UG103.12: Application Development Fundamentals: Silicon Labs Connect

This document describes the features and functions of Silicon Labs Connect, including its device types, network topologies, and its stack structure.

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

KEY POINTS

- Devices
- Network topology
- Stack structure

1. Introduction

Silicon Labs is developing products designed to meet the demands of customers as we move to an ever-connected world of devices in the home, what is often referred to as the IoT (Internet of Things). At a high level the goals of IoT for Silicon Labs are to:

- Connect all the devices in the home with best-in-class mesh networking, whether with Ember ZigBee PRO or other emerging standards.
- Leverage the company's expertise in low-power, constrained devices.
- Enhance established low-power, mixed-signal chips.
- Provide low-cost bridging to existing Ethernet and Wi-Fi devices.
- Enable cloud services and connectivity to smartphones and tablets that promote ease of use and a common user experience for customers.

Achieving all of these goals will increase adoption rates and user acceptance for IoT devices in the Connected Home.

One such challenge is managing devices requiring low power consumption, such as battery-powered devices where long battery life is essential. To meet this challenge Silicon Labs has developed the Silicon Labs Connect stack. Connect provides a fully-featured, easily-customizable wireless networking solution optimized for devices that require low power consumption and are used in a simple network topology. Connect is configurable to be compliant with regional communications standards worldwide. Each RF configuration is designed for maximum performance under each regional standard.

The Silicon Labs Connect stack supports many combinations of radio modulation, frequency and data rates. The stack provides support for end nodes, coordinators, and range extenders. It includes all wireless MAC (Medium Access Control) layer functions such as scanning and joining, setting up a point-to-point or star network, and managing device types such as sleepy end devices, routers, and coordinators. With all this functionality already implemented in the stack, users can focus on their end application development and not worry about the lower-level radio and network details.

The Connect stack should be used in applications with simple network topologies, such as a set of data readers feeding information directly to a single central collection point (star or extended star topology), or a set of nodes in the same range exchanging data in a single-hop fashion (direct devices). It does not provide a full mesh networking solution such as that provided by the EmberZNet PRO or Silicon Labs Thread stacks.

The Connect stack is part of the Silicon Labs Flex SDK (Software Development Kit), installed through Simplicity Studio. Connect runs on top of RAIL (Radio Abstraction Interface Layer), also included with the Flex SDK. RAIL provides an intuitive, easily-customizable radio interface layer that is designed to support proprietary or standards-based wireless protocols. For more information, see *UG103.13: Application Development Fundamentals: RAIL*.

The Connect stack supports efficient application development through its "building block" plug-in design. When used with the Simplicity Studio IDE (Integrated Development Environment), developers can easily select the functions should be included in the application. The resulting applications are completely portable, in that they can be recompiled for different regions, different MCUs, and different radios.

2. Devices

The Connect stack supports the following device types:

Star coordinator: The star coordinator forms and manages the star or extended star network. The star coordinator also communicates with other range extenders and end nodes. Each Connect star network has a single coordinator.

Star range extender: A device between the star coordinator and one or more star end nodes that can be used to extend the range of the star end nodes. Each range extender can serve up to 32 star end nodes.

Star end node: Joins to a star coordinator or a star range extender.

Direct device: A device able to send and receive messages from other (direct) devices in range on the same PAN, with no star topology restrictions. Such a device does not relay messages.

An example of a Connect network is a network of temperature and humidity sensor end nodes installed throughout a home. Each end node periodically takes a reading and transmits that data either directly to a coordinator (sink) or, for those sensors placed farther away from the coordinator, to a range extender. The range extenders take data from the sensors and forward them to the coordinator. The coordinator not only forms and manages the network, but also sends the compiled data to an environmental management system that is part of another network.

Another example is a topology of two minimally-featured nodes that exchange data in both directions. This topology can be used as a generic wire replacement.

A third example is a topology of N direct devices all in range that exchange data between any pair of two nodes in both directions in a single-hop fashion.

3. Network Topology

The Connect stack supports three topologies, shown in the figure below:

- Point to Point
- Star
- Extended Star
- Single Hop Direct

A Point to Point network provides simple communications between two devices: a star coordinator ( ) and a star end node ( ).

A Star network has a single star coordinator hub communicating with multiple star end nodes. All communication is through the star coordinator.

An Extended Star network includes a star range extender ( ) between the star coordinator and star end node(s) in one or more arms of the star. Communications between the star coordinator and the far star end node(s) flow through one of the star range extenders.

A Single Hop Direct network includes two or more direct nodes ( ) communicating with each other in a single hop fashion.

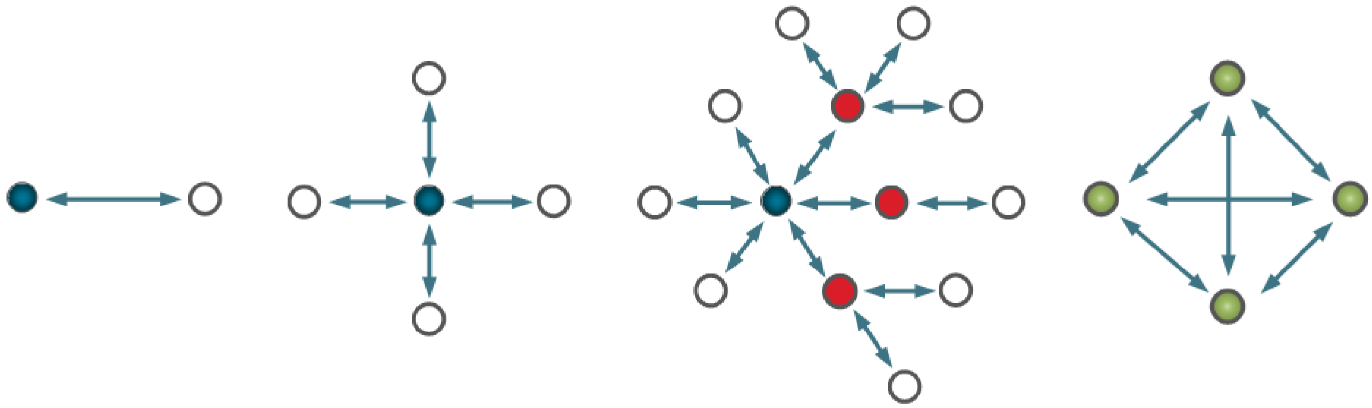


Figure 3.1. Point to Point, Star, Extended Star and Single Hop Direct Network Topologies

4. Stack Structure

The Connect stack provides code organized into three functional layers, as shown in the following figure:

- PHY (physical)
- MAC
- Network

The PHY and MAC layers are based on the IEEE 802.15.4-2006 standard (IEEE 802.15.1-2006 Specification, http://standards.ieee.org/findstds/standard/802.15.4-2006.html). The Network layer is based on a proprietary protocol.

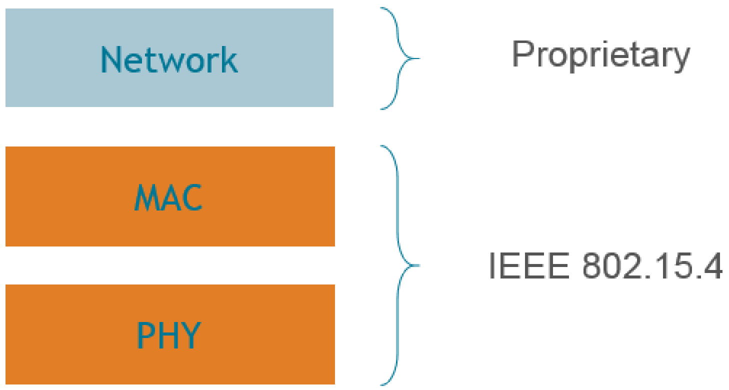


Figure 4.1. Connect Stack Layers

Finally, the Connect Application Framework provides a complete tool and API infrastructure over the underlying stack layers. Functionality within the Application Framework and the Connect stack layers is provided in the form of individual building blocks called plugins. Details of the plugins for each layer are provided in the *Silicon Labs Connect Application Framework API Reference* included in the stack documentation.

4.1 PHY and MAC Layers

The IEEE 802.15.4-2006 specification is a standard for wireless communication that defines the MAC and PHY operating at subGHz frequencies as well as at 250kbps in the 2.4GHz band. 802.15.4 was designed with low power in mind.

The 802.15.4 MAC layer is used for basic message handling and congestion control. The network layer builds on these underlying mechanisms to provide reliable end-to-end communications in the network. The MAC layer includes a CSMA (Carrier Sense Multiple Access) mechanism for devices to listen for a clear channel, as well as a link layer to handle retries and acknowledgement of messages for reliable communications between adjacent devices. The MAC layer also provides security functionality (authentication, encryption, and replay attack protection). The MAC auxiliary header indicates which security scheme is used for that packet. Optional security schemes that can be implemented through plugins include XXTEA (Corrected Block Tiny Encryption Algorithm) and AES (Advanced Encryption Standard). The destination node looks at the auxiliary header and uses the correct security scheme (if it supports it) to de-crypt and authenticate the incoming packet.

One of the characteristics derived from the need for low power and limiting the BER (Bit Error Rate) is enforcing smaller sized packets to be sent over the air. These can be up to a maximum of 127 bytes at the PHY layer. The MAC layer payload can vary depending on the security options and addressing type as illustrated in the figure below.

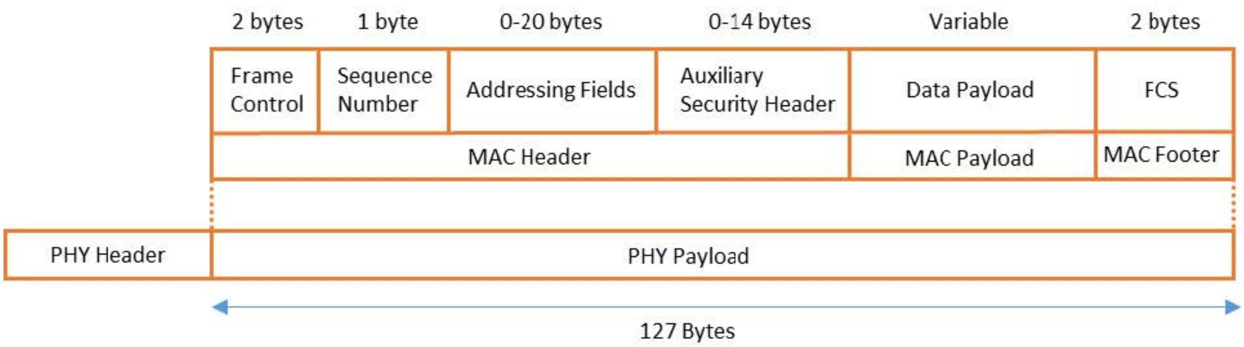


Figure 4.2. 802.15.4 MAC Payload

Nodes are provided with both short (2-byte) and long (8-byte) identifiers. A network is identified by a 2-byte PAN ID.

4.2 Network Layer

For star topology devices, the proprietary network layer provides network formation and full routing support, meaning every node in the network can communicate to any other node in the network in both directions. Routing is transparent to the application layer. Routing is not performed for direct devices.

The network formation functionality offers an association mechanism that, while it is similar to that in the 802.15.4 protocol, has been improved and made more secure by providing a special encrypted association request command not present in the 802.15.4 protocol. Network formation also includes centralized address allocation at the star coordinator. Finally, it provides APIs for commissioning the node network parameters such as node ID, PAN ID, channel, transmission power. Direct devices join a network using the commission-ing APIs.

4.3 Application Framework

The Connect Application Framework leverages the Ember Application Framework v6 and its bookkeeping functionality implemented in callbacks such as `init()`, `tick()`, and `stackStatus()`. Application Framework plugins can provide callbacks and can implement the callbacks of other plugins.

4.4 Functionality Blocks

Every Connect application includes the following functionality blocks. The HAL (hardware abstraction layer) and Simulated EEPROM functionality blocks reside below the Connect stack. The PHY, event system, and message builder/parser functionality blocks are part of the stack itself.

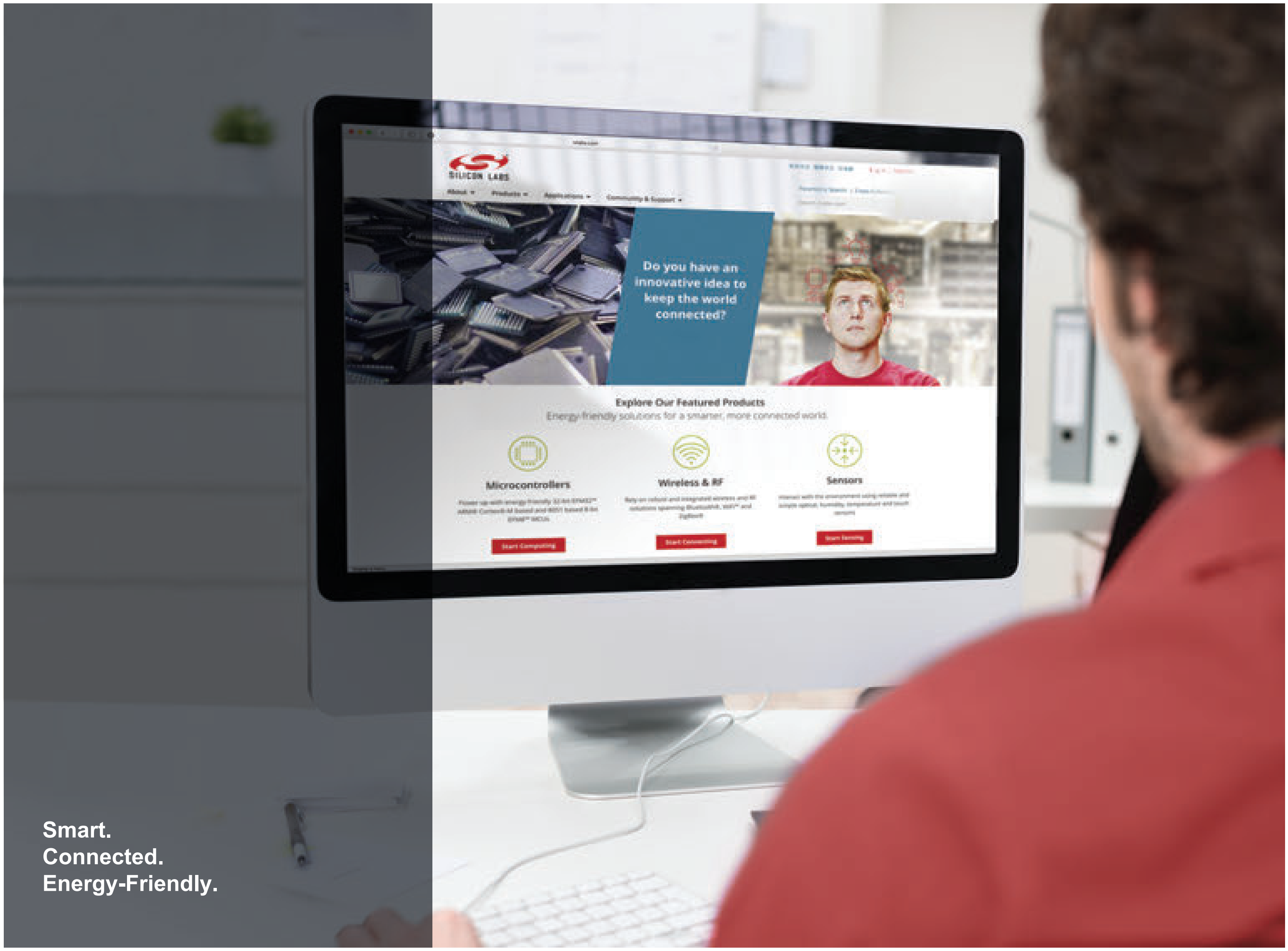
- **HAL:** Drivers for devices and their peripherals such as SPI (Serial Peripheral Interface), UART (Universal Asynchronous Receiver/ Transmitter), timers, and so on.
- **Simulated EEPROM:** Wear-leveled persistent storage of network and application data.
- **Event system:** System that allows the stack and the application to schedule code to run after some specified time interval. Events are also useful when an ISR (Interrupt Service Routine) needs to initiate an action that should run outside the ISR context.
- **PHY:** Software module that interfaces with the transceiver over the SPI bus and provides basic radio TX, RX, and radio sleep functionality.
- **Message builder/parser:** Provides a 15.4-like PHY/MAC packet format builder/parser and a proprietary network layer format builder/parser.
- **Dynamic memory allocation:** A generic lightweight module that provides dynamic memory allocation and garbage collection.

5. Next Steps

See either *QSG138: Getting Started with the Silicon Labs Flex Software Development Kit for the Wireless Gecko (EFR32™) Portfolio* or *QSG137: Getting Started with the Silicon Labs Flex Software Development Kit for the EZR32* for instructions on using Simplicity Studio and the WSTK to develop a Connect-based application using the Flex SDK.

For more details on application development using Silicon Labs Connect, refer to *UG235: Silicon Labs Connect User Guide*.

See *AN902: Building Low Power Networks with the Silicon Labs Connect Stack* provides instructions specific to low-power implementations.



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

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

http://www.silabs.com