The Many Communication Modes of Bluetooth® LE

PAwR – Periodic Advertising with Responses

Introduction

In the world of Bluetooth, the term *advertising* applies to a collection of different communication modes, all of which have certain things in common. Advertising entails *connectionless communication* and the repeated transmission of packets. Usually, advertising devices *broadcast* their packets so that they are available to multiple receiver devices. Typically, the application data in advertising packets changes infrequently or even not at all.

Extended advertising is a term that first appeared in version 5.0 of the Bluetooth Core Specification. It describes a form of advertising that is more advanced than the original advertising capability that the first version of Bluetooth Low Energy (LE) defined, and which is now known as *legacy advertising*.

Amongst the qualities that make extended advertising more advanced than legacy advertising is the fact that many more radio channels are used. Larger application data payloads are also supported.

The different Bluetooth communication modes are more formally called *logical transports*. Several of the communication modes entail a connectionless, advertising capability. In previous articles in this series, we've explored the advertising broadcast logical transport (ADVB) in both its legacy and extended forms, and the periodic advertising (PADVB) logical transport.

In this article we turn our attention to the newest Bluetooth LE logical transport, Periodic Advertising with Responses which is known as PAwR for short.

Note: *Before reading this article, it is recommended that the previous articles in this series are read first.*

Understanding Periodic Advertising with Responses

As was discussed in the previous article in this series, periodic advertising (i.e. the PADVB logical transport) is characterized by a fixed rate, regular schedule of events during which the Broadcaster transmits advertising packets. Observers can acquire details of the Broadcaster's schedule and synchronize their scanning activity to it.

The PAwR logical transport shares these basic characteristics but is designed to meet a very basic need that is not met by PADVB. PAwR supports the *bidirectional* exchange of application data between a Broadcaster and a potentially very large number of other devices whereas PADVB supports the transmission of application data in one direction only, from Broadcaster to Observers.

PAwR is not the only logical transport that supports bidirectional communication of application data. Devices can also perform bidirectional application data communication over LE-ACL connections but it's common for devices to support no more than say 4 concurrent connections. In contrast, PAwR can support bidirectional communication with thousands of devices.

Note: *Periodic advertising (PADVB) is sometimes referred to as periodic advertising* ***without*** *responses to distinguish it from its newer cousin, periodic advertising* ***with*** *responses (PAwR).*

The key motivation for developing the PAwR logical transport was to provide a standards-based Bluetooth transport for a type of product known as an electronic shelf label (ESL). But as you'd expect of any well-designed communications stack, the PAwR logical transport is in no way coupled to ESL and is a generalized transport that should support a range of applications in the future. But it is ESL that was the source of many of the requirements that make PAwR what it is today. We'll revisit ESL in Section 4.

Overview

PAwR is similar to PADVB in that:

1. The Broadcaster's transmission of packets is governed by a regular, unvarying periodic schedule.
2. Observer devices can acquire the Broadcaster's schedule and synchronize their own activities to it.
3. The general communication topology can be described as one-to-many.
4. The Communication method is connectionless.

Key differences between PAwR and PADVB are that:

1. PAwR supports bidirectional application data communication
2. The PAwR scheduling scheme is more complicated than that of a PADVB advertising train
3. A PAwR Broadcaster can initiate a connection with one of its Observers. PADVB does not have this capability.
4. PAwR enables communication with subsets of the synchronized Observers.
5. The design of PAwR assumes that application data payloads will change frequently.

In the following sections, these points will be explored further.

Advertising Modes

PAwR can be used in one advertising mode only. Advertising is always non-connectable and non-scannable. The AdvMode always has a value of 0x00 to indicate this.

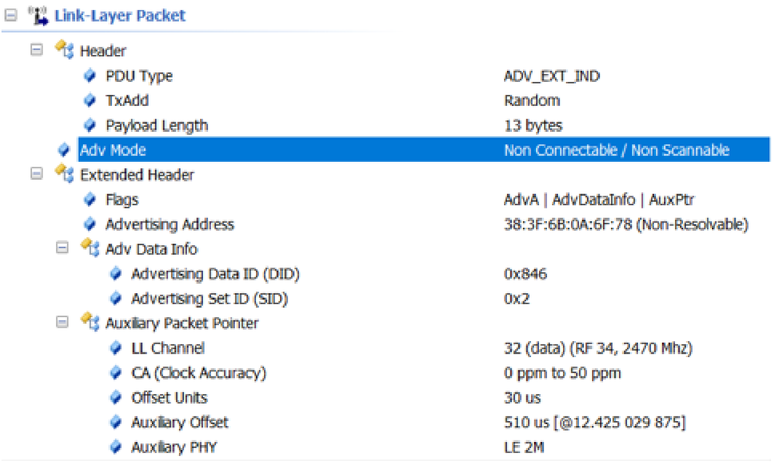

Figure 1 ADV_EXT_IND Packet Contents (Ellisys Analyzer Application)

Device Roles

The advertising device assumes the GAP Broadcaster role and devices that receive advertisements assume the GAP Observer role.

States

The Link Layer defines a formal state machine, depicted in Figure 2, below.

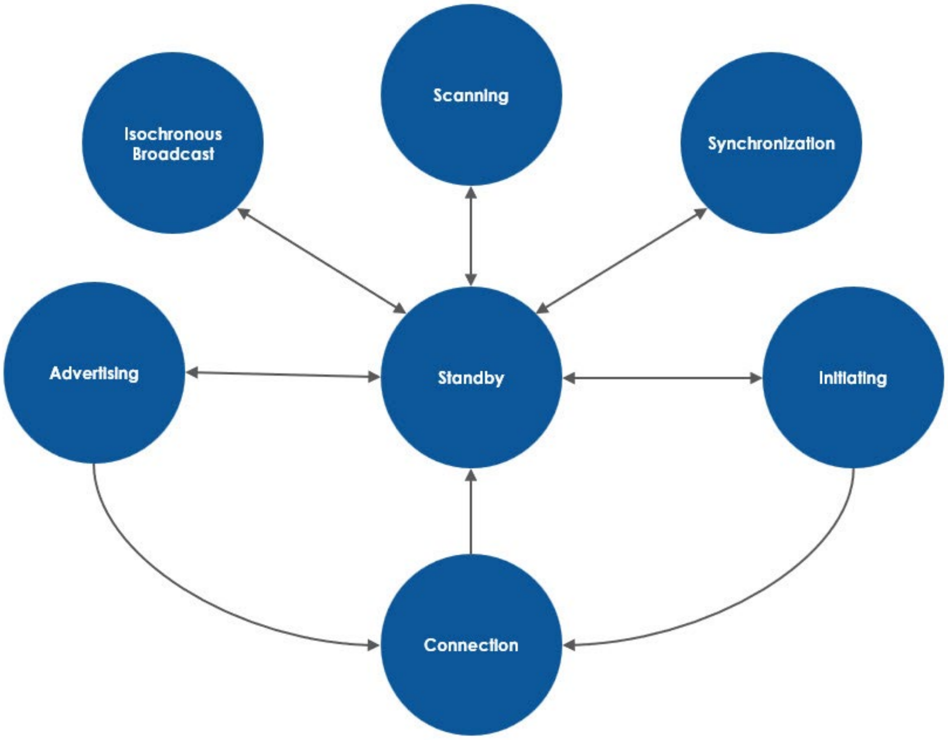

Figure 2 Link Layer State Machine

The Synchronization state is relevant to PAwR and has two sub-states: Synchronizing and Synchronized. These are depicted in Figure 3.

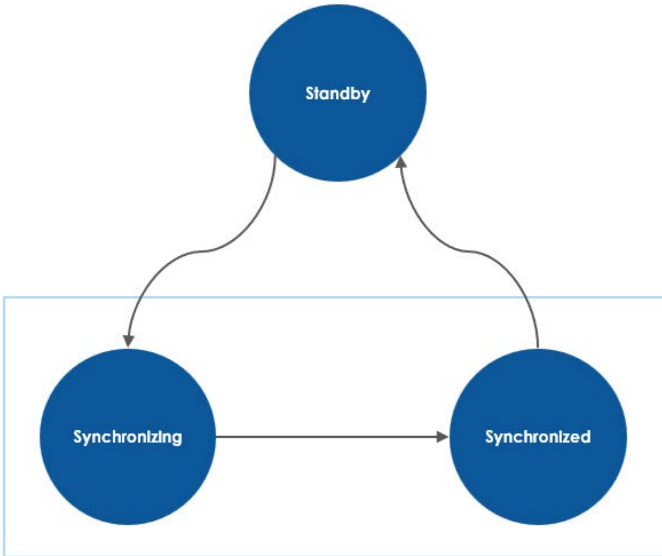

Figure 3 The Synchronization substates

Observer devices can acquire information about the schedule of the PAwR advertising device and having done so, the Host can instruct the Controller to enter the Synchronizing state. Once a PDU from the PAwR broadcaster has been received, the Controller's state changes to Synchronized.

Packets, PDUs and Radio Channels

PDUs and Relationships

The Broadcaster device typically transmits AUX_SYNC_SUBEVENT_IND PDUs. Occasionally, it might transmit an AUX_CONNECT_REQ PDU to initiate the establishment of an LE-ACL connection with one of the Observer devices.

The Broadcaster's transmission schedule and other parameters of the PAwR advertising train are published in AUX_ADV_IND PDUs in much the same way that equivalent parameters are published for a PADVB advertising train. There are additional parameters for PAwR though, and to accommodate the publication of these extra parameters, the Additional Controller Advertising Data (ACAD) field is used as well as the Sync Info field.

Figure 4 below illustrates the PDUs transmitted by the Broadcaster.

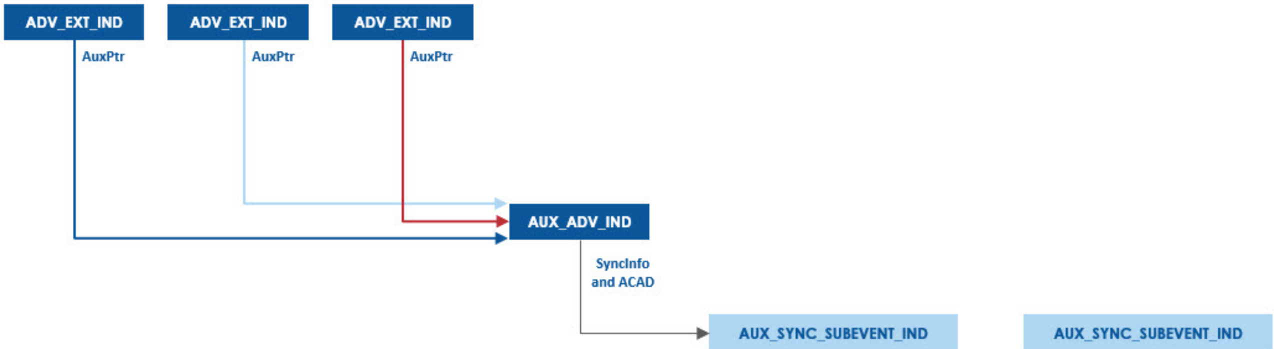

Figure 4 The primary PDU types transmitted by a Broadcaster in PAwR

Figure 5 shows the ACAD field, following the Sync Info field, in an AUX_ADV_IND packet (Ellisys Bluetooth Analyzer Application). These fields (and their subfields) are further discussed in the Scheduling and Scanner Synchronization sections, below, and in the Ellisys Bluetooth Analyzer Examples section.

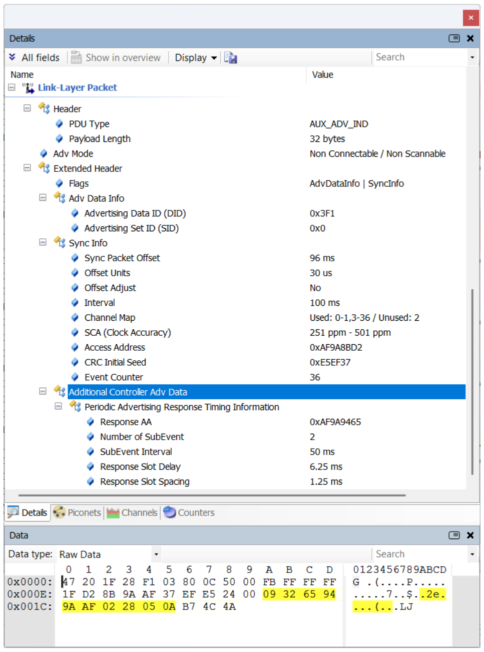

Figure 5 Sync Info and ACAD fields (ACAD highlighted) in AUX_ADV_IND

Observers sometimes transmit AUX_SYNC_SUBEVENT_RSP PDUs in reply to AUX_SYNC_SUBEVENT_IND PDUs. These response PDUs are received and processed by the Broadcaster.

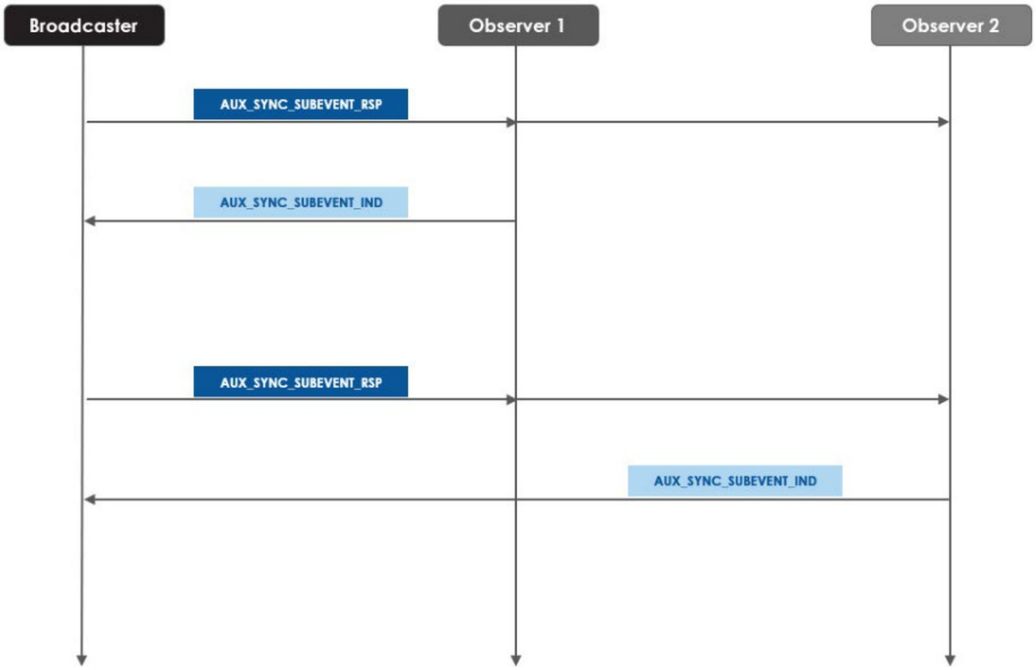

Figure 6 Observers transmitting response PDUs

Radio Channels

Like periodic advertising *without* responses (PADVB), PAwR uses the 37 general-purpose channels. A channel is selected using Channel Selection Algorithm #2 (CSA #2) and response PDUs are transmitted by Observers on the same channel that the corresponding AUX_SYNC_SUBEVENT_IND PDU was received on.

The ADVB extended advertising communication mode is used to publish PAwR parameters, including synchronization information, and this uses all 40 Bluetooth LE channels.

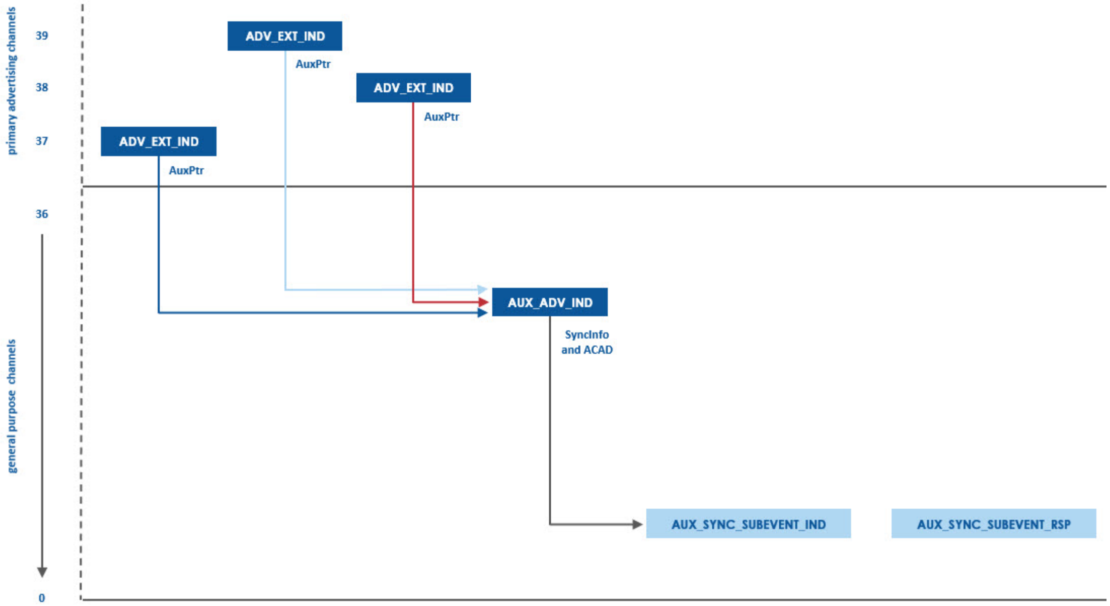

Figure 7 - PAwR (and ADVB$_E$) radio channel use

Scheduling

For scheduling purposes, PAwR divides time into a series of events, subevents, and response slots.

A PAwR event starts every *periodic advertising interval* ms.

An event is divided into a number of subevents, and this is specified in a parameter called *num subevents* which has a maximum value of 128.

A subevent starts at the same time as an event and then every *subevent interval* ms. Channel selection using CSA #2 takes place ahead of each subevent

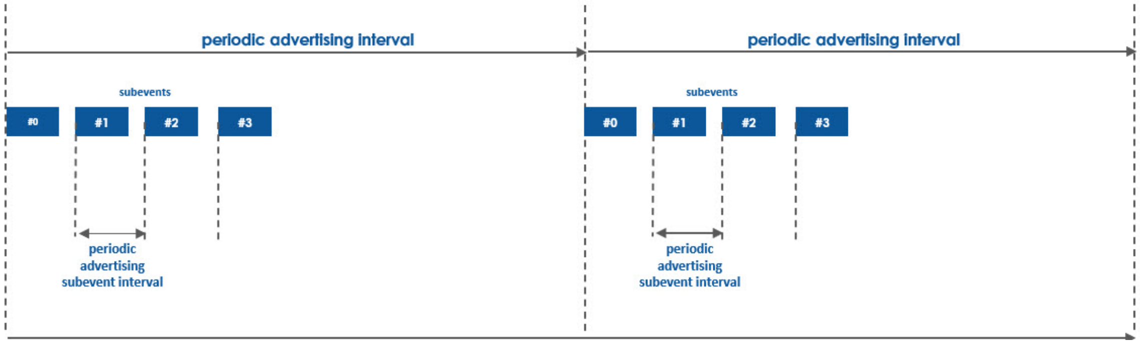

Figure 8 Scheduling of PAwR subevents within PAwR events with Num Subevents = 4

Each subevent consists of a time slot during which a packet can be transmitted by the Broadcaster, followed by a delay called *response slot delay* and then a number of time slots which Observers may use for the transmission of response PDUs. Response slots are spaced along the timeline by a parameter called *response* slot spacing. Figure 9 illustrates the structure of a PAwR subevent.

Observers must use a particular *access address* in packets containing response PDUs. The value to be used is supplied in a synchronization parameter called RspAA.

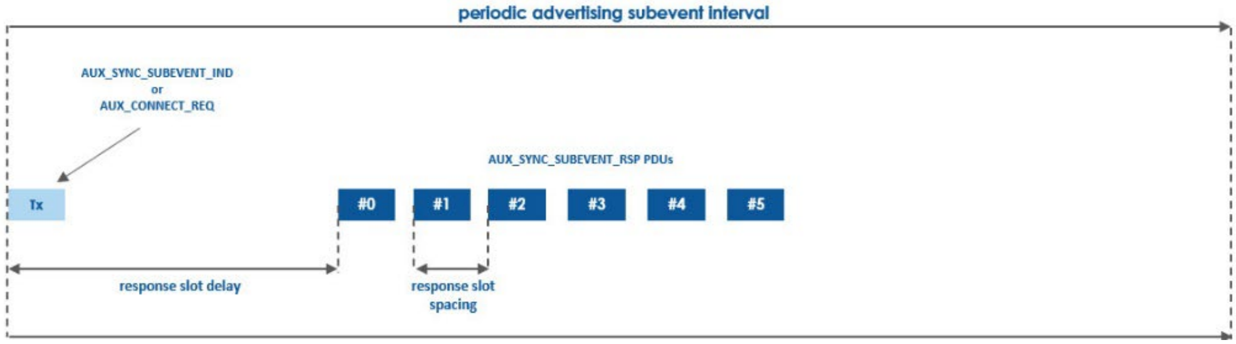

Figure 9 Structure of a PAwR subevent.

Scanner Synchronization

Broadcast Synchronization Information

Periodic advertising *with* responses addresses most aspects of scanner synchronization in the same way as for periodic advertising *without* responses[1].

The SyncInfo field of AUX_ADV_IND PDUs includes subfields that allow the calculation of an offset time relative to the first bit of the packet, and this indicates when the next PAwR event will start.

The periodic advertising interval, channel map and periodic event counter are also to be found in the SyncInfo field. Further parameters that PAwR needs are published in the ACAD field which is another field defined by the Common Extended Advertising Payload Format (CEAPF).

Table 1 below summarizes the parameters and where they are to be found during synchronization.

[1] See the article on periodic advertising (PADVB) to review this information.

Parameter (s)	Found in	Description
Offset Base, Offset Units and Offset Adjust	SyncInfo	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 PAwR event will start.
periodic advertising interval	SyncInfo	Defines how often a PAwR event starts.
ChM	SyncInfo	This is the channel map as used by PAwR. 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*.
periodic event counter	SyncInfo	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.
num subevents	ACAD	Defines the number of PAwR subevents per event.
subevent interval	ACAD	Indicates the elapsed time between the start of PAwR subevents.
response slot delay	ACAD	Specifies the elapsed time between the start of a subevent and the start of the first response slot.
response slot spacing	ACAD	Indicates the elapsed time between the start of adjacent response slots.
RspAA	ACAD	The access address that must be used in packets containing response PDUs.

Table 1 PAwR synchronization parameters

The ACAD field contains data packaged in AD Types[2]. For PAwR parameters the Periodic Advertising Response Timing Information AD Type is defined.

PAST

The Periodic Advertising Sync Transfer (PAST) Link Layer procedure can be used to pass PAwR synchronization to an Observer device over a LE-ACL connection.

Periodic advertising without responses uses the LL_PERIODIC_SYNC_IND PDU to send synchronization parameters over a connection. PAwR however, uses the LL_PERIODIC_SYNC_WR_IND PDU. A LL_PERIODIC_SYNC_WR_IND PDU can contain all the same fields as a LL_PERIODIC_SYNC_IND PDU plus those fields defined by the Periodic Advertising Response Timing Information AD Type.

The Application Layer and Synchronization

The data broadcast in SyncInfo and ACAD fields or shared using the PAST procedure does not provide all of the synchronization information that a PAwR Observer device needs and leaves unanswered questions. A device could, for example, scan all subevents, some subevents or just one specific subevent but which of these options applies is not indicated using either of the two synchronization methods. How this is dealt with will be covered in the Application Concerns section, below.

---

[2] AD Types were introduced in the legacy advertising article.

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 errors is included in both of these fields.

The Physical Layer

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

AUX_SYNC_SUBEVENT_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

Configuring the Broadcaster for PAwR involves similar steps to those used to configure PADVB. However, application data is not supplied during initial configuration. Additional parameters are also required.

Importantly, if the Num_Subevents parameter in the LE Set Periodic Advertising Parameters command is greater than zero, PAwR will be used rather than PADVB and when this is the case, the command also requires values to be supplied for the Subevent_Interval, Response_Slot_Delay, Response_Slot_Spacing, and Num_Response_Slots parameters.

The Observer

Configuration of a PAwR Observer must include indicating to the Controller the PAwR subevent or subevents to synchronize with. This is accomplished using the LE Set Periodic Sync Subevent HCI command which allows an array of 1 or more subevent numbers to be specified. The basis for deciding which subevent or subevents to synchronize with is the concern of the application layer, however.

Application Data

The Broadcaster

PAwR is designed with the assumption that application data will change every subevent. This does not necessarily have to be the case, and this will vary from application to application. But to accommodate the idea, the Controller regularly asks the Host for the application data for a series of future subevents by sending an LE Periodic Advertising Subevent Data Request HCI event. The Bluetooth Core Specification recommends that as much data is requested as the Controller has memory available so that the Controller always has a significant cache of application data at the ready and HCI exchanges are reduced.

The Host uses the LE Set Periodic Advertising Subevent Data command to send data to the Controller for use in AUX_SYNC_SUBEVENT_IND PDUs.

The Observer

The Observer must provide its Controller with application data to use in response slots. It does this using the LE Set Periodic Advertising Response Data HCI command.

Reliability

PAwR shares the same characteristics as periodic advertising *without* responses with respect to reliability. It's connectionless, uses all 37 general-purpose channels with channel selection algorithm #2, and has no transport layer acknowledgement system.

However, it is possible for the application layer to implement a simple acknowledgement system based on response PDUs, so in this respect, PAwR stands to offer a more reliable one-to-many transport than PADVB.

Energy Efficiency

PAwR was designed to be energy efficient and for the types of product that would use it to be able to run off coin cell batteries for literally years.

Key to the energy efficiency of PAwR are two factors:

1. Observers scan during brief subevents and depending on application layer logic will often scan only during one subevent. The minimal time spent scanning reduces the energy that needs to be used significantly.
2. The basis for selecting subevent(s) to scan on comes from application layer logic. This results in a high probability that data received by the Observer is relevant to it.

Security

Encrypted Advertising may be used with PAwR to protect the confidentiality of application data. PAwR is anonymous with no advertiser address included in PDUs and so the tracking of Broadcaster devices engaged in PAwR cannot be achieved using the device's Bluetooth address.

Any static, distinct pattern of data in transmissions can be the basis for tracking a device. The fact that PAwR is likely to be transporting frequently changing application data payloads makes it automatically good at avoiding this possibility.

Latency

PAwR, with its ability to accommodate new application data for each subevent has all the appearances of a messaging transport. But it's important to note that the scheduling system with its intervals, events and subevents can mean that the time that could elapse from the start of a use case at the application layer to the transmission of all resultant

AUX_SYNC_SUBEVENT_IND PDUs and receipt of all responses could take tens of seconds depending on configuration values. PAwR is not intended for real-time or even near real-time messaging applications.

Electronic Shelf Labels

Overview

Electronic shelf labels (ESLs) are used in large retail stores and as suggested by the name, act as labels for goods that are on sale and that are typically presented on shelves. They have a display which can usually show images as well as text, usually have one or more colored LEDs and sometimes have sensors.

Figure 10 Typical Electronic Shelf Label

ESLs support wireless communication. Historically, proprietary 2.4 GHz protocols tended to be used but at the same time that Bluetooth Core Specification v5.4 was released in 2023, the Electronic Shelf Label Profile specification was also released, paving the way for standardization in ESL products.

The ESL profile is designed around a hub/spoke, one-to-many topology with a single device called the *Access Point (AP)* at the center, communicating with ESL devices which are potentially numbered in the thousands. The AP can send commands to ESLs, for example to change the image displayed, to enable or disable an LED, or to request a sensor value.

PAwR is used as the logical transport for most AP and ESL communication. The AP sends *ESL commands* in the payload of AUX_SYNC_SUBEVENT_IND PDUs. ESLs send ESL response messages in AUX_SYNC_SUBEVENT_RSP PDUs.

Occasionally, the AP will connect to an ESL by sending an AUX_CONNECT_REQ PDU and then execute defined operations that rely on an LE-ACL connection such as the upload of an image file.

ESL and PAwR

ESL Addresses

During their initial set-up, each ESL is assigned an address that is unique in the AP’s network of ESLs. The address consists of two parts:

- An 8-bit ESL identifier
- A 7-bit Group ID

From this we can conclude that a network of ESLs, controlled by a single AP can consist of a maximum of 128 groups, each of which can contain up to 255 ESLs. In total therefore, 32,640 ESLs can be members of an ESL network with a single AP at its heart.

ESLs and PAwR Subevents

The ESL Group ID has an important role. It is the Group ID that indicates to an ESL which PAwR subevent it is to listen on. As such, all ESLs in the same group listen during the same PAwR subevent and therefore they all receive the same AUX_SYNC_SUBEVENT_IND PDUs broadcast by the AP.

An AUX_SYNC_SUBEVENT_IND PDU received by an ESL contains an array of one or more ESL commands. Each command includes an ESL address, the Group ID part of which is the same for all commands in the PDU. On receiving a PDU and extracting the ESL commands, an ESL checks the ESL ID of each command against its own ID and only executes commands where there is an ID match.

PAwR response slot allocation

The index position of a command in the ESL command array indicates the number of the response slot to be used by the ESL to which the command is addressed. For example, if an ESL finds its ID in the address of a command and the command is second in the command array (i.e. it has an index value of 1), it uses PAwR response slot #1 for its AUX_SYNC_SUBEVENT_RSP transmission.

ESLs and PAwR Synchronization

In summary, the ESL Profile defines a way for PAwR synchronization to be completed. PAwR scheduling parameters are acquired from broadcast AUX_ADV_IND PDUs or using the PAST procedure as usual but each ESL is assigned a subevent to listen in by its allocated Group ID, and response slots are allocated dynamically from the position of commands in the command array of an ESL message.

Ellisys Bluetooth Analyzer Examples

Using an Ellisys Bluetooth Analyzer system with real devices allows us to see PAwR PDUs exchanged by devices in real-time.

A series of screenshots follow and trace through the packets and PDUs involved in the communication of periodic advertising synchronization information and those of the associated PAwR advertising train itself.

Two Nordic Semiconductor nRF54L15 developer boards were used to generate the Bluetooth exchanges. Each was running a sample Zephyr application with one acting as the Broadcaster and the other as an Observer. The Observer is assigned subevent #0 to listen to and within that subevent, response slot #0 to use for AUX_SYNC_SUBEVENT_RSP PDU transmission. The subevent and response slot assignment is made by writing the two parameters as two single byte values to a GATT characteristic using the attribute protocol over an LE-ACL connection. We can see this in the screenshot from the Ellisys Bluetooth Analyzer shown in Figure 11.

Central: "test_ext_adv" D1:00:2D:79:94:DD (Static) <-> Peripheral: "PAwR sync sample" D9:5A:9B:E8:08:C4 (Static)		
ATT Write Transaction (12345678-1234-5678-1234-56789ABCDEF1: 00 00)	OK	2 bytes (00 00)
ATT Write Request Packet (12345678-1234-5678-1234-56789ABCDEF1: 00 00)	OK	2 bytes (00 00)
ATT Write Response Packet	OK	

Figure 11 Application layer using LE-ACL to assign subevent #0 and response slot #0

PAwR synchronization parameters are shown in the AUX_ADV_IND PDU in Figure 12.

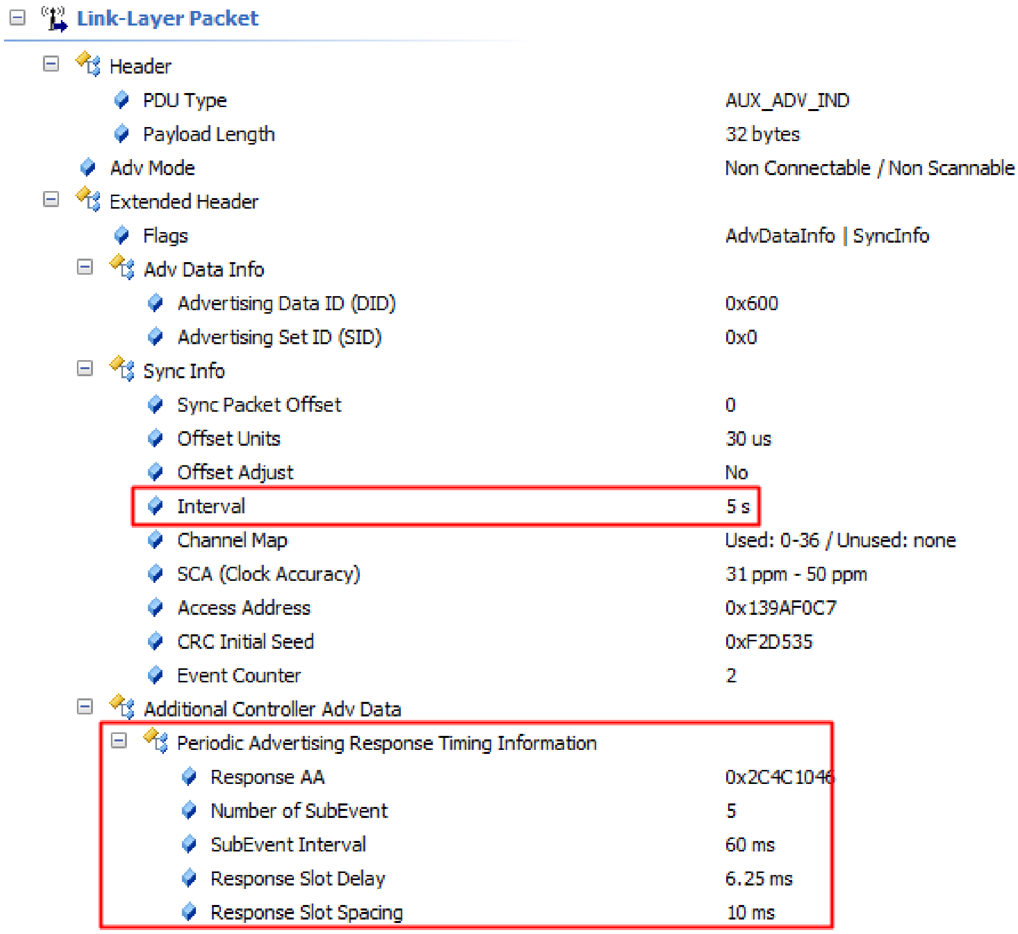

Figure 12 AUX_ADV_IND including SyncInfo and ACAD PA response timing info

Here we can see that the periodic advertising interval is set to 5 seconds. Each periodic event consists of 5 subevents, each spaced at an interval of 60 ms. So, every 5 seconds there should be a burst of 5 subevent transmissions in rapid succession.

And the Ellisys Bluetooth Analyzer confirms this as is shown in Figure 13 .

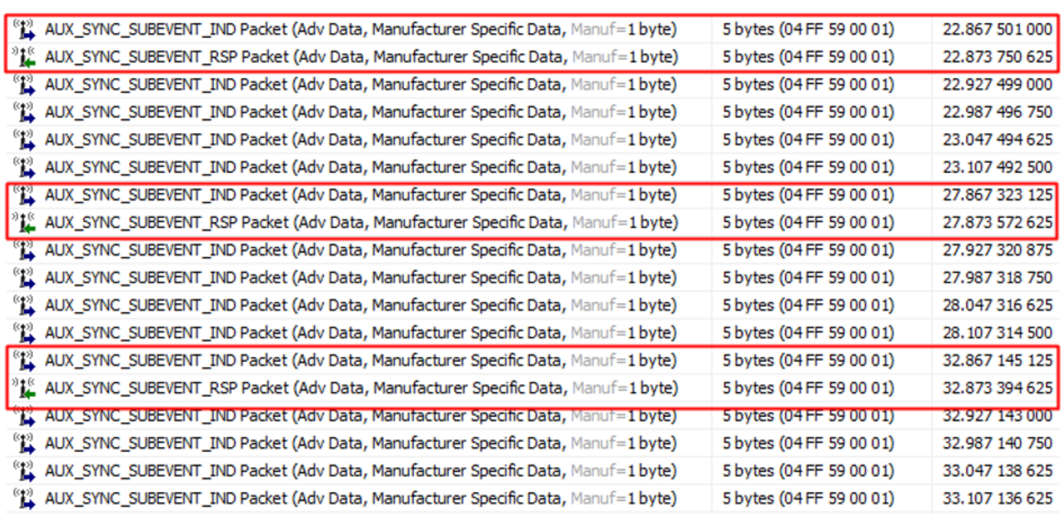

Figure 13 A series of subevent broadcasts and responses

Figure 13 above shows groups of 5 AUX_SYNC_SUBEVENT_IND PDUs transmitted every 5 seconds. This is evidence of the 5 subevents occurring every 5 seconds (the periodic advertising interval).

The test scenario consists of a single Observer only and it's been assigned subevent #0 and response slot #0. Consequently, the first AUX_SYNC_SUBEVENT_IND PDU in each group of five is followed by an AUX_SYNC_SUBEVENT_RSP PDU transmitted by the Observer in the first response slot.

We can also see that the Broadcaster is using the Manufacturer Specific Data AD Type in the AdvData payload field. It contains the 16-bit company ID of Nordic Semiconductor in little endian format (0x5900) followed by a hard coded value of 0x01. The AUX_SYNC_SUBEVENT_RSP PDUs simply echo back the same Manufacturer Specific Data AD Type and its value.

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 with responses.

Property	Comment
Topology	one-to-many (1:m)
Transmitters vs Receivers	All participating devices can both transmit and receive.
Application Data Transmission Direction	Bidirectional. Application data can be transmitted in both directions, from the advertiser to scanners and vice versa.
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. Typically devices are organised into subsets with all devices in each subset receiving packets at the same time.
Radio Channels	The 37 Bluetooth LE general purpose channels are used by PAwR.
Scalability	PAwR allows the Broadcaster to send data to a theoretically unlimited number of devices so in this respect, PAwR is highly scalable. No more than 255 Observer devices can send responses back to the Broadcaster in each subevent. PDU chaining using AUX_CHAIN_IND PDUs is not permitted with PAwR and so the maximum amount of application data that can be transmitted per event is smaller than with periodic advertising without responses.
Choice of PHY	Transmission of AUX_SYNC_SUBEVENT_IND and AUX_SYNC_SUBEVENT_RSP 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 2 PAwR 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 PAwR can be used to send a message to one Observer, a specific subset of Observers or all Observer devices in a single transmission.

How Observers that are synchronized to the same PAwR advertising train at an event level are assigned and synchronized to specific subevents is entirely an application layer concern. This makes for a very flexible system which is able to act as a transport for numerous message-oriented communication scenarios. Note that it's always best to use an applicable profile specification if one exists for your application.

***Did you know that packets in a PAwR 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 PAwR applications to include this field.

Next in the Series

In this article we've explored the periodic advertising with responses (PAwR) communication mode of Bluetooth LE

In the next article in this series, we'll explore isochronous communication.