Electronic Design

By BRIAN BEDROSIAN, Vice President of Marketing,
IoT Compute and Wireless Business Unit,
Cypress Semiconductor

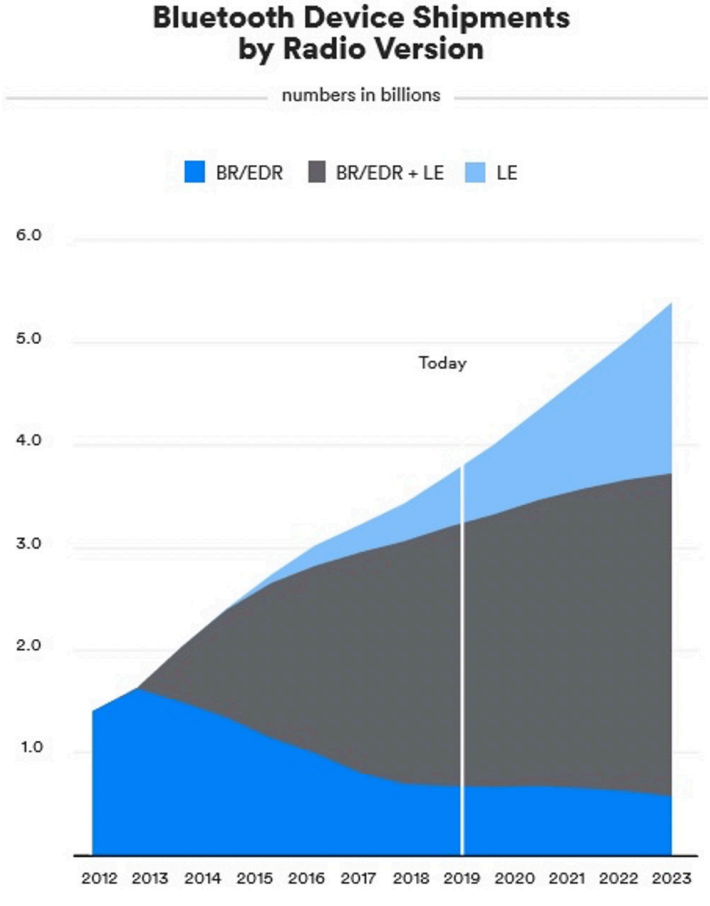
Design Considerations for Bluetooth Mesh Across Industrial, Home Environments

BLE Mesh technology expands Bluetooth's reach into a multitude of applications, but it brings to bear numerous design tradeoffs in both hardware and software. Here's an introduction to BLE Mesh and an overview of those tradeoffs framed in a smart-lighting context.

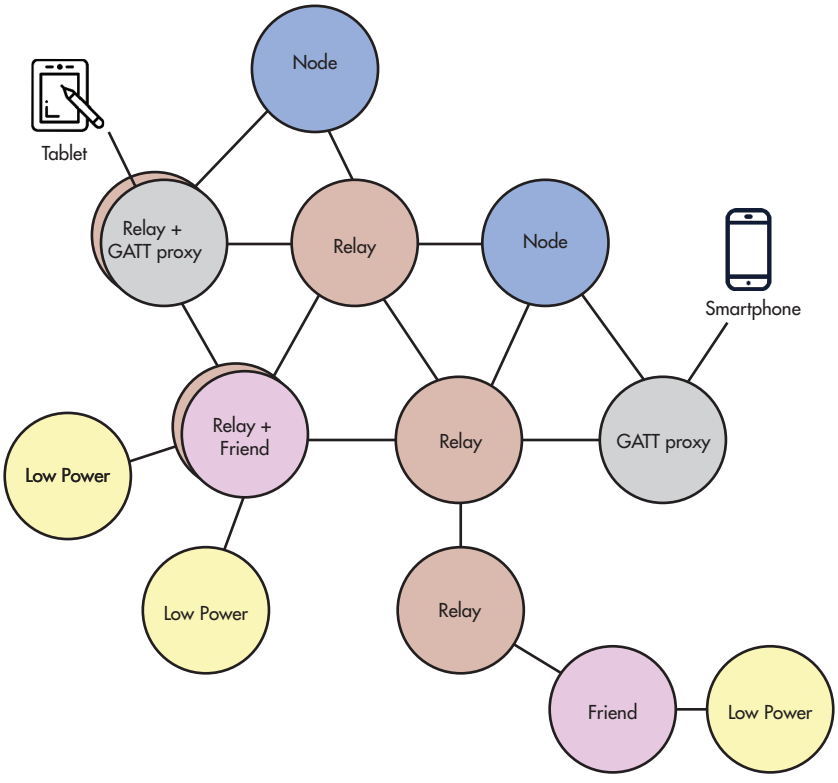
Bluetooth is a ubiquitous communications protocol with countless applications in consumer electronics, healthcare, industrial automation, and asset tracking. With Bluetooth Low Energy (BLE) Mesh now added as a network layer, there are even greater opportunities for simultaneous control and monitoring of hundreds—even thousands—of devices. These new capabilities come with added complexity for developers, though.

Bluetooth's many advantages have given rise to its now-ubiquitous presence. The Bluetooth standards are maintained and advanced by the Bluetooth Special Interest Group (SIG), which as of this writing has some 33,000 members in 150 countries. The original standards specified point-to-point (1:1) connections, with multipoint (one-to-many or 1:m) and mesh (many-to-many or m:m) added later. Classic Bluetooth supports 1:1 and 1:m communications with both a Basic Rate (BR) and an Enhanced Data Rate (EDR). Bluetooth Low Energy (BLE) is the only mode that supports m:m mesh networking.

The advent of BLE Mesh networking has further expanded the scale and scope of potential applications for Bluetooth. With support for over 30,000 network nodes, BLE Mesh can handle applications that



1. The adoption of Bluetooth continues broadly across end markets, with the Low Energy mode taking an increasingly prominent role over time. (Source: Bluetooth SIG)



2. The virtually unlimited scalability and high resiliency of BLE Mesh makes it suitable for a wide variety of new and demanding use cases.

span large buildings, healthcare enterprises, and campuses.

Bluetooth adoption has grown rapidly and is forecasted to continue apace *(Fig. 1)*. Such widespread use has made Bluetooth networking ubiquitous across virtually all platforms, including smartphones and tablets, smart watches, laptops, and peripherals, ranging from keyboards and mice to speakers and headsets. Such ubiquity across interfaces and widespread interoperability across brands establishes an installed base and developer/user familiarity that makes Bluetooth an obvious choice for many new applications.

BLE Mesh Overview

Every device in a BLE Mesh network must meet fundamental requirements identified in the specifications. This section provides an overview of those requirements using terminology adopted by the Bluetooth SIG.

Mesh-Network Topology

A mesh-network topology enjoys two significant advantages: virtually unlimited scalability and high resiliency, both of which have contributed to the protocol's popularity among product design engineers. These advantages derive from the many-to-many communications that form multiple paths throughout the network from source to destination *(Fig. 2)*.

The m:m connections assure successful communications even when multiple nodes have failed or been taken out of service, whether temporarily or permanently. Putting it another way: BLE Mesh networks can expand far and wide with no single points of failure.

Mesh-Node Types

The scalable topology enables BLE Mesh networks to support a theoretical maximum of 32,767 nodes—a number that places no practical limits on real-world applications. The standards define four types of nodes, and any single node can be configured to support multiple types:

- *Relay Nodes* retransmit or relay received messages to propagate them throughout the mesh network. Messages are only relayed when their time-to-live (TTL) value is greater than zero. Except for Low Power Nodes, all BLE Mesh devices should support this capability.
- *Low Power Nodes (LPNs)* are used primarily for battery-powered, low-duty-cycle sensors. To minimize power consumption, LPNs are normally assigned a companion "Friend Node" to serve as an intermediary for messages.
- *Friend Nodes* receive messages on behalf of their assigned LPN(s), storing them in a queue for later delivery. Each LPN periodically "wakes up" and polls its Friend Node to receive any new messages that might be in its queue.
- *Proxy Nodes* relay messages between the connection-oriented General ATTribute (GATT) Bearer and the Advertising Bearer in the BLE Mesh network. This feature enables devices that support BLE (but not the BLE Mesh stack) to communicate with the mesh network without any need for a dedicated gateway or other special provision.

Mesh-Node Element(s)

Every node is required to have a primary identification element that defines its basic functionality. They also may optionally have one or more secondary elements to define additional functionality. For example, a switch (the primary element) might have an occupancy sensor as a secondary element, too. Or, an occupancy sensor (the primary element) might also have a light-level sensor as a secondary element.

Mesh-Network Addresses

There are four types of addresses in a BLE Mesh network, all of which are assigned during the secure provisioning process. Note that addresses are assigned to elements, which means that a node with multiple elements will have multiple

addresses.

• *Unicast addresses* uniquely identify each individual element to enable point-to-point communications.

• *Group addresses* represent multiple elements to enable multicast communications. The Bluetooth SIG has defined four Fixed Group addresses: All-proxies, All-friends, All-relays, and All-nodes.

• *Virtual addresses* create virtual groups of elements or nodes to enable additional, dynamic multicast communications capabilities.

• *Unassigned addresses* identify elements that have yet to be provisioned with their Unicast, Group, and/or Virtual addresses.

Mesh-Node Models

BLE Mesh nodes employ one of three different types of models: Client, Server, or Control. These models are determined by a node's basic function or functions, as it's possible to implement more than one model in a single node.

• *Server models* contain and expose the state of an element; for example, a luminaire being on or off or at some intermediate brightness level.

• *Client models* interact with Server models by sending and receiving messages; for example, when a switch is used to turn off or dim a luminaire.

• *Control models* combine Client and Server models in a single node, and typically include control logic (i.e., rules and behaviors). For example, an outdoor luminaire with an ambient light sensor may be configured to turn on at dusk and off at dawn and to turn on and off an indoor entryway luminaire.

Messages and Messaging

There are two categories of messages in a BLE Mesh network: Access messages for implementing an application and Control messages to manage the operation of the mesh network. Access messages are particularly important to product design engineers, as these are the means for requesting, sending, or changing the state values of elements; for example, turning on or off a luminaire.

The three types of Access messages are GET, SET, and STATUS. GET messages are sent to request state values from elements or groups of elements, which send STATUS messages in response. SET messages are sent to change state values in elements or groups of elements, which normally acknowledge the change by sending a STATUS message. SETs can also be unacknowledged, in which case no STATUS message is sent in response to the change. In addition to GET and SET responses, STATUS messages can be initiated by elements to periodically report their state value(s).

Communications occur within a BLE Mesh network as a managed flood of messages. "Flood" conveys how messages flow throughout the entire mesh topology while being "managed" to ensure efficient and effective use of available bandwidth. Key to a managed flood is the publish/subscribe group messaging used. Any node can publish or send a message, and every node is configured to subscribe to or act on only certain messages it receives, with all others being relayed as needed. The combination of these two aspects help give the BLE Mesh network its industry-leading price/performance, scalability, and dependability.

Device Provisioning

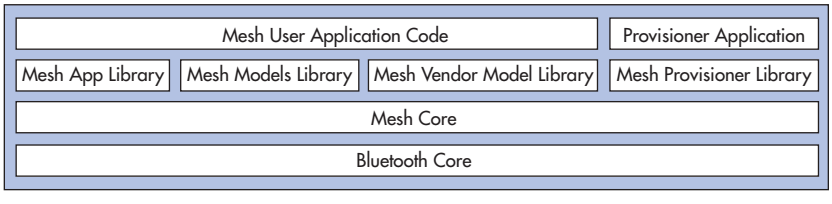
All devices installed must be provisioned before they can join a BLE Mesh network. The provisioning is normally performed by an application running on a smartphone, tablet, or PC. This is a distinct advantage of BLE Mesh because a Provisioner Application and, optionally, Mesh User Application Code *(Fig. 3)* can be run from a mesh participant. Provisioning is a deterministic and secure process that involves the exchange of keys for mutual authentication.

Mesh-Network Security

The BLE Mesh protocols were designed with robust security provisions. Provisioning, authentication, and messaging are subject to strong encryption. Network, application, and device security can all be applied separately using different keys, which provides the means to have multiple entities managing the different elements. These provisions protect against a full spectrum of physical and virtual attacks, including brute force, replay, man-in-the-middle, and trashcan, and provide for user-data privacy.

Software Architecture

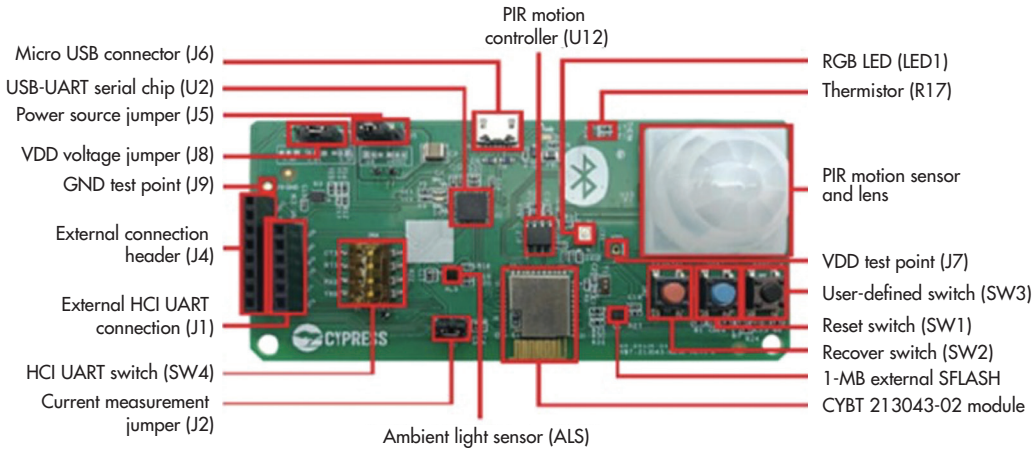
Figure 3 depicts the layers of software in BLE Mesh nodes. Note how the Mesh User and Provisioner Applications are located at the top of the architecture, above the libraries and core functions, to make them independent of the underlying BLE Mesh network. Note also how the Mesh Vendor Model Library makes it possible to add advanced, proprietary features while maintaining compatibility with the Bluetooth standards.



3. The layered architecture enables software engineers to focus their development efforts exclusively on the Mesh User and Provisioner applications and not the BLE Mesh network.

Developing BLE Mesh Products

This section highlights the BLE Mesh development process, including the tools used, and explores key design considerations. While BLE Mesh networks are suitable for myriad applications and use cases, smart lighting is used here as an example for two reasons. One is that the standards were



4. Shown here is an evaluation board used for prototyping and developing smart-lighting products. Not shown is the battery compartment on the back. (Source: Cypress Semiconductor)

established with lighting as a basic use case. The other is the popularity of such "smart building" applications that are forecast to experience a compound annual growth rate (CAGR) of 46% in unit shipments through 2023, according to research conducted for the Bluetooth SIG.

The basic development process begins by defining the device's configuration, which must include, at a minimum, its node type, element(s), model, and all hardware- and/or application-specific states and callback functions. The next step involves implementing the callback functions for both the application and the BLE Mesh network.

Various open-source and product-specific tools are normally used during the development process from beginning to end. A specific and more detailed example of the effort involved is available in the Getting Started with Bluetooth Mesh application note published by Cypress Semiconductor.

For products supporting BLE Mesh networks, design considerations involve, at a minimum, device functions (or elements), mesh-network size, antenna range, memory requirements, power consumption, and cost. Separate design consideration must be given to the provisioning, management, and other software that runs on a smartphone, tablet, or PC.

As is common in all product development efforts, tradeoffs are often needed among the various design considerations. The need for such tradeoffs will be addressed here in the context of three products commonly used in smart-lighting applications: switches, sensors, and luminaires. As will be shown, the design consideration at the center of most tradeoffs is power consumption.

To maximize the versatility and, therefore, the benefit of smart lighting, users will want to be able to deploy switches and sensors virtually anywhere. Because some devices might be deployed in locations that aren't easily accessible, the use of energy harvesting with rechargeable batteries might be a desirable feature. This is especially true for sensors that measure illumination or sense the presence/absence of occupants. Physical switches, by contrast, are by their very nature readily accessible and can, therefore, be designed with replaceable primary batteries when located where ac power isn't available.

For BLE Mesh networks, it's advantageous for battery-powered devices to be designed as Low Power Nodes, which depend on the availability of Friend Node functionality. This requirement can be noted in the product's documentation and/or provided in another product in a family, such as a luminaire, which is assured to have an external power source.

Because a major advantage of BLE Mesh networks is their scalability, they can grow quite large. Though the maximum number of nodes and a high number of hops are rarely limiting factors, products should be designed to work in small-scale deployments that need to span a large area, potentially outdoors, with relatively few nodes. In these use cases, it may be necessary to facilitate increasing a product's antenna range and/or providing Relay Node functionality in a separate (and optionally dedicated) product.

Power consumption is inextricably linked to transmit range, and the BLE Mesh standards give design engineers some powerful (pun intended) capabilities to make the desired tradeoffs that might be needed. One such capability is being able to expand the range without increasing power consumption by decreasing the bandwidth. The converse capability is also possible; that is, boosting the bandwidth by decreasing the range, again while not increasing the power consumption.

Other aspects of a product can elevate the importance of its power consumption. For example, how sensitive does a sensor need to be, and how frequently does it need to poll for status changes? More frequent communication means more power consumption, heightening the need for a larger primary battery or energy harvesting for a rechargeable battery.

Figure 4 shows an example of an evaluation board that can be used for product prototyping and development. Note the inclusion of three features commonly needed in smart-lighting applications: LEDs for luminaires, switches, and a PIR motion detector for use in an occupancy sensor. The module

at the center of the board's bottom edge contains the CPU, memory, and antenna required for the BLE Mesh network, as well as for running the application software.

Given the ubiquity of Bluetooth in smartphones, tablets, and PCs, these systems are normally used for provisioning, configuring, and managing BLE Mesh products, such as the switches, sensors, and luminaires utilized in smart-lighting applications. As noted above, because the Mesh User and Provisioner Applications are layered above the BLE Mesh libraries and core functions, software developers can focus their efforts exclusively on the application and not the network. Here's a sampling of some of the functions Mesh User and Provisioner Applications might need to support:

- Create and delete BLE Mesh networks and groups
- Provision and remove individual nodes
- Configure publications and subscriptions
- Publish GET messages to query the states of elements
- Publish SET messages, which for lighting applications might include On/Off, Level, Lightness and Lightness Hue, Saturation, Lightness Color Temperature, and Delta UV
- Publish SET messages for vendor data and vendor models
- Perform over-the-air (OTA) firmware upgrades

Choosing BLE Mesh Components

One additional design consideration not covered in the previous section is cost, which is always an important factor in the development of any product. Cost always has two dimensions: designing the product and manufacturing it. Choosing the most cost-effective BLE Mesh components also has two dimensions: the silicon and its software. The silicon is the system-on-chip (SoC) or system-in-package (SiP) modules, and the software is the development tools that accompany them.

With interoperability as the raison d'être for standards, the fundamental requirement when selecting BLE Mesh components is certified compliance with the Bluetooth SIG standards. This applies equally to both the silicon and the software, including the full Bluetooth BR/EDR and BLE Mesh protocol stacks and all pertinent libraries. Using SoCs, SiPs, and other components certified by the Bluetooth SIG eliminates the need for design engineers to conduct rigorous qualification and interoperability tests.

As of this writing, certification is available for version 5.0 of the Bluetooth core specifications and for version 1.0 of the BLE Mesh specifications. Version 2.0 of the BLE Mesh specifications are expected to be published in 2020.

When choosing a BLE Mesh platform, designer engineers should seek a solution that meets most or all of the following criteria:

- A family of modules to accommodate different needs—from the basic battery-powered sensor to the most sophisticated devices that might be needed now and in the foreseeable future.
- Fully integrated modules that minimize the need for external components, accelerate time-to-market, and reduce development and manufacturing costs.
- Ultra-low-power designs with the types of antennae and transmit powers needed to accommodate all anticipated node-to-node distances.
- Adequate CPU, memory (flash and RAM), and I/O for all foreseeable applications and upgrades.

For applications in which the BLE Mesh network may need to communicate with a Wi-Fi network, some form of gateway functionality will be required. For example, a home security system could be used to turn on or off certain lights to simulate people being at home or be deactivated with an authorized code. In these situations, an SoC or SiP combo-module that supports both Wi-Fi and BLE networking simplifies the design effort.

Because the software-development environment, tools, and libraries are just as important as the silicon, endeavor to find a solution that meets most or all of the following criteria:

- An easy-to-use integrated development environment (IDE) that abstracts the complexity of the underlying protocols
- Software development kits (SDKs), sample software, prototyping hardware, and a developer community to assist in software development and testing efforts
- Reference software designs for control applications running on the Android, iOS, Linux, and Windows operating systems

Bluetooth is already ubiquitous in "personal area networking" applications, and the advent of BLE Mesh significantly expands both the scale and scope of potential use cases for this popular protocol. The combination of virtually unlimited scalability and high resiliency afforded by BLE Mesh networking now enables Bluetooth applications to span buildings, campuses, and even entire cities. While the smart-lighting application used here serves as a good example, the potential use cases for BLE Mesh are limited only by the imagination.

Its ubiquity also gives Bluetooth another important advantage: the availability of proven techniques, tools, and software needed to develop new products. Hardware and software engineers interested in learning more about these resources are encouraged to review the documents and links listed in the References section below.

The many resources now available make it easy to get started with BLE Mesh. Proof-of-concept designs can be created quickly and affordably using inexpensive evaluation boards, IDEs, and SDKs. A solid head start may also be available in open-source or vendor-supplied sample application software. As a leading supplier of Bluetooth-certified silicon and software, Cypress Semiconductor provides an array of resources

at www.cypress.com/products/ble-bluetooth.

By offering enormous upside potential with minimal downside risk and a short time-to-market, BLE Mesh is destined to become the network of choice for a growing number of residential, commercial, and industrial applications.

Brian Bedrosian is Vice President of Marketing for the IoT Compute and Wireless Business Unit at Cypress Semiconductor, where he is responsible for strategic marketing and business development for the company's MCU, Wi-Fi, and Bluetooth business lines. Previously he was Sr. Director of Broadcom's IoT business, where he led the company's emerging embedded Wi-Fi and Bluetooth Smart business in audio, healthcare, smart home, consumer appliances and industrial systems.

Brian has 20 years of experience in digital communications and wireless technologies, including mobile satellite, microwave radio, DSL, and 802.11x. He has held positions at LevelOne Communications, Applied Signal Technology, and Globespan-Virata, and received an Electrical Engineering degree from UC Davis.

References

Bluetooth Mesh Networking - An Introduction for Developers

Bluetooth-Mesh-Paving-the-Way-for-Smart-Lighting

BLE Mesh specifications: https://www.bluetooth.com/specifications/mesh-specifications

Getting Started with Bluetooth Mesh (AppNote AN227069)