

# element14 Essentials: Wireless Protocol I

[element14 Learning Center](#)



## Wireless Protocol I: Bluetooth

Sponsored by



[1. Introduction](#)

[2. Objective](#)

[3. The 2.4 GHz  
Ecosystem](#)

[4. A Brief History of  
Bluetooth](#)

[5. The Bluetooth  
Standard](#)

[6. Bluetooth 5.0](#)

[7. Bluetooth Low Energy  
\(BLE\)](#)

[8. Bluetooth Devices](#)

[Parts Used](#)

[Test Your Knowledge ▶](#)

### 1. Introduction

Legend has it that Harald Blaatand ("Bluetooth") Gormsen was the Danish king who united parts of Denmark and Norway and, eventually, other parts of Scandinavia, and who may have also introduced Christianity into Denmark. Thus, when Jim Karbach, an engineer at Intel® in the late 90s, decided to name the wireless standard for uniting personal use devices he chose Bluetooth.

### 2. Objective

The objective of this learning module is to acquaint you with the Bluetooth standard.

***Upon completion of this learning module, you will be able to:***

- Demonstrate a deeper understanding of Bluetooth
- Explain some key features of the Bluetooth standard
- Explain the enhancements in Bluetooth 5.0
- Explain some characteristics of BLE
- Explain Bluetooth beacons
- Dazzle your friends with a piece of Norse/IT trivia (see intro paragraph)

### 3. The 2.4 GHz Ecosystem

Before we begin looking at Bluetooth in detail, it is worth discussing the wireless ecosystem in which Bluetooth and Bluetooth devices operate. We have discussed some aspects of this wireless ecosystem in Part II of the element14 Essentials series on [Wireless Microcontrollers \(MCU 2\)](#).

As discussed in the Essentials course MCUs 2, Bluetooth is a technology that operates in the 2.4 GHz unlicensed ISM band. The 2.4 GHz band is one of the spectrum bands that regulatory bodies have made available for unlicensed use. The FCC in the United States, and other similar bodies around the world, have made most of the 2.4 GHz band available for such unlicensed use, for the sake of compatibility. This means that the 2.4 GHz band is available for anyone to deploy any kind of wireless or, more specifically, Radio Frequency (RF) application, as long as they do not exceed the mandate on maximum transmit power (if you are interested in the specifics, see: <http://afar.net/tutorials/fcc-rules/>). This has led to a considerable amount of innovation in the 2.4 GHz ISM band, such as products like baby monitors and cordless phones. Two of the

most dominant wireless technologies of the last 20 years, Wi-Fi and Bluetooth, have both been developed for this band. In addition, Wi-Fi is also being developed for the 5.8 GHz unlicensed ISM band.

While this plethora of devices and technologies has been a great boon for the marketplace, it has also been a great bane, because every device that is using the 2.4 GHz band is capable of interfering with, and disrupting the communications of, other 2.4 GHz band users. To compound the problem, one of the most popular household appliances, the microwave oven, emits microwave radiation precisely in the 2.4 GHz band. So yes, if the popcorn is popping, perhaps your Wi-Fi is dropping.

We will later see that Bluetooth has some unique approaches to managing this complicated ecosystem, both to minimize the effects of external interference on Bluetooth communication and to reduce the impact of Bluetooth on other communications. For now, remember that it is important for anyone seeking to build applications using Bluetooth to keep the ecosystem in mind.

#### 4. A Brief History of Bluetooth

The concept of Bluetooth originated with the Ericsson cellphone company (with the logo based on a melding of Scandinavian runes letters H and B for Harald Bluetooth). Engineers at Ericsson were looking for an efficient and wireless method of attaching other devices to the cellphone. Ericsson worked with a number of industry practitioners, such as Intel®, IBM, Nokia, and Toshiba to form the original Bluetooth Special Interest Group (SIG) [<https://www.bluetooth.com>] in 1996. Other companies like Microsoft, Apple, and Lenovo have since joined the Bluetooth SIG. The Bluetooth SIG is the entity in charge of the standardization of Bluetooth.

Even though the first versions of the Bluetooth standard (1.0 and 1.0B) were released a few years after the inception of the Bluetooth SIG, there were a number of problems and errors reported with those versions, and thus Bluetooth did not truly take hold until the stable 1.1 version. The 1.1 version was also ratified by the IEEE Working Group on Personal Area Networks (IEEE 802.15.1) in 2002, which also resulted in wider adoption of the standard. Bluetooth 2.0, released in 2004, introduced an improvement called the Enhanced Data Rate (EDR), resulting in wide growth and adoption of Bluetooth in the industry. Since then, there have been regular updates to the standard, for example ver. 2.1 (2007), ver. 3.0 (2009, introduced HS for High Speed Transport), ver 4.0 (2010, which introduced LE, low energy), 4.1 (2013), 4.2 (which contained enhancements for IoT), and finally ver. 5.0 in the summer of 2016.

In many ways, the growing ubiquity of cellphones, and especially the smartphone, has led to the widespread adoption of Bluetooth. The combination of a solid technology and standard, coupled with the right market forces, has driven the growth of Bluetooth.

In this learning module we will not discuss every version in great detail, instead we will explore the key features of Bluetooth as it stands today. Specifically, we will try to understand essentially how Bluetooth works, and what enhancements to expect with version 5.0, as well as what features are unique to Bluetooth Low Energy (BLE).

It is interesting to note that Bluetooth was the working name for the standard being developed in the late 90s; other names considered at that time were RadioWire and PAN, for Personal Area Network. According to the SIG, the PAN name was found to have too many trademark collisions even though it was the favored choice initially, and a trademark search could not be completed in time for the launch of the first standard. I think one can safely say that Bluetooth is a much more distinctive and memorable name!

#### 5. The Bluetooth Standard

As mentioned above, the Bluetooth standard operates in the 2.4 GHz band, and thus the Radio Frequency (RF) being utilized is in the microwave range. Several design decisions in the Bluetooth protocol and standards are driven by its being in this dense and complex 2.4 GHz ecosystem.

Bluetooth operates between 2402 MHz and 2480 MHz, and has 1 MHz wide channels, resulting in 79 channels. BLE uses 2 MHz, leading to 40 channels. Basic Rate (BR) operation uses Gaussian Frequency Shift Keying (GFSK) to achieve a data rate of 1 Mbps on each channel. Enhanced Data Rate (EDR) allows for faster data rates through the use of Differential Quadrature Phase Shift Keying (DQPSK). Note that this is the raw data rate and includes overhead; it is typical to expect a data throughput of 800 Kbps with ver. 4.2.

All of the channels are used for data transmission somewhat simultaneously in a Frequency Hopping Spread Spectrum (FHSS) scheme. This means that the active channel of communication is switched, or "hopped," between these 79 channels. Bluetooth hops 1600 times a second, for a dwell time of 625 microseconds per channel.

All 79 channels may not be used in every region of the world, depending on local regulations; for example, in France, Spain and Japan there are only 23 channels available. Additionally, when Bluetooth experiences continuous interference in certain channels they are removed temporarily from the hop sequence, resulting in fewer than 79 channels being used, but ensuring that there are more robust communications. In this sense Bluetooth is more resistant to your popcorn popping in the microwave!

A central notion in Bluetooth is that of a piconet. Bluetooth was created for the use of personal devices, and thus was designed for devices that are in close proximity with one another. In communication terminology, devices that are networked in close proximity are called Personal Area Networks (PAN). The basic network in a PAN is the piconet. Each piconet has a Master (M) device and a number of Slave (S) devices. The Bluetooth standard was developed in such a way that allows for the development of very cheap and low-power peripherals; thus, the Slaves in a Bluetooth piconet are far less complex than the Master. The Master defines the hopping sequence inside a piconet, and directs Slaves regarding when they can send data, when they should power down, and so forth.

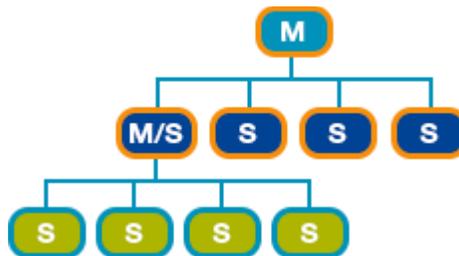


Figure 1: Bluetooth Piconet and Scatternet

Once the hopping sequence is set up communication proceeds, with different devices taking turns communicating by transmitting in predefined slots. Typically, even slots are reserved for the Master's transmissions and odd slots for the Slaves. The Master defines which slots belong to which Slaves. A Slave in the piconet can function as a Master for a secondary piconet, thus extending the piconet and making it a scatternet. See the figure to the left. The core specification dictates that a piconet can have a maximum of 1 Master and 7 Slaves.

Let us now discuss some key elements of Bluetooth in more detail.

### - 5.1 Links

Bluetooth has two connection link types available: Asynchronous Connectionless Link (ACL), and Synchronous Connection Oriented (SCO) Link. Which link is used in a specific situation depends on what kind of information is being transferred.

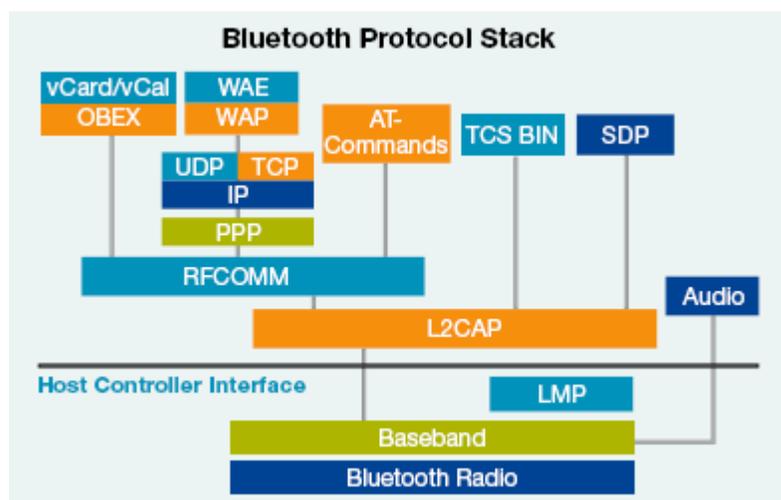
ACL uses a framed data transfer link. The objective here is to be able to transfer data objects that are not time dependent, such as files, contacts, and so on. A set of error detection and correction mechanisms is included. Forward Error Correction (FEC) is used to reduce the need for retransmissions. Both unidirectional and bidirectional data transfer are supported. With bidirectional mode, there is support for symmetric (same data

rate in both directions) and asymmetric (much higher data rate in one direction, a common need in many applications) data transfer.

SCO is meant for streaming communications. This is typically used for audio, for example in phone headsets and Bluetooth speakers. Early versions did not include any way of adding Acknowledgements (ACKs), but since ver. 1.2, extended SCO (eSCO) added in support for ACKs.

## - 5.2 Links

The Bluetooth SIG actually defines a complete protocol stack, starting from the application layer down to a physical layer. What complicates things is that separate stacks are possible for the different profiles, of which there are now over 20. No wonder the original specification was 1,500 pages long! Moreover, the protocol stack is not exactly compatible with the OSI or TCP/IP models. The IEEE Working Group defined how Bluetooth is integrated into the traditional Internet (802) stack. This was done by focusing mainly on Link Layer down. There is a Link Layer Control and Adaptation Protocol (L2CAP) that in some sense bridges Bluetooth at the low level with any standard application or application/transport setup at the higher layers.



L2CAP allows multiple applications to use the Bluetooth wireless transmissions. Therefore, this provides a multiplexing function, which is somewhat unusual. Incidentally, audio and control functions can be directly accessed by an application, thus bypassing the L2CAP layer. Above the L2CAP is a middleware layer that provides some standard transport functions, such as RFCOMM, for connecting a variety of peripherals, and Telephony, for use with speech-oriented applications. Below the L2CAP layer is a baseband layer that performs functions similar to a traditional MAC layer.

Another fundamental element of the Bluetooth protocol stack is the Service Discovery Protocol (SDP), which enables devices to discover what services are offered by other devices in a piconet, and what parameters are required in order to connect with those devices.

## - 5.3 The Connection Process, Pairing, and Bonding

Establishing a Bluetooth connection happens in phases:

- **Phase 1 (Inquiry):** It is reasonable to assume that the first time two devices connect they know absolutely nothing about each other. In this situation, one or both of the devices first has to scan to see what other Bluetooth devices are within range. This process, which appears as scanning to the end user, is called *Inquiry* in Bluetooth terminology. In the *Inquiry* phase one device will send out an *Inquiry* message and the other device will respond with its ID and address. In order for a device to respond to an *Inquiry* it must be *discoverable*.

■ **Phase 2 (Paging):** Once two devices know of each other's existence, i.e. one device has discovered the other and ID information has been exchanged, the next step is for them to try to establish a connection. During this phase a passkey or PIN code, which is usually 4 digits, is authenticated between the two devices. The *Paging* process is initiated by the Master device, which pages the specific Slave device.

■ **Phase 3 (Pairing):** Once the passkey has been successfully authenticated, a short-term key is generated that will be used throughout a session. The two devices are then *paired* with each other.

If the end user so indicates, the security information in the pairing details can be stored for use in a future session. In this case, the two devices will pair automatically when they are within range of each other, and they are said to have *bonded*.

Once a connection has been established it can be in one of several modes. See the table below for a look at connection modes.

**Table: Bluetooth Connection Modes**

Active	Both devices actively communicate with each other (though, as discussed before, the connection will typically be asymmetric).
Sniff	For saving power, a device will go into a power down mode and only come on at regular intervals.
Hold	The master can direct a slave device to go to sleep or power down for a pre-specified "hold" interval.
Park	The master can "park" a slave device. The slave will go to sleep or power down mode with only the ability to receive a "wake-up" command from the master.

### - 5.4 Classes

Bluetooth devices can belong to one of several power classes, depending on their transmit power and range. The table below shows the classes in the ver. 4.2 core specification.

Class 1	20	100	100 m
Class 2	4	2.5	10 m
Class 3	0	1	1 m

### - 5.5 Profiles

As we have already discussed, Bluetooth defines a number of application profiles. These dictate not only the kinds of applications that can use Bluetooth, but also actually define the protocol stack specific to a given application profile. Certain audio profiles, for example, can bypass the L2CAP layer that was discussed earlier, and a number of profiles have to use the RFCOMM middleware layer. The table below lists some Bluetooth profiles.

Advanced Audio Distribution Profile	AADP	For streaming stereo quality from a source to a receiver ( <i>sink</i> )
Audio/Video Remote Control Profile	AVRCP	Self-explanatory
Basic Imaging Profile	BIP	Controlling an imaging device
Basic Printing Profile	BPP	Self-explanatory

File Transfer Profile	FTP	Self-explanatory
Generic Object Exchange Profile	GOEP	Transferring of objects, ex: vCards
Hands-Free Profile	HFP	Self-explanatory
Headset Profile	HSP	Self-explanatory
Human Interface Device Profile	HID	For keyboards, mice, and other devices for human interaction
Personal Area Networking Profile	PANP	To form a PAN, an ad-hoc network
Service Discovery Application Profile	SDAP	Application profile for SDP
Serial Port Profile	SPP	Emulating a wireless Serial connection
Synchronization Profile	SYNC	Self-explanatory
Video Distribution Profile	VDP	Streaming video

## 6. Bluetooth 5.0

In typical Bluetooth tradition, the ver. 5.0 specification is over 2,800 pages long. Version 5 doubles the data rate, quadruples the range under certain circumstances, and allows for 8 times longer packets, which implicitly increases throughput, as compared to older versions of Bluetooth. The base data rate is doubled from 1 Mbps to 2 Mbps, resulting in an effective throughput increase from 800 Kbps to 1400 Kbps. In turn, this also means that the maximum possible data rate is increased from 24 Mbps to 48 Mbps. The range enhancements are due to the inclusion of Forward Error Correction (FEC). Depending on the FEC scheme being used (S=2 or S=8), the range can be increased significantly, because the addition of the FEC mechanism allows for a more robust communication, albeit at a lower data rate. The figure below summarizes the choices of Physical Layers (PHY) for ver. 5. LE 1M is the existing PHY from older versions of Bluetooth. LE 2M is the double base data rate PHY. The two LE Coded modes are the PHYs that have FEC.

Symbol Rate	1 MS/S	1 MS/S	1 MS/S	2 MS/S
Data Rate	1 Mbit/s	500 Kbit/s	125 Kbit/s	2 Mbit/s
Error Detection	CRC	CRC	CRC	CRC
Error Correction	NONE	FEC	FEC	NONE
Range Multiplier (approx)	1	2	4	0.8
Bluetooth 5 Requirement	Mandatory	Optional	Optional	Optional

*Figure: Bluetooth 5 PHY options. Source: <https://blog.bluetooth.com/exploring-bluetooth-5-going-the-distance> (MS/S means Million Symbols per Second. Note how the base data rate drops significantly with the use of FEC. Thus, it is a trade-off between range and speed.)*

Version 5 is better suited for use with IoT devices because of an enhanced broadcast capability. Message sizes are now increased to 255 bytes from 31 bytes, allowing for a more compelling use case for beacons, which we will discuss in more detail in the next section.

## 7. Bluetooth Low Energy (BLE)

Bluetooth is a fairly low-power technology, built with power-saving features in mind; consider the various low-power connection modes previously discussed. Since the early 2000s there has been a lot of interest in non-broadband wireless applications where various devices — sensors, for example — collect data for some application, such as a sensor network. Nokia developed a standard called Wibree for use in alternative, non-user-centric wireless applications. In order to maintain compatibility with as many devices as possible, an effort has been made to make Wibree as compatible with Bluetooth as possible; in fact, it was also referred to as "baby Bluetooth." Developments in Wibree continued through to the introduction of the Bluetooth ver. 4 standard, at which point it was integrated into the core specification as Bluetooth Low Energy (BLE). (Please note that BLE has also been referred to as Bluetooth Smart by the Bluetooth SIG.)

At the time of the introduction of ver. 4, it had become clear that there was a need for protocols that addressed connecting low computation devices with the rest of the internet (Internet of Things or IoT applications). In addition, BLE has been considered an IoT and Machine-to-Machine (M2M) networking-friendly technology. Developments in Bluetooth ver. 5 in terms of enhanced Beacon and broadcast capability also indicate a full embrace of IoT applications by the Bluetooth SIG.

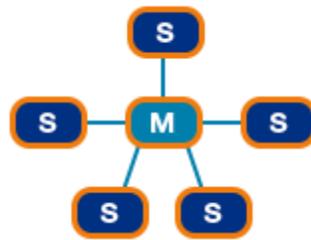


Figure 2: Bluetooth LE, Star Topology

One of the most important features of BLE is that it sacrifices data rate for the sake of conserving power. The base data rate still remains the same as the regular ("classic") flavors of Bluetooth, but the effective throughput is significantly reduced. For ver. 4, the effective throughput is 305 Kbps. BLE has built-in 128-bit AES encryption, and is capable of supporting any number of active slave devices, as compared to the 7 allowed by classic; the topology here is no longer a scatternet in a multiple slave scenario but a star. The reason for both of those topology-related specifications is the same: the applications envisioned can have a single Master and any number of Slave devices (sometimes referred to as "things"). The power consumption is reduced by 50% to 99%, depending on the specific use case scenario compared to Bluetooth classic.

BLE introduces some specialized application profiles of its own. Some of these are shown in the table below.

Blood Pressure Profile	BLP	Healthcare
Continuous Glucose Monitor Profile	CGMP	Healthcare
Cycling Speed and Cadence Profile	CSCP	Sports & Fitness
Running Speed and Cadence Profile	RSCP	Sports & Fitness
Environmental Sensing Profile	ESP	Generic
"Find me" Profile	FMP	Proximity Sensing
Proximity Profile	PXP	Proximity Sensing

### - 7.1 Beacons

One of the most interesting aspects of BLE is the inclusion of a technology called beacons. Beacons are small devices ("things") that periodically, generally about once every 100 msec, broadcast a message with the ID of

the device that is emitting the message. Any listening device that is within range, such as a smartphone or a higher computation-capable controller, could then use that information for an appropriate application. Some potential examples include: geolocation and location-aware services, other proximity-based applications, industrial automation, sensing applications, smart homes, smart shopping and retail, event management, and possibly many others.

Some of the initial applications so far have been little more than glorified RFID, but BLE is more active than passive RFID. With the advent of Bluetooth ver. 5 — and its much larger message sizes — it becomes possible to carry a richer set of information on the broadcast messages. Including the extremely long range possibilities with BLE in ver. 5, plus its FEC feature, perhaps we can look forward to more exciting IoT and smart applications of BLE. Beacons are already in place and used in asset tracking in transportation systems in San Francisco, London, and Moscow.

Eddystone and iBeacon are existing BLE beacon protocols, with the main difference being the kind of information that can be sent by the beacon. iBeacon is more limited, because it only sends the ID. Eddystone can send a URL and Telemetry sensing data, in addition to being able to send an ID. In ver. 5 it is possible to send the ID, URL, and data.

## 8. Bluetooth Devices

There are obviously a whole array of devices that we use in our day-to-day lives that utilize Bluetooth, ranging from our PCs and smartphones to items like smartwatches and fitbits. To design these devices, a sampling of the Bluetooth development tools currently available are discussed in this section.

### - 8.1 Overview

When developing standalone applications utilizing Bluetooth there are a range of development options available. For example, when developing for the Arduino, one can opt for the [Bluefruit modules](#) from Adafruit, or go with an offering from the makers of the Arduino, such as the Arduino BT. The Arduino BT board is supplied with a [Bluegiga Bluetooth module](#) from Silicon Labs. Development boards such as the Raspberry Pi have begun incorporating Bluetooth into the main board, as in the [Raspberry Pi 3](#). Texas Instruments [SimpleLink](#) boards also incorporate Bluetooth in their Wireless MCUs module. Many development boards and MCUs will have options or variants that include Bluetooth.

It is worth looking at some of the options that are available from Silicon Labs, especially the Blue Gecko series. Silicon Labs offers many options in Bluetooth products, as well as offerings tailored for use in Mesh. Silicon Labs offers starter kits, such as the [Low Energy SiP Module](#) and the EFR32 Blue Gecko Low Energy Wireless. These starter kits enable quick development and deployment of Bluetooth applications, and there is added support through the [Simplicity Studio](#) software suite. More details on the Bluetooth Software stack from Silicon Labs are available [here](#).

### - 8.2 Developing with a Wireless SoC or a Wireless Module

Whether you are designing low-energy Bluetooth beacons, mesh-network devices, or IoT sensor-to-cloud solutions, you'll likely need to make use of the development tools, which often include example applications, documentation, and other resources to get started with your own development. This section will cover some of the components and evaluation boards available.

One of the first things to consider is whether to use a wireless system-on-a-chip (SoC) or a wireless module. A wireless system-on-a-chip (SoC) is mounted onto the product's printed circuit board (PCB). It's less expensive than a wireless module, but designing with it may be costly because board design time, regulatory and standards certification costs, and time-to-market issues are generally higher for an SoC reference design than a wireless module.



*EFR32BG13 Bluetooth SoC Microcontroller QFN48*

An example of a wireless SoC is the EFR32BG13 Bluetooth LE IC. It is a 40 MHz ARM Cortex-M4 microcontroller and provides a 19 dBm maximum power output. Built with low-power Gecko Technology, it has fast wake-up times and energy-saving modes. With a receive sensitivity of -103.3 dBm, it supports a full DSP instruction set and floating point unit to speed computation. This device includes 512 kB of flash and 64 kB of RAM. Software and SDKs are available to provide support for Bluetooth low energy (BLE), Bluetooth 5, and Bluetooth mesh networking.

The other design option is to use a wireless module which has the SoC inside. A majority of the design is already done, including a fully-characterized PCB with RF optimization and antenna layout, shielding, timing components (crystals), external bill of materials (BOM), regulatory approvals, and standards certifications. Modules remove unknown risks of designing with a wireless SoC, depending on volume, time-to-market urgency, risk tolerance, and resources, among other factors.



*BGM121/BGM123 Blue Gecko Bluetooth<sup>®</sup> SiP Module*

An example of a wireless module is the BGM121/BGM123 Blue Gecko Bluetooth<sup>®</sup> SiP Module, which is ideal for applications where ultra-small size (51 mm<sup>2</sup>), reliable high-performance RF, low power consumption, and easy application development are important. The BGM121/BGM123 also integrates an ultra-robust antenna, which requires minimal PCB, plastic, and metal clearance. It also integrates a Bluetooth 4.2 compliant Bluetooth stack and can also run end-user applications on-board, or alternatively as a network co-processor over one of the host interfaces.

Once you complete your cost-benefit analysis on whether to use an SoC or a module, you'll need to conduct an evaluation. To facilitate your evaluation, development boards are available to make the process fairly easy. Here are two examples:



## EFR32BG13 Blue Gecko Bluetooth Low Energy Wireless SoC Starter Kit

For SoC evaluation, the Silicon Labs Wireless SoC Starter Kit for the EFR32BG13 Blue Gecko will get you started. The Wireless Starter Kit Mainboard contains sensors and peripherals, enabling easy demonstration of some of the EFR32's capabilities. An on-board J-Link debugger allows debugging of the attached radio board, as well as providing a debug connection for external hardware. A plug-in radio board contains the reference design for the EFR32BG13 chip itself, including the RF section and device-specific hardware. The kit includes a Bluetooth Low Energy Software Stack and integrated debug adapter. This enables developers to quickly establish a Bluetooth connection and evaluate Blue Gecko SoCs. With the supporting Simplicity Studio suite of tools, developers can take advantage of graphical wireless application development and visual energy profiling and optimization. This kit supports Bluetooth 5, Bluetooth beacon, and Bluetooth mesh specifications.



## Blue Gecko Bluetooth<sup>®</sup> Low Energy System-In Package (SiP) Module Wireless Starter Kit

To speed up evaluation of SiP wireless modules, Silicon Labs has the Blue Gecko Bluetooth Low Energy Module Wireless Starter Kit for BGM11S, BGM121 and BGM123 SiP modules. The wireless starter kit comes with the Bluetooth Low Energy module radio board, a main board with coin-cell battery holder, USB and Ethernet connections, display and connections to all the module's peripheral interfaces, as well as an extension board. This kit supports Bluetooth LE and Bluetooth 5 advertising. Provided with the kit is Silicon Labs' Bluetooth software SDK, featuring example applications, documentation, and other resources to help you get started with your own development.

### - 8.3 Internet of Things

What if you'd like to collect and deliver data to your cloud application in your IoT development project? That can be easily accomplished with a Bluetooth-enabled, sensor-driven platform. Here's an example of how to add BLE connectivity to your sensor or actuator applications:



## RD-0057-0201 Thunderboard<sup>™</sup> React Bluetooth MCU

The Thunderboard<sup>™</sup> React Kit is a low-cost BLE solution that collects and delivers data to the cloud. The Thunderboard<sup>™</sup> React is a cloud-connected, Bluetooth<sup>®</sup> Smart-enabled, sensor-driven platform that enables customers to demo, evaluate, and develop their own unique applications. It uses Silicon Labs' BGM111 Bluetooth Smart module as a wireless system-on-a-chip (SoC) to collect various sensor data and deliver it to the cloud through Bluetooth Low Energy (BLE)-enabled iOS/Android mobile apps. It has onboard relative humidity and temperature, ambient light and UV, omni-polar hall-effect sensor, and a 6-axis motion sensor. The Thunderboard<sup>™</sup> React firmware is pre-compiled and has source application firmware. It features a sleep

mode for battery savings, as well as beacon notifications to interface with beacon-compatible mobile apps. It also has Bluetooth-compatible GATT service profiles.

\*Trademark. **Silicon Labs®** is a trademark of **Silicon Laboratories, Inc.** Other logos, product and/or company names may be trademarks of their respective owners.

## Test Your Knowledge

Are you ready to demonstrate your knowledge of the Bluetooth wireless protocol? **Then take a quick 15-question multiple choice quiz to see how much you've learned from this Essentials Wireless Protocol 1 module.**

[Take Quiz](#)