The Many Communication Modes of Bluetooth® LE

PADVB – Periodic Advertising

Understanding Periodic Advertising

In previous articles, two communication modes, each of which are variants of the Advertising Broadcast (ADVB) Link Layer logical transport were examined. The first of the two, legacy advertising ($ADVB_L$) is the original and only form of advertising that the first version of Bluetooth Low Energy (LE) supported. The second is called extended advertising.

But the term "extended advertising" applies to a collection of different advertising capabilities that are associated with several of the Bluetooth LE logical transports, not just ADVB.

Note: *When used with the ADVB logical transport, extended advertising is referred to in these articles as $ADVB_E$.*

What $ADVB_L$ and $ADVB_E$ have in common is that the scheduling of packet transmission on the primary advertising channels is *irregular*, perturbed in time by a random delay value.

Extended advertising also features in two other logical transports, periodic advertising broadcast (PADVB) and periodic advertising with responses (PAwR). Both of these logical transports feature *regular*, transmission schedules and the ability for scanning devices to discover and synchronize their scanning with that schedule.

Periodic advertising (PADVB) is the focus of this article.

Note: *Before reading this article, it is recommended that the previous articles in this series on both legacy advertising ($ADVB_L$) and extended advertising ($ADVB_E$) are read first.*

Overview

Extended advertising made its debut in V5.0 of the Bluetooth Core Specification. It was introduced because of a number of issues relating to what we now know as legacy advertising. For example:

1. Legacy advertising only uses the 3 primary advertising channels and with the rapidly growing number of Bluetooth devices in the world, there was a risk that they would become congested and that this would affect user experience.
2. The amount of application data that can be transported using legacy advertising is constrained by very small PDUs and fields.
3. The uncoordinated, independent transmit and receive behaviors of advertising devices and scanners have pros and cons but can lead to slower application data transfer rates, lower reliability in getting application data to its destination, and poor energy efficiency.

The first two of these issues are addressed by extended advertising in all of its forms because they all share the following characteristics:

- They make use of all 40 of the Bluetooth LE radio channels.
- The AdvData field in which application data is transported, allows up to 254 bytes rather than the maximum of 31 bytes that applies in legacy advertising, larger application data payloads can be fragmented and transmitted in a series of related PDUs in a process known as PDU chaining, and all application data transmission takes place in the 37 general purpose channels rather than the 3 primary advertising channels.

The third issue is dealt with in both periodic advertising broadcast (PADVB) and periodic advertising with responses *(PAwR)*.

PADVB involves the regular transmission of PDUs to an unvarying, regular schedule. That schedule can be established by scanning devices either from AUX_ADV_IND PDUs that are associated with the periodic advertising train or over a connection using a Link Layer procedure called Periodic Advertising Sync Transfer (PAST). Having acquired the periodic advertising schedule, Observer devices can then synchronize their scanning schedule with it to more efficiently and reliably receive advertising packets.

Advertising Modes

In PADVB, as with other forms of extended advertising, the advertising mode is indicated directly using a field called *AdvMode*. PADVB allows the advertising mode to be *non-connectable and non-scannable undirected* only and this is represented by a value of zero in the AdvMode field.

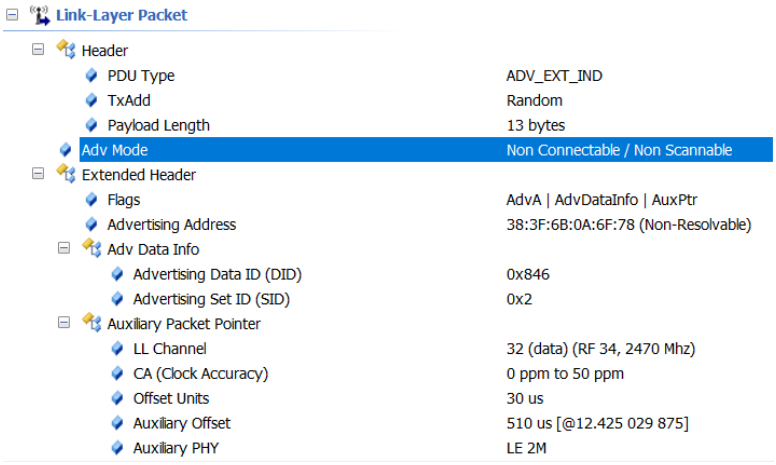


Figure 1 ADV Mode Highlighted in ADV_EXT_IND Packet (Ellisys Analyzer Application)

Device Roles

Periodic advertising must be non-connectable and non-scannable and therefore the advertising device assumes the GAP *Broadcaster* role and scanning devices, the GAP *Observer* role.

States

The Link Layer defines a formal state machine which is depicted in

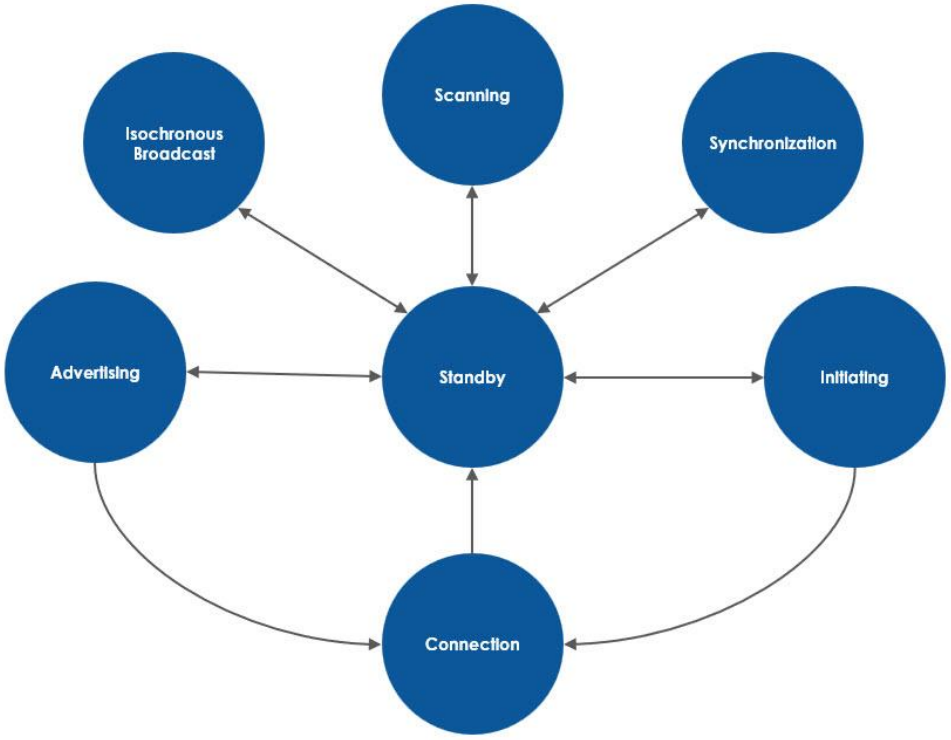


Figure 2 below.

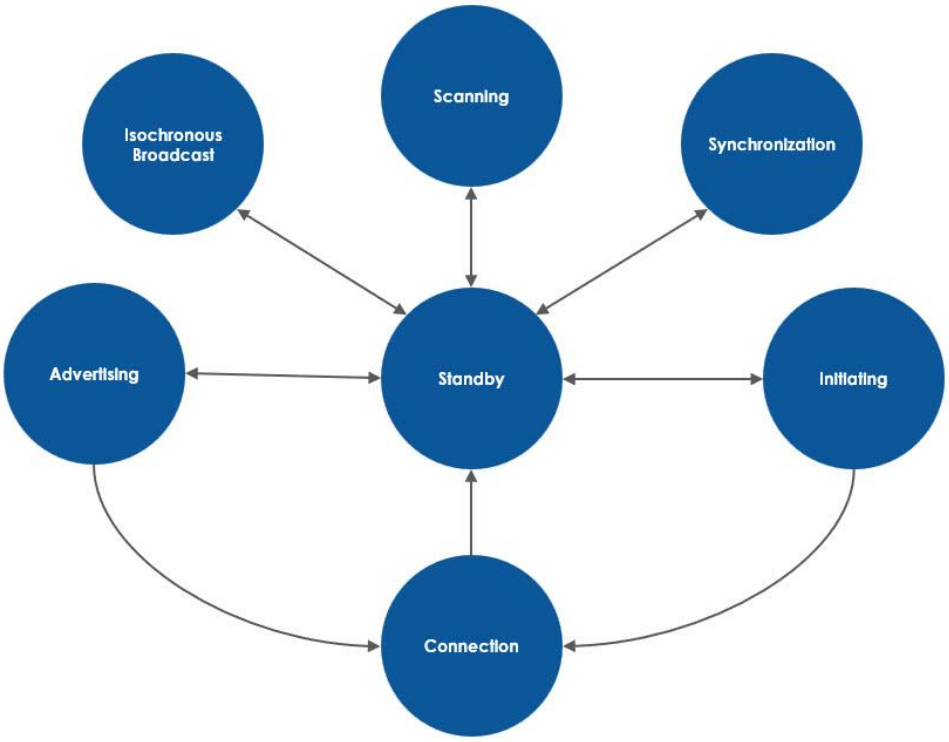


Figure 2 The Link Layer state machine

The Synchronization state is relevant to PADVB and has two sub-states, Synchronizing and Synchronized. These are depicted in

Figure 3, below.

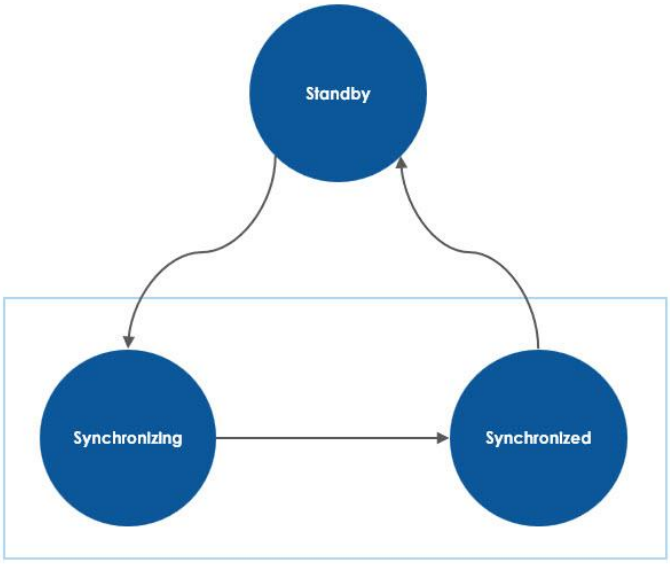


Figure 3 The Synchronization substates

Having acquired information about the schedule of the advertising device either from broadcast AUX_ADV_IND packets or over a connection, the Host can instruct the Controller to enter the Synchronizing state. Once a PDU from the periodic advertising broadcast has been received, the state changes to Synchronized.

Packets, PDUs and Radio Channels

PDUs and Relationships

Periodic advertising involves the regular transmission of AUX_SYNC_IND PDUs.

AUX_ADV_IND PDUs, which are auxiliary PDUs of ADV_EXT_IND PDUs, can include a field called SyncInfo. When present, SyncInfo contains data that describes the schedule of an associated periodic advertising train.

PADVB allows PDU chaining. This involves splitting a large payload into fragments which are then transmitted in a series of linked PDUs which include the AUX_CHAIN_IND type. To this end, an AUX_SYNC_IND PDU can point to an auxiliary AUX_CHAIN_IND PDU.

Thus, the relationship between PDU types is as shown in Figure 4.

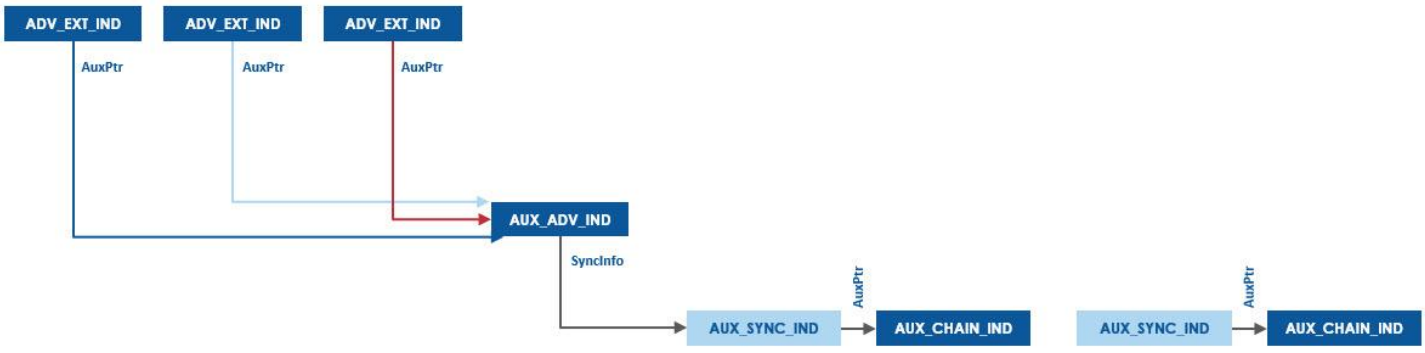


Figure 4 Relationship between PDU types in periodic advertising

Note that no AuxPtr field links adjacent AUX_SYNC_IND PDUs. Scanning devices only need to find the data in the SyncInfo once to then be able to scan at the right time and on the right channel to receive each periodic advertising

PDU. AuxPtr is used to locate auxiliary packets only such as the AUX_ADV_IND PDU from a superior ADV_EXT_IND PDU or an AUX_CHAIN_IND PDU from a superior AUX_SYNC_IND PDU.

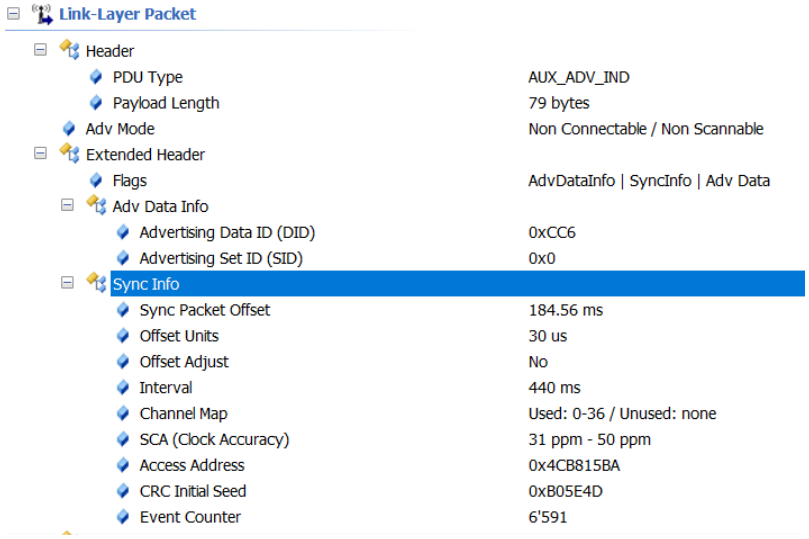


Figure 5 AUX_ADV_IND Packet (Partial) with Sync Info Field Details

Radio Channels

All periodic advertising transmission of AUX_SYNC_IND PDUs takes place over the 37 Bluetooth LE general purpose channels.

That said, as shown in Figure 6, $ADVB_E$ extended advertising is used as a transport for periodic advertising synchronization information, and the same data can also be sent over an LE-ACL connection which requires device discovery to have taken place before the connection can be formed. Therefore, if we include either of the two synchronization approaches in our consideration of the radio channels involved in periodic advertising overall, the answer is that all 40 channels are used. If we focus solely on the transmission of AUX_SYNC_IND PDUs, which strictly speaking constitutes the periodic advertising train, then only the general-purpose channels are used.

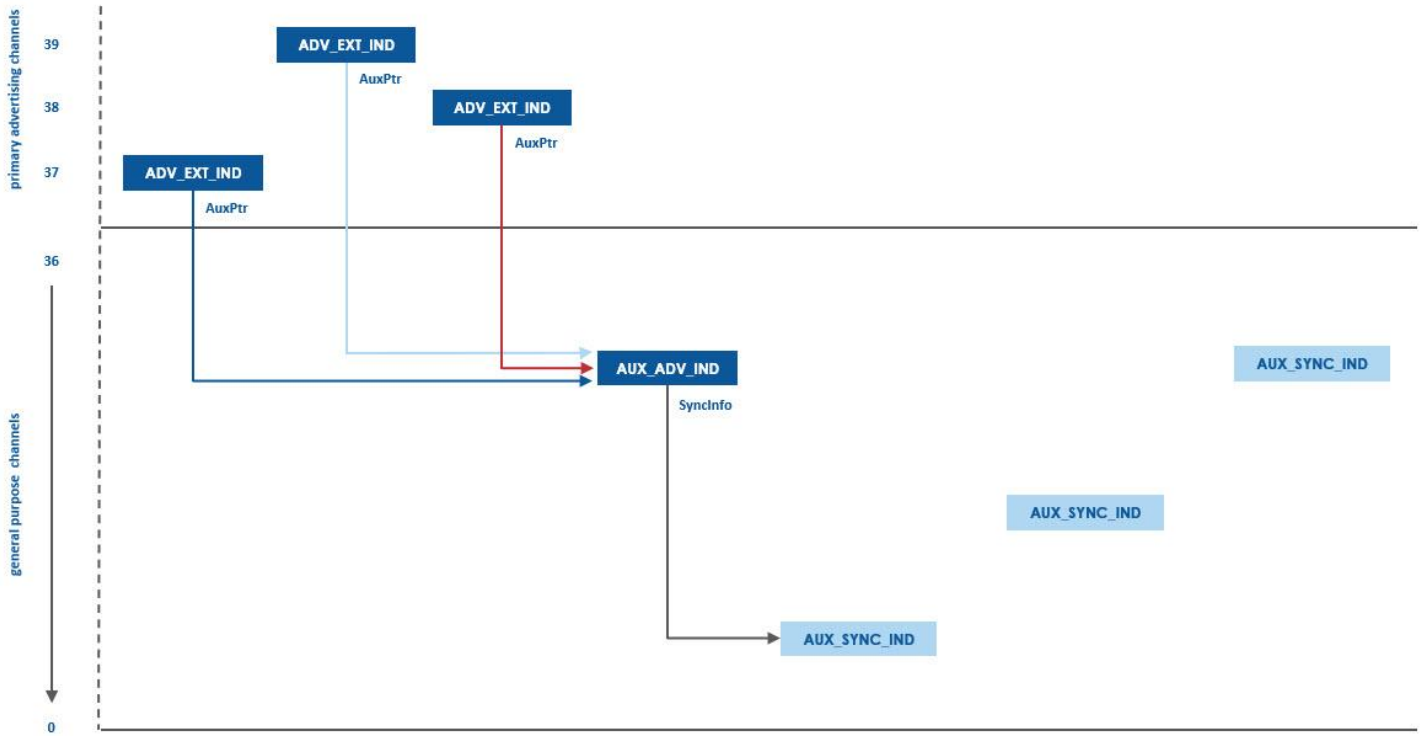


Figure 6 PADVB (and ADVB$_E$) radio channel use

PADVB uses Channel Selection Algorithm #2 (CSA#2) for the selection of channels over which to transmit AUX_SYNC_IND PDUs.

Scheduling

PADVB defines the periodic advertising interval which is the period of time that elapses between the transmission of AUX_SYNC_IND PDUs that form the periodic advertising train. A periodic advertising event starts every periodic advertising interval and ends when an AUX_SYNC_IND PDU and each of its auxiliary AUX_CHAIN_IND PDUs have been transmitted. In contrast to ADVBE extended advertising, no random delay value affects the scheduling of events.

Periodic advertising events and the periodic advertising interval are independent of the advertising event and advertising interval that governs any associated ADVBE extended advertising.

Figure 7 below illustrates the main scheduling parameters used in periodic advertising.

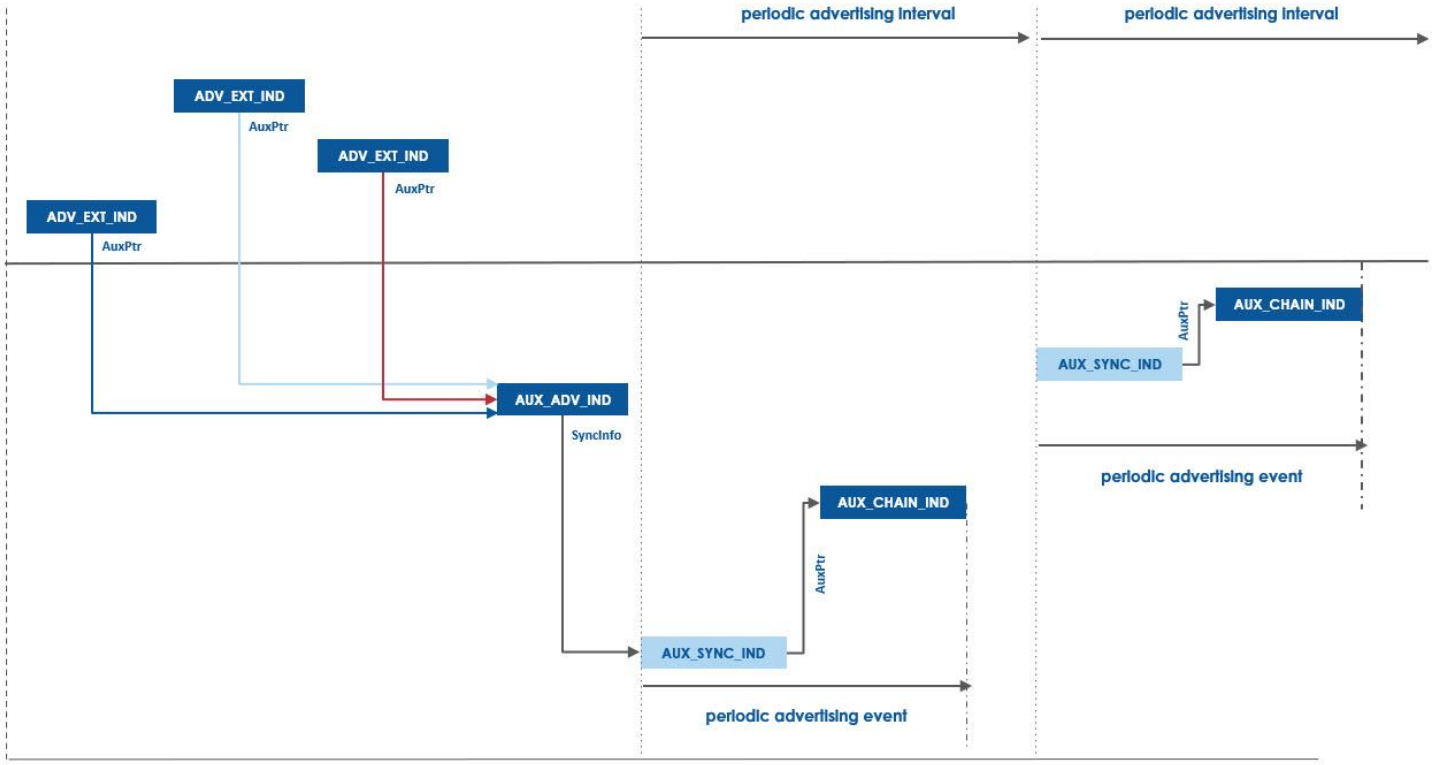


Figure 7 Scheduling of periodic advertising activity

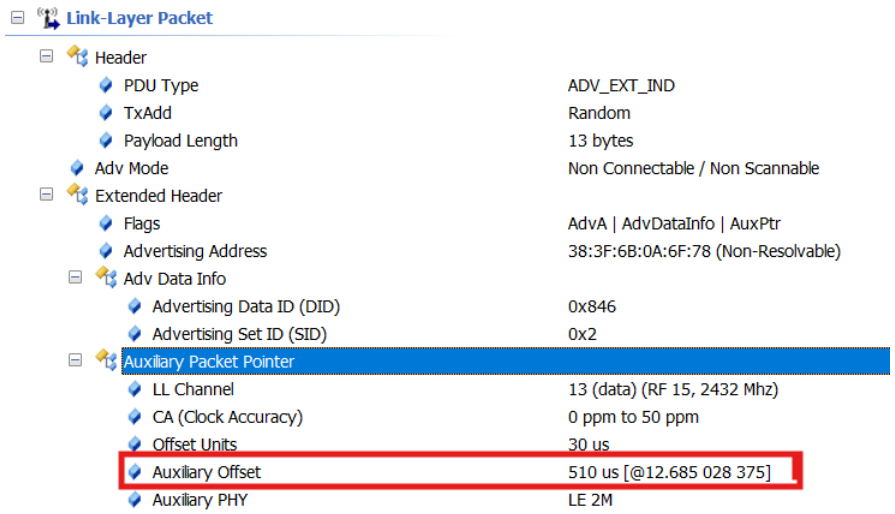


Figure 8 ADV_EXT_IND Packet - Note the analyzer timestamp indicating the location of the AUX_ADV packet.

Scanner Synchronization

Scanners can synchronize their schedule with that of a periodic advertiser in either of two ways: Broadcast Synchronization Information or PAST.

Broadcast Synchronization Information

Extended advertising PDUs of all types use the Common Extended Advertising Payload Format (CEAPF). This defines

standard payload fields, and rules for their inclusion or exclusion.

One of the CEAPF fields is called SyncInfo. It's an optional field in the AUX_ADV_IND PDU and, if present, provides the information a scanning device needs to find and synchronize with a periodic advertising train. The key fields that it includes are:

- **Offset Base, Offset Units and Offset Adjust.** These fields allow the scanner to calculate an offset time, relative to the time at which the first bit of the current packet was received. This indicates when the next AUX_SYNC_IND PDU of the periodic advertising train will be transmitted.
- **Interval.** This is the elapsed time between the start of each pair of successive AUX_SYNC_IND PDUs in the periodic advertising train. The minimum value of the Interval field is 7.5 ms.
- **ChM.** This is the channel map used by the periodic advertising train. It takes the form of a bit map with bit values indicating whether channels whose index corresponds to the bit position are used or unused.
- **PeriodicEventCounter.** This is a 16-bit counter field that is incremented every periodic advertising interval and wraps to zero when 0xFFFF has been reached. It is used in the CSA#2 algorithm, and it is by sharing the current value with scanning devices that they are able to synchronize their selection of channel indices with those selected by the advertising device.

Figure 9 below shows the offset time as calculated from subfields of SyncInfo and the use of the Interval parameter as the periodic advertising interval.

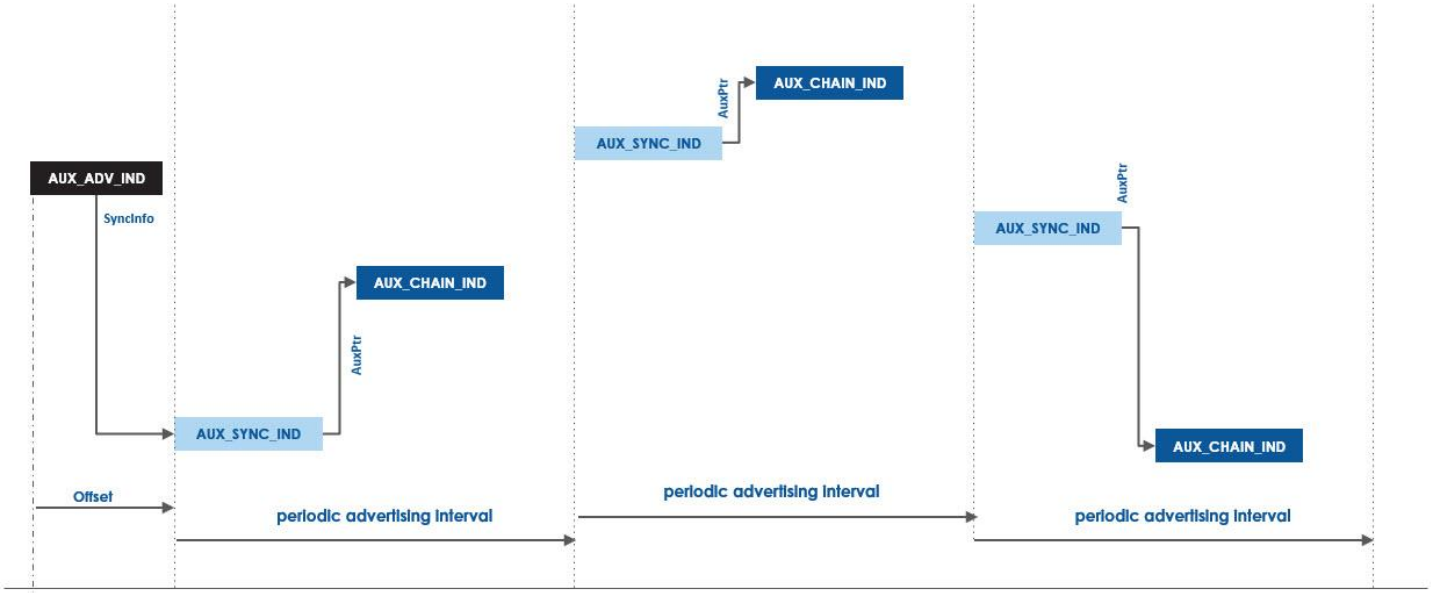


Figure 9 Synchronization using Broadcast SyncInfo

PAST

The Periodic Advertising Sync Transfer (PAST) procedure is a Link Layer control procedure that allows periodic advertising synchronization information **to be provided from one device to another over an LE-ACL connection**.

The Link Layer specification defines the LL_PERIODIC_SYNC_IND PDU which is used to pass the synchronization information between devices. It includes a SyncInfo field which has the same definition as when used within an AUX_ADV_IND PDU.

A common use of PAST is to allow a device that has already acquired synchronization information to share it with another device that wishes to synchronize with the same periodic advertising train but without undertaking the

relatively expensive (in terms of energy use) process of scanning for an AUX_ADV_IND PDU that contains the SyncInfo field. This allows a power rich device like a smartphone to scan for and acquire SyncInfo from broadcast AUX_ADV_INFO PDUs and then pass it to a more constrained device like an earbud. This process is illustrated in Figure 10 below.

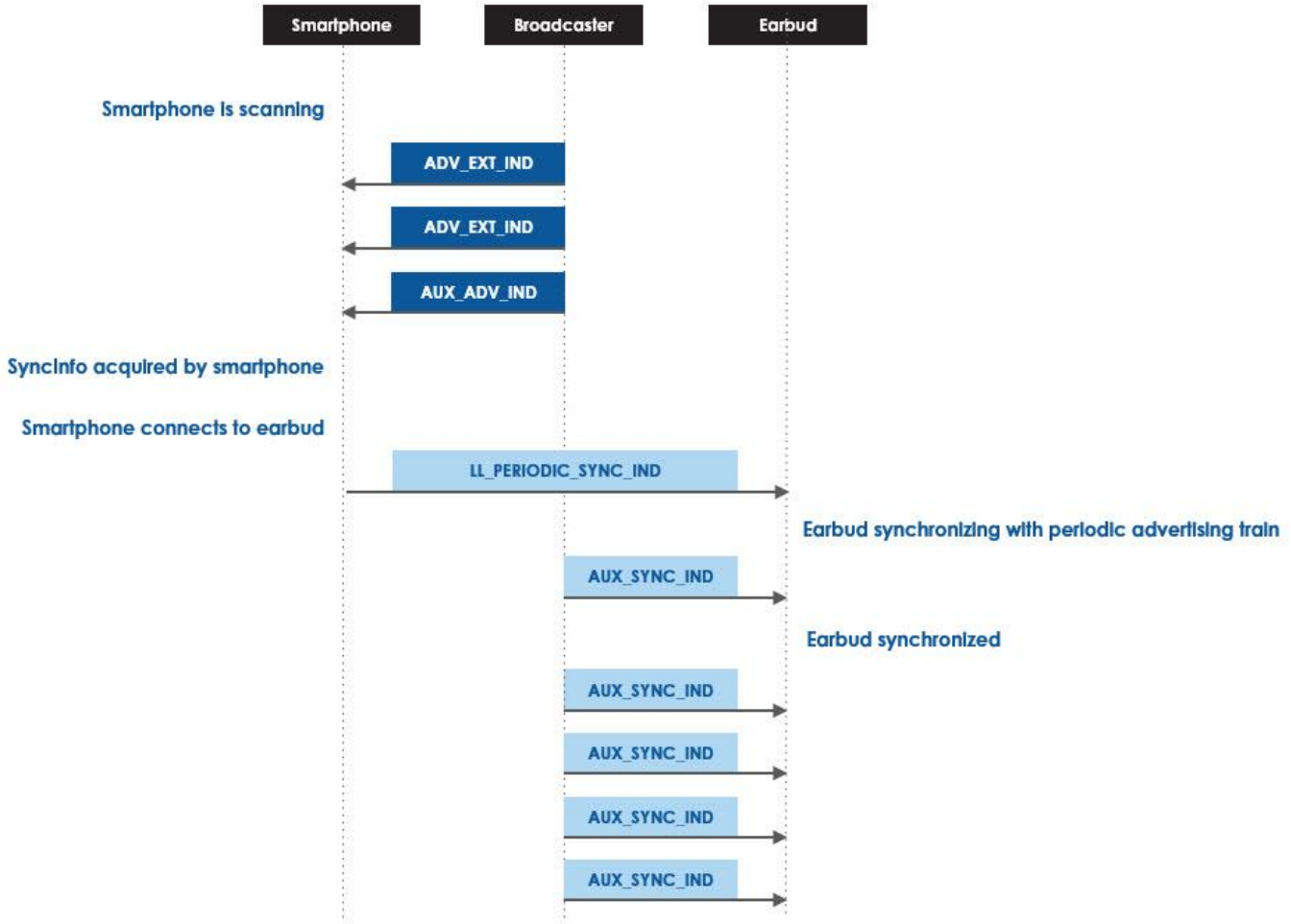


Figure 10 Smartphone providing an earbud with SyncInfo data using PAST

Clock Accuracy

Both the AuxPtr field and the SyncInfo field provide an offset time value which indicates when the Observer should scan for the associated packet. All devices have internal clocks that allow timers to be used for such events. But all clocks exhibit a degree of inaccuracy and so an indication of the accuracy of the Broadcaster's clock, expressed as a parts per million (ppm) measure of the range of possible error is included in both of these fields.

The Physical Layer

Extended advertising PDUs transmitted on the general purpose channels, including periodic advertising AUX_SYNC_IND PDUs, may use the LE 1M, LE 2M or LE Coded PHYs.

AUX_SYNC_IND PDUs must use the same PHY as the AUX_ADV_IND PDU that points to them via the SyncInfo field.

Application Concerns

Configuration

Applications typically use APIs to configure and enable periodic advertising. Such APIs rely on the Host Controller Interface (HCI) which provides a series of commands and events relating to periodic advertising, synchronization and scanning.

API documentation should be reviewed to establish the appropriate details for the relevant platform.

The Broadcaster

A number of configuration steps are involved in configuring and enabling periodic advertising. Each involves one or more HCI commands.

- Step 1 - Configure extended advertising

  The *LE Set Extended Advertising Parameters* HCI command lets the Host application set parameters such as minimum and maximum acceptable advertising interval values, and PHY preferences. An Advertising_Handle parameter identifies the advertising set to which the parameters relate and is used in subsequent HCI commands to identify the same set.

- Step 2 - Configure periodic advertising

  The *LE Set Periodic Advertising Parameters* HCI command allows key parameters such as a range of acceptable periodic advertising interval values to be specified. The Advertising_Handle parameter is used to link these parameters to the same advertising set whose associated extended advertising parameters were configured in step 1.

- Step 3 - Provide the periodic advertising data

  The Host application supplies data to be included in advertising PDUs in one or more calls to the *LE Set Periodic Advertising Data* HCI command. Data can be supplied in fragments.

- Step 4 - Enable periodic advertising

  The Host application must enable periodic advertising for it to start. It does so using the HCI *LE Set Periodic Advertising Enable* command. If extended advertising for the advertising set identified in the command parameters has not yet been enabled, periodic advertising waits until this has been done before starting.

- Step 5 - Enable extended advertising

  Using the *LE Set Extended Advertising Enable* HCI command, extended advertising is enabled for the set specified. If periodic advertising has previously been enabled for the same set, it now starts.

The Observer

Using broadcast SyncInfo

The Observer application must go through a number of configuration steps to receive periodic advertising data.

- Step 1 - Configure and enable scanning for extended advertising

  Configuring and enabling scanning for extended advertising is a precursor to being able to synchronize with a periodic advertising train since the necessary SyncInfo data is transmitted in AUX_ADV_IND PDUs.

  Configuration involves the HCI *LE Set Extended Scan Parameters* command and scanning is enabled using *LE Extended Scan Enable*.

- Step 2 - Synchronize with the periodic advertising train

  To synchronize with a periodic advertising train, an application must invoke the *LE Periodic Advertising Create Sync* command. The command has various parameters and includes the ability to indicate whether or not the delivery of advertising data in HCI reports should be immediately enabled, and whether or not duplicates should be filtered.

  When the Controller has received its first AUX_SYNC_IND PDU and synchronization has been achieved, an HCI *LE Periodic Advertising Sync Established* event is sent to the Host.

Using PAST

To use the PAST procedure, configuring and enabling scanning is not required. Instead, an LE-ACL connection must be established to the device which can supply the synchronization data. The *LE Periodic Advertising Sync Transfer* HCI command can then be used to initiate PAST and includes a connection handle parameter that identifies the LE ACL connection to be used.

- Step 3 - Enable receipt of periodic advertising reports (optional)

  The application can defer enabling the receipt of periodic advertising until some time after synchronizing with the periodic advertising train and enable it when required using HCI *LE Set Periodic Advertising Receive Enable* command.

Application Data

The Broadcaster

As stated, the application layer supplies data to the Controller to be included in periodic advertising AUX_SYNC_IND PDUs using the *LE Set Periodic Advertising Data* HCI command.

Data must be encapsulated in a series of applicable AD Types as with all advertising. It can be fragmented and sent to the Controller in a series of calls to the *LE Set Periodic Advertising Data* command. The command includes a parameter called Operation which is used to indicate whether a fragment is the first in the series, an intermediate fragment, or the final fragment.

If periodic advertising has not yet been enabled when data is supplied to the Controller, it is retained. When periodic advertising is enabled, the supplied data is used in AUX_SYNC_IND PDUs and if necessary, auxiliary AUX_CHAIN_IND PDUs.

If periodic advertising has already been enabled, the data supplied replaces that which was already in use (if any). In this way, applications can change the data being advertised when necessary.

The Observer

Periodic advertising data is delivered to Observer devices in an HCI event type called LE Periodic Advertising Report.

If necessary, the Controller may fragment received data and transfer it to the Host in a series of advertising reports. A field in the event called Data Status indicates to the Host whether or not there is more data to follow in subsequent reports.

Reliability

PADVB is classified as connectionless communication but is different to ADVB in that scanner and advertiser schedules are synchronized. This can result in more reliable communication since there's less risk of broadcast packets being missed due to the scanner being out of sync with the broadcaster.

PADVB does not include a system of acknowledgements but if broadcast data is not varying frequently, this should not be an issue since synchronized scanners will have multiple opportunities to receive a particular set of broadcast data.

The use of the 37 general purpose channels and a randomized channel selection algorithm makes PADVB robust in the face of interference.

Security

Encrypted Advertising as described in previous articles in this series may be used with periodic advertising.

Periodic advertising is anonymous with no advertiser's address or target device address included in AUX_SYNC_IND PDUs, and so privacy issues relating to device tracking using addresses is not an issue. For the threat of device tracking to be fully addressed however, applications should consider ways of varying the application data payload so that it cannot form a static pattern and be used for tracking purposes.

Ellisys Bluetooth Analyzer Examples

Bluetooth extended advertising is a little complicated. And periodic advertising even more so. But using an Ellisys Bluetooth Analyzer system with real devices provides clarity and insight into the theoretical behaviors that are defined in the Bluetooth Core Specification.

A series of screenshots follow which trace through the packets and PDUs involved in the communication of periodic advertising synchronization information and those of the associated PADVB advertising train itself. The device generating the depicted packets is a Nordic nRF54L15 developer board, running a sample periodic advertising application that is included with the Zephyr open-source operating system.

Figure 11 below shows a series of extended advertising packets followed by the first AUX_SYNC_IND PDU of a periodic advertising chain. We can see at a glance that there are three ADV_EXT_IND (one per primary advertising channel), an auxiliary advertising packet and then an AUX_SYNC_IND PDU that belongs to an associated periodic advertising train.

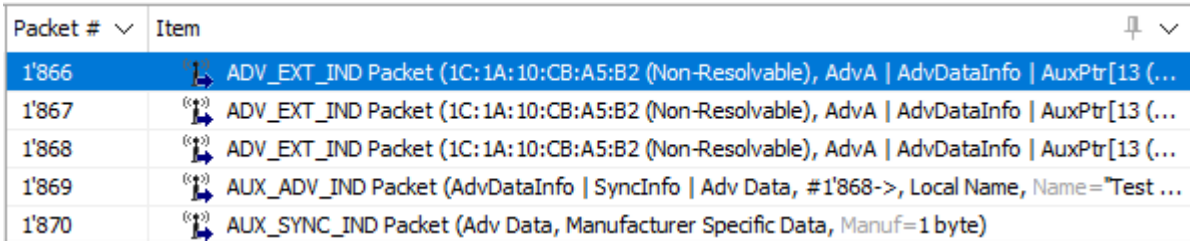


Figure 11 Extended advertising and the first periodic advertising packet in Ellisys Bluetooth Analyzer software

Figure 12 below shows a detailed breakdown of one of the ADV_EXT_IND PDUs shown in Figure 11 (obtained by selecting on one of the rows shown in Figure 11). We can see that it was transmitted on primary advertising channel #37 using the LE 1M PHY.

The AuxPtr field indicates that an auxiliary packet will be transmitted on channel index 13 using the LE 2M PHY and that this will occur 1.110 ms from the time at which the first bit of this packet was received. Helpfully, this time is shown as a timestamp with a value of 14.267018875 seconds. A clock accuracy measure of 0 - 50 ppm is also provided.

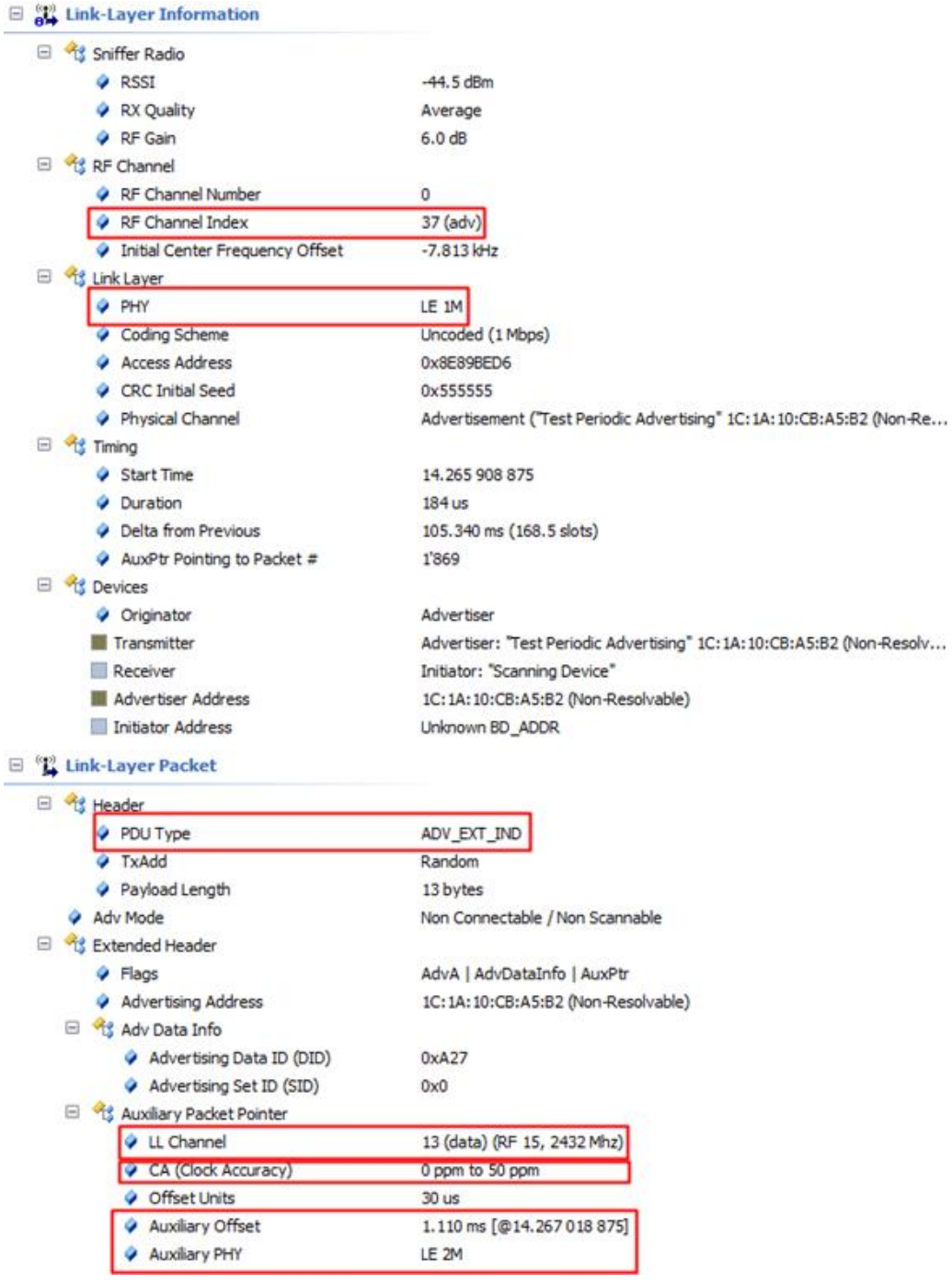


Figure 12 An ADV_EXT_IND PDU pointing to an auxiliary PDU in Ellisys Bluetooth Analyzer software

Figure 13 shows the details of the auxiliary packet. It contains an AUX_ADV_IND PDU that was transmitted as expected, on channel index 13 using the LE 2M PHY.

The time that the start of this packet was received has a time stamp allocated by the Analyzer tool of 14.267046375 seconds. The predicted transmission time indicated on the superior ADV_EXT_IND PDU was 14.267018875. Subtracting one from the other yields a value of -0.0000275 or 27.5 ppm which is within the stated accuracy of 0 - 50 ppm.

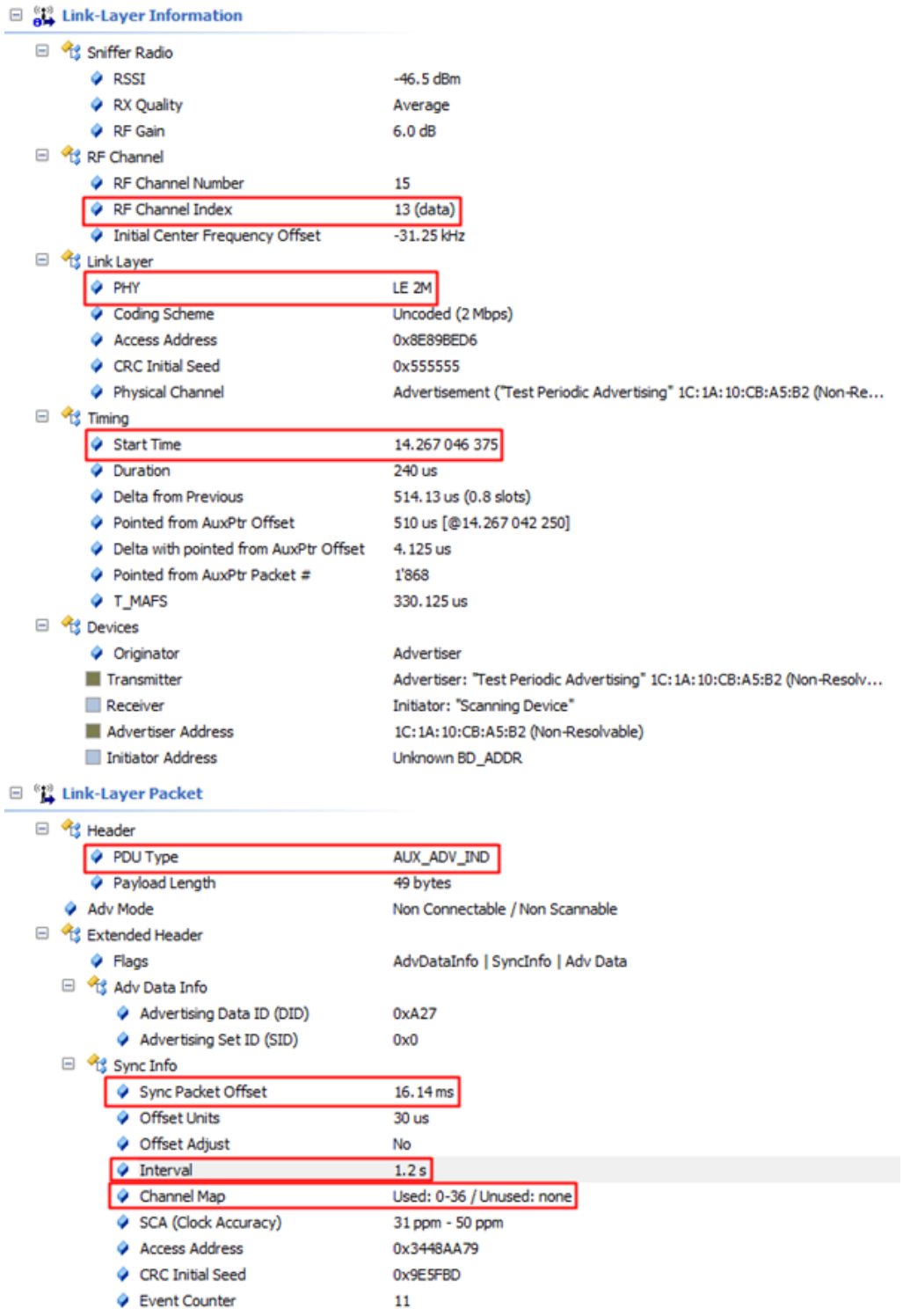


Figure 13 AUX_ADV_IND in Ellisys Bluetooth Analyzer software

The AUX_ADV_IND PDU in Figure 13 contains a SyncInfo field. The calculated Sync Packet Offset value indicates that the next AUX_SYNC_IND PDU of the associated periodic advertising train will be transmitted in 16.14 ms. It also indicates that the periodic advertising interval in use by this periodic advertising train is 1.2 seconds and that all 37 of the general purpose channels are currently marked as in use.

Figure 14 shows the first AUX_SYNC_IND PDU transmitted after the AUX_ADV_IND PDU in Figure 13. It was transmitted on channel index 1 using the LE 2M PHY. It has an AdvData field which contains only one AD Type. This is the Manufacturer Data field, which has two subfields. The company ID used in this case was a reserved value, which elsewhere in the Analyzer tool we can see is 0xFFFF (used for testing only). The Manufacturer Specific Data field contains the value 0x01.

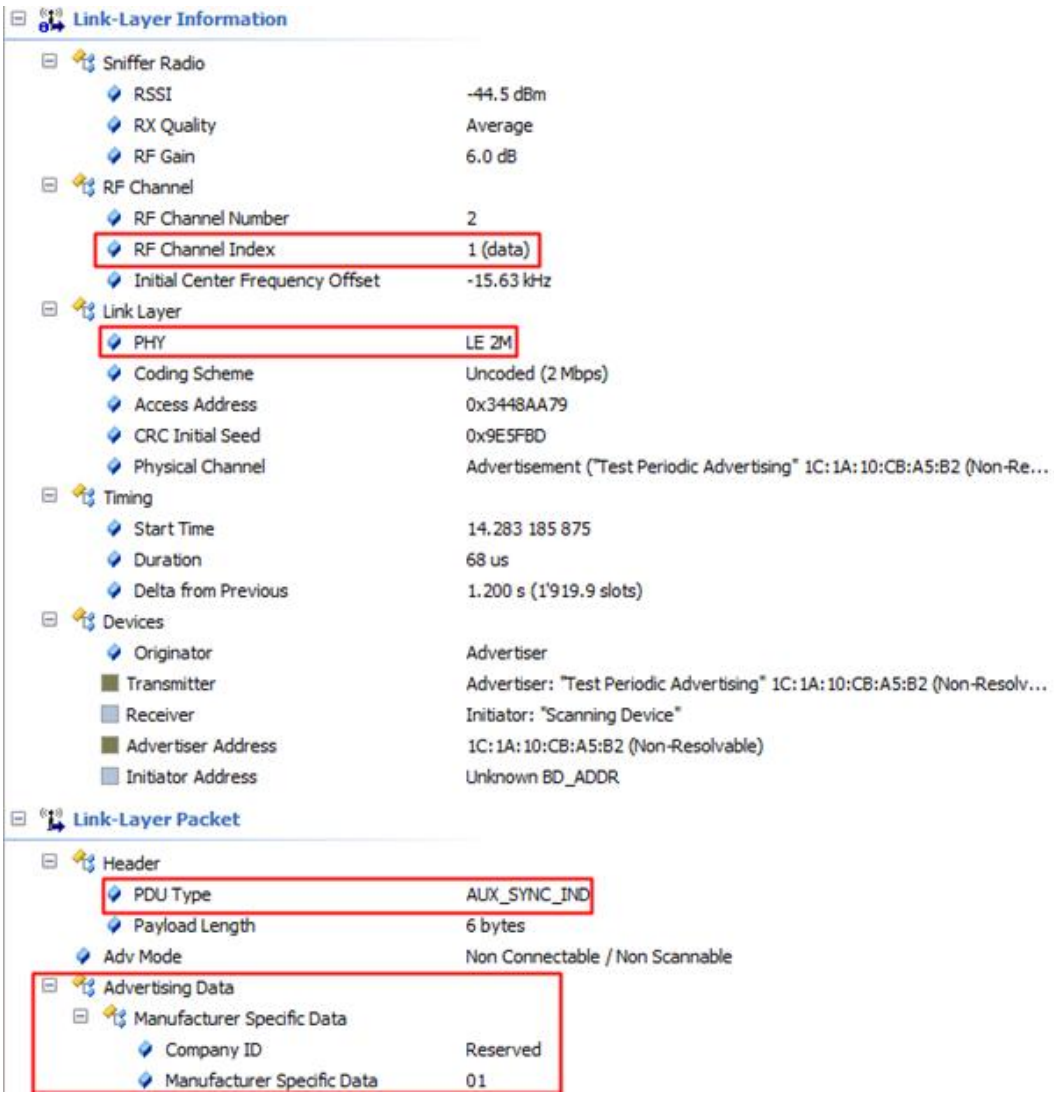


Figure 14 AUX_SYNC_IND in Ellisys Bluetooth Analyzer software

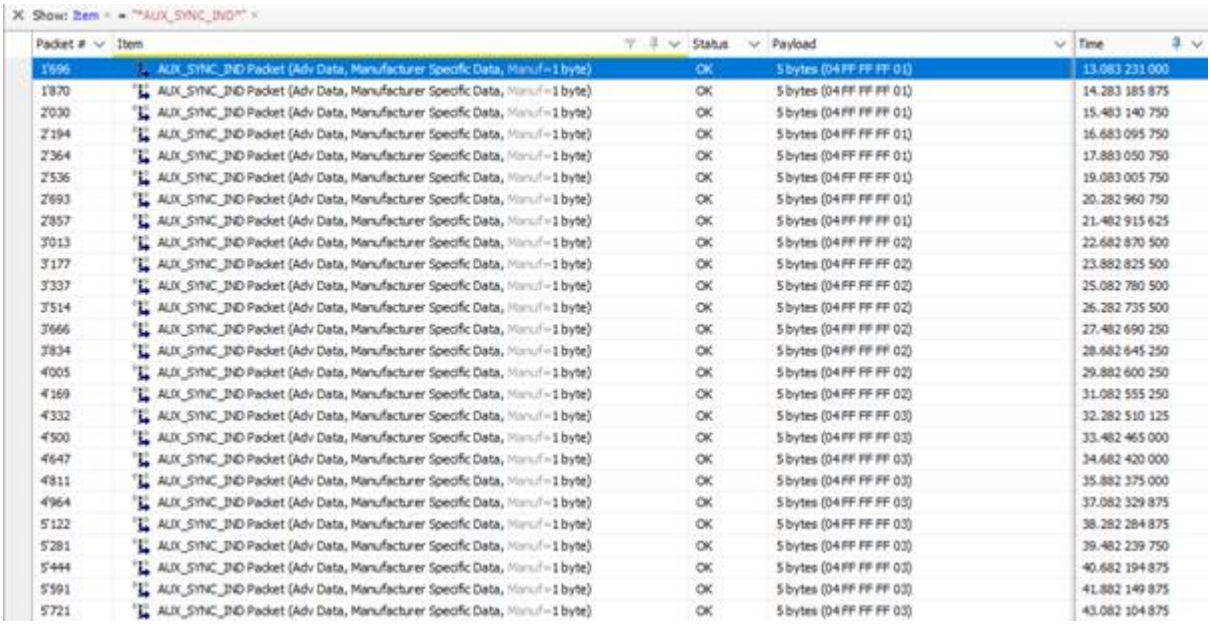
Note that after periodic advertising has started, the Zephyr sample code cycles the content of the manufacturer data AD Type in the AdvData field through the integer values 0x01, 0x02 and 0x03, changing the current advertised value every 10 seconds. See Figure 15 below.

```
for (int i = 0; i < 3; i++) {
    k_sleep(K_SECONDS(10));

    mfg_data[2]++;

    printk("Set Periodic Advertising Data...");
    err = bt_le_per_adv_set_data(adv, per_adv_ad, ARRAY_SIZE(per_adv_ad));
    if (err) {
        printk("Failed (err %d)\n", err);
        return 0;
    }
    printk("done.\n");
}
```

Figure 15 Code loop changing the advertised data every 10 seconds

Using a filter pattern of Item= "*AUX_SYNC_IND*", it's easy to isolate all AUX_SYNC_IND PDUs in the Ellisys Bluetooth Analyzer "Packets" view. This is shown in Figure 16 below.

In the Payload column, we can see the Manufacturer Specific Data field (the final byte) changing, as expected. We can also see evidence of the periodic advertising interval of 1.2 seconds at work by examining the Time column.



Packet #	Item	Status	Payload	Time
1'696	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 01)	13.083 231 000
1'870	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 01)	14.283 185 875
2'030	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 01)	15.483 140 750
2'194	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 01)	16.683 095 750
2'364	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 01)	17.883 050 750
2'536	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 01)	19.083 005 750
2'693	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 01)	20.282 960 750
2'857	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 01)	21.482 915 625
3'013	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 02)	22.682 870 500
3'177	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 02)	23.882 825 500
3'337	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 02)	25.082 780 500
3'514	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 02)	26.282 735 500
3'666	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 02)	27.482 690 250
3'834	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 02)	28.682 645 250
4'005	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 02)	29.882 600 250
4'169	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 02)	31.082 555 250
4'332	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	32.282 510 125
4'500	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	33.482 465 000
4'647	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	34.682 420 000
4'811	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	35.882 375 000
4'964	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	37.082 329 875
5'122	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	38.282 284 875
5'281	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	39.482 239 750
5'444	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	40.682 194 875
5'591	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	41.882 149 875
5'721	AUX_SYNC_IND Packet (Adv Data, Manufacturer Specific Data, Manuf=1 byte)	OK	5 bytes (04 FF FF FF 03)	43.082 104 875

Figure 16 List of AUX_SYNC_IND PDUs in Ellisys Bluetooth Analyzer software

Communication Mode Properties

The first article in this series on The Many Communication Modes of Bluetooth LE introduced a set of properties that can be useful in comparing one communication mode with another. Here is the table of properties for periodic advertising.

Property	Comment
Topology	Undirected: one-to-many (1:m).
Transmitters vs Receivers	Advertising devices transmit only. Scanners receive only.
Application Data Direction	One-way. Application data can be transmitted only from the advertising device to scanning devices.
Connected or Connectionless?	Connectionless but coordinated.
Data and Time	Asynchronous.
Receiver Concurrency	An unlimited number of scanning devices can receive periodic advertising packets at the same time.
Radio Channels	The 37 Bluetooth LE general purpose channels are used by PADVB. The 3 primary advertising channels are involved in acquiring synchronization information.
Scalability	Periodic advertising is always undirected and can reach a theoretically unlimited number of devices so in this respect, PADVB is highly scalable. PADVB is a form of extended advertising and therefore allows significantly more application data to be transmitted per advertising event than $ADVB_L$.
Choice of PHY	Transmission of AUX_SYNC_IND PDUs uses either LE 1M, LE 2M or LE Coded but the selected PHY must match that of the associated AUX_ADV_IND PDU(s).

Table 1 PADVB Properties

Did You Know?

As we approach the end of this article we'll close with a series of interesting and useful additional points about periodic advertising.

Did you know that a new logical transport called Periodic Advertising with Responses (PAwR) was introduced in version 5.4 of the Bluetooth Core Specification?

PADVB only allows the transmission of application data by the Broadcaster device. PAwR allows periodic advertising to be used and for Observer devices to transmit their own application data back to the Broadcaster. PAwR will be the subject of the next article in this series.

Did you know that the core specification states that advertising pointing to a periodic advertising train shall not be anonymous?

This means that an AUX_ADV_IND PDU that includes the SyncInfo field must always have a device address in the AdvA advertiser address field. You can see an example of this in Figure 13.

Did you know that packets in a periodic advertising train may include a Constant Tone Extension?

A Constant Tone Extension (CTE) is used in the Bluetooth direction finding system and so it can be useful to some applications to include this field. That said, the Host can indicate to the Controller whether or not periodic advertising trains either with or without the CTE field should be considered relevant.

Did you know that neither Android or iOS currently provide an API for synchronizing with a periodic advertising train?

The chances are good that this will change. The new Bluetooth LE audio technology, Auracast involves periodic advertising and it's been announced that Android 16 will support Auracast. No official announcement has been made regarding Apple and Auracast yet. Whether supporting Auracast will result in APIs that allow direct synchronization with periodic advertising from an app remains to be seen, of course.

Next in the Series

In this article we've explored the periodic advertising (PADVB) communication mode of Bluetooth LE.

In the next article in this series, we'll explore the periodic advertising with responses (PAwR) communication mode.