

Bluetooth® Toolbox Release Notes



MATLAB®



How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Bluetooth® Toolbox Release Notes

© COPYRIGHT 2022 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

R2022a

Introducing Bluetooth Toolbox	1-2
Bluetooth physical layer processing	1-2
Coexistence modeling between Bluetooth and WLAN	1-2
Bluetooth location and direction-finding and ranging capabilities	1-3
Bluetooth transmitter and receiver testing and support for software-defined radio (SDR)	1-3
Link-level simulation and analysis in the presence of RF and channel impairments	1-4
Multinode communication in Bluetooth mesh, piconet, and LE Audio networks	1-4
C and C++ code generation support	1-5

R2022a

Version: 1.0

New Features

Introducing Bluetooth Toolbox

MathWorks Bluetooth Toolbox provides standard-based tools to design, simulate, and verify Bluetooth communications systems. It supports test waveform generation, golden reference verification, and Bluetooth network modeling.

With the toolbox, you can configure, simulate, and analyze end-to-end Bluetooth communication links. You can create and reuse test benches to verify that your designs, prototypes, and implementations comply with the Bluetooth standard, including Bluetooth basic rate/enhanced data rate (BR/EDR) and low energy (LE). You can also assess coexistence, interference, localization, and LE Audio scenarios by modeling multiple layers of the Bluetooth protocol stack.

Bluetooth physical layer processing

- Generate, decode, and visualize Bluetooth waveforms. Use the `bluetoothWaveformGenerator` and `bluetoothIdealReceiver` functions for Bluetooth BR/EDR waveforms and the `bleWaveformGenerator` and `bleIdealReceiver` functions for Bluetooth LE waveforms. For an example showing how to generate and visualize Bluetooth LE waveforms with different physical layer (PHY) modes, see “Bluetooth LE Waveform Generation and Visualization”.
- To compute the Bluetooth BR/EDR or LE packet duration given the type of Bluetooth packet, the specified PHY transmission mode, and the payload length, use the `bluetoothPacketDuration` function.
- The PHY features of the toolbox enable you to add radio frequency (RF) impairments and path loss to the generated Bluetooth waveforms. For examples showing how to add RF impairments and path loss to generated Bluetooth waveforms, see “Generate Bluetooth LE Waveform and Add RF Impairments” and “Generate and Attenuate Bluetooth BR/EDR Waveform in Industrial Environment”, respectively.

For more information, see “PHY Modeling”.

Coexistence modeling between Bluetooth and WLAN

- You can model, simulate, and visualize noncollaborative coexistence between Bluetooth and WLAN and mitigate interference by using adaptive frequency hopping (AFH). See the following examples.
 - “Noncollaborative Bluetooth LE Coexistence with WLAN Signal Interference” — This example shows how to simulate Bluetooth LE coexistence with WLAN signal interference. Add WLAN signal interference from a baseband file or by using WLAN Toolbox™ features. You can also add a custom channel selection algorithm.
 - “PHY Simulation of Bluetooth BR/EDR, LE, and WLAN Coexistence” — This example performs PHY simulation to model homogenous and heterogeneous noncollaborative coexistence between Bluetooth BR/EDR, LE, and WLAN. Perform AFH by classifying the channels as good or bad and compute the bit error rate (BER) and signal-to-interference plus noise ratio (SINR).
 - “End-to-End Bluetooth BR/EDR PHY Simulation with WLAN Interference and Adaptive Frequency Hopping” — This example performs an end-to-end Bluetooth BR/EDR simulation and computes the BER and packet error rate (PER) values of the Bluetooth BR/EDR waveforms in the presence of WLAN interference with AFH and basic frequency hopping.
 - “Bluetooth Full Duplex Data and Voice Transmission in MATLAB” — This example shows how to model and simulate a full duplex communication in a Bluetooth piconet with WLAN interference and with support for AFH. You can compute the PER of each Bluetooth node in the

presence of WLAN interference and with AFH. Additionally, you can add your own channel classification algorithm to analyze the simulation results.

- Select Bluetooth LE channel index for connection, periodic advertising, and isochronous events by using the `bleChannelSelection` System object™. For an example showing how to select Bluetooth LE channel index, see “Bluetooth LE Channel Selection Algorithms”.
- Generate Bluetooth BR/EDR hopping sequence for inquiry, paging, and connection procedures by using the `bluetoothFrequencyHop` object.

For more information, see “Coexistence Modeling”.

Bluetooth location and direction-finding and ranging capabilities

- Estimate the angle of arrival (AoA) or angle of departure (AoD) by using the `bleAngleEstimate` function. To parameterize this function, use the `bleAngleEstimateConfig` configuration object and the associated object functions. For more information about AoA and AoD direction finding capabilities, see “Bluetooth Location and Direction Finding” and “Parameterize Bluetooth LE Direction Finding Features”.
- Estimate the range between two Bluetooth BR/EDR or LE devices by using the `bluetoothRange` function. Use the `bluetoothRangeConfig` object to parameterize this function.
- The `bleCTEIQSample` function enables you to perform in-phase and quadrature (IQ) sampling on the constant tone extension (CTE) field of the Bluetooth LE packet. Using this function, you can estimate the AoA and AoD between the Bluetooth LE transmitter and receiver.
- To estimate the position of a Bluetooth LE node, use the `blePositionEstimate` function. The “Bluetooth LE Positioning by Using Direction Finding” reference example enables you to estimate the 2-D or 3-D position of a Bluetooth LE node by implementing Bluetooth direction finding functionality and the triangulation-based location estimation technique. You can measure the positioning accuracy of the Bluetooth LE node related to the bit energy-to-noise density ratio (Eb/No).
- The “Bluetooth LE Direction Finding for Tracking Node Position” example shows how to track the Bluetooth LE node position by using Bluetooth direction finding functionalities and position estimation techniques. Simulate the direction finding packet exchange in the presence of radio frequency (RF) front end impairments, path loss model, and additive white Gaussian noise (AWGN) and measure the positioning accuracy at each node position.

For more information, see “Localization”.

Bluetooth transmitter and receiver testing and support for software-defined radio (SDR)

- The `bluetoothRFPHYTestConfig` object enables you to set Bluetooth LE RF-PHY transmitter and receiver test configuration parameters based on RF-PHY.TS.p15.
- Generate Bluetooth BR/EDR or LE test waveforms by using the `bluetoothTestWaveform` function. The function supports all the PHY transmission modes of Bluetooth BR/EDR and LE. To parameterize this function, use the `bluetoothTestWaveformConfig` object.
- These reference examples show how to perform RF tests on Bluetooth BR/EDR waveforms.
 - “Bluetooth BR/EDR Power and Spectrum Tests”
 - “Bluetooth BR RF-PHY Transmitter Tests for Modulation Characteristics, Carrier Frequency Offset, and Drift”

- “Bluetooth EDR RF-PHY Transmitter Tests for Modulation Accuracy and Carrier Frequency Stability”
- These reference examples show how to perform RF-PHY tests on Bluetooth LE waveforms.
 - “Bluetooth LE IQ samples Coherency and Dynamic Range Tests”
 - “Bluetooth LE Modulation Characteristics, Carrier Frequency Offset and Drift Tests”
 - “Bluetooth LE Blocking, Intermodulation and Carrier-to-Interference Performance Tests”
 - “Bluetooth LE Output Power and In-Band Emissions Tests”
- You can generate, transmit, and receive Bluetooth BR/EDR and LE waveforms by using the ADALM-PLUTO radio. For more information, explore these examples.
 - “Bluetooth BR/EDR Waveform Generation and Transmission Using SDR”
 - “Bluetooth BR/EDR Waveform Reception Using SDR”
 - “Bluetooth LE Waveform Generation and Transmission Using SDR”
 - “Bluetooth LE Waveform Reception Using SDR”

For more information, see “Test and Measurement”.

Link-level simulation and analysis in the presence of RF and channel impairments

The toolbox provides reference examples that enable you to perform end-to-end Bluetooth BR/EDR and LE simulation.

- You can evaluate and analyze the link-level performance in the presence of RF impairments, path loss, and WLAN interference. For more information, see these examples.
 - “End-to-End Bluetooth BR/EDR PHY Simulation with Path Loss, RF Impairments, and AWGN”
 - “End-to-End Bluetooth LE PHY Simulation Using Path Loss Model, RF Impairments, and AWGN”
 - “End-to-End Bluetooth LE PHY Simulation with AWGN, RF Impairments and Corrections”
- Compute PER and BER for the Bluetooth BR/EDR and LE waveforms. For examples showing how to compute PER and BER, see “Bluetooth LE Bit Error Rate Simulation with AWGN” and “End-to-End Bluetooth BR/EDR PHY Simulation with AWGN, RF Impairments and Corrections”.

For more information, see “End-to-End Simulation”.

Multinode communication in Bluetooth mesh, piconet, and LE Audio networks

Use these features and reference examples to simulate multinode communication in Bluetooth mesh, piconet, and LE Audio networks.

- The `bluetoothLENode` object and the associated object functions enable you to create and configure a Bluetooth LE node. Configure a link layer (LL) connection between a Bluetooth LE Central and Peripheral by using the `bluetoothLEConnectionConfig` object and the associated object function. For more information, see the “Create, Configure, and Simulate Bluetooth LE Network” example.

-
- The `bluetoothMeshProfileConfig` object enables you to configure mesh profile parameters at a Bluetooth LE node based on Bluetooth Mesh Profile v1.0.1. To establish friendship between a Friend node and a Low Power node (LPN) in a Bluetooth mesh network, use the `bluetoothMeshFriendshipConfig` object and the associated object function. For more information, see “Bluetooth Mesh Flooding in Wireless Sensor Networks” and “Energy Profiling of Bluetooth Mesh Nodes in Wireless Sensor Networks” examples.
 - Use the `bluetoothLEBIGConfig` object to set broadcast isochronous group (BIG) configuration parameters between a Bluetooth LE isochronous broadcaster and a synchronized receiver. For more information, see the “Create, Configure, and Simulate Bluetooth LE Broadcast Audio Network” example.

The “Estimate Packet Delivery Ratio of LE Broadcast Audio in Residential Scenario” example shows how to estimate the packet delivery ratio (PDR) of Bluetooth LE audio isochronous broadcast streams in a residential scenario. You can add WLAN interference and a custom path loss model to the wireless channel and explore the PDR performance of the LE audio broadcast network. The example enables you to visualize PDR at different receiver locations in the residential scenario through a heatmap.

- The “Evaluate the Performance of Bluetooth QoS Traffic Scheduling with WLAN Signal Interference” example shows how to evaluate the performance of the Bluetooth scheduler. You can simulate multiple applications with different quality-of-service (QoS) requirements (throughput and latency) in the presence of WLAN signal interference. You can also plug-in your own custom scheduler and analyze the throughput and latency performance.
- The protocol data unit (PDU) generation and decoding functions and objects enable you to generate and decode Bluetooth LE LL, the logical link control and adaptation protocol (L2CAP), the generic access profile (GAP), and the attribute protocol (ATT) layer PDUs. For examples showing how to generate and decode PDUs, see “Generate and Decode Bluetooth Protocol Data Units”, “Bluetooth LE Link Layer Packet Generation and Decoding”, and “Bluetooth LE L2CAP Frame Generation and Decoding”.
- To export Bluetooth LE LL packets to packet capture (PCAP) and packet capture next generation (PCAPNG) files, use the `blePCAPWriter` object.

For more information, see “Multinode Communication”.

C and C++ code generation support

Bluetooth Toolbox supports ANSI®/ISO® compliant C/C++ code generation. For an alphabetized list of features that support C/C++ code generation, see Bluetooth Toolbox - Functions and Objects Filtered by C/C++ Code Generation.

