UG103.10: RF4CA Fundamentals

This document describes the ZigBee RF4CE specification, with notes about considerations when implementing an RF4CE solution. It includes a basic description of RF4CE device types, the network formation process, power saving, and security.

Silicon Labs' *Application Development Fundamentals* series covers topics that project managers, application designers, and developers should understand before beginning to work on an embedded networking solution using Silicon Labs chips, networking stacks such as EmberZNet PRO or Silicon Labs Bluetooth Smart, and associated development tools. The documents can be used a starting place for anyone needing an introduction to developing wireless networking applications, or who is new to the Silicon Labs development environment.

KEY POINTS

- RF4CE devices
- Network formation
- Topology
- Power saving
- Security
- Transmission modes
- Frequency agility
- Profiles

1. Introduction

The ZigBee RF4CE specification describes mechanisms for building remote control (RC) networks for simple, robust, low cost communication for consumer electronic (CE) devices. RF4CE provides simple networking and application layers on top of the IEEE 802.15.4 standard in the 2.4 GHz frequency band. A multiple star network topology is used in RF4CE with a variety of transmission options including both broadcast and unicast with optional MAC-level acknowledgement and optional network-level security. Frequency agility and standard power saving mechanisms help ensure that RF4CE products are able to meet consumer expectations for reliability and long life. These features together enable manufacturers to build a diverse range of remote control products, including home entertainment devices and keyless entry systems. At the discretion of the application, RF4CE networks are multi-vendor interoperable. A number of ZigBee-developed application profiles as well as manufacturer-specific profiles are already available.

2. Definitions

- Controller – a network participant that has ZigBee RF4CE functionality
- Originator – the device from which a transmission is sent
- Recipient – the device to which a transmission is sent
- Target – a network coordinator that has ZigBee RF4CE functionality

3. Devices

RF4CE networks consist of controller nodes and target nodes. The fundamental difference between these two device types is that targets may create their own networks while controllers are only capable of joining existing networks. In ZigBee PRO terms, controllers are roughly analogous to end devices while targets are more similar to coordinators.

3.1 Controllers

Controllers are network participants, which means they can join existing networks. An example of a controller device is a handheld remote control for a television or set-top box. Consumers will typically interact with a controller to operate a target some distance away. Before communicating with other nodes, controllers must first join to an existing network using a discovery and pairing process described in section 4. Network Formation.

Controllers are frequently battery operated and therefore usually operate in a low power mode in order to meet consumer expectations for battery longevity. The RF4CE specification offers a specific power saving mechanism to allow nodes to sleep for extended periods while still allowing other nodes to communicate with them. A node that wishes to conserve power will enable its receiver for some duration within a larger period. The duty cycle of sleepy nodes helps them save power while also facilitating communication to them by other nodes in the network. Power saving is described in more detail in section 6. Power Saving.

Because most actions begin with a user operating a controller, controllers often perform the originator role in RF4CE networks and the terms “controller” and “originator” are frequently synonymous. This is not always the case, however, so care must be taken not to conflate the two concepts.

3.2 Targets

Targets are network coordinators, which mean they can create new networks. In addition to being capable of forming networks, targets may also join networks created by other targets. The equivalent in ZigBee PRO is a coordinator that may choose to join an existing network as a router instead of forming its own network. The ability to form a network is the key distinction between a controller and a target, but it is important to remember that either device type can join a network.

Typical examples of a target device are televisions or set-top boxes. These types of products are often packaged with a dedicated remote control from the same manufacturer. The product itself plus its dedicated remote control comprise the most basic example of a complete RF4CE network. A television, for example, would create a network in which to operate and the included remote control would join to that network in order to interact with the television. The network parameters could be specified during production so the two products are already joined together before reaching the consumer or the network creation and the joining process could be completed during an initial setup procedure initiated by the consumer.

Targets frequently perform the recipient role in RF4CE networks. As before, it is important to remember that “target” and “recipient” are not always synonymous. It is possible for a target to have its own network and to be joined with another network created by a separate target. For example, a set-top box will typically have its own network with its own dedicated remote. It may also join to the network of the television to which it is physically connected. In a setup like this, the set-top box would be the recipient on its own network and an originator on the network belonging to the television.

4. Network Formation

RF4CE networks consist of one or more devices paired to a target. The network joining process is comprised of two distinct steps: discovery and pairing. The discovery procedure is used to identify targets within radio range. Once a potential target is identified, the pairing process is used to join with that device. Discovery usually precedes pairing, but a device is free to pair with other devices whose identities are made known to it through out-of-band means. For example, a bar code printed on a set-top box may provide the information to a remote control for pairing.

4.1 Discovery

Discovery is the process by which devices learn about other devices in the vicinity. Its primary purpose is to identify targets with which to pair. The device that initiates discovery is known as the originator while devices that are discovered are known as recipients. Typically, discovery is performed by controllers, but targets are permitted to act as originators as well.

During discovery, the originator periodically transmits discovery request messages on each of the RF4CE channels. These messages are typically directed to all nodes within range, but may also be transmitted to a specific node or to all nodes within an existing network. The discovery request includes information about the originator, including its capabilities and vendor information. Additionally, the originator specifies which application device types it is attempting to discover. A multi-function remote control, for example, may wish to discover only televisions.

Because ZigBee RF4CE does not have parent-child relationships like in ZigBee PRO, discovery can only identify other nodes that have their receiver enabled and are within immediate range of the originator. Originators are free to repeat discovery until an application- or profile-specific condition has been satisfied. For example, discovery may continue for a fixed duration or until some number of responses have been received. Repeating the discovery process increases the likelihood of locating devices within range.

When a target receives a discovery request, it examines the originator information and the requested application device type contained in the request in order to determine whether to respond. A set-top box, for example, should not to respond to a discovery request for a television. Similarly, a device may choose to respond only to requests from the same manufacturer. Ignoring discovery requests is one way that a target can exert control over the devices that join its network.

If a target decides to respond to a discovery request, it transmits a discovery response back to the originator. The response includes information about the recipient, including its capabilities and application device type.

Targets may also enable an automatic discovery mode. When activated, the stack will automatically determine whether to respond to discovery requests based on whether the capabilities advertised by the originator and the requested application device type are compatible with the local node. If so, the stack will automatically respond with the information for the local node. Otherwise, the stack will ignore the request. Automatic discovery mode ends after an application-specified duration or if a response is sent, whichever comes first.

As the originator receives discovery responses from recipients, it uses the recipient information to decide whether the recipient is an acceptable target. At the conclusion of the discovery process, the application will have collected a list of potential targets. Typically, the originator will attempt to pair with one or more of the targets in the list. The decisions about which targets are acceptable and whether to proceed to pairing are application and profile specific.

4.2 Pairing

Pairing is the process by which devices establish bidirectional links with other devices. Both controllers and targets can act as pairing originators and initiate the pairing process. Regardless of whether the originator is a controller or target, the pairing recipient must be a target. Controllers may not pair with other controllers.

Once an originator identifies a target that it wants to pair with, either through discovery or some out-of-band mechanism, pairing may begin. The originator starts the process by transmitting a pairing request to the recipient. Pair requests, like discovery requests, include information about the originator and the target uses this information to decide how to respond. If the target wishes to accept the pair request, it transmits a pair response back to the originator with an indication of success. Otherwise, it responds with an indication of failure.

If a pairing is successful and if the originator and recipient both support security, a key exchange procedure is then attempted. The key exchange establishes a link key that is used to encrypt messages sent between the originator and recipient. This procedure is described in section 7. Security.

Once the pairing and the key exchange, if applicable, completes successfully, a bidirectional link between the two nodes is established. Both the originator and recipient will store information about the other node in an entry in its pairing table. The pairing table includes addressing information for the other node and is stored persistently. Each entry is assigned a unique pairing reference. Once a pairing is established, the application sends messages to the other node using the pairing reference assigned to that node. The stack will translate the pairing reference to the actual network address before transmitting the message. Similarly, when a message is received, the stack will determine the pairing reference of the sender based on the source address and will provide that reference to the application instead of the actual source address. In this way, addressing is abstracted and simplified for the application and the application is therefore not required to store any specific addressing information for other nodes. A consequence of this abstraction, however, is that nodes cannot communicate with each other unless they have previously paired. The stack will ignore incoming messages from and refuse to transmit outgoing messages to unpaired devices.

Both controllers and targets may pair with any number of other devices. The capacity of the pairing table is implementation specific, although targets must be able to maintain simultaneous pairings with at least five other devices. For example, supporting multiple pairings permits a television to be controllable by either its factory-supplied remote control or a separate multi-function remote control. Similarly, a multi-function remote control is able to control, for example, a television and set-top box by virtue of supporting multiple pairings.

Either side of a pairings may decide to remove a pairing after it has been created. The ability to unilaterally remove a pairing enables a device to determine which other devices are permitted to communicate with it. This is essential in a consumer environment where devices may change frequently with each new generation of technology. A television, for example, that supports five pairings may need to preemptively remove a pairing entry for a remote control that is infrequently used in order to accommodate a new remote control. Similarly, a single-function remote control will need to remove and replace its single pairing entry each time it pairs with a new device. This functionality ensures devices are not permanently tied to old devices that no longer exist.

5. Topology

The following figure shows an example of a typical RF4CE network. The network includes three target nodes: a television, a DVD player, and a set-top box. Each target has formed its own network and is paired with a dedicated remote control in that network. A multi-function remote control has also paired with all of the targets. By pairing with multiple targets, the multi-function remote control has connected the separate networks into a larger network with a multiple-star topology.

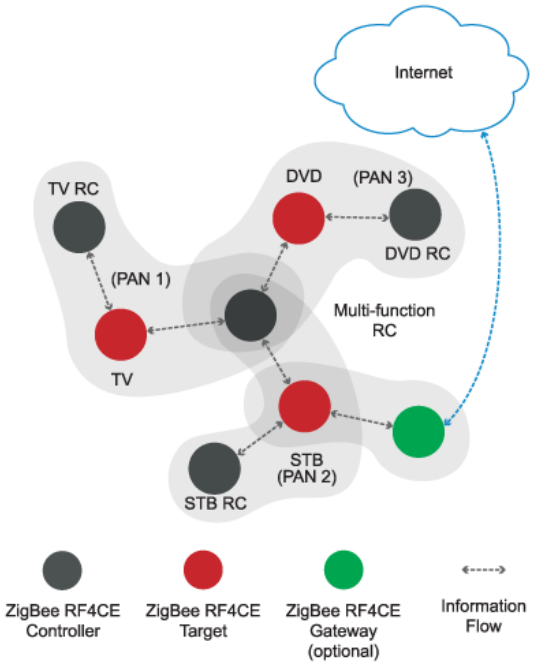


Figure 5.1. Multiple Star Topology

To communicate with the various target devices, the multi-function remote control will switch to the channel of the target and assume its personal area network (PAN) identifier. The controller will use the network address allocated during pairing to communicate with the target. The channel, PAN identifier, and network addresses of the local node and the target node are stored persistently in the pairing table.

The figure also shows a gateway device that is providing Internet connectivity to the set-top box. Other types of gateways are also possible. For example, a gateway could provide connectivity to a ZigBee PRO Home Automation (HA) network operating in the same location as the RF4CE network. Gateways are application-specific and outside the scope of the RF4CE specification.

6. Power Saving

In order to maximize battery life, RF4CE defines two power-saving mechanisms. Each application is free to manage its own power consumption in a way most suitable to its particular characteristics and usage. Although power saving is traditionally used for controllers, both mechanisms are also available for use by targets.

6.1 Direct Control

RF4CE explicitly permits the application to directly control the state of its receiver and, ultimately, its sleep schedule during normal operation. The application may, for example, wake up and enable its receiver indefinitely. This is recommended for a line-powered device, such as a television, when it is turned on. Conversely, the application may also disable its receiver and sleep indefinitely. A remote control may wish to enter such a dormant state if some duration of time has elapsed since a button press or if it senses that the user is no longer holding the remote.

The application can also enable and disable its receiver on its own schedule to achieve fine-grained control of the power consumption of the device. Unlike end devices in ZigBee PRO, sleepy devices in RF4CE do not periodically check in with a parent, so devices are free to sleep for extended durations without network-layer consequences. However, application profiles may impose restrictions that effectively limit the sleep duration, so it is important to understand the constraints of the broader environment in which the device operates.

6.2 Duty Cycling

Although RF4CE permits direct control over the receiver state, leaving the application in control can severely reduce the ability for other nodes to communicate with the device. Fortunately, the receiver does not need to be enabled all of the time for other nodes to effectively communicate with the device. Because of retries at various levels, a device that regularly enables its receiver may still receive messages in a timely manner. This method of power savings is called duty cycling.

The RF4CE specification allows a device to set an active period and a duty cycle. At the start of each duty cycle, the stack will enable the receiver for the duration of the active period. When the active period elapses, the stack will disable the receiver until the end of the current duty cycle. When the duty cycle elapses, the process repeats with the receiver again enabled for a new active period. This is shown in the following figure.

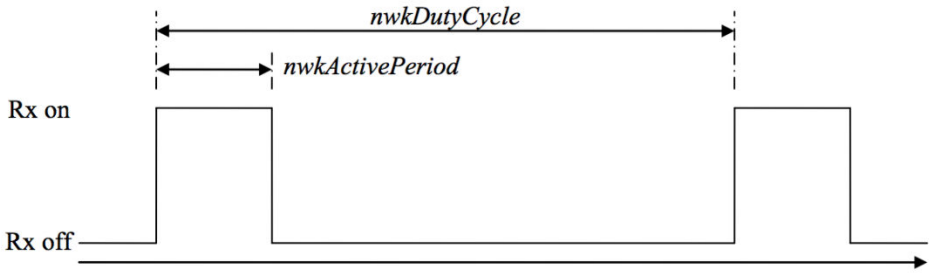


Figure 6.1. Duty Cycling

By setting appropriate active period and duty cycle parameters, the application can achieve predictable power consumption and, because of the regularity of the enabled receiver, other nodes will be able to effectively and reliably communicate with the device.

7. Security

The RF4CE specification provides a security mechanism based on link keys, similar to those used in the Smart Energy profile in ZigBee PRO. As in ZigBee PRO, link keys provide confidentiality and authenticity between pairs of nodes while also preventing replay attacks. However, the mechanism for deriving link keys is very different in RF4CE.

During pairing, the originator and recipient each indicate if they support security. Following a successful pairing, if both nodes support security, the recipient will generate a random 128-bit security link key to be used for communication between the pair. The recipient then transmits a series of key seed messages to the originator. The key seed messages contain random data mixed with the link key itself. In aggregate, the data contained in the key seed messages is sufficient to derive the link key. Once all of the key seed messages are successfully transmitted to the originator, the originator will test the validity of the link key by transmitting an encrypted ping request containing random data to the recipient. If the recipient receives and can decrypt the ping request, it will send a ping response back to the originator with the same random data. If the originator receives and decrypts the ping response, the link key is known to be valid and will be used for future secure communication between the originator and recipient. Messages are encrypted using the AES-128-CCM* algorithm, as in ZigBee PRO.

Because the key seed data is used to derive the link key, any malicious eavesdropping node can subvert the security of the pairing if it is able to receive all of the key seed messages. However, the security risks of this scheme are reduced by the inherent lossiness and asymmetry of wireless networks. As the number of key seed messages increases, the probability that the originator and an eavesdropper successfully receive the same set of key seed messages decreases. To reduce the opportunity for an eavesdropper to receive the complete set of key seed messages, the recipient transmits the messages without retries and at a reduced power level. If the originator fails to acknowledge a key seed message, either because it did not receive the message or because the recipient did not receive the acknowledgement, the recipient will generate a new message to replace it. If an eavesdropper misses any of the key seed messages, it will not be able to derive the link key.

8. Transmission Modes

RF4CE offers a number of transmission options. The various options allow devices to balance power consumption and reliability in a way most suited to their particular use cases.

As in ZigBee PRO, messages can be transmitted with or without a request for an 802.15.4-level acknowledgement. If the sender requests acknowledgement, the receiver will transmit an acknowledgement upon receipt of the message. If an acknowledgement is not received, the message is assumed to have been lost.

A mode unique to RF4CE is channel agility. In single-channel mode, the stack will attempt to transmit the message on the last-known channel of the network, as in ZigBee PRO. If the network has changed channels, the intended recipient may not receive the message. If the originator instead uses channel agility or multiple-channel mode, an unacknowledged message will automatically cause the stack to retransmit the message on the other channels. This is an important part of frequency agility, as described in section 9. Frequency Agility.

Finally, as in ZigBee PRO, messages may be unicast or broadcast in RF4CE. Unlike ZigBee PRO, however, recipients will automatically drop messages from devices to which they are not paired. This rule applies to both unicasts and broadcasts. All nodes that wish to communicate with each other must be paired.

9. Frequency Agility

RF4CE uses the 2.4 GHz frequency band, just like ZigBee PRO. To reduce interference from other devices in this band, RF4CE avoids channels commonly used by other protocols and instead restricts itself to channels 15, 20, and 25. At startup, targets first perform an energy scan across the permitted channels in order to identify the best available channel on which to operate. When other nodes pair with the target, they will record the channel and use it for transmitting messages.

If the target determines that its current channel is congested and that network performance may suffer as a result, it may select a new channel from the permitted set of channels. There is no explicit notification of a channel change, so other nodes paired to the target will not know about the change. Instead, the other nodes must implicitly learn of the change by detecting failed transmissions to the target.

When other nodes attempt to send a message to the target, they will transmit on the original channel and, therefore, the message will not be heard or acknowledged by the target. The other nodes will react to the unacknowledged message by attempting to retransmit it on each of the three RF4CE channels in succession. Eventually, the message will be transmitted on the new channel and the target will receive and acknowledge the message. When this happens, the other node will record the new channel and use it for all subsequent messages to the target.

This simple mechanism allows the network to react to interference and for nodes to realign with each other through the normal course of communication. However, the mechanism relies on the use of two optional transmission modes: acknowledged messages and multi-channel transmissions. Without acknowledgements, a node will be unable to detect that the target has moved. When sending, the application must also explicitly specify to the stack that multi-channel transmissions are to be used. Otherwise, single-channel mode is utilized and the stack will not automatically retransmit the message on the other channels even in the event of an unacknowledged message.

10. Profiles

The ZigBee RF4CE specification deals exclusively with network-level commissioning and communication. Application-level functionality is contained within ZigBee- or manufacturer-defined profiles. Each profile builds upon the networking layer and adds additional commissioning steps as well as attributes and commands. RF4CE profiles are similar to clusters in the ZigBee Cluster Library (ZCL) used in ZigBee PRO.

10.1 Generic Device Profile

The Generic Device Profile (GDP) is meant to serve as the basis for other profiles. It provides a standard commissioning scheme as well as a set of commands for performing basic functions that are applicable to a range of applications.

One key feature of the GDP profile is a standard commissioning mechanism. The commissioning process includes the standard discovery and pairing process, but augments it with an algorithm to rank and sort potential targets, a way to perform application-specific configuration, and a standard way to recover and continue in the event of pairing failures.

Following a successful pairing, the GDP profile enters a configuration procedure during which each applicable application profile is given an opportunity to exchange information specific to the profile. These application-specific configuration procedures allow a single GDP-based pairing to configure multiple GDP-based profiles in a single operation from the perspective of the user.

Following a successful pairing and configuration, the GDP profile defines a validation procedure. Only if the validation procedure succeeds are the devices considered bound and able to communicate. The validation procedure is implementation specific, but typically involves a challenge-response mechanism. For example, when a remote pairs with a television, the television may generate and display a string of numbers that the user must then enter on the remote. The validation procedure involves the user in the pairing to ensure the intended devices are paired.

The GDP profile also provides attributes and a way to exchange attribute values between paired devices. The GDP profile itself defines a set of attributes, and higher-level profiles are able to add their own sets as well. Attributes in the GDP profile are similar to attributes in the ZCL. One notable change is that GDP devices are required to store the attribute values of the devices to which they are paired. Typically, attribute values are exchanged during pairing and configuration. When a device wants to communicate with another device, it checks its local copy of the attribute value from the other device to determine how to interact with it. For example, during commissioning, a television may indicate it supports an interactive menu by setting a particular attribute value and transmitting it to the remote. If the user presses the menu button on the remote, the remote will first check to see whether the television has indicated support for the button. This feature allows devices to communicate using messages and options that are appropriate.

10.2 ZigBee Remote Control

The ZigBee Remote Control (ZRC) profile is intended to be used for consumer remote control applications. The original 1.0 version of the profile was called Consumer Electronics Remote Control (CERC) profile, but it was renamed to ZRC with version 1.1.

Versions 1.0 and 1.1 of the ZRC profile defined a simple push-button pairing mechanism to commission devices. In order to pair, both the originator and the recipient push a button to enter the commissioning state. Pairing will only succeed if the originator identifies exactly one possible target. This simple approach helps ensure that remotes can unambiguously pair with a specific target.

ZRC 1.0 and 1.1 also include a set of commands for controlling a target from a remote. The commands reflect the state of a button press on a remote. For example, if a user presses the power button on a remote, the remote transmits a user control press message with the identifier of the power button. If the user continues to hold the button down, the remote transmits repeat messages. When the user releases the button, the remote transmits a release message. The set of available buttons are taken from the HDMI Consumer Electronics Control (HDMI-CEC) specification. It is also possible for ZRC 1.0 and 1.1 devices to discover which HDMI-CEC actions are supported by the paired device.

The ZRC profile has been redesigned with version 2.0. It is now based on GDP 2.0 and therefore uses its commissioning scheme. The commands to control a target have been extended to support actions from other specifications in addition to HDMI-CEC. The various sets of actions are referred to as action banks. The RF4CE working group has identified a list of available action banks and has provided for manufacturer-defined action banks as well.

The commissioning scheme and over-the-air commands used by ZRC 2.0 are not inherently compatible with ZRC 1.0 or 1.1. To address this backwards compatibility problem, ZRC 2.0 devices are required to also support ZRC 1.1. In this way, ZRC 1.x and ZRC 2.x devices can coexist and interoperate.

10.3 Multiple System Operators

The Multiple System Operators (MSO) profile is a manufacturer-specific profile originally developed by Comcast and subsequently contributed to CableLabs, a not-for-profit research and development consortium of cable operators. The MSO profile is based on the ZRC 1.1 profile with extensions added to enhance commissioning.

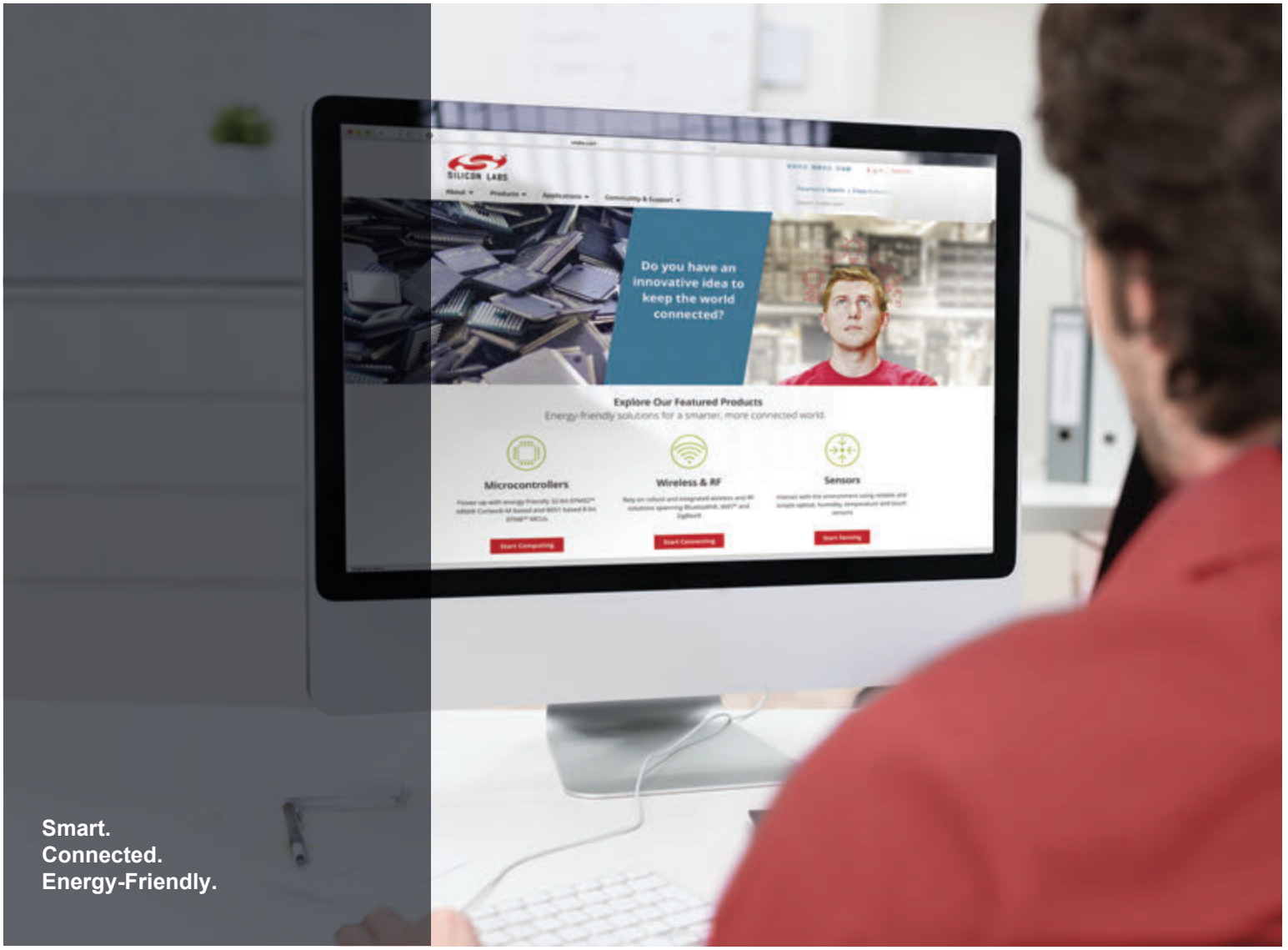
After pairing and key exchange, MSO devices enter a procedure to validate the pairing. This process is similar to the validation procedure provided by the GDP profile.

Although the specification is shared with a consortium, it is not intended to allow devices from different manufacturers to interoperate. In fact, the pairing mechanism explicitly forbids devices to pair with other devices that do not share the same vendor identification.

Although the MSO profile is similar to the ZRC profiles, it is not compatible with them. A device is, however, free to implement MSO and ZRC at the same time.

11. Next Steps

The EmberZNet software includes a certified ZigBee RF4CE networking stack and reference implementations for the GDP, ZRC 1.1 and 2.0, and MSO profiles. Customers are encouraged to use the included sample applications to gain familiarity with RF4CE in general and the Silicon Labs offering in particular. Each of the applications demonstrate how devices discover and pair with each other as well as how messages are sent and received. The applications are available for use after loading the EmberZNet installation in Ember AppBuilder. Ember Desktop includes support for decoding the network- and application-layer messages in RF4CE and provides additional insight into the operation of RF4CE networks.



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS®

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com