

# AI-01862 Zigbee Sniffer

Owner	© Rigel Arcayan
Tags	Project
Created time	@July 22, 2025 9:24 AM

Can I use ZBDongle-E Zigbee 3.0 USB as Zigbee sniffer with Wireshark? Is the Zigbee ZBDongle-E the best sniffer hardware I need or there is something else? How to use ZBDongle-E as sniffer in Wireshark?

Yes, you absolutely **can use the ZBDongle-E Zigbee 3.0 USB as a Zigbee sniffer with Wireshark**.<sup>1</sup> It's a popular and effective choice for this purpose, especially given its affordability and the EFR32MG21 chipset it uses.

## Is the ZBDongle-E the best sniffer hardware?

The ZBDongle-E is a **very good and highly recommended option** due to its:

- **Affordability:** It's one of the most cost-effective sniffers available.<sup>2</sup>
- **Chipset (EFR32MG21):** This chip is capable and well-supported for sniffing.
- **Community Support:** Being popular, there's a good amount of online documentation and community help available.

However, calling it "the best" depends on your specific needs and budget. Other notable options for Zigbee sniffing include:

- **Texas Instruments CC2531/CC2652P/CC1352P:** These are widely used and have good support with TI's SmartRF Packet Sniffer software, which can then pipe data to Wireshark. The CC2652P and CC1352P are more powerful than the older CC2531.
- **Nordic nRF52840 DK or Dongle:** Nordic Semiconductor offers the nRF Sniffer for 802.15.4, which integrates well with Wireshark and is a strong contender, especially if you're already working with Nordic's ecosystem.<sup>3</sup>
- **Silicon Labs Ember-based dongles (e.g., CEL MeshConnect):** These are also robust options, often used with Java-based sniffer software that feeds into Wireshark.

For most users, especially those getting started with Zigbee sniffing, the **ZBDongle-E offers an excellent balance of cost and capability**.

## How to use ZBDongle-E as a sniffer in Wireshark:

Using the ZBDongle-E as a sniffer with Wireshark generally involves these steps:

### 1. Flash Sniffer Firmware:

- The ZBDongle-E typically comes pre-flashed with Zigbee coordinator firmware.<sup>4</sup> To use it as a sniffer, you'll need to **flash specific sniffer firmware** onto it.
- You'll likely find this firmware (often a `.gbl` or `.hex` file) on the Sonoff website or in community repositories like the one by ErkSponge on GitHub, which specifically provides sniffer firmware for the ZBDongle-E.<sup>5</sup>
- The flashing process usually involves a specific tool (e.g., a dedicated flashing utility or a tool like `cc-tool` for Linux). Refer to the firmware provider's instructions carefully.

### 2. Install Wireshark and the necessary Extcap/Sniffer Software:

- **Wireshark:** Download and install the latest version of Wireshark from its official website.<sup>6</sup>
- **Extcap/Sniffer Application:** Wireshark doesn't directly interface with the dongle. It needs an "Extcap" (External Capture) plugin or a separate sniffer application to read data from the dongle and pipe it to Wireshark.
  - For the ZBDongle-E (EFR32MG21 chip), you'll often use a tool like the one from ErkSponge's GitHub repository, which includes an `Extcap_802.15.4.exe` file for Windows. This file needs to be copied into Wireshark's extcap folder (you can find the path in Wireshark's "Help → About Wireshark" under the "Folders" tab).
  - Alternatively, some setups might use a Java-based sniffer application (like `ZigbeeSniffer.jar` from Z-Smart Systems) that creates a data pipe for Wireshark to connect to.<sup>7</sup>

### 3. Identify the Dongle's COM Port (Windows) or Device Path (Linux):

- Plug in the ZBDongle-E to your computer.
- **Windows:** Open Device Manager (Win+X, M) and look under "Ports (COM & LPT)" to find the COM port assigned to your dongle (e.g., COM3, COM6).
- **Linux:** The dongle will likely appear as something like `/dev/ttyUSB0` or similar.

### 4. Configure Wireshark:

- **Start the Extcap/Sniffer Application (if applicable):** If you're using a separate sniffer application (like `ZigbeeSniffer.jar`), start it *before* Wireshark. You'll typically specify the COM port/device path and the Zigbee channel you want to sniff.

- **Start Wireshark:**
  - In Wireshark, you should see the new capture interface related to your dongle (e.g., a COM port or a named pipe). Select it.
  - **Channel Selection:** If the Extcap doesn't handle channel selection automatically, look for a gear icon next to the interface in Wireshark to configure the Zigbee channel you wish to sniff.
  - **Start Capture:** Click the "Start" button (green fin) to begin capturing packets.

## 5. Decrypt Zigbee Traffic (Crucial!):

- Most Zigbee networks use encryption. Without the network key, you'll see encrypted data, which is largely unreadable.
- **Add Network Key:**
  - Go to **Edit > Preferences > Protocols > ZigBee**.
  - Under "Pre-configured keys," click "Edit."
  - Add your Zigbee network's "Trusted Network Key." The default "Zigbee Alliance" key is often `5A6967426565416C6C69616E63653039`.
  - You'll also need your specific **Zigbee Network Key**. This key is unique to your hub/coordinator. You can often retrieve it from your Zigbee hub's configuration (e.g., Home Assistant's `configuration.yaml` or `coordinator_backup.json`, or by sniffing the "Transport Key" when a new device joins the network, as it's often sent unencrypted during the joining process). Ensure you enter it with the correct byte order (usually "Normal").
- Once these keys are added, Wireshark should be able to decrypt the Zigbee traffic, allowing you to see the application layer data.

### Important Notes:

- **Firmware is Key:** The correct sniffer firmware is essential for the dongle to operate in sniffing mode.
- **Admin Privileges:** You might need to run the sniffer application and/or Wireshark with administrator privileges.
- **Single Channel:** A Zigbee sniffer typically listens on only one channel at a time. If your Zigbee network uses multiple channels, you'll need to switch the sniffer's channel to capture traffic on different parts of your network.

- **Interference:** USB 3.0 ports can sometimes cause interference with 2.4 GHz Zigbee signals.<sup>8</sup> Using a USB 2.0 extension cable can help.

By following these steps, you should be able to effectively use your ZBDongle-E as a Zigbee sniffer with Wireshark to analyze your Zigbee network traffic.

# Comprehensive Guide to Zigbee 3.0 Packet Sniffing with ZBDongle-E and Wireshark

## I. Executive Summary

The SONOFF ZBDongle-E, equipped with the EFR32MG21 chipset, presents a highly capable and cost-effective solution for Zigbee 3.0 packet sniffing when configured with appropriate sniffer firmware. This repurposing of a readily available USB dongle is a prevalent and effective methodology for advanced network diagnostics. Wireshark, globally recognized as a premier network protocol analyzer, serves as the indispensable front-end for dissecting and analyzing captured Zigbee traffic, offering profound insights into the intricate behaviors of a Zigbee network.

Achieving successful and meaningful Zigbee sniffing necessitates a series of critical technical steps. These include flashing dedicated sniffer firmware onto the dongle, employing an intermediary application or an Extcap plugin to reliably feed raw data to Wireshark, and meticulously configuring crucial decryption keys within Wireshark for clear packet interpretation.

The ZBDongle-E distinguishes itself as a robust and accessible choice for Zigbee sniffing, primarily owing to its EFR32MG21 chip and the extensive support it receives from the open-source community for sniffer firmware development. A critical aspect of effective analysis is decryption: both the universally common Zigbee Trust Center Link Key and the network-specific Transport Key are indispensable for deciphering encrypted Zigbee messages. Without these cryptographic keys, captured data remains largely unintelligible. Furthermore, environmental factors, notably radio frequency interference from pervasive 2.4 GHz Wi-Fi networks and electromagnetic interference emanating from USB 3.0 ports, significantly influence the reliability and integrity of sniffed data. Proactive management of these factors through meticulous planning and strategic hardware placement is therefore paramount.

## II. Understanding Zigbee Sniffing

### What is a Zigbee Sniffer and its Purpose?

A Zigbee sniffer constitutes a specialized combination of hardware and software meticulously engineered to passively capture and display raw data frames transmitted across a Zigbee network. This capability fundamentally differs from the operational modes of a device acting as an active coordinator or a router within the network. The primary function of a sniffer is to observe low-level network communications, a process invaluable for debugging complex network issues that prove intractable through higher-level Application Programming Interfaces (APIs) or standard network logs. This granular observation facilitates in-depth investigations into protocol-specific anomalies and communication failures.

It is important to recognize that Zigbee sniffing is typically employed as a diagnostic tool for targeted analysis of specific events or problems, rather than for continuous, round-the-clock network monitoring. Attempting to capture an excessive volume of data can rapidly become unmanageable, hindering effective analysis.

## **The Role of Wireshark in Analyzing Zigbee Traffic**

Wireshark is a powerful, free, and open-source network protocol analyzer widely utilized for inspecting network traffic. Its extensive capabilities for dissecting various network protocols make it an ideal platform for in-depth Zigbee analysis. Wireshark excels at presenting the contents of captured Zigbee packets in a human-readable format, thereby enabling detailed debugging of low-level protocol issues and providing a clear understanding of communication flows.

Crucially, Wireshark does not directly interface with Zigbee hardware. Instead, it relies on an external dongle and associated sniffer software to capture the raw radio data. This raw data is then converted into a standardized format, such as the Zigbee Encapsulation Protocol (ZEP) transmitted over UDP, which Wireshark can then process and dissect. This multi-layered architecture is fundamental to the operation of a Zigbee sniffer with Wireshark. The process involves the dedicated hardware (the ZBDongle-E), running specialized sniffer firmware, which then communicates with an intermediary software component (acting as a data pipe or an Extcap plugin). This intermediary component, in turn, streams the captured data to Wireshark for comprehensive analysis and visualization. Understanding this modular design is paramount, as successful setup and troubleshooting require a correct configuration and operational integrity at each distinct layer. A malfunction or misconfiguration at any point within this chain can impede the entire sniffing process, preventing the capture or proper display of Zigbee traffic.

## **III. ZBDongle-E as a Zigbee Sniffer**

### **Compatibility and Capabilities of the ZBDongle-E (EFR32MG21 Chip)**

The SONOFF Zigbee 3.0 USB Dongle Plus Model "ZBDongle-E" is specifically engineered around the Silicon Labs EFR32MG21 chipset. This particular chip is a modern and highly capable component for contemporary Zigbee 3.0 applications. The ZBDongle-E is broadly recognized and recommended as a robust Zigbee radio adapter across various open-source smart home platforms, including the ZHA integration within Home Assistant. This widespread endorsement underscores its strong compatibility and reliable performance within the broader Zigbee ecosystem.

While the ZBDongle-E typically ships pre-flashed with EZNet 6.10.3 coordinator firmware, enabling its use as a Zigbee gateway, it possesses the inherent capability to be reprogrammed with specialized sniffer firmware to facilitate passive packet capture. The foundation of the ZBDongle-E on the Silicon Labs EFR32MG21 chip provides a significant advantage for sniffing operations. This chipset benefits from a mature Silicon Labs development ecosystem and, critically, robust and active open-source community support. Projects such as

[darkxst/silabs-firmware-builder](#) actively develop and provide sniffer firmware builds specifically for the ZBDongle-E. This strong community backing ensures the availability of specialized firmware and continuous development, positioning the ZBDongle-E as a more viable and enduring choice for sniffing compared to older, less actively maintained chipsets, such as the TI CC2531, which has seen reduced validation with Wireshark over time.

## **Distinction: Coordinator/Router Firmware vs. Sniffer Firmware**

The ZBDongle-E's factory-installed firmware enables it to function as a Zigbee coordinator, acting as the central hub for a Zigbee network, or it can be re-flashed with router firmware to extend the network's signal range. However, when configured for sniffing, the dongle operates in a specialized, passive mode with distinct firmware. In this sniffer mode, the device cannot simultaneously participate as an active node (either a coordinator or a router) within the Zigbee network while performing packet capture. This represents a critical functional limitation inherent to most sniffer hardware.

This functional limitation implies a practical trade-off for individuals utilizing the dongle. If a user possesses only a single ZBDongle-E, they cannot simultaneously employ it as their primary smart home Zigbee coordinator and actively sniff network traffic. Such a scenario would necessitate either dedicating the dongle exclusively to sniffing (thereby taking the smart home network offline during diagnostic periods) or acquiring a second dongle specifically for sniffing purposes. This consideration is vital for resource planning and maintaining network availability, especially in production smart home environments.

## **IV. Exploring Alternative Zigbee Sniffer Hardware**

While the ZBDongle-E offers a compelling balance of cost and capability for Zigbee sniffing, understanding alternative hardware options is essential for a comprehensive perspective. These alternatives range from other widely adopted consumer-grade dongles to highly specialized, professional-grade protocol analyzers.

## Detailed Comparison of Popular Options

- **Texas Instruments (TI) based radios (e.g., CC2531, CC2652P/CC1352P):**
  - **Legacy CC2531:** Older TI CC2531 dongles were once a common choice for Zigbee sniffing. However, Texas Instruments itself indicates that the CC2531EMK firmware has not been validated with Wireshark for some time. Furthermore, these chips are generally no longer recommended for general Zigbee coordination due to deprecated hardware, outdated firmware, and performance limitations, particularly in larger networks exceeding 15-20 devices. Flashing new firmware onto a CC2531 typically requires an external debugger, which adds both cost and complexity to the setup process. These older devices are also often limited to sniffing a single channel without a true promiscuous mode.
  - **Modern CC2652P/CC1352P:** Newer TI chips, such as the CC2652P and CC1352P, are recommended for robust Zigbee coordination. The CC1352P offers an additional capability for sub-1 GHz radio communication, though this is not relevant for standard 2.4 GHz Zigbee operations. While these chips are powerful and well-suited for coordinator roles, their dedicated sniffer setup procedures may differ from those of the older CC2531.
- **Nordic Semiconductor (e.g., nRF52840 Dongle):**
  - The Nordic nRF52840 Dongle is highly regarded as a valuable tool for both learning and debugging 802.15.4 protocols, including Thread and Zigbee. It integrates seamlessly with Wireshark via a dedicated Extcap plugin, providing a more streamlined and user-friendly experience compared to relying on external data piping applications. This dongle offers real-time packet display enriched with critical metadata, such as timestamps, Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI), and channel information, all of which are vital for in-depth network analysis. All necessary software, including firmware and the Wireshark plugin, is freely available and supports multiple operating systems, including Windows, Linux, and macOS.
- **Other Silicon Labs-based dongles (e.g., SMLIGHT SLZB-06M, SLZB-07, Home Assistant Connect ZBT-1):**

- These devices often leverage the same EFR32MG21 or similar Silicon Labs chipsets as the ZBDongle-E. Consequently, they benefit from the same robust community firmware development and convenient web flasher tools, rendering them equally viable for sniffing applications. Certain models, such as the SMLIGHT SLZB-06M, notably offer Power over Ethernet (PoE) connectivity. While PoE does not directly enhance sniffing capabilities, it provides significant advantages for network stability and flexible placement when the dongle is used as a coordinator, by mitigating USB interference and allowing the device to be positioned optimally away from the host machine.
- **High-end/Professional Sniffers (e.g., Ellisys Bluetooth Tracker, Frontline X500TM Wireless Protocol Analyzer):**
  - These devices represent the pinnacle of protocol analysis tools, with acquisition costs typically ranging from approximately \$25,000 to over \$45,000. Many are based on Software Defined Radio (SDR) technology, which allows their firmware to be updated to support future versions of wireless specifications, providing unparalleled longevity and adaptability. They offer highly advanced features and multi-protocol capture capabilities, enabling simultaneous analysis of technologies like Bluetooth, Wi-Fi, and 802.15.4. However, their prohibitive cost and inherent complexity generally render them unsuitable for the typical technical enthusiast or home user.

## Key Factors for Selecting a Sniffer

When selecting a Zigbee sniffer, several factors should be considered:

- **Cost:** The ZBDongle-E and Nordic nRF52840 dongle offer highly cost-effective entry points into Zigbee sniffing.
- **Performance & Features:** Evaluate the device's ability for real-time capture, the richness of captured metadata (e.g., RSSI, LQI), and whether the sniffer supports a promiscuous mode (capturing all packets on a selected channel).
- **Ease of Use & Setup:** Prioritize devices with clear, comprehensive documentation, readily available sniffer firmware, and simplified flashing processes (e.g., web flashers, direct Wireshark Extcap integration).
- **Community Support:** An active and engaged community provides invaluable resources, including alternative firmware, troubleshooting guides, and supplementary tools.
- **Chipset:** The underlying radio chip (e.g., Silicon Labs EFR32MG21, Nordic nRF52840, Texas Instruments CC2652P) dictates compatibility with specific sniffer firmware and

software ecosystems.

While the ZBDongle-E is an excellent, affordable choice for dedicated Zigbee sniffing, its reliance on a USB interface makes it inherently more susceptible to environmental interference, particularly from USB 3.0 ports or adjacent Wi-Fi networks. More advanced, PoE-enabled dongles, such as certain SMLIGHT models, offer superior network stability and placement flexibility. These advantages can make them a more robust choice for a permanent Zigbee coordinator setup, as they mitigate USB interference and allow for optimal positioning away from the host machine. This distinction highlights that the optimal dongle for passive sniffing may not always be the optimal choice for actively managing a production smart home network, even if the hardware possesses the technical capability to serve both functions.

## Zigbee Sniffer Hardware Comparison

Device/Chipset	Key Features	Pros	Cons	Estimated Price Range
<b>SONOFF ZBDongle-E</b> (EFR32MG21)	Zigbee 3.0, EFR32MG21 chip, USB 2.0	Cost-effective, strong community support, widely compatible with open-source platforms, good performance for sniffing.	Requires firmware flashing (can be complex), susceptible to USB 3.0/Wi-Fi interference.	~\$15-25
<b>Nordic nRF52840 Dongle</b>	Zigbee, Thread, BLE, 802.15.4, USB	Low cost, excellent Wireshark Extcap integration, real-time display with rich metadata (RSSI, LQI), cross-platform support, all software free.	Requires firmware flashing, can drop packets occasionally (hardware limitation for single channel).	~\$10-30
<b>TI CC2531 USB Dongle</b> (Legacy)	Zigbee, 802.15.4, USB	Very low cost, basic sniffing capability.	Deprecated hardware/firmware, not officially validated with	~\$5-15 (often used, less available new)

			Wireshark, requires external debugger for flashing, limited to single channel, struggles with larger networks.	
<b>TI CC2652P/CC1352P USB Stick (Modern)</b>	Zigbee 3.0, CC2652P/CC1352P chip, USB, Power Amplifier (P models)	Robust performance as coordinator, good for sniffing (with proper firmware), strong signal.	May require specific sniffer firmware not as readily available as ZBDongle-E, higher cost than basic dongles.	~\$50-100
<b>SMLIGHT SLZB-06M/SLZB-07 (EFR32MG21)</b>	Zigbee 3.0, EFR32MG21 chip, USB, <b>PoE option (06M)</b>	Strong community support, robust, PoE option for stable coordinator placement (mitigates USB interference).	Higher cost than basic USB dongles, PoE benefits more for coordinator than sniffer.	~\$40-70 (USB), ~\$80-120 (PoE)
<b>High-end Professional Analyzers (e.g., Ellisys, Frontline)</b>	Multi-protocol (Zigbee, BLE, Wi-Fi), SDR-based, advanced features, autonomous captures	Unparalleled analysis depth, future-proof, simultaneous multi-protocol capture, highly reliable.	Extremely high cost, complex software, often Windows-only.	\$25,000 - \$45,000+

Export to Sheets

## V. Step-by-Step Guide: Setting up ZBDongle-E for Wireshark Sniffing

Setting up the ZBDongle-E for Zigbee packet sniffing with Wireshark involves several distinct stages, each requiring careful execution.

### 1. Initial Setup and Driver Installation

The first crucial step involves preparing the host computer and ensuring proper recognition of the ZBDongle-E.

- **Operating System Considerations:**

- For Windows and macOS environments, manual installation of the appropriate USB-to-serial drivers is typically a prerequisite.
- Linux and Raspberry Pi OS environments often offer a more "Plug & Play" experience, with necessary drivers either pre-installed or automatically detected upon connecting the dongle.
- If the host system runs Home Assistant or other applications within a Virtual Machine (VM) environment, it is imperative to configure the VM to correctly pass through and expose the USB port to which the dongle is connected. Failure to do so will prevent the VM from recognizing the device.

- **Installing Necessary USB-to-Serial Drivers:**

- It is essential to identify the specific serial chip integrated into your ZBDongle-E. Newer manufacturing batches (those with Serial Numbers 30623XXXXX and above, located on the back of the device) typically utilize the CP210x serial chip, whereas older revisions may incorporate the CH9102 chip. This distinction is critical for selecting and installing the correct driver.
- Download and install the appropriate driver: for CP210x chips, drivers can be obtained from Silicon Labs (e.g., `CP210x_VCP_Windows` for Windows, `Mac_OSX_VCP_Driver` for macOS). For CH9102 chips, if not automatically recognized by the operating system, the `CH343SER` driver should be manually downloaded and installed. The presence of different serial chips within the same product line is a common hardware revision challenge that users must be aware of to prevent initial driver installation failures, which can be a significant early hurdle. Correctly identifying the chip and installing the corresponding driver is a fundamental prerequisite for the dongle's functionality.
- After driver installation, confirm device recognition. On Windows, check Device Manager under "Ports (COM & LPT)" for the corresponding USB device. On Linux/macOS, commands like `ls -l /dev/serial/by-id` or `ls /dev/tty.*` can reveal the serial port path.

## 2. Flashing Sniffer Firmware

The ZBDongle-E, by default, functions as a Zigbee coordinator. To enable its sniffing capabilities, it must be flashed with specialized sniffer firmware.

- **Necessity of Sniffer Firmware:** A Zigbee sniffer dongle operates in a distinct mode, capturing low-level frames from the air and passing them to Wireshark. This requires specific firmware that is different from the standard coordinator or router firmware.
- **Obtaining Sniffer Firmware:**
  - For the ZBDongle-E (EFR32MG21), suitable sniffer firmware can be found in community-maintained repositories. A prominent source is the `darkxst/silabs-firmware-builder` GitHub repository, which provides unofficial Zigbee Coordinator and Thread (OpenThread) firmware builds, including sniffer firmware.
  - Alternatively, a precompiled `.jar` file for the ZSmart Systems sniffer, which works with EmberZNet adapters like the ZBDongle-E, can be fetched from ZSmart Systems sniffer releases on GitHub. This sniffer application will place the dongle into the required low-level frame capture mode.
- **Flashing the Firmware:**
  - **Warning:** Re-flashing the device will overwrite its factory coordinator firmware, meaning the USB dongle will no longer function as a Zigbee coordinator until it is re-flashed with coordinator firmware.
  - **Entering Bootloader Mode:** To flash new firmware, the ZBDongle-E must be put into bootloader mode. This typically involves unplugging the dongle, carefully opening its case (if necessary), locating and holding the BOOT button (usually near the USB connector), and then plugging the dongle back into the computer while continuing to hold the BOOT button for approximately 5 seconds before releasing it.
  - **Using a Flashing Tool:**
    - A recommended tool for flashing Silicon Labs dongles is `universal-silabs-flasher`, a Python-based utility. This tool can be installed within a Python virtual environment to maintain system cleanliness (

```
pip install universal-silabs-flasher
```

).
    - Once installed, the firmware (`.gbl` file) can be flashed using a command similar to: 

```
universal-silabs-flasher --device /dev/tty.usbserial-210 flash --firmware ~/Downloads/your_sniffer_firmware.gbl --allow-cross-flashing
```

 (adjusting paths and serial port accordingly).

- **Web-based Flasher:** An easier alternative for ZBDongle-E is the Silabs Firmware Flasher, a web-based tool accessible via Chrome-based browsers. This tool simplifies the flashing process by directly connecting to the dongle via the browser's WebSerial API. Before using, ensure no other applications are using the serial port and, for Linux users, that the user account is added to the `dialout` group.

### 3. Setting up the Sniffer Application/Extcap

Wireshark requires an intermediary application or an Extcap plugin to receive the raw Zigbee data from the dongle.

- **Role of the Intermediary Software:** The sniffer application (e.g., ZSmart Systems sniffer) or Extcap acts as a converter, reading messages from the dongle's serial port and sending them through a data pipe (often TCP/UDP port 17754 for Zigbee Encapsulation Protocol, ZEP) that Wireshark can attach to.
- **Installing and Running the Sniffer Application (e.g., ZSmart Systems Sniffer):**
  - Download the precompiled `.jar` file for the ZSmart Systems sniffer.
  - Open a terminal or command prompt and run the sniffer application using Java:
 

```
java -jar ZigbeeSniffer.jar -baud 115200 -flow {OPTION} -port {PORT} -c {CHANNEL} .
```

    - `{OPTION}`: Replace with `none` for Sonoff Dongle-E and SLZB-06m, or `hardware` for HUSBZB-1/SkyConnect, depending on the adapter's flow control.
    - `{PORT}`: On Windows, find the COM port in Device Manager (e.g., `COM3`). On Linux, it will be a path like `/dev/ttyUSB0` or `/dev/tty.SLAB_USBtoUART`.
    - `{CHANNEL}`: Set this to the Zigbee channel of the network you intend to sniff.
  - It is critical to start this sniffer application and ensure it is actively streaming data *before* launching Wireshark. If Wireshark is stopped, the sniffer application may need to be restarted as well.
- **Installing Wireshark Extcap (for ZBDongle-E specific builds):**
  - Some community-developed sniffer firmware for the ZBDongle-E may come with a dedicated Wireshark Extcap plugin (e.g., `Extcap_802.15.4.exe` for Windows).
  - To install, copy this executable file into Wireshark's Extcap folder. This folder can be located by opening Wireshark, navigating to `Help > About Wireshark`, selecting the

"Folders" tab, and finding the "Global Extcap path" or "Personal Extcap path". After copying, close and restart Wireshark for the Extcap to be loaded.

## 4. Configuring Wireshark for Zigbee Capture

With the dongle flashed and the intermediary sniffer application running, Wireshark needs to be configured to capture and decrypt Zigbee traffic.

- **Wireshark Installation:** Download and install Wireshark from its official website. For Windows, also install Npcap, ensuring the "Npcap Loopback Adapter" option is selected during installation.
- **Starting Capture in Wireshark:**
  - Launch Wireshark (often as administrator for proper interface access).
  - From the Wireshark start screen, select the appropriate interface. If the sniffer application is running on the same computer, the "Loopback" interface is typically used. Otherwise, select the network interface through which the ZEP data is being piped.
  - Apply a display filter in Wireshark's filter bar to focus only on Zigbee traffic. The default ZEP port is 17754, so the filter `udp.port==17754` is effective. An additional filter like `udp.port==17754 &&!icmp` can remove ICMP "Destination unreachable" messages if they clutter the capture.
- **Decryption Keys:** Zigbee messages are encrypted, and without the correct keys, the captured data will be unintelligible. Two primary keys are required for decryption:
  - **Trusted Network Key (Trust Center Link Key):** This key is common to most Zigbee networks. The standard key is `5A:69:67:42:65:65:41:6C:6C:69:61:6E:63:65:30:39`.
  - **Network Encryption Key (Transport Key):** This key is unique to each Zigbee network installation. It can be obtained in several ways:
    - **During Device Join:** When a new Zigbee device joins the network, the hub transmits the Transport Key unencrypted. By capturing traffic during a device's joining process, the Transport Key can be read from the "Transport Key" field within the "ZigBee Network Layer Data" → "ZigBee Security Header" of the "Device Announcement" message.
    - **From Configuration Files:** If using a Zigbee coordinator software like Zigbee2MQTT, the network key might be found in `coordinator_backup.json` (under

`network_key > key` ) or `configuration.yaml` (under `network_key` ). These values may need to be converted to a colon-separated hexadecimal format for Wireshark.

- **Adding Keys in Wireshark:**

1. Navigate to `Edit > Preferences > Protocols > ZigBee` .
2. Set the `Security Level` to `AES-128 Encryption, 32-bit Integrity Protection` .
3. Click on `Pre-configured keys > Edit` .
4. Add the Trusted Network Key ( `5A:69:67:42:65:65:41:6C:6C:69:61:6E:63:65:30:39` ) with `Byte Order Normal` .
5. Add the Network Transport Key (obtained previously) as a second entry, also with `Byte Order Normal` .
6. Press `OK` to apply changes.

- **Protocol Dissection and Expert Information:**

- To ensure comprehensive display of Zigbee protocols, navigate to `Analyze > Enabled Protocols` in Wireshark and enable all protocols containing "Zigbee" in their description.
- Utilize Wireshark's powerful display filters for targeted analysis. Examples include `zbee.zdp` for Zigbee Device Profile messages or `zbee_zcl_general.gp` for Zigbee Cluster Library Green Power messages. The Wireshark Display Filter Reference provides an extensive list of available fields.
- Wireshark's "Expert Information" dialog (accessible via `Analyze > Expert Information` ) highlights anomalies and points of interest within the capture. These entries are categorized by severity (Chat, Note, Warn, Error) and group (e.g., Decryption, Malformed, Protocol, Sequence), providing a quick overview of potential network issues. While a useful starting point, interpreting expert information requires contextual understanding of the network being analyzed.

## VI. Optimizing Zigbee Sniffing and Troubleshooting Common Issues

Effective Zigbee sniffing can be impacted by various environmental and configuration factors. Addressing these proactively is crucial for reliable packet capture and analysis.

### Environmental Interference

- **USB 3.0 Interference:** USB 3.0 ports are known to generate significant noise and radio interference in the 2.4 GHz band, which directly overlaps with Zigbee frequencies. This interference can lead to issues such as pairing difficulties, device dropouts, and timeouts. It is highly recommended to use a USB extension cable between the computer's USB port and the Zigbee dongle to minimize this interference.
- **Wi-Fi Coexistence:** Zigbee and Wi-Fi both operate in the 2.4 GHz spectrum. When both protocols transmit on overlapping channels, interference can degrade the quality of the Zigbee signal.
  - **Channel Planning:** Wi-Fi's three non-overlapping channels (1, 6, and 11) overlap with various Zigbee channels (11-22). Even Zigbee channels 25-26 can be affected by Wi-Fi channel 11's sideband lobes. To mitigate this, careful channel planning is essential. For instance, if Wi-Fi is on channel 11, Zigbee might perform better on channels 15 or 20, or vice versa. It is advisable to set static Wi-Fi channels to avoid unpredictable interference from dynamic channel switching.
  - **Physical Placement:** Positioning the Zigbee dongle as far as possible from Wi-Fi access points and other sources of electromagnetic interference (EMI), such as dryer motors, can significantly improve signal integrity. Optimal placement for a Zigbee adapter is typically in the middle of the house, away from walls, ceilings, and floors.

## Software and Driver Issues

- **Virtual Machine USB Passthrough:** When running Home Assistant or other host systems on a virtual machine, the USB ports are often not exposed to the VM by default. This prevents the Zigbee dongle from being recognized. Users must explicitly configure the VM software (e.g., VirtualBox, VMware) to pass through the specific USB port connected to the dongle. A VM restart may be necessary after configuration.
- **Linux `brltty` Rule:** On Linux systems, some EmberZNet adapters (including the ZBDongle-E, which uses the 1a86:7523 USB identifier) may be misidentified by the `brltty` udev rule, preventing proper recognition. Disabling this rule by commenting out the relevant line in `/usr/lib/udev/rules.d/85-brltty.rules` (e.g., `# ENV{PRODUCT}=="1a86/7523/*", ENV{BRLTTY_BRAILLE_DRIVER}="bm", GOTO="brltty_usb_run"`) and then re-plugging the dongle can resolve the issue.
- **Administrator Privileges:** Both the intermediary sniffer application (e.g., TiWcPc, ZSmart Systems sniffer) and Wireshark should often be run with administrator

privileges to ensure proper access to network interfaces and data pipes.

- **Incorrect Serial Chip Driver:** As discussed, the ZBDongle-E can use different serial chips (CP210x or CH9102) depending on its manufacturing batch. Installing the incorrect driver for the specific chip can lead to recognition failures. Verifying the dongle's serial number to determine the chip type and installing the corresponding driver is crucial.

## Network-Related Troubleshooting

- **Zigbee Router Devices:** A robust Zigbee network relies on a strong mesh, which is built by mains-powered router devices (e.g., smart plugs, light bulbs) that extend coverage and increase device capacity. If a network primarily consists of battery-powered end devices without sufficient routers, devices may lose connection. Flashing additional Zigbee dongles with router firmware and strategically placing them can significantly improve network stability.
- **Device Pairing Proximity:** When pairing new battery-powered Zigbee devices, it is best practice to initiate the pairing process close to the device's final installation location, or at least near a strong routing device. Some devices tend to "stick" to the router they initially paired with, and if that initial connection is too far from their operational spot, they may struggle to maintain connectivity even if closer routers exist.
- **Chatty Devices:** Certain Zigbee devices, particularly some Tuya-branded products, can be excessively "chatty," spamming the network with frequent updates (e.g., every 0.1 seconds for some presence sensors). This high volume of traffic can overwhelm the Zigbee network, especially older coordinator sticks, leading to network crashes and device disconnects. Identifying and managing such devices (e.g., by adjusting their reporting intervals if possible) is important for network health.
- **Sniffer Channel Matching:** For successful packet capture, the sniffer dongle must be configured to listen on the exact Zigbee channel that the target network is operating on. If the network's channel changes (e.g., due to automatic channel selection by the coordinator), the sniffer's channel setting must be updated accordingly. Sniffers typically operate on a single channel at a time and do not natively "hop" across channels to follow a dynamic network.

## VII. Conclusions and Recommendations

The ZBDongle-E is a highly effective and accessible tool for Zigbee 3.0 packet sniffing when properly configured. Its foundation on the EFR32MG21 chip, coupled with robust

community support for sniffer firmware, positions it as a superior choice compared to older, less maintained alternatives like the TI CC2531. The process, while multi-faceted, is well-documented within the open-source community, making it achievable for technical enthusiasts.

Successful Zigbee sniffing hinges on three critical pillars:

1. **Hardware Preparation:** Repurposing the ZBDongle-E with dedicated sniffer firmware and ensuring correct USB-to-serial driver installation, accounting for potential variations in serial chips.
2. **Data Pipelining:** Establishing a reliable connection between the dongle and Wireshark via an intermediary sniffer application or an Extcap plugin, which translates raw radio data into a Wireshark-compatible format.
3. **Decryption and Analysis:** Crucially, configuring Wireshark with both the common Trust Center Link Key and the network-specific Transport Key is non-negotiable for intelligible packet dissection. Without these keys, the captured data remains encrypted and largely useless for debugging.

While the ZBDongle-E excels as a sniffer, it is important to understand its functional distinction: it cannot simultaneously act as a sniffer and an active network participant (coordinator or router). For users with a single dongle, this necessitates a choice between network operation and diagnostic sniffing. Furthermore, environmental factors, particularly 2.4 GHz Wi-Fi interference and USB 3.0 electromagnetic interference, demand careful attention to channel planning and physical dongle placement to ensure clean and reliable packet capture.

### **Recommendations:**

1. **Dedicated Sniffer Dongle:** For continuous smart home operation and simultaneous diagnostic capabilities, consider acquiring a second ZBDongle-E (or a similar EFR32MG21-based dongle) to dedicate solely to sniffing, thereby avoiding disruption to the primary Zigbee network.
2. **Optimize Physical Placement:** Always use a USB extension cable with the ZBDongle-E to mitigate USB 3.0 interference. Position the sniffer dongle (and your primary Zigbee coordinator) strategically, away from Wi-Fi access points, microwaves, and other sources of 2.4 GHz electromagnetic interference.
3. **Strategic Channel Planning:** Proactively plan your Wi-Fi and Zigbee channels to minimize overlap. Utilize Wi-Fi channels 1, 6, or 11, and select a non-overlapping Zigbee channel (e.g., Zigbee 25 or 26 if Wi-Fi is on 1 or 6, or Zigbee 15 if Wi-Fi is on 11).

4. **Prioritize Decryption:** Understand that Zigbee packet analysis is severely limited without decryption keys. Prioritize obtaining and correctly configuring both the Trusted Network Key and your unique Network Transport Key in Wireshark.
5. **Leverage Community Resources:** The open-source community provides invaluable resources, including alternative firmware, web flashers, and troubleshooting forums. Actively engage with these resources for the latest tools and solutions.
6. **Targeted Sniffing:** Employ the sniffer for specific diagnostic tasks rather than continuous monitoring. This approach optimizes resource usage and simplifies data analysis.

Sources used in the report

[dongle.sonoff.techStep 1: Connecting to the Operating System - SONOFF Dongle Opens in a new window](#) [reddit.comDifference between these ZigBee sticks? : r/homeassistant - Reddit Opens in a new window](#) [novelbits.ioBLE Sniffer Basics + Comparison Guide \(Updated 2025\) - Novel Bits Opens in a new window](#) [zigbee2mqtt.ioSupported Adapters - Zigbee2MQTT Opens in a new window](#) [e2e.ti.comCC2531EMK: No packets captured from Wireshark using ZBOSS Sniffer - Zigbee & Thread forum - TI E2E Opens in a new window](#) [wireshark.orgGo Deep | Display Filter Reference: Index - Wireshark Opens in a new window](#) [community.home-assistant.ioIntegration to debug Zigbee firmware? - Home Assistant Community Opens in a new window](#) [dongle.sonoff.techGetting Started - SONOFF Dongle Opens in a new window](#) [itead.ccZigbee 3.0 USB Dongle Plus | ZBDongle-E | ITEAD STUDIO OFFICIAL Opens in a new window](#) [community.smarthings.comZigbee Sniffer Recommendations - Connected Things - SmartThings Community Opens in a new window](#) [sonoff.techSONOFF Zigbee 3.0 USB Dongle Plus | ZBDongle-E Opens in a new window](#) [reddit.comFinally figured out the source of my Zigbee network issues.. \(1-year of troubleshooting\) Opens in a new window](#) [smarthomejunkie.netHow to Troubleshoot Zigbee Issues in Home Assistant Opens in a new window](#) [wireshark.org11.4. Control Protocol Dissection - Wireshark Opens in a new window](#) [opensmarthouse.orgZigbee network sniffer - OpenSmartHouse Opens in a new window](#) [sonoff.techHow to Flash Firmware on SONOFF ZBDongle-E: Step-by-Step Tutorial Opens in a new window](#) [reddit.comHow to Flash Sonoff ZBDongle-E \(Zigbee Router Firmware\) on macOS - Reddit Opens in a new window](#) [wireshark.orgWireshark • Go Deep Opens in a new window](#) [community.home-assistant.ioZigbee network optimization: a how-to guide for avoiding radio frequency interference + adding Zigbee Router devices \(repeaters/extenders\) to get a stable Zigbee network mesh with best possible range and coverage by fully utilizing Zigbee mesh networking - Home Assistant Community Opens in a new window](#) [metageek.comZigBee and Wi-Fi Coexistence - MetaGeek Opens in a new window](#) [e2e.ti.comCC1352P: Questions about CC1352 vs.](#)

[CC2652 - Sub-1 GHz forum - TI E2E Opens in a new window nordicsemi.com](#)[nRF Sniffer for 802.15.4 - nordicsemi.com Opens in a new window nordicsemi.com](#)[nRF Sniffer for Bluetooth LE Development tool - Nordic Semiconductor Opens in a new window wireshark.org](#)[7.4. Expert Information - Wireshark Opens in a new window wireshark.org](#)[Display Filter Reference: ZigBee ZCL Green Power - Wireshark Opens in a new window wireshark.org](#)[Display Filter Reference: ZigBee Device Profile - Wireshark Opens in a new window darkxst.github.io](#)[Silabs Firmware Flasher | Web based flasher for ZB-GW04 and ... Opens in a new window github.com](#)[darkxst/silabs-firmware-builder: Silicon Labs firmware builder - GitHub Opens in a new window zigbee2mqtt.io](#)[Sniff Zigbee traffic | Zigbee2MQTT Opens in a new window diysolarforum.com](#)[Zigbee Sniffer | DIY Solar Power Forum Opens in a new window github.com](#)[NordicSemiconductor/nRF-Sniffer-for-802.15.4 - GitHub Opens in a new window docs.nordicsemi.com](#)[nRF Sniffer for 802.15.4 - Technical Documentation Opens in a new window community.element14.com](#)[Review of Texas Instruments CC2531 USB Evaluation Module Kit - element14 Community Opens in a new window community.element14.com](#)[Review of Texas Instruments CC2531 USB Evaluation Module Kit - element14 Community Opens in a new window github.com](#)[ErkSponge/Sniffer\\_802.15.4\\_SONOFF\\_USB\\_Dongle\\_Plus\\_E: Sniffer firmware for SONOFF Zigbee 3.0 USB DONGLE Plus-E - GitHub Opens in a new window community.hubitat.com](#)[How I setup Wireshark to capture Zigbee messages - Code Share - Hubitat Opens in a new window opensmarthouse.org](#)[Wireshark ZigBee Sniffer - OpenSmartHouse Opens in a new window home-assistant.io](#)[Zigbee Home Automation \(ZHA\) - Home Assistant](#)













