11.011	P6.011	P3.043
171	P18.02	P11.012	P6.012	P3.044
172	P18.03	P11.013	P6.013	P3.045
173	P18.04	P11.014	P6.014	P3.046
174	P18.05	P11.015	P6.015	P3.047
175	P18.06	P11.016	P6.016	P3.048
176	P18.07	P12.01	P6.017	P3.049
177	P18.08	P12.02	P6.018	P3.050
178	P18.09	P12.03	P6.019	P3.051
179	P18.010	P12.04	P6.020	P3.052
180	P19.01	P12.05	P6.021	P3.053
181	P19.02	P12.06	P6.022	P3.054
182	P19.03	P12.07	P6.023	P3.055
183	P19.04	P12.08	P6.024	P3.056
184	P19.05	P12.09	P6.025	P3.057
185	P19.06	P12.010	P6.026	P3.058
186	P19.07	P12.011	P6.027	P3.059
187	P19.08	P12.012	P6.028	P3.060
188	P19.09	P12.013	P6.029	P3.061
189	P19.010	P12.014	P6.030	P3.062
190	P20.01	P12.015	P6.031	P3.063
191	P20.02	P12.016	P6.032	P3.064
192	P20.03	P13.01	P7.01	P4.01
193	P20.04	P13.02	P7.02	P4.02
194	P20.05	P13.03	P7.03	P4.03
195	P20.06	P13.04	P7.04	P4.04
196	P20.07	P13.05	P7.05	P4.05

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
197	P20.08	P13.06	P7.06	P4.06
198	P20.09	P13.07	P7.07	P4.07
199	P20.010	P13.08	P7.08	P4.08
200	P21.01	P13.09	P7.09	P4.09
201	P21.02	P13.010	P7.010	P4.010
202	P21.03	P13.011	P7.011	P4.011
203	P21.04	P13.012	P7.012	P4.012
204	P21.05	P13.013	P7.013	P4.013
205	P21.06	P13.014	P7.014	P4.014
206	P21.07	P13.015	P7.015	P4.015
207	P21.08	P13.016	P7.016	P4.016
208	P21.09	P14.01	P7.017	P4.017
209	P21.010	P14.02	P7.018	P4.018
210	P22.01	P14.03	P7.019	P4.019
211	P22.02	P14.04	P7.020	P4.020
212	P22.03	P14.05	P7.021	P4.021
213	P22.04	P14.06	P7.022	P4.022
214	P22.05	P14.07	P7.023	P4.023
215	P22.06	P14.08	P7.024	P4.024
216	P22.07	P14.09	P7.025	P4.025
217	P22.08	P14.010	P7.026	P4.026
218	P22.09	P14.011	P7.027	P4.027
219	P22.010	P14.012	P7.028	P4.028
220	P23.01	P14.013	P7.029	P4.029
221	P23.02	P14.014	P7.030	P4.030
222	P23.03	P14.015	P7.031	P4.031
223	P23.04	P14.016	P7.032	P4.032
224	P23.05	P15.01	P8.01	P4.033
225	P23.06	P15.02	P8.02	P4.034
226	P23.07	P15.03	P8.03	P4.035
227	P23.08	P15.04	P8.04	P4.036
228	P23.09	P15.05	P8.05	P4.037
229	P23.010	P15.06	P8.06	P4.038
230	P24.01	P15.07	P8.07	P4.039
231	P24.02	P15.08	P8.08	P4.040
232	P24.03	P15.09	P8.09	P4.041
233	P24.04	P15.010	P8.010	P4.042
234	P24.05	P15.011	P8.011	P4.043
235	P24.06	P15.012	P8.012	P4.044
236	P24.07	P15.013	P8.013	P4.045
237	P24.08	P15.014	P8.014	P4.046
238	P24.09	P15.015	P8.015	P4.047
239	P24.010	P15.016	P8.016	P4.048
240	P25.01	P16.01	P8.017	P4.049
241	P25.02	P16.02	P8.018	P4.050

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
242	P25.03	P16.03	P8.019	P4.051
243	P25.04	P16.04	P8.020	P4.052
244	P25.05	P16.05	P8.021	P4.053
245	P25.06	P16.06	P8.022	P4.054
246	P25.07	P16.07	P8.023	P4.055
247	P25.08	P16.08	P8.024	P4.056
248	P25.09	P16.09	P8.025	P4.057
249	P25.010	P16.010	P8.026	P4.058
250	P26.01	P16.011	P8.027	P4.059
251	P26.02	P16.012	P8.028	P4.060
252	P26.03	P16.013	P8.029	P4.061
253	P26.04	P16.014	P8.030	P4.062
254	P26.05	P16.015	P8.031	P4.063
255	P26.06	P16.016	P8.032	P4.064
256	P26.07	P17.01	P9.01	P5.01
257	P26.08	P17.02	P9.02	P5.02
258	P26.09	P17.03	P9.03	P5.03
259	P26.010	P17.04	P9.04	P5.04
260	P27.01	P17.05	P9.05	P5.05
261	P27.02	P17.06	P9.06	P5.06
262	P27.03	P17.07	P9.07	P5.07
263	P27.04	P17.08	P9.08	P5.08
264	P27.05	P17.09	P9.09	P5.09
265	P27.06	P17.010	P9.010	P5.010
266	P27.07	P17.011	P9.011	P5.011
267	P27.08	P17.012	P9.012	P5.012
268	P27.09	P17.013	P9.013	P5.013
269	P27.010	P17.014	P9.014	P5.014
270	P28.01	P17.015	P9.015	P5.015
271	P28.02	P17.016	P9.016	P5.016
272	P28.03	P18.01	P9.017	P5.017
273	P28.04	P18.02	P9.018	P5.018
274	P28.05	P18.03	P9.019	P5.019
275	P28.06	P18.04	P9.020	P5.020
276	P28.07	P18.05	P9.021	P5.021
277	P28.08	P18.06	P9.022	P5.022
278	P28.09	P18.07	P9.023	P5.023
279	P28.010	P18.08	P9.024	P5.024
280	P29.01	P18.09	P9.025	P5.025
281	P29.02	P18.010	P9.026	P5.026
282	P29.03	P18.011	P9.027	P5.027
283	P29.04	P18.012	P9.028	P5.028
284	P29.05	P18.013	P9.029	P5.029
285	P29.06	P18.014	P9.030	P5.030
286	P29.07	P18.015	P9.031	P5.031

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
287	P29.08	P18.016	P9.032	P5.032
288	P29.09	P19.01	P10.01	P5.033
289	P29.010	P19.02	P10.02	P5.034
290	P30.01	P19.03	P10.03	P5.035
291	P30.02	P19.04	P10.04	P5.036
292	P30.03	P19.05	P10.05	P5.037
293	P30.04	P19.06	P10.06	P5.038
294	P30.05	P19.07	P10.07	P5.039
295	P30.06	P19.08	P10.08	P5.040
296	P30.07	P19.09	P10.09	P5.041
297	P30.08	P19.010	P10.010	P5.042
298	P30.09	P19.011	P10.011	P5.043
299	P30.010	P19.012	P10.012	P5.044
300	P31.01	P19.013	P10.013	P5.045
301	P31.02	P19.014	P10.014	P5.046
302	P31.03	P19.015	P10.015	P5.047
303	P31.04	P19.016	P10.016	P5.048
304	P31.05	P20.01	P10.017	P5.049
305	P31.06	P20.02	P10.018	P5.050
306	P31.07	P20.03	P10.019	P5.051
307	P31.08	P20.04	P10.020	P5.052
308	P31.09	P20.05	P10.021	P5.053
309	P31.010	P20.06	P10.022	P5.054
310	P32.01	P20.07	P10.023	P5.055
311	P32.02	P20.08	P10.024	P5.056
312	P32.03	P20.09	P10.025	P5.057
313	P32.04	P20.010	P10.026	P5.058
314	P32.05	P20.011	P10.027	P5.059
315	P32.06	P20.012	P10.028	P5.060
316	P32.07	P20.013	P10.029	P5.061
317	P32.08	P20.014	P10.030	P5.062
318	P32.09	P20.015	P10.031	P5.063
319	P32.010	P20.016	P10.032	P5.064
320	P33.01	P21.01	P11.01	P6.01
321	P33.02	P21.02	P11.02	P6.02
322	P33.03	P21.03	P11.03	P6.03
323	P33.04	P21.04	P11.04	P6.04
324	P33.05	P21.05	P11.05	P6.05
325	P33.06	P21.06	P11.06	P6.06
326	P33.07	P21.07	P11.07	P6.07
327	P33.08	P21.08	P11.08	P6.08
328	P33.09	P21.09	P11.09	P6.09
329	P33.010	P21.010	P11.010	P6.010
330	P34.01	P21.011	P11.011	P6.011
331	P34.02	P21.012	P11.012	P6.012

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
332	P34.03	P21.013	P11.013	P6.013
333	P34.04	P21.014	P11.014	P6.014
334	P34.05	P21.015	P11.015	P6.015
335	P34.06	P21.016	P11.016	P6.016
336	P34.07	P22.01	P11.017	P6.017
337	P34.08	P22.02	P11.018	P6.018
338	P34.09	P22.03	P11.019	P6.019
339	P34.010	P22.04	P11.020	P6.020
340	P35.01	P22.05	P11.021	P6.021
341	P35.02	P22.06	P11.022	P6.022
342	P35.03	P22.07	P11.023	P6.023
343	P35.04	P22.08	P11.024	P6.024
344	P35.05	P22.09	P11.025	P6.025
345	P35.06	P22.010	P11.026	P6.026
346	P35.07	P22.011	P11.027	P6.027
347	P35.08	P22.012	P11.028	P6.028
348	P35.09	P22.013	P11.029	P6.029
349	P35.010	P22.014	P11.030	P6.030
350	P36.01	P22.015	P11.031	P6.031
351	P36.02	P22.016	P11.032	P6.032
352	P36.03	P23.01	P12.01	P6.033
353	P36.04	P23.02	P12.02	P6.034
354	P36.05	P23.03	P12.03	P6.035
355	P36.06	P23.04	P12.04	P6.036
356	P36.07	P23.05	P12.05	P6.037
357	P36.08	P23.06	P12.06	P6.038
358	P36.09	P23.07	P12.07	P6.039
359	P36.010	P23.08	P12.08	P6.040
360	P37.01	P23.09	P12.09	P6.041
361	P37.02	P23.010	P12.010	P6.042
362	P37.03	P23.011	P12.011	P6.043
363	P37.04	P23.012	P12.012	P6.044
364	P37.05	P23.013	P12.013	P6.045
365	P37.06	P23.014	P12.014	P6.046
366	P37.07	P23.015	P12.015	P6.047
367	P37.08	P23.016	P12.016	P6.048
368	P37.09	P24.01	P12.017	P6.049
369	P37.010	P24.02	P12.018	P6.050
370	P38.01	P24.03	P12.019	P6.051
371	P38.02	P24.04	P12.020	P6.052
372	P38.03	P24.05	P12.021	P6.053
373	P38.04	P24.06	P12.022	P6.054
374	P38.05	P24.07	P12.023	P6.055
375	P38.06	P24.08	P12.024	P6.056
376	P38.07	P24.09	P12.025	P6.057

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
377	P38.08	P24.010	P12.026	P6.058
378	P38.09	P24.011	P12.027	P6.059
379	P38.010	P24.012	P12.028	P6.060
380	P39.01	P24.013	P12.029	P6.061
381	P39.02	P24.014	P12.030	P6.062
382	P39.03	P24.015	P12.031	P6.063
383	P39.04	P24.016	P12.032	P6.064
384	P39.05	P25.01	P13.01	P7.01
385	P39.06	P25.02	P13.02	P7.02
386	P39.07	P25.03	P13.03	P7.03
387	P39.08	P25.04	P13.04	P7.04
388	P39.09	P25.05	P13.05	P7.05
389	P39.010	P25.06	P13.06	P7.06
390	P40.01	P25.07	P13.07	P7.07
391	P40.02	P25.08	P13.08	P7.08
392	P40.03	P25.09	P13.09	P7.09
393	P40.04	P25.010	P13.010	P7.010
394	P40.05	P25.011	P13.011	P7.011
395	P40.06	P25.012	P13.012	P7.012
396	P40.07	P25.013	P13.013	P7.013
397	P40.08	P25.014	P13.014	P7.014
398	P40.09	P25.015	P13.015	P7.015
399	P40.010	P25.016	P13.016	P7.016
400	P41.01	P26.01	P13.017	P7.017
401	P41.02	P26.02	P13.018	P7.018
402	P41.03	P26.03	P13.019	P7.019
403	P41.04	P26.04	P13.020	P7.020
404	P41.05	P26.05	P13.021	P7.021
405	P41.06	P26.06	P13.022	P7.022
406	P41.07	P26.07	P13.023	P7.023
407	P41.08	P26.08	P13.024	P7.024
408	P41.09	P26.09	P13.025	P7.025
409	P41.010	P26.010	P13.026	P7.026
410	P42.01	P26.011	P13.027	P7.027
411	P42.02	P26.012	P13.028	P7.028
412	P42.03	P26.013	P13.029	P7.029
413	P42.04	P26.014	P13.030	P7.030
414	P42.05	P26.015	P13.031	P7.031
415	P42.06	P26.016	P13.032	P7.032
416	P42.07	P27.01	P14.01	P7.033
417	P42.08	P27.02	P14.02	P7.034
418	P42.09	P27.03	P14.03	P7.035
419	P42.010	P27.04	P14.04	P7.036
420	P43.01	P27.05	P14.05	P7.037
421	P43.02	P27.06	P14.06	P7.038

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
422	P43.03	P27.07	P14.07	P7.039
423	P43.04	P27.08	P14.08	P7.040
424	P43.05	P27.09	P14.09	P7.041
425	P43.06	P27.010	P14.010	P7.042
426	P43.07	P27.011	P14.011	P7.043
427	P43.08	P27.012	P14.012	P7.044
428	P43.09	P27.013	P14.013	P7.045
429	P43.010	P27.014	P14.014	P7.046
430	P44.01	P27.015	P14.015	P7.047
431	P44.02	P27.016	P14.016	P7.048
432	P44.03	P28.01	P14.017	P7.049
433	P44.04	P28.02	P14.018	P7.050
434	P44.05	P28.03	P14.019	P7.051
435	P44.06	P28.04	P14.020	P7.052
436	P44.07	P28.05	P14.021	P7.053
437	P44.08	P28.06	P14.022	P7.054
438	P44.09	P28.07	P14.023	P7.055
439	P44.010	P28.08	P14.024	P7.056
440	P45.01	P28.09	P14.025	P7.057
441	P45.02	P28.010	P14.026	P7.058
442	P45.03	P28.011	P14.027	P7.059
443	P45.04	P28.012	P14.028	P7.060
444	P45.05	P28.013	P14.029	P7.061
445	P45.06	P28.014	P14.030	P7.062
446	P45.07	P28.015	P14.031	P7.063
447	P45.08	P28.016	P14.032	P7.064
448	P45.09	P29.01	P15.01	P8.01
449	P45.010	P29.02	P15.02	P8.02
450	P46.01	P29.03	P15.03	P8.03
451	P46.02	P29.04	P15.04	P8.04
452	P46.03	P29.05	P15.05	P8.05
453	P46.04	P29.06	P15.06	P8.06
454	P46.05	P29.07	P15.07	P8.07
455	P46.06	P29.08	P15.08	P8.08
456	P46.07	P29.09	P15.09	P8.09
457	P46.08	P29.010	P15.010	P8.010
458	P46.09	P29.011	P15.011	P8.011
459	P46.010	P29.012	P15.012	P8.012
460	P47.01	P29.013	P15.013	P8.013
461	P47.02	P29.014	P15.014	P8.014
462	P47.03	P29.015	P15.015	P8.015
463	P47.04	P29.016	P15.016	P8.016
464	P47.05	P30.01	P15.017	P8.017
465	P47.06	P30.02	P15.018	P8.018
466	P47.07	P30.03	P15.019	P8.019

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
467	P47.08	P30.04	P15.020	P8.020
468	P47.09	P30.05	P15.021	P8.021
469	P47.010	P30.06	P15.022	P8.022
470	P48.01	P30.07	P15.023	P8.023
471	P48.02	P30.08	P15.024	P8.024
472	P48.03	P30.09	P15.025	P8.025
473	P48.04	P30.010	P15.026	P8.026
474	P48.05	P30.011	P15.027	P8.027
475	P48.06	P30.012	P15.028	P8.028
476	P48.07	P30.013	P15.029	P8.029
477	P48.08	P30.014	P15.030	P8.030
478	P48.09	P30.015	P15.031	P8.031
479	P48.010	P30.016	P15.032	P8.032
480	P49.01	P31.01	P16.01	P8.033
481	P49.02	P31.02	P16.02	P8.034
482	P49.03	P31.03	P16.03	P8.035
483	P49.04	P31.04	P16.04	P8.036
484	P49.05	P31.05	P16.05	P8.037
485	P49.06	P31.06	P16.06	P8.038
486	P49.07	P31.07	P16.07	P8.039
487	P49.08	P31.08	P16.08	P8.040
488	P49.09	P31.09	P16.09	P8.041
489	P49.010	P31.010	P16.010	P8.042
490	P50.01	P31.011	P16.011	P8.043
491	P50.02	P31.012	P16.012	P8.044
492	P50.03	P31.013	P16.013	P8.045
493	P50.04	P31.014	P16.014	P8.046
494	P50.05	P31.015	P16.015	P8.047
495	P50.06	P31.016	P16.016	P8.048
496	P50.07	P32.01	P16.017	P8.049
497	P50.08	P32.02	P16.018	P8.050
498	P50.09	P32.03	P16.019	P8.051
499	P50.010	P32.04	P16.020	P8.052
500	P51.01	P32.05	P16.021	P8.053
501	P51.02	P32.06	P16.022	P8.054
502	P51.03	P32.07	P16.023	P8.055
503	P51.04	P32.08	P16.024	P8.056
504	P51.05	P32.09	P16.025	P8.057
505	P51.06	P32.010	P16.026	P8.058
506	P51.07	P32.011	P16.027	P8.059
507	P51.08	P32.012	P16.028	P8.060
508	P51.09	P32.013	P16.029	P8.061
509	P51.010	P32.014	P16.030	P8.062
510	P52.01	P32.015	P16.031	P8.063
511	P52.02	P32.016	P16.032	P8.064

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
512	P52.03	P33.01	P17.01	P9.01
513	P52.04	P33.02	P17.02	P9.02
514	P52.05	P33.03	P17.03	P9.03
515	P52.06	P33.04	P17.04	P9.04
516	P52.07	P33.05	P17.05	P9.05
517	P52.08	P33.06	P17.06	P9.06
518	P52.09	P33.07	P17.07	P9.07
519	P52.010	P33.08	P17.08	P9.08
520	P53.01	P33.09	P17.09	P9.09
521	P53.02	P33.010	P17.010	P9.010
522	P53.03	P33.011	P17.011	P9.011
523	P53.04	P33.012	P17.012	P9.012
524	P53.05	P33.013	P17.013	P9.013
525	P53.06	P33.014	P17.014	P9.014
526	P53.07	P33.015	P17.015	P9.015
527	P53.08	P33.016	P17.016	P9.016
528	P53.09	P34.01	P17.017	P9.017
529	P53.010	P34.02	P17.018	P9.018
530	P54.01	P34.03	P17.019	P9.019
531	P54.02	P34.04	P17.020	P9.020
532	P54.03	P34.05	P17.021	P9.021
533	P54.04	P34.06	P17.022	P9.022
534	P54.05	P34.07	P17.023	P9.023
535	P54.06	P34.08	P17.024	P9.024
536	P54.07	P34.09	P17.025	P9.025
537	P54.08	P34.010	P17.026	P9.026
538	P54.09	P34.011	P17.027	P9.027
539	P54.010	P34.012	P17.028	P9.028
540	P55.01	P34.013	P17.029	P9.029
541	P55.02	P34.014	P17.030	P9.030
542	P55.03	P34.015	P17.031	P9.031
543	P55.04	P34.016	P17.032	P9.032
544	P55.05	P35.01	P18.01	P9.033
545	P55.06	P35.02	P18.02	P9.034
546	P55.07	P35.03	P18.03	P9.035
547	P55.08	P35.04	P18.04	P9.036
548	P55.09	P35.05	P18.05	P9.037
549	P55.010	P35.06	P18.06	P9.038
550	P56.01	P35.07	P18.07	P9.039
551	P56.02	P35.08	P18.08	P9.040
552	P56.03	P35.09	P18.09	P9.041
553	P56.04	P35.010	P18.010	P9.042
554	P56.05	P35.011	P18.011	P9.043
555	P56.06	P35.012	P18.012	P9.044
556	P56.07	P35.013	P18.013	P9.045

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
557	P56.08	P35.014	P18.014	P9.046
558	P56.09	P35.015	P18.015	P9.047
559	P56.010	P35.016	P18.016	P9.048
560	P57.01	P36.01	P18.017	P9.049
561	P57.02	P36.02	P18.018	P9.050
562	P57.03	P36.03	P18.019	P9.051
563	P57.04	P36.04	P18.020	P9.052
564	P57.05	P36.05	P18.021	P9.053
565	P57.06	P36.06	P18.022	P9.054
566	P57.07	P36.07	P18.023	P9.055
567	P57.08	P36.08	P18.024	P9.056
568	P57.09	P36.09	P18.025	P9.057
569	P57.010	P36.010	P18.026	P9.058
570	P58.01	P36.011	P18.027	P9.059
571	P58.02	P36.012	P18.028	P9.060
572	P58.03	P36.013	P18.029	P9.061
573	P58.04	P36.014	P18.030	P9.062
574	P58.05	P36.015	P18.031	P9.063
575	P58.06	P36.016	P18.032	P9.064
576	P58.07	P37.01	P19.01	P10.01
577	P58.08	P37.02	P19.02	P10.02
578	P58.09	P37.03	P19.03	P10.03
579	P58.010	P37.04	P19.04	P10.04
580	P59.01	P37.05	P19.05	P10.05
581	P59.02	P37.06	P19.06	P10.06
582	P59.03	P37.07	P19.07	P10.07
583	P59.04	P37.08	P19.08	P10.08
584	P59.05	P37.09	P19.09	P10.09
585	P59.06	P37.010	P19.010	P10.010
586	P59.07	P37.011	P19.011	P10.011
587	P59.08	P37.012	P19.012	P10.012
588	P59.09	P37.013	P19.013	P10.013
589	P59.010	P37.014	P19.014	P10.014
590	P60.01	P37.015	P19.015	P10.015
591	P60.02	P37.016	P19.016	P10.016
592	P60.03	P38.01	P19.017	P10.017
593	P60.04	P38.02	P19.018	P10.018
594	P60.05	P38.03	P19.019	P10.019
595	P60.06	P38.04	P19.020	P10.020
596	P60.07	P38.05	P19.021	P10.021
597	P60.08	P38.06	P19.022	P10.022
598	P60.09	P38.07	P19.023	P10.023
599	P60.010	P38.08	P19.024	P10.024
600	P61.01	P38.09	P19.025	P10.025
601	P61.02	P38.010	P19.026	P10.026

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
602	P61.03	P38.011	P19.027	P10.027
603	P61.04	P38.012	P19.028	P10.028
604	P61.05	P38.013	P19.029	P10.029
605	P61.06	P38.014	P19.030	P10.030
606	P61.07	P38.015	P19.031	P10.031
607	P61.08	P38.016	P19.032	P10.032
608	P61.09	P39.01	P20.01	P10.033
609	P61.010	P39.02	P20.02	P10.034
610	P62.01	P39.03	P20.03	P10.035
611	P62.02	P39.04	P20.04	P10.036
612	P62.03	P39.05	P20.05	P10.037
613	P62.04	P39.06	P20.06	P10.038
614	P62.05	P39.07	P20.07	P10.039
615	P62.06	P39.08	P20.08	P10.040
616	P62.07	P39.09	P20.09	P10.041
617	P62.08	P39.010	P20.010	P10.042
618	P62.09	P39.011	P20.011	P10.043
619	P62.010	P39.012	P20.012	P10.044
620	P63.01	P39.013	P20.013	P10.045

Modbus Register Address	Essential Data Object Reference (Objects configured per page)			
	x10	x16	x32	x64
621	P63.02	P39.014	P20.014	P10.046
622	P63.03	P39.015	P20.015	P10.047
623	P63.04	P39.016	P20.016	P10.048
624	P63.05	P40.01	P20.017	P10.049
625	P63.06	P40.02	P20.018	P10.050
626	P63.07	P40.03	P20.019	P10.051
627	P63.08	P40.04	P20.020	P10.052
628	P63.09	P40.05	P20.021	P10.053
629	P63.010	P40.06	P20.022	P10.054
630	P64.01	P40.07	P20.023	P10.055
631	P64.02	P40.08	P20.024	P10.056
632	P64.03	P40.09	P20.025	P10.057
633	P64.04	P40.010	P20.026	P10.058
634	P64.05	P40.011	P20.027	P10.059
635	P64.06	P40.012	P20.028	P10.060
636	P64.07	P40.013	P20.029	P10.061
637	P64.08	P40.014	P20.030	P10.062
638	P64.09	P40.015	P20.031	P10.063
639	P64.010	P40.016	P20.032	P10.064

Extra Data

Extra Data contains 1024 values. These values are arranged in 16 pages of 64 objects.

Once the interface is started, the values may be accessed using any of the supported Modbus function codes. These values appear in registers that follow on from the existing Essential Value registers – i.e. Registers 640...1663 on Commander, or 1280...2303 on ObSys.

For a complete list of Modbus registers and the Extra Data page/object reference, download the spreadsheet from http://resource.northbt.com/driver/Modbus_RegisterList.xlsx

Driver Versions

Version	Build Date	Details
1.0	20/6/2008	Released for ObSys platform
2.0	11/10/2010	Modbus server feature added
2.0	1/4/2011	Released for Commander platform
2.0	31/10/2012	Default unit type now 'Modbus\Default' to match Modbus driver
2.0	26/9/2013	Fixed problem with writing register bit (decode K) objects Serial address defaulted to 255
3.0	08/2/2016	ExtraData compatibility added 64-bit value support added Decode types H & I, now read before writing value. Added formulas 37 and 40 Added ability to read in blocks of 6 sig figs, for 32-bit and 64-bit values
3.0	18/6/2018	Fix reading 64-bit float values when 0 (previously returned blank)

Next Steps...

If you require help, contact support on 01273 694422 or visit www.northbt.com/support



North Building Technologies Ltd
+44 (0) 1273 694422
support@northbt.com
www.northbt.com

This document is subject to change without notice and does not represent any commitment by North Building Technologies Ltd.

ObSys and Commander are trademarks of North Building Technologies Ltd. All other trademarks are property of their respective owners.

© Copyright 2018 North Building Technologies Limited.

Author: BS
Checked by: JF

Document issued 18/06/2018.