Creating a Smart Energy Device
Using Digi's XBee® SE RF Module

Introduction
Overview
The ZigBee Smart Energy public profile is a collection of device descriptions and functionality that enables utilities to intelligently manage energy loads, monitor energy usage and optimize consumption in a secure environment.

Digi International has created the XBee and XBee-PRO SE RF Modules to provide a major building block for manufacturers developing any of the eight Smart Energy devices specified by the ZigBee Smart Energy public profile. These include:

- ESP (Energy service portal)
- Metering device
- In-premise display device
- PCT (Programmable communicating thermostat)
- Load controller
- Smart appliance
- Prepayment terminal
- Range extender

Manufacturers must implement a defined communications protocol to ensure Smart Energy devices are able to intercommunicate simply and securely.

Digi's XBee SE RF module features a smart energy firmware load that is customized specifically for the ZigBee Smart Energy profile. The diagram below illustrates how an XBee SE operates in a smart energy device, including what the XBee does and what the host microcontroller should do.
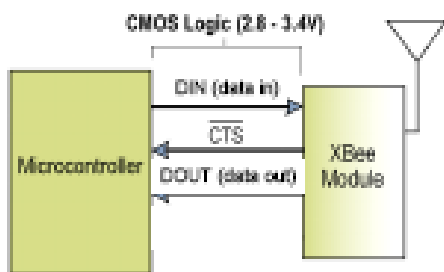
The Microcontroller Firmware:
- Is the brains of the device
- Writes a driver to talk to the XBee API via UART
- Responds to ZigBee Device Profile commands (e.g. simple descriptor, active endpoints, match descriptor requests)
- Supports the required and any optional clusters for the specific smart energy device
- Implements mandatory attributes and commands, as well as any optional attributes and commands to be supported by the device.

The XBee SE Firmware:
- Is the wireless ZigBee interface of the device
- Provides a simple serial API to a certified ZigBee stack
- Manages forming/joining networks
- Supports key establishment cluster and authentication
- Features configuration registers to allow simple customization and diagnostics of advanced ZigBee features including: Security and networking, Enabling/disabling joining, End device interactions (polling, sleep durations, etc.), Many-to-one routing.

By using the XBee SE module as part of a Smart Energy device, developers can focus on supporting the high level application communications protocols without digging into the technical intricacies of the ZigBee stack. In addition, XBee and XBee-PRO SE RF modules have been fully certified by the FCC, ETSI and other approval agencies, eliminating the need for RF engineering, testing and manufacturing.

Firmware Loads

The XBee Smart Energy firmware is a firmware load for XBee Series 2 RF modules. This means developers can convert XBee ZB and ZNet 2.5 RF modules into SE modules simply by changing firmware. Look for the Smart Energy Conversion Kit on www.digi.com/standards/smart-energy to do this. The XBee Smart Energy firmware currently supports the following variations:

- 31xx (API Coordinator)
- 33xx (API Router)
- 39xx (API End device)

Changing from coordinator to router or router to end device requires a new firmware load. Firmware can be uploaded serially to the XBee either programmatically or using a serial firmware update tool such as Digi's X-CTU. See the XBee SE RF module manual for details.

Reference Documentation

This document references several ZigBee standards documents available to members of the ZigBee Alliance. At the time of this writing, the following documents were used as references:

Document Description	Document Number
ZigBee Specification	053474r17
ZigBee Cluster Library Specification	075123r02
ZigBee Smart Energy Specification	075356r15
ZigBee Smart Energy Profile Test Specification	07538r17

The XBee SE product manual includes additional information on the XBee SE firmware. The product manual can be downloaded on www.digi.com/standards/smart-energy.

Recommendations to Manufacturers

We recommend manufacturers consider the following steps when creating Smart Energy devices:
- Obtain a copy of the ZigBee Specifications listed above
- Subscribe to the email list and consider attending the weekly AMI teleconferences to stay abreast of upcoming specification changes
- Stay in touch with se.developer@digi.com for technical support while building the product
- Obtain test certificates (see the XBee SE RF module manual Appendix D for details on certificates)
- Certify the Smart Energy device with a ZigBee test house
- Obtain production certificates

Remaining Sections

The remaining sections in this application note cover the following important topics related to developing a Smart Energy device:
- Forming / Joining a Smart Energy Network
- XBee Commissioning
- ZigBee Device Profile (ZDP) Support
- ZigBee Cluster Library (ZCL) Support
- Implementing Smart Energy Devices
- APS Encryption
- Trust Center Requirements

- Network Management
- Smart Energy Device Certification

Forming/Joining a Smart Energy Network

The XBee is designed to make forming and joining a smart energy network simple. The following sections describe the default XBee behavior for each device type. Refer to the XBee SE manual for details.

Coordinator

The coordinator is configured to start a network on one of the 14 channels in the default scan channels list (11 – 24). The coordinator performs an energy scan and active scan to ensure it selects a channel with minimal energy where a PAN ID conflict will not occur. Once the coordinator forms a network, it sends a modem status API frame out its UART indicating the coordinator started, and its Associate LED pin (pin 15) toggles at a rate of one blink per second. The AI command, if queried, returns a value of 0, indicating a successful start. The actual operating channel and PAN ID can be queried with the CH, OP and OI commands. See the XBee SE manual for details.

By default, the coordinator selects a random 64-bit and 16-bit PAN ID for the network. The coordinator is also configured as the trust center and network manager. It supports key establishment on endpoint 0x5E and can respond to all of the key establishment commands, allowing joining devices to authenticate and obtain a new link key. It will respond to active endpoint, match descriptor or simple descriptor requests that are sent to it by joining devices. For example, joining devices may query the coordinator for an endpoint that supports the key establishment cluster using a match descriptor request. (Applications can optionally respond to these and other ZDO requests with custom endpoint and/or cluster ID information using the AO command. See the "ZigBee Device Profile" support section later in this app note.)

Permit-joining time is defined by the NJ setting, which has a default of 0xFE, or 254 seconds. Once the coordinator starts a network and the permit-join time completes, joining must be enabled in order for additional devices to join the network. Joining can be enabled by changing the NJ value, by sending a CB command with a parameter of 2, or by using the commissioning push button feature. See the SE manual for details on the commissioning push button feature.

Router

The router is configured to attempt to join a network up to three times when first powered on. By default, the router will scan 14 of the 16 channels (11 – 24) and join any PAN that is allowing joining. After the router joins a network, it will attempt to discover an endpoint on the trust center that supports key establishment by sending a match descriptor request to the coordinator. If the trust center responds with a match descriptor response, the router will attempt to perform key establishment.

When the router joins a network (before key establishment begins), it will send a modem status API frame out its UART indicating the router has joined a network. Once key establishment completes, it will send an API frame out the UART indicating the device authenticated successfully, and its Associate LED pin (pin 15) toggles at a rate of two blinks per second. The AI command, if queried, returns a value of 0, indicating a successful join to a smart energy network.

If the three join attempts fail, the AI command may be queried to identify the cause of failure. The AI register has a large number of defined status values; see the XBee SE manual for definitions. A non-zero value indicates the reason for the failure. AI is set to 0x30 if the router joins a network, but has not yet authenticated. Authentication can take several seconds. The router will attempt joining again when either a CB command is sent with a parameter of 1, or by using the commissioning push button feature. See the SE manual for details on the commissioning push button feature.

NOTE: By default the XBee SE has key establishment disabled. If key establishment is enabled (EO command), the XBee router requires a certificate installed in order to join a smart energy network. See the XBee SE RF module manual Appendix D for details on certificates.

End Device

The end device is configured to attempt to join a network up to three times when first powered on. By default, the end device will scan 14 of the 16 channels (11 – 24) and join any PAN that is allowing joining. After the end device joins a network, it will attempt to discover an endpoint on the trust center that supports key establishment by sending a match descriptor request to the coordinator. If the trust center responds with a match descriptor response, the end device will attempt to perform key establishment.

When the end device joins a network (before key establishment begins), it will send a modem status API frame out its UART indicating the end device has joined a network. Once key establishment completes, it will send an API frame out the UART indicating the device authenticated successfully, and its Associate LED pin (pin 15) toggles at a rate of two blinks per second. The AI command, if queried, returns a value of 0, indicating a successful join to a smart energy network.

If the three join attempts fail, the AI command may be queried to identify the cause of failure. A non-zero value indicates the reason for the failure. AI is set to 0x30 if the end device joins a network, but has not yet authenticated. Authentication can take several seconds. The end device will attempt joining again when either a CB command is sent with a parameter of 1, or by using the commissioning push button feature. See the SE manual for details on the commissioning push button feature.

Once the end device has joined a network, it sends a poll request to its parent device to see if the parent received any RF data in its behalf. The end device continues polling while awake, disabling its receiver between polls if the parent has no data for it.

NOTE: By default, the XBee SE has key establishment disabled. If key establishment is enabled (EO command), the XBee end device requires a certificate installed in order to join a smart energy network. See the XBee SE RF module manual Appendix D for details on certificates.

XBee Commissioning

The XBee SE firmware includes AT commands that can be set or queried to manage commissioning attributes. In most cases, the default values should suffice. The following table maps smart energy startup attributes to AT commands.

Startup Attribute Set Parameter	AT Command	Description	Range / Default
Short Address	MY	Read the 16-bit network address a device receives after joining a ZigBee network.	0x0000 – 0xFFFB
Extended PAN ID	ID, OP	ID – Sets the 64-bit PAN ID the device will join. Setting it to 0 will join any PAN ID (default). OP – Reads the actual 64-bit PAN ID the device joined. If ID is set to a non-zero value, OP will report the same value as ID.	0 – 0xFFFFFFFFFFFFFFFF Default: 0 (join any PAN ID)
16-bit PAN ID	II, OI	II – Sets the initial 16-bit PAN ID. Setting it to 0xFFFF will join any 16-bit PAN ID. (This value should rarely be set to a non-default value.) OI – Reads the actual 16-bit PAN ID the device joined.	0 – 0xFFFF Default: 0xFFFF (join any 16-bit PAN ID)
Channel Mask	SC	Set / read the 16-bit scan channels bitmask. Each bit represents one of the channels in the 2.4 GHz band from channel 11 (2405 MHz) to	0 – 0xFFFF Default: 0x3FFF

		channel 26 (2480 MHz). For example, setting SC to 0x0042 would scan channels 12 and 17. Note: The XBee-PRO doesn't support channels 25 and 26.	
Stack Profile	ZS	Read the ZigBee stack profile.	0x02
Trust Center Address	-	The 16-bit address of the trust center.	0x0000
Link Key	KY	Set the initial 128-bit link key. This command register cannot be read.	16-byte value Default: 0x0001
Trust Center Network Key	NK	Set the 128-bit network key (coordinator only). This command register cannot be read.	16-byte value Default: 0x0001
Use Insecure Join	-	Determines if insecure joins can be used during joining.	FALSE
Scan Attempts	-	Specifies the maximum number of times a device should attempt to scan in order to find a ZigBee router or coordinator to join.	3
Time between scans	-	Delay between consecutive active scans when attempting to discover a ZigBee network to join.	1 second
Concentrator Flag / Concentrator Discovery Time	AR	Set/read the concentrator flag and discover time value. If AR is set to 0xFF, the device is not a concentrator. Otherwise, the device sends a many-to-one route discovery broadcast every AR * 10 seconds. If AR is set to 0, a single broadcast will be sent.	0 – 0xFF Default: 0xFF (Not a concentrator)
Concentrator Radius	BH	Set/read the maximum broadcast radius for all broadcast transmissions, including the many-to-one route request.	0 – 0x1E Default: Should be 11

Sending AT Commands

The AT Command and AT Command Queue Parameter API frames (0x08 and 0x09 respectively) are used to set or query AT command values.

The AT Command and AT Command Queue Parameter frames are outlined in the XBee SE manual.

General Commissioning Practices

The default configuration for XBee routers and end devices allows the devices to join any PAN ID that is allowing joining on any of 14 channels (channels 11 – 24). The XBee can support all 16 channels; the XBee-PRO cannot support the higher two channels. The PAN ID and scan channel masks are configurable with the ID, II and SC commands. In general, routers and end devices should leave the ID, II and SC commands at their default values to ensure the device can join any PAN ID.
Smart energy devices must have a valid certificate installed. See the XBee SE RF module manual Appendix D for details on certificates.

ZigBee Device Profile (ZDP) Support

The ZigBee Device Profile defines an interface for managing ZigBee devices, and performing device and service discovery. See the ZigBee specification for a complete list of features included in the ZDP.

Most ZDP messages involve a command / response interaction. For example, a request message such as a network address request or match descriptor request might be sent to a remote device. The remote device that the request was intended for would respond with the appropriate response.

By default, the XBee smart energy firmware responds to all received ZDP request messages. However, many of the ZDP messages are intended to discover services (e.g., endpoints, cluster IDs, etc.) that are supported on the device. Therefore, the application must receive ZDP request messages and send properly formatted responses.

For the application layer to receive ZDP request messages, the AO command must be set to 3 (non-default value). When AO is set to 3, the XBee firmware will no longer send responses to many of the received ZDP requests. The application is responsible to manage all ZDP requests that are sent out the UART.

With AO set to 3 (enable ZDP passthrough), the XBee will process the following ZDP messages:

- Network Address Request (cluster ID 0x0000)
- IEEE Address Request (cluster ID 0x0001)
- Node Descriptor (cluster ID 0x0002)
- Power Descriptor (cluster ID 0x0003)
- Most Mgmt ZDP messages (cluster IDs 0x0030 – 0x0034, 0x0036, 0x0038)

The application must respond to:

- Simple Descriptor Request (CID 0x0004)
- Active Endpoints Request (CID 0x0005)
- Match Descriptor Request (CID 0x0006)
- Unsupported ZDP requests

Example 1 – Simplified Code to Manage ZDP Messages

```
// Received RF data packet
If(( packet->profileID == ZIGBEE_DEVICE_PROFILE_ID) && (packet->destEndpoint ==
ZIGBEE_DEVICE_OBJECTS_ENDPOINT)){
 Switch (packet->clustered){
  Case SIMPLE_DESC_REQ_ZDP:
    sendSimpleDescResponse();
    break;
  case ACTIVE_EP_REQ_ZDP:
   sendActiveEndpointsResponse();
   break;
  case MATCH_DESC_REQ_ZDP:
   sendMatchDescriptorResponse();
   break;
  case USER_DESC_REQ_ZDP:
   sendUserDescResponse();
   break;
  default:
   sendUnsupportedResponse();
 }
```

ZigBee Cluster Library (ZCL) Support

The ZigBee Cluster Library specification provides a foundation to retrieve information from ZigBee devices using a command format that can be understood among all ZigBee devices. The library is a collection of clusters, where each cluster in the library is a collection of commands and attributes that together define an interface to a specific functionality. For example, the TIME cluster can be used to synchronize devices in the network together with a similar timestamp. The TIME cluster as defined in the ZCL includes attributes to specify time-related information such as UTC time and a local time zone. In most cases, these attributes can be queried or set.

The following table shows several standard ZigBee clusters with some of the attributes and commands they support. Note that each attribute and command has a unique identifier within the scope of the cluster.

Table 1 – Sample Clusters, Attributes and Commands

Cluster (Cluster ID)	Server Side Attributes (Attribute ID)	Commands (Command ID)
Basic (0x0000)	- Application Version (0x0001) - Hardware Version (0x0003) - Model Identifier (0x0005)	- Reset to defaults (0x00)
Identify (0x0003)	- Identify Time (0x0000)	- Identify (0x00) - Identify Query (0x01)
Time (0x000A)	- Time (0x0000) - Time Status (0x0001) - Time Zone (0x0002)	
Thermostat (0x0201)	- Local Temperature (0x0000) - Occupancy (0x0002)	- Setpoint raise / lower (0x00)

Client/Server

Clusters are typically implemented in a client/server model. The server of a cluster typically stores attributes that can be manipulated or queried by the client of the cluster. In some cases, attributes may be present on the client of a cluster.

Commands that manipulate or read attributes are typically sent from a client device and received by the server. The response is then sent by the server device to the client. The simple descriptor of a device lists the input clusters and output clusters supported on the device. Input clusters are synonymous with server clusters, and the output clusters are the client clusters supported on the device.

Public profiles such as Smart Energy or Home Automation require their respective devices to implement a defined set of functions. Some of these functions may be defined in the profile itself, while others may be defined in the ZCL. Smart Energy devices, for example, must support the basic and time clusters in the ZCL. This document will explore several clusters in detail.

ZCL Frame Format

The ZigBee Cluster Library defines a simple message service that includes a ZCL frame header and ZCL payload. All ZCL commands that are transmitted to devices in the network must follow the ZCL frame formatting conventions as defined in the ZCL specification.

The ZCL frame header is defined as follows (see ZigBee Cluster Library, section 2.3.1):

Table 2 – ZCL Frame Header

Name	Size (Bytes)	Description
ZCL Frame control	1	Bitfield value that defines the command type and other information relevant to the ZCL command. The ZCL frame control field is decoded in the following table.
Manufacturer Code	0 / 2	This field is included when a ZCL command is a manufacturer specific (proprietary) extension to a profile. It is only included if the Manufacturer Specific bit in the ZCL frame control byte is set.
Transaction Sequence Number	1	A sequence number or identifier for the ZCL command. This value is set to an arbitrary value.
Command Identifier	1	This specifies the cluster command being used.

The bitfield for the ZCL frame control field in the ZCL header is enumerated below.

Table 3 – ZCL Frame Control Bitfield

Name	Bits	Description
Frame Type	0-1	00 – Command acts across the entire profile 01 – Command is specific to a cluster 10-11 – Reserved
Manufacturer Specific	2	Set the bit if the command refers to a manufacturer specific extension to a profile. If set, the manufacturer code field shall be present in the ZCL frame header. Otherwise, the manufacturer code field is omitted.
Direction	3	Set the bit if the command is being sent from the server side of a cluster to the client side. Clear the bit if the command is sent from the client side to the server side.
Disable Default Response	4	This determines if the default response command should be returned. See ZigBee Cluster Library specification, section 2.4.12 for details.
Reserved	5-7	Set to 0.

General Commands

The Frame Type bits of the ZCL Frame Control bitfield specifies if the command applies to the entire profile, or if the command is specific to a cluster. For example, the Smart Energy profile defines a load control event command as part of the demand response and load control cluster. This command is specific to a cluster.

In contrast, general commands are defined across an entire profile and not specific to a particular cluster. General commands are most often used to perform some operation on attributes such as discover, read, write and report. When these commands are sent, the Frame Type bits in the ZCL Frame Control must be set to 00. The following table lists some of the general commands. See ZigBee Cluster Library specification, section 2.4 for a complete listing of commands.

Table 4 – Listing of Some General Commands

Command (Command ID)	Description
Read Attributes (0x00)	Used to read one or more attributes on a remote device.
Read Attributes Response (0x01)	Generated in response to a read attributes command.
Write Attributes (0x02)	Used to change one or more attributes on a remote device.
Write Attributes Response (0x04)	Sent in response to a write attributes command.
Configure Reporting (0x06)	Used to configure a device to automatically report on the values of one or more of its attributes.

Report Attributes (0x0A)	Used to report attributes when report conditions have been satisfied.
Discover Attributes (0x0C)	Used to discover the attribute identifiers on a remote device.
Discover Attributes Response (0x0D)	Sent in response to a discover attributes command.

XBee API

The XBee API includes simple frames to transmit and receive ZigBee APS messages. The transmission frame allows data to be sent using any endpoints, cluster ID and profile ID combination. The data payload of the API frame can be constructed to send ZCL messages.

ZigBee Explicit Transmission (0x11)

The XBee SE firmware also includes an explicit receive API frame that indicates the endpoints, cluster ID and profile ID used in each received RF packet. The explicit receive frame must be enabled on an API device by setting the AO command to 1 or 3.

The data payload in the explicit transmit request is transmitted over-the-air exactly how the data is received. When sending ZDO or ZCL messages, 2-byte and 4-byte values must be sent in little endian byte order to be properly decoded by the receiver. For example, if a ZDO frame requires sending the destination 16-bit address in the ZDO payload, and if the destination 16-bit address is 0x1234, the data should be ordered as 0x34 0x12 bytes in the API payload.

ZigBee Explicit Receive Indicator (0x91)

Similar to ZDO and ZCL transmissions, received ZDO and ZCL data should have all 2-byte and 4-byte parameters ordered in little-endian byte order.

ZigBee Cluster Library API Example

The following example sends ZCL commands to a device in a ZigBee network. This example uses an arbitrary endpoint and profile ID. In a real ZigBee network, to discover which clusters are supported on a device, we must first identify the endpoints on the device, and then interrogate each endpoint to find what cluster and profile ID(s) are supported. The Active Endpoints ZDO can be sent to a device to get a list of endpoints, and the Simple Descriptor ZDO can then be sent to query the list of clusters and the profile ID on each endpoint.

Example – Read the hardware version (attribute ID 0x0003) from the basic cluster (general cluster ID = 0x0000) from a device in the network. Suppose the device has a 64-bit address of 0x0013A200 40401234. The following table shows how the API can send this command.

	Frame Fields				Offset	Example	Description
A P I P a c k e t	Start Delimiter				0	0x7E	
	Length				MSB 1	0x00	Number of bytes between the length and checksum.
					LSB 2	0x19	
	Frame Specific Data	Frame Type			3	0x11	Explicit transmit API frame identifier.
		Frame ID			4	0x01	Identifies the UART data frame for the host to correlate with a subsequent transmit status. If set to 0, no transmit status frame will be sent out the UART.
		64-bit Destination Address			MSB 5	0x00	64-bit address of the destination device (big endian byte order). For unicast transmissions, set to the 64-bit address of the destination device,
					6	0x13	
					7	0xA2	
					8	0x00	

				Offset	Value	Description
				9	0x40	or to 0x0000000000000000 to send a unicast to the coordinator. Set to 0x000000000000FFFF for broadcast.
				10	0x40	
				11	0x12	
				12	0x34	
	16-bit Destination Address			MSB 13	0xFF	16-bit address of the destination device (big endian byte order). Set to 0xFFFE for broadcast, or if the 16-bit address is unknown.
				LSB 14	0xFE	
	Source Endpoint			15	0x5E	Set to the source endpoint on the sending device. (0x5E arbitrarily selected)
	Destination Endpoint			16	0x5E	Set to the destination endpoint on the remote device. (0x5E arbitrarily selected)
	Cluster ID			MSB 17	0x00	Set to the cluster ID that corresponds to the ZCL command being sent. 0x0000 = Basic Cluster
				LSB 18	0x00	
	Profile ID			MSB 19	0x01	Set to the profile ID supported on the device. (0x0109 is the smart energy profile.)
				LSB 20	0x09	
	Broadcast Radius			21	0x00	Sets the maximum number of hops a broadcast transmission can traverse. If set to 0, the transmission radius will be set to the network maximum hops value.
	Transmit Options			22	0x00	All bits must be set to 0.
	Data Payload	ZCL Frame Header	Frame Control	23	0x00	Bitfield that defines the command type and other relevant information in the ZCL command. See the ZCL specification for details.
			Transaction Sequence Number	24	0x01	A sequence number used to correlate a ZCL command with a ZCL response. (The hardware version response will include this byte as a sequence number in the response.) The value 0x01 was arbitrarily selected.
			Command ID	25	0x00	Since the frame control "frame type" bits are 00, this byte specifies a general command. Command ID 0x00 is a Read Attributes command.
		ZCL Payload	Attribute ID	26	0x03	The payload for a "Read Attributes" command is a list of Attribute Identifiers that are being read. Note the 16-bit Attribute ID (0x0003) is sent in little endian byte order (0x0300). All multi-byte ZCL header and payload values must be sent in little endian byte order.
				27	0x00	
Checksum				28	0xFA	0xFF minus the 8 bit sum of bytes from offset 3 to this byte.

Implementing Smart Energy Devices

Each smart energy device must support a defined set of commands and attributes to interoperate with other smart energy devices, and to obtain product certification. This section identifies the various smart energy device types and the required clusters, attributes, and commands for each. Details can be found in the ZigBee Cluster Library and ZigBee Smart Energy specifications.

The following device types are defined in the Smart Energy profile:

- ESP (Energy service portal)
- Metering device
- In-premise display device
- PCT (Programmable communicating thermostat)
- Load controller
- Smart appliance
- Prepayment terminal
- Range extender

The requirements for each device are elaborated in the following sections. Note that the key establishment cluster must be supported on all smart energy devices. (The key establishment cluster has been implemented in the XBee firmware, so it is omitted from the tables below.)

Energy Service Portal

The energy service portal is intended to connect the energy management devices in the smart energy network with the utility company's connection network. See section 6.3.1 of the smart energy specification for details.

Digi offers a selection of Ethernet and cellular gateway devices with support for ESP functionality. These products enable OEMs to quickly deploy energy service portal devices for smart energy networks. Contact the Digi sales team for details.

Name	Value	ZigBee Standards Reference
Device Attributes		
Device ID	0x0500 (ESP)	SE spec 5.9
General ZCL Clusters		
Basic (server)	0x0000	SE spec 5.11 ZCL spec 3.2.2
Time (client)	0x000A	SE spec 5.11 ZCL spec 3.12.3
Smart Energy Clusters		
Message (server)		SE spec, D.5.2
Price (server)		SE spec, D.4.2
Demand Response and Load Control (server)		SE spec, D.2.2
Time (server)		SE spec 5.11 ZCL spec 3.12
Messaging Commands		
Messaging Server Command: Display Message	0x00 (command ID)	SE spec, D.5.2.3
Messaging Server Command: Cancel Message	0x01 (command ID)	SE spec, D.5.2.3

Price Commands		
Price Server: GetCurrentPrice	0x00	SE spec, D.4.2.3
DRLC Commands		
DRLC Server Command: Load Control Event	0x00 (command ID)	SE spec, D.2.2.3.1
DRLC Server Command: Cancel Load Control Event	0x01 (command ID)	SE spec, D.2.2.3.2
DRLC Server Command: Cancel All Load Control Events	0x02 (command ID)	SE spec, D.2.2.3.3

Metering Device

The metering device consists of a ZigBee device and a meter (electricity, gas, water, heat, etc.). This device may report meter readings when queried or periodically on its own. See section 6.3.2 of the smart energy specification for details.

The meter must support the clusters and attributes listed in the following table. The SE specification also lists various optional clusters (time, prepayment, price and message), attributes and commands that are not included here.

Name	Value	ZigBee Standards Reference
Device Attributes		
Device ID	0x0501 (Metering)	SE spec 5.9
General ZCL Clusters		
Basic (server)	0x0000	SE spec 5.11 ZCL spec 3.2.2
Smart Energy Clusters		
Simple Metering (server)		SE spec, 6.3.2.1
Metering Attributes		
Metering Server: Reading Information Set: CurrentSummationDelivered	0x0000	SE spec, D.3.2.2.1
Metering Server: Meter Status Set: Status	0x0200	SE spec, D.3.2.2.3
Metering Server: Formatting Attribute Set: UnitofMeasure	0x0300	SE spec, D.3.2.2.4
Metering Server: Formatting Attribute Set: SummationFormatting	0x0303	SE spec, D.3.2.2.4
Metering Server: Formatting Attribute Set: MeteringDeviceType	0x0306	SE spec, D.3.2.2.4

The Explicit API frames (0x11 and 0x91) can be used to send and receive smart energy messages. To develop a metering device using the XBee SE firmware, the application must do the following:

- Respond correctly to the ZDP requests (defined earlier)
- Respond to ZCL attribute requests to query or set the simple metering client attributes

- Support any desired optional clusters and/or commands

In-Premise Display

The in-premise display relays energy consumption data to the user through a display. See section 6.3.3 of the smart energy specification for details.

Name	Value	ZigBee Standards Reference
Device Attributes		
Device ID	0x0502 (In premise display)	SE spec 5.9
General ZCL Clusters		
Basic (server)	0x0000	SE spec 5.11 ZCL spec 3.2.2
Smart Energy Clusters		
At least one of the Price, Simple Metering, or Messaging (client) clusters must be implemented.		

Programmable Communicating Thermostat (PCT)

This device can control the heating and cooling systems in a facility. See section 6.3.6 of the smart energy specification for details.

Name	Value	ZigBee Standards Reference
Device Attributes		
Device ID	0x0503 (PCT)	SE spec 5.9
General ZCL Clusters		
Basic (server)	0x0000	SE spec 5.11 ZCL spec 3.2.2
Time (client)	0x000A	SE spec 5.11 ZCL spec 3.12.3
Smart Energy Clusters		
Demand Response and Load Control (client)		SE spec, D.2.3
DRLC Attributes		
DRLC Client Attribute: Utility Enrolment Group	0x0000	SE spec, D.2.3.2.1
DRLC Client Attribute: Start Randomize Minutes	0x0001	SE spec, D.2.3.2.2
DRLC Client Attribute: Stop Randomize Minutes	0x0002	SE spec, D.2.3.2.3
DRLC Client Attribute: Device Class Value	0x0003	SE spec, D.2.3.2.4
DRLC Commands		
DRLC Server Command: Load Control Event	0x00 (command ID)	SE spec, D.2.2.3.1
DRLC Server Command:	0x01 (command ID)	SE spec, D.2.2.3.2

Name	Value	ZigBee Standards Reference
Cancel Load Control Event		
DRLC Server Command: Cancel All Load Control Events	0x02 (command ID)	SE spec, D.2.2.3.3
DRLC Client Command: Report Event Status	0x00 (Command ID)	SE spec, D.2.3.3.1
DRLC Client Command: Get Scheduled Events	0x01 (Command ID)	SE spec, D.2.3.3.2

Load Controller

The load controller is a device that can receive demand response and load control events to either supply power to a load, or to shed a load for a specified time. Examples include water heaters, exterior lighting control, pool pumps, etc. See section 6.3.5 of the smart energy specification for a detailed description.

The following table outlines mandatory information that should be used when developing a load controller using the XBee SE firmware. The SE specification includes information on many optional clusters, attributes, and commands that are not included here.

Name	Value	ZigBee Standards Reference
Device Attributes		
Device ID	0x0504	SE spec 5.9
General ZCL Clusters		
Basic (server)	0x0000	SE spec 6.1 ZCL spec 3.2.2
Time (client)	0x000A	SE spec 5.11, 6.3.5 ZCL spec 3.12.3
Smart Energy Clusters		
Demand Response and Load Control (client)	0x0701	SE spec, D.2.3
ZCL Attributes		
Basic Device Information	0x0000 (ZCL Version)	ZCL spec 3.2.2
DRLC Attributes		
DRLC Client Attribute: Utility Enrolment Group	0x0000	SE spec, D.2.3.2.1
DRLC Client Attribute: Start Randomize Minutes	0x0001	SE spec, D.2.3.2.2
DRLC Client Attribute: Stop Randomize Minutes	0x0002	SE spec, D.2.3.2.3
DRLC Client Attribute: Device Class Value	0x0003	SE spec, D.2.3.2.4
DRLC Commands		
DRLC Server Command: Load Control Event	0x00 (command ID)	SE spec, D.2.2.3.1
DRLC Server Command: Cancel Load Control Event	0x01 (command ID)	SE spec, D.2.2.3.2
DRLC Server Command: Cancel All Load Control Events	0x02 (command ID)	SE spec, D.2.2.3.3

DRLC Client Command: Report Event Status	0x00 (Command ID)	SE spec, D.2.3.3.1
DRLC Client Command: Get Scheduled Events	0x01 (Command ID)	SE spec, D.2.3.3.2

The Explicit API frames (0x11 and 0x91) can be used to send and receive smart energy messages. To develop a load controller using the XBee SE firmware, the application must do the following:

- Respond correctly to the ZDP requests (defined earlier)
- Respond to the DRLC server commands with the correct DRLC Client Command (response).
- Respond to ZCL attribute requests to query or set the DRLC client attributes
- Properly shed loads as required based on the received DRLC load control event messages.
- Support any desired optional clusters and/or commands.

Smart Appliance

This device type is for appliances that contain intelligence to participate in energy management activities. It is still being defined. See section 6.3.5 of the smart energy specification for details.

Name	Value	ZigBee Standards Reference
Device Attributes		
Device ID	0x0505 (smart appliance)	SE spec 5.9
General ZCL Clusters		
Basic (server)	0x0000	SE spec 5.11 ZCL spec 3.2.2
Time (client)	0x000A	SE spec 5.11 ZCL spec 3.12.3
Smart Energy Clusters		
Price (Client)		SE spec, D.2.2
Price Commands		
Get Current Price	0x00 (command ID)	SE spec, D.4.2.3

Prepayment Terminal

This device is still being defined. The smart energy specification states that the information for this device should be used as reference material until the prepayment cluster is completely defined. See section 6.3.8 in the SE spec.

Name	Value	ZigBee Standards Reference
Device Attributes		
Device ID	0x0506 (Prepayment Terminal)	SE spec 5.9
General ZCL Clusters		
Basic (server)	0x0000	SE spec 5.11 ZCL spec 3.2.2
Time (client)	0x000A	SE spec 5.11 ZCL spec 3.12.3
Smart Energy Clusters		
Price (client)		SE spec, 6.3.8
Time (client)		SE spec, 6.3.8
Prepayment (server)		SE spec, 6.3.8

Prepayment (client)		SE spec, 6.3.8
Price Commands		
Get Current Price	0x00 (command ID)	SE spec D.4.2.3

Range Extender

The range extender is a simple router that is used to extend the RF range between smart energy devices. Range extenders may be deployed as necessary to promote adequate RF coverage in a deployment. The range extender must be a router, and it cannot support any device functionality of the other smart energy device types. See section 6.3.6 of the smart energy specification for details.

The following table outlines mandatory information that the wall router must support.

Name	Value	ZigBee Standards Reference
Device Attributes		
Device ID	0x0008	SE spec 5.9
General ZCL Clusters		
Basic (server)	0x0000	SE spec 6.1 ZCL spec 3.2.2
ZCL Attributes		
Basic Device Information	0x0000 (ZCL Version)	ZCL spec 3.2.2

APS Encryption

ZigBee defines encryption at both the network and APS layers. The network key is used to encrypt data at the network layer. It is obtained during joining and shared among all devices in the network. The link key is used to encrypt data at the APS layer. In smart energy, the link key is known only to the source and destination devices, providing end-to-end encryption.

In smart energy, network encryption is always applied and APS encryption is required only for certain messages. APS encryption can be applied on a per-packet basis by setting the transmit options byte in the Explicit transmit API frame (0x11). All smart energy clusters, along with the time and commissioning clusters require APS encryption. (See table 5.10 in section 5.4.6 in the smart energy specification for more information.)

When the XBee receives an RF packet, the receive options byte in the explicit receive API frame (0x91) indicates if the packet was encrypted with APS encryption or not. The receive options byte will have a 0x20 bit value if the packet was APS encrypted.

Trust Center Requirements
Registering Devices

In order to join a network, smart energy devices must be registered with the trust center. (In rare cases, joining devices can be pre-configured with the trust center link key if known.) The content in this section applies if devices will be registered with the trust center (i.e., the trust center link key is not known). The XBee API includes a Register Device API frame (frame ID = 0x24) that can be used to register devices with the XBee coordinator (trust center). This API stores a 64-bit address and initial link key for a given device into the trust center's key table. Up to 14 devices can be registered with the trust center. Once a device joins a smart energy network and completes key establishment, the key table entry for that device is updated with the key that was generated during key establishment. To remove the device (and its key) from the key table, send a Register Joining Device API frame with the key field omitted.

Many-to-One and Source Routing

The smart energy trust center must support many-to-one and source routing. To do this, the AR command on the XBee coordinator (trust center) should be set to a value less than 0xFF (default). This will send a periodic many-to-one route request broadcast transmission every (AR * 10) seconds.

To support source routing, the XBee should store received route record frames (frame ID = 0xA1). It can then use the create source route frame (0x21) to send the correct source route to the XBee on demand, just prior to sending data to a remote.

See the XBee SE manual for more information on many-to-one and source routing.

Network Management

The ZigBee Device Profile includes a number of management primitives that can detect RF interference, change the operating channel and discover devices and services in the network.

The Explicit transmit and receive API frames in the XBee firmware allow applications to take advantage of these features in the ZigBee Device Profile. The following table maps a ZDO to a particular function. Note that some functions are optional; meaning devices from different vendors may or may not respond to the request.

Feature	ZDO	Cluster ID	Mandatory / Optional
Read Child Tables	Network Address Request	0x0000	M
Discover Devices	Match Descriptor Request	0x0006	M
Discover Devices	Device Announce	0x0013	M
Read Neighbor Tables	Mgmt LQI Request	0x0031	O
Read Routing Tables	Mgmt Rtg Request	0x0032	O
Remove Devices	Mgmt Leave Request	0x0034	O
Enable Joining	Mgmt Permit Joining Request	0x0036	M
Energy Detection	Mgmt Network Update	0x0038	O

Smart Energy Device Certification

Smart Energy end products must be run through a series of tests by a ZigBee accredited certification lab (e.g., NTS, TUV Rheinland). These test labs often offer a test suite that can be rented, prior to certification testing, to ensure the device responds correctly to all Smart Energy messages. Companies must be members of the ZigBee Alliance in order to certify a Smart Energy end product. See http://www.zigbee.org for details.