Features and Benefits of Bluetooth Connectivity

SILICON LABS

FOREWORD

Thanks to all of the home-automation–related smart gadgets that we use in our daily lives and incorporate within our homes and workplaces, our world is indeed becoming “smarter.” But with all of these convenient, fancy and “must-have” smart gadgets, we are seeing a number of competing wireless network technologies. As a result, the ultimate success of a functional and easy-to-use multiprotocol wireless, smart-home ecosystem hinges on one key parameter: interoperability.

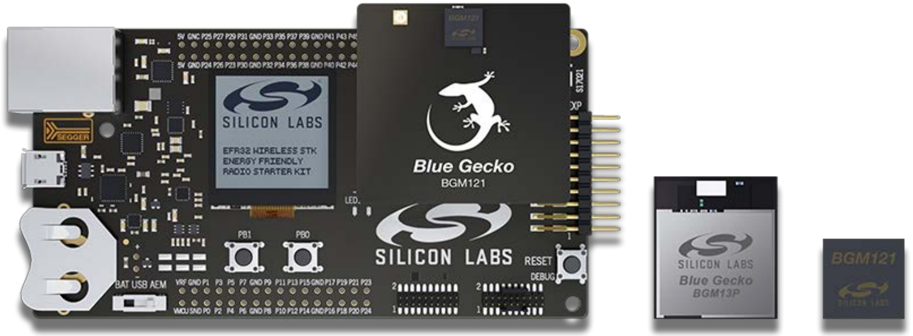
In this e-book, we discuss a variety of Bluetooth technologies, make a case for where and why they should be used within our smart homes and highlight some suppliers of multiprotocol modules and ICs and device makers that use them.

Bluetooth is already at the heart of many digital products including audio, video, input accessories and file sharing. Bluetooth home automation is often chosen because of how little power it uses. For those who are concerned about energy consumption and carbon footprint, low-power protocols like Bluetooth are a great fit. Bluetooth is already included in so many devices that it’s one of the fastest-growing sectors of home automation.

This e-book explores all of the different types, features and use cases for Bluetooth connectivity.

Table of Contents

Bluetooth 5



Mesh range, speed and bandwidth are the core improvements introduced in Bluetooth 5. Doubling the speed and quadrupling the range of Bluetooth 4, Bluetooth 5 has added many additional improvements over previous versions. Data length extensions (DLE) allow for eight times the previous message length. These new enhancements of Bluetooth 5 focus on increasing functionality for home automation and the IoT. Two new features for Bluetooth 5 deserve a closer look. ■

Learn more about the new features of Bluetooth 5 technology and how they enable the next wave of IoT applications read this whitepaper by Silicon Labs.

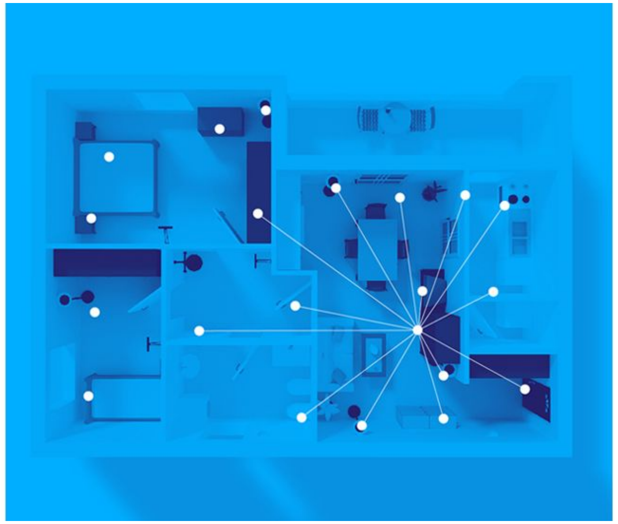
The highlights of Bluetooth mesh networking technology

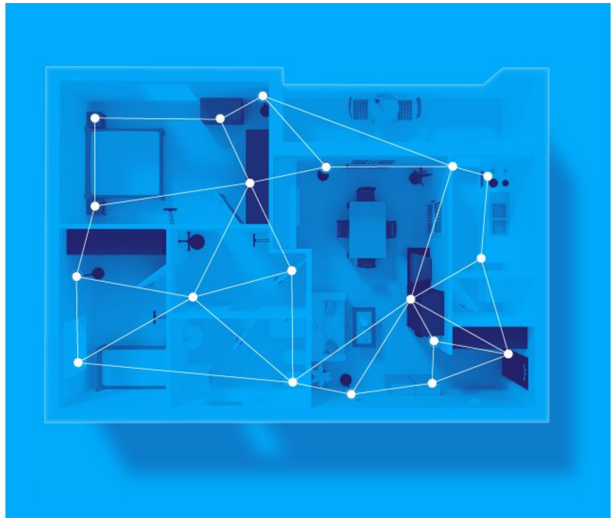
Bluetooth is one of today's most predominant low-power wireless technologies and is extremely familiar to wireless device users and developers. Prior to last summer, Bluetooth network typologies were confined to two-way communications between two devices, beaconing or a single hub and several satellite devices that could only communicate with that hub (star network). Though beaconing—a one-to-many broadcast-based technology used for point-of-interest information delivery—indoor positioning, asset tracking and two-way communications have created many possibilities for Bluetooth IoT use, the latest Bluetooth updates have evolved to the next generation of network architectures. Last summer, the Bluetooth Low Energy (LE) standard was upgraded with the option of mesh as a new network topology. Mesh networking enables a massive number of Bluetooth nodes to operate as a single large network, enabling new application sectors and use cases.

The key to Bluetooth mesh is the ability for each node to act as a viable operator in a network and also have the ability to relay messages from other nodes in the network. With one, or several, nodes connected to the internet, a Bluetooth mesh network can then seamlessly become an internet of thing (IoT) network that is readily scalable and comparably straightforward to develop.

Where Bluetooth Mesh Shines

In a traditional star network, every satellite node must be connected to a central node. Though suited for well-defined, static and small-sized networks, the limitations of this network topology make scaling, development, mobile and dynamic networks infeasible due to cost and design challenges. Another limitation of star network topologies is also range, as each satellite node must be able to reach back to the hub node in order for a star network to function. With mesh, however, multi-hop communications allow any viable node in range to retransmit a node message to whichever node is the desired destination. This is where Bluetooth mesh shines, as a mesh network topology can be designed to scale to massive numbers—currently specified to 32,000—of nodes and adapt to changing conditions of the network.





Star vs Mesh Network Topology for Bluetooth Devices

Star networks require a centralized hub to operate, where mesh networks can leverage nodes within the network to relay messages for enhanced scalability

Bluetooth Mesh Network Scalability and Reliability

With the IoT revolution happening in virtually every industry, many applications that previously relied on hard-line reliability but suffered from hard-line scalability issues are beginning to adopt wireless solutions. These applications benefit from low-power mesh technology, as the scalability of such systems is unprecedented compared to hard-line and traditional point-to-point wireless systems. What's more, mesh also doesn't suffer from the reliability issues that plagued previous point-to-point wireless systems.

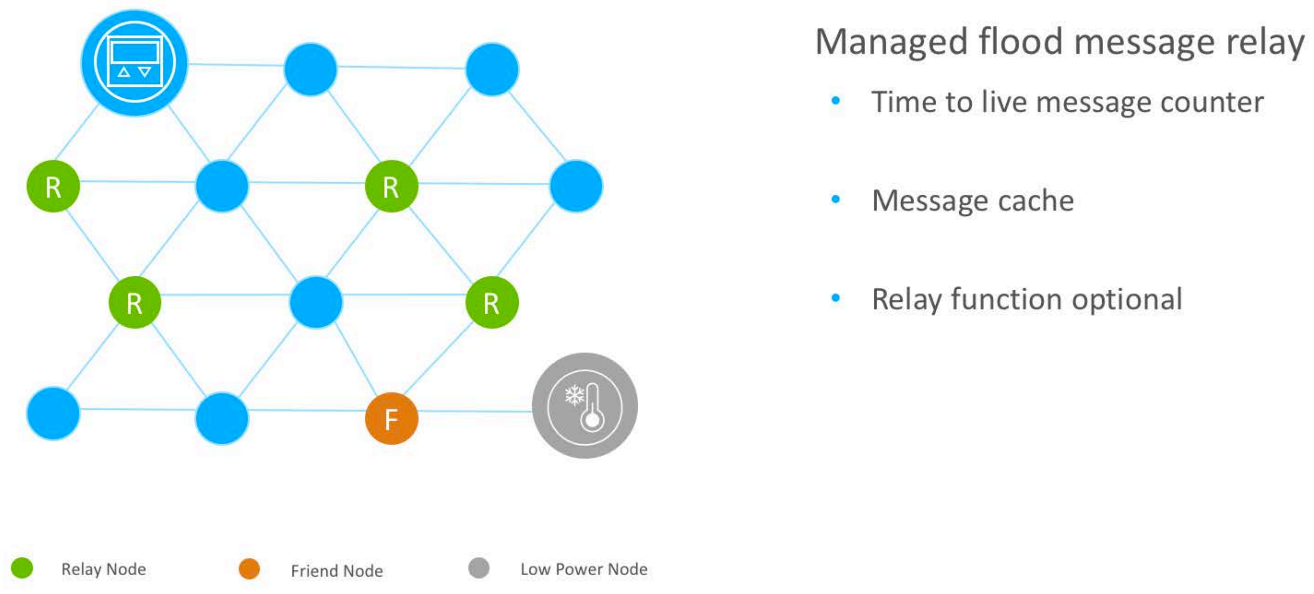
Scalability in a Bluetooth mesh is a matter of developers enabling a mesh network to grow based on the nodes that are provisioned to operate with that network. Provisioning can be done as a very secure multi-authorization approach, or made as simple as a user opting in with a push of a virtual smartphone button. Hence, mesh deployment can be done in one large single installation, or nodes can be added as the need arises for additional network coverage, sensor density or other scenarios that benefit from additional nodes. Moreover, through the Generic Attribute Profile (GATT), standard Bluetooth LE devices can communicate with a Bluetooth LE mesh network through proxy nodes, further allowing a mesh network to scale based on need.

Where point-to-point and hard-line network typologies typically revolve around a single hub, and liabilities exist for each connection to the hub and the hub's infrastructure, a mesh network topology can leverage each node in the network as a potential communication link and/or hub (relay node). This enables self-healing and routing optimization that can enhance the reliability and efficiency of a mesh network compared to a star network. This function is called multipath delivery, wherein a message can be transmitted from one node and received by the end recipient through a variety of paths through the mesh network.

Bluetooth Mesh Interoperability and Low-Power Beacons

Interoperability is a major concern with many new standards and technologies, as it may alienate or allow for partial obsolescence of still-viable prior technologies. However, with Bluetooth mesh, previous revisions of Bluetooth LE, such as those with smartphones and tablets, can still communicate with a Bluetooth mesh through the GATT. Proxy nodes are simply Bluetooth mesh nodes that have been approved to connect to other Bluetooth LE devices, enabling two-way communication with a Bluetooth mesh.

A significantly growing use case of Bluetooth LE is beaconing. Beaconing is when an event triggers a node to transmit information, which could include sensor information, location or point-of-interest information. There are two major beacon standards in the market: iBeacon from Apple and EddyStone from Google, each with individual benefits and use cases. Now, any Bluetooth mesh node could incorporate one or both beacon standards and could be turned into a virtual Bluetooth beacon while operating as a Bluetooth mesh node.



Star networks require a centralized hub to operate, where mesh networks can leverage nodes within the network to relay messages for enhanced scalability

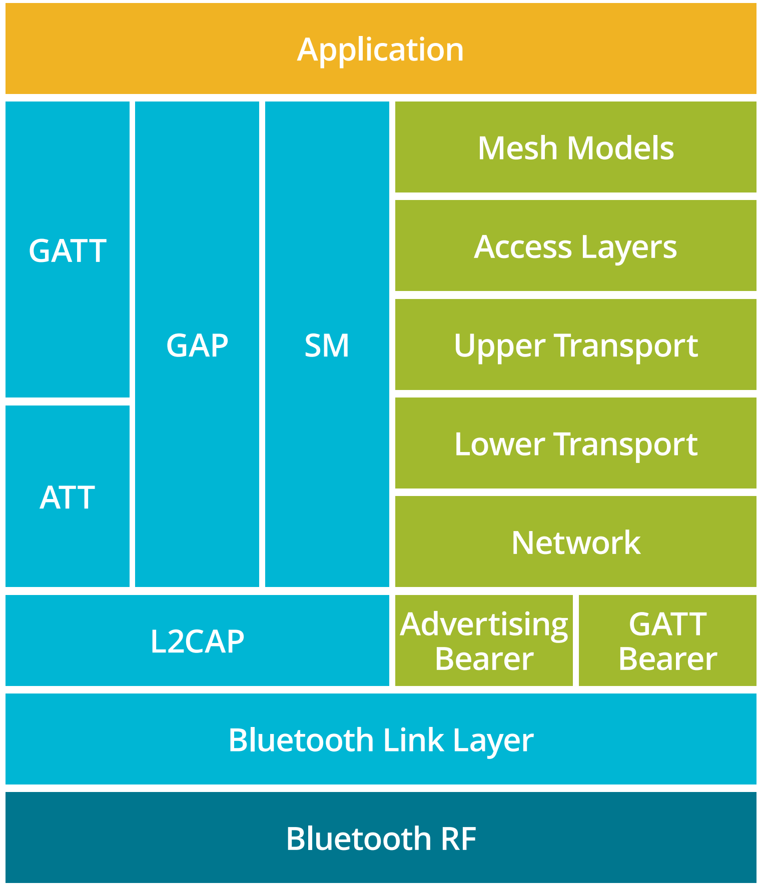
This can enable new use cases and even a business model such as, for example, connected indoor lighting vendors and now start to offer new services such as indoor positioning, asset tracking and point-of-interest information delivery.

Bluetooth Mesh Developer Benefits

Fortunately for developers that already have mature Bluetooth LE stacks and deployments, Bluetooth mesh is complementary to the established Bluetooth LE protocol. Using a supplier's Bluetooth Mesh SDK, a mesh network can rapidly be developed and optimized for a given application, which is readily done by a developer familiar with previous Bluetooth revisions and SDKs. Moreover, previous technologies can communicate with new Bluetooth mesh devices via the GATT, meaning that only technologies that require the mesh capability need to require mesh development. Lastly, just like with many Bluetooth devices, a user device setup and provisioning can all easily be enabled through a smartphone/tablet and app interface.

Bluetooth Mesh and Home Automation

Smart light bulbs, thermostats, vents, motion sensors, and many other smart home IoT applications are gaining in popularity. Typically, these home automation devices use a variety of different wireless standards and technologies and require various interconnected hubs and network routing throughout a home. Bluetooth mesh is promising as a protocol to simplify the smart



The versatility of the Bluetooth stack enables Bluetooth mesh networks to communicate with Bluetooth LE devices, such as a user's/customer's smartphone/tablet, through proxy nodes in the mesh network.

home automation process by enabling a growing mesh network of devices readily set up and provisioned with a user's smartphone or tablet.

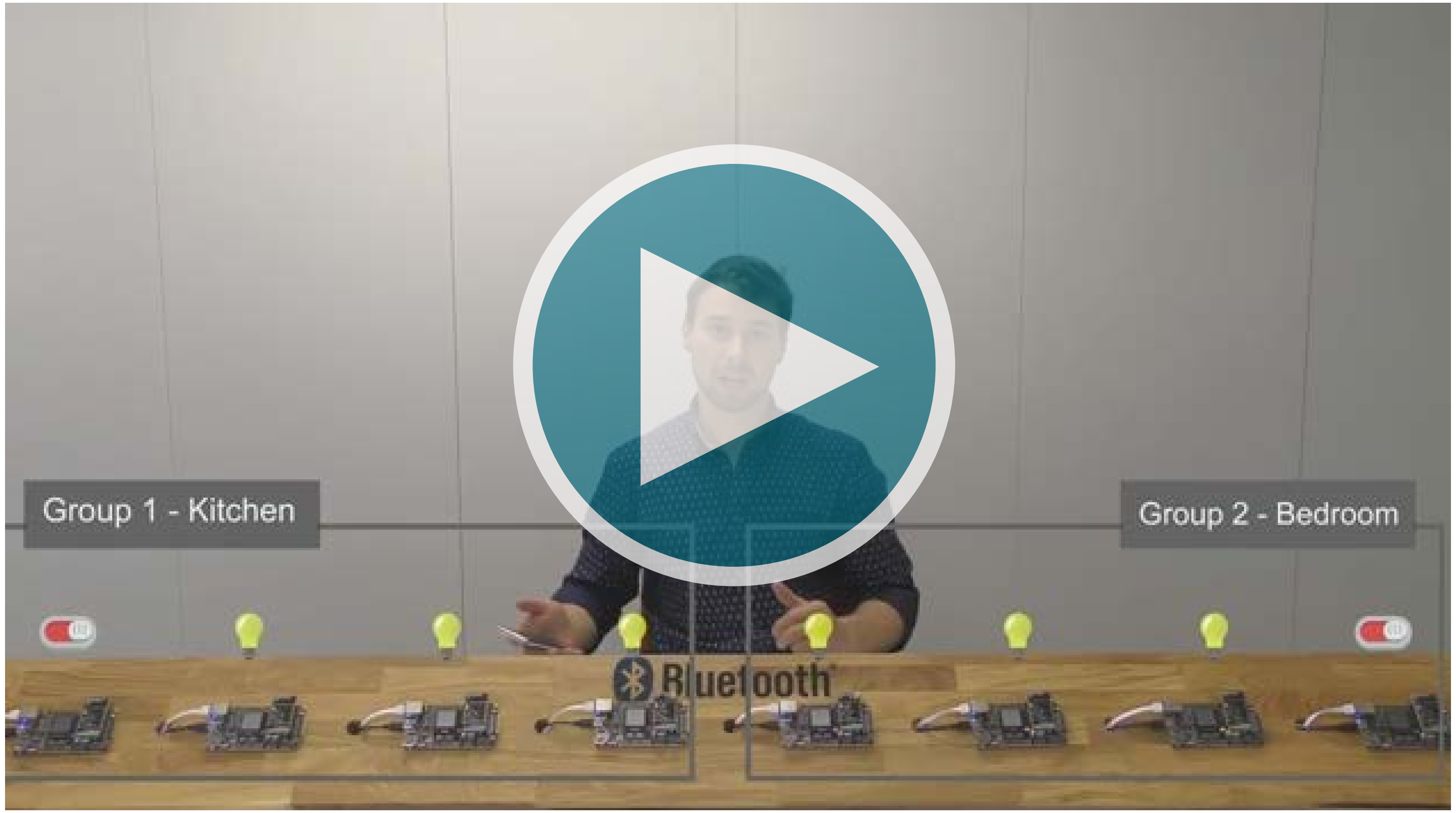
The benefit to a user is that a mesh network of connected devices could be used to multipath relay messages, which simultaneously increases reliability and network coverage. With just a few nodes throughout the house, a Bluetooth mesh could reach every corner of even a large home and prevent outages of service if a single node fails. For example, if a few of the nodes in a home lighting Bluetooth mesh are connected to a home network, even if one of the home-network-connected nodes fails, the messages from the rest of the mesh could still reach the user's controlling/reporting device. Traditionally, if the hub fails, the whole network is unusable. This could quite literally be the difference between lights on and lights off.

Bluetooth Mesh in Retail, Smart Building and Warehouse Automation Applications

Many commercial and industrial environments are benefiting from IoT augmentation. Some of these applications include asset tracking, beaconing for location services, advanced lighting control, environmental monitoring, patient monitoring and other automation functions. What Bluetooth mesh is capable of doing is simplifying the deployment of a multi-function network capable of massive scaling without modifying the underlying infrastructure of the network. This is comparably powerful potential, given that many wireless communication technologies require complex development or separate networks to perform multiple functions.

For example, a single Bluetooth mesh network could support enhanced customer information and interaction services and asset tracking beacons in a retail store's warehouse while simultaneously supporting the asset tracking and beacons reporting the status of a store shelf. Such an integrated system could go beyond simple reporting of low shelf stock and could actually be used to automatically direct an employee to the exact location of the item in the warehouse and back to the shelf or to the location of a potential customer with the item of interest identified on their smartphone.

Bluetooth mesh can also be used in low-latency scenarios. Current Silicon Labs benchmarks demonstrate that a Bluetooth mesh solution can achieve a latency below 10 milliseconds per hop, with single packet payload that is up to 11 bytes of data (test data has demonstrated 6 hop round-trip latencies at about 110 ms, or 55-ms one-way latency). Naturally, as the network or payload size increases, the latency will also increase. Fortunately for some applications, the network size has a smaller impact on latency than the payload size. This means



that optimizing a particular network with functional nodes and relay nodes can be used to reduce the latency of a network while maintaining coverage and reliability. Silicon Labs benchmarks also show that large Bluetooth mesh can perform well. In a 240-node network test by Silicon Labs, with proper relay selection in place, the mesh achieved above 99% reliability and 98% of the packets arriving at the nodes in 60 milliseconds or less.

Conclusion

Though Bluetooth mesh has only recently emerged with chipsets and SDKs from manufacturers and will likely benefit from future enhancements to the standard, mesh networking with the Bluetooth standard is poised to offer many unique features to home, commercial and industrial automation. These benefits surround Bluetooth mesh's capability of easily developing and deploying a scalable mesh network that is both extremely reliable and can be readily leveraged to perform multiple tasks. This capability is further enhanced as a Bluetooth mesh network can be composed of nodes with a variety of functions, such as friend nodes supporting low-power nodes and proxy nodes supporting intercommunication between a Bluetooth mesh and a non-mesh Bluetooth LE device. Given the flexibility and ease of development, it will be exciting to see what solutions developers are able to achieve with the latest in low-power mesh networking technology. ■

Blue Gecko Bluetooth Smart Modules (BGM13P)

- Pre-certified module with integrated crystals, RF passives and antenna enable fast time to market
- Ideal for battery powered applications
- Full support for Bluetooth 5 (2M PHY and LE Long Range) and Bluetooth Mesh

» SHOP NOW

The Silicon Labs EFR32BG Bluetooth development kit is the easiest and fastest way to start building your own BLE application. This kit is designed to speed evaluation time featuring powerful EFR32 Blue Gecko modules. Learn more.

Mesh Networking Wireless SoC for Zigbee & Thread (EFR32MG)

- Industry-leading energy efficiency and wake up times
- Supports Zigbee, Thread and Bluetooth 5
- Suited for applications requiring high performance

» SHOP NOW

Blue Gecko Bluetooth Low Energy SoC (EFR32BG)

- Full support for Bluetooth 5 (2M PHY and LE Long Range) and Bluetooth Mesh
- Ideal for battery powered applications
- Sub-GHz and Bluetooth LE can be run simultaneously on a single chip

» SHOP NOW

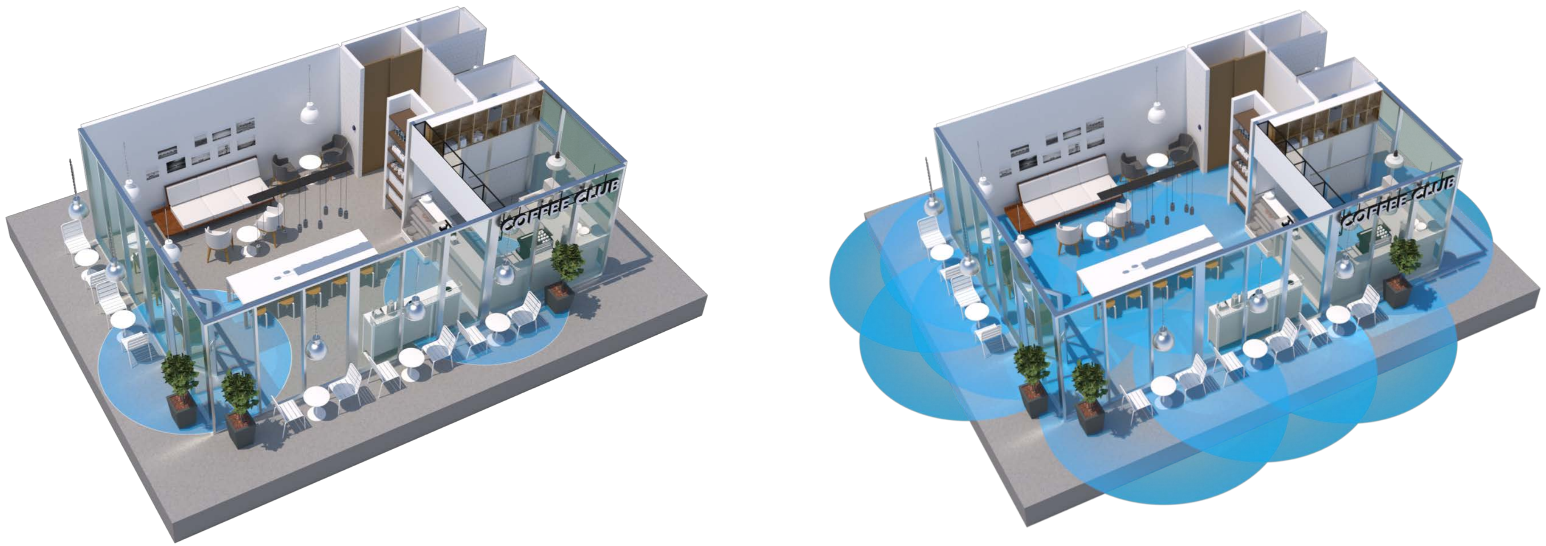
RESOURCES

1. http://pages.bluetooth.com/mesh.html
2. https://www.bluetooth.com/specifications/mesh-specifications?utm_campaign=mesh&utm_source=bluetooth&utm_medium=web&utm_term=paper&utm_content=paper-mesh-hdwr-pltfrm-link
3. https://www.silabs.com/products/wireless/learning-center/bluetooth/bluetooth-mesh
4. https://www.silabs.com/documents/login/application-notes/an1137-bluetooth-mesh-network-performance.pdf

Expand Device Capability with Multiprotocol Bluetooth and Zigbee Connectivity

Multiprotocol connectivity provides a unique approach to add functionality being requested by consumers and businesses. To provide the necessary scalability and robustness in home or building automation scenarios device-to-device communication over a mesh network is an ideal implementation. At the same time being able to setup, control, or monitor an individual device or a set of devices directly from a smartphone is a feature being requested to simplify consumer experience and provide more immediate diagnostic information to technicians for installation.

Value added services such as providing proximity based advertisements in retail settings, transmitting system health information for technicians, and tracking assets in warehouses can be delivered through connected devices such as lights. At the same time, there is a desire to participate in multiple ecosystems whether it's Alexa, Apple HomeKit or Google Home, each with their own protocols or integration requirements. By supporting multiple protocols on an individual device we are able to address a number of the needs we just discussed.



Improving beacon density with multifunction lights

Providing New Experiences with Multiple Wireless Protocols

Let's examine how the experience in home automation scenarios can be improved with devices that support multiple protocols. Zigbee provides whole home coverage with its mesh capabilities and can provide control from outside the home via a gateway. However, with multiprotocol support, we can further expand the use case to include the phone, which has Bluetooth low energy, for local control and location aware services. This video illustrates how a smartphone app can connect to a smart door lock over Bluetooth to unlock it.

By supporting both Bluetooth and Zigbee connectivity the door lock unlocks after receiving Bluetooth communication and is simultaneously able to send a Zigbee message to turn on the living room lights.

Using proximity aware services such as Bluetooth beacons when the smartphone is brought into the bedroom, the lights can send a Bluetooth beacon message to allow the consumer to turn on all or some of the lights in the room.

In retail or commercial settings, there is a desire to make use of technologies such as Bluetooth Beacons to provide location based advertisements, track assets and also develop heat maps of foot traffic. One of the challenges for wide scale adoption is the need for dedicated beacon devices. For device life cycle management, the range of connectivity also has impact on the logistics of updating devices.

By integrating Bluetooth beacons into other connected infrastructure such as lighting we can build out large scale and dense beacon coverage areas. Instead of having to deploy both connecting lights and beacons, a connected Light or Luminaire can also serve as the Bluetooth beacon. This can provide a more cost-effective means to improve beacon density than deploying separate dedicated Beacon devices with an added advantage of not requiring battery powered beacon devices that must be monitored and maintained.

Multiprotocol makes additional uses cases possible as well. For example, over-the-air updates can take a long time over a mesh network, but the higher throughput of Bluetooth can provide quicker transfer of update images without consuming the bandwidth of the mesh network.

Finding a Cost-Effective Means to Support Multiple Protocols

One of the challenges faced to deliver these improved experiences with support of multiple protocols has been the requirement to have multiple chips or SoCs, one for each protocol. With multiprotocol chips, devices now have the flexibility to run different protocols. The below table describes some common examples of multiprotocol devices

Multiprotocol Type	Description
Programmable	Devices are programmed to run one of a number of protocols in manufacturing or via updates in the field or over the air.
Switched	Devices change which wireless protocol is being used by bootloading a new firmware image when the device is already deployed in the field.
Dynamic	Devices simultaneously run multiple wireless protocols on one chip, using a time-slicing mechanism to share the radio
Multiradio	Devices achieve dedicated operation of multiple protocols without any trade-offs with multiple radios

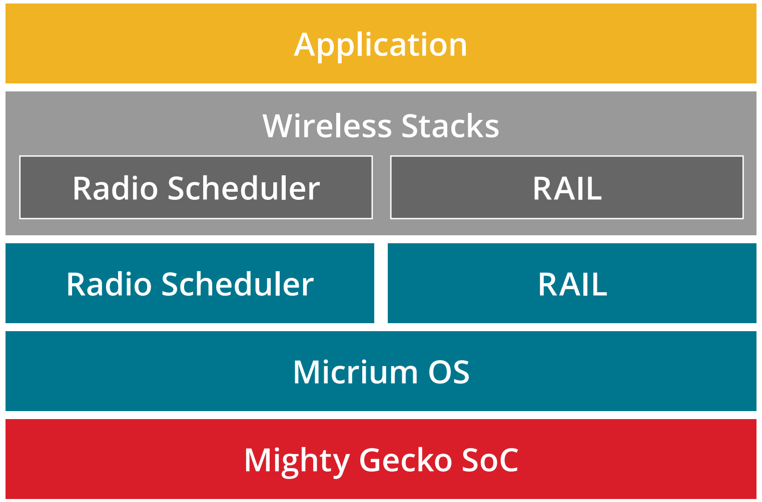
Table 1: Wireless Multiprotocol Scenarios

Single chip solutions that combine advances in software and hardware from companies like Silicon Labs enables devices to support both Zigbee and Bluetooth to address the use cases discussed so far. By using one SoC wireless sub-system BoM cost can be reduced by 40% by eliminating two radios and also simplify PCB design by eliminating the need to address the possible interference between two radios in a design.

Simultaneous Execution of Multiple Protocols on a Single Radio

Let's take a more detailed looked at how dynamic multiprotocol scheduling works to support multiple protocols with a single radio. A Zigbee router always has its radio in receive mode when not transmitting. This is so that other devices in the network can always send packets to it, or route through it. Because of the low duty cycle for Zigbee traffic and the retry mechanisms in the Zigbee networking stack, it is possible for the Zigbee Router to change its radio to another protocol for short periods without dropping any messages at an application level. This allows us to time-slice Zigbee and Bluetooth communication on the same chip. In addition to Zigbee routing, Silicon Labs dynamic multiprotocol technology supports Bluetooth connections, and Bluetooth beacons.

The protocol connection interval is configurable to match application requirements. For Bluetooth beacons, the radio only needs about 1ms to transmit a beacon and the connection interval between beacons is typically no shorter than 100ms. For high speed OTA firmware updates, the device will likely need to be configured to support much longer Bluetooth connection periods. These examples lie on opposite ends of the spectrum; however, with a configurable connection interval, the multiprotocol solution from Silicon Labs provides a flexible framework to meet the unique needs of different applications. To enable effective



multiprotocol communication Silicon Labs has made a number of investments in both software and hardware. Wireless protocol stacks from Silicon Labs are architected to share the same low-level radio drivers and libraries (RAIL). Making use of RAIL ensures a consistent API and interface to share the radio.

In addition, the Radio Scheduler manages the requests from protocols for access to the radio while the Micirum OS kernel manages the sharing of resources between the stacks. The multiprotocol scheduling from Silicon

Labs takes the protocols being scheduled into account and uses a priority based scheduling methodology. Bluetooth requires a fixed connection interval for effective operation while Zigbee, with its MAC retry approach is more forgiving. For this reason, for Zigbee and Bluetooth multiprotocol operation, Bluetooth operates at a higher priority. Because of the unified architecture of wireless stacks using RAIL, radio scheduler and Micirum OS the system is able to use a priority based scheduling approach to balance Zigbee and Bluetooth operation.

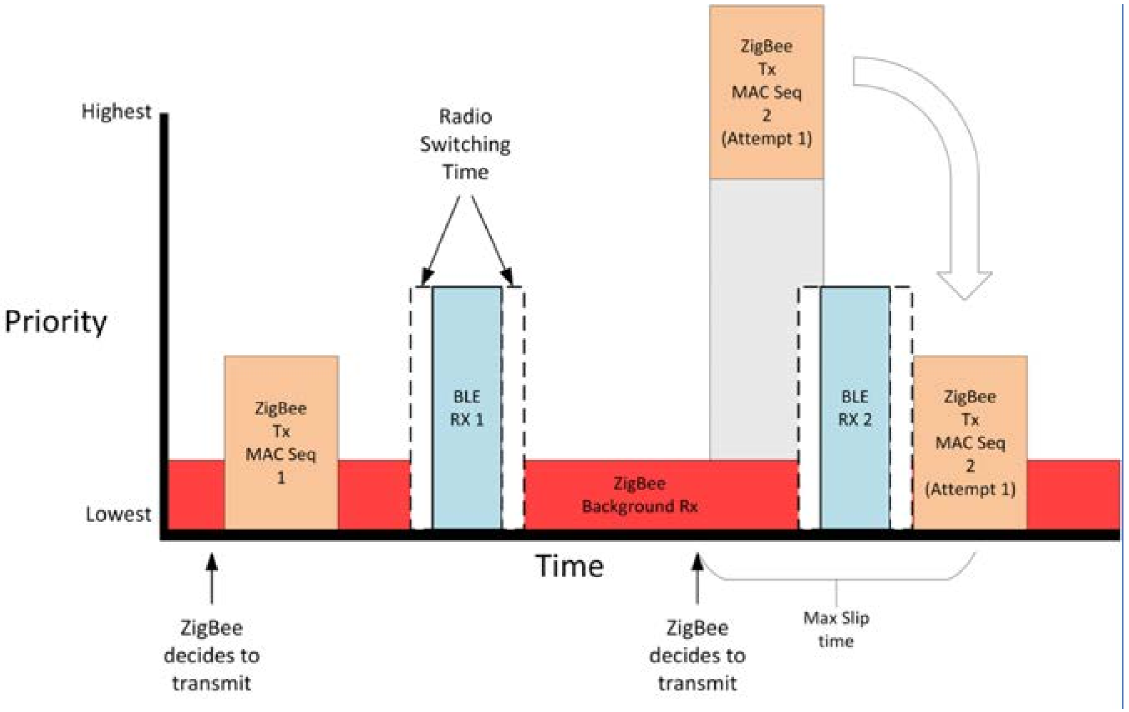


Figure 1: Zigbee background receive with Bluetooth LE having priority

Scheduling Requirements for Single Radio Zigbee and Bluetooth Operation

A number of scheduling scenarios may be required to enable the correct operation of Zigbee and Bluetooth with a single radio. The scheduler can be configured to make either protocol the higher priority with regards to radio access. However, the most likely configuration would be to make Bluetooth connections and beacons the higher priority and leave the radio in Zigbee receive mode when doing nothing else.

In Figure 1, we can see that the low-priority Zigbee receive is the default, but when a Zigbee transmission is required, it interrupts that process. This is normal behavior for a Zigbee device. When a Bluetooth LE connection is scheduled, this takes precedent, and the scheduler switches out of Zigbee receive mode in time to be available for the Bluetooth connection. If the scheduler has a request for a Zigbee transmission that would exceed the time available on the radio before the next Bluetooth connection or beacon, the scheduler will reschedule the Zigbee transmission to occur after the Bluetooth activity has completed.

If the transmission time for a Zigbee packet exceeds what was expected, perhaps due to backoffs or clear channel assessment,

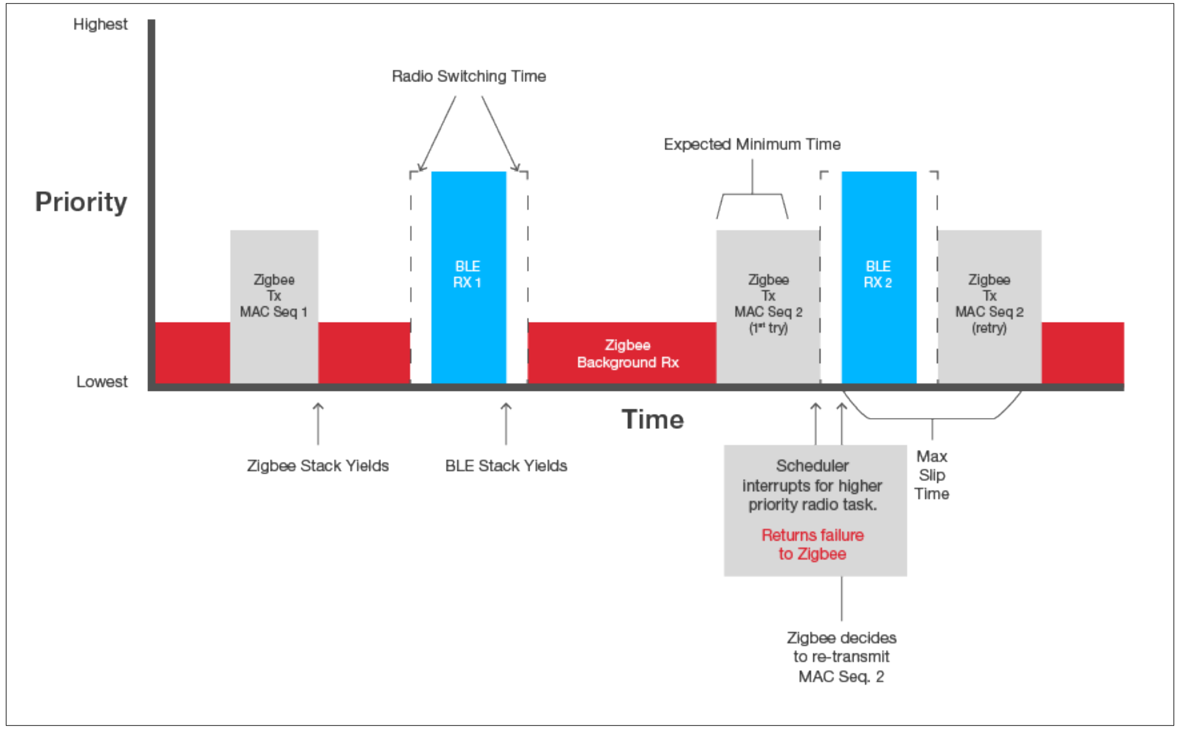


Figure 2: Bluetooth connection interrupts Zigbee transmission

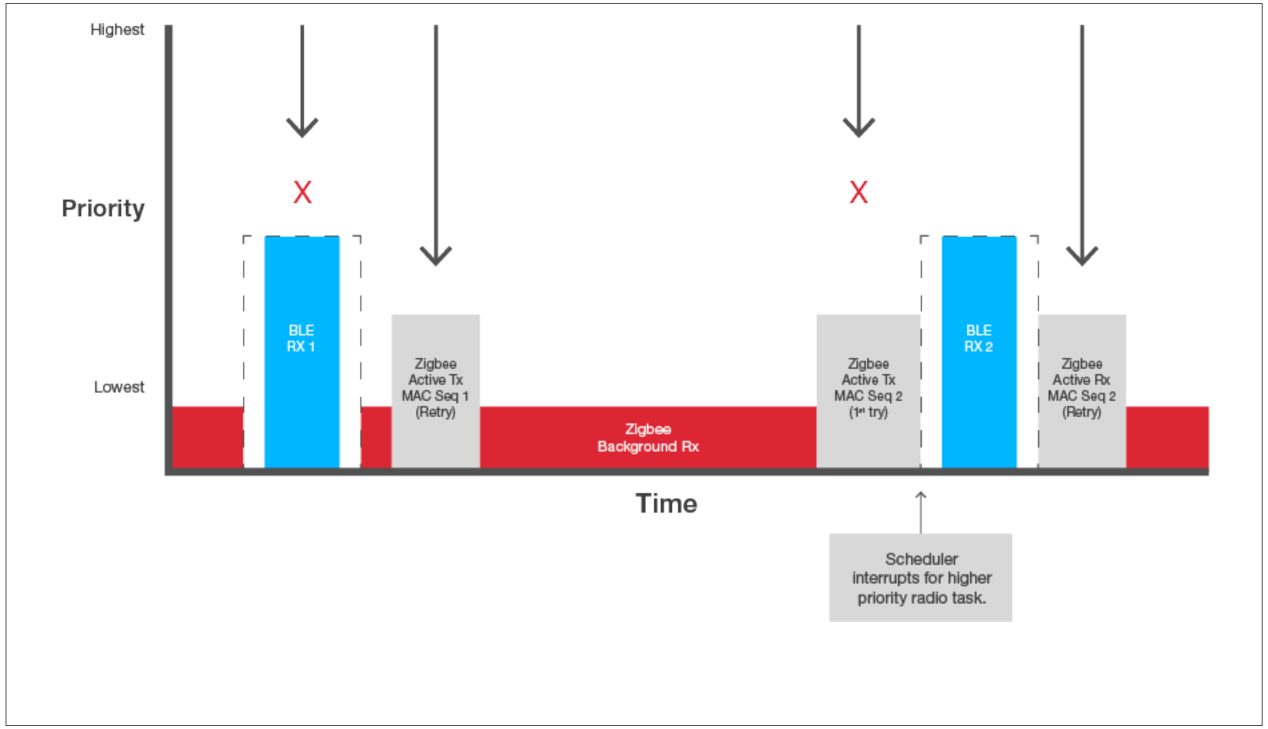


Figure 3

the scheduler can interrupt that transmission and switch to Bluetooth. To the Zigbee stack, this looks like a failed attempt, so it retries the transmission, and this time it succeeds, as shown in Figure 2.

Similarly, if a remote Zigbee node attempts to send a packet to the device while it is in the middle of a Bluetooth connection or beacon, the device cannot receive the packet, but the sending device will retry (IEEE 802.15.4 MAC retry), and the packet will be received on the second attempt. Also, if the device is in the middle of receiving a Zigbee packet when a Bluetooth connection or beacon is due, the scheduler can interrupt the packet reception, and the sending device will not receive an acknowledgement. As a result, it will retry the transmission and succeed on the second attempt. Figure 3. shows both scenarios.

The radio scheduler must handle a variety of scenarios to manage conflicts between wireless protocols, but the individual protocol stacks do not have any awareness of one another, only that they must request access to the radio and whether their transmission or reception has been successful.

For additional radio scheduling examples please refer to the Dynamic Multiprotocol Users Guide.

Evaluating Dynamic Multiprotocol Performance

In order to understand device behavior when running multiple protocols, it is important to measure and compare performance under multiple configurations. For the case of Zigbee and Bluetooth running on the same SoC and single radio the scenarios may include:

- Zigbee throughput vs Bluetooth connection(s) and/or advertisement intervals
- Zigbee latency vs Bluetooth connection(s) and/or advertisement intervals
- Zigbee throughput or latency vs varying Bluetooth packet types and sizes
- Zigbee retries and network behavior for varying Bluetooth connections and/or advertisements

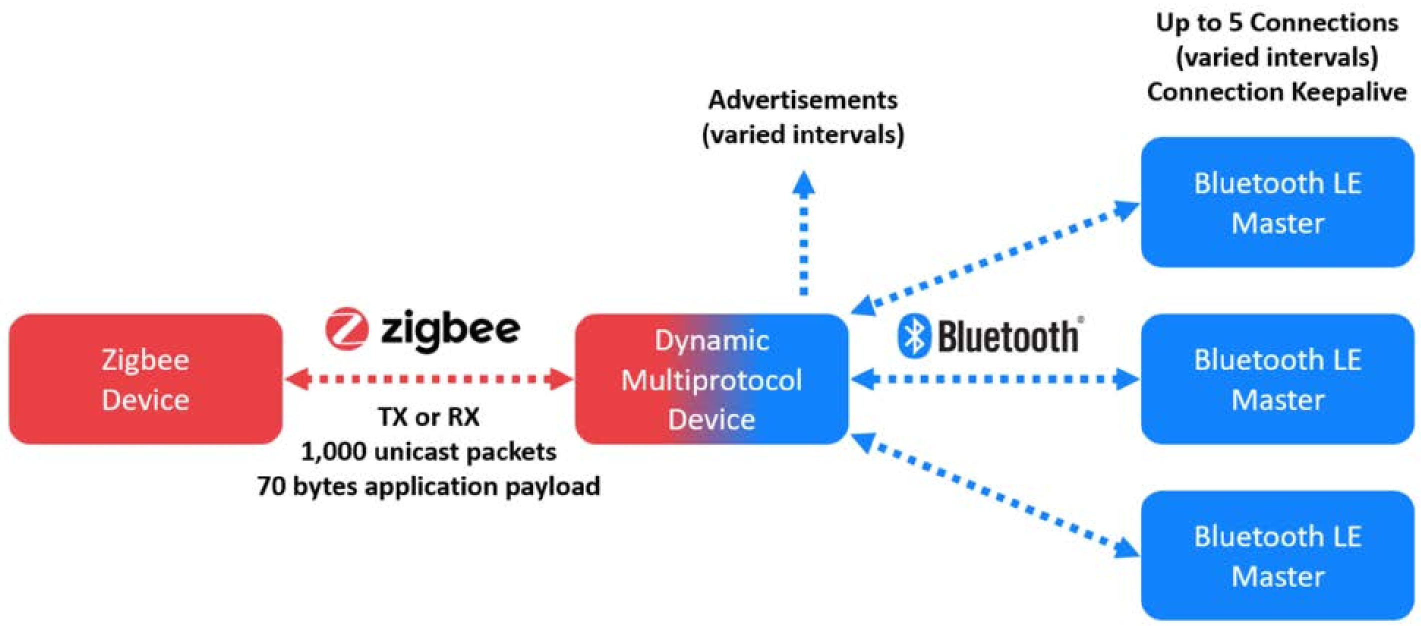
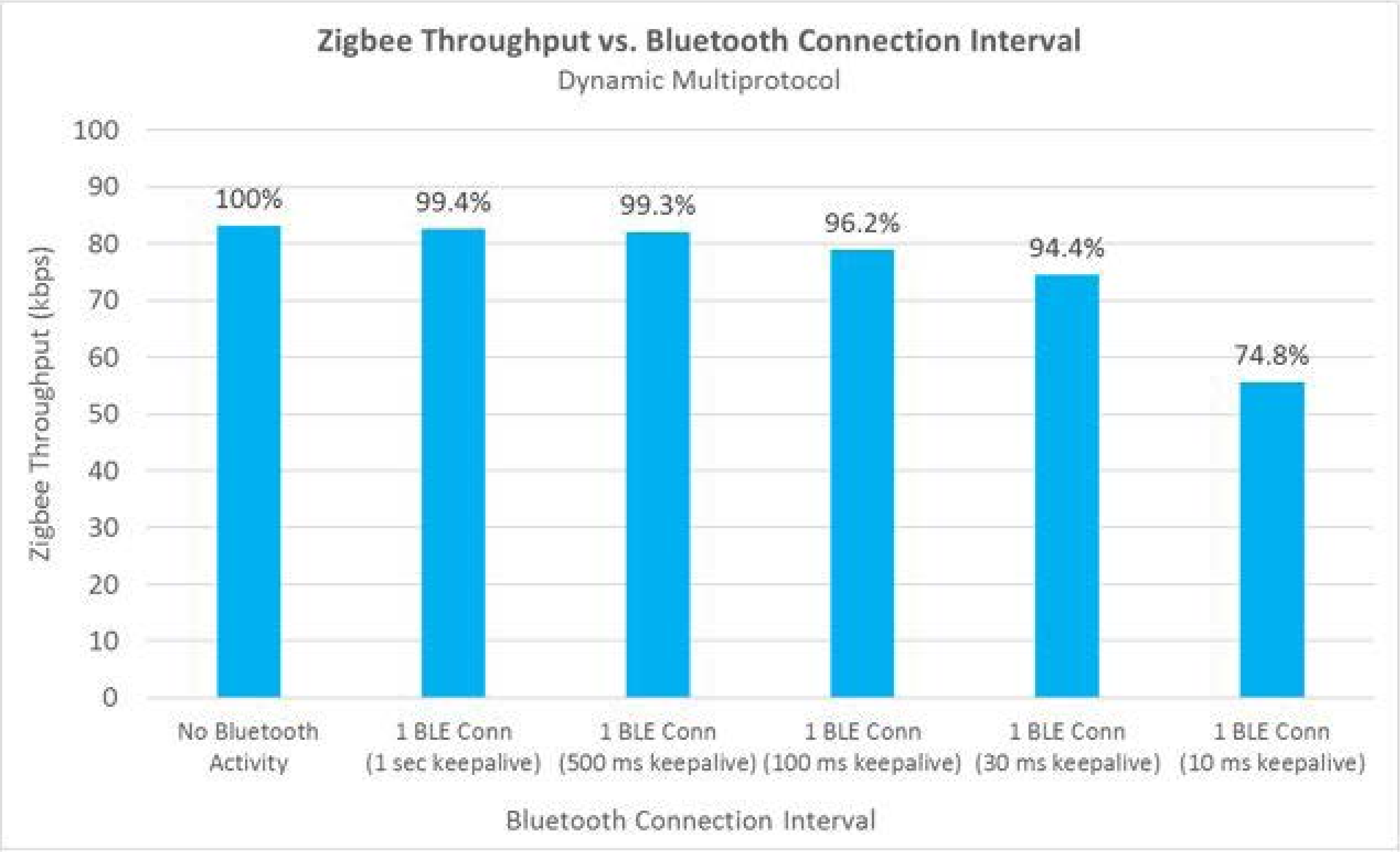


Figure 4: Dynamic Multiprotocol Test Setup

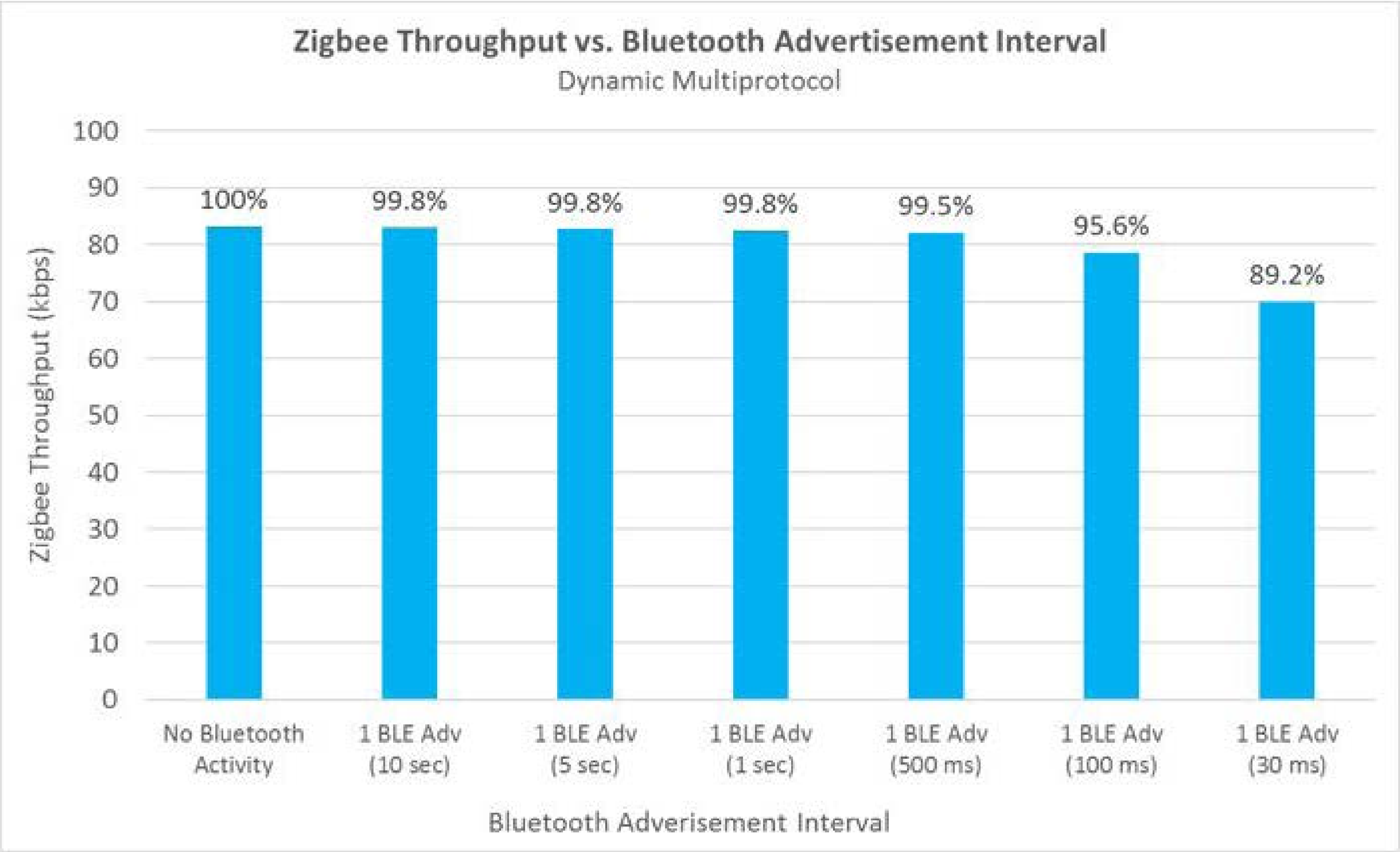
Using the test setup outlined in Figure 4 a sample test executed on a Silicon Labs Wireless Gecko STK board using a radiated test setup gives the following results:

For the results displayed 802.15.4 MAC and Zigbee NWK layer retries were enabled while Zigbee APS layer retries were not. The device was configured to transmit 70 bytes of payload across one hop while the Bluetooth connection was maintained with a keepalive



at the noted connection interval. As the Bluetooth connection interval is reduced, the number of Bluetooth connection events increase and the Zigbee throughput decreases due to the decreased radio time on the Zigbee network. Note that 100% end-to-end message reliability was achieved and, although throughput decreased due to longer data transfer, no Zigbee application messages were lost.

To verify the impact of advertisement intervals the device was configured to transmit Bluetooth advertisements at varying intervals instead of maintaining a Bluetooth connection. Since Bluetooth advertisements packets are larger than Bluetooth connection keepalives, they have a slightly higher impact on Zigbee throughput for the same time interval. Advertisement intervals as short as 0.5 seconds have little impact on Zigbee throughput and should meet the needs of most use cases.



Designing Systems with Multiprotocol Connectivity

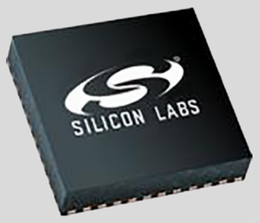
With dynamic multiprotocol hardware and software it is now feasible to combine the benefits of multiple protocols in cost effective manner on a single SoC. Home automation, asset tracking and retail advertising can benefit from combining Zigbee and Bluetooth connectivity on a device.

Each device and application has unique needs that require software configurability of items such as Bluetooth connection intervals. Before embarking on development is important to ensure the underlying software and hardware architecture is designed for effective resource sharing of the radio and enables advanced scheduling scenarios.
In addition, testing and performance benchmarks should be defined with the specific application and system use cases in mind to ensure proper operation in the field. ■

Practical Bluetooth Low Energy and Sub-GHz Integration Approaches

The adoption of Bluetooth technology in mesh and point-to-point networks shows no sign of fatigue. Meanwhile, sub-GHz radios continue to proliferate, particularly in smart home, commercial and industrial applications. In this article, we present practical multiband integration approaches for Bluetooth and sub-GHz technologies.

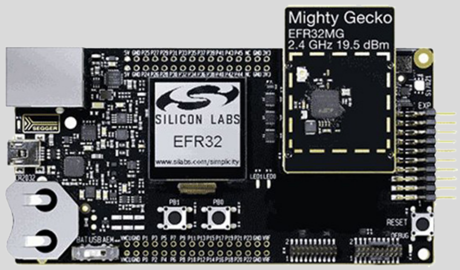
Blue Gecko Bluetooth Smart Modules (BGM13P)



- Pre-certified module with integrated crystals, RF passives and antenna enable fast time to market
- Ideal for battery powered applications
- Full support for Bluetooth 5 (2M PHY and LE Long Range) and Bluetooth Mesh

» SHOP NOW

Blue Gecko Bluetooth Low Energy SoC (EFR32BG)



- Full support for Bluetooth 5 (2M PHY and LE Long Range) and Bluetooth Mesh
- Ideal for battery powered applications
- Sub-GHz and Bluetooth LE can be run simultaneously on a single chip

» SHOP NOW

The Proliferation of Sub-GHz Devices

Wireless technologies functioning at frequencies below 1 GHz (sub-GHz) have been trending in popularity with the emergence of Low Power Wide Area Networks (LPWAN) and Wi-Fi standards such as HaLow (IEEE 802.11ah) and White-Fi (IEEE 802.11af). This trend comes from the inherent ability for signals to travel greater distances (> 1 km) with better object penetration and immunity to interference. This wider coverage range has broad implications in IoT for smart city, building and home automation, smart farming and industrial IoT (IIoT).

According to ON World, LPWAN services are set to reach $75 billion by 2025, serving more than 30 different applications in a broad range of market segments globally. And with more than 3 billion anticipated connected devices by 2021, it is no surprise the sub-GHz market is booming.

The Potential Hurdle of Relying on External Wi-Fi and Cellular Networks

While the number of IoT devices using the sub-GHz spectrum is growing, there is still a considerable barrier to the progress of these technologies—the lack of compatibility with existing devices including smartphones, tablets and laptops. Typically, sub-GHz IoT networks operate in a star topology where sensor nodes initiate transmissions with a gateway or base station that is often hardwired (e.g., Ethernet cable) for a high-throughput connection to network servers (See Figure 1).

Therefore, the only way to remotely control an end device or a gateway is often through a remote web-based app or Cli (Command-line interface) terminal that connects to a local Wi-Fi or cellular network. There may be scenarios where these networks are not always available or reliable. While this may not be an issue for some IoT applications, it is critical for industrial and commercial applications where the network is used to operate, maintain and upgrade equipment.

Benefits of a Multiprotocol (Bluetooth + Sub-GHz) Platform

For most LPWAN networks including LoRa, Sigfox, and NB-IoT, an additional radio is required to communicate with existing user equipment (e.g.: smartphone, tablet, etc.). Bluetooth Low Energy (LE) offers a platform where the device setup, operation and maintenancecan be performed locally with a smartphone or tablet. Moreover, firmware over-the-air (OTA) updates can be performed much more rapidly with a higher throughput Bluetooth LE 5 connection (~2 Mbps) than LPWAN narrowband (NB) or ultra-narrowband (UNB), which only supports modulation schemes from 1 kbps to 100 kbps.

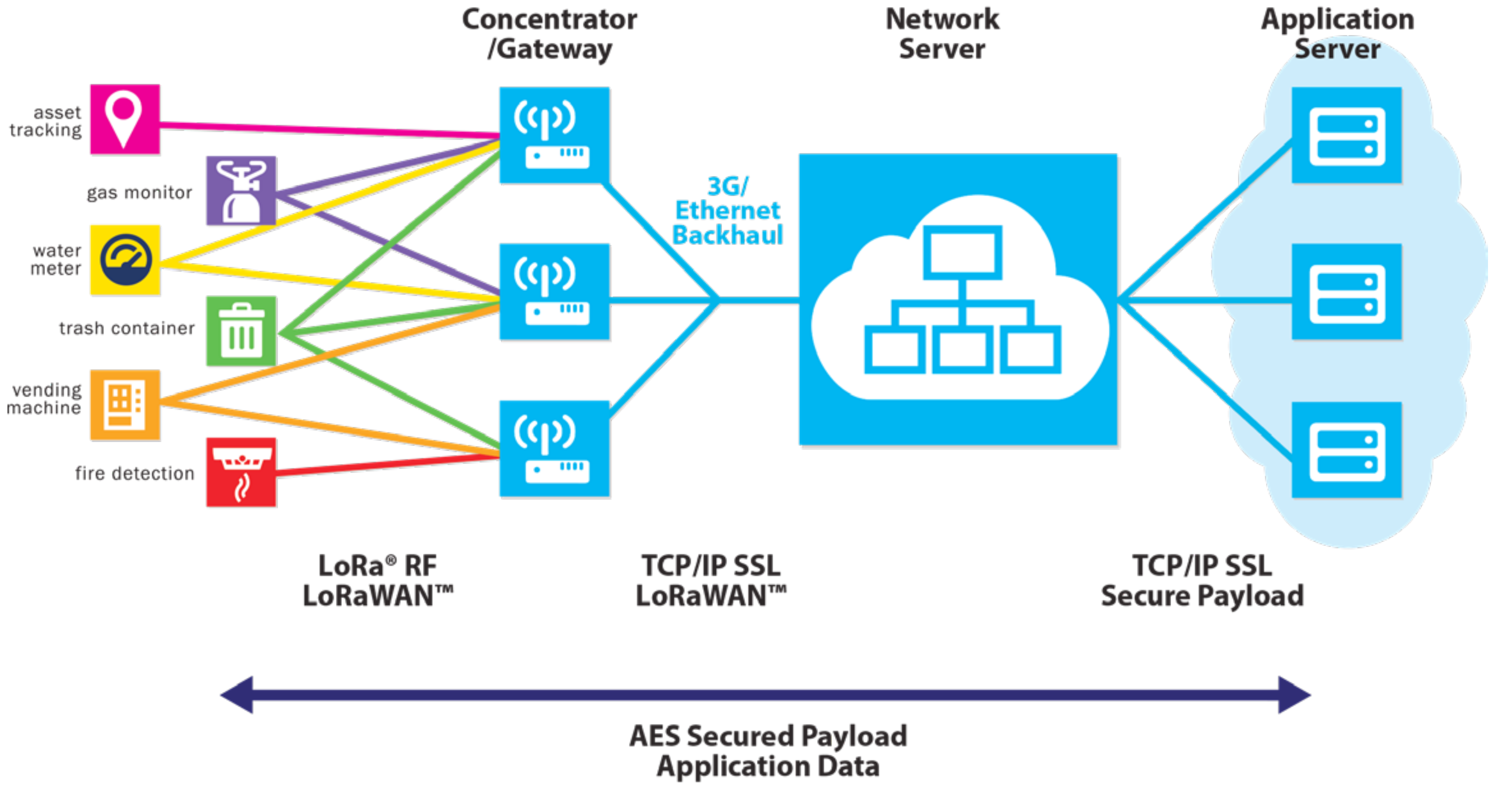


Figure 1: LPWAN technology LoRa network diagram with a star topology. [Credit: Semtech]

Multiprotocol SoCs for the Setup and Management of Smart Meters

Sub-GHz frequencies are often used for the dissemination of collected data from a system of tens to hundreds of smart meters, which could be in the form of a proprietary Field Area Network (FAN) or an open LPWAN platform. The FAN uses a star topology in a many-to-one communications architecture where multiple smart meters wirelessly link data to a local server connected to a gateway. The collected information is then sent over a Wide Area Network (WAN) to a geographically separated corporate office/ billings system. Any upgrades to the smart meters must then occur through the gateway over the internet or through a hardwired connection to the meter itself. Typically, OTA firmware upgrades are performed to add functionality, address technical or security related issues. Since these upgrades are critical to the long-term functionality of smart meter equipment, utility companies must be able to plan around the failure rate of OTA upgrades based upon traffic, interference and a number of other parameters.

A smart meter equipped with an SoC that can function in both sub-GHz frequencies and Bluetooth LE (2.4 GHz) can provide both long-range and local short-range wireless communication. By adding Bluetooth connectivity, utility technicians can now use a mobile app to control, communicate or update smart meters in the field. This type of redundancy built into the system can potentially reduce the operational overhead of utility companies by having multiple backup plans to perform wireless maintenance of smart meters. This technology can also cross industry verticals and be applied to agricultural and industrial applications with industrial area meters to monitor parameters such as flow rate (See Figure 2).

Applications for Bluetooth Beacons and Sub-GHz

Bluetooth beacons recently have become the leading proximity sensing technology and accounted for nearly 80 percent of the proximity sensors deployed globally in 2015. Bluetooth beacon shipments are expected to surpass 400 million units by 2020. This exponential growth rate can be attributed to a wide array applications, including:

- Indoor navigation
- Contactless payment
- Proximity marketing
- Automatic check-in
- Real-time location systems (RTLS)
- Asset tracking

Bluetooth LE beacons can be applied in medium-accuracy short-range/indoor (3-5 m) location services where a smartphone can detect the proximity to a beacon and know the general location (room/department) of the signal's source. This can be in the form of either sending promotions (e.g., coupons, sales, etc.) to a shopper in the vicinity of a retail store or allowing a user to navigate to a location via a mobile app regardless of whether or not cellular data is available (e.g., train station, park, etc.). Another example that is growing in popularity is the real-time tracking of high-value assets such as plant tools in industrial applications, shipping containers in logistics applications, or even hospital equipment in healthcare applications.

An SoC that integrates sub-GHz and Bluetooth LE radio modules can effectively serve all of the above applications as they also often leverage the sub-GHz spectrum – reducing wireless subsystem cost substantially. As shown in Figure 2, the sub-GHz frequencies are used in many transportation, medical, consumer, utilities, agricultural and industrial applications and clearly overlap with many Bluetooth applications.

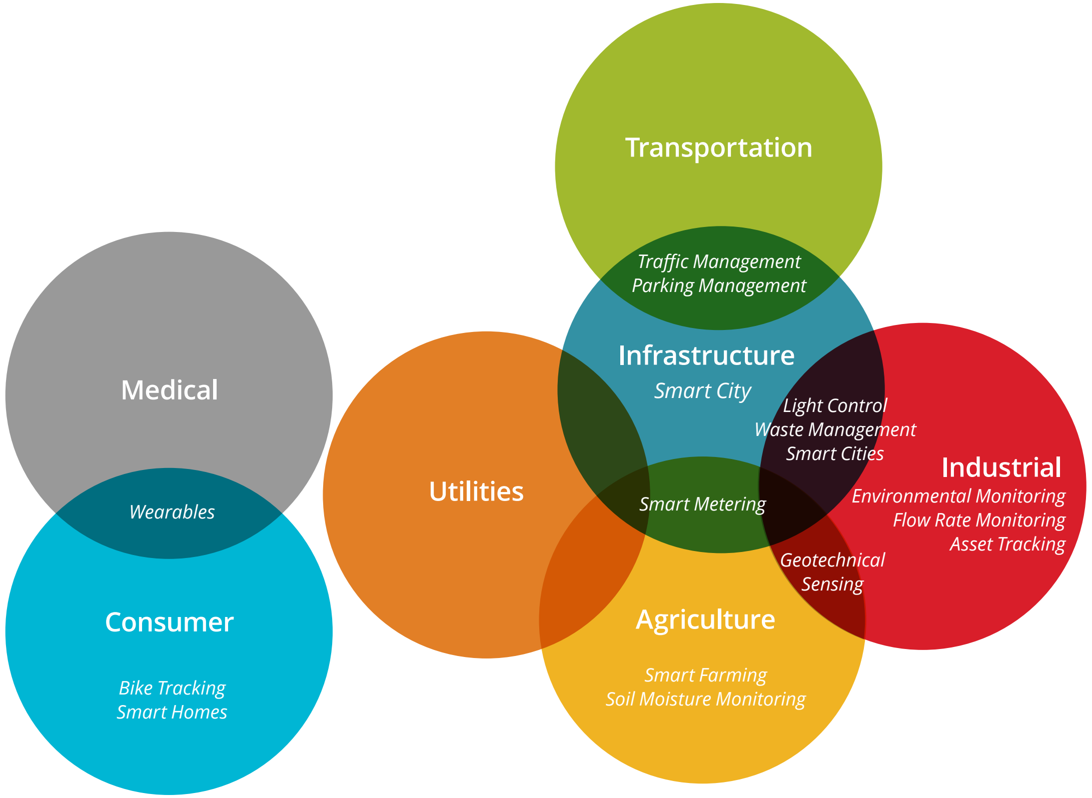


Figure 2: Industries that leverage Sub-GHz technologies and their respective applications.

Integrated Bluetooth Beacons with Connected Lighting for Location-Based Services

A key potential application for multiprotocol SoCs is sub-GHz connected lighting equipment in malls and retail centers. A chip that supports simultaneous Bluetooth and sub-GHz connectivity could deliver a Bluetooth beacon for the dissemination of localized content while also allowing remote control of infrastructure lighting (See Figure 3). Connected lighting infrastructure can be used to increase beacon density in place of dedicated beacons, enhancing location accuracy and in turn, driving better mobile engagement. A connected lighting network that also broadcasts beacons provides a scalable platform to deliver automation, energy savings, and Real-Time Location Systems (RTLS) in commercial, retail and industrial settings.

How Bluetooth LE and Sub-GHz Can Coexist on One Chip

Integration of multiple radio modules in one chip is possible through the slotting of transmissions. Most sub-GHz systems are able to save on battery life due to their narrowband modulation schemes as well as the infrequent nature of transmissions from the sensor nodes; this enables end devices

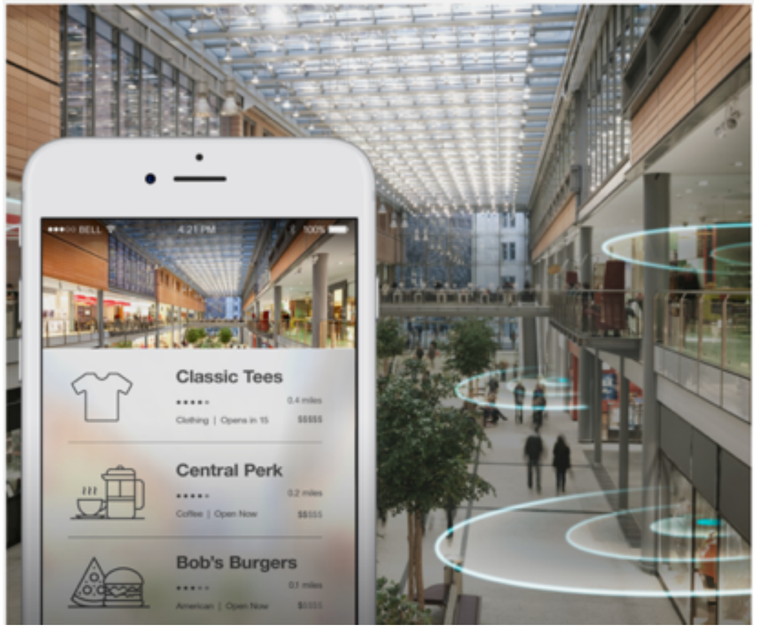


Figure 3: Bluetooth beacons have retail-based applications where a wayfinding phone application could also send store-specific content such as coupons to a customer's smartphone.

to operate in a low current consumption states – receive, idle and sleep. While sub-GHz end devices such as smart meters or smart lights are operating in receive mode, other devices supporting Bluetooth LE such as smartphones or tablets can transmit to the sub-GHz devices. This built-in intelligent scheduling mechanism allows Bluetooth and sub-GHz protocols to be leveraged on the same chip.

For proximity sensing applications, the typical Bluetooth beacon requires 1 ms to transmit a beacon. A beacon interval—the frequency at which a beacon transmits its advertising packet—is typically quite large (>100 ms). This amount of time enables the sub-GHz radio module to be in a listening mode for the majority of the time to reliably control lighting. For smart meter applications, an OTA firmware upgrade would likely require much more time for a Bluetooth connection. This is sustainable due to the infrequent nature of a localized OTA upgrade.

Conclusion

IoT devices have proliferated throughout all industries enabling a seamless method

Figure 4: Scheduling BLE and Sub-GHz for multi-protocol applications.

to monitor and track parameters remotely. Sub-GHz frequencies fill an underserved niche of low- to medium-throughput long-range links with a potential challenge of not being able to connect nodes to the internet without an IoT gateway. Bluetooth presents the opportunity to enhance functionality of sub-GHz connected products by enabling localized access directly to these devices. This can be achieved through a multiprotocol SoC and some clever scheduling techniques. Moreover, trending Bluetooth beacon technology can be used in tandem with many sub-GHz technologies, providing an innovative platform for location-based services. ■

Plug-and-play: Bluetooth Xpress module simplifies home automation product design

Although already prolific, IoT devices are still gaining momentum, saturating nearly every industry vertical from medical to industrial, infrastructure to home and more. Bluetooth connectivity has successfully filled a niche for short-range wireless links as an effective cable replacement technology in many applications. Moreover, virtually every existing device (e.g., smartphone, tablet, laptop, etc.) is Bluetooth-enabled, allowing for a massive array of opportunities for data collection or control applications via a smartphone.

Still, while Bluetooth connectivity in user devices does lower the barriers for remote control of IoT end devices, it is only one piece of the design and development puzzle of an IoT system. Adding Bluetooth connectivity to an end device is a multi-faceted problem, crossing the firmware development, RF hardware design, mobile app development and secure cloud connectivity disciplines. This article will cover some of the difficulties that the average IoT system developer could fall into when trying to add Bluetooth to a design by running through a simple point-to-point system with a smartphone-controlled lamp use case. In addition, it explains how implementing the SiLabs Bluetooth Xpress module could mitigate, if not eliminate, some major obstacles by offering what is essentially a DIY Bluetooth-based home automation platform.

Firmware development

Bluetooth firmware at a glance

The Bluetooth protocol stack is implemented in software with two separate parts: A host stack and a controller stack. The host stack (e.g., MCU, PC, smartphone, etc.) contains high-layer data to control the Bluetooth host controller (typically the separate Bluetooth IC). It is often implemented as part of an OS or with installable drivers on top of the OS. The controller stack—often used for a separate Bluetooth chip that includes the radio and microprocessor—contains the lower-layer data to control the radio unit. The host controller interface (HCI) is therefore the layer that enables control of the host controller, or Bluetooth IC. For PC-to-Bluetooth-IC communication, this transport layer is generally USB, but for MCU-to-Bluetooth-IC communication, the HCI is often the serial UART interface. There are highly embedded System-on-Chip (SoC) systems for wearable applications such as Bluetooth headsets in which the wireless MCU can support the application code as well as the Bluetooth host and controller code.

Most home automation applications do not have the stringent dimensional requirements of wearables, factoring out the need for SoC firmware on a single MCU. Embedded home automation will likely involve an MCU and a separate Bluetooth IC. This type of communication is known as the Network Co-Processor (NCP) model, wherein the application runs on the host MCU while the host and controller code run on a target MCU host and target MCUs communicate

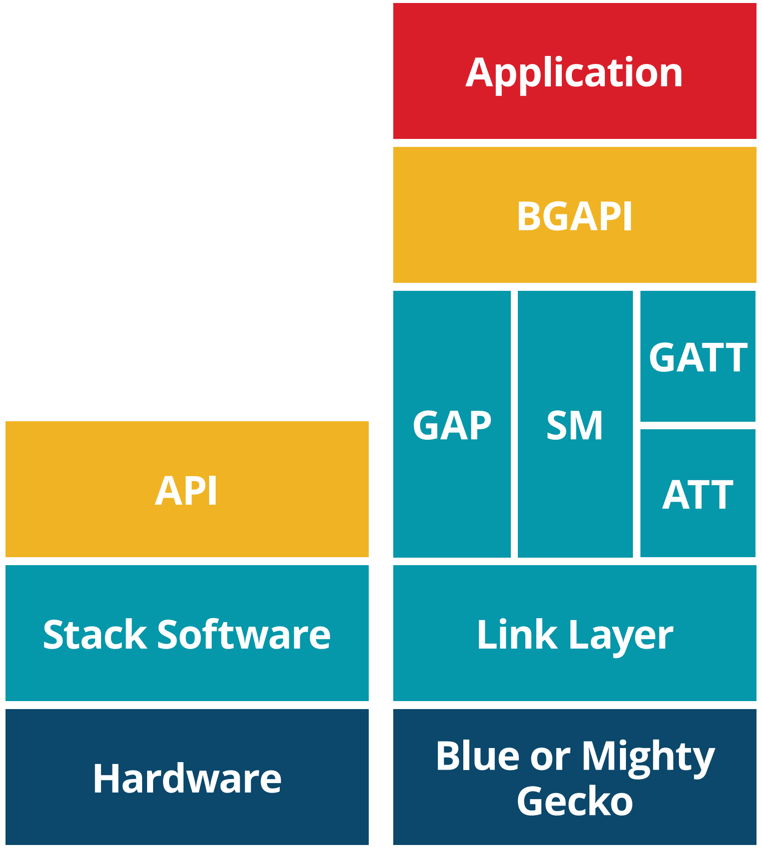


Figure 1: Bluetooth stack architecture block diagram.

through a serial interface. The NCP-MCU relationship in embedded designs has a range of complexity involved in developing the Bluetooth stack.

The Bluetooth Stack

As shown in Figure 1, the Bluetooth stack involves several major components. The link layer controls the low-level communication over the physical interface, using the HCI to send data to the high-layers in the stack. The Generic Access Profile (GAP) controls advertising and determines how devices interact, allowing for a Bluetooth device to be visible while managing its connections. The Generic Attribute Profile (GATT) then takes over after the GAP establishes a connection and details the nature of the exchange of data. As defined by Bluetooth SIG, the GATT profile describes a use case, roles and general behaviors based on the GATT functionality. Therefore, in order to establish a seamless connection between two devices for a particular application, bytes of data must be transferred from one end-point to the other through the GATT profile hierarchy. Naturally, this opens up a Pandora's Box of unknowns and potential complications. The benefit of the Bluetooth Xpress module is namely that this is all taken care of "under the hood".

Firmware Development Considerations

There are many Bluetooth stack solutions in which custom APIs allow configurability for specific embedded applications. To gain this knowledge in and of itself requires expertise. A designer involved in writing Bluetooth firmware must become familiar with vendor tools and their respective Bluetooth stack APIs to generate a NCP. This process can take a significant amount of time considering an expertise in Bluetooth firmware has to be developed and then that knowledge has to then be applied to the specific embedded application.

While the programmable MCU approach generally relies on a vendor-specific Integrated Development Environment (IDE), the Bluetooth Xpress module does not. The interface is human readable and can be readily communicated with using either the Simplicity Studio's Xpress Configurator Application, or a generic terminal program right out of the box.

A primer for configuring the Bluetooth Xpress module

This example uses the Blue Gecko Xpress (BGX) expansion board (BGX13P) with the BGX13 Bluetooth Xpress module. The module includes a UART-to-USB bridge device to enable simple firmware development with a PC via the USB interface. Communication with the BGX module would ultimately occur over the device's UART interface. The first step would be to install the Silicon Labs Simplicity Studio Integrated Development Environment (IDE) drivers that enable the expansion kit to communicate with Simplicity Studio over a serial COM port connection. Next, open up the Xpress Configurator App—a program that communicates with the BGX device via the USB interface to configure parameters such as UART baud rate, GPIOs, modes of operation, system configurations, as well as advertising and connection configurations. As an added note, while the IDE can be installed there is only a need to leverage the Xpress Configurator GUI for configuration via USB. Furthermore, any terminal interface can be employed. The BGX will appear as a COM port.

Smartphone control of the BGX device can be accomplished through the use of the BGX commander application that can be readily installed on any Android or iOS device. As shown in Figure 2, this application can

Figure 2: OTA support for the BGX module is offered out of the box with the BGX commander mobile application. This allows for remote configuration of Bluetooth capabilities regardless of the type of smartphone being utilized.

remotely send messages to the Bluetooth module which will, in turn, appear on the Xpress Configurator (or terminal emulator of choice) over the serial connection. The applications offered through Simplicity Studio offer a relatively straightforward GUI that allows users to readily switch between stream mode and command mode in order to configure a Bluetooth module. Stream mode provides a raw data interface where the MCU writes a byte to Bluetooth Xpress via its UART RX pin. Then, Bluetooth Xpress handles all the Bluetooth activity, and the byte appears in a buffer on the mobile application—or—the byte appears across the TX pin of the receiving Bluetooth Xpress product. Command mode allows a user to adjust the default settings of the Bluetooth module such as the power mode, pin code and baud rate to better fit a particular embedded application. It should also be noted that the reconfiguration of the Bluetooth module settings can occur remotely through the BGX commander application.

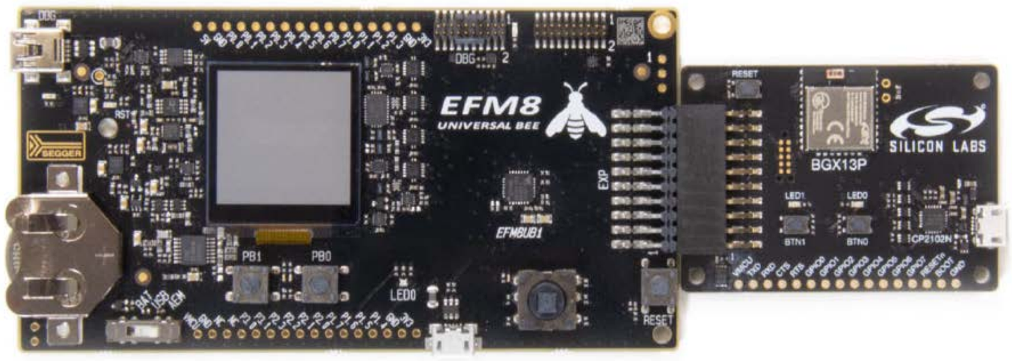


Figure 3: BGX Expansion Board connected to EFM8.

Bluetooth Xpress module for Bluetooth-controlled lamp application

The BGX module can add Bluetooth capabilities to nearly any home appliance, including a lamp. This can be accomplished by leveraging a relay, what with the fact that most 3.3-V or 5-V MCUs will not be able to directly operate a relatively high-voltage lamp (12 V and up). Relays offer electrical isolation between the lamp and the controlling circuitry (MCU + BGX) by allowing smaller controlling voltages to cause a relay to close. This enables an electrical path from a wall outlet to the controlled circuit—allowing for 12 V across the lamp—while disabling (causing an open circuit) the electrical path from the MCU to the wall outlet. As an additional note, many LED-based lamps

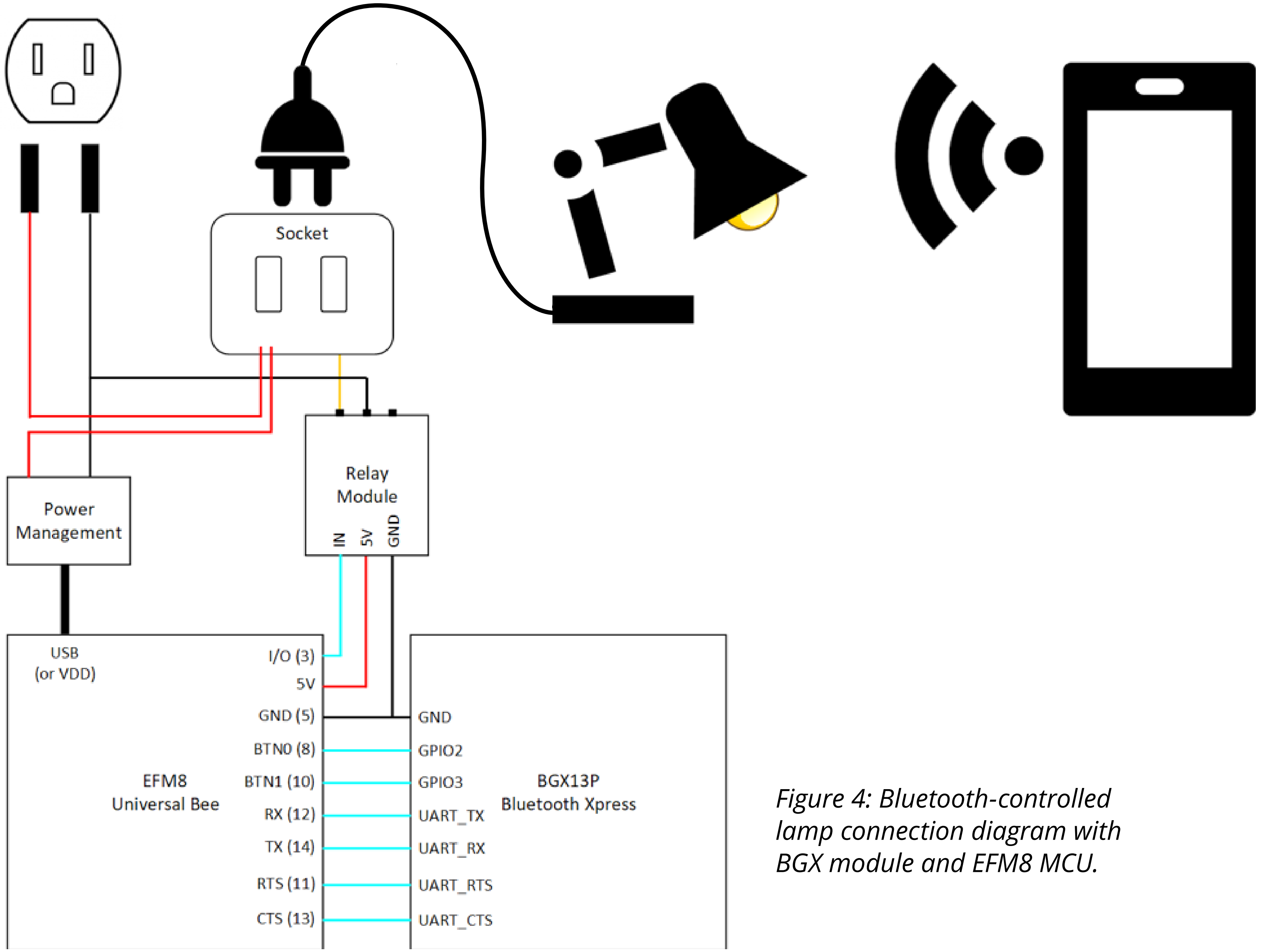


Figure 4: Bluetooth-controlled lamp connection diagram with BGX module and EFM8 MCU.

operate at low enough voltages, making it so that a relay may not be necessary.

This example leverages the EMF8 Universal Bee board, which can be used in tandem with the BGX expansion board for prototyping purposes. The expansion board includes 20 header pins to directly connect with the MCU EMF8 starter kit, as shown in Figure 3.

The diagram in Figure 4 illustrates the general setup of the Bluetooth-enabled lamp, wherein the BGX is connected to the host MCU's (EMF8) UART Rx, Tx, RTS and CTS ports as well as its Vcc and GND ports. The relay connects to a digital input/output pin for control. The relay is connected to the wall outlet to be readily controlled by 5-V logic from the MCU via a safety relay box that protects the controlling MCU and Bluetooth module. The relay box includes the relay and power management circuitry that ultimately powers both the MCU (and, thus, the BGX) as well as the lamp from the wall outlet. The final product could be rapidly implemented with the MCU and system-in-package (SiP) BGX module in a custom PCB. This ultimately allows for a small-form-factor IoT product.

Traditionally, the firmware development involved in maintaining a simple point-to-point link such as this would require significant knowledge of the Bluetooth stack and use of a GATT table. This is in contrast to the minimal development required when implementing the Bluetooth Xpress solution; a designer can send/receive data without knowing how the Bluetooth protocol is used to get bytes of data from one endpoint to another. And, the Xpress framework effectively interfaces with a PC or smartphone regardless of its native OS. This, in effect, provides a base to not only rapidly generate a product but also minimize the learning curve associated with developing an expertise in engineering an IoT product design.

Hardware development

Bluetooth certification considerations

A common challenge to developing a Bluetooth product is the Bluetooth qualifications that go along with it. In the interest of interoperability of any and all Bluetooth products, the Bluetooth Special Interest Group (SIG) released a number of specifications whereby manufacturers must submit their designs for intensive qualification testing prior to receiving a license for use. Having pre-certified hardware from SIG's End Product Listing (EPL) right out of the box minimizes the time associated with certifying Bluetooth under a company's specific brand. As stated on the Bluetooth website, the qualification process can be accomplished without required testing with designs that fall under one of three conditions:

- The device leverages another organization's qualified Bluetooth end product (or subsystem) without modifying its Bluetooth capabilities.
- The device is being redistributed under a new brand name from a third-party manufacturer that has already qualified their Bluetooth end product (or subsystem).
- The device involves a combination of already qualified Bluetooth end products (or subsystems) without modifying their Bluetooth capabilities.

As a note, a Bluetooth end product is typically a combination of two to three subsystems including a Bluetooth controller (radio, HCI), Bluetooth host subsystem (protocols and profiles) and a profile subsystem. As shown in Figure 5, the testing qualification process tests the radio and software capabilities of the device to enable interoperability between Bluetooth devices regardless of the vendor diversity. The radio qualification verifies the conformance to the Bluetooth specification, while the software qualification verifies the Bluetooth stack's conformance to the Bluetooth specification. Both of these allow for a component or subsystem listing under SIG. An EPL combines these without the inherent limitations of the licensing that comes with them. In other words, there is much more flexibility to develop practically any type of Bluetooth product based on combinations of certified subsystems.

Developing an embedded hardware and RF expertise

Not only are Bluetooth conformance tests a significant consideration for the design and manufacturability of an end product, but the PCB layout and antenna selection itself require a fairly large amount of RF expertise combined with Bluetooth expertise. Considerations such as space, cost and antenna radiation pattern as well as its respective matching circuitry is highly relevant. Integrated antennas need to consider the surrounding components, casing and the space to the ground plane in order to minimize signal loss. Typical tests for any wireless system include the Total Radiated Power (TRP) and Total Isotropic Sensitivity (TIS) to ensure that the product falls within FCC power restrictions and that the receiver has a minimum packet-error rate (PER) at various RF levels. These tests can be performed OTA for highly integrated Bluetooth products, but that requires pricy anechoic or reverberation chambers. If a user leverages a module with an onboard antenna, then the significant performance tests are already overcome. A user that chooses to incorporate an external antenna must then consider the cost of re-certification. A company that chooses to start an embedded Bluetooth product from scratch must then consider the cost of an RF engineer to design an RF front end, the

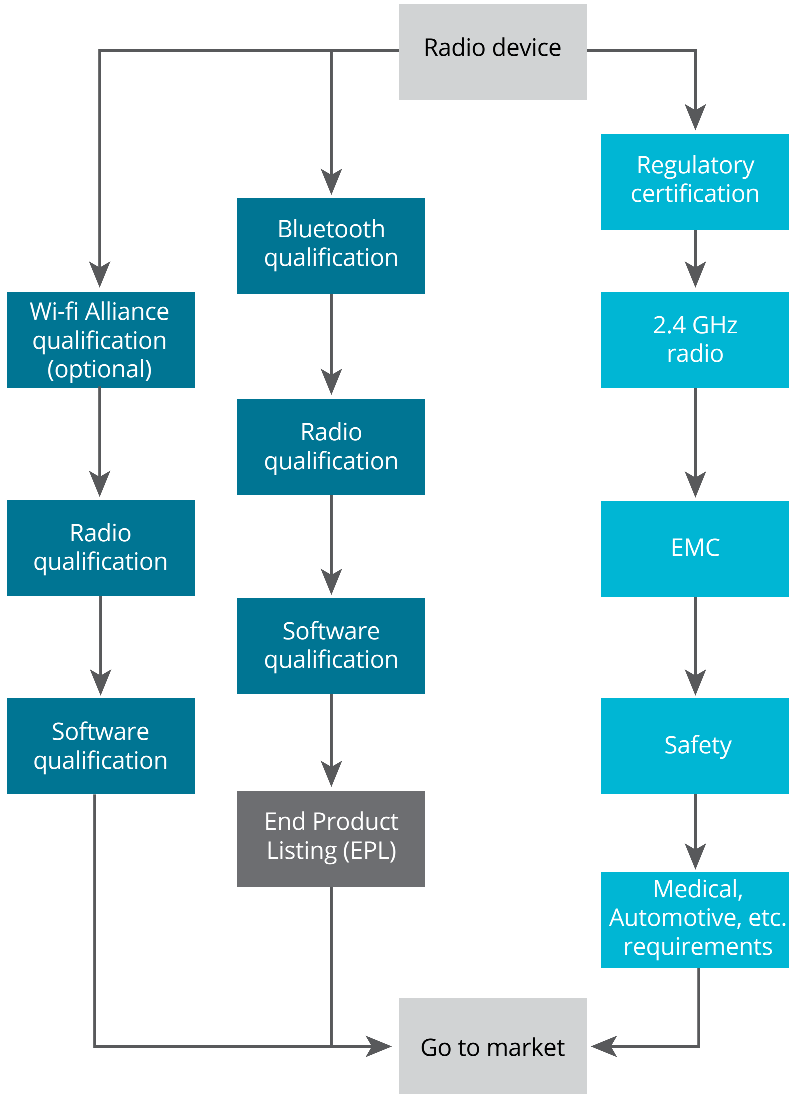


Figure 5: The Bluetooth qualification process includes intensive radio performance tests and software qualification prior to receiving certification.

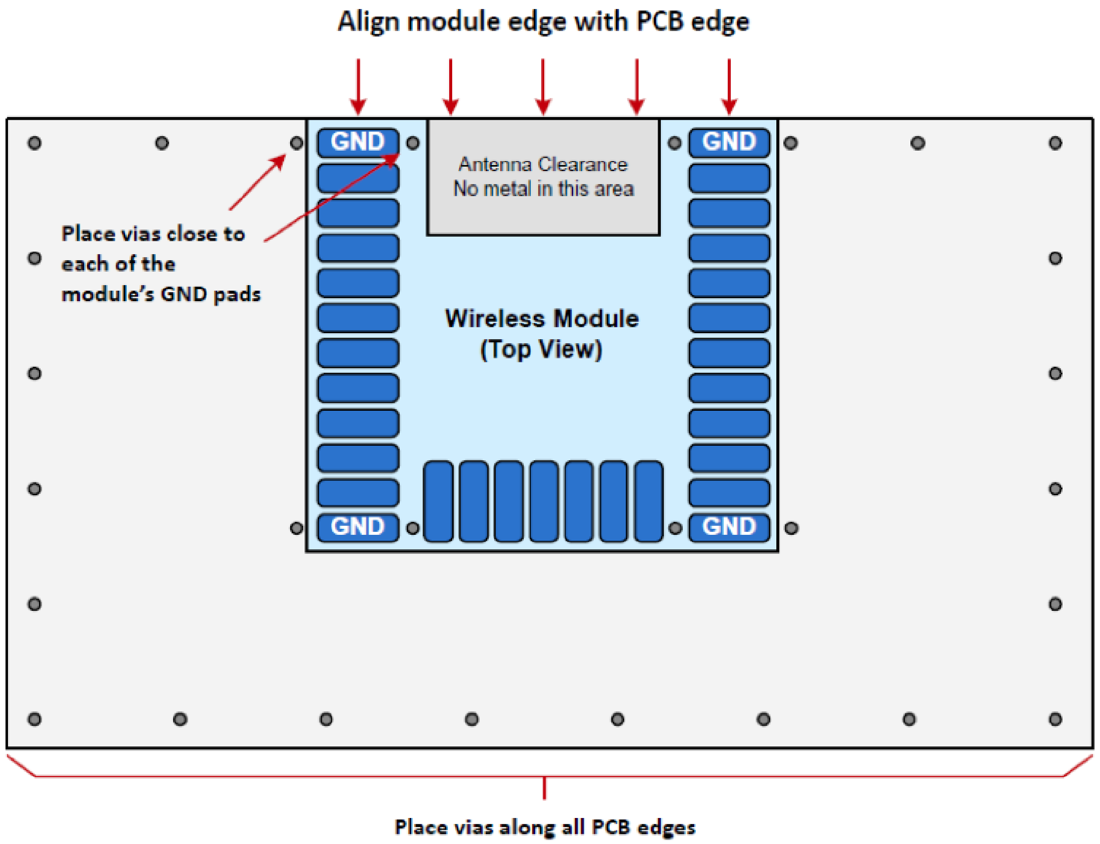


Figure 6: The SiP form of the BGX module can be placed on the edge of a PCB without taking up much real estate.

cost of renting or purchasing the required lab equipment (e.g., network analyzer, anechoic chamber, signal analyzer, etc.) and the cost of Bluetooth radio certification—simply to develop the hardware interface.

Leveraging the Bluetooth Xpress module for Bluetooth-controlled lamp

The BGX module comes with an embedded chip antenna that has been certified wherein the transmit and receive characteristics are well-defined. For an embedded application, the SiP form of the BGX module can be mounted on the edge of a PCB for optimal RF performance (Figure 6). The small-form-factor, highly integrated BGX hardware simplifies the process of adding Bluetooth capabilities to any home appliance or the process of designing a Bluetooth-enabled product from scratch.

Conclusion

This article illustrated some of the ways that the Bluetooth Xpress solution can simplify the process of generating a Bluetooth-enabled lamp. This can be applied across the home automation industry, whereby the user-friendly GUI and source code enables simplified application layer development all the way down to hardware control. The BGX module can, in essence, be a plug-and-play component and a strong tool in the repertory of any embedded IoT development company with certified hardware out of the box. ■

Conclusion

This e-book has explored the many different types, features and use cases for Bluetooth connectivity. Are you ready to get started with Bluetooth development? Then head to SiliconLabs.com where you can check out the latest demos, app notes and tutorials to get up and running.

Want to speak with us directly? Contact Silicon Labs now.