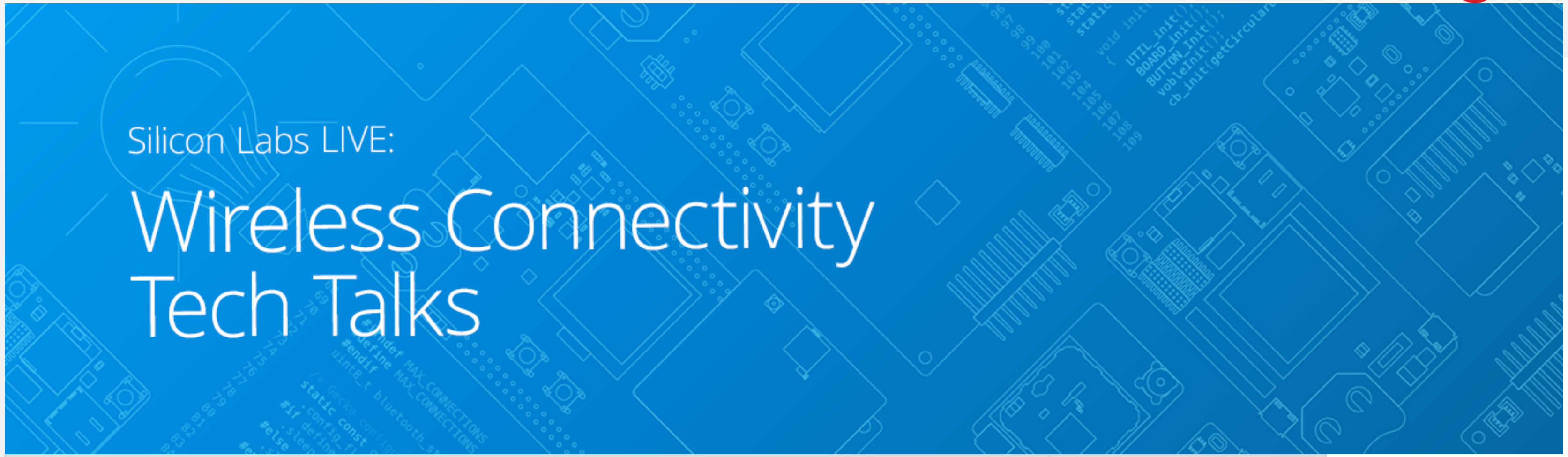
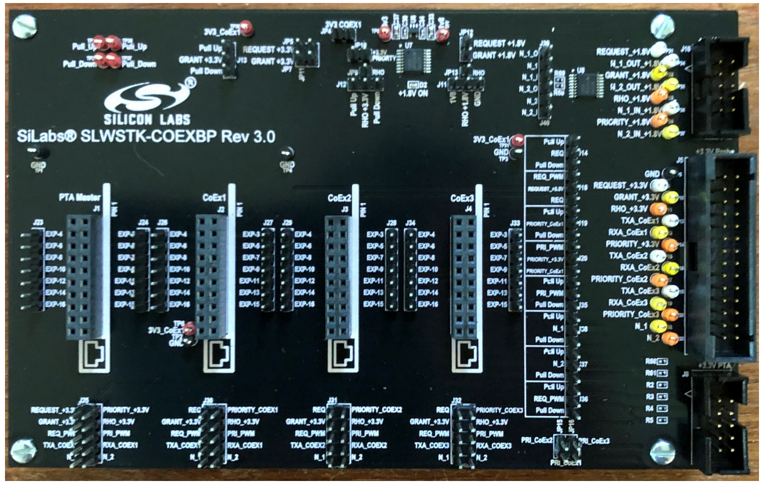
Tech Talks LIVE Schedule – Presentation will begin shortly



Topic	Date
Optimize a Battery Supply Using the Energy Friendly PMIC	Tuesday, June 2
Zigbee Software Structure: Learn about Plugins and Callbacks	Thursday, June 4
Multiprotocol Wireless: Real Application of Dynamic Multiprotocol	Tuesday, June 9
Wireless Coexistence	**Thursday, June 11**
Bluetooth Software Structure: Learn the APIs and State Machines	Tuesday, June 16
Add a Peripheral to a Project in No Time: With 32-bit Peripheral Github Library	Thursday, June 18
OpenThread Software Structure: Learn about Resources and Examples	Tuesday, June 23
Talk with an Alexa: Using Zigbee to Connect with an Echo Plus	Thursday, June 25

Fill out the survey for a chance to win a SLWSTK-COEXBP co-existence eval board!



Find Past Recorded Sessions at:
https://www.silabs.com/support/training

WELCOME

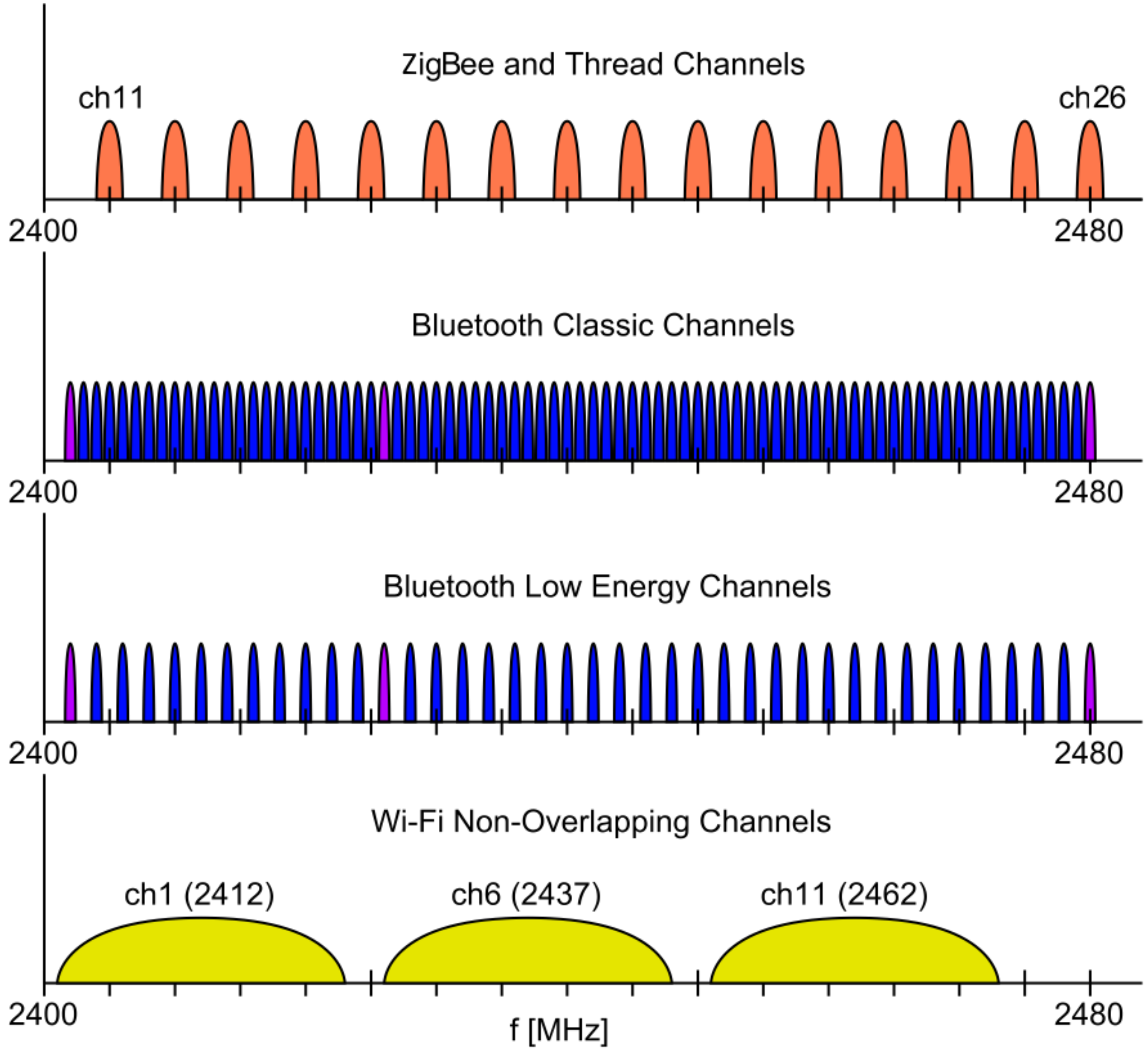
SILICON LABS

Silicon Labs Wireless Co-Existence

JUNE 2020

2.4GHz ISM Band Coexistence Challenge

- Multiple wireless protocols share the same 2.4GHz ISM Band: Wi-FI, Bluetooth, and IEEE 802.15.4 (ZigBee, Thread)
- These wireless protocols have different modulation schemes, channel frequencies and bandwidth but overlap when collocated
- Signals from one wireless protocol look like unwanted noise for the other protocols
- If the desired receive signal is weaker than the noise, the radio will be unable to properly receive messages

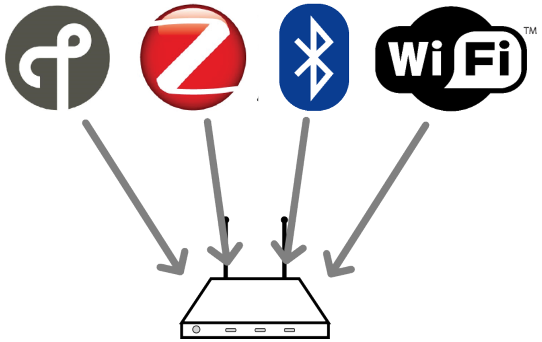


Coexistence Challenges Are Increasing

- Historically, wireless devices seemed to “just work” even without specifically addressing coexistence issues

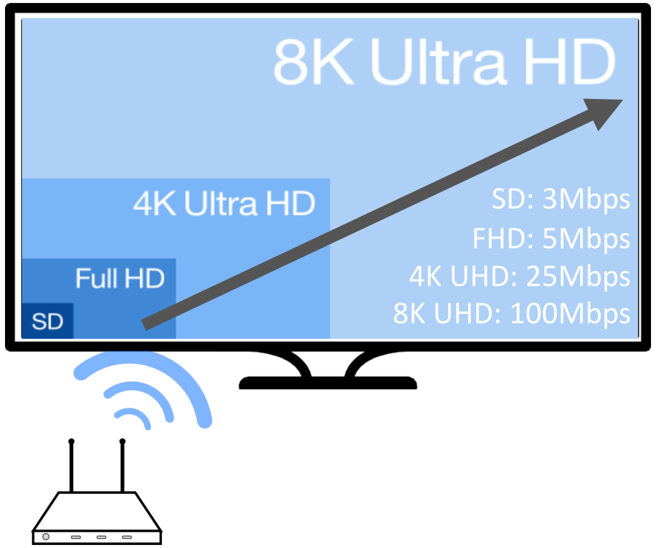
- Unfortunately, this is no longer the case due to three trends:

Greater device integration & radio colocation



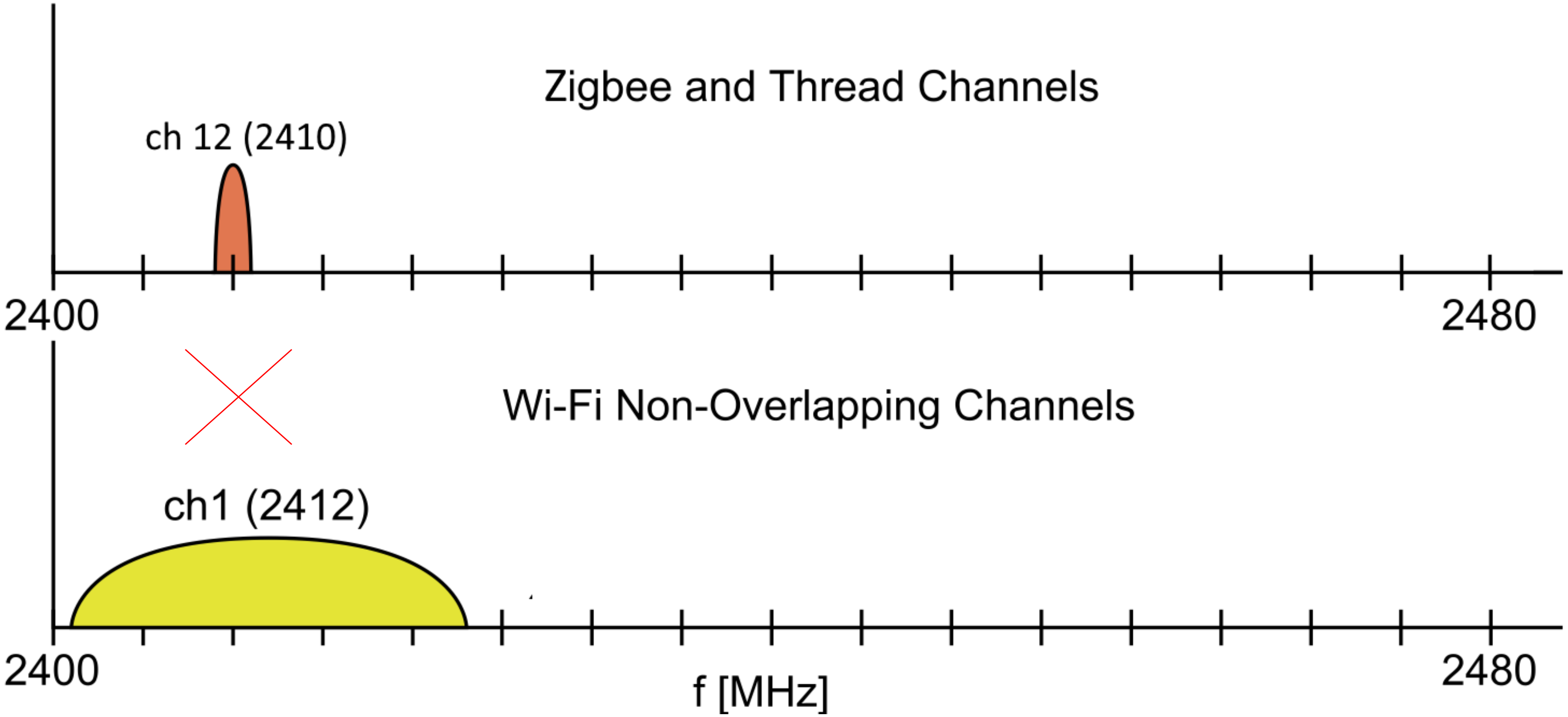
Increased Wi-Fi transmit power (+30dBm)

Larger throughput needed for Wi-Fi streaming



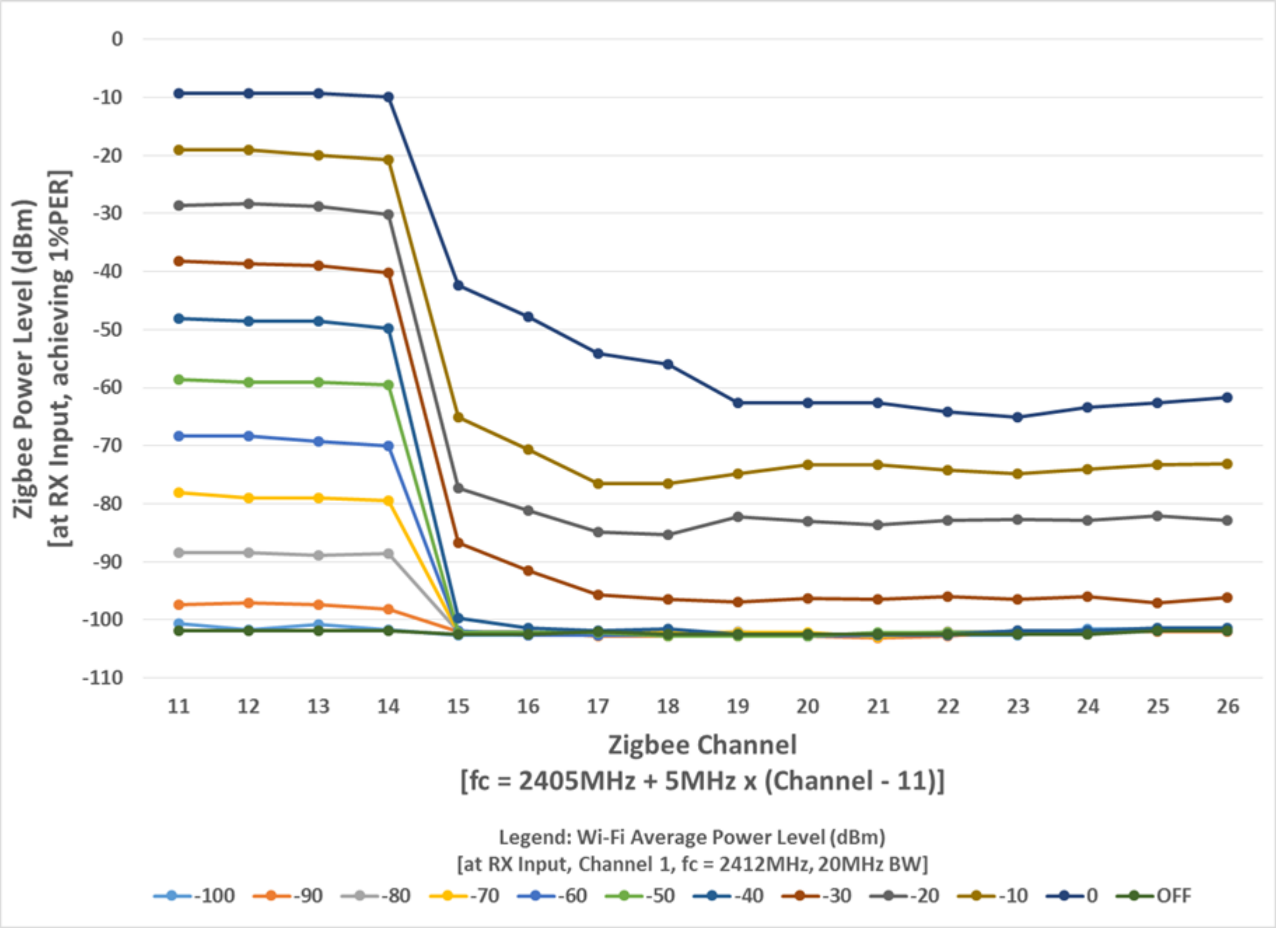
Coexistence Impact: Co-Channel

- 2.4 GHz IoT RX is generally blocked when the 2.4 GHz Wi-Fi interference is co—channel and stronger at the receiver than the signal being received from the remote IoT device
- Zigbee uses Clear Channel Assessment (CCA) to test the channel prior to transmit, and transmit is blocked with energy detection > -75 dBm per 802.15.4 spec



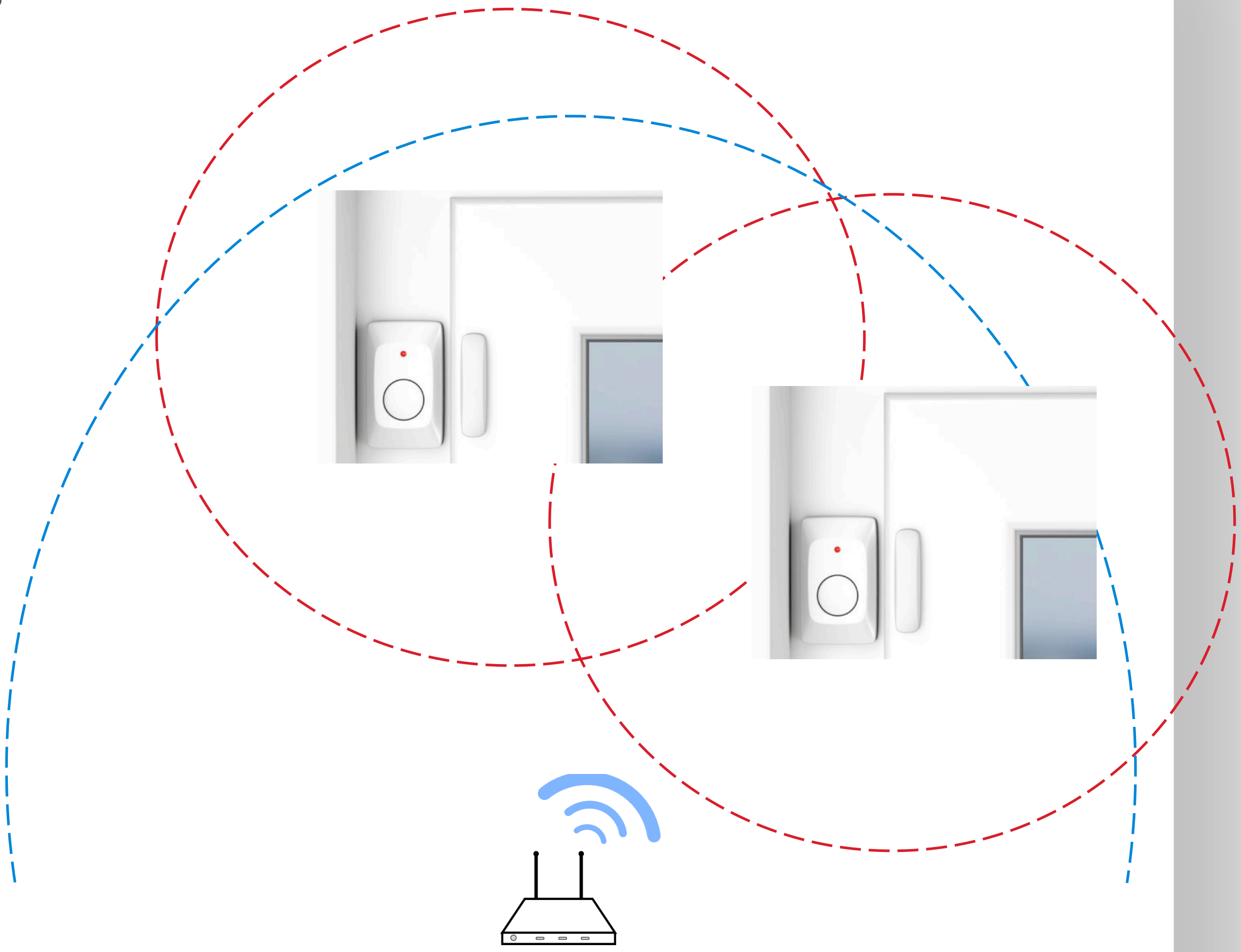
Coexistence Impact: Receiver Blocking / Selectivity

- 2.4 GHz IoT RX sensitivity is degraded by high amplitude 2.4GHz Wi-Fi
  - Even if Wi-Fi and IoT are on different channels, the Wi-Fi energy is still in-band for the LNA, AGC, Mixer, etc.
- In this example, if Wi-Fi energy is below -40 dBm, far-away Zigbee channels are unaffected.
  - This generally covers the cases where the Wi-Fi and IoT are physically separated
- Multi-protocol gateways are another story
  - High power 2.4 GHz Wi-Fi TX in the same enclosure as IoT is difficult to isolate



Coexistence Issues for End Devices

- Use Case:
  - ZigBee security system with battery powered sensors
  - Wi-Fi co-located (within radio range)
  - Wi-Fi traffic is used for streaming videos
- Customer effect:
  - Delayed or missing reports
  - Reduced battery life
- Impact:
  - High rate of retries
  - Latency due to CCA failures
- Strategies to improve performance:
  - Unmanaged Coexistence

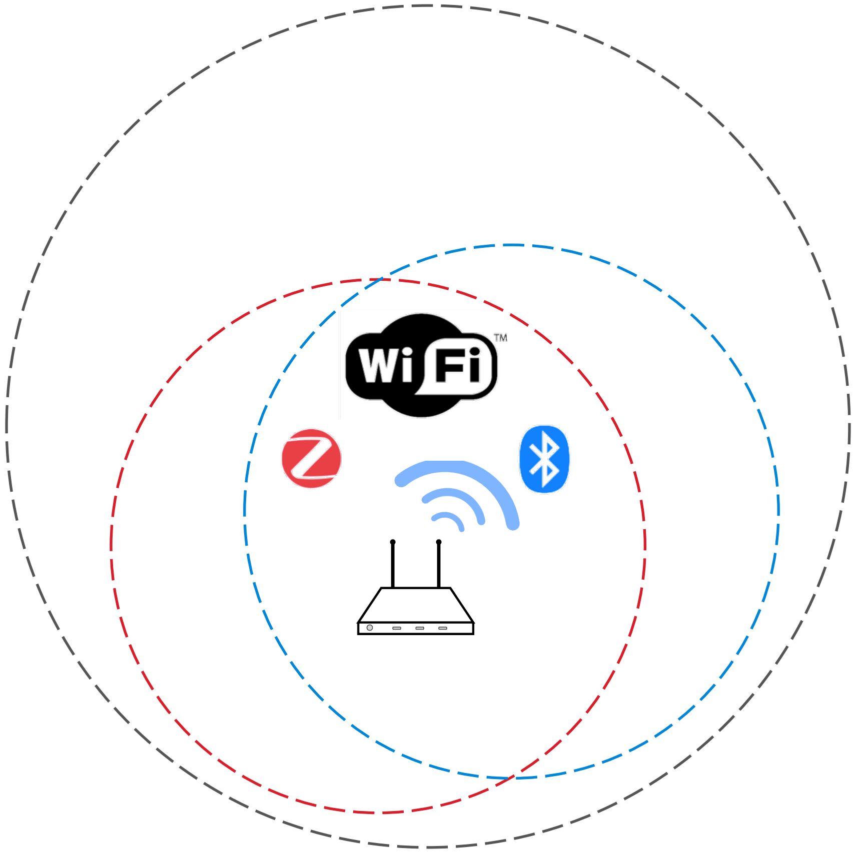


Coexistence Issues for Gateways

- Use Case:
  - ZigBee, Bluetooth, Wi-Fi in a small form factor
  - Device used as Wi-Fi Access Point
  - Device is also used as a home IOT gateway
    - Communicates with sensors (door/window, presence, etc)
    - Communicates with lights, door locks, shade controllers, etc.

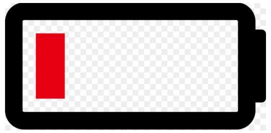
- Customer effect when 2.4G Wi-Fi is under high TX load:
  - Poor command responsiveness for Zigbee devices
  - Delayed or missing Zigbee reports
  - Dropped Bluetooth connections
  - Reduced battery life on Zigbee devices due to retries

- Strategies to Improve Performance:
  - Unmanaged coexistence
  - Managed coexistence



Why retries need to be minimized

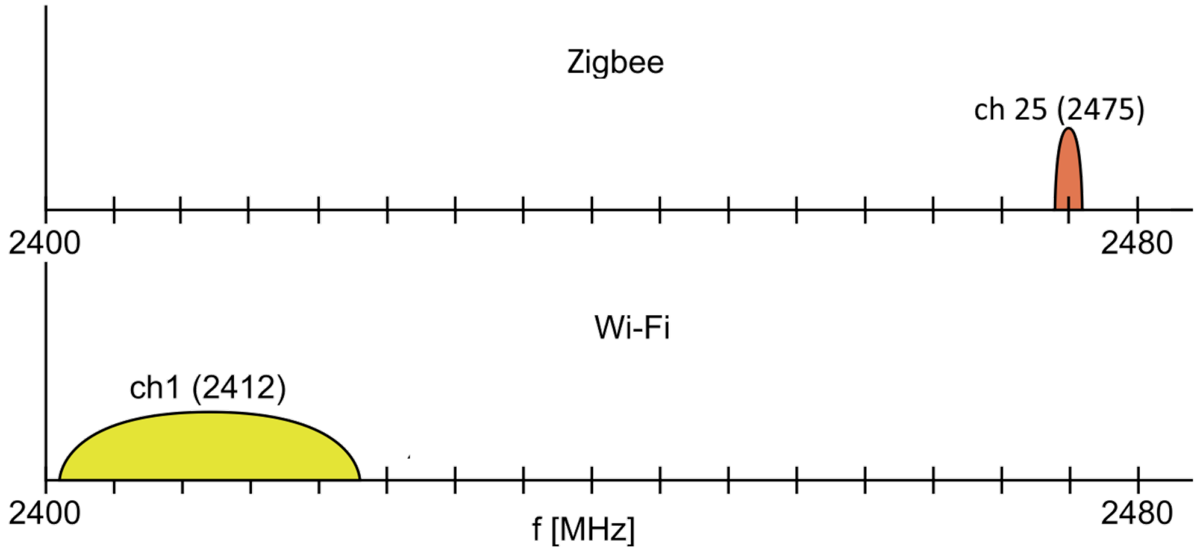
- Battery powered devices
  - Messages from window sensors, door locks, etc. are asynchronous: they may arrive at any time
  - Key concern is battery life
    - Wi-Fi Interference leads to more retries
    - Retries lead to more power consumption
    - Short battery life = more frequent replacement



- Latency
  - Some applications have latency requirements (e.g. light must turn on in <200ms)
  - CCA failures and retries increase latency

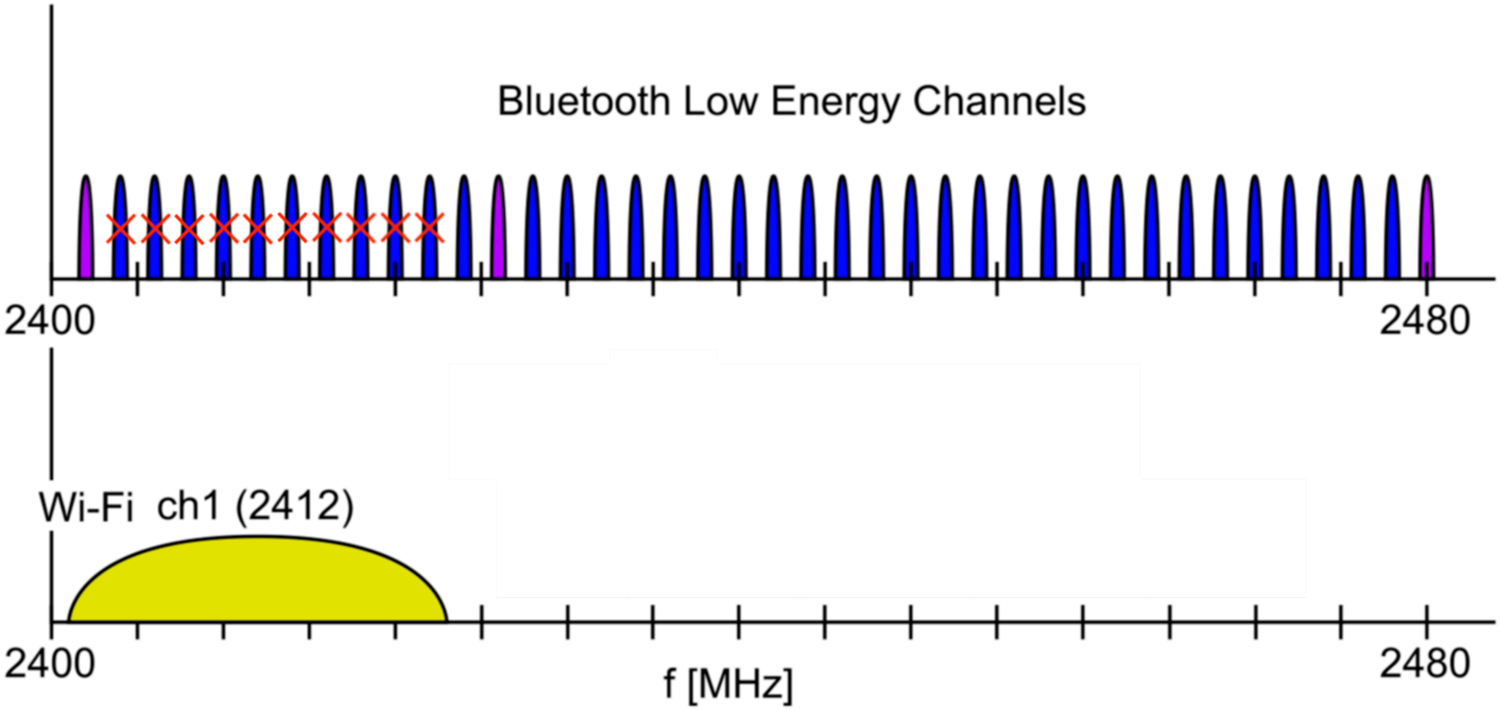
Improving Coexistence: Unmanaged

- Frequency separation
  - If possible, select ZigBee channel as far from 2.4 GHz Wi-Fi channel as possible
  - Use Bluetooth Low Energy channel map
- Antenna Isolation
  - Provide as much isolation between IoT and 2.4 GHz Wi-Fi antennas as possible
  - If IoT and 2.4 GHz are in separate physical units, implement install guidelines for minimum clearance
- Use 20 MHz Wi-Fi bandwidth (avoid 40 MHz)
- Rely on protocol Retry Mechanisms



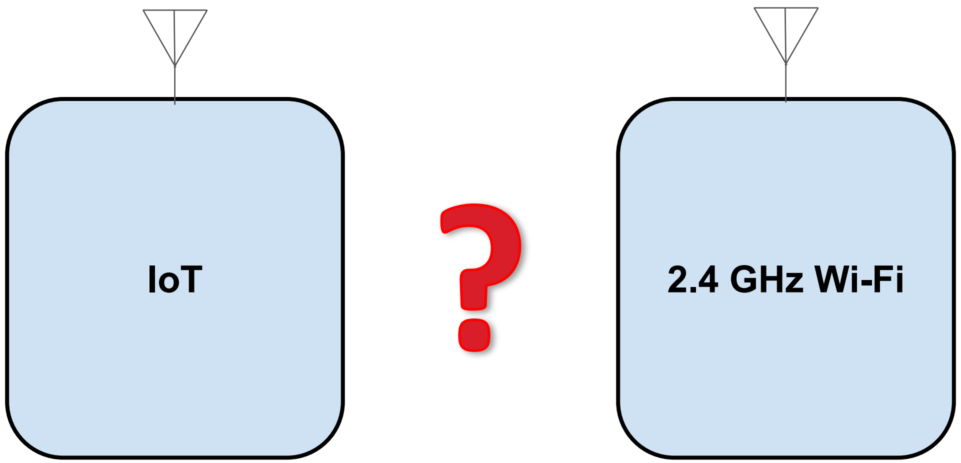
Unmanaged Coexistence: Bluetooth Data Channel Management

- Bluetooth Low Energy uses two types of frequency diversity
  - Advertising/beacons uses three channels – low, mid, and high advertising channels
  - Connection master can dynamically configure the channel map for frequency hopping on data channels to avoid channels occupied by other protocols
- If EFR32 is the connection master:
  - `gecko_cmd_le_gap_set_data_channel_classification()`



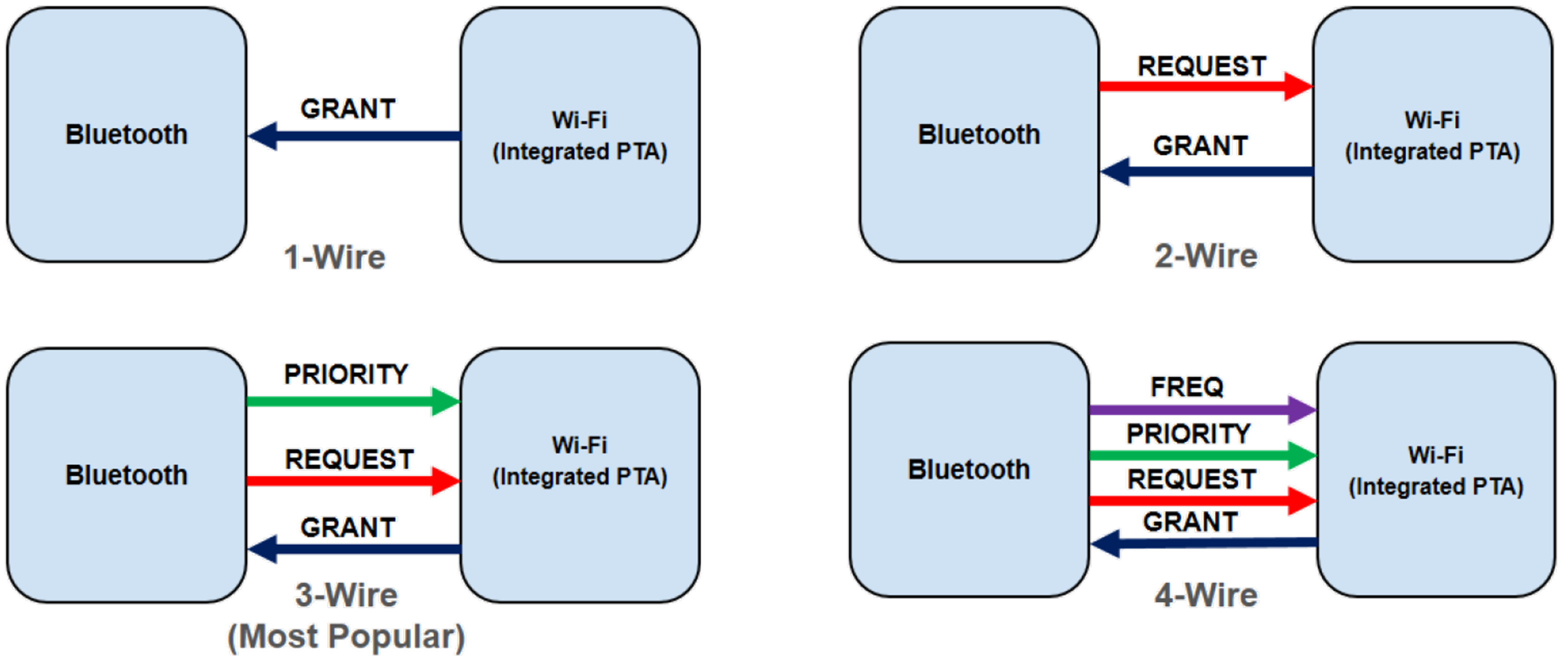
Improving Coexistence: Managed

- Separate the radio activity in time
- Requires coordination between the radios
- Is there an easy way for the different radios sharing the 2.4 GHz band within a single product to communicate with each other to accomplish this?



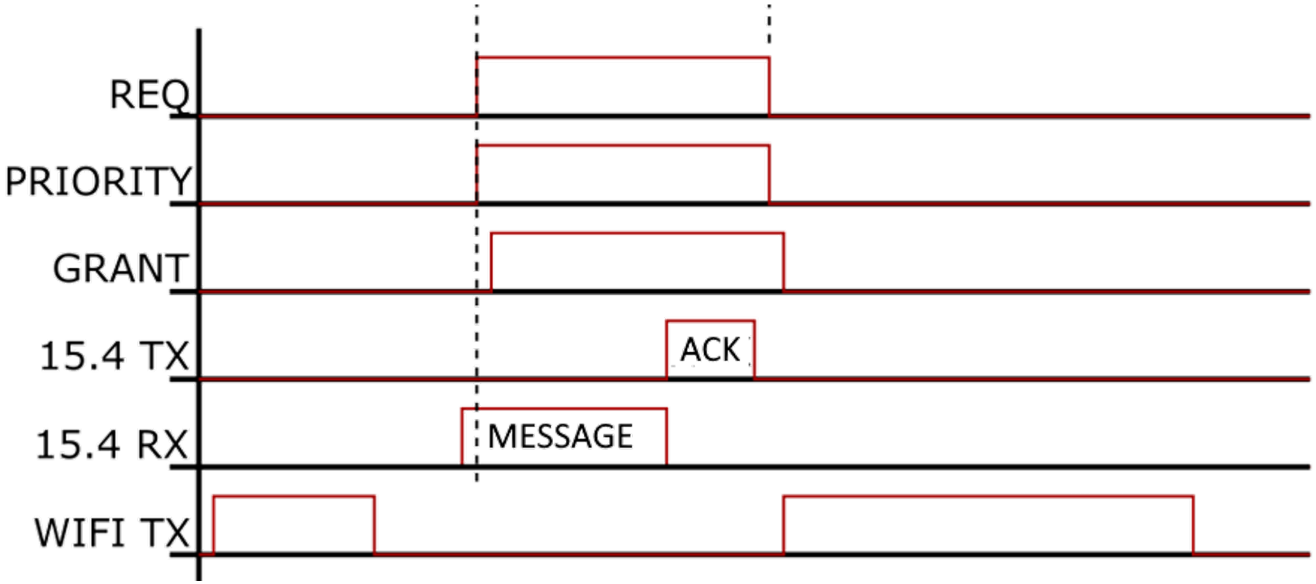
Managed Coexistence: Packet Traffic Arbitration

- Packet Traffic Arbitration (PTA) is a *recommendation* provided in 802.15.2: “Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands”.
- It was originally written for Bluetooth Classic/Wi-Fi coexistence and includes four different wiring configurations



Packet Traffic Arbitration Basics

- IoT device asserts REQUEST and optionally asserts the PRIORITY signal
- If the Wi-Fi device can grant airtime, it asserts the GRANT signal back to the IoT device(s)
- The Wi-Fi device is expected to stop transmitting prior to asserting GRANT, and is expected to not begin a new transmission while GRANT is asserted
- When the IoT transaction is completed, the IoT device de-asserts REQUEST and the Wi-Fi device follows by de-asserting GRANT.



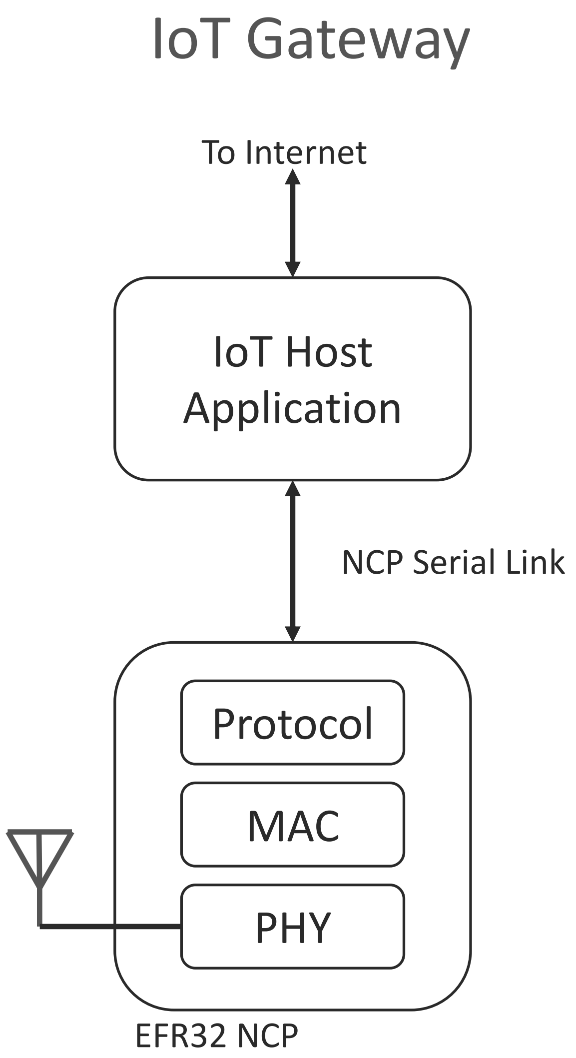
Managed Coexistence: Packet Traffic Arbitration

- Because PTA is based on recommendations and not requirements, there is variability of implementation by different Wi-Fi and Bluetooth IC manufacturers.
- To mitigate against this, Silicon Labs' interface is flexible in terms of the number of signals used, signal drive, and signal polarity.
  - 1- to 3- wire supported (4-wire not supported)
  - 3-wire recommended
- We have tested our IoT radio stacks across many Wi-Fi chips from many Wi-Fi vendors
- The results show a clear benefit to using managed coexistence

NOTE: Managed coexistence (PTA) is recommended to be deployed in addition to (not in place of) the unmanaged coexistence techniques previously mentioned!

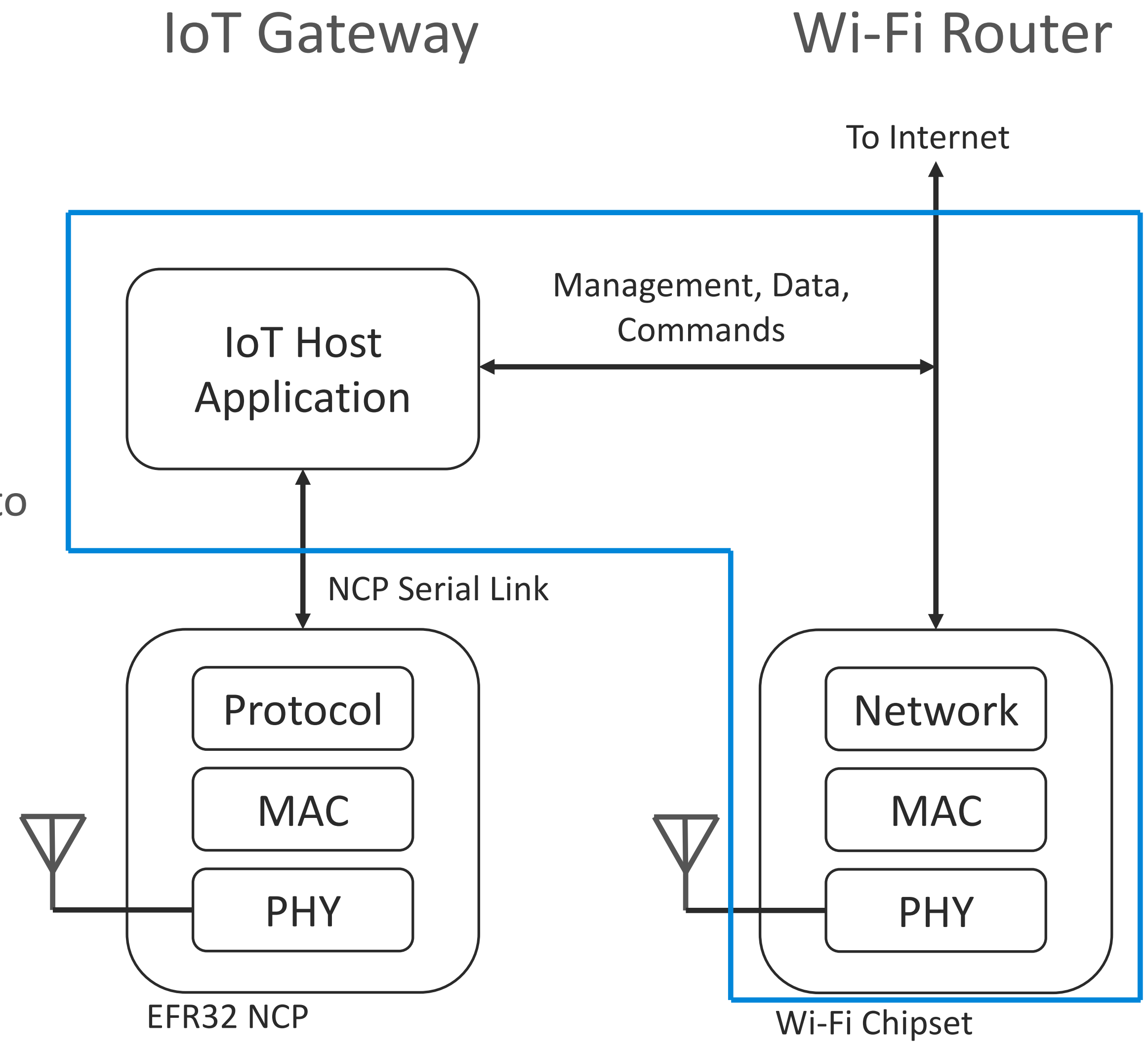
IoT Gateway – Managed Coex Example

- The IoT Gateway function is typically split between a network co-processor (NCP) and a host application processor
  - EFR32 SoCs are well suited to run the NCP firmware, which handles the low-level protocol stack functionality
  - Application-level processing is done on a resource-rich host
- Both the NCP firmware and the Gateway Application code are available as examples in our Bluetooth and Zigbee SDKs
  - Customers typically only wish to modify the application code



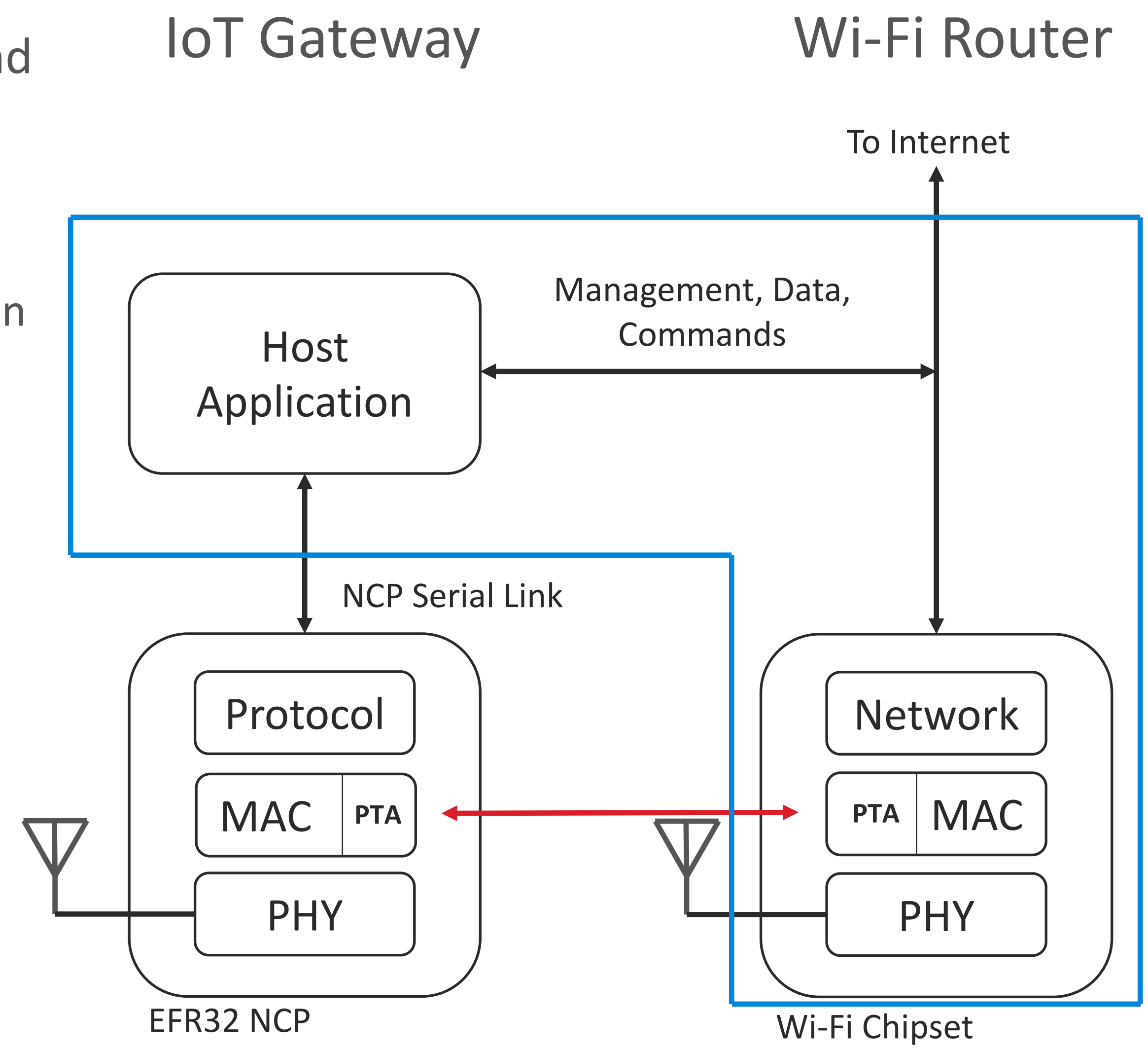
IoT Gateway – Managed Coex Example

- The ZigBee host application is commonly run directly on the Wi-Fi host processor
  - Because the ZigBee Gateway host application is lightweight, it can run on a Wi-Fi host processor without affecting Wi-Fi performance
  - This also provides an easy method of connectivity to the Internet



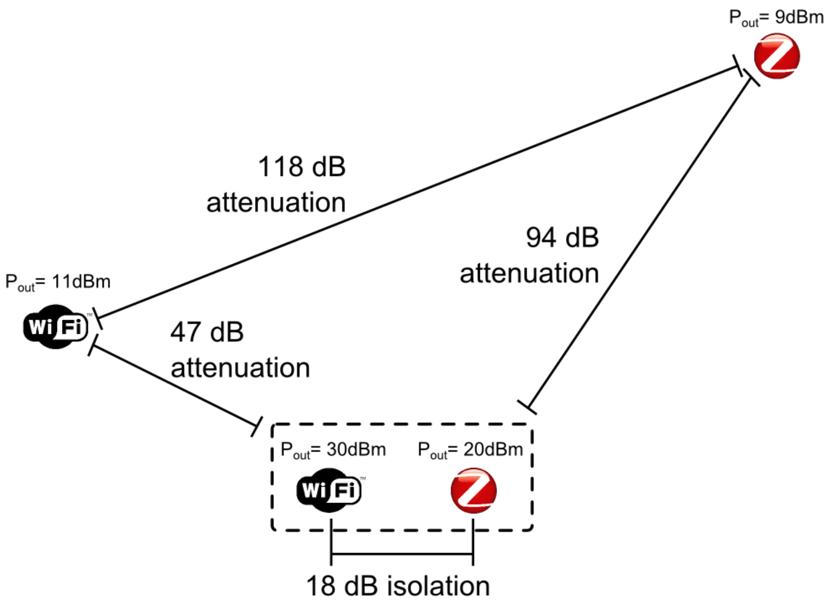
IoT Gateway – Managed Coex Example

- Connect PTA signals between the EFR32 chip and the Wi-Fi chip so that the radios can coordinate
  - Note that PTA is implemented at the MAC layer: Users do not have to rewrite any application code in order to use it



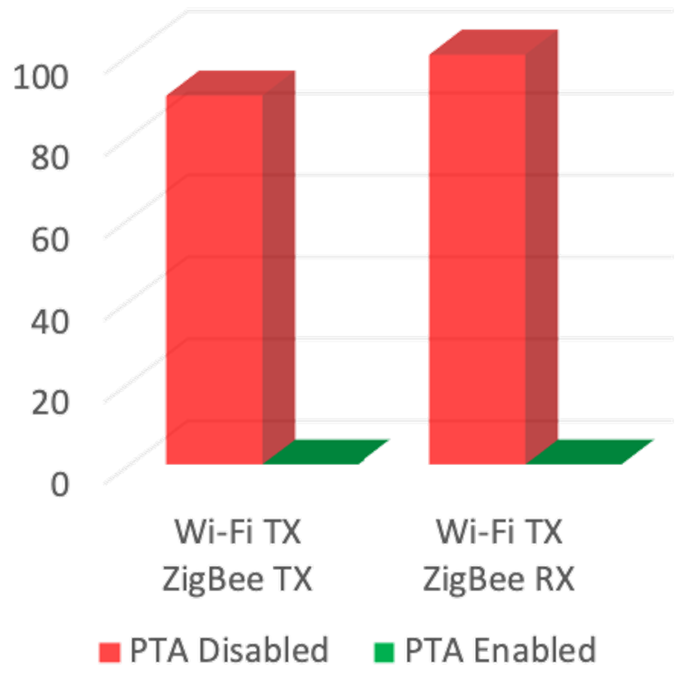
Managed Coexistence: Packet Traffic Arbitration (PTA)

- Use PTA signals (REQUEST, PRIORITY, and GRANT) to coordinate Wi-Fi and Zigbee radios TX and RX operations
- Dashed box represents a co-located Wi-Fi access point + ZigBee gateway
- Traffic is simultaneously pushed through the Wi-Fi link and the ZigBee network



Test conditions: Wi-Fi CH1, 20Mbps UDP, MCS7, 20MHz BW; Zigbee CH24 at 10 kbps

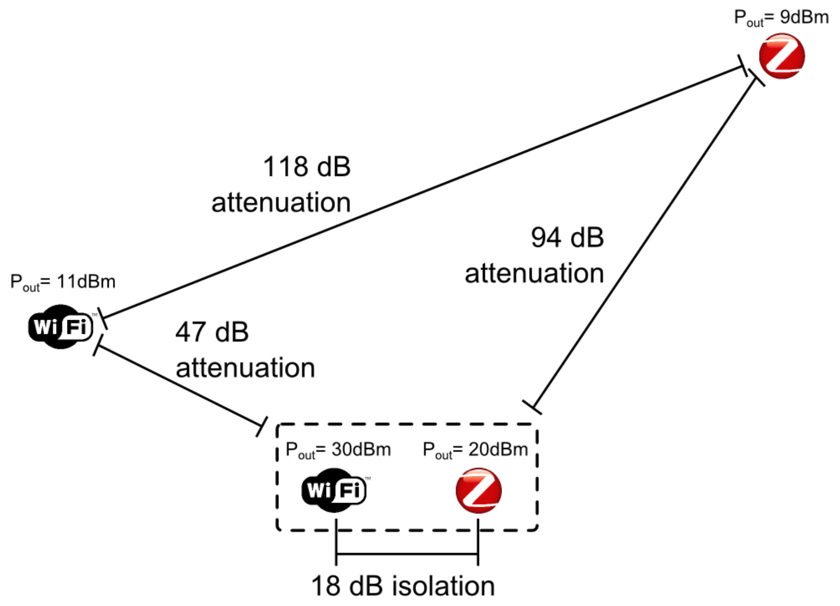
ZigBee Message Failure Rate vs. Test Conditions



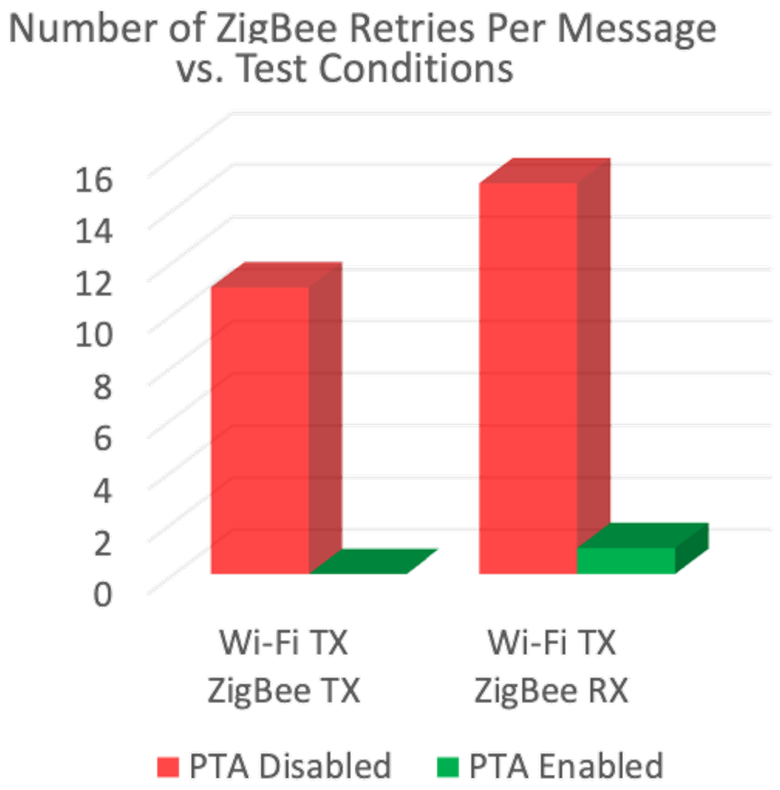
Enabling PTA decreases Zigbee message failure rate from >90% to 0%, results in lower latency

Advantages of using Silicon Labs PTA solution (2)

- Dashed box represents a co-located Wi-Fi access point + ZigBee gateway
- Traffic is simultaneously pushed through the Wi-Fi link and the ZigBee network



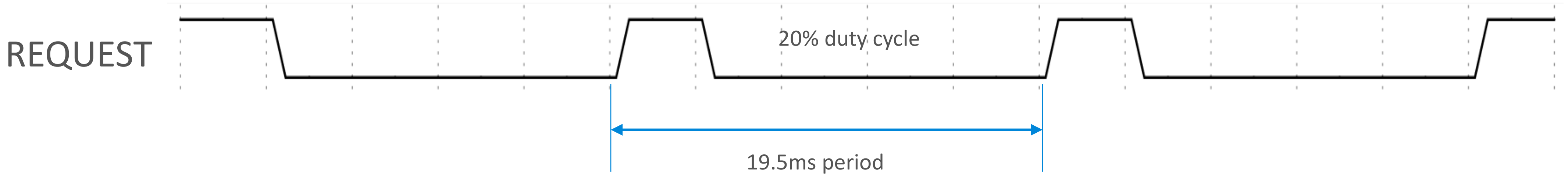
Test conditions: Wi-Fi BW at 20 MHz, 20 Mbps; Zigbee at 10 kbps



Enabling PTA reduces Zigbee retry traffic, results in increased battery life

PTA Strategies: REQUEST PWM

- If co-located 2.4 GHz Wi-Fi TX duty cycle is high, the IoT device may not be able to hear any packets, which makes managed coexistence ineffective for IoT RX
- IoT device needs to successfully receive the preamble and sync of the incoming packet in order to determine that an IoT packet is incoming to assert REQUEST
  - Zigbee: 160us
  - Bluetooth Low Energy (1Mbps PHY): 40us
- One solution to this is to have the IoT device(s) make periodic PTA requests to establish regular listening windows during which the Wi-Fi device will not transmit
- REQUEST PWM makes receiving inbound ZigBee packets and Bluetooth advertising more deterministic. Maximum impact on the 2.4 GHz Wi-Fi throughput is the duty cycle percentage



PTA Strategies: REQUEST PWM

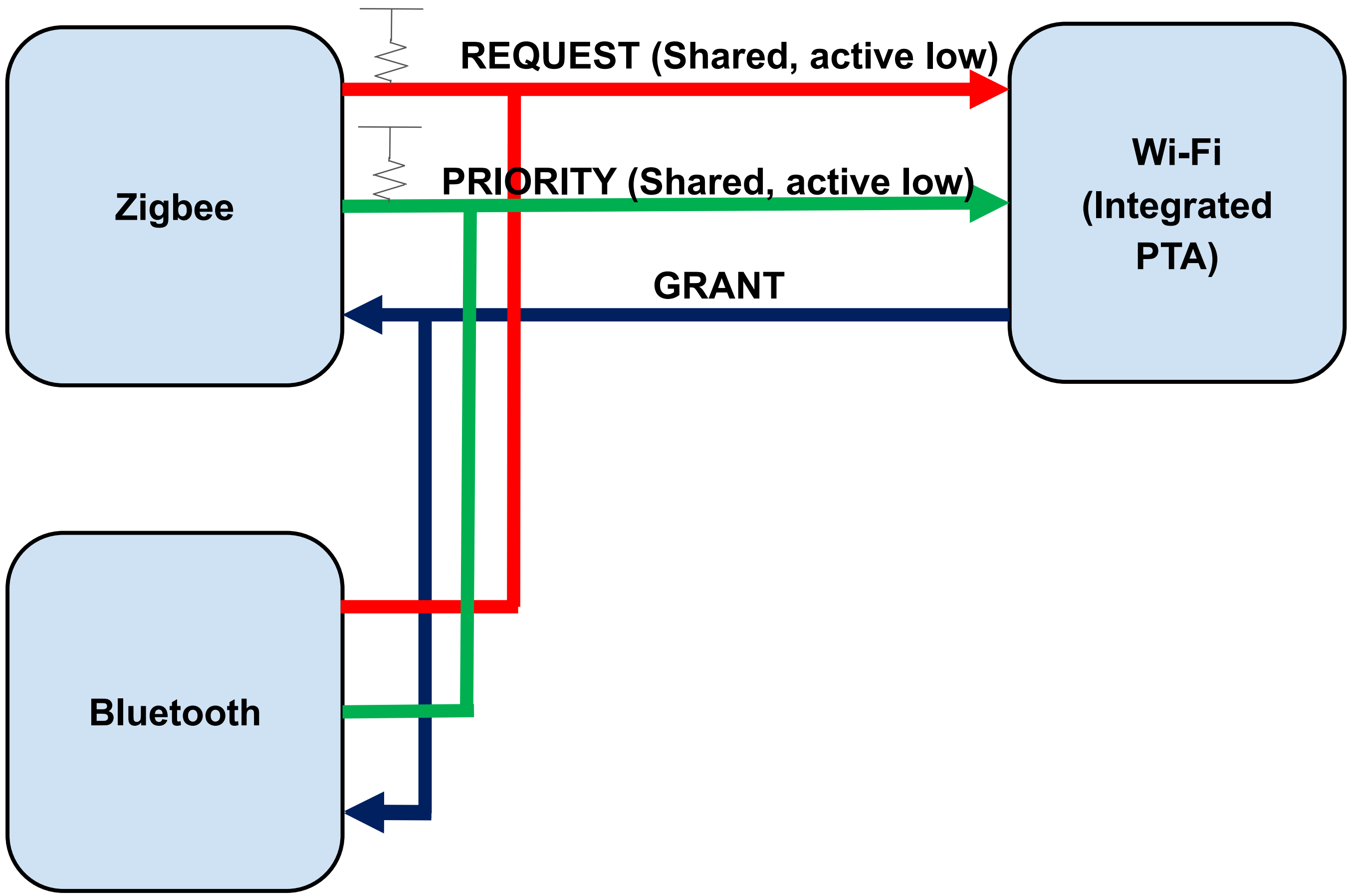
- Here's an example of high duty cycle Wi-Fi with successful ZigBee RX due to a 19.5ms, 20% duty cycle REQUEST PWM
- Green arrows show successfully received packets
- Red boxes surround packets that were transmitted from the remote device but not received

PTA Strategies: Priority

- For static priority, the state of the PRIORITY signal communicates the level of PRIORITY of the IoT transaction
  - Example: ZigBee TX = low priority, ZigBee RX = high priority
- For directional PRIORITY, the PRIORITY signal is either high or low for a typically 20µs duration after REQUEST asserted, but switches to low during receive operation and high during transmit operation.
- PRIORITY provides the Wi-Fi PTA master more insight into the IoT request so that it can more efficiently balance IoT requests against Wi-Fi priorities
- Our Zigbee SDK has a priority escalation feature, which can escalate a low priority request to high priority after a specified number of consecutive failures

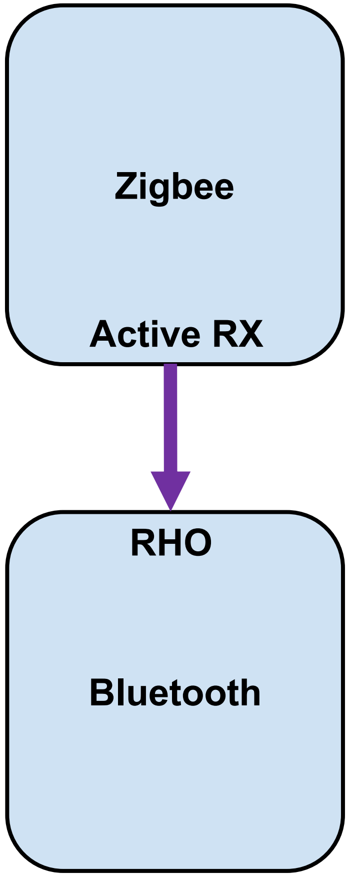
PTA Strategies: Shared PTA

- Silicon Labs provides the ability to share the PTA among several EFR32 IoT devices
- REQUEST (and optionally PRIORITY) can be implemented with an open source or open drain driver on the EFR32 with an external pullup/pulldown resistor
- The Wi-Fi PTA master only sees a single REQUEST signal, so it doesn't care whether there are multiple IoT devices
- The EFR32 devices will poll REQUEST prior to asserting, so this also allows arbitration between the IoT protocols on the separate EFR32 devices



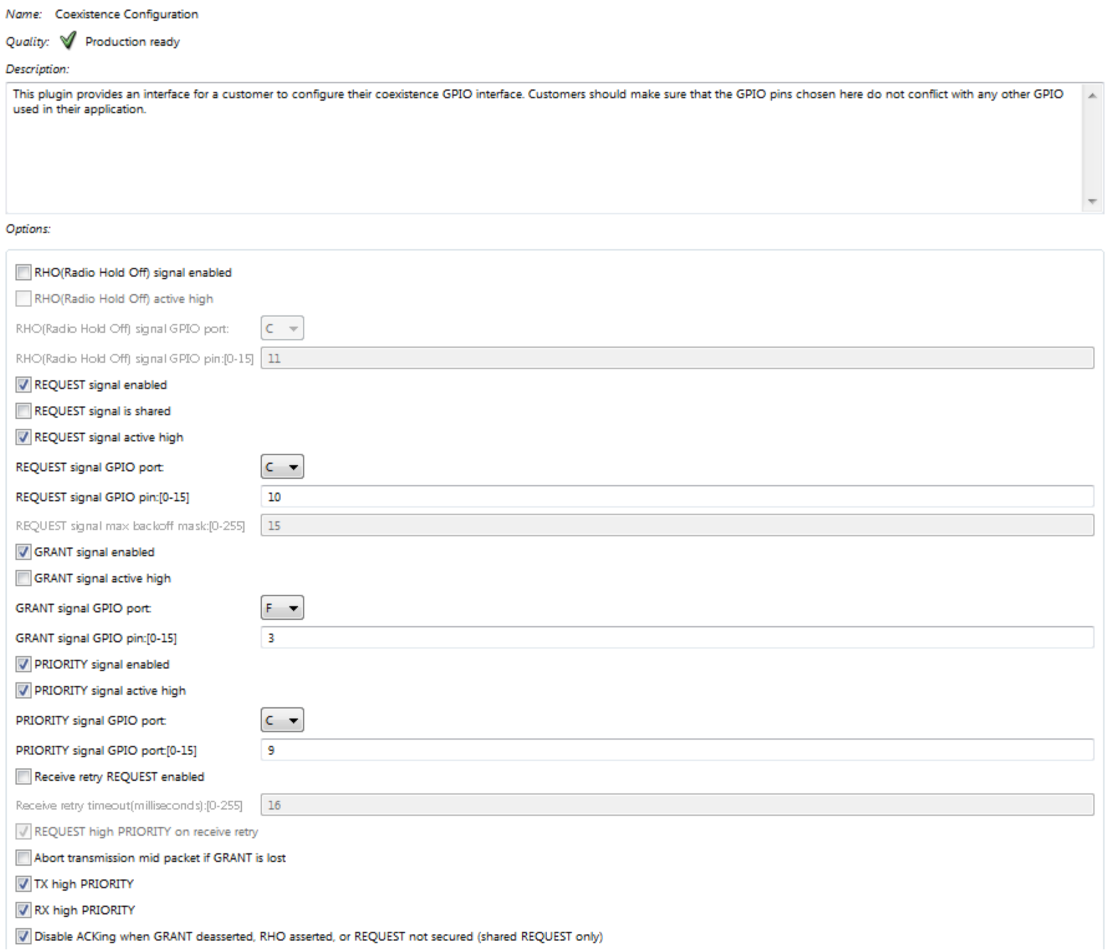
PTA Strategies: Radio Hold-Off

- Our stacks provide the ability to designate a Radio Hold-Off (RHO) pin
- This pin prevents the device from transmitting while asserted
- Use Cases
  - Provides a separate method of prioritizing IoT behavior between radios (Zigbee RX activity -> Bluetooth RHO)
  - Can also be useful to implement on a multi-radio design to avoid FCC co-located transmitter requirements



EFR32 PTA: Zigbee Configuration

- App Note AN1017 – Zigbee Coexistence
  - An advanced version of this app note can be provided under NDA
    - Includes detailed Wi-Fi Coexistence/PTA test setup and results
- Coexistence configured via Appbuilder plugin
- Configure the GPIO to match the PTA hardware configuration and WiFi chip expectations;
  - REQUEST/GRANT/PRIORITY enabled, which pin and active high or low
  - Radio Hold-Off enable/setup
  - Default priority for RX and TX
- Example shown is PTA for single EFR32 with typical time-slotted 3-wire Wi-Fi/PTA



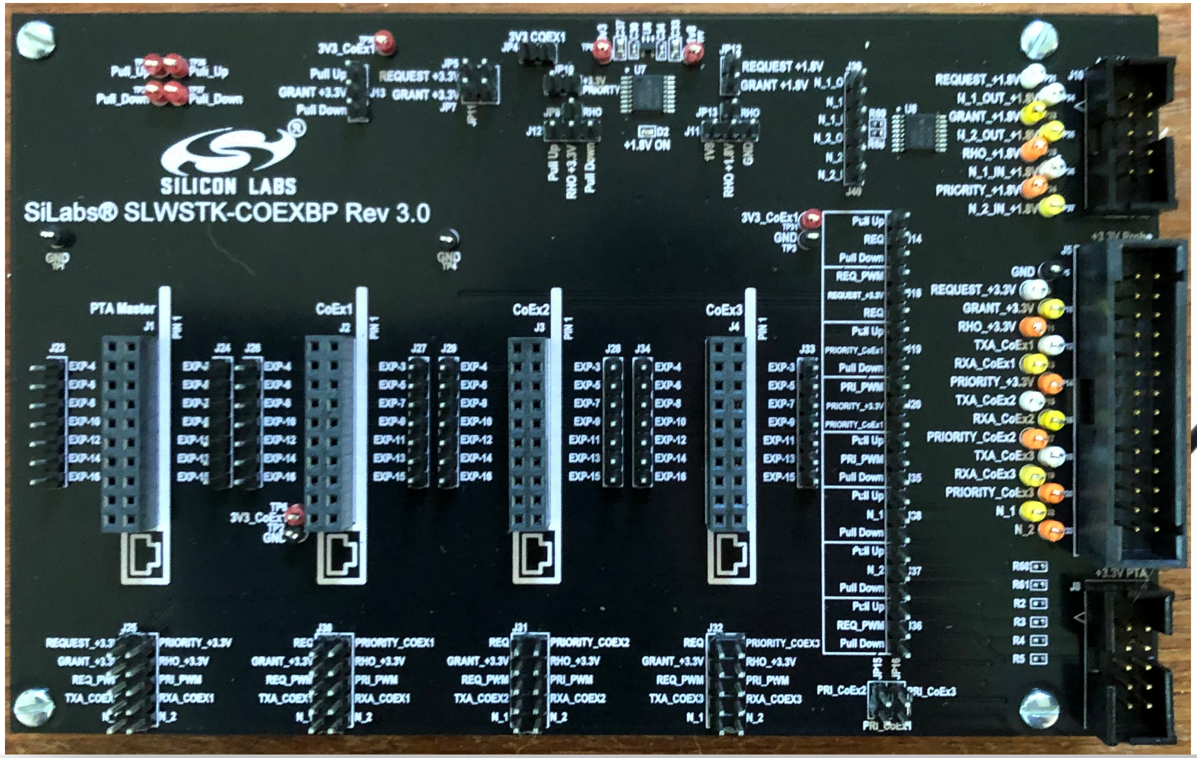
EFR32 PTA: Bluetooth Configuration

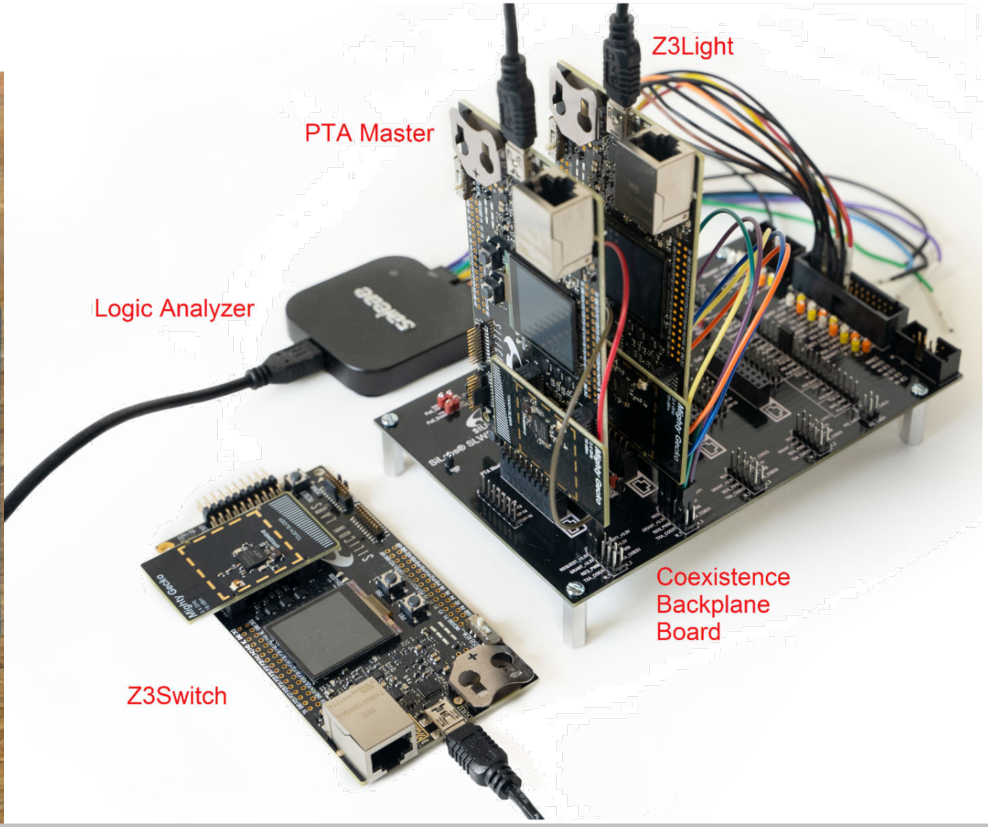
- App Note AN1128 – Bluetooth Coexistence
  - An advanced version of this app note can be provided under NDA
    - Includes detailed Wi-Fi Coexistence/PTA test setup and results
- Coexistence configured via header file (hal-config.h)
- Configure the GPIO to match the PTA hardware configuration and WiFi chip expectations;
  - REQUEST/GRANT/PRIORITY enabled, which pin and active high or low
  - Radio Hold-Off enable/setup
  - Default priority
- Example shown is PTA for single EFR32 with typical time-slotted 3-wire Wi-Fi/PTA

```
// $[COEX]
#define HAL_COEX_ENABLE                        (1)

#define BSP_COEX_REQ_PIN                       (10U)
#define BSP_COEX_REQ_PORT                      (gpioPortC)
#define BSP_COEX_REQ_ASSERT_LEVEL              (1)
#define HAL_COEX_REQ_WINDOW                    (50U)
#define HAL_COEX_REQ_SHARED                    (0)
#define HAL_COEX_REQ_BACKOFF                   (15U)

#define BSP_COEX_GNT_PIN                       (3U)
#define BSP_COEX_GNT_PORT                      (gpioPortF)
#define BSP_COEX_GNT_ASSERT_LEVEL              (0)
#define HAL_COEX_TX_ABORT                      (0)

#define BSP_COEX_PRI_PIN                       (12U)
#define BSP_COEX_PRI_PORT                      (gpioPortD)
#define BSP_COEX_PRI_ASSERT_LEVEL              (1)
#define HAL_COEX_PRIORITY_DEFAULT              (1)
#define HAL_COEX_PRI_SHARED                    (0)

#define HAL_COEX_PWM_DEFAULT_ENABLED           (0)
#define HAL_COEX_PWM_REQ_PERIOD                (39U)
#define HAL_COEX_PWM_REQ_DUTYCYCLE             (20U)
#define HAL_COEX_PWM_PRIORITY                  (0)
```

Coexistence Development Kit (SLWSTK-COEXBP)

- SLWSTK-COEXBP is designed as a platform to demonstrate and prototype PTA managed coexistence with EFR32
- Supports up to three EFR32 PTA slave devices (implemented on a single WSTK each)
- Supports an EFR32 based PTA master emulator or header connections to an external Wi-Fi PTA master





Additional Information

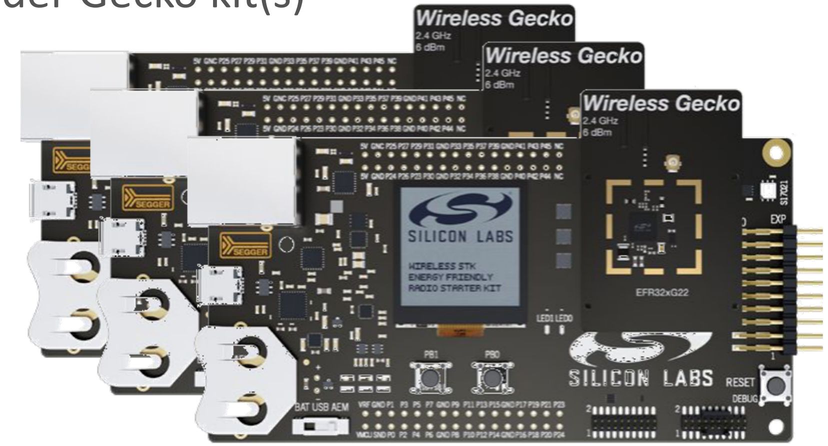
- AN1017: Zigbee Coexistence with Wi-Fi https://www.silabs.com/documents/public/application-notes/an1017-coexistence-with-wifi.pdf
- AN1128: Bluetooth Coexistence with Wi-Fi https://www.silabs.com/documents/public/application-notes/an1128-bluetooth-coexistence-with-wifi.pdf
- UG350: Silicon Labs Coexistence Development Kit (SLWSTK-COEXBP) https://www.silabs.com/documents/public/user-guides/ug350-coexistence-development-kit.pdf
- Contact Silicon Labs Support via our Support Portal
  - Visit: https://siliconlabs.force.com

Conclusions

- Unmanaged coexistence can make IoT networks perform better when in proximity to Wi-Fi networks
- On designs with co-located 2.4 GHz radios, the combination of unmanaged and managed coexistence (PTA) can provide deterministic IoT performance by:
  - Bounding the remote retries for battery powered ZigBee nodes
  - Ensuring stable Bluetooth connections
  - Allowing the deterministic reception of Bluetooth beacons
- Silicon Labs offers built-in support for PTA on both the Bluetooth and ZigBee stacks for EFR32

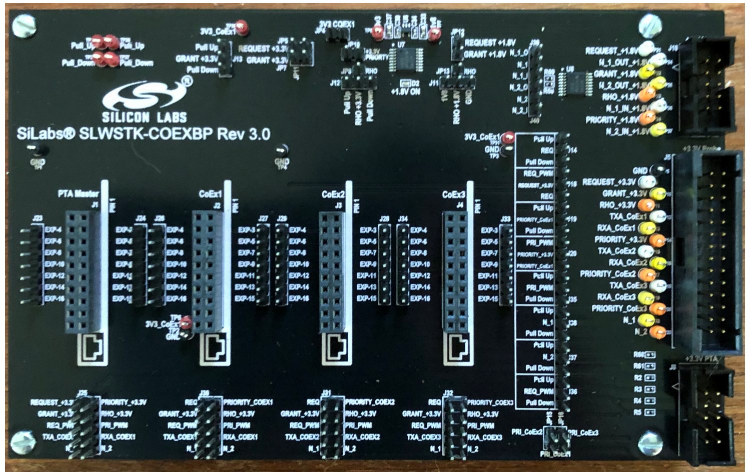
Getting Started

1. Order Gecko kit(s)



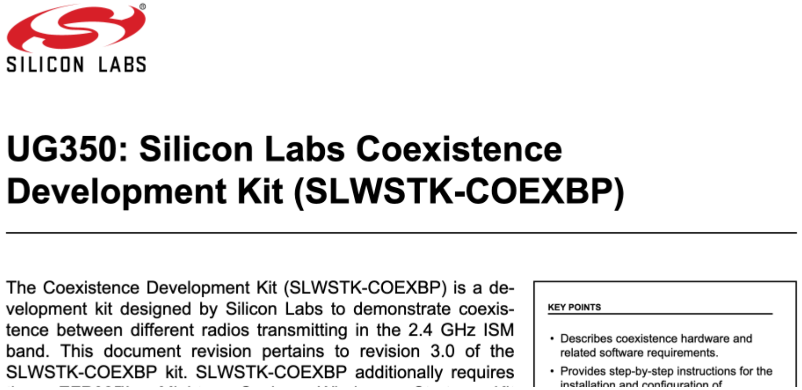
SLWSTK6000B

2. Order SLWSTK-COEXBP



3. Download and install Simplicity Studio.

4. Follow the instructions in UG350.



SILICON LABS

UG350: Silicon Labs Coexistence Development Kit (SLWSTK-COEXBP)

The Coexistence Development Kit (SLWSTK-COEXBP) is a development kit designed by Silicon Labs to demonstrate coexistence between different radios transmitting in the 2.4 GHz ISM band. This document revision pertains to revision 3.0 of the SLWSTK-COEXBP kit. SLWSTK-COEXBP additionally requires

KEY POINTS

- Describes coexistence hardware and related software requirements.
- Provides step-by-step instructions for the

BG22 Virtual Workshop

Learn how to develop and deploy more powerful, efficient, and secure IoT products with your own BG22 Thunderboard – free for all registrants!

New Sessions Open for June

10:00AM –11:30 AM CST - T, W, Th

(Other sessions available for Asia Pacific and Europe)

Register today! https://www.silabs.com/about-us/events/virtual-bluetooth-workshop

works with

BY SILICON LABS

SEPTEMBER 9–10, 2020 | VIRTUAL

workswith.silabs.com

TWO DAYS OF TECHNICAL TRAINING FROM BEGINNER TO ADVANCED

Q & A Session