

Object ID	Object name	Object name / description
3711	Demand Method - Current	Current demand method: 0 = Thermal demand 1 = Timed interval sliding block 2 = Timed interval block 3 = Timed interval rolling block 4 = Input synchronized block 5 = Input synchronized rolling block 6 = Command synchronized block 7 = Command synchronized rolling block 8 = Clock synchronized block 9 = Clock synchronized rolling block
3721	Demand Method - Input Metering	Input metering demand method: 0 = Thermal demand 1 = Timed interval sliding block 2 = Timed interval block 3 = Timed interval rolling block 4 = Input synchronized block 5 = Input synchronized rolling block 6 = Command synchronized block 7 = Command synchronized rolling block 8 = Clock synchronized block 9 = Clock synchronized rolling block

### I/O multi-state input objects

The following table lists the Multi-state Input objects that provide information about meter's I/O configuration.

Object ID	Object name	Description
7274, 7298, 7322, 7346	Digital Input 1 Mode Digital Input 2 Mode Digital Input 3 Mode Digital Input 4 Mode	Digital Input 1 to 4 control mode 0 = Normal (Alarm) 1 = Demand Interval Sync Pulse 2 = Multi-tariff Control 3 = Input Metering
9673, 9681	Digital Output Mode 1 Digital Output Mode 2	Digital output 1 and 2 control mode 0 = External 1 = Demand Sync 2 = Alarm 3 = Energy

## EtherNet/IP

EtherNet/IP is the name given to the Common Industrial Protocol (CIP), as implemented over standard Ethernet (IEEE 802.3 and the TCP/IP protocol suite).

### EtherNet/IP features overview

The EtherNet/IP and related features are available in firmware version 2.5.2 and above for PM5560, PM5563 and PM5563RD meter models, and firmware version 10.6.3 and above for PM5561 meter model.

The CIP application layer defines a set of application objects and device profiles that define common interfaces and behaviors. In addition, CIP communication services enable end-to-end communication between devices on the different CIP networks.

EtherNet/IP maps the CIP communication services to Ethernet and TCP/IP, enabling multi-vendor interoperability between devices on Ethernet as well as with the other CIP networks.

EtherNet/IP defines two primary types of communications:

- Cyclical Exchanges (Implicit Exchanges)
- Messaging (Explicit Exchanges)

## Cyclical Exchanges (Implicit Exchanges)

The tables below gives description of the assembly sets supported by PM55xx meters.

### Assembly input 100

Word Number	Parameter
0	Current A
2	Current B
4	Current C
6	Current N
8	Current G
10	Current Avg
12	Current Unbalance A
14	Current Unbalance B
16	Current Unbalance C
18	Current Unbalance Worst
20	Voltage A-B
22	Voltage B-C
24	Voltage C-A
26	Voltage L-L Avg
28	Voltage A-N
30	Voltage B-N
32	Voltage C-N
34	–
36	Voltage L-N Avg
38	Voltage Unbalance A-B
40	Voltage Unbalance B-C
42	Voltage Unbalance C-A
44	Voltage Unbalance L-L Worst
46	Voltage Unbalance A-N
48	Voltage Unbalance B-N
50	Voltage Unbalance C-N
52	Voltage Unbalance L-N Worst
54	Active Power A
56	Active Power B
58	Active Power C
60	Active Power Total
62	Reactive Power A
64	Reactive Power B
66	Reactive Power C

Word Number	Parameter
68	Reactive Power Total
70	Apparent Power A
72	Apparent Power B
74	Apparent Power C
76	Apparent Power Total
78	Power Factor A
80	Power Factor B
82	Power Factor C
84	Power Factor Total
86	Displacement Power Factor A
88	Displacement Power Factor B
90	Displacement Power Factor C
92	Displacement Power Factor Total
94	Frequency
96	Active Energy Delivered (Into Load)
98	Active Energy Received (Out of Load)
100	Active Energy Delivered + Received
102	Active Energy Delivered- Received
104	Reactive Energy Delivered
106	Reactive Energy Received
108	Reactive Energy Delivered + Received
110	Reactive Energy Delivered - Received
112	Apparent Energy Delivered
114	Apparent Energy Received
116	Apparent Energy Delivered + Received
118	Apparent Energy Delivered - Received

### Assembly output 150

Word Number	Parameter
0	Dummy parameter
1	Dummy parameter

## Messaging (Explicit Exchanges)

The following objects can be accessed through explicit exchanges by PM55XX meters.

### Object classes

The object classes are detailed in the following table:

Object class	Class ID	No. of instances	Description
Identity	01 hex	1	Supports the reset service
Message router	02 hex	1	Explicit message connection
Assembly	04 hex	2	Defines I/O data format
Connection manager	05 hex	1	Manages the internal resources associated with both

Object class	Class ID	No. of instances	Description
			I/O and explicit messaging conditions
TCP/IP interface	F5 hex	1	TCP/IP configuration
Ethernet link	F6 hex	1	Counter and status information
Port object	F4 hex	1	Describes the communication interfaces that are present on the meter and visible to CIP
Base energy	4E hex	1	Acts as an energy supervisor for CIP energy implementations
Electrical energy	4F hex	1	Provides unified electrical energy reporting capability for CIP enabled devices and processes

### Identity object (01 hex)

The identity object provides identification and status information about the meter.

#### Class code

Hexadecimal	Decimal
01 hex	1

#### Class attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	1
2	Get	Max instances	UINT	1
3	Get	Number of instances	UINT	7
6	Get	Max ID of class attributes	UINT	7
7	Get	Max ID of instance attributes	UINT	7

#### Instance attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Vendor ID	UINT	243
2	Get	Device type	UINT	43
3	Get	Product code	UINT	10241
4	Get	Revision	Struct of: USINT USINT	1 1
5	Get	Status	WORD	Summary status of meter
6	Get	Serial number	UDINT	Serial number of meter
7	Get	Product name	SHORT_STRING	PM5560 or PM5561 or PM5563

**Supported Class and Instance services**

Class Service code	Instance Service code	Service name	Description
01 hex	01 hex	Get_Attribute_All	Return all attributes
0E hex	0E hex	Get_Attribute_Single	Return single attribute
–	05 hex	Reset	Reset the communication module of the PM55XX meters

**Message router object (02 hex)****Class code**

Hexadecimal	Decimal
02 hex	2

**Class attributes**

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	1
2	Get	Max instances	UINT	1
3	Get	Number of instances	UINT	1
4	Get	Optional attribute list	Struct of: NumbOfAttrs Attr List	NumbOfAttrs = 2 Attr1 = 1 Attr2 = 2
6	Get	Max ID number of class attribute	UINT	7
7	Get	Max ID number of instance attribute	UINT	2

**Instance attributes**

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Supported objects	Struct of: UINT Array of UINT	Number of objects = 0x09 Object IDs = 0x1 0x2 0x4 0x6 0xf4 0xf5 0xf6 0x4e 0x4f
2	Get	Max number of connections	UINT	0x20

**Supported Class and Instance services**

Service code	Service name	Description
01 hex	Get_Attribute_All	Return all attributes
0E hex	Get_Attribute_Single	Return single attribute

**Assembly object (04 hex)****Class code**

Hexadecimal	Decimal
04 hex	4

**Class attributes**

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	0x02

**Instance attributes**

Attribute ID	Access	Name	Data type	Semantic / Value
3	Set	Data	Array of Byte	Refer to Assembly input 100 and Assembly output 150 tables of the <b>Cyclical Exchanges (Implicit Exchanges)</b> for the description of the assembly sets supported by PM55XX meters.

**Supported Class and Instance services**

Service code	Service name	Description
0E hex	Get_Attribute_Single	Return single attribute

**Connection manager object (06 hex)****Class attributes**

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	1
2	Get	Max instances	UINT	1
3	Get	Number of instances	UINT	1
4	Get	Optional attribute list	Struct of: NumbOfAttrs Attr List	NumbOfAttrs = 8 Attr1 = 1 Attr2 = 2 Attr3 = 3 Attr4 = 4 Attr5 = 5 Attr6 = 6 Attr7 = 7 Attr8 = 8
6	Get	Max ID number of class attributes	UINT	7
7	Get	Max ID number of instance attributes	UINT	8

**Instance attributes**

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Open requests	UINT	Number of forward open service requests received
2	Get	Open format rejects	UINT	Number of forward open service requests which were rejected due to bad format
3	Get	Open resource rejects	UINT	Number of forward open service requests which were rejected due to lack of resources
4	Get	Open other rejects	UINT	Number of forward open service requests which were rejected for reasons other

Attribute ID	Access	Name	Data type	Semantic / Value
				than bad format or lack of resources
5	Get	Close requests	UINT	Number of forward close service requests received
6	Get	Close format rejects	UINT	Number of forward close service requests which were rejected due to bad format
7	Get	Close other rejects	UINT	Number of forward close service requests which were rejected for reasons other than bad format
8	Get	Connection timeouts	UINT	Total number of connection timeouts that have occurred in connections controlled by this connection manager

### Supported Class and Instance services

Class Service code	Instance Service code	Service name	Description
01 hex	01 hex	Get_Attribute_All	Return all attributes
0E hex	0E hex	Get_Attribute_Single	Return single attribute
–	54 hex	Forward_Open	Opens a connection
–	4E hex	Forward_Close	Closes a connection

### TCP/IP interface object (F5 hex)

#### Class attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	4
2	Get	Max instances	UINT	1
3	Get	Number of instances	UINT	1

#### Instance attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Configuration status	DWORD	0 = The interface configuration attribute has not been configured 1 = The interface configuration attribute contains configuration obtained from BOOTP, DHCP or non-volatile storage
2	Get	Configuration capability	DWORD	Bit 0 = 1 (TRUE) shall indicate that the meter is capable of obtaining its network configuration through BOOTP  Bit 1 = 1 (TRUE) shall indicate that the meter is capable of resolving host names by querying a DNS server  Bit 2 = 1 (TRUE) shall indicate that the meter is capable of obtaining its network configuration through DHCP  Bit 3 = 1 (TRUE) shall indicate that the meter is capable of sending its host name in the DHCP request  Bit 4 = 1 (TRUE) shall indicate that the Interface configuration attribute is settable  Bit 5-31: reserved

Attribute ID	Access	Name	Data type	Semantic / Value
3	Get	Configuration control	DWORD	Bits 0-3 start-up configuration 0 = The meter shall use statically-assigned IP configuration values 1 = The meter shall obtain its interface configuration values through BOOTP 2 = The meter shall obtain its interface configuration values through DHCP 3-15 = Reserved for future use Bit 4 = 1 (TRUE), the meter shall resolve host names by querying a DNS server Bit 5-31: reserved
4	Get	Physical link object	Struct of: UINT Padded EPATH	Path size Path: Logical segments identifying the physical link object
5	Get	Interface configuration	Struct of: UDINT UDINT UDINT UDINT UDINT String	IP address (0: no address configured) Network mask (0: no network mask configured) Gateway address (0: no address configured) Name server address (0: no address configured) Name server address 2 (0: no address configured) Domain name
6	Get	Host name	String	ASCII characters. Maximum length is 64 characters. Shall be padded to an even number of characters (pad not included in length). A length of 0 shall indicate no Host Name is configured
13	Get/Set	Encapsulation inactivity timeout	UINT	Number of seconds of inactivity before TCP connection or DTLS session is closed 0 = Disable 1-3600 = timeout in seconds Default = 120

### Supported Class and Instance services

Class Service code	Instance Service code	Service name	Description
–	01 hex	Get_Attribute_All	Return all attributes
0E hex	0E hex	Get_Attribute_Single	Return single attribute
–	10 hex	Set_Attribute_Single	Write one attribute

### Ethernet link object (F6 hex)

#### Class attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	4
2	Get	Max instances	UINT	2
3	Get	Number of instances	UINT	2
4	Get	Optional attribute list	Struct of: NumbOfAttrs	NumbOfAttrs = 3



Attribute ID	Access	Name	Data type	Semantic / Value
			Attr List	Attr1 = 7 Attr2 = 8 Attr4 = 10
6	Get	Max ID number of class attributes	UINT	7
7	Get	Max ID number of instance attributes	UINT	11

### Instance attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Interface speed	UDINT	Interface speed currently in use
2	Get	Interface flags	DWORD	<p>Bit 0: Link status indicates whether the Ethernet 802.3 communications interface is connected to an active network. 0 indicates an inactive link; 1 indicates an active link</p> <p>Bit 1: Half/Full duplex indicates the duplex mode currently in use. 0 indicates that the interface is running half duplex; 1 indicates full duplex</p> <p>Bit 2-4: Negotiation status</p> <ul style="list-style-type: none"> <li>0: Auto-negotiation in progress</li> <li>1: Auto-negotiation and speed detection not successful</li> <li>2: Auto negotiation not successful but detected speed duplex was defaulted</li> <li>3: Successfully negotiated speed and duplex</li> <li>4: Auto-negotiation not attempted. Forced speed and duplex</li> </ul> <p>Bit 5: Manual setting require reset</p> <ul style="list-style-type: none"> <li>0: Indicates that the interface can activate changes to link parameters (autonegotiate, duplex mode, interface speed) automatically</li> <li>1: Indicates that the meter requires a reset service be issued to its identity object in order for the changes to take effect</li> </ul> <p>Bit 6: Local hardware error</p> <ul style="list-style-type: none"> <li>0: Indicates that the interface detects no local hardware error</li> <li>1: Indicates that a local hardware error is detected. The meaning of this is product-specific</li> </ul> <p>Bit 7-31: Reserved shall be set to zero</p>
3	Get	Physical address	Array of 6 USINTs	MAC layer address
6	Set	Interface control	Struct of: WORD UINT	<p>Bit 0: Auto-negotiate</p> <p>Bit 1: Forced duplex mode</p> <p>Bit 2-15: Reserved</p>
7	Get	Interface type	USINT	0x02
8	Get	Interface state	USINT	Current state of the interface : operational (0x01), disabled(0x02), etc

Attribute ID	Access	Name	Data type	Semantic / Value
10	Get	Interface label	SHORT_STRING	Human readable identification
11	Get	Interface capability	Struct of: DWORD USINT	Capability bits Speed/Duplex options

### Supported Class and Instance services

Service code	Service name	Description
01 hex	Get_Attribute_All	Return all attributes
0E hex	Get_Attribute_Single	Return single attribute

### Port object (F4 hex)

#### Class code

Hexadecimal	Decimal
F4 hex	299

#### Class attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	2
2	Get	Max Instance	UINT	1
3	Get	Num Instances	UDINT	1
8	Get	Entry Port	UINT	1
9	Get	Port Instance Info	ARRAY of UINT STRUCT of UINT	Array of structures containing instance attributes 1 and 2 from each instance

#### Instance attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Port Type	UINT	1
2	Get	Port Number	UDINT	1
3	Get	Link Object	STRUCT of: UINT Padded EPATH	Path Length Link Path
4	Get	Port Name	SHORT_STRING	Communication interface associated with this instance
7	Get	Port Number and Node Address	Padded EPATH	Port Segment containing the Port Number of this port and the Link Address of this meter on this port
10	Get	Port Routing Capabilities	DWORD	Bit string that defines the routing capabilities of this port

**Supported Class and Instance services**

Service code	Service Name	Description
0x0E	Get Attribute Single	Used to read a Port Class attribute value. This service is required if any of the Port Class attributes are supported

**Base energy object (4E hex)****Class code**

Hexadecimal	Decimal
4E hex	78

**Class attributes**

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	2

**Instance attributes**

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Energy/Resource Type	UINT	0: Generic
2	Get	Base Energy Object Capabilities	UINT	0: Energy Measured
3	Get	Energy Accuracy	UINT	Specifies the accuracy of power and energy metering results
7	Get	Consumed Energy Odometer	ODOMETER	The consumed energy value in kWh <sup>2</sup>
8	Get	Generated Energy Odometer	ODOMETER	The generated energy value in kWh
9	Get	Total Energy Odometer	SIGNED_ODOMETER	The total net energy value
10	Get	Energy Transfer Rate	REAL	The time rate of energy consumption or production
12	Get	Energy Type Specific Object Path	Struct of: UINT Padded EPATH	Path to energy type specific object instance

**Supported Class and Instance services**

Service code	Service name	Description
0E hex	Get_Attribute_Single	Used to read a base energy class attribute value

**Odometer and Signed\_Odometer structure principle**

Data type structure		Description of data type element	Semantic / Value
ODOMETER STRUCT of:	SIGNED_ODOMETER STRUCT of:	–	–
UINT	INT	$\times 10^n$	$\pm \text{Unit1} \times 10^n$
UINT	INT	$\times 10^{n+3}$	$\pm \text{Unit1} \times 10^{n+3}$
UINT	INT	$\times 10^{n+6}$	$\pm \text{Unit1} \times 10^{n+6}$
UINT	INT	$\times 10^{n+9}$	$\pm \text{Unit1} \times 10^{n+9}$

Data type structure		Description of data type element	Semantic / Value
UINT	INT	$x10^{n+12}$	$\pm \text{Unit1} \times 10^{n+12}$
The valid range of n shall be a SINT between 0 and -15.			

### Odometer type in Kilowatt-hour units and $n = -3$

$x10^{n+12}$	$x10^{n+9}$	$x10^{n+6}$	$x10^{n+3}$	$x10^n$
Terawatt-hours (kWh $\times 10^9$ )	Gigawatt-hours (kWh $\times 10^6$ )	Megawatt-hours (kWh $\times 10^3$ )	Kilowatt-hours (kWh)	Watt-hours (kWh $\times 10^{-3}$ )

### Electrical energy object (4F hex)

#### Class code

Hexadecimal	Decimal
4F hex	79

#### Class attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Revision	UINT	2

#### Instance attributes

Attribute ID	Access	Name	Data type	Semantic / Value
1	Get	Real Energy Consumed Odometer	ODOMETER	The total real energy consumed
2	Get	Real Energy Generated Odometer	ODOMETER	The total real energy generated
3	Get	Real Energy Net Odometer	SIGNED ODOMETER	Specifies the accuracy of power and energy metering results
4	Get	Reactive Energy Consumed Odometer	ODOMETER	The consumed energy value in kWh <sup>2</sup>
5	Get	Reactive Energy Generated Odometer	ODOMETER	The generated energy value in kWh
6	Get	Reactive Energy Net Odometer	SIGNED ODOMETER	The total net energy value
7	Get	Apparent Energy Odometer	ODOMETER	The total apparent energy consumed Range from 0 kVAh to 999,999,999.999 kVAh
9	Get	Line Frequency	REAL	Hz 0.0...999.9 $\times 10^{21}$
10	Get	L1 Current	REAL	Amps (A) 0.0...999.9 $\times 10^{21}$
11	Get	L2 Current	REAL	Amps (A) 0.0...999.9 $\times 10^{21}$
12	Get	L3 Current	REAL	Amps (A) 0.0...999.9 $\times 10^{21}$
13	Get	Average Current	REAL	Amps (A) 0.0...999.9 $\times 10^{21}$
14	Get	Percent Current Unbalance	REAL	Percent 0.0...100.0
15	Get	L1-N Voltage	REAL	Volts (V) RMS 0.0...999.9 $\times 10^{21}$

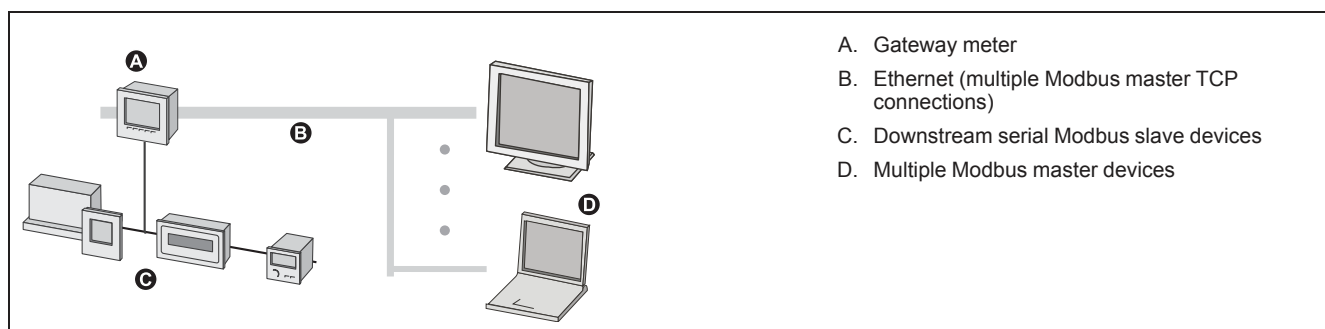
Attribute ID	Access	Name	Data type	Semantic / Value
16	Get	L2-N Voltage	REAL	Volts (V) RMS 0.0...999.9x10 <sup>21</sup>
17	Get	L3-N Voltage	REAL	Volts (V) RMS 0.0...999.9x10 <sup>21</sup>
18	Get	Average L-N Voltage	REAL	Volts (V) RMS 0.0...999.9x10 <sup>21</sup>
19	Get	L1-L2 Voltage	REAL	Volts (V) RMS 0.0...999.9x10 <sup>21</sup>
20	Get	L2-L3 Voltage	REAL	Volts (V) RMS 0.0...999.9x10 <sup>21</sup>
21	Get	L3-L1 Voltage	REAL	Volts (V) RMS 0.0...999.9x10 <sup>21</sup>
22	Get	Average L-L Voltage	REAL	Volts (V) RMS 0.0...999.9x10 <sup>21</sup>
23	Get	Percent Voltage Unbalance	REAL	Percent voltage deviation between phases 0.0...100.0
24	Get	L1 Real Power	REAL	Watts (W) ±...999.9x10 <sup>21</sup>
25	Get	L2 Real Power	REAL	Watts (W) ±...999.9x10 <sup>21</sup>
26	Get	L3 Real Power	REAL	Watts (W) ±...999.9x10 <sup>21</sup>
27	Get	Total Real Power	REAL	Watts (W) ±...999.9x10 <sup>21</sup>
28	Get	L1 Reactive Power	REAL	Volt-amps reactive (VAR) ±...999.9x10 <sup>21</sup>
29	Get	L2 Reactive Power	REAL	Volt-amps reactive (VAR) ±...999.9x10 <sup>21</sup>
30	Get	L3 Reactive Power	REAL	Volt-amps reactive (VAR) ±...999.9x10 <sup>21</sup>
31	Get	Total Reactive Power	REAL	Volt-amps reactive (VAR) ±...999.9x10 <sup>21</sup>
32	Get	L1 Apparent Power	REAL	Volt-amps (VA) 0.0...999.9x10 <sup>22</sup>
33	Get	L2 Apparent Power	REAL	Volt-amps (VA) 0.0...999.9x10 <sup>22</sup>
34	Get	L3 Apparent Power	REAL	Volt-amps (VA) 0.0...999.9x10 <sup>22</sup>
35	Get	Total Apparent Power	REAL	Volt-amps (VA) 0.0...999.9x10 <sup>22</sup>
36	Get	L1 True Power Factor	REAL	Percent -100...100
37	Get	L2 True Power Factor	REAL	Percent -100...100
38	Get	L3 True Power Factor	REAL	Percent -100...100
39	Get	Three Phase True Power Factor	REAL	Percent -100...100
40	Get	Phase Rotation	UINT	0 = None 1 = ABC 2 = ACB
41	Get	Associated Base Energy Object Path	Struct of: UINT Padded EPATH	Path to associated Base Energy Object instance

**Class and Instance services**

Service code	Service name	Description
0E hex	Get_Attribute_Single	Used to read a electrical energy class attribute value

## Modbus Ethernet gateway

The meter's Ethernet gateway feature extends the meter's functionality by allowing Ethernet access to serial devices connected to the meter's RS-485 serial communications port.



A Modbus master device (such as an energy management system) can communicate through the gateway meter to a serial network of devices connected to the gateway meter's serial port(s). The meter receives Modbus TCP/IP data on TCP port 502, translates it to Modbus RTU then forwards it to the addressed slave device.

This functionality allows the use of monitoring software to access information from slave devices for data collection, trending, alarm/event management, analysis, and other functions.

## Ethernet gateway implementation

There is specific implementation information to consider when using your meter as an Ethernet gateway.

### Firmware support

The Ethernet gateway functionality is available on firmware version 2.0.1 or later.

### Addressing

You can use slave address 255 or the Unit ID configured in the gateway meter's serial settings to send a request to the gateway-enabled meter itself. Messages addressed with other unit IDs are forwarded by the gateway meter to the RS-485 slave devices.

### Broadcast messages

The gateway meter always processes broadcast messages (in other words, messages sent to Unit ID 0). You can configure whether or not broadcast messages are forwarded to the slave devices.

### Modbus master TCP/IP connections

The maximum number of Modbus master TCP connections allowed for the Ethernet gateway is configurable. It is the same as the maximum number of total Modbus TCP/IP connections that are configured on the gateway-enabled meter.