# DT0107
# Design tip

## Slot allocation and multiple connection timing strategy for BlueNRG, BlueNRG-MS, BlueNRG-1 and BlueNRG-2

Andrea Vitali

| Main components | |
|---|---|
| BlueNRG-MS | Bluetooth Low Energy Network Processor supporting Bluetooth 4.1 core specification |
| BlueNRG-1 | Bluetooth Low Energy System On Chip |
| BlueNRG-2 | Bluetooth Low Energy Wireless System-On-Chip |

## Purpose and benefits

This design tip explains the best strategy to allocate different types of slots (advertising, scanning and multiple connections slots) when using Bluetooth® low energy network processors, such as BlueNRG and BlueNRG-MS, or Bluetooth® low energy system-on-chip, such as BlueNRG-1 and BlueNRG-2.

Benefits:

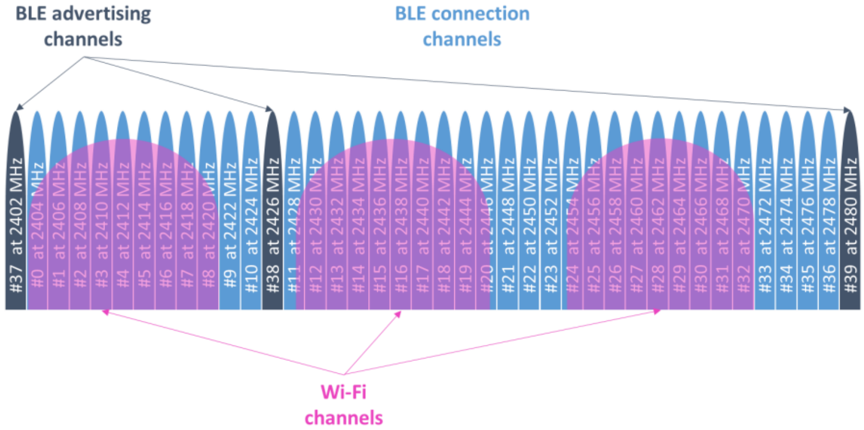
- Learn how slot allocation works, in order to have slot of different types (advertising, scanning and multiple connection slots) coexist successfully.
- Learn how to select optimal interval and length for connection slots, to maximize throughput and minimize latency, especially when there are multiple connections with different timing requirements.

## Physical (PHY) and Link Layer (LL)

The Physical Layer (PHY) is a 1 Mbps adaptive frequency-hopping Gaussian Frequency Shift Keying (GFSK) radio. It operates in the license free 2.4 GHz ISM band at 2400-2483.5 MHz. There are 40 radio frequency (RF) channels, with 2 MHz spacing, centered at 2402 + n * 2 MHz, where n goes from 0 to 39.

Three RF channel (#37 at 2402 MHz, #38 at 2426 MHz and #39 at 2480 MHz) are used for “advertising”; the other 37 channels (from #0 to #36) are used for bi-directional data “connections”. Advertising channels are carefully positioned to avoid interference with Wi-Fi channels. See Figure 1.

**Figure 1. BLE advertising and connection channels**




The Link Layer (LL) defines how devices can use the radio to transmit or receive data. There are several states (see Figure 2):

- Stand-by: no transmission or reception
- Advertising: transmission of advertising packets in advertising channels; this is used by devices whose role is “broadcaster” or “peripheral”.
- Scanning: reception of advertising packets; this is used by devices whose role is “observer” or “central”.
- Connection: transmission and reception; this is used by devices whose role is “central” or “peripheral”.

Typically, “peripheral” devices advertise their existence to a scanning “central” device; after discovery, the central device is able to connect to peripheral devices. **The central is the master** because it can initiate the connection and controls the timing of the communication, on the opposite peripherals are slaves.

**Table 1. Link Layer (LL) states and corresponding slot type**

| Slot type | LL state | GAP roles | Notes |
|---|---|---|---|
| ADV | Advertising | Peripheral (slave)<br>Broadcaster | In ADV slots, the device broadcasts advertisements.<br>• ADV_NONCONN_IND if not-connectable and not scannable<br>• ADV_SCAN_IND if not-connectable but **scannable**<br>• ADV_IND if **connectable** and **scannable**<br>• ADV_DIRECT_IND if **connectable** but not scannable |
| SCAN | Scanning | Central (master)<br>Observer | In SCAN slots, the device looks for advertisements.<br>• Passive scan: does not send additional SCAN_REQuests.<br>• Active scan: sends **SCAN_REQ to scannable devices**. |
| CONN | Connection | Central (master)<br>Peripheral (slave) | In CONN slots, the device exchange data with other devices.<br>• The device (Master) initiate connections with **CONNECT_REQ to connectable devices** (Slaves), and defines the timing of CONN slots.<br>• In a CONN slot: Master sends data to Slave (M>S), then Slave sends data to Master (S>M), and the cycle repeats until the connection event is over. |

# Generic Access Profile (GAP)

The Generic Access Profile (GAP) defines the basic requirements of a BLE device. There are four GAP roles (see Figure 3):

- Broadcaster: transmit advertising packets. Optionally it can also receive.
- Observer: receives advertising packets. Optionally it can also transmit.
- Peripheral: transmit advertising packets to advertise its existence; during connections, transmit and receives data from one Central device (master).
- Central: receives advertising packets to discover devices (scanning); during connections, transmit and receives to one or more Peripheral devices (slaves).

When advertising, the device can specify if it is **connectable** (can accept CONN_REQests and establish a connection) and if it is **scannable** (can accept SCAN_REQuests and reply with SCAN_RESponse). There are four possible combinations (see Figure 2).

Usually a broadcaster is not-connectable (ADV_NONCONN_IND or ADV_SCAN_IND), while a peripheral is connectable (ADV_IND or ADV_DIRECT_IND).

**Table 2. Generic Access Profile (GAP) roles and corresponding Link Layer (LL) state**

| GAP roles | LL state | Notes |
|---|---|---|
| Broadcaster | Advertising | Broadcaster broadcasts advertisements. Does not accept CONN_REQuests.<br>• ADV_NONCONN_IND not-connectable and not-scannable<br>• ADV_SCAN_IND not-connectable but **scannable**: accepts and answers **SCAN_REQuests** |
| Observer | Scanning | Observer looks for advertisements.<br>• Passive scan: does not send SCAN_REQ<br>• Active scan: sends **SCAN_REQests** to **scannable** devices. |
| Central | Scanning<br>Connection | Central scan for Peripherals, then sends **CONN_REQests** to **connectable** devices.<br>• Passive scan: does not send SCAN_REQ<br>• Active scan: sends **SCAN_REQests** to **scannable** devices. |
| Peripheral | Advertising<br>Connection | Peripheral advertise its presence to Central, then it accepts **CONN_REQuests**.<br>• ADV_IND **connectable and scannable**: accepts and answers **SCAN_REQ**<br>• ADV_DIRECT_IND **connectable** but not scannable |

# Slot allocation APIs

In general, one must specify the slot interval (periodicity) and the slot length (duration). See Figure 4 for valid ranges and units.

## Advertising slot (ADV)

The advertising slot (ADV) is allocated by calling the function **aci_gap_set_discoverable()** specifying its type, min and max interval (periodicity). If min and max interval are both zero, a default value is used. The length is fixed at 14.6 ms and includes a random delay (max 10 ms) which is added before the event to reduce collisions with other advertisers. The average delay, 5 ms, is added to min and max intervals specified in the call to the function.

**ADV_DIRECT_IND (connectable but not scannable) with high duty cycle is special**: interval parameters are ignored; the interval is set to 3.75 ms and the length is set to 2.25 ms. There is no random delay. Advertising automatically stops after 1.28 s (timeout).

**Figure 2. Valid range for slot interval and length**

| Slot type | Interval range | Interval units | Length range | Length Units | Notes |
|---|---|---|---|---|---|
| ADV_NONCONN_IND (not-connectable and not-scannable) | 100 ms to 10.24 s<br>Default min = 100 ms<br>Default max = 150 ms | 0.625 ms | **14.6 ms**<br>fixed by FW<br><br>Length + 1.5 ms<br><= Interval | 0.625 ms | Best allocated first, before SCAN slot (because SCAN interval and length may be ignored).<br><br>Best if interval is long and multiple of 10 ms (see notes below for CONN slots).<br><br>The fixed length includes a **random delay (0 to 10 ms) which is added** before the event to reduce collisions with other advertisers.<br><br>The **intervals specified are increased by 5 ms**, the average delay. |
| ADV_SCAN_IND (not-connectable but scannable) | | | | | |
| ADV_IND (connectable and scannable) | 20 ms to 10.24 s<br>Default min = 30 ms<br>Default max = 60 ms | | | | |
| ADV_DIRECT_IND Low duty cycle (connectable but not-scannable) | | | | | |
| **ADV_DIRECT_IND High duty cycle** (connectable but not-scannable) | **3.75 ms**<br>fixed by FW | n/a | **2.25 ms**<br>fixed by FW<br><br>Length + 1.5 ms<br>== Interval | n/a | **Automatic stop at timeout: 1.28 s** |
| SCAN | *May be ignored, only advisory*<br><br>2.5 ms to 10.24 s | 0.625 ms | *May be ignored, only advisory*<br><br>2.5 ms to 10.24 s<br><br>Length + 1.5 ms<br><= Interval | 0.625 ms | Best allocated after ADV slot, because **SCAN interval and length may be ignored**.<br><br>Best setting is to use same interval as for connections and the largest possible length.<br><br>It is replaced by CONN slot upon connection. |
| CONN | 7.5 ms to 4 s | **1.25 ms** | **3.75 ms** to 4 s<br><br>**4.5 ms** to 4 s when Data Length Extension is supported<br><br>Length + 1.5 ms<br><= Interval | 0.625 ms | To leave space for other connections, use long interval and short length.<br><br>To accommodate connections with shorter periodicity, best interval is multiple of 10 ms as it is still multiple of 1.25 ms when divided by 2, 4, 8.<br><br>**Latency**: 0 to 500. Actual interval = anchor period * (latency + 1).<br><br>**Supervision timeout**: 100 ms to 32 s (10 ms units). It must be larger than 2*Interval*(N+1), where N is the Latency. |

## Scanning slot (SCAN)

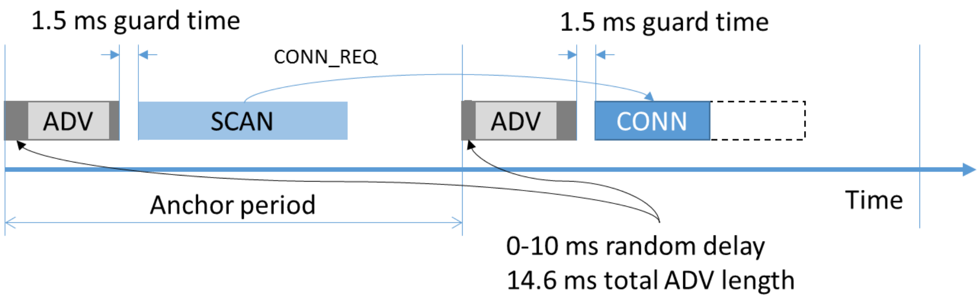
The scanning slot (SCAN) is allocated by calling the **aci_gap_start_gen_disc_proc()** function and specifying its interval and length. Note that specified parameters are only advisory and may be ignored. In particular, to achieve the target duty cycle (requested length / requested interval), when the actual interval is shorter, the length is reduced up to 25% of requested length.

If the scan window length is too short, the reception of advertise packets may be difficult, making the discovery of connectable devices slower.

**The best setting is to set the scan interval to be the same as the connection interval, and make the scan window length to be as large as the scan interval.**

Note that the SCAN slot is replaced by a CONN slot when the connection is established. See Figure 5.

**Figure 3. SCAN slot replaced by CONN slot when connection is established**



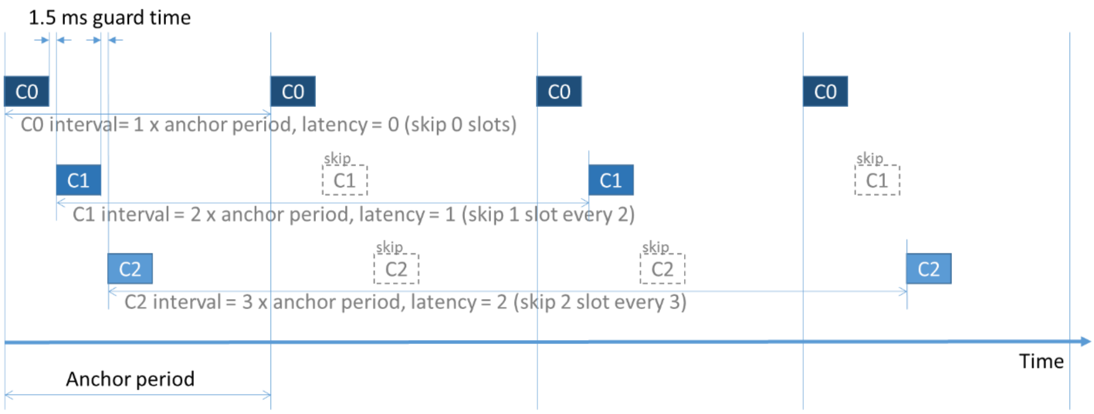

### Connection slot (SCAN)

The connection slot (CONN) is allocated by the central device (master) calling the **aci_gap_create_connection()** function and specifying its min and max interval, min and max length, the latency and the supervision timeout.

The latency is the number of connection events that the device can skip before the next connection. If the latency is set to N, the actual interval for the connection event is equal to the requested interval * (N+1). See Figure 6 for an example.

The supervision timeout is the maximum time between two correctly received packets before the connection is considered lost. **It must be larger than 2 * Interval * (N+1).**

**Figure 4. Example: setting Latency = 0, 1 or 2 for CONN slots**




## Slot allocation strategy

The time base is defined when the first slot is allocated. The beginning of the slot becomes the anchor point and the interval becomes the **anchor period**. The slot is followed by 1.5 ms guard time, therefore the length can be at most equal to interval – 1.5 ms.

Subsequent slots are allocated only if all constraints on interval and length are satisfied:

- There must be a multiple or sub-multiple of the anchor period that is in the requested min and max interval range (with the exception of the SCAN slot for which the requested interval is only advisory).
- There must be enough space left in the selected anchor period to accommodate the requested length plus 1.5 ms guard time (with the exception of the SCAN slot for which the requested length is only advisory).

Therefore, when the n-th slots is allocated, the anchor period can be kept as is or it can be reduced (See Figure 7):

- If N * anchor period satisfies the constraint on the interval, and if there is enough space left, the anchor period is kept as is and the latency of the new slot is set to N-1.
- If anchor period / N satisfies the constraint on the interval, and it is still multiple of 1.25 ms, and it has enough space for all existing slots plus the new slot (including their guard times), the anchor period is reduced and the latency of all other slots is multiplied by N (with the exception of the ADV slot which cannot reduce the anchor period and which cannot be allocated in this case).

**Figure 5. Anchor period can be kept as is or it can be reduced.**

| Slot type | No anchor period | Anchor period < min requested interval | Anchor period > max requested interval |
|---|---|---|---|
| ADV | Anchor period is set to average of min + 5 ms and max + 5 ms requested interval | **ADV is not allocated because interval cannot be increased by setting the latency.**<br>(anchor period < min + 5 ms)<br><br>Anchor period is kept as is. | **ADV slot is not allocated because anchor period cannot be reduced.**<br>(anchor period > max + 5 ms)<br><br>Anchor period is kept as is. |
| SCAN | Anchor period is set to requested interval | **Requested interval is ignored,** while length may be reduced up to 25% to achieve target duty.<br><br>Anchor period is kept as is. | **Requested interval is ignored.** Note that duty cycle will be lower than requested length/interval.<br><br>Anchor period is kept as is. |
| CONN | Anchor period is set to average of min and max requested interval | Find a multiple of the anchor period that is in the specified range<br>(min <= N * anchor period <= max).<br><br>Anchor period is kept as is.<br>Latency of new CONN slot is set to N-1. | Find sub-multiple of anchor period that is in the specified range<br>(min <= anchor period / N <= max).<br><br>**Anchor period divided by N if still multiple of 1.25 ms.** Latency of other CONN slots is multiplied by N. |

## Best strategy

- If ADV slot is used, allocate ADV slot first to set the anchor period. A possible SCAN slot will adapt because its parameters are only advisory. Subsequent CONN slots will use a multiple (latency>0) or sub-multiple of this anchor period. Specify a long interval to accommodate all slots. Specify a multiple of 10 ms to accommodate CONN slots with shorter interval, as it is still a multiple of 1.25 ms when divided by 2, 4 or 8.
- If SCAN slot is used, allocate it after other slots. It will adapt to existing anchor period. It will be replaced by CONN slot upon connection.

- If multiple CONN slots are used, the longest possible interval and the shortest possible length should be used to accommodate all slots. Specify an interval multiple of 10 ms to accommodate CONN slots with shorter interval, or use a multiple of the shortest interval if it is known in advance.

If the CONN slots are all the same, the minimum connection interval to maximize the throughput can be computed as follows:

**CONN interval = N slaves * (CONN length + 1.5 ms guard time)**

**Round to the next multiple of 1.25 and add 1.25 ms for margin**.The additional 1.25 ms is needed to account for internal rounding: 1.5 is not a multiple of 0.625 and, whenever a new slot is allocated, it is rounded to 1.875 or 1.25 alternatively by the embedded radio stack. There must be enough space for the guard time to be 1.875 ms for the last slot.

After a CONN slot is allocated, the user may call **aci_hal_get_anchor_period():** this function returns the connection interval and the remaining time between the last allocated connection slot and the end of the interval. The time interval is measured in 0.625 ms units. The 1.5 ms guard time is already subtracted.

### Interval, Length and Throughput for CONNection slots

To accommodate the target data rate, the connection length (duration) must be long and/or the connection interval (periodicity) must be short, to increase the connection frequency.

On the opposite to accommodate many slots: the connection length (duration) must be short and/or the interval (periodicity) must be long, reducing the connection frequency and the achievable data rate.

During the connection event, the master sends data to the slave (M>S), then the slave sends data to the master (S>M), and the cycle repeats until the connection event is over. Each data transfer requires a minimum of 80 usec (0 byte payload) and a maximum of 328 usec (20 bytes payload) and is followed by 150 usec guard time.

If the master does not send data, and the slave sends 20 bytes with each transfer, then each cycle requires 708 usec (80 + 150 + 328 + 150 usec). The maximum achievable throughput from slave to master can be estimated as follows:

**Throughput bits/s = Floor(Length ms / 0.708 ms) * 20 bytes * 8 bits / Interval ms**

As an example:

- Interval = 7.5 ms (shortest), Length = Interval – 1.5 ms, Throughput = 170.6 kbit/s
- Interval = 4000 ms (longest), Length = Interval – 1.5 ms, Throughput = 225.8 kbit/s

The effective throughput can be lower than the maximum achievable for several reasons:

- If there are losses on air, each packet is automatically re-transmitted until successfully acknowledged (or until the connection drops, if the supervision timeout expires). One must plan a connection length that accommodates re-transmissions. More re-transmissions are required when the transmitter power is reduced (low-power applications), or when the antenna efficiency is low (compact PCB antennas), or in presence of strong near interferences in the 2.4 GHz band.

- Transmission buffers can overflow while re-transmissions are in progress, or while the device is waiting for the connection slot to begin. The transmission buffer in all BlueNRG ASICs can accommodate up to 8 data packets (this is the maximum number of packets that can be transmitted when the connection interval is 7.5 ms and the length is 6 ms). If more buffering is needed, it must be handled at the application level.

- The device on the other end may have or set limitations. Smartphones cannot allow that a single application takes over the BLE radio stack. Therefore, the minimum connection interval may be higher than 7.5 ms (reducing the connection frequency and the achievable throughput). Transmission buffers can be limited for the same reason or because of limitations of the radio front-end.

## Concurrent Master and Slave roles

BlueNRG-MS, BlueNRG-1 and BlueNRG-2 support the concurrent master and slave roles. The device is the master of its slaves, but it is also the slave of at least another master. The other master may impose connection timings, which are in conflict with timings in the local network. The followings rules are adopted to guarantee the connectivity:

- A slave slot begins, with timing imposed by the other master; the end of this slave slot overlaps with the beginning of a master slot: the length of the slave slot is reduced, if this is possible (CONN length >= 3.75 ms); otherwise, a round-robin policy is adopted.

- A master slot begins with timing controlled locally; the end of this master slot overlaps with the beginning of a slave slot: a round-robin policy is adopted.

The round-robin policy ensures that all devices, the master and the other master(s), have their chance to complete the connection event in consecutive intervals. If in one interval a device wins, in the next interval the other device wins, and so on.

BlueNRG-MS can support a connection to up to 2 other masters (N = 0, 1, 2), while maintaining the connections to up to 8-N slaves (Mode 3, Master & Slave) or 4-N slaves (Mode 4, Master & Slave with simultaneous Advertising & Scanning).

BlueNRG-1 and BlueNRG-2 can support connections to up to 2 other masters (N = 0, 1, 2), while maintaining the connections to up to 8-N slaves (simultaneous Advertising and Scanning is always possible).

## Support material

| Related design support material |
|---|
| Databrief DB2956, STEVAL-STLKT01V1, SensorTile development kit |
| Databrief DB3446, STEVAL-IDB008V2, Evaluation platform based on BlueNRG-2 |
| **Documentation** |
| Datasheet DS10691, BlueNRG-MS, Upgradable Bluetooth® low energy network processor |
| Datasheet DS12166, BlueNRG-2, Bluetooth® low energy wireless system-on-chip |
| Programming manual, PM0237, BlueNRG, BlueNRG-MS stacks programming guidelines |
| Programming manual, PM0257, BlueNRG-1, BlueNRG-2 BLE stack programming guidelines |

## Revision history

| Date | Version | Changes |
|---|---|---|
| 26-Mar-2018 | 1 | Initial release |
| 23-Nov-2018 | 2 | Added constraint on Supervision timeout. Added best setting for Scan Window Length and Scan Interval. Added minimum connection length when Data Length Extension is supported. Added formula for min CONN interval and max throughput taking into account margin for internal rounding. |

## IMPORTANT NOTICE – PLEASE READ CAREFULLY