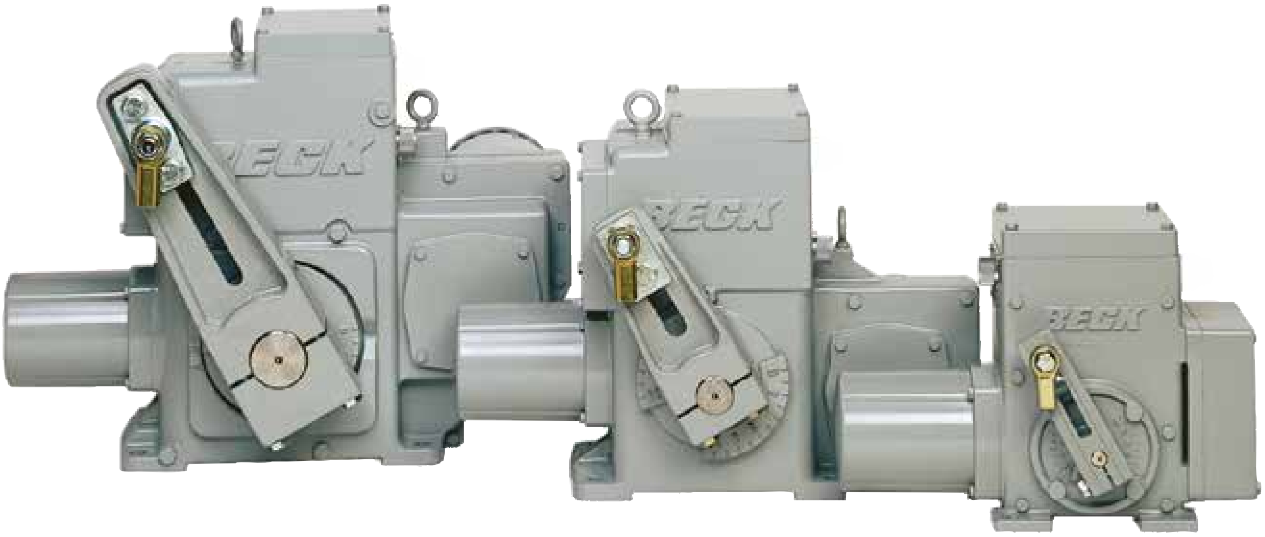
80-1103-03
Rev. 01.9

SUPPLEMENT TO MANUAL:
- 80-1103-00

MODEL GROUP 11
MODBUS RTU & TCP INTERFACE SUPPLEMENT

The instructions and procedures for the Installation, Operation, Calibration and Maintenance of Beck Group 11 Actuators are the same as listed in the above manual, except for the differences pertaining to the Modbus RTU & TCP interface which are detailed herein (covers Modbus interfaces with FC17 Server IDs 0-2).



ELECTRIC ACTUATORS FOR INDUSTRIAL PROCESS CONTROL

TABLE OF CONTENTS

Modbus RTU

PRECAUTIONARY INFORMATION

WARNING

Repeated changes to actuator configuration parameters within a control algorithm may result in process instability and actuator damage.

NOTE: When network security is a concern, the actuator should be placed in write protect mode.

INSTALLATION

TERMINAL BLOCK ASSIGNMENTS

Terminal Block Assignments		
Terminal	**Common Name**	**Modbus Name**
AA	DATA+	D1
BB	DATA-	D0
DD	COMMON	COMMON

RECOMMENDED CABLE

The following cable has been documented as meeting Modbus interface requirements:

- Belden 3106A
- Alpha Wire 6454

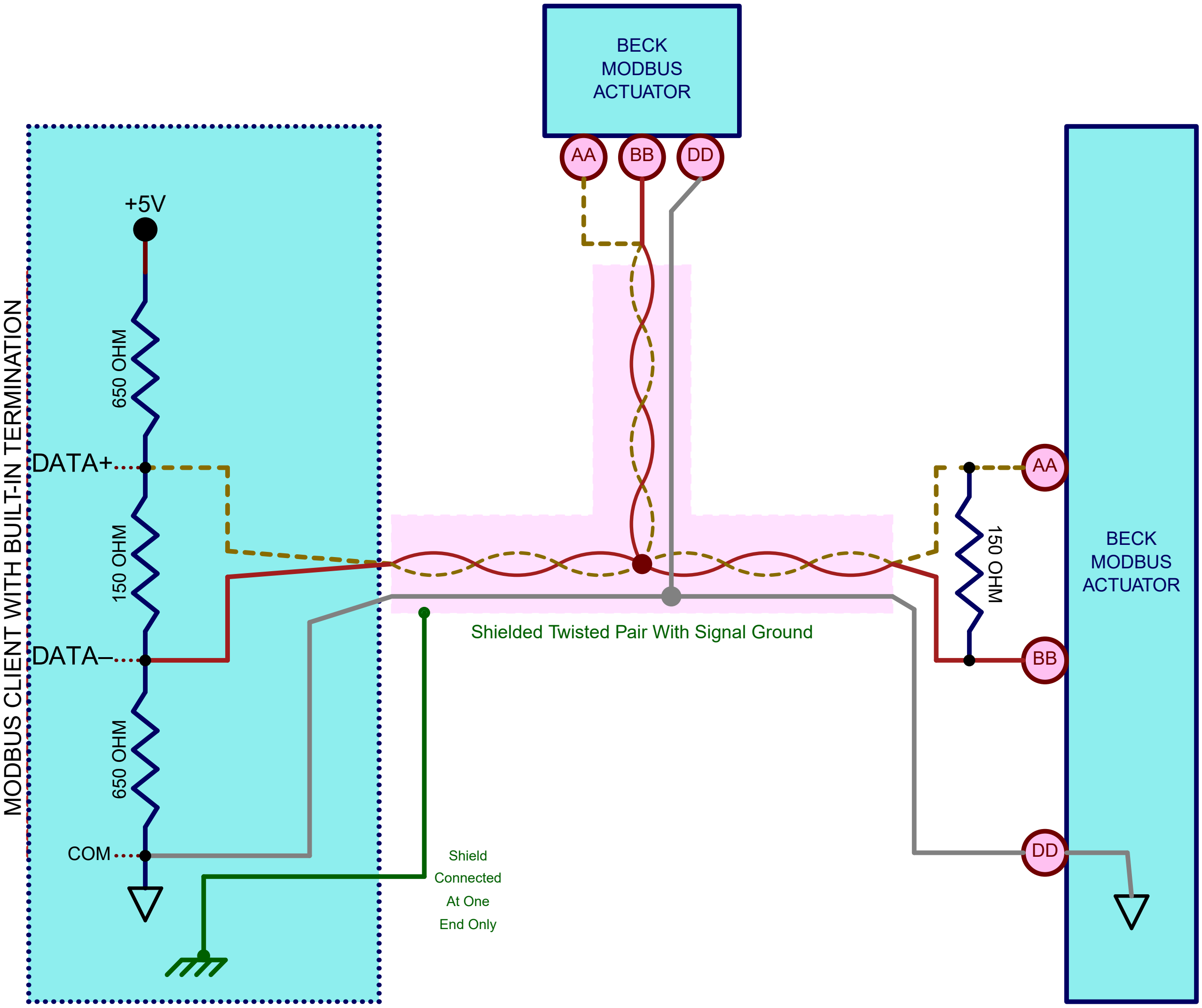
TERMINATION AND LINE POLARIZATION

For error-free data exchange, line termination and line polarization (as specified in the Modbus standard) are required. The rules of termination in the Modbus specification are:

- Line termination shall be placed at the two ends of the bus that are the greatest wire-length distance apart.
- Line polarization shall be provided at only one location on the bus, typically near the Modbus client.

Because Modbus RTU is a multi-drop network, there is no provision for built-in termination or line polarization in the actuator. When needed, termination hardware can be connected at the actuator terminal block.

Because line polarization is typically provided at the Modbus client, Beck Actuators do not provide bias voltage for this function. The diagram below shows a typical termination and line polarization connection for a two actuator network:



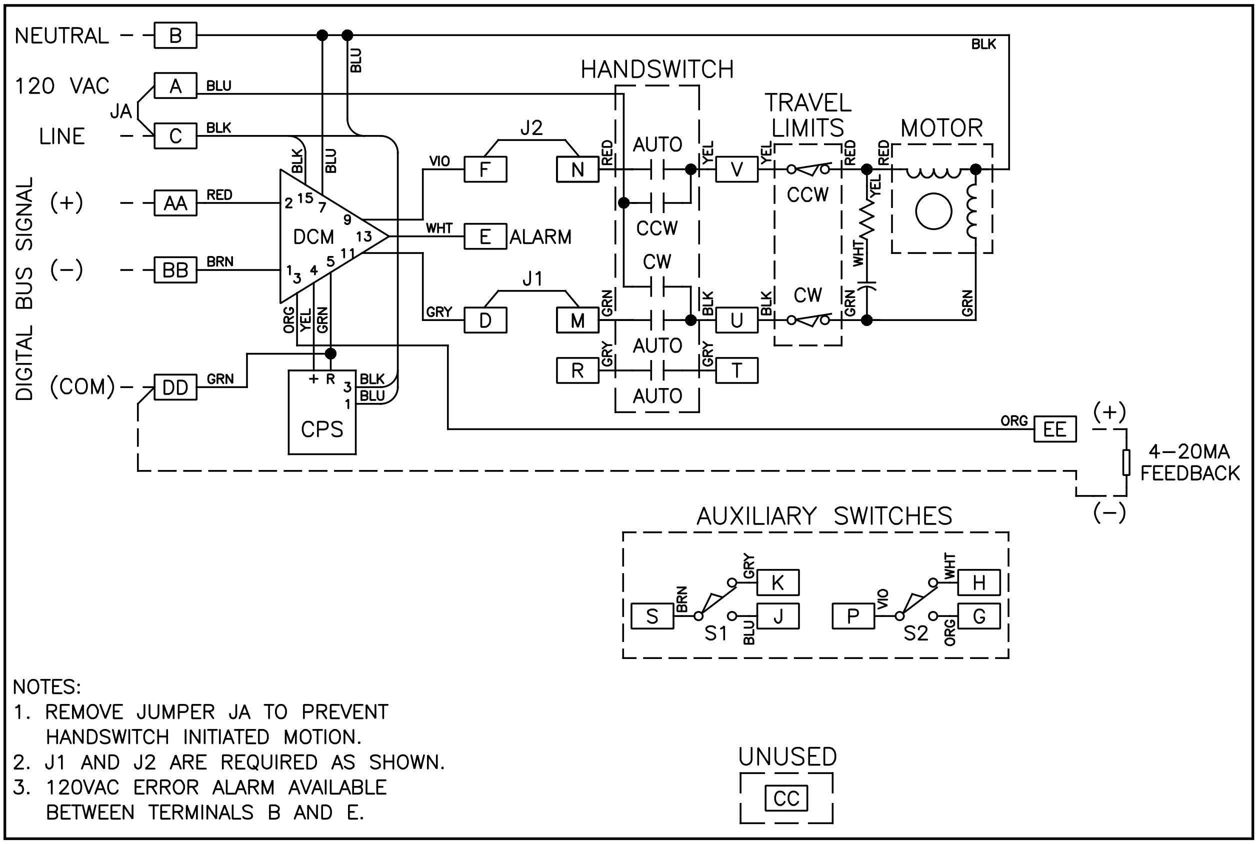
INSTALLATION

For control systems that do not provide for termination or line polarization, a separate device must be added to provide it. One such module is the Loytec LT-B4 (shown below).

Photo courtesy of Loytec

TYPICAL WIRING DIAGRAM

A typical Group 11 product wiring diagram is shown below.



NOTES:
1. REMOVE JUMPER JA TO PREVENT HANDSWITCH INITIATED MOTION.
2. J1 AND J2 ARE REQUIRED AS SHOWN.
3. 120VAC ERROR ALARM AVAILABLE BETWEEN TERMINALS B AND E.

TYPICAL GROUP 11 WIRING DIAGRAM FOR MODBUS RTU

MODBUS INTERFACE SETUP

MODBUS CONFIGURATION INTERFACES

The DCM-3 has three configuration interfaces available for Modbus setup: the Pushbuttons, the Modbus interface itself and the Serial port.

Pushbuttons

Two pushbutton combinations are available to reset the Modbus interface to standard and factory defaults (see the “Modbus Defaults Pushbutton Combinations” section).

Modbus Interface

It is possible to change the Modbus baudrate, parity and polling address via the Modbus interface itself using the following steps:

- If the baud rate, parity or polling address are not known, use the pushbuttons to set the DCM to known Modbus defaults.
- Establish Modbus communications to the DCM.
- Change the polling address to the desired value by writing to Register 40018.
- Reestablish communications at the new polling address.
- Change the parity to the desired value by writing to Register 40017.
- Reestablish communications with the new parity configuration.
- Change the Baudrate to the desired rate by writing to Register 40016.
- Reestablish communications at the new Baudrate.
- Once the basic communications settings have been updated, further modification to the DCM configuration is possible by changing any of the configuration holding registers listed in the “Holding Register Parameter Map”.

Serial Port

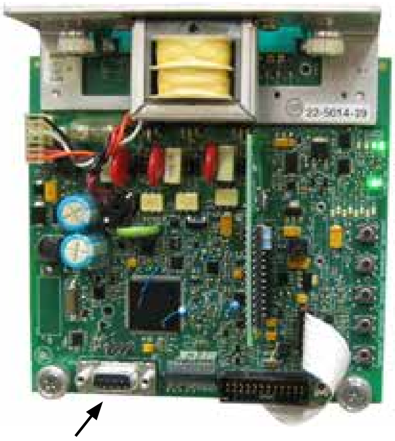
CAUTION

Changes made to the actuator through Serial communications may not necessarily be reflected in asset management systems. Be sure to verify any changes made serially and make manual corrections to the asset management system if necessary.

The Serial port interface is accessible by connecting a computer to the DCM-3 using the DB9 connector.

DB9 to USB Cable

Laptop Computer



DB9 Serial Port

MODBUS INTERFACE SETUP

Serial Port (cont'd)

Serial terminal software is required on the computer to communicate to the DCM-3. This serial terminal software is produced by third-parties who provide installation instructions and in some cases, technical support.

Hyperterminal® Software

One third-party software package is Hyperterminal®. This package was formerly bundled with Microsoft® Windows®, but is presently a separate application available for purchase from Hilgraeve, Inc. (http://www.hilgraeve.com).

Detailed instructions on connecting to the DCM using Hyperterminal® software is available in the full Group 11 Instruction Manual (80-1103-00).

Terra Term Software

A free and open source serial terminal software package called Terra Term has been found to work successfully on Windows 10 Pro Version 1803. This package is available for download at: https://osdn.net/projects/ttssh2/.

Connecting to the DCM with Terra Term is a two step process:

1. After the program is started, select the "Serial" option and choose the serial port corresponding to the DCM connection.
2. From the "SETUP" menu select "Serial Port" and set the speed to 1200, set the data to 8bits, set the parity to none, set the stop bits to 1 and set the flow control to "NONE."

From the main Terra Term window, confirm that the previous two steps were successful by hitting the ENTER key several times and verifying that the DCM responds with a "DCM-3<X.XX>" prompt (where X.XX is the firmware revision of the DCM).

Beck Serial Communications Cables

DB9 to USB-A
P/N 20-0511-60

DB9 to DB9
P/N 20-0511-14

*Note cables may differ in appearance from depiction.

Hyperterminal® is a product of Hilgraeve, Inc.

MODBUS PHYSICAL INTERFACE

Modbus is a serial digital interface that transmits information over a twisted pair of wires plus a common wire. The electrical interface for Modbus is a 5V level differential signal scheme called RS- 485 (or EIA-485). The Modbus terminology for this interface is Modbus “Two-wire” (2W-MODBUS).

BASIC MODBUS COMMUNICATIONS SETUP

Data Rate

The actuator must be set to the same data rate as the control system. Beck Modbus actuators support the following data rates: 300, 600, 1200, 2400, 4800, 9600, 19200 and 38400 bps. There are 2 ways to select the Modbus baudrate:

- Using the Serial command "modbaud" followed by the desired baudrate.
- Using the Modbus baudrate holding register *40016* with the following codes:

Code	Baudrate
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	19200
7	38400

Parity

Parity is used as a means of error checking individual Modbus characters. The actuator must be set to the same parity format as the control system.

EVEN, ODD or NO parity is available. Parity can be set through the “modparity” serial command or the modbus parity register (40017). Both of these options use the following codes:

Code	Parity/Stop Bit
0	Even Parity/One Stop Bit
1	Odd Parity/One Stop Bit
2	NO Parity/Two Stop Bits
3	NO Parity/One Stop Bit

Polling Address

Each actuator on the Modbus network must be assigned a unique polling address. This address is a number that ranges from 1 to 247. There are two ways to configure the polling address:

- Using the Serial command “modpolladdr”.
- Using the Modbus baudrate holding register: *40018*.

Note that address 246 is the default and is normally reserved for commissioning.

ADDITIONAL MODBUS SETUP SERIAL COMMANDS

In addition to the “modbaud”, “modparity” and “modpolladdr” commands described earlier, there are additional commands available to configure other aspects of the Modbus interface. These additional commands are:

MODENDIAN Command

This command sets byte order for modbus floating point and 32 bit register sets. More information is available in the “Modbus Message” section. The selection options are:

Command	Value
modendian 0	Big Endian
modendian 1	Middle-Little Endian
modendian 2	Little Endian
modendian 3	PDP Endian

MODSCALING Command

This command sets the scaling factors for scaled integers types. There are 6 options:

Command	Range	Limits
modscaling 0	0-10000 counts representing 0% to 100%	-500 (-5%) to 10500 (105%)
modscaling 1	0-1000 counts representing 0% to 100%	-50 (-5%) to 1050 (105%)
modscaling 2	0-100 counts representing 0% to 100%	-5 (-5%) to 105 (105%)
modscaling 3	0-255 counts representing 0% to 100%	-12 -4.7%) to 267 (104.7%)
modscaling 4	0-4095 counts representing 0% to 100%	-204 (-4.98%) to 4299 (104.98%)
modscaling 5	0-65535 counts representing 0% to 100%	0 (0%) to 65535 (100% demand/position, 120% torque)

MODCONFIG Command

This command takes two arguments. The command modifies normal DCM Modbus behavior. The first argument controls whether modbus client broadcast messages are ignored. The second argument controls whether the demand status value is forced to “good”. The demand status value is available for control systems that map control system Analog Output Function Blocks to Modbus. The MODCONFIG command fields are:

modconfig [Broadcast Override: 0=Ignore/1=Accept] [Demand Status: 0=Ignore/1=Active]

modconfig 0 0	Broadcast messages are ignored. Demand status ignored.
modconfig 0 1	Broadcast messages are ignored. Demand status value is active.
modconfig 1 0	Broadcast messages are accepted. Demand status ignored.
modconfig 1 1	Broadcast messages are accepted. Demand status value is active.

The default configuration is: **modconfig 1 0.**

Configuration Lockdown Serial Command (Write Protect)

When the DCM is placed in configuration lockdown, all attempts to change Modbus configuration parameters will be rejected. Modbus process parameters will still be writable.

The default state of conguration lockdown/write protection is OFF (i.e., writes are allowed).

Configuration lockdown/write protection can be switched on or off using the “wrprot” serial command:

- **wrprot 0**: writes are allowed
- **wrprot 1**: writes are blocked

Write protection can also be switched off using the eeprom_select parameter at Register 40074.

Since the Modbus protocol does not define a response code to denote write protect status, exception code 4 (server device failure) will be returned when write protect mode is active when an attempt to change a configuration parameter is made.

Network Configuration Serial Command

The Beck DCM has built-in support for multiple communication protocols. In most cases, a DCM will be set at the factory to operate as a Modbus server; however, in some cases when the DCM is setup from scratch, it is necessary to manually turn on the Modbus interface.

The basic serial command for accomplishing this is the “network” command.

- network 4: (Modbus hardware active)

The “network 4” setting is required for normal Modbus operation. There is no Modbus access available for this function.

Modbus Default Settings

Factory and Standard Defaults

When a Modbus actuator ships from the factory, the Modbus interface will be configured to match customer requirements for baudrate, parity and server address. These customer-supplied defaults are stored in the DCM as *factory defaults*.

In the absence of a customer directive, *standard defaults* are stored. Standard Default settings are the same for all DCMs. They are:

- Server Address: The standard default server address is **246**.
- Baud Rate: The standard default baud rate is **19200**.
- Parity: The standard default parity is **EVEN**.

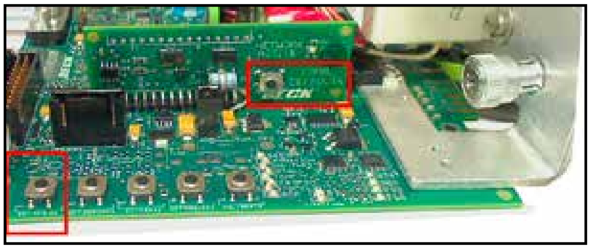
Modbus Defaults Pushbutton Combinations

Because access to the serial command port is not always available, the DCM pushbutton interface has been expanded to support resetting Modbus communication parameters to factory or standard default values.

Since analog demand is disabled when using the Modbus interface, the DCM pushbuttons related to analog demand have been re-purposed for this function.

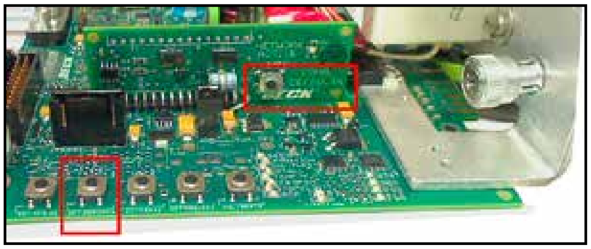
Restore Standard Modbus Defaults Pushbutton Combination: Standard defaults will be restored with the following pushbutton combination:

[COMM. DEFAULTS] + [DEM 0%]



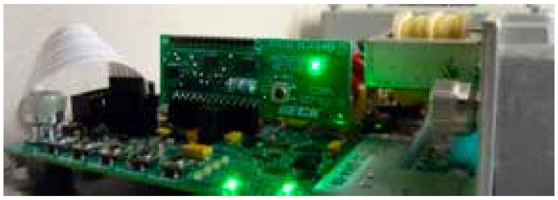
Restore Factory Modbus Defaults Pushbutton Combination Factory defaults will be restored with the following pushbutton combination:

[COMM. DEFAULTS] + [DEM 100%]



NETWORK ACTIVITY LED

The Modbus Network Activity LED is located on the Modbus interface board, as shown below.



The Network Activity LED blinks only when a Modbus message addressed to the DCM with a proper CRC-16 value is received. Because Modbus clients have no specified polling or timeout intervals, the Network Activity LED will continue to blink for 15 seconds after the last valid message has been received.

HANDSWITCH AND LIMIT SWITCH SIGNALS

The Group 11 Handswitch bypasses the DCM to control the actuator directly (see “Typical Wiring Diagram” on Page 4). Because of this, the DCM is not capable of sensing the Handswitch or limit switch status.

Because the Handswitch and limit switch signals are not available, they cannot be transmitted via the provided Modbus Discrete Inputs (10040 to 10045) or Input Register 30044. For Group 11 actuators these Modbus items always read zero. (See Note 1 on page 24).

MODBUS BINARY SIGNAL MAP

The majority of binary parameters are available individually as Discrete Inputs or as part of a bit array within an Input Register. The relationship between these two formats is described in the diagram on the next page. (See page 24 for the availability notes).

Note that binary parameters represented by Discrete Inputs 10001 through 10032 (Input Registers 30041 and 30042) can be masked using the ‘‘alarmoutmask’’ serial command. See the Group 11 manual (80-1103-00) for instructions on using this command.

GROUP 11 MODBUS BINARY SIGNAL MAP

(DCM Rev. 3.24) Rev. 1.2

Alarm Mask Bit	Description	Availability Note	Discrete	Input Register
	Torque/Thrust out of accurate measurement range.	(1)	Discrete 10038	Register 30043-BIT 11
	Ambient Temperature out of accurate measurement range.		Discrete 10037	Register 30043-BIT 10
	Demand out of accurate measurement range.		Discrete 10036	Register 30043-BIT 9
	Position out of accurate measurement range.		Discrete 10035	Register 30043-BIT 8
	Position feedback is overridden with a fixed value.		Discrete 10034	Register 30043-BIT 0
31	Eventlog Memory Failure.		Discrete 10032	Register 30042-BIT 15
30	Excessive Power Supply Ripple Detected.		Discrete 10031	Register 30042-BIT 14
27	Analog Demand has been detected while under PAT Control.		Discrete 10028	Register 30042-BIT 11
26	Analog Demand has been detected while under HART/FF/Profibus/ Modbus Control.		Discrete 10027	Register 30042-BIT 10
25	Invalid Discrete Input (Overrides) signal combination detected.		Discrete 10026	Register 30042-BIT 9
24	Serial Communications Hardware Failure.		Discrete 10025	Register 30042-BIT 8
23	Unused		Discrete 10024	Register 30042-BIT 7
22	Unused		Discrete 10023	Register 30042-BIT 6
21	Unused		Discrete 10022	Register 30042-BIT 5
18	Unused		Discrete 10019	Register 30042-BIT 2
17	Demand signal is over allowable limits.		Discrete 10018	Register 30042-BIT 1
16	Invalid DCM pushbutton combination detected.		Discrete 10017	Register 30042-BIT 0
15	Memory Failure.		Discrete 10016	Register 30041-BIT 15
14	Temperature Sensor Hardware Failure.		Discrete 10015	Register 30041-BIT 14
13	Position Loss of Signal.		Discrete 10014	Register 30041-BIT 13
12	Internal DCM hardware problem (I2C Interface).		Discrete 10013	Register 30041-BIT 12
11	Demand A/D converter problem.		Discrete 10012	Register 30041-BIT 11
10	Position A/D converter problem.		Discrete 10011	Register 30041-BIT 10
9	Torq/Thrust Hardware Failure: Bad sensor signal or A/D converter problem.	(1)	Discrete 10010	Register 30041-BIT 9
8	RTC (timekeeping IC) Failure. Eventlog Information will not be reliable.		Discrete 10009	Register 30041-BIT 8
7	Position Signal Out of Normal Range.		Discrete 10008	Register 30041-BIT 7
6	Temperature Out of Normal Range.		Discrete 10007	Register 30041-BIT 6
5	Feedback Loop is Open.		Discrete 10006	Register 30041-BIT 5
4	G11: Stop/Limit (motor line open or limit switch closed). G22 & G57: DC Volts Low.		Discrete 10005	Register 30041-BIT 4
3	Torq/Thrust Stop.	(1)	Discrete 10004	Register 30041-BIT 3
2	Stall.		Discrete 10003	Register 30041-BIT 2
1	Torq/Thrust High.	(1)	Discrete 10002	Register 30041-BIT 1
0	Demand Loss of Signal.		Discrete 10001	Register 30041-BIT 0

MODSTAT Serial Command

The “MODSTAT” serial command provides a snapshot on what the actuator will send for position and what it received for demand. “MODSTAT” also includes information on the Modbus interface configuration which can be read with a single command.

A typical “MODSTAT” command output is as follows:

```
*** Modbus Reporting for Present Position: [49.93]
Register 30002 [0x4247] (49.933)
Register 30001 [0xBBC1] (49.933)
Register 30008 [0x01f3] (Type 1: 0% = 0 span = 1000)
*** Modbus Reporting for Present Demand: [49.93]
Register 40002 [0x4247] (49.933)
Register 40001 [0xBBC1] (49.933)
Register 40007 [0x01f3] (Type 1: 0% = 0 span = 1000)
Modbus Demand Address: Latest-- | RECEIVED: [ 40007 ] |
Modbus Coils: NONE
Modbus Network Silent Seconds Count: [ 9 ]
Modbus Message Count: [ 6 ]
Modbus CRC Error Count: [ 0 ]
Modbus Exception Count: [ 0 ]
Baudrate: 19200
modparity [0-3]: [Code 0 = 8 Data/EVEN/1 Stop]
Modbus Polling Addr: [ 246 ]
Floating Point Format: [Endian 0: (ABCD translated to ABCD)]
Scaled Integer: Type 1 Configuration [(0%=0 CNTS)(SPAN=1000 CNTS)])
Poll Addr 0 Broadcast Messages are ACCEPTED.
Demand Status Ignored
OK
```

In this example, the actuator reports that there are no Modbus Coils that are active and that it has been 9 seconds since a Modbus message has been processed on the wire (Modbus Network Silent Seconds Count) and that there have been zero CRC and Exception errors.

Demand (Setpoint) Hierarchy

For Group 11 Beck Actuators, there is a fixed hierarchy of sources that determine the actual setpoint controlling the output shaft position. This hierarchy is:

- 1. Handswitch Setting.
- 2. Limit Switch Status.
- 3. Sensor and DCM Diagnostic Status.
- 4. Logic-Level Discrete Inputs.
- 5. Network (Modbus) Setpoint. Since the Modbus control interface has the capability of handling multiple setpoint formats and sources, an additional hierarchy is established to determine what the effective network setpoint will be. This network setpoint hierarchy is:
    - 5.1 The Actuator Stop Override Coil (00001).
    - 5.2 The Actuator 100% Demand Override Coil (00002) or the Actuator 0% Demand Override Coil (00003).
    - 5.3 The latest modulating setpoint in Scaled Integer (demand_scaled at Register 40007 or floating point registers 40001 and 40002).

The following table defines the relationship between the override coils and the setpoint holding registers:

Network Setpoint: Override Coil/Setpoint Holding Register Relationship				
Actuator Stop Override Coil	**Actuator 0% Override Coil**	**Actuator 100% Override Coil**	**Condition**	**Outcome**
1	X	X	Actuator Stop Override Coil is asserted	The actuator ceases all movement. All other network setpoint sources are ignored. The actuator Handswitch and logic level overrides can bypass this coil.
0	1	1	Illegal Override Coil Request	When both the 0% and 100% override coils are asserted, the Stop Override Coil is automatically asserted.
0	1	0	Actuator 0% Override Coil is asserted	The actuator moves to the 0% setpoint. With the exception of the Actuator Stop Override Coil, all other network setpoint sources are ignored. The actuator Handswitch and logic level overrides can bypass this coil.
0	0	1	Actuator 100% Override Coil is asserted	The actuator moves to the 100% setpoint. With the exception of the Actuator Stop Override Coil, all other network setpoint sources are ignored. The actuator Handswitch and logic level overrides can bypass this coil.
0	0	0	All Actuator Overrides are Inactive	The actuator will not move when the overrides become inactive, but will move to the next setpoint holding register value when it is received.

There are 2 methods of determining the active Network Setpoint:

- Modbus Demand Address Readback Registers 30076 and 30077.
- The “MODSTAT” Serial command.

Modbus Demand Address Readback Registers

To determine which network setpoint is controlling the actuator, two input registers are available to determine the active source:

- **Last Setpoint Address Received**: The Modbus Address of the last received setpoint (Input Register 30077).
- **Last Setpoint Address Accepted**: The last accepted setpoint (Input Register 30076).

For either Register (30077 or 30076), the value 0xFFFF (65535) indicates that no applicable demand address has been accessed.

For example, if the the stop coil was previously asserted after a setpoint has been received at Address 40007 these registers will report:

- Register 30076 (last accepted setpoint) will contain the decimal value of 00003 (since the stop coil outranks Register 40007).
- Register 30077 (last received setpoint) will contain a value of 40007 decimal (0x9C47).

Demand Setpoint Reading with “MODSTAT” Serial Command

The “MODSTAT” serial command will display each demand source and which is controlling the actuator. In the previous example, the stop coil has been asserted and has blocked an incoming setpoint from Address 40007.

The active network demand source is described by the the line that begins “Modbus Demand Address.” It shows that latest demand address RECEIVED was 40007 while the present ACCEPTED address is 00003 (the stop coil):

```
------------------------------------------------------------------------------------------------------------------------------------------
*** Modbus Reporting for Present Position: [49.94]
Register 30002 [0x4247] (49.935)
Register 30001 [0xBDE3] (49.935)
Register 30008 [0x01f3] (Type 1: 0% = 0 span = 1000)
*** Modbus Reporting for Present Demand: [49.93]
Register 40002 [0x4247] (49.933)
Register 40001 [0xBBC1] (49.933)
Register 40007 [0x01f3] (Type 1: 0% = 0 span = 1000)
Modbus Demand Address: Latest-- | RECEIVED: [ 40007 ] | ACCEPTED: [ 00003 ] |
Modbus Coils: StopCoil |
Modbus Network Silent Seconds Count: [ 9 ]
Modbus Message Count: [ 22 ]
Modbus CRC Error Count: [ 0 ]
Modbus Exception Count: [ 2 ]
Baudrate: 19200
modparity [0-3]: [Code 0 = 8 Data/EVEN/1 Stop]
Modbus Polling Addr: [ 246 ]
Floating Point Format: [Endian 0: (ABCD translated to ABCD)]
Scaled Integer: Type 1 Configuration [(0%=0 CNTS)(SPAN=1000 CNTS)])
Poll Addr 0 Broadcast Messages are ACCEPTED.
Demand Status Ignored
OK
------------------------------------------------------------------------------------------------------------------------------------------
```

Process Variable Input Register Block

Key process variables in Scaled integer format can be retrieved in a single Function Code 4 transaction by reading input registers 30038 to 30041. This single FC04 command returns:

- 30038: Duplicate scaled position (mirrors register 30008)
- 30039: Duplicate scaled torque (mirrors register 30009)
- 30040: Duplicate scaled demand (mirrors register 40007)
- 30041: Alarm Bytes 1 and 0 (mirrors discrete inputs 10001 to 10016)
- 30042: Alarm Bytes 2 and 3 (mirrors discrete inputs 10017 to 10032)

MODBUS TROUBLESHOOTING

Communications Failure Troubleshooting Checklist

The main symptom of a communications failure is when the Modbus controller is transmitting to the actuator, and the Network Activity LED is not blinking. Here are some suggestions to solve this problem:

- Check the communications link for shorts and opens.
- Verify that the terminal connections are correct: Modbus RTU is polarity sensitive. If the data lines are reversed, communications cannot occur. A signal ground connection is mandatory and must be connected to the proper terminal.
- Verify that line termination at both ends of the network is installed. Termination is installed between the DATA+ and DATA- lines. Paragraph 3.4.5 of the Modbus Serial specification suggests that either a 150 Ohm resistor or a 1nF Capacitor in series with a 120 Ohm resistor are appropriate terminations.
- Two Line polarization resistors should be installed at the Modbus controller. Paragraph 3.4.6 of the Modbus Serial specification recommends a value between 450 and 650 Ohms. A 650 Ohm value supports a greater number of devices on the segment.
- Use an appropriate tool such as an oscilloscope to verify correct signal levels at the actuator terminals. The differential voltage between DATA+ and DATA- should be between 1.5V and 6V. Long cable runs may require the use of a repeater.
- Verify that the Actuator baudrate, parity and polling address settings match the Modbus controller settings.
- Verify the actuator DCM is configured for Modbus using the “network” serial command. The network setting must be four (network 4).
- Verify that the Modbus controller is transmitting frequently enough to the actuator to keep the Network Activity LED blinking. If the controller transmits a Modbus message at a rate of less than every 15 seconds, the Network Activity LED will stop blinking despite the fact that there is nothing wrong with the connection.

Communications Errors Troubleshooting Checklist

Diagnostics built into modern control systems are capable of reporting communications errors. Such errors include:

- ○ Message Retries: More than one transmission of a particular request was required for a successful response from the actuator.
- ○ Message Timeouts: A message request was abandoned because the actuator never responded.
- ○ CRC Errors: A message from the actuator contained a CRC value that did not match the Modbus controller calculated value.

Many of the causes of communications failures can contribute to communications errors, so it is a good idea to verify the items in the previous checklist with the emphasis on confirming proper line termination, polarization, signal levels and link settings. Other possible communications error sources are:

- Verify that there is only one Modbus device for each polling address on each communications link. Actuators configured with duplicate polling addresses will interfere with each other.
- Verify that there are no additional line terminators installed other than the two connected at the far ends of the network. Adding a third line terminator may allow some devices to communicate and block others.
- Line polarization is important for error-free communications. A separately powered RS-485 polarization and termination module may be required for control systems with optically isolated RS-485 ports.
- For shielded cable, verify that the shield is connected on only one end to Earth. Connecting both ends can sometimes result in a circulating ground current that can interfere with communications.
- Verify that the parity setting on the actuator matches the setting on the Modbus controller.
- Long communication links can introduce sporadic errors. A repeater may be required to strengthen signal levels to eliminate the errors.

Communications Errors Troubleshooting Checklist (cont'd)

- Verify that there is only one active Modbus client of a given link at one time. Simultaneous client requests cause errors.
- Use of transmission cables not designed for RS-485 can be the cause of communications errors.
- Communication links that pass near sources of strong electromagnetic waves may be susceptible to communications errors. Relocate them to solve the problem.

Actuator Setpoint Troubleshooting Checklist

Once error-free communications have been established, there can still be issues in controlling the actuator. For example: A setpoint is transmitted to the actuator (e.g. via Register 40007) and the actuator transmits a correct Modbus response message but does move to the new setpoint value. For this case, the following items should be verified:

- Verify that the Handswitch is in AUTO.
- Verify that the digital override signals are inactive.
- Verify that there are no issues with the actuator setup, such as a position sensor error.
- Verify that none of the Modbus coils is active (by checking the last accepted demand address listed with the MODSTAT serial command or Input Register 30076). Note that all incoming setpoints will be ignored until Modbus coils are cleared.
- When using the Demand floating point register set (Registers 40001 and 40002), verify the endian format is correct and that the value is in the correct 0% to 100% range.
- When using the Demand scaled integer register (40007), verify the value being sent matches the expected limits configured in Register 40006 or the modscaling serial command.

Actuator Configuration Troubleshooting Checklist

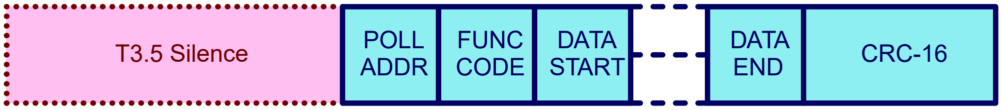
When configuring the actuator via Modbus, the actuator should be in a condition so that any control process will not be upset. If a particular configuration value is rejected by the actuator, the following items should be checked:

- Verify that the actuator is not in write protect mode. Write protect mode can be modified by using the “wrprot” serial command or Holding Register 40074.
- Verify that the configuration value being written is legal. For example, writing a Modbus address of “400” to Holding Register 40018 will be rejected since Modbus addresses must be between 1 and 247.
- In writing floating point values: Make sure that Modbus controller’s endian configuration matches that of the actuator. If the endian configurations do not match, a legal value may be transformed into an illegal one.
- A write to only one of the floating point registers in a two register set will be ignored.
- On heterogeneous device networks, a broadcast message to a different device type may be interpreted by the Beck actuator as a command to change a configuration parameter. Because of this, it is recommended that under these circumstances, the actuator should be configured to ignore broadcast messages.

MODBUS MESSAGE FORMATS

Overview

Beck Modbus actuators support the Modbus RTU message format. A Modbus RTU message consists of the address, function code, data and checksum. There is a response silent interval (t3.5) of approximately 4 character periods before the actuator responds to a message request as required by the Modbus specification.



Polling Address

The polling address is the assigned address that the Beck Actuator will respond to (all other addresses are ignored). The Beck Modbus interface polling address can be configured to a value between 1 and 247. There are 2 ways to configure the polling address:

- Modbus holding register: **40018**.
- Serial command: **modpolladdr**.

Broadcast Messages

Broadcast Messages (Address 0) are messages that are received by all Modbus devices on a network and are acted upon without responding.

In homogeneous networks (for example, a network consisting of only Beck Actuators), a single broadcast message can be used to send a setpoint or to set a coil for all connected devices.

In heterogeneous networks of mixed device types, a broadcast message intended for one group of devices may trigger an unintended function in another device, possibly upsetting a control process. Because of the potential problems of broadcast messages, a serial command to disable processing of these messages has been provided: The “modconfig” serial Command.

Function Codes Supported

Modbus function codes define different message formats. Beck Modbus actuators support the following function codes:

Function Code	Description	Exception Codes
01	Read Multiple Coils (1-2000)	01-04
02	Read Multiple Discrete Inputs (1-2000)	01-04
03	Read Multiple Holding Registers (1-125)	01-04
04	Read Multiple Input Registers (1-125)	01-04
05	Write Single Coil	01-04
06	Write Single Register	01-04
08	Diagnostics	01, 03 or 04
15	Write Multiple Coils (1-1968)	01-04
16	Write Multiple Registers (1-123)	01-04
17	Report Server ID	01 or 04

Function Code 8 Message Format

Function Code 8 is used for testing the Serial Interface and returning diagnostic information. This function also provides for a series of diagnostic counters that track the DCM's view of the quality of the communications link. This function also provides a loopback test that simply echos out incoming data to verify the link quality. All diagnostic counters are unsigned 32 bit numbers.

Hex	Decimal	Name	Description
0x00	00	Return Query Data (Loopback)	Maximum test message length is 32 bytes.
0x01	01	Restart Communications Option	Restart the actuator serial line port and clear all communications event counters.
0x02	02	Return Diagnostic Register	Return the actuator diagnostic register.
0x0A	10	Clear Counters and Diagnostic Register	Clear all counters and the diagnostic register. Counters are also cleared upon power–up.
0x0B	11	Return Bus Message Count	Return the quantity of messages that the remote device has detected on the communications system since its last restart.
0x0C	12	Return Bus Communication Error Count	Return the quantity of CRC errors encountered by the actuator since its last restart, clear counters operation, or power–up.
0x0D	13	Return Bus Exception Error Count	Return the quantity of MODBUS exception responses returned by the actuator since its last restart, clear counters operation, or power–up.

The Diagnostic register has the following format:

Bitmap 1: Function Code 8 Diagnostic Register (R)

MEMORY_FAIL	TEMPERRCODE	POSLOSCODE	IIC_FAIL	DMNDERRCODE	POSERRCODE	TORQERRCODE	RTC_FAIL	DEVICE_MALFUNCTION	CONFIGURATION_CHANGED	COLD_START	MORE_STATUS_AVAILABLE	PV_ANALOG_OUTPUT_FIXED	PV_ANALOG_OUTPUT_SATURATED	NON_PV_OUT_OF_LIMITS	PV_OUT_OF_LIMITS	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Defaults

Register Fields

Bit	Mask	Meaning
MEMORY_FAIL	0x8000	Set if the self test detects a memory error. (Same as Discrete Input 10016)
TEMPERRCODE	0x4000	Error in temperature measurement. (Same as Discrete Input 10015)
POSLOSCODE	0x2000	Error in position sensor measurement. (Same as Discrete Input 10014)
IIC_FAIL	0x1000	Internal IIC interface error. (Same as Discrete Input 10013)
DMNDERRCODE	0x0800	Analog Demand Signal A/D conversion error. (Same as Discrete Input 10012)
POSERRCODE	0x0400	Position Signal A/D conversion error. (Same as Discrete Input 10011)
TORQERRCODE	0x0200	Torque signal input or conversion error. (Same as Discrete Input 10010)
RTC_FAIL	0x0100	Real time clock hardware error. (Same as Discrete Input 10009)
DEVICE_MALFUNCTION	0x0080	Unspecified actuator malfunction.
CONFIGURATION_CHANGED	0x0040	Configuration has changed.
COLD_START	0x0020	DCM has reset/Cold start.
MORE_STATUS_AVAILABLE	0x0010	Additional status is available in Discrete Inputs 10001-10032 or Input Registers 30041 and 30042 (See Modbus Binary Signal Map on Page 11).
PV_ANALOG_OUTPUT_FIXED	0x0008	Analog output feedback current is fixed.
PV_ANALOG_OUTPUT_SATURATED	0x0004	Return for analog output feedback is saturated.
NON_PV_OUT_OF_LIMITS	0x0002	Analog Demand is out of limits.
PV_OUT_OF_LIMITS	0x0001	Position Sensor value is out of limits.

Function Code 17 Message Format

Function Code 17 is used by control systems to identify server devices. Unlike other Modbus RTU messages, the results from this function are ASCII. The format of the response is:

FC17 Byte	Description
0	Server ID: Enumerated value DCM-3 = 0 , DCM-2 (G22) = 1 , Group 11 = 2.
1	Run Indicator Equals 0x00 if both CCW and CW inhibitors are not equal to zero. Equals 0xFF if either CCW or CW inhibitor is zero.
2 to 32	MFG ID, Revision, and Device ID: BECK XX.XX/DM-YY NNNNNN. Padded with spaces.
33 to 54	Actuator Serial Number: GG-MMM-WWWWWW-XX-YY. Padded with spaces.
55 to 65	Position Sensor Serial Number: CPS NNNNNN . Padded with spaces.

Process Variable Data Formats

Scaled Integer Register Set

The Scaled Integer Register Set represents a 0% to 100% of full scale value for the position, setpoint or torque process variables using a 16 bit signed integer. To support the many numeric scaling formats used in industry, direct emulation of the most popular set of typical actuator interfaces have been provided for.

The following input (read only) registers are part of this register set:

- Position (position_scaled at Register 30008)
- Torque (torque_scaled at Register 30009)

The following holding (read/write) registers are part of this set:

- Demand (demand_scaled at Register 40007)
- Scaling Type (scaling_type at Register 40006)

To set the type of scaling, the scaling type parameter (scaling_type) can be set using:

- The “modscaling” Beck command.
- The “scaling_type” (40006) holding register via function code 6 or 16.

See the description of the “modscaling” Beck command for the details on configuring this parameter.

Floating Point Register Set

IEEE-754 is the standard used to define the format of a floating point number. The Beck Modbus actuator provides the following IEEE-754 Floating-Point format register sets:

- Demand (Registers 40001 and 40002)
- Position (Registers 30001 and 30002)
- Torque (Registers 30004 and 30005)

All units are in percent and represent percentage of full scale value. Note floating-point holding registers are grouped as a protected Modbus holding register set.

Floating Point Byte Order

While the IEEE-754 standard defines the floating point numeric format, it does not control how the underlying bytes are stored in a particular system. In computing, the sequence in which bytes are arranged to make up a larger component is called endianness. While Modbus appears to be natively “big endian”, there is no assurance that a control system will assemble the 4 floating point bytes received for each process variable in that fashion. To handle systems which are other than “big endian”, the Beck Modbus interface provides for 4 different endian formats. Those formats are:

MODBUS MESSAGE FORMATS

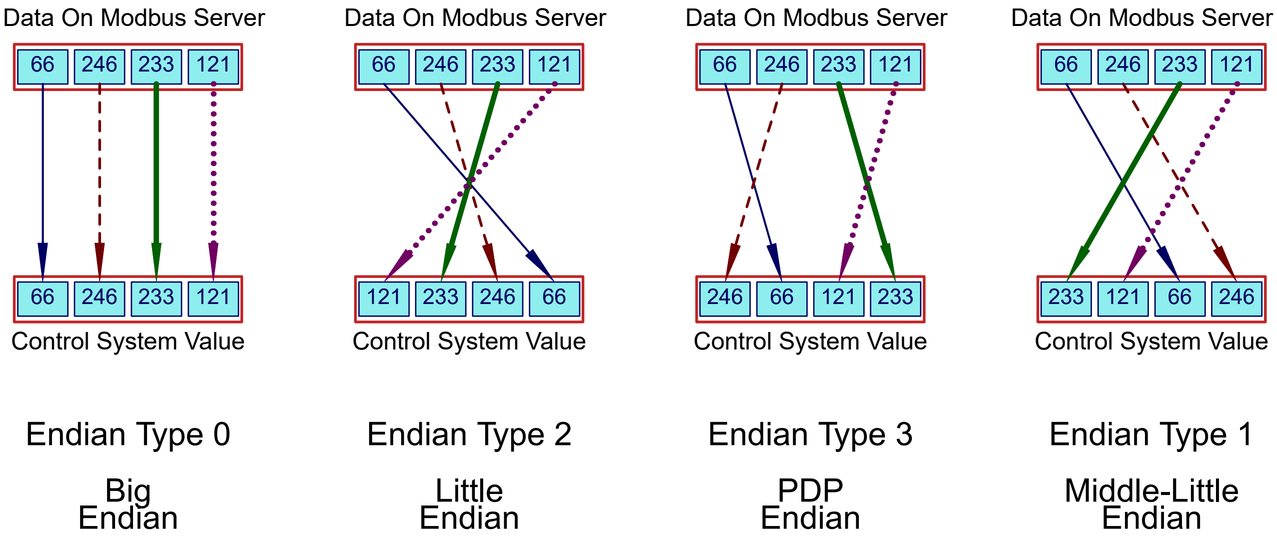
Floating Point Byte Order (cont'd)

- Endian Type 0: Big Endian.
- Endian Type 1: Middle-Little Endian.
- Endian Type 2: Little Endian.
- Endian Type 3: PDP Endian.

In the example below, the floating point number “123.456” is translated into the 4 different endian formats:

Modbus Floating Point Example

Value 123.456 (0x42F6E979 / Bytes 66,246,233 and 121)



Endian Configuration Options

Endianness is configurable using the following options:

- The “modendian” Beck Command.
- The endian_format Modbus holding register at 40004.

Protected Modbus Holding Register Sets

Because the basic Modbus data unit is 16 bits, certain parameters that exceed that size are made up of more than one register. If only one of the registers were updated, it would be possible to corrupt the underlying parameter by mating two unrelated pieces of a larger data item. The Beck Modbus interface provides for protection against this possibility by implementing the following restrictions for multi-register Modbus parameters:

- Contiguous Write Restrictions: All registers in a set must be written in contiguous Modbus transactions in any order.
- Timing Restrictions: All registers in a set must be written within a certain time window or the value will be rejected.

Register sets of this type are called Protected Modbus Holding Register Sets. These sets are shown on the next page.

Group A	40002	demand_value_float_1
	40001	demand_value_float_0
Group B	40020	deadband_value_1_0
	40021	deadband_value_1_1
Group C	40022	deadband_value_2_0
	40023	deadband_value_2_1
Group D	40030	feedback_range_lower_0
	40031	feedback_range_lower_1
Group E	40032	feedback_range_upper_0
	40033	feedback_range_upper_1
Group F	40037	trq_thrust_span_0
	40038	trq_thrust_span_1
Group G	40039	trq_thrust_zero_0
	40040	trq_thrust_zero_1
Group H	40041	trq_thrust_limit_upper_0
	40042	trq_thrust_limit_upper_1
Group I	40045	shutdown_trq_thrust_level_0
	40046	shutdown_trq_thrust_level_1
Group J	40049	drive_stall_time_0
	40050	drive_stall_time_1
Group K	40052	travelspan_value_0
	40053	travelspan_value_1
Group L	40054	maximum_rated_travel_span_engunits_0
	40055	maximum_rated_travel_span_engunits_1
Group M	40057	overtravel_annunciation_level_0
	40058	overtravel_annunciation_level_1
Group N	40062	drive_gear_ratio_0
	40063	drive_gear_ratio_1
Group O	40064	drive_motor_poles_0
	40065	drive_motor_poles_1
Group P	40069	inch_per_rev_0
	40070	inch_per_rev_1

String Register Format

Multiple Modbus registers are also be used to transfer strings. The order of string transfer is independent of the floating point endian format parameter at Register 40004. Strings are mapped to consecutive Modbus registers beginning with the first register in the dataset. The first register contains the first two characters of the string with the high order byte storing the first character and the low order byte storing the second character. The remaining characters are mapped the same way in the register set. Parameters with odd numbers of bytes are padded with spaces.

Ambient Temperature Format

Ambient temperature is represented as two 16 bit registers with signed unscaled representation for Celsius and Fahrenheit. The registers are:

- Celsius: ambient_value_degc at Register 30032.
- Fahrenheit: ambient_value_degf at Register 30033.

MODBUS PARAMETER MAP

The Modbus Parameter Maps that follow describe the Coils, Discrete Inputs, Input Registers and Holding Registers for all Beck products. Where indicated in the "**Note**" column, the following apply:

Parameter Map Notes

Note 1: The indicated parameter may be available if optional hardware is installed and enabled.

Note 2: The Register is a bit array. Not all bits will be active for this model (see the Modbus Binary Signal Map on page 12).

COIL PARAMETER MAP

Coil Parameter Map					
Coil No.	**Hex Address**	**ID**	**Type**	**Description**	**Note***
00001	0x0000	actuator_stop_override_coil_0x01	Process	ACTUATOR_STOP_OVERRIDE_COIL: Setting this coil activates the STOP override coil which causes the actuator to stop at the present and reject all transmitted setpoints. All other setpoint coils (0PCT and 100PCT) will be ignored when this coil is set. All override coils must be cleared to return to automatic control. See “Demand (Setpoint) Hierarchy” Section on Page 14.	
00002	0x0001	actuator_100pct_override_coil_0x02	Process	ACTUATOR_100PCT_OVERRIDE_COIL: Setting this coil activates the 100PCT setpoint override coil. See “Demand (Setpoint) Hierarchy” Section on Page 14.	
00003	0x0002	actuator_0pct_override_coil_0x04	Process	ACTUATOR_0PCT_OVERRIDE_COIL: Setting this coil activates the 0PCT override coil. See “Demand (Setpoint) Hierarchy” Section on Page 14.	

**See above for "Note" column description.*

DISCRETE INPUT PARAMETER MAP

Beck Modbus Basic Parameter Set: Discrete Input Address Map					
DI No.	**Hex Address**	**ID**	**Type**	**Description**	**Note***
10001	0x0000	xmtr_specific_status_0_0x01	Process	ALARM_DEMAND_LOS: The Demand Signal is outside of the intended limits (see Demand Setup menu).	
10002	0x0001	xmtr_specific_status_0_0x02	Process	ALARM_TRQ_THRUST_OVERRANGE: The Torque/Thrust is greater than the output rating.	1
10003	0x0002	xmtr_specific_status_0_0x04	Process	ALARM_STALL: Stall condition has been detected (see **Stall Time** in the **General Setup** menu).	
10004	0x0003	xmtr_specific_status_0_0x08	Process	ALARM_OVERTRQ_THRUST_PROTECTION: Motor power has been removed due to excessive output torque.	1
10005	0x0004	xmtr_specific_status_0_0x10	Process	ALARM STOP LIMIT: Power line to motor appears open.	
10006	0x0005	xmtr_specific_status_0_0x20	Process	ALARM_FEEDBACK_DISCONNECTED: The Feedback Signal is enabled but the loop is open.	
10007	0x0006	xmtr_specific_status_0_0x40	Process	ALARM_TEMPERATURE_OVERRANGE: The temperature is outside of -40F to 185F.	
10008	0x0007	xmtr_specific_status_0_0x80	Process	ALARM_POSITION_OVERRANGE: The Position Signal is less than -5% or greater than 105%.	
10009	0x0008	xmtr_specific_status_1_0x01	Process	ERROR_RTC: Real Time Clock hardware failure.	
10010	0x0009	xmtr_specific_status_1_0x02	Process	ERROR_TRQ_THRUST_SENSING: Torque/Thrust sensing error.	1
10011	0x000A	xmtr_specific_status_1_0x04	Process	ERROR_POSITION_SENSING: Position sensing error.	
10012	0x000B	xmtr_specific_status_1_0x08	Process	ERROR_DEMAND_SENSING: Demand sensing error.	
10013	0x000C	xmtr_specific_status_1_0x10	Process	ERROR_DCM_I2C: I2C Interface has failed.	
10014	0x000D	xmtr_specific_status_1_0x20	Process	ERROR_POSITION_LOS: Position signal in LOS.	
10015	0x000E	xmtr_specific_status_1_0x40	Process	ERROR_TEMPERATURE_AD: Temperature A/D Fail.	

**See above for "Note" column description.*

Beck Modbus Basic Parameter Set: Discrete Input Address Map

DI No.	Hex Address	ID	Type	Description	Note*
10016	0x000F	xmtr_specific_status_1_0x80	Process	ERROR_RAM_ROM: Memory failure.	
10017	0x0010	xmtr_specific_status_2_0x01	Process	ILLEGAL_PB_PRESS: Local control button pressed while locked-out	
10018	0x0011	xmtr_specific_status_2_0x02	Process	DMND_OVER_LIMIT: Demand Signal is out of limit	
10019	0x0012	xmtr_specific_status_2_0x04	Process	UNUSED	
10022	0x0015	xmtr_specific_status_2_0x20	Process	UNUSED	
10023	0x0016	xmtr_specific_status_2_0x40	Process	UNUSED	
10024	0x0017	xmtr_specific_status_2_0x80	Process	UNUSED	
10025	0x0018	xmtr_specific_status_3_0x01	Process	UNDEFINED_3_0: Xmtr Alarm 3 Bit 0	
10026	0x0019	xmtr_specific_status_3_0x02	Process	UNDEFINED_3_1: Xmtr Alarm 3 Bit 1	
10027	0x001A	xmtr_specific_status_3_0x04	Process	DMND_DET_UNDER_DIGITAL: Loop Current Detected while under Modbus/FF Control.	
10028	0x001B	xmtr_specific_status_3_0x08	Process	DMND_DET_UNDER_PAT: Loop Current Detected while set for LOS PAT.	
10031	0x001E	xmtr_specific_status_3_0x40	Process	UNDEFINED_3_6: Xmtr Alarm 3 Bit 6	
10032	0x001F	xmtr_specific_status_3_0x80	Process	UNDEFINED_3_7: Xmtr Alarm 3 Bit 7	
10034	0x0021	analog_output_fixed1_0x01	Process	ERROR_FIXED_FDBK: Feedback is in fixed mode	
10035	0x0022	analog_output_saturated1_0x01	Process	ERROR_POSITION_OOL: Position out of accurate measurement range	
10036	0x0023	analog_output_saturated1_0x02	Process	ERROR_DEMAND_OOL: Demand out of accurate measurement range	
10037	0x0024	analog_output_saturated1_0x04	Process	ERROR_AMBIENT_OOL: Temperature out of accurate measurement range	
10038	0x0025	analog_output_saturated1_0x08	Process	ERROR_TRQ_THRUST_OOL: Torque/Thrust out of accurate measurement range	1
10053	0x0034	led_status_0x0040	Process	POSITION_GREEN_LED: The Output shaft position has reached the limits configured for the Green Indicator.	
10054	0x0035	led_status_0x0080	Process	POSITION_RED_LED: The Output shaft position has reached the limits configured for the Red Indicator.	

**See page 24 for "Note" column description.*

MODBUS PARAMETER MAP

INPUT REGISTER PARAMETER MAP

Input Register Parameter Map

Reg. No.	Hex Address	ID	Type	Description	Note*
30001	0x0000	position_value_float_0	Process	position_value_float: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Position: The present shaft position as measured by the CPS sensor expressed as a percentage. Zero percent position corresponds to the voltage entered in the 'CPS Zero%' value in the position sensor menu. Also see 'Travel'.	
30002	0x0001	position_value_float_1	Process	position_value_float: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Position: The present shaft position as measured by the CPS sensor expressed as a percentage. Zero percent position corresponds to the voltage entered in the 'CPS Zero%' value in the position sensor menu. Also see 'Travel'.	
30003	0x0002	position_value_float_status	Process	Type: ENUMERATED Position Status: The status associated with position_value_float. Used to recreate a standard DCS Floating Point Structure(DS-65). The value will equal either 'good' if there are no parameter-related alarms or 'bad' if any alarm is active.	
30004	0x0003	torque_value_float_0	Process	torque_value_float: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Torque/ Thrust: The load Torque/Thrust measured at the output shaft– expressed as percentage of the maximum rated Torque/Thrust.	1
30005	0x0004	torque_value_float_1	Process	torque_value_float: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Torque/ Thrust: The load Torque/Thrust measured at the output shaft– expressed as percentage of the maximum rated Torque/Thrust.	1
30006	0x0005	torque_value_float_status	Process	Type: ENUMERATED Torque/Thrust Status: The status associated with torque_value_float. Used to recreate a standard DCS Floating Point Structure– DS-65. The value will equal either 'good' if there are no parameter-related alarms or 'bad' if any alarm is active.	1
30008	0x0007	position_scaled	Process	Type: UNSIGNED_INTEGER The output shaft position expressed in percent of span scaled to the range chosen by the 'scaling_type' parameter.	
30009	0x0008	torque_scaled	Process	Type: UNSIGNED_INTEGER Torque/Thrust: The load Torque/Thrust measured at the output shaft-expressed as percentage of the maximum rated Torque/Thrust scaled to the range chosen by the 'scaling_type' parameter.	1
30014	0x000D	beck_drive_designation_readonly_0	Configuration	beck_drive_designation_readonly: Parameter set part 01 of 10 (Byte indexes 01 and 00). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30015	0x000E	beck_drive_designation_readonly_1	Configuration	beck_drive_designation_readonly: Parameter set part 02 of 10 (Byte indexes 03 and 02). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30016	0x000F	beck_drive_designation_readonly_2	Configuration	beck_drive_designation_readonly: Parameter set part 03 of 10 (Byte indexes 05 and 04). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	

**See page 24 for "Note" column description.*

Input Register Parameter Map					
Reg. No.	**Hex Address**	**ID**	**Type**	**Description**	**Note***
30017	0x0010	beck_drive_designation_readonly_3	Configuration	beck_drive_designation_readonly: Parameter set part 04 of 10 (Byte indexes 07 and 06). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30018	0x0011	beck_drive_designation_readonly_4	Configuration	beck_drive_designation_readonly: Parameter set part 05 of 10 (Byte indexes 09 and 08). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30019	0x0012	beck_drive_designation_readonly_5	Configuration	beck_drive_designation_readonly: Parameter set part 06 of 10 (Byte indexes 11 and 10).Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30020	0x0013	beck_drive_designation_readonly_6	Configuration	beck_drive_designation_readonly: Parameter set part 07 of 10 (Byte indexes 13 and 12). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30021	0x0014	beck_drive_designation_readonly_7	Configuration	beck_drive_designation_readonly: Parameter set part 08 of 10 (Byte indexes 15 and 14). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30022	0x0015	beck_drive_designation_readonly_8	Configuration	beck_drive_designation_readonly: Parameter set part 09 of 10 (Byte indexes 17 and 16). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30023	0x0016	beck_drive_designation_readonly_9	Configuration	beck_drive_designation_readonly: Parameter set part 10 of 10 (Byte indexes 19 and 18). Type: ASCII(20) Present value of the Drive Serial Number: Unique serial number associated with the Beck drive. This number is stamped on the nameplate.	
30025	0x0018	beck_software_revision_0	Configuration	beck_software_revision: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT DCM Software Revision: The version of software operating in this Beck circuit board.	
30026	0x0019	beck_software_revision_1	Configuration	beck_software_revision: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT DCM Software Revision: The version of software operating in this Beck circuit board.	
30027	0x001A	device_id_0	Configuration	device_id: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: UNSIGNED_ INTEGER(4) Device Id: Unique serial number associated with the digital control module (DCM). This number is printed on a label affixed to the module CPU.	
30028	0x001B	device_id_1	Configuration	device_id: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: UNSIGNED_ INTEGER(4) Device Id: Unique serial number associated with the digital control module (DCM). This number is printed on a label affixed to the module CPU.	
30032	0x001F	ambient_value_degc	Process	Type: INTEGER Ambient Value Degrees C: The ambient temperature in Celsius measured at the DCM circuit board expressed as a single signed 16 bit integer with no scaling.	

**See page 24 for "Note" column description.*

Input Register Parameter Map					
Reg. No.	**Hex Address**	**ID**	**Type**	**Description**	**Note***
30033	0x0020	ambient_value_degf	Process	Type: INTEGER Ambient Value Degrees F: The ambient temperature in Fahrenheit measured at the DCM circuit board expressed as a single signed 16 bit integer with no scaling.	
30038	0x0025	position_scaled_duplicate	Process	Type: INTEGER A copy of the position_scaled register placed near the status bits. This duplicate allows process variables and status to be read as one multi-register block. The output shaft position expressed in percent of span scaled to the range chosen by the 'scaling_type' parameter.	
30039	0x0026	torque_scaled_duplicate	Process	Type: INTEGER A copy of the position_scaled register placed near the status bits. This duplicate allows process variables and status to be read as one multi-register block. Torque/Thrust: The load Torque/Thrust measured at the output shaft - expressed as percentage of the maximum rated Torque/Thrust scaled to the range chosen by the 'scaling_type' parameter.	1
30040	0x0027	demand_scaled_duplicate	Process	Type: INTEGER A copy of the position_scaled register placed near the status bits. This duplicate allows process variables and status to be read as one multi-register block. The effective demand value. If in 'HOLD' (Modbus) Mode - editing this value will control the motor set point unless an alarm condition exists. It is expressed in percent of span scaled to the range chosen by the 'scaling_type' parameter.	
30041	0x0028	xmtr_specific_status_1_0	Process	Type: UNSIGNED_INTEGER Register representation of discrete inputs xmtr_specific_status_1 and 0.	2
30042	0x0029	xmtr_specific_status_3_2	Process	Type: UNSIGNED_INTEGER Register representation of discrete inputs xmtr_specific_status_3 and 2.	2
30043	0x002A	analog_output_fixed1_saturated1	Process	Type: UNSIGNED_INTEGER Register representation of discrete inputs analog_output_fixed1 and analog_output_saturated1.	2
30045	0x002C	ccw_ext_movement_status_2_1	Process	Type: UNSIGNED_INTEGER Register representation of the counter-clockwise inhibitor bit array. When this value equals zero– the actuator can proceed in the counter-clockwise direction unless otherwise blocked.	
30046	0x002D	cw_ext_movement_status_2_1	Process	Type: UNSIGNED_INTEGER Register representation of the clockwise inhibitor bit array. When this value equals zero– the actuator can proceed in the clockwise direction unless otherwise blocked.	
30057	0x0038	drive_accumulated_runtime_0	Process	drive_accumulated_runtime: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Accumulated Running Time: The total accumulated motor run time since factory shipment.	
30058	0x0039	drive_accumulated_runtime_1	Process	drive_accumulated_runtime: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Accumulated Running Time: The total accumulated motor run time since factory shipment.	
30059	0x003A	drive_overtorques_0	Process	drive_overtorques: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Overthrusts: The total number of times excessive Thrust was detected at the output shaft since factory shipment.	1
30060	0x003B	drive_overtorques_1	Process	drive_overtorques: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Overthrusts: The total number of times excessive Thrust was detected at the output shaft since factory shipment.	1

**See page 24 for "Note" column description.*

Input Register Parameter Map

Reg. No.	Hex Address	ID	Type	Description	Note*
30061	0x003C	drive_peak_torque_0	Process	drive_peak_torque: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Peak Thrust: The maximum thrust sensed at the output shaft since factory shipment.	1
30062	0x003D	drive_peak_torque_1	Process	drive_peak_torque: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Peak Thrust: The maximum thrust sensed at the output shaft since factory shipment.	1
30063	0x003E	drive_reversals_0	Process	drive_reversals: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Reversals: The total number of times the motor started in a direction opposite to the previous start since factory shipment.	
30064	0x003F	drive_reversals_1	Process	drive_reversals: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Reversals: The total number of times the motor started in a direction opposite to the previous start since factory shipment.	
30065	0x0040	drive_run_time_0	Process	drive_run_time: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Last Run Time: The duration of the last motor movement.	
30066	0x0041	drive_run_time_1	Process	drive_run_time: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Last Run Time: The duration of the last motor movement.	
30067	0x0042	drive_stalls_0	Process	drive_stalls: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Stalls: The total number of Stall conditions since factory shipment.	
30068	0x0043	drive_stalls_1	Process	drive_stalls: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Stalls: The total number of Stall conditions since factory shipment.	
30069	0x0044	drive_starts_0	Process	drive_starts: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Starts: The total number motor starts since factory shipment.	
30070	0x0045	drive_starts_1	Process	drive_starts: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Starts: The total number motor starts since factory shipment.	
30071	0x0046	setup_num_overtorques_0	Process	setup_num_overtorques: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT StartUp Over-Thrusts: The number of times the output shaft torque exceeded the Thrust Alarm (’ThrAlrmUpLmt’) threshold during installation.	1
30072	0x0047	setup_num_overtorques_1	Process	setup_num_overtorques: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT StartUp Over-Thrusts: The number of times the output shaft torque exceeded the Thrust Alarm (’ThrAlrmUpLmt’) threshold during installation.	1
30073	0x0048	setup_pk_torque_0	Process	setup_pk_torque: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT StartUp Peak Thrust: The peak thrust recorded during installation.	1
30074	0x0049	setup_pk_torque_1	Process	setup_pk_torque: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT StartUp Peak Thrust: The peak thrust recorded during installation.	1
30076	0x004B	demand_modbus_address_number_last_accepted	Process	Type: UNSIGNED_INTEGER The last ACCEPTED demand (setpoint) Modbus address number stated as an integer value. The number may be that of a coil or holding register. The value 0xFFFF (65535) indicates that no such address has been received.	
30077	0x004C	demand_modbus_address_number_last_rcvd	Process	Type: UNSIGNED_INTEGER The last RECEIVED demand (setpoint) Modbus address number stated as an integer value. The number may be that of a coil or holding register. The value that was transmitted from this Modbus address may have been rejected because it was invalid or the assertion of an actuator override coil (coils 0001 0002 or 0003). The value 0xFFFF (65535) indicates that no such address has been received.	

**See page 24 for "Note" column description.*

MODBUS PARAMETER MAP

Input Register Parameter Map					
Reg. No.	**Hex Address**	**ID**	**Type**	**Description**	**Note***
30079	0x004E	position_raw_value_0	Process	position_raw_value: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Position Sensor Voltage: Displays the present value of the internal Position Sensor voltage– which can be measured at TP4(+) and TP1(-) on the DCM. This value can be edited in the 'Calibration Trim' menu for trimming the position sensor A/D converter.	
30080	0x004F	position_raw_value_1	Process	position_raw_value: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Position Sensor Voltage: Displays the present value of the internal Position Sensor voltage– which can be measured at TP4(+) and TP1(-) on the DCM. This value can be edited in the 'Calibration Trim' menu for trimming the position sensor A/D converter.	
30081	0x0050	trq_thrust_raw_value_0	Process	trq_thrust_raw_value: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Torque/ Thrust TK counts: Displays the present value of the load sensor– TK units represent the internal Torque/Thrust Sensor signal level.	1
30082	0x0051	trq_thrust_raw_value_1	Process	trq_thrust_raw_value: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Torque/ Thrust TK counts: Displays the present value of the load sensor– TK units represent the internal Torque/Thrust Sensor signal level.	1
30083	0x0052	ambient_value_high_0	Process	ambient_value_high: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT	
30084	0x0053	ambient_value_high_1	Process	ambient_value_high: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT	
30085	0x0054	ambient_value_low_0	Process	ambient_value_low: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT	
30086	0x0055	ambient_value_low_1	Process	ambient_value_low: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT	
30088	0x0057	line_frequency	Configuration	Type: INTEGER Line Frequency: Connection to 50 Hz or 60 Hz power mains.	
30094	0x005D	hs_is_logic_level	Configuration	When this register value is greater than zero, the Handswitch works through the DCM and is available via Modbus. When the value is zero, the Handswitch signals are not available and will always read zero.	
30095	0x005E	limits_are_logic_level	Configuration	When this register value is greater than zero, the limit switches work through the DCM and are available via Modbus. When the value is zero, the limit switch signals are not available and will always read zero.	

**See page 24 for "Note" column description.*

HOLDING REGISTER PARAMETER MAP

Holding Register Parameter Map					
Reg. No.	**Hex Address**	**ID**	**Type**	**Description**	**Note***
40001	0x0000	demand_value_float_0	Process	demand_value_float: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Demand Value: Effective Demand signal. If in 'HOLD' (Modbus) Mode– editing this value will control the motor set point unless an alarm condition exists. Valid values are between -5% and 105%.	
40002	0x0001	demand_value_float_1	Process	demand_value_float: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Demand Value: Effective Demand signal. If in 'HOLD' (Modbus) Mode– editing this value will control the motor set point unless an alarm condition exists. Valid values are between -5% and 105%.	
40003	0x0002	demand_value_float_status	Process	Type: INTEGER Demand Status: The status associated with demand_value_float. Used to recreate a standard DCS Floating Point Structure (DS-65). The value will equal either 'good' if there are no parameter-related alarms or 'bad' if any alarm is active.	
40004	0x0003	endian_format	Configuration	Type: ENUMERATED The byte order for all dual Modbus registers representing a 32 byte parameter. This is an enumerated value with a valid range of 0-3. Big Endian=0| Middle-Little Endian=1|Little Endian= 2|PDP Endian=3	
40006	0x0005	scaling_type	Configuration	Type: ENUMERATED The type of scaling applied to the scaled integer register set. The scaling range is an enumerated value. The default value is 0.	
40007	0x0006	demand_scaled	Process	Type: INTEGER The effective demand value. If in 'HOLD' (Modbus) Mode- editing this value will control the motor set point unless an alarm condition exists. It is expressed in percent of span scaled to the range chosen by the 'scaling_type' parameter.	
40013	0x000C	drive_direction	Configuration	Type: ENUMERATED Drive Direction: Defines the direction of output shaft rotation for an increasing Demand signal. Rotation direction is viewed facing the exposed output shaft end.	
40014	0x000D	drive_operating_mode	Process	Type: ENUMERATED Operating Mode: Defines the fundamental manual or automatic Position control mode. 'Follow' mode will cause positioning according to the analog Demand signal. 'Hold' mode will cause positioning according to the HART/FF Interface Demand Value. 'Stay' mode will cause the output to remain stationary and to maintain the present position. In 'Stay' mode– the motor handwheel can NOT be freely turned. 'Stop' mode removes power from the motor. In 'Stop' mode– the motor handwheel can be freely turned. 'Run' modes will cause the shaft to move to the end of travel. All operating modes can be overridden by the control drive Handswitch.	

***See page 24 for "Note" column description.**

MODBUS PARAMETER MAP

Holding Register Parameter Map					
Reg. No.	Hex Address	ID	Type	Description	Note*
40016	0x000F	modbus_baudrate	Configuration	Type: ENUMERATED The baudrate selector for the Modbus interface. Selection values are: 0=300... 1=600... 2=1200... 3=2400... 4=4800... 5=9600... 6=19200... 7=38400.	
40017	0x0010	modbus_parity	Configuration	Type: ENUMERATED The parity selector for the Modbus interface. Selection values are: 0=EVEN/ONE STOP BIT ... 1=ODD/ONE STOP BIT ... 2=NONE/TWO STOP BITS ... 3=NONE/ONE STOP BIT.	
40018	0x0011	modbus_address	Configuration	Type: INTEGER The active address of the Modbus interface. Valid values are 1-247. The default value is 246.	
40020	0x0013	deadband_value_1_0	Configuration	deadband_value_1: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Step Size: The typical change in Demand signal– expressed in percent of span– that can occur before the output shaft will reposition.	
40021	0x0014	deadband_value_1_1	Configuration	deadband_value_1: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Step Size: The typical change in Demand signal– expressed in percent of span– that can occur before the output shaft will reposition.	
40022	0x0015	deadband_value_2_0	Configuration	deadband_value_2: Parameter set part 01 of 02 (Byte indexes 01 and 00). Type: FLOAT Max Error: The maximum difference– expressed in percent of span– that the automatic control will allow before the output shaft will reposition.	
40023	0x0016	deadband_value_2_1	Configuration	deadband_value_2: Parameter set part 02 of 02 (Byte indexes 03 and 02). Type: FLOAT Max Error: The maximum difference– expressed in percent of span– that the automatic control will allow before the output shaft will reposition.	
40025	0x0018	utility_command_register	Process	Type: ENUMERATED Used to initiate system utilities based on the value in the register. Utility command codes are: 21930 which unstalls the Actuator and 44236 which is used to test communications by blinking the Acknowledge LED on the actuator DCM. During a write operation the register will respond with the value transmitted. When read the register will always equal zero.	
40027	0x001A	pot_power_installed	Configuration	Type: ENUMERATED Potentiometer Power: Defines whether the optional Potentiometer Power Supply function is installed in the DCM.	
40028	0x001B	smart_feedback_installed	Configuration	Type: ENUMERATED Feedback Output: Defines whether the optional Feedback Output function is installed in the DCM.	
40029	0x001C	io_select	Configuration	Type: ENUMERATED Feedback Select: Defines whether the DCM I/O terminal is used to supply a Feedback output signal– to supply power to a Position Sensor potentiometer– or is unused.	

**See page 24 for "Note" column description.*

Holding Register Parameter Map					
Reg. No.	**Hex Address**	**ID**	**Type**	**Description**	**Note***
40030	0x001D	feedback_range_lower_0	Configuration	feedback_range_lower: Parameter set part 01 of 02 (Byte indexes 01 and 00). Type: FLOAT Feedback Range Lower: The value of the feedback signal– in mA– that corresponds to a 0% output shaft position. Values can range between 3.0 mA and 16.0mA	
40031	0x001E	feedback_range_lower_1	Configuration	feedback_range_lower: Parameter set part 02 of 02 (Byte indexes 03 and 02). Type: FLOAT Feedback Range Lower: The value of the feedback signal– in mA– that corresponds to a 0% output shaft position. Values can range between 3.0 mA and 16.0mA	
40032	0x001F	feedback_range_upper_0	Configuration	feedback_range_upper: Parameter set part 01 of 02 (Byte indexes 01 and 00). Type: FLOAT Feedback Range Upper: The value of the feedback signal– in mA– that corresponds to a 100% output shaft position. Values can range between 7.0mA and 21.0mA	
40033	0x0020	feedback_range_upper_1	Configuration	feedback_range_upper: Parameter set part 02 of 02 (Byte indexes 03 and 02). Type: FLOAT Feedback Range Upper: The value of the feedback signal– in mA– that corresponds to a 100% output shaft position. Values can range between 7.0mA and 21.0mA	
40035	0x0022	trq_thrust_sensor_installed	Configuration	Type: ENUMERATED Thrust Sensor: Defines whether the optional Thrust Sensing function is installed in the DCM.	1
40036	0x0023	trq_thrust_sensor_enabled	Configuration	Type: ENUMERATED Torque/Thrust Display: Defines whether the optional Torque/Thrust Sensing function is enabled in the DCM. If the actuator does not have a torque/thrust sensor– this feature must be disabled.	1
40037	0x0024	trq_thrust_span_0	Configuration	trq_thrust_span: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Torque/Thrust Constant: The value of the load sensor span that corresponds to 100% span. TK units represent the internal Torque/Thrust Sensor signal level.	1
40038	0x0025	trq_thrust_span_1	Configuration	trq_thrust_span: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Torque/Thrust Constant: The value of the load sensor span that corresponds to 100% span. TK units represent the internal Torque/Thrust Sensor signal level.	1
40039	0x0026	trq_thrust_zero_0	Configuration	trq_thrust_zero: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Torque/Thrust Null: The value of the load sensor that corresponds to 0%. TK units represent the internal Torque/Thrust Sensor signal level.	1

**See page 24 for "Note" column description.*

MODBUS PARAMETER MAP

Holding Register Parameter Map					
Reg. No.	Hex Address	ID	Type	Description	Note*
40040	0x0027	trq_thrust_zero_1	Configuration	trq_thrust_zero: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Torque/Thrust Null: The value of the load sensor that corresponds to 0%. TK units represent the internal Torque/Thrust Sensor signal level.	1
40041	0x0028	trq_thrust_limit_upper_0	Configuration	trq_thrust_limit_upper: Parameter set part 01 of 02 (Byte indexes 01 and 00). Type: FLOAT Torque/Thrust Alarm Level: Torque/Thrust values greater than this will generate an alarm. This alarm function does not cause the motor to stop. Also see 'ShutDwnTrq/ShutDwnThr'.	1
40042	0x0029	trq_thrust_limit_upper_1	Configuration	trq_thrust_limit_upper: Parameter set part 02 of 02 (Byte indexes 03 and 02). Type: FLOAT Torque/Thrust Alarm Level: Torque/Thrust values greater than this will generate an alarm. This alarm function does not cause the motor to stop. Also see 'ShutDwnTrq/ShutDwnThr'.	1
40044	0x002B	overtrq_thrust_protection	Configuration	Type: ENUMERATED Over-thrust Protection: Defines whether detection of excessive thrust will remove power from the motor.	1
40045	0x002C	shutdown_trq_thrust_level_0	Configuration	shutdown_trq_thrust_level: Parameter set part 01 of 02 (Byte indexes 01 and 00). Type: FLOAT Shutdown Thrust Level: Power will be removed if thrust is greater than this level. This will cause a shutdown alarm. Set this value greater than 'AlarmLevel' to have an alarm before shutdown. Also see 'Ovt Prot'.	1
40046	0x002D	shutdown_trq_thrust_level_1	Configuration	shutdown_trq_thrust_level: Parameter set part 02 of 02 (Byte indexes 03 and 02). Type: FLOAT Shutdown Thrust Level: Power will be removed if thrust is greater than this level. This will cause a shutdown alarm. Set this value greater than 'AlarmLevel' to have an alarm before shutdown. Also see 'Ovt Prot'.	1
40048	0x002F	stall_protection	Configuration	Type: ENUMERATED(1) stall_protection: Removes motor power if stall time is reached. Also see 'Stall Time'.	
40049	0x0030	drive_stall_time_0	Configuration	drive_stall_time: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Stall Time: The maximum allowable time for the motor to run before the desired position is reached. After this time the Stall condition is set. If stall protection is enabled– power will also be removed from the motor. This value should be set slightly longer than the end to end travel timing. Also see 'StallProt'.	

**See page 24 for "Note" column description.*

Holding Register Parameter Map					
Reg. No.	**Hex Address**	**ID**	**Type**	**Description**	**Note***
40050	0x0031	drive_stall_time_1	Configuration	drive_stall_time: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Stall Time: The maximum allowable time for the motor to run before the desired position is reached. After this time the Stall condition is set. If stall protection is enabled– power will also be removed from the motor. This value should be set slightly longer than the end to end travel timing. Also see ’StallProt’.	
40052	0x0033	travelspan_value_0	Configuration	travelspan_value: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT Travel: the number of degrees/inches/mm of output shaft travel for a 100% span. Edit this value to use a stroke shorter than the allowable ’MaxTravel’. Values higher than ’MaxTravel’ are rejected. If values less than ’MaxTravel’ are rejected– ’CPS Zero%’ may have to be adjusted.	
40053	0x0034	travelspan_value_1	Configuration	travelspan_value: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT Travel: the number of degrees/inches/mm of output shaft travel for a 100% span. Edit this value to use a stroke shorter than the allowable ’MaxTravel’. Values higher than ’MaxTravel’ are rejected. If values less than ’MaxTravel’ are rejected– ’CPS Zero%’ may have to be adjusted.	
40054	0x0035	maximum_rated_travel_span_engunits_0	Configuration	maximum_rated_travel_span_engunits: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT The maximum available travel distance of the output shaft. This value is entered manually– and must correspond to the actuator design. If a shortened stroke is desired– edit ’Travel’.	
40055	0x0036	maximum_rated_travel_span_engunits_1	Configuration	maximum_rated_travel_span_engunits: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT The maximum available travel distance of the output shaft. This value is entered manually– and must correspond to the actuator design. If a shortened stroke is desired– edit ’Travel’.	
40056	0x0037	overtravel_annunciation	Configuration	Type: ENUMERATED Limit Switch Indication: Defines whether contacting a limit switch– outside the normal travel range of 0% to 100%– will cause an error condition.	
40057	0x0038	overtravel_annunciation_level_0	Configuration	overtravel_annunciation_level: Parameter set part 01 of 02 (Byte indexes 01 and 00). Type: FLOAT overtravel_annunciation_level is used in conjunction with overtravel_annunciation to set a zone at the ends of travel where STOP/LIMIT and overtorque alarms will be supressed.	

**See page 24 for "Note" column description.*

MODBUS PARAMETER MAP

Holding Register Parameter Map					
Reg. No.	Hex Address	ID	Type	Description	Note*
40058	0x0039	overtravel_annunciation_level_1	Configuration	overtravel_annunciation_level: Parameter set part 02 of 02 (Byte indexes 03 and 02). Type: FLOAT overtravel_annunciation_level is used in conjunction with overtravel_annunciation to set a zone at the ends of travel where STOP/LIMIT and overtorque alarms will be supressed.	
40062	0x003D	drive_gear_ratio_0	Configuration	drive_gear_ratio: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT The gear ratio from the motor to the output shaft.	
40063	0x003E	drive_gear_ratio_1	Configuration	drive_gear_ratio: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT The gear ratio from the motor to the output shaft.	
40064	0x003F	drive_motor_poles_0	Configuration	drive_motor_poles: Parameter set part 01 of 02 (Byte indexes 01 and 00).Type: FLOAT The number of magnetic poles on the rotor.	
40065	0x0040	drive_motor_poles_1	Configuration	drive_motor_poles: Parameter set part 02 of 02 (Byte indexes 03 and 02).Type: FLOAT The number of magnetic poles on the rotor.	
40074	0x0049	eeprom_select	Configuration	Type: ENUMERATED The EEPROM memory selector utility function. eeprom_select=249 will reset the DCM. eeprom_select=252 will recall the default parameter values for the model. eeprom_select=253 will save the present parameter values to factory defaults. eeprom_select=509 will restore the factory defaults. All other values are ignored.	

**See page 24 for "Note" column description.*

MODBUS TCP CONFIGURATION

Overview

The Modbus TCP Ethernet Communications Module provides a bridge between modern Ethernet-based control systems and legacy Modbus RTU (RS-485) Beck actuators. The module consists of a stainless steel conduit with the conversion electronics sealed inside. The core conversion module is powered by the Lantronix xPort Modbus TCP-RTU gateway.

BECK MODBUS TCP ADAPTER

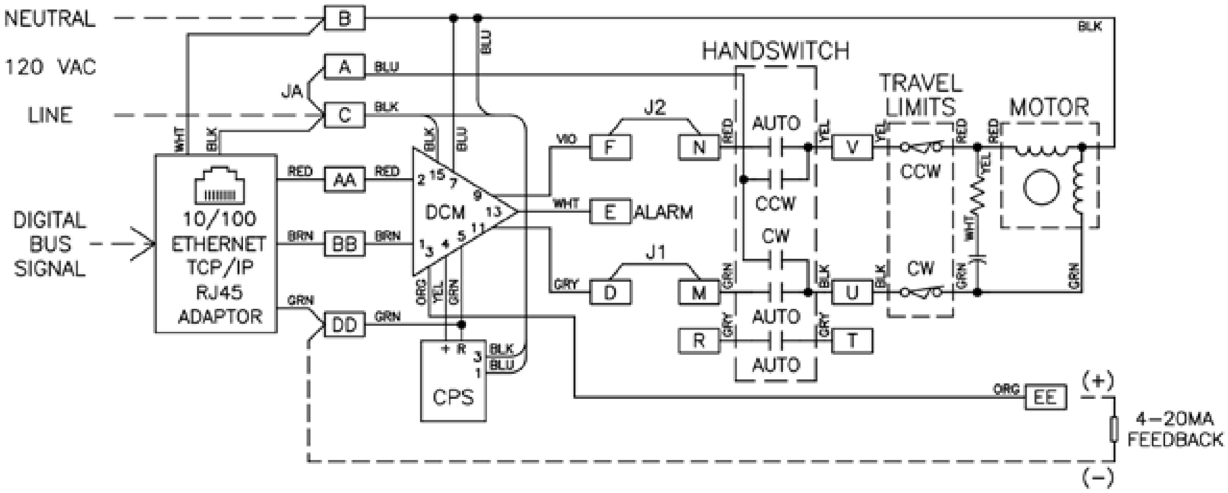
INSTALLED BECK MODBUS TCP ADAPTER

Connections

One end of the module has wires that are designed to be connected to a Beck Actuator terminal block. This end is screwed into the signal connection conduit entrance. The function of the wires is to provide power to the module and to connect to the actuator Modbus RTU interface.

The wires are connected to the following terminals:

COLOR	SIGNAL	TERMINAL
BLACK	AC LINE	C
WHITE	AC NEUTRAL	B
RED	MODBUS RTU DATA_P	AA
BROWN	MODBUS RTU DATA_N	BB
GREEN	MODBUS RTU COMMON	DD



TYPICAL GROUP 11 WIRING DIAGRAM FOR MODBUS TCP

The other end of the module that faces away from the actuator has a standard RJ-45 Ethernet connector for the control network connection:

RJ-45 ETHERNET CONNECTOR

MODBUS TCP CONFIGURATION

Modbus TCP Ethernet Communications Module Setup Overview

The Modbus TCP Ethernet Communications Module must be configured so that the Modbus RTU side of the module matches the actuator configuration.

Standard Configuration Defaults

The following table lists standard configuration defaults for Beck Lantronix Modbus-TCP xPort-05 based products:

Option	Default Value
IP Address	STATIC IP: 192.168.100.64
Default Gateway	STATIC IP: 192.168.100.1
Netmask	255.255.255.0
Modbus RTU Serial Interface	19200,8,E,1,RS485
Security Option	Unsecured/no Passwords

Modbus RTU Polling Address

The Modbus TCP Ethernet Module default configuration is a pass-through gateway which transmits the underlying Modbus message without modification. Consequently, the polling address field in the message must match that of the Group 11 actuator.

To simplify control system programming, it is recommended that the default Modbus RTU polling address of 246 be used in all Modbus TCP applications so that the same underlying messages can be used for all actuators in the network.

Mandatory Configuration Settings

Each Modbus TCP Module is set at the factory to certain mandatory configuration settings with no changes required by the user.

The mandatory settings required for proper operation are:

- The serial interface *MUST* be set for "**RS485**".
- The "Protocol" *MUST* be configured as "**Modbus/RTU, Server(s) attached**".
- Pin CP1 *MUST* be configured as "**RS_485 Output Enable, non-inverted**".

Network Security

⚠ **WARNING**

This product is intended only for use on properly secured and firewall-protected industrial control system networks.

The Beck Modbus TCP module is based on the Lantronix xPort device. Lantronix recommended best practices for secure operation of this device should be reviewed prior to installation: https://cdn.lantronix.com/wp-content/uploads/pdf/Best-Practices-for-IoT-Device-Security.pdf.

The latest Lantronix security notices are available at: https://www.lantronix.com/support/security-updates/.

Configuration Password

According to the **xPort User Guide**, Lantronix recommends that the user should always enable the enhanced password setting and create a strong, sixteen character password.

Some jurisdictions may require that a random password be set prior to actuator shipment. In this case, a printed copy of the password certificate will be shipped with the actuator.

Lantronix xPort Information

Additional documentation on the Lantronix xPort is available at: http://www.lantronix.com.

The xPort Modbus Protocol User Guide is available at: http://www.lantronix.com/wp-content/uploads/pdf/Modbus_Protocol_UG_Rev_J.pdf.

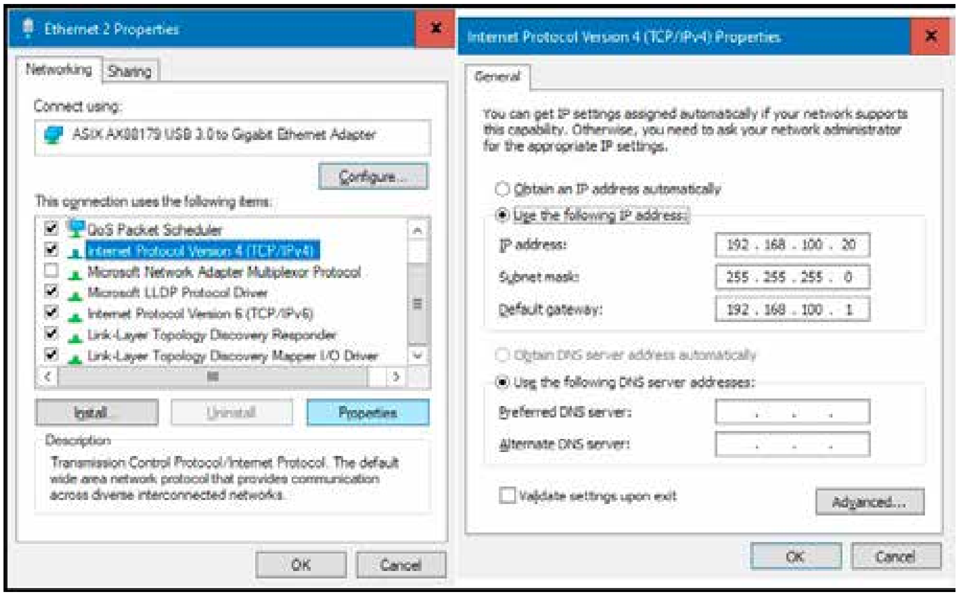
The xPort Hardware User guide is available at: https://cdn.lantronix.com/wp-content/uploads/pdf/XPort_UG.pdf.

Modbus TCP Module Configuration Tools

The Lantronix Modbus-TCP xPort-05 is configured using free software and inexpensive hardware.

Hardware: Ethernet Communications Adapter

Any standard USB-to-Ethernet adapter can be used to interface to the Lantronix Modbus-TCP xPort-05. Windows device manager should set the interface to an address in the 192.168.100.XX/24 Subnet (NOTE: Address 192.168.100.64 must not be used as the test Ethernet port IP address as this is the default Modbus TCP Adapter device address).



Connect using:

ASIX AX88179 USB 3.0 to Gigabit Ethernet Adapter

This connection uses the following items:

- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- □ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Link-Layer Topology Discovery Responder
- ☑ Link-Layer Topology Discovery Mapper I/O Driver

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 100 . 20

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 100 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Software: Configuration Tools

There are two configuration tools available, both of which are free:

- **Lantronix DeviceInstaller Software**:

  This is a free windows application from Lantronix which is available at:

  https://www.lantronix.com/products/deviceinstaller/

- **Telnet Command**:

  The Telnet command is a standard command line utility provided by all operating systems.

* *Note that the user will need sufficient privileges to configure the USB-to-Ethernet adapter to settings that agree with the Device Under Test.*

BECK®

HAROLD BECK & SONS, INC.

Made in USA

11 TERRY DRIVE • NEWTOWN, PENNSYLVANIA 18940 • USA
PHONE: 215-968-4600 • FAX: 215-860-6383 • E-MAIL: sales@haroldbeck.com
www.haroldbeck.com

Rev. 01.9
3/23