Everything you need to know about Bluetooth LE advertising

Nordic Tech Webinar

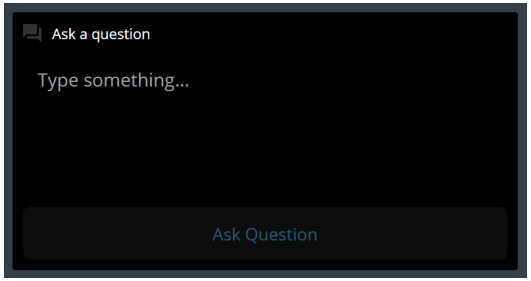
Hung Bui / Senior Application Engineer

Håkon Helsing / Senior Application Engineer

September 2020

Practicalities

- Duration: 50-60 min
- Questions are encouraged!
- Please type questions in the top of the right sidebar
  - All questions are anonymous
  - Try to keep them relevant to the topic
- I will answer questions towards the end
- The chat is not anonymous, and should **not** be used for questions
- If you have more questions, please use DevZone
- A recording of the webinar will be available together with the presentation at webinars.nordicsemi.com



{ DevZone

Today's hosts

Hung Bui

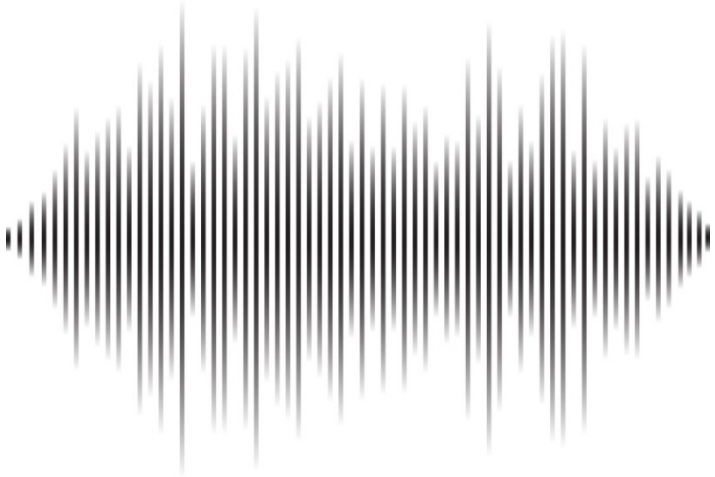
Senior Application Engineer

Håkon Helsing

Senior Application Engineer

Agenda

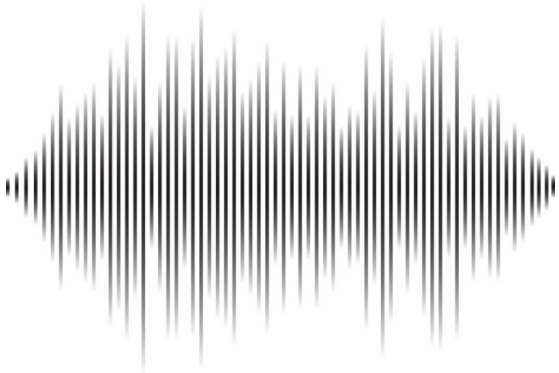
- Basics
- Advertising Extensions
- Advertising types
- Advertising data format
- nRF Connect SDK API and example walk through
- Advertisiment analysis with nRF Sniffer for Bluetooth LE



Basics

What is advertising?

- A Broadcaster advertises to broadcast data
  - Data
  - RSSI
  - Direction Finding I/Q data
- A Peripheral advertises to broadcast data and/or to make itself discoverable for a connection
- Two types of advertising:
  - Legacy advertising
  - Extended advertising



GAP roles and Link Layer states

GAP role	Link Layer state
Broadcaster	Advertising
Observer	Scanning
Peripheral	Advertising Connection (Slave)
Central	Scanning Initiating Connection (Master)

All roles can also be in the standby state

Bluetooth Core Specification 5.2, Vol 3, Part C, 2.2.2

Roles (GAP)

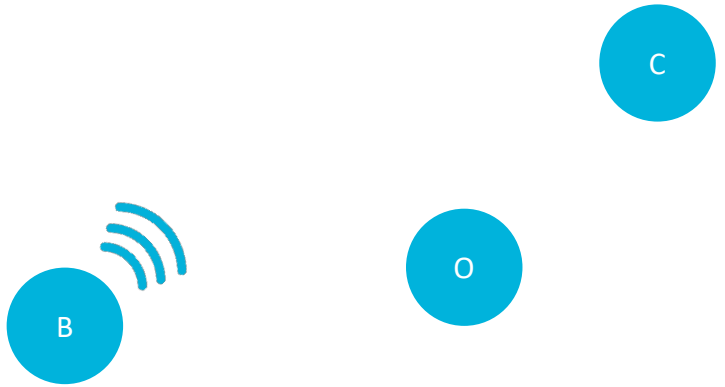
B Broadcaster

O Observer

P Peripheral

C Central

Roles (GAP)



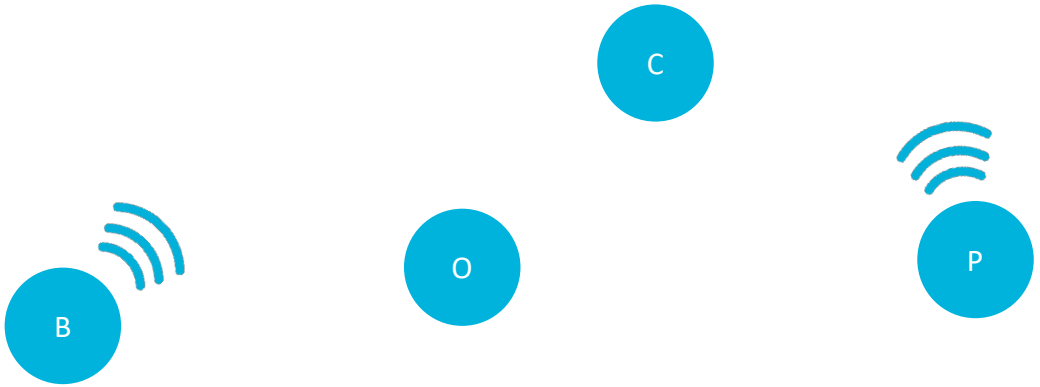
B Broadcaster

O Observer

P Peripheral

C Central

Roles (GAP)



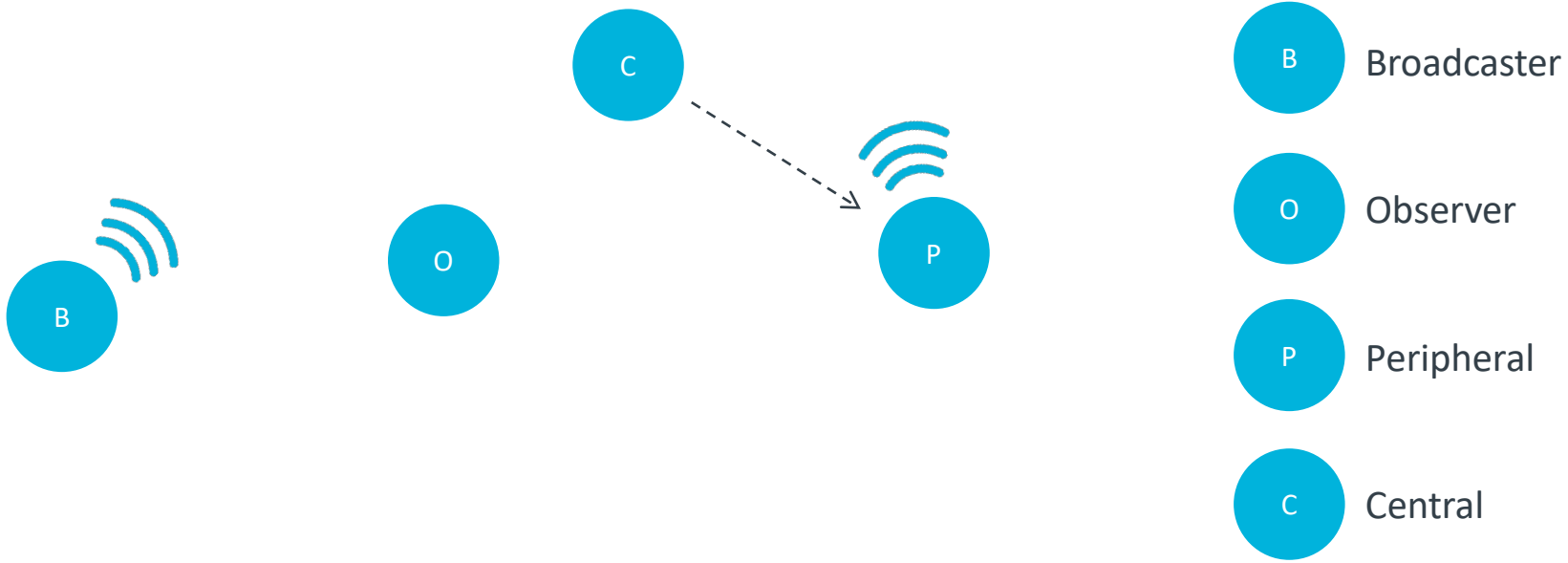
B Broadcaster

O Observer

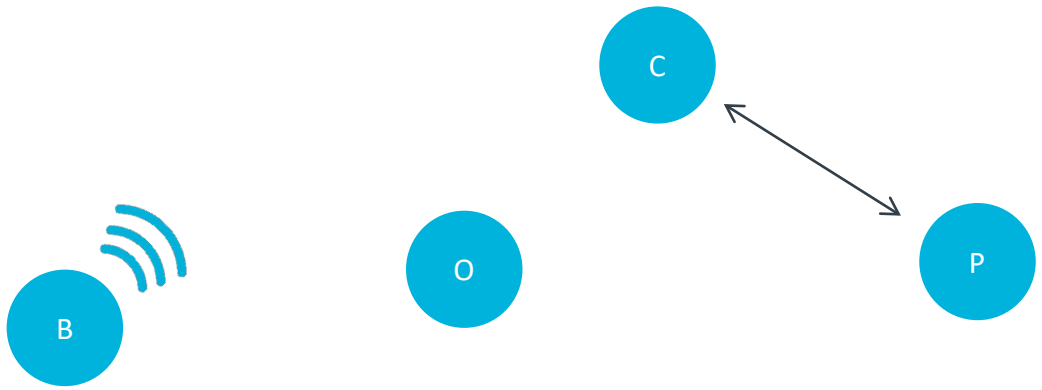
P Peripheral

C Central

Roles (GAP)



Roles (GAP)



B Broadcaster

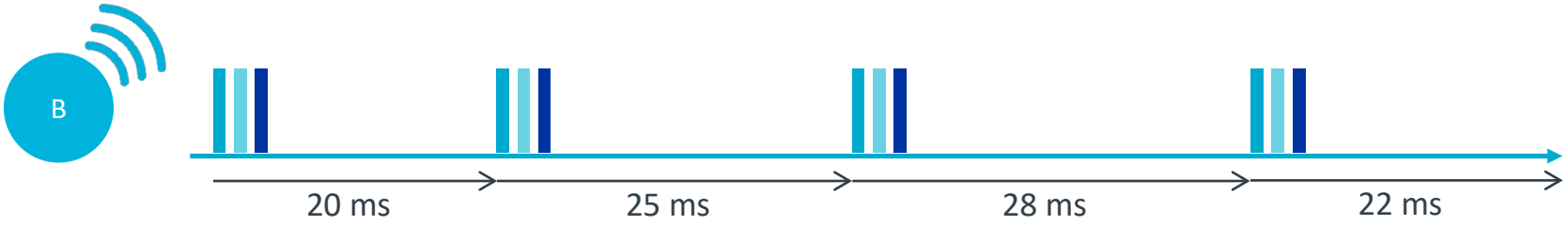
O Observer

P Peripheral

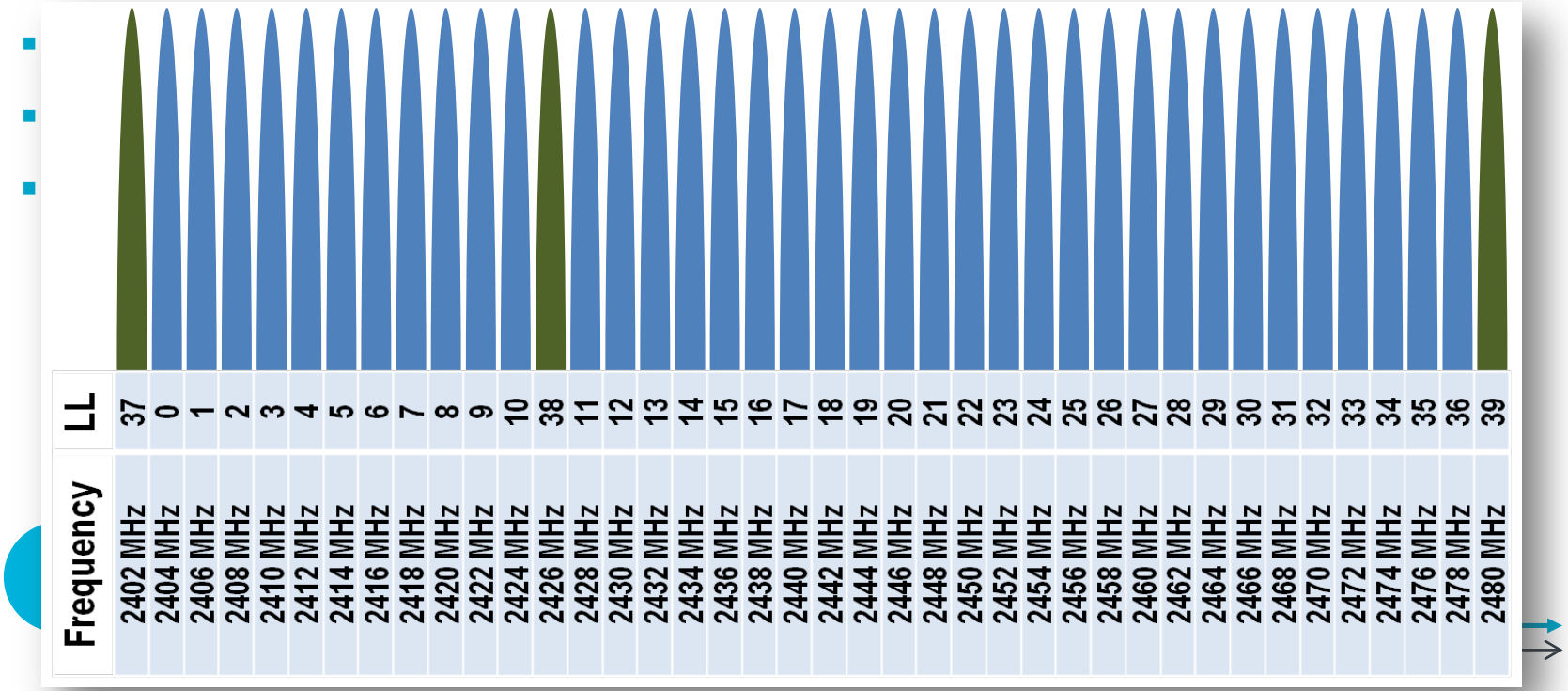
C Central

Advertising and scanning

- Advertising interval = 20 ms
- 0-10 ms random delay
- Advertising on channel

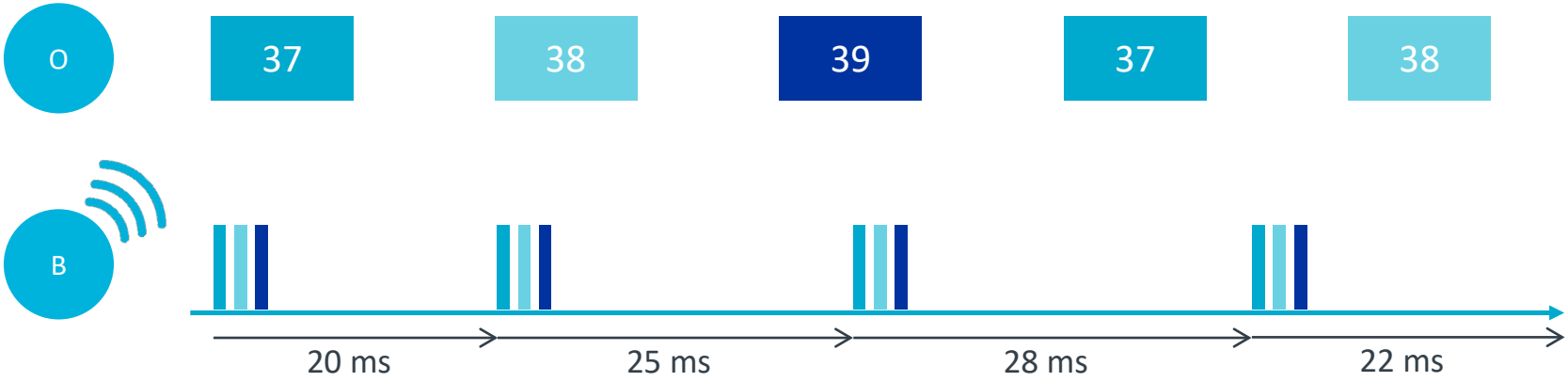


Advertising and scanning



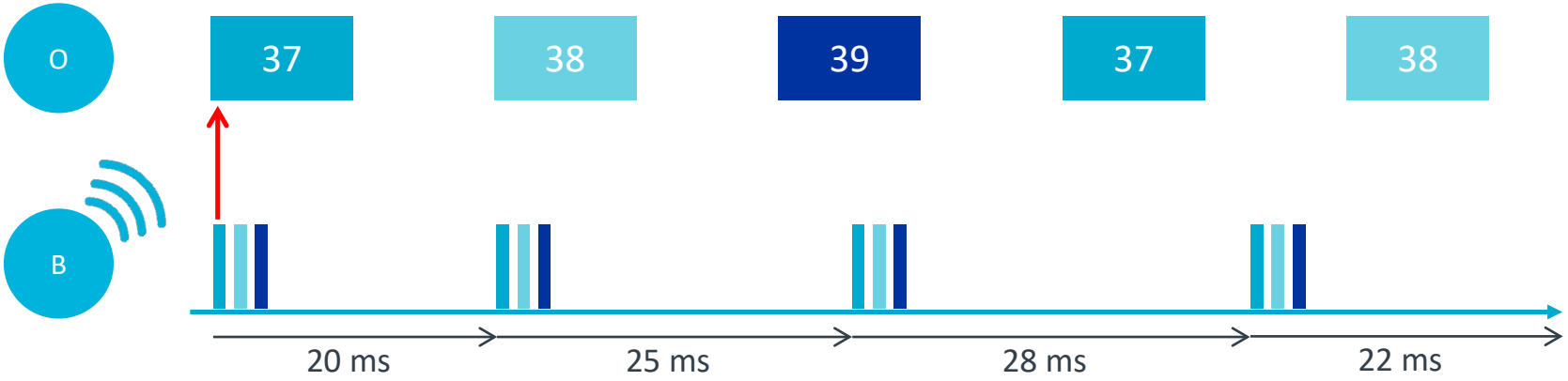
Advertising and scanning

- Scan interval = 20 ms
- Scan window = 10 ms



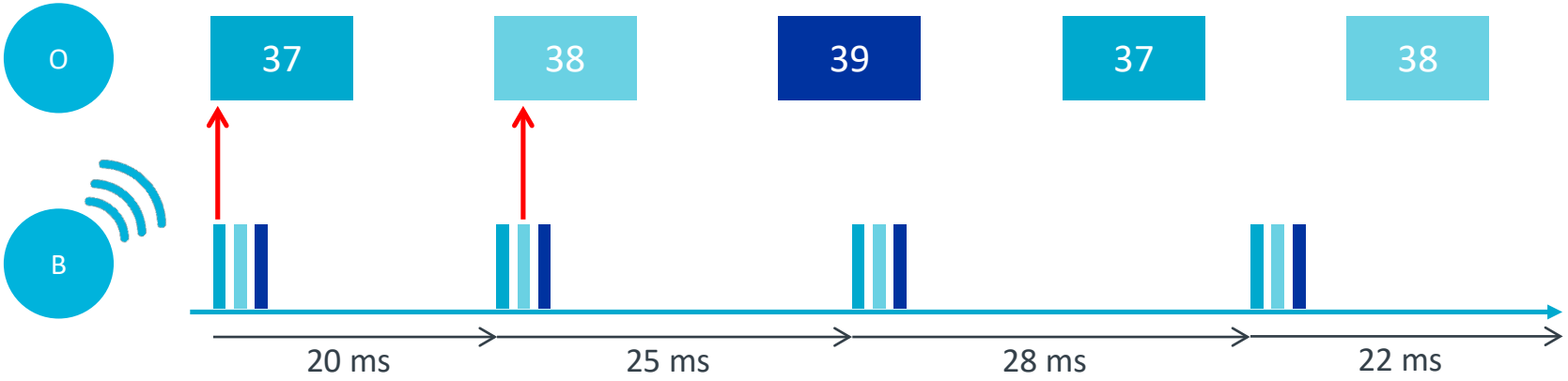
Advertising and scanning

- Scan interval = 20 ms
- Scan window = 10 ms



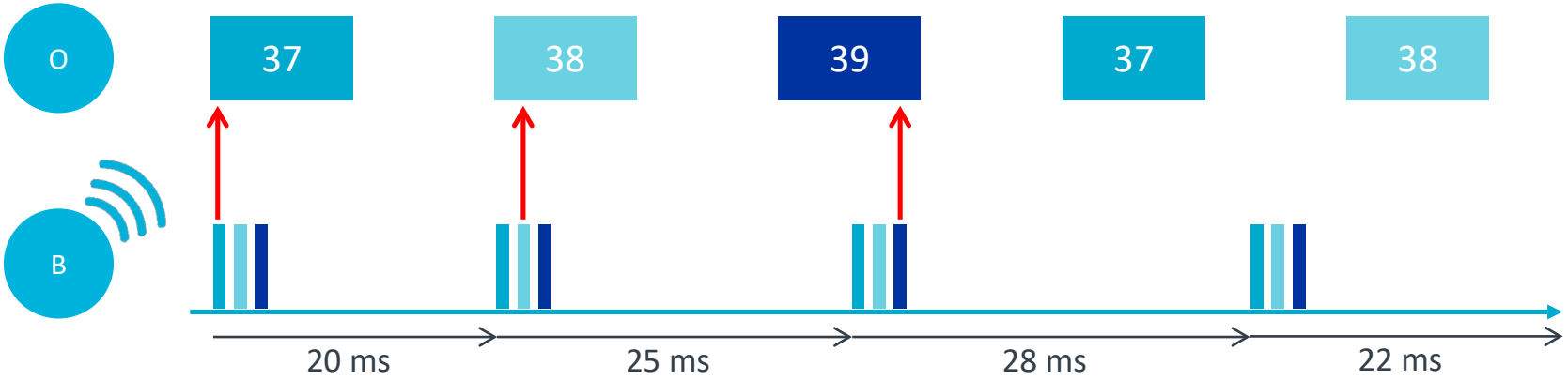
Advertising and scanning

- Scan interval = 20 ms
- Scan window = 10 ms



Advertising and scanning

- Scan interval = 20 ms
- Scan window = 10 ms



Advertising types

Legacy advertising types

Type	Scannable	Connectable
ADV_IND	Yes	Yes
ADV_SCAN_IND	Yes	
ADV_NONCONN_IND		
ADV_DIRECT_IND		Yes

Bluetooth Core Specification 5.2, Vol 6, Part B, 2.3.1

Legacy advertising types

Type	Scannable	Connectable
ADV_IND	Yes	Yes
ADV_SCAN_IND	Yes	
ADV_NONCONN_IND		
ADV_DIRECT_IND		Yes

- Extended advertising types
  - ADV_EXT_IND
  - AUX_ADV_IND
  - AUX_SYNC_IND
  - AUX_CHAIN_IND

Bluetooth Core Specification 5.2, Vol 6, Part B, 2.3.1

GAP roles and advertising types

GAP role	Advertising types
Broadcaster	ADV_SCAN_IND ADV_NONCONN_IND
Peripheral	ADV_IND ADV_SCAN_IND ADV_NONCONN_IND ADV_DIRECT_IND

Bluetooth Core Specification 5.2, Vol 3, Part C, 2.2.2

Scan request and scan response

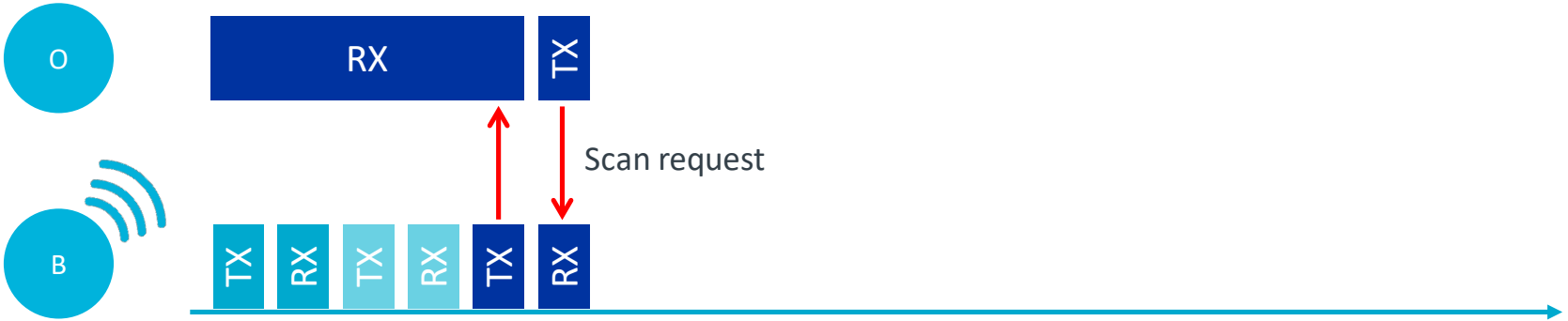
- ADV_SCAN_IND and ADV_IND are scannable

Scan request and scan response

- ADV_SCAN_IND and ADV_IND are scannable

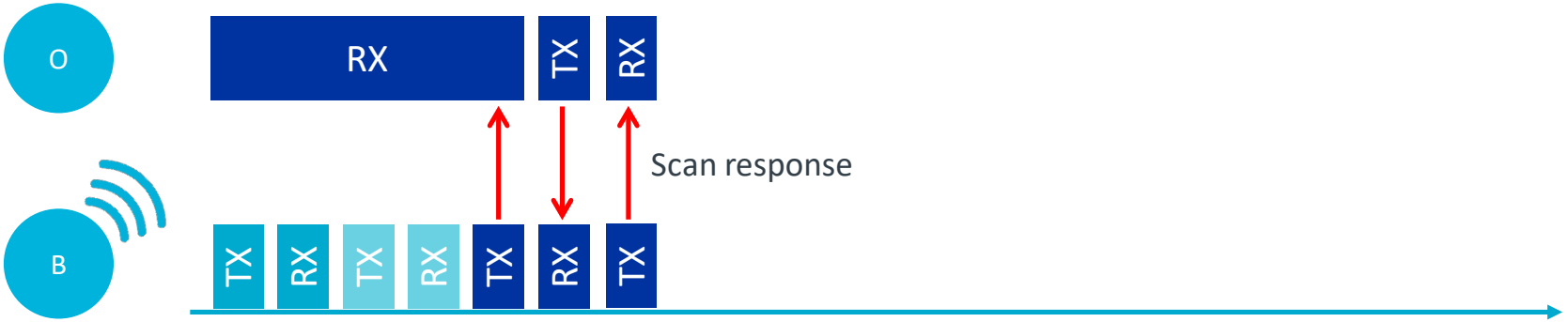
Scan request and scan response

- ADV_SCAN_IND and ADV_IND are scannable



Scan request and scan response

- ADV_SCAN_IND and ADV_IND are scannable

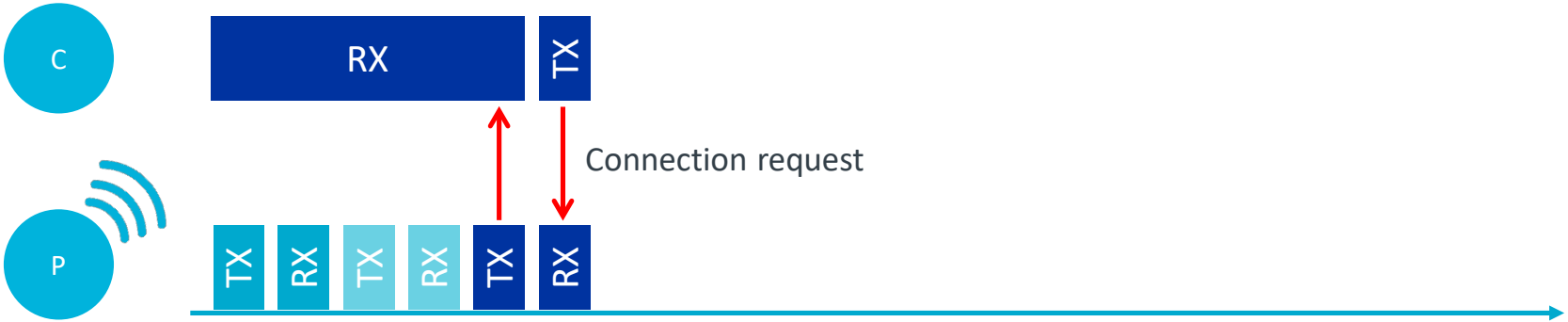


Connection request

- ADV_IND and ADV_DIRECT_IND are connectable

Connection request

- ADV_IND and ADV_DIRECT_IND are connectable



Non-connectable advertising

- ADV_NONCONN_IND
- Not connectable or scannable
- No RX -> Lower power consumption

Advertising interval

- ADV_NONCONN_IND, ADV_SCAN_IND, ADV_IND
  - 20 ms to 10.24 seconds, in steps of 0.625 ms
  - 0-10 ms random delay
- Trade-off between power consumption and device discovery time
- Scan interval and scan window
- Leverage scanner if possible

Bluetooth Core Specification 5.2, Vol 6, Part B, 4.4.2.2.1

Advertising interval - ADV_DIRECT_IND

- Directed advertising
- Used to quickly reconnect to a known Central
- Two duty cycle options:
  - Low duty cycle
    - The time between two advertisements shall be less or equal to 10 ms
  - High duty cycle
    - The time between two advertisements shall be less or equal to 3.75 ms
    - Maximum duration of 1.28 s

Bluetooth Core Specification 5.2, Vol 6, Part B, 4.4.2.4

White list

- An advertiser can use a white list to filter out unwanted scan requests and/or connection requests
- The white list contains device addresses and/or Identity Resolving Keys (IRKs) that the advertiser shall accept scan requests and/or connection requests from, filtering away all other.
  - These typically belong to devices it has bonded with
  - IRKs are shared after bonding, and are used to resolve private resolveable addresses

Bluetooth Core Specification 5.2, Vol 6, Part B, 4.3


Advertising data

Advertising PDU

- Advertising PDU:

Header 16 bits	Payload (1-255 octets)

- Advertising PDU header:

PDU type 4 bits	RFU 1 bit	ChSel 1 bit	TXAdd 1 bit	RXAdd 1 bit	Length 8 bits

Advertising PDU header

- Advertising PDU Header:

PDU type 4 bits	RFU 1 bit	ChSel 1 bit	TXAdd 1 bit	RXAdd 1 bit	Length 8 bits

PDU type	PDU name
0b0000	ADV_IND
0b0001	ADV_DIRECT_IND
0b0010	ADV_NONCONN_IND
0b0110	ADV_SCAN_IND

Advertising PDU header

- Advertising PDU header:

PDU type 4 bits	RFU 1 bit	ChSel 1 bit	TXAdd 1 bit	RXAdd 1 bit	Length 8 bits

- RFU: Reserved for future use
- ChSel: 1 if LE Channel Selection Algorithm #2 is supported
- TXAdd: 0 if address is public, 1 if random
- RXAdd: 0 if the target's address is public, 1 if random
- Length: Payload length

Advertising PDU payload

- Advertising PDU:

Header 16 bits	Payload (1-255 octets)

ADV_IND payload	
AdvA (6 octets)	AdvData (0-31 octets)

ADV_NONCONN_IND payload	
AdvA (6 octets)	AdvData (0-31 octets)

ADV_DIRECT_IND payload	
AdvA (6 octets)	TargetA (6 octets)

ADV_SCAN_IND payload	
AdvA (6 octets)	AdvData (0-31 octets)

AdvA – Advertiser address

- Public (TXAdd = 0)
  - Unique address that can be recognized
- Random (TXAdd = 1)
  - Static
  - Private
    - Non-resolvable
    - Resolvable

Advertising data format

- Advertising PDU:

Header 16 bits	Payload (1-255 octets)

ADV_IND payload	
AdvA (6 octets)	AdvData (0-31 octets)

ADV_NONCONN_IND payload	
AdvA (6 octets)	AdvData (0-31 octets)

ADV_DIRECT_IND payload	
AdvA (6 octets)	TargetA (6 octets)

ADV_SCAN_IND payload	
AdvA (6 octets)	AdvData (0-31 octets)

Advertising data format

AD structure 1	AD structure 2	…	AD structure N

Length 1 octet	AD type n octets	AD data Length – n octets

- With ADV_IND, ADV_NONCONN_IND and ADV_SCAN_IND the sum of the structures can be maximum 31 bytes, **with overhead**

Advertising data format – AD type

- AD type data format is defined in Part A in Bluetooth Core Specification Supplement (CSS)
- Most common AD types:
  - Service UUID
  - Local Name
  - Flags
  - Manufacturer Specific Data

Advertising data format – AD type

- Service UUID
  - Typically included in connectable advertisements so that Centrals know supported service(s) without connecting. For example should a Heart Rate Sensor include the Heart Rate Service UUID
- Local Name
  - Typically included in connectable advertisements so that a human user can select to connect. For example «Heart rate sensor».
  - Can be shortened (0x08) or complete (0x09)

Advertising data format – AD type

- Flags (0x01)
  - Shall be included when any of the Flag bits are non-zero and the advertising packet is connectable (ADV_IND)
  - 5 bits:
    - Limited Discoverable Mode
    - General Discoverable Mode
    - BR/EDR Not Supported
    - Simultaneous LE and BR/EDR to Same Device Capable (Controller)
    - Simultaneous LE and BR/EDR to Same Device Capable (Host)

Advertising data format – AD type

- Manufacturer Specific Data (0xFF)
  - Typically used to include custom data
  - The two first octets shall contain a company identifier from the company identifier assigned numbers (free to obtain for Bluetooth SIG members)

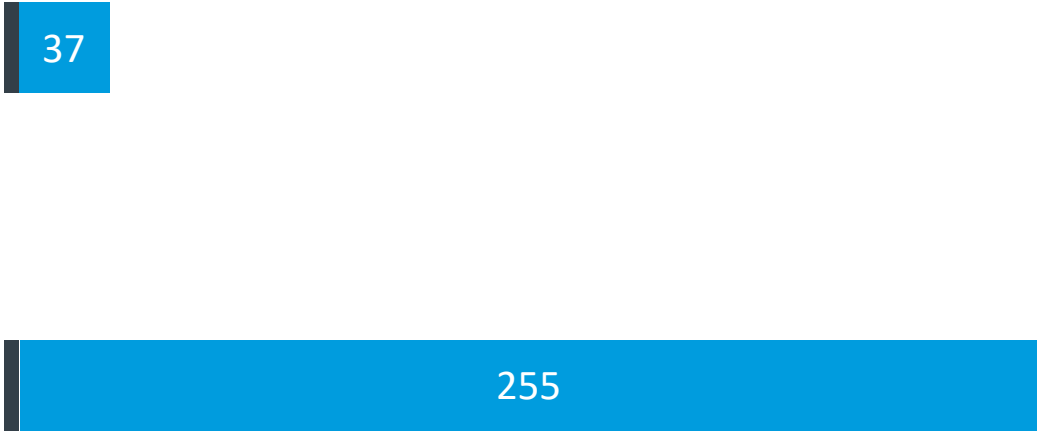
Advertising Extensions

Why use Advertising Extensions?

- Increases advertising data length
- Allows advertising on data channels
- Enables long range connection establishment
- Chaining
- Periodic advertising

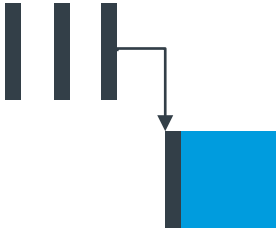
Increases Advertising data length

- Legacy Advertising
  - 2 bytes header
  - 37 bytes payload
  - 31 bytes advertising data
- Advertising Extensions
  - 2 bytes header
  - 255 bytes payload
  - 254 bytes advertising data
  - 0-63 extended header



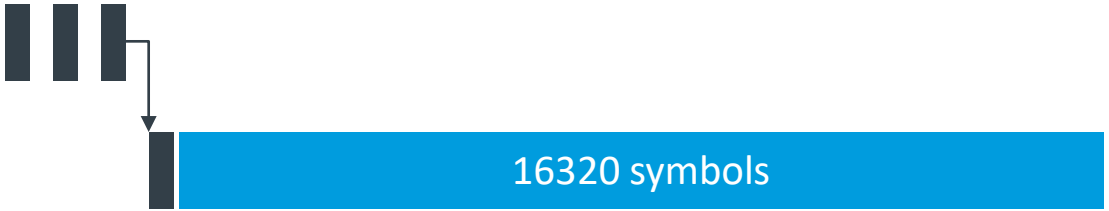
Advertising on data channels

- Legacy Advertising
  - Complete payload repeated on the advertising channels
- Advertising extensions
  - Header is repeated on the advertising channels
  - Payload is only transmitted once



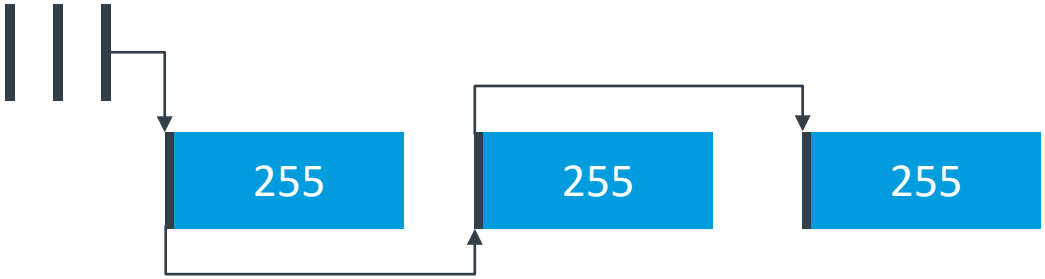
Advertising on data channels

- Longer packets and coding
  - Congested advertising channels
- Reduces contention and duty cycle



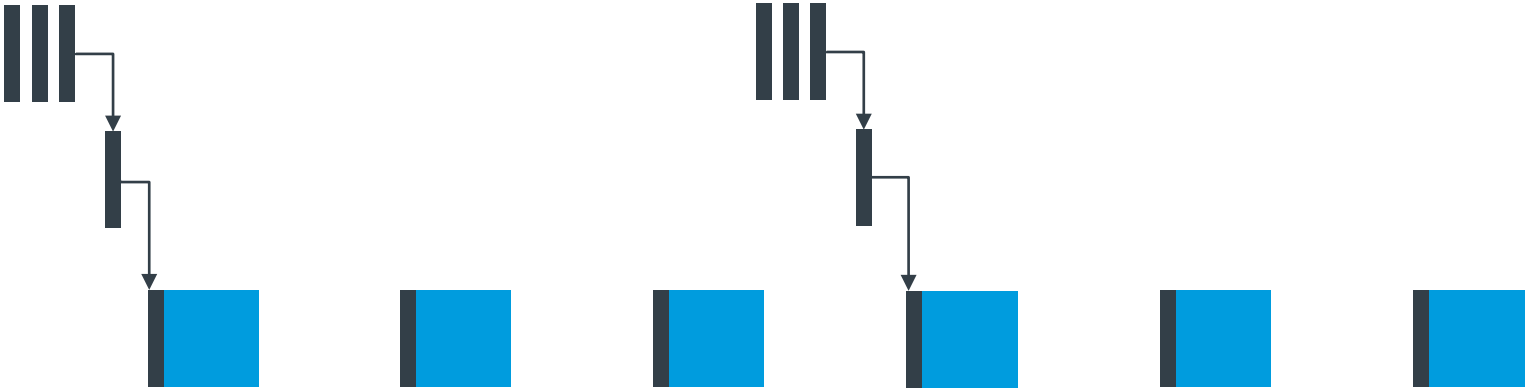
Chaining

- Advertisements can be chained together to extend the amount of advertising data



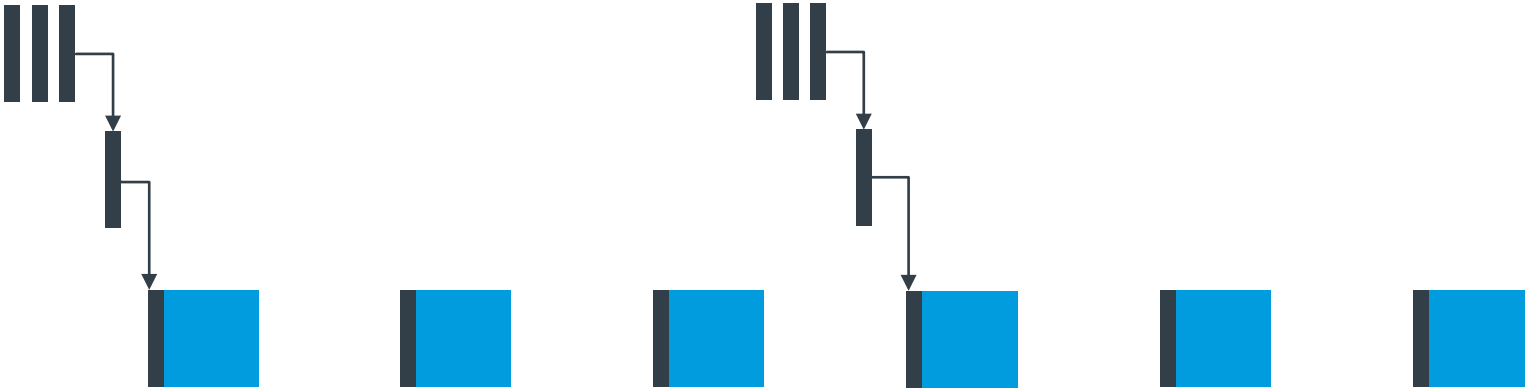
Periodic Advertising

- Enables synchronized broadcasting of advertising data
- Happens at a deterministic interval allowing true connectionless broadcasting



Periodic Advertising

- Connectionless Direction Finding

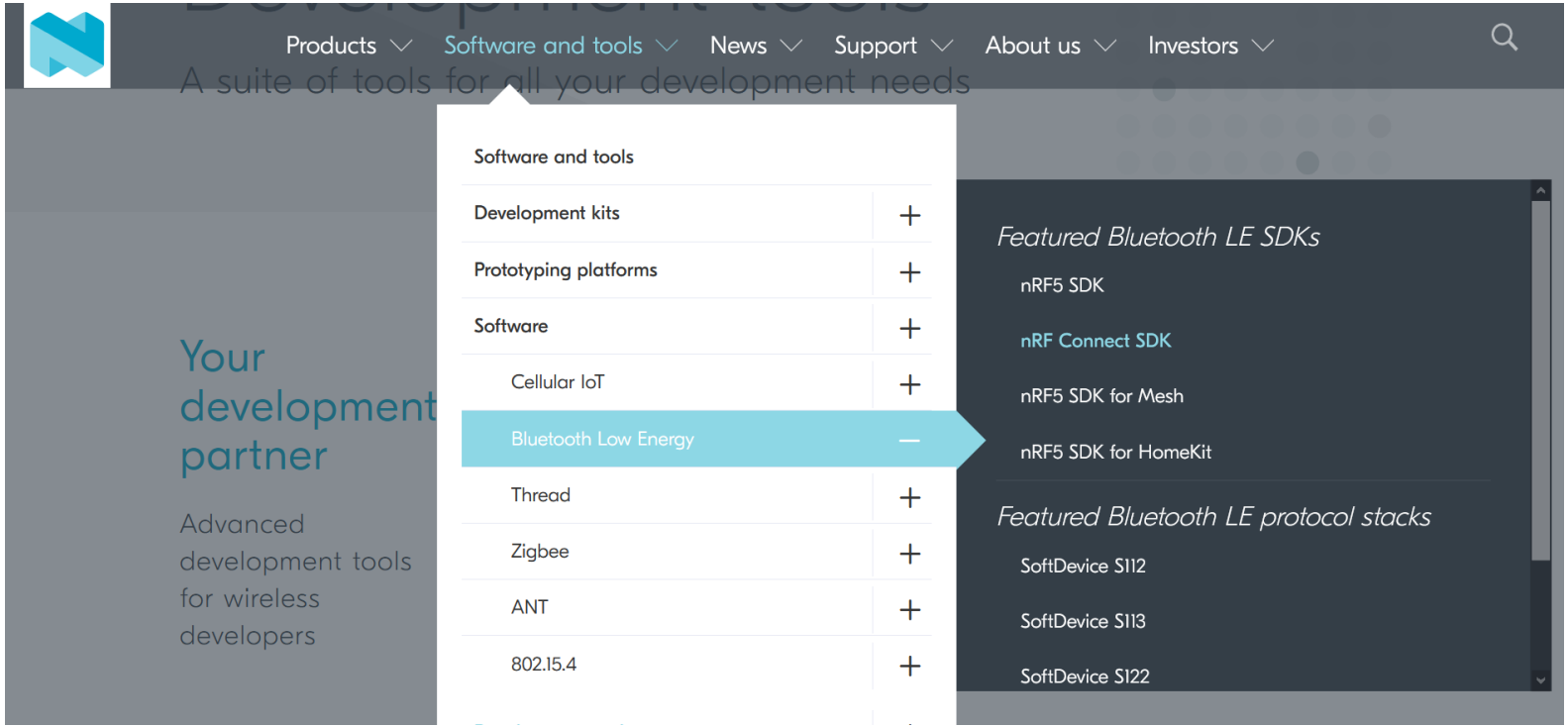


nRF Connect SDK

API, example walkthrough and demo

Demo setup

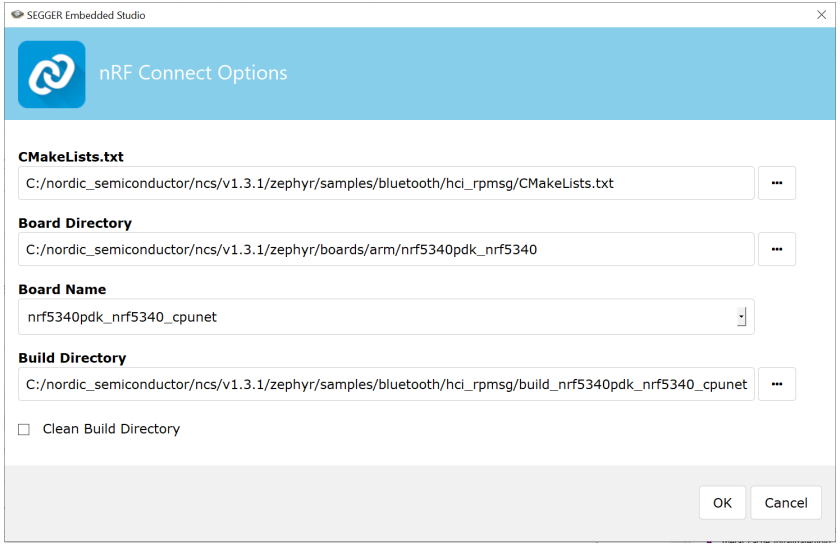
nRF Connect SDK



SEGGER Embedded Studio

Preparing the network core of nRF5340

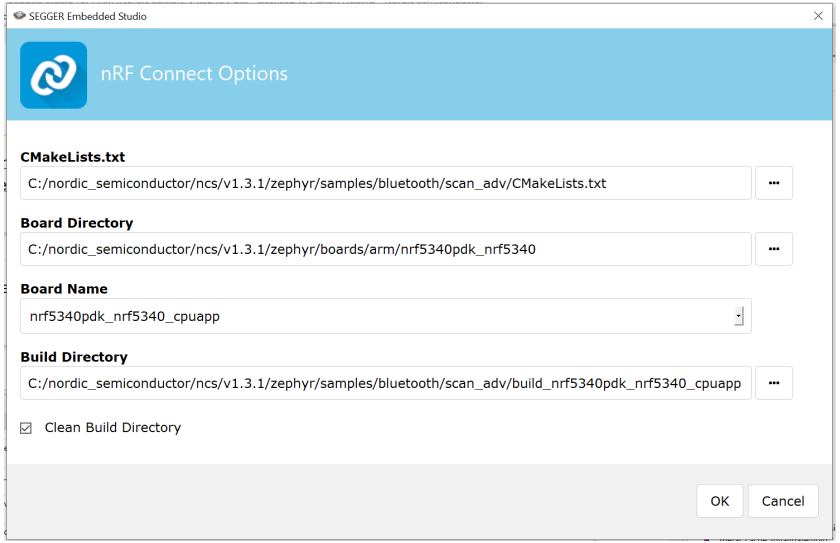
- nRF 52 series devices may skip this step
- Prepare nRF5340 PDK with the *hci_rpmsg* sample
- Select the Network core as shown
- Build and run the sample



SEGGER Embedded Studio

Project setup

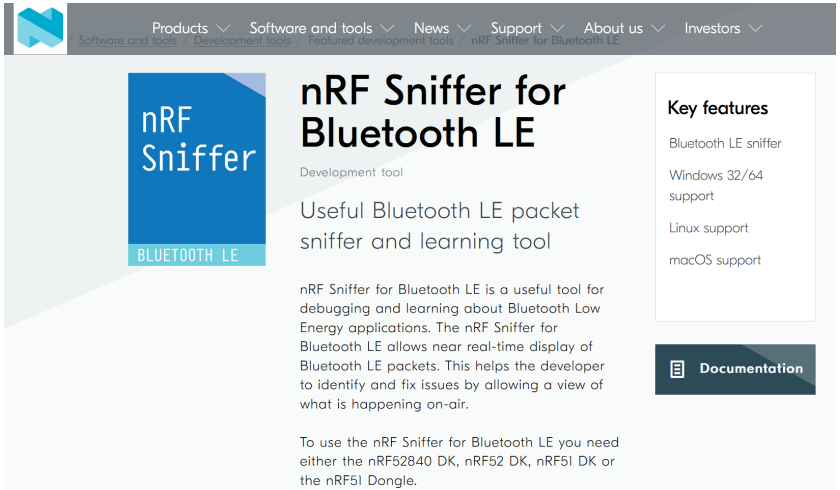
- Locate the *scan_adv* sample
- Select your board directory
- Select the Application core
- Click ‘Build and run’ to flash the sample



Demo

nRF Sniffer for Bluetooth LE

- Find the downloadvand instructions at nordicsemi.com



Demo

Examples

- Example 1: scan_adv sample, unmodified from Zephyr Project
- Example 2: Static address, name, Company ID
- Example 3: Advertising interval, set address
- Example 4: Scan response, send more data(128-bit UUID)

Example 1

scan_adv

Example 1 - scan_adv sample

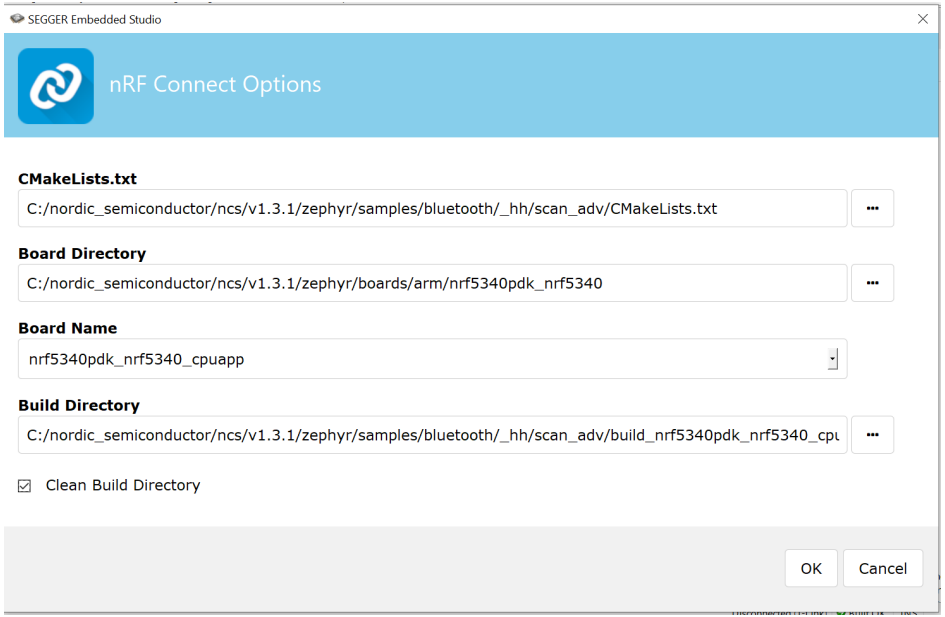
- [nRF Sniffer for Bluetooth LE window]

Example 2

Static address, name, Company ID

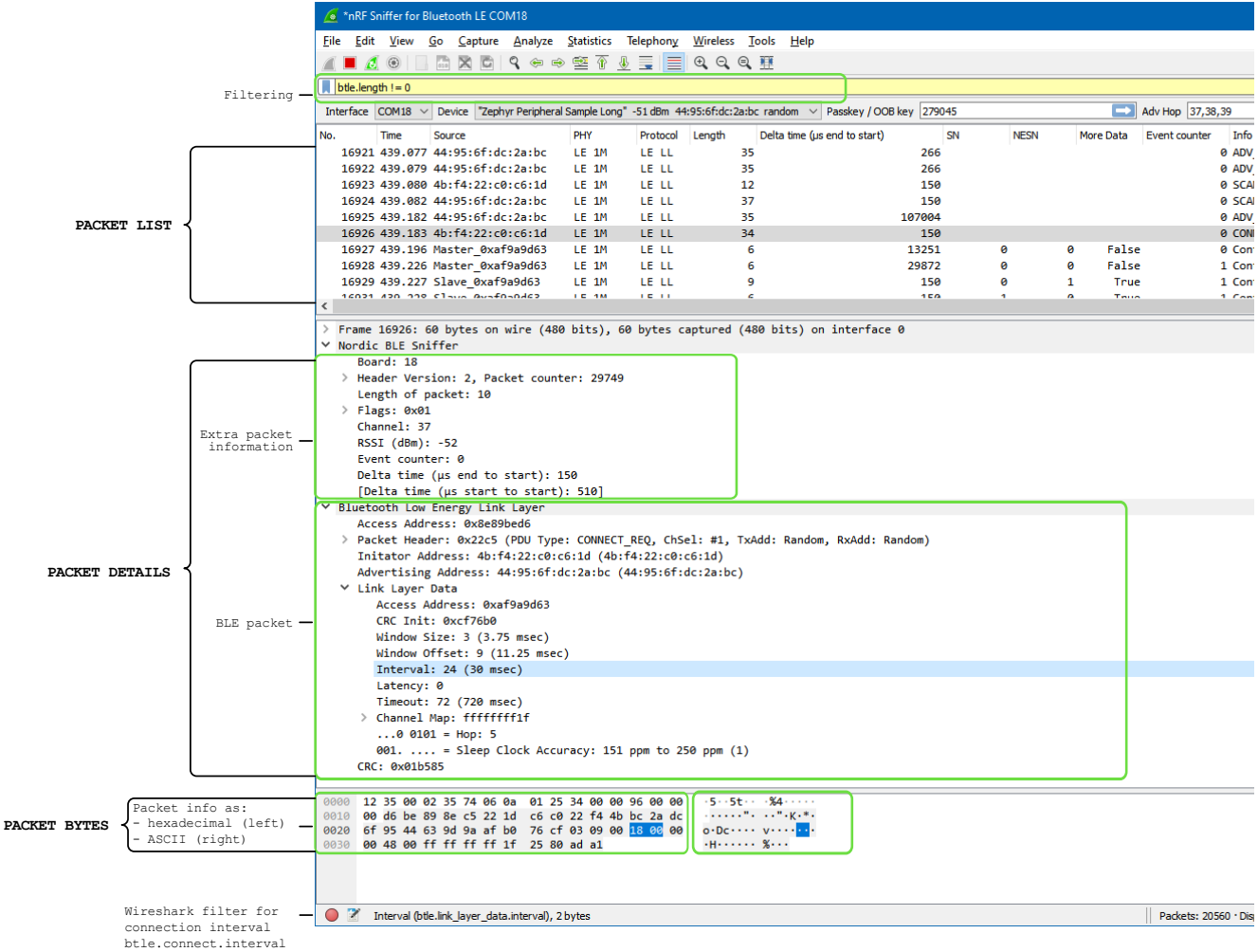
Example 2

- [SES-window]



Example 2

- [Wireshark window]



Example 3

Advertising interval and set address

Example 3

- [SES-window]

Example 3

- [Wireshark window]

Example 4

Scan response, send more data(128-bit UUID)

Example 4

- [SES-window]

Example 4

- [Wireshark window]

Q&A