# AN5508

Application note

# Social Distancing Detection using Bluetooth® Low Energy

## Introduction

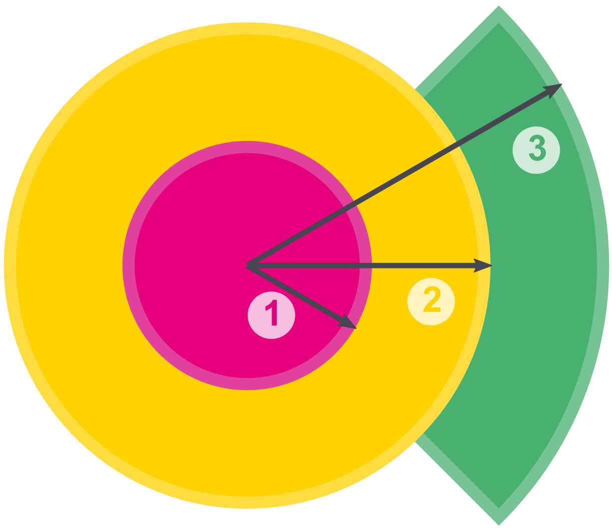
This social distancing application help users maintain safe distances and minimize the risk of contracting infectious diseases such as COVID-19 by generating an alert when two people come within a minimum set distance of each other, usually 2 meters or less. The application involves wireless Bluetooth® Low Energy (BLE) communication between wearable nodes based on the BlueNRG-1 or BlueNRG-2 systems on chip.

The nodes simultaneously advertise their presence and scan for the presence of other similar beacons in range, while deploying LowPower Modes during periods of inactivity to conserve battery power.

Advertising is performed on three channels and no connection or response packet are required between the devices during advertising and scanning activities (ADV_NONCONN_IND).

**Figure 1. Distance zones**

1. Alert zone = 2 meters
2. Pre-alert zone = 3 meters
3. Safe zone = more than 3 meters




**RELATED LINKS**

*Visit the Bluetooth website for more information on Bluetooth specification version 4.0 [vol 0] and V5.0*

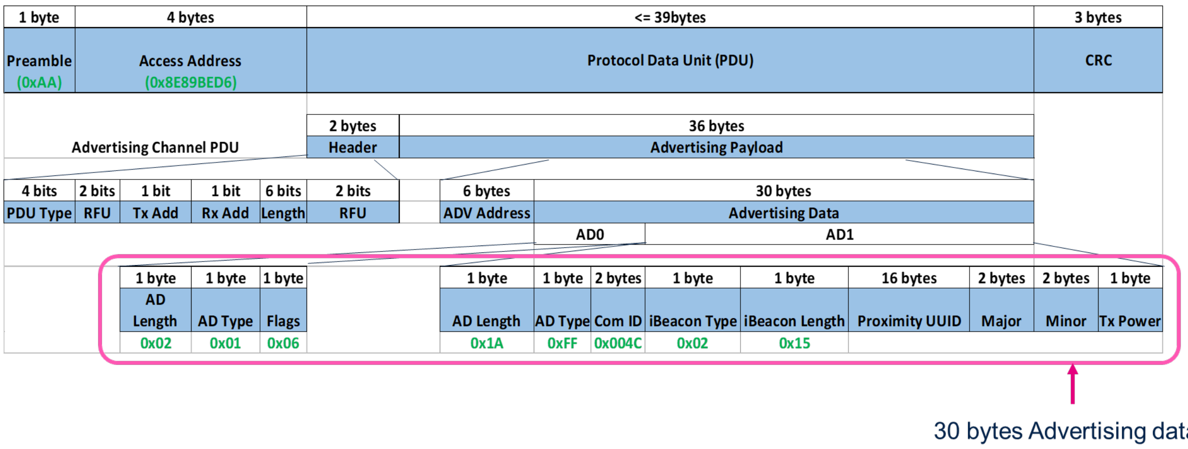
# 1 iBeacon standard

We chose the iBeacon advertising format for this application because it is relatively straightforward, it offers reliable performance, and works on both iOS and Android operating systems.

The standard uses a 30-byte packet length comprised of:

- 3-byte header: (AD length, AD type and flags)
- 27-byte payload: (AD length, AD type, Company ID, iBeacon type, iBeacon length, UUID, major number, minor number and transmit output power) parameters as shown in the figure below.

**Figure 2. iBeacon advertising packet**




Other beacon standards like Eddystone can also be used for transmitting data to other nodes, but these implement different packet protocols.

## RELATED LINKS

*See the Getting Started with iBeacon document for more information*

# 2 Overview of proximity detection

The proximity detection algorithm handles simultaneous advertising and scanning.

**Figure 3. Proximity detection based on RSSI**

1. Advertising:
- Generate Beacon (known Tx Power in beacon)
2. Scanning:
- Beacon detected
- Check RSSI
- Store device ID and time in Flash


**RELATED LINKS**

*Visit the following location for more information on how to calculate distances based on RSSI signal strength*

## 2.1 System startup and initialization

After power on, the system firmware perfoms the following sequence:
1. initializes the BLE stack
2. initializes the accelerometer:
   – calculates its gain and offset values
3. configures general discoverable mode operation:
   – `AD_TYPE` set to Non connectable undirected advertising mode (`ADV_NONCONN_IND`)
   – no connection between devices, only advertising and passive scanning
4. sets the scan window and scan interval parameter values:
   – the defaults can be changed according to user requirements

**Table 1. Default advertising and scanning intervals**

| Parameters | Normal Mode | High Probability Detection Mode | Comments |
|---|---|---|---|
| ADV_INTERVAL_MIN | N = 240 (150ms) | N= 160 (100ms) | Time = N * 0.625ms |
| ADV_INTERVAL_MAX | N = 240 (150ms) | N = 160 (100ms) | |
| SCAN_INTERVAL | N = 400 (250ms) | N = 240 (150ms) | |
| SCAN_WINDOW | N = 320 (200ms) | N = 160 (100ms) | |

## 2.2 System operation

As soon as a device detects a beacon, the `hci_le_advertising_report_event` call back function is invoked.

It provides the information that Bluetooth device has received some information during a passive scan.

The Controller maintains the queue of these advertising packets and provides the information from multiple devices in one LE Advertising Report event.

The function derives the Tx power and RSSI value and stores the samples individually in the `sBeacon` data structure, along with the current time stamp and node index value.

Currently in this structure storage of few samples is done (for example 20, this is configurable in software) from same device. If no beacons are received from the same address over a certain period of time, the data is flushed.

If the device is continuously receiving samples from the same device then samples will be stored in round robin format.

The configurable `MAX_NODE_COUNT` parameter sets the maximum number of different devices that samples will be stored for (default is 20).

The weighted mean and average filter are applied to remove the variations in the RSSI signal due to external factors such as absorption, interference and diffraction.

The distance from a detected beacon is determined from the following equation:

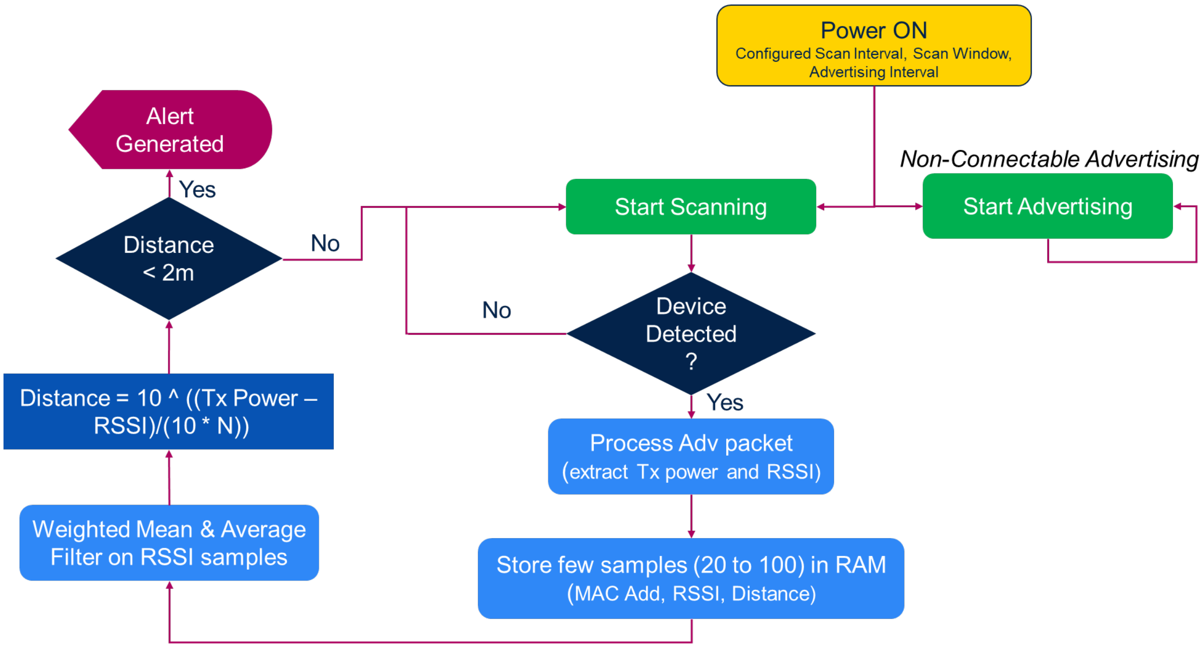
$$\text{Distance} = 10^{\frac{\text{TxPower}-\text{RSSI}}{10*\text{N}}}$$

where:

- TxPower = transmit power configured in the beacon packet and defined in the application
- RSSI = filtered Received Signal Strength Indicator
- N = Environmental factor; constant normally between 2 and 4 (4 is used in application)

The application signals when the calculated Distance value falls below a set threshold (e.g., 2 meters) with a flashing LED or buzzer alert.

**Figure 4. Flowchart of algorithm logic**

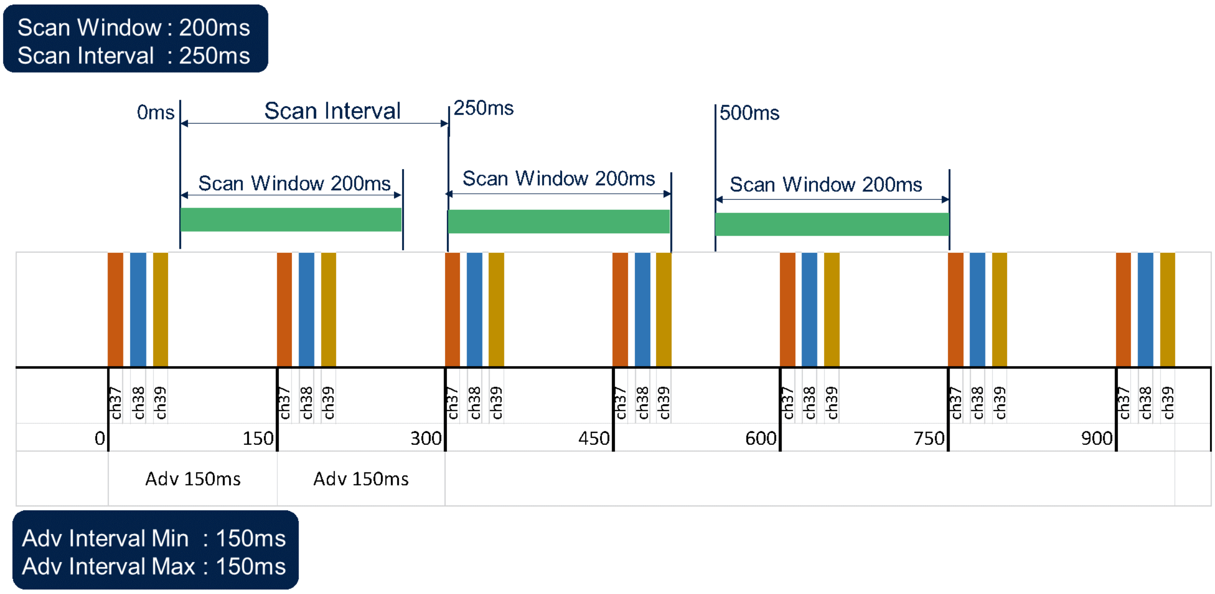



The beacon does not advertise continuously, but according to defined minimum and maximum advertising intervals (i.e., 150 ms in normal mode).

*Note:* *Adjusting the advertising interval and scan window parameters can impact battery life significantly.*

The interval must be an integer multiple of 0.625 ms (Time = N * 0.625 ms) from 20 ms to 10.24 s. But for non-connectable undirected event `type(ADV_NONCONN_IND)`, the minimum advertising interval can be 20 ms. There is also a random delay of 0 to 10 ms in each advertising packet, which is generated by the Link Layer.

**Figure 5. Advertising intervals and scan windows**




## 2.3 RSSI filtering techniques

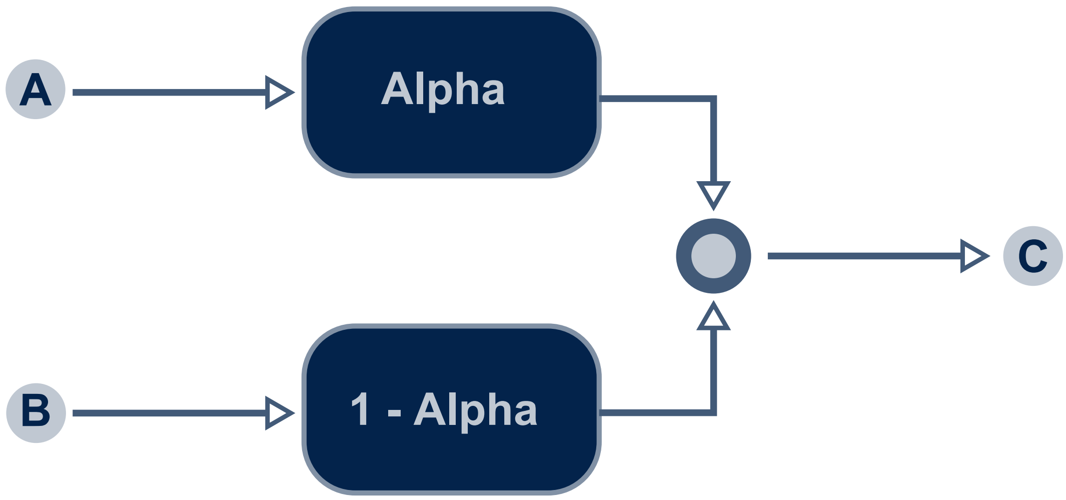
RSSI filtering is required to limit the potentially high variation in received signal strength due to multi-path reflections and environmental factors.

### 2.3.1 Weighted mean filter

Weighted mean filter is also known as feedback filtering because previous RSSI values contribute to the calculation of the filtered output.

Suppose A and B are measured values for signal C. Signal C is the estimated value after passing A and B through filters. As the noise measured in A or B might be high or low frequency, the low-pass filter Alpha is complemented by the high-pass filter 1 – Alpha, where Alpha is a value between 0 and 1.

**Figure 6. Advertising and scanning**



As the source of A and B derive from the same signal, their noise components should be similar in nature, and a weighted mean between two measured values can act as a basic complimentary filter. Experiments on different values of alpha showed a minimal variation for a value of alpha between 0.6 to 0. 7.

For an alpha of 0.6, 60% of the weighting is applied to the previous RSSI measurement and 40% to the current measurement. The current value of RSSI is then replaced with the filtered value and saved in RAM.

```
/************** Weighted Mean/Feedback Filtering ******************************/

int8_t Weighted_Mean(int8_t aPrevious_RSSI,int8_t aCurrent_RSSI)
{
  int8_t filter_rssi = 0;
  filter_rssi = (int8_t)(((1 - ALPHA) * aCurrent_RSSI) + (ALPHA * aPrevious_RSSI));
  return filter_rssi;
}
```

### 2.3.2 Average filter

Average filtering, or moving average filtering, further reduces noise by taking the average of a series of complementary or weighted mean filter outputs saved in RAM.

```
/****************** Average Filter ************************************/
int16_t Average_Filter(uint8_t aNodeIndex,uint8_t aSample_count)
{
  int16_t Total_RSSI=0;
  for(uint8_t count=0;count <= aSample_count ; count++)
  {
    Total_RSSI = Total_RSSI + sBeacon[aNodeIndex].rssi_value[count] ;
  }
  return (Total_RSSI / (aSample_count + 1)) ;
}
```

## 2.4 Accounting for tilt angles

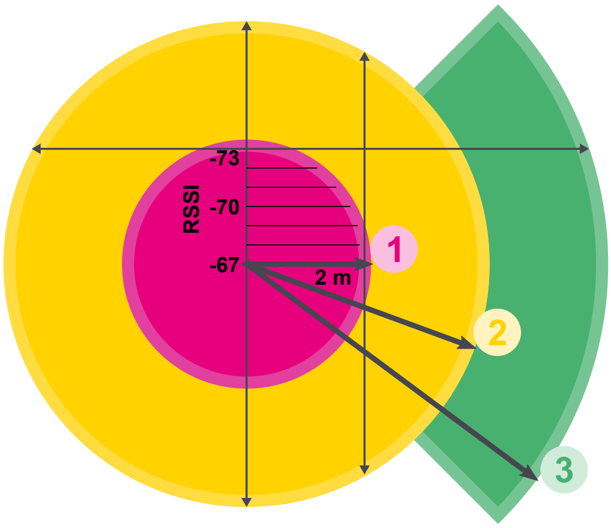
The results from testing indicate that the difference in tilt angle between a beacon and the detecting device has a greater impact on RSSI and distance measurement accuracy than their relative positions around each other.

The firmware therefore includes a lookup table to account for different tilt angles and ensure the best possible RSSI band capture for a node at the nominal safe distance of 2 meters.

The Relative tilt vs RSSI lookup table contains values derived from testing and can be refined by the user with further information regarding relative angles at similar distances. Conditional statement are derived from the preferred bands for RSSI sampling in the danger zone, where the distance between nodes is less than 2 meters.

**Figure 7. RSSI signal strength over distance**

1. Alert zone = 2 meters
2. Pre-alert zone = 3 meters
3. Safe zone = more than 3 meters




```
Cconst int table[ ][7] =
       {{Tiltx1, Tilty1, Tiltz1, Theta, RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1, Tilty1, Tiltz1 , Theta + 90, RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1, Tilty1, Tiltz1 , Theta + 180,  RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1, Tilty1, Tiltz1 , Theta + 270,  RSSI Min, RSSI Max, RSSI Avg},

       {Tiltx1 + 90, Tilty1, Tiltz1 + 90, Theta, RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1 + 90, Tilty1, Tiltz1 + 90 , Theta + 90, RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1 + 90, Tilty1, Tiltz1 + 90 , Theta + 180,  RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1 + 90, Tilty1, Tiltz1 + 90 , Theta + 270,  RSSI Min, RSSI Max, RSSI Avg},

       {Tiltx1, Tilty1 + 90, Tiltz1 + 90, Theta, RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1, Tilty1 + 90, Tiltz1 + 90 , Theta + 90, RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1, Tilty1 + 90, Tiltz1 + 90 , Theta + 180,  RSSI Min, RSSI Max, RSSI Avg},
       {Tiltx1, Tilty1 + 90, Tiltz1 + 90 , Theta + 270,  RSSI Min, RSSI Max, RSSI Avg},
       };
```

**Table 2. RSSI lookup table**

| Relative tilt X (deg) | Relative tilt Y (deg) | Relative tilt Z (deg) | Min RSSI (dBm) | Comments |
|---|---|---|---|---|
| 0 | 0 | 0 | -67 | Min RRSI band inside 2m varies from -67 to -73 |
| <20° | 20 < Y < 90 | 20 < Z < 90 | -69 | |
| <20 | < 20 | < 20 | -70 | |
| 20 < X < 90 | < 20 | 20 < Z < 90 | -73 | |
| Multiple combinations to be added | | | | |

## RELATED LINKS

*See the paper for Compensation of the Antenna Polarization Misalignment in the RSSI Estimation*

*4 Test results on page 11*

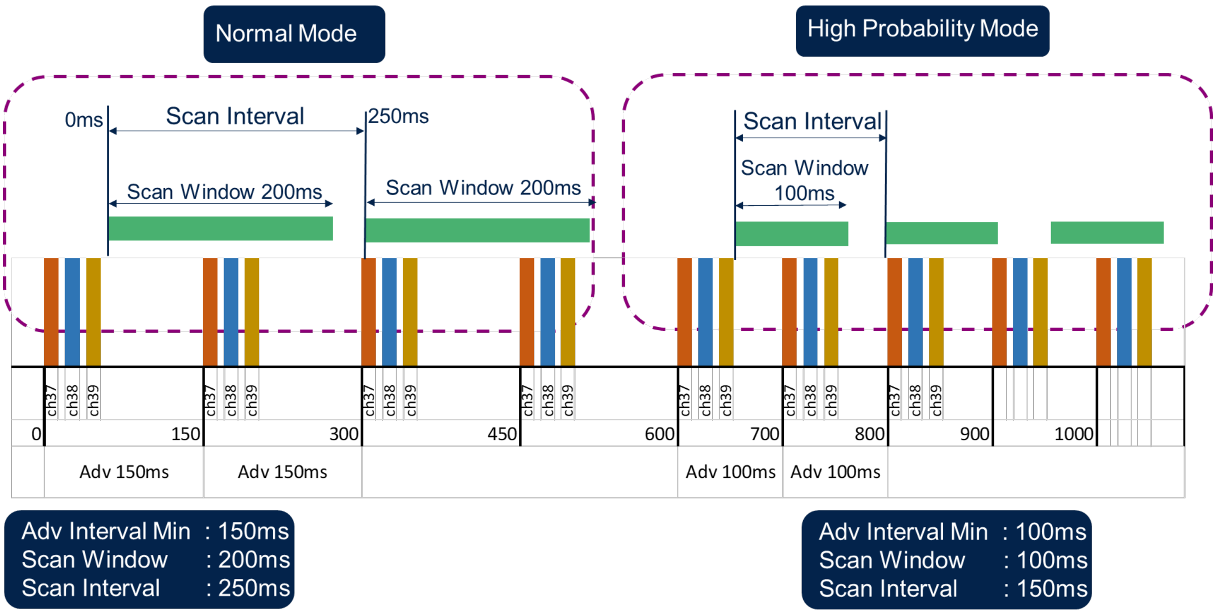
# 3 Device operating modes

## 3.1 High probability detection

The initial device state is the active Normal Mode with a 150 ms advertising interval, 250 ms scanning interval, and 200 ms scanning window.

The device enters High Detection Mode if it detects more beacons than a configurable threshold value (default is 5) within 500 ms. In this mode, the advertising interval is raised to 100 ms, the scanning interval to 150 ms, and scanning window to 100 ms.

**Figure 8. High detection scanning and advertising intervals**




## 3.2 Low power operation

Low Power Mode allows the device to save power and increase battery life when no beacon is detected or when the device remains stationary for a certain amount of time.

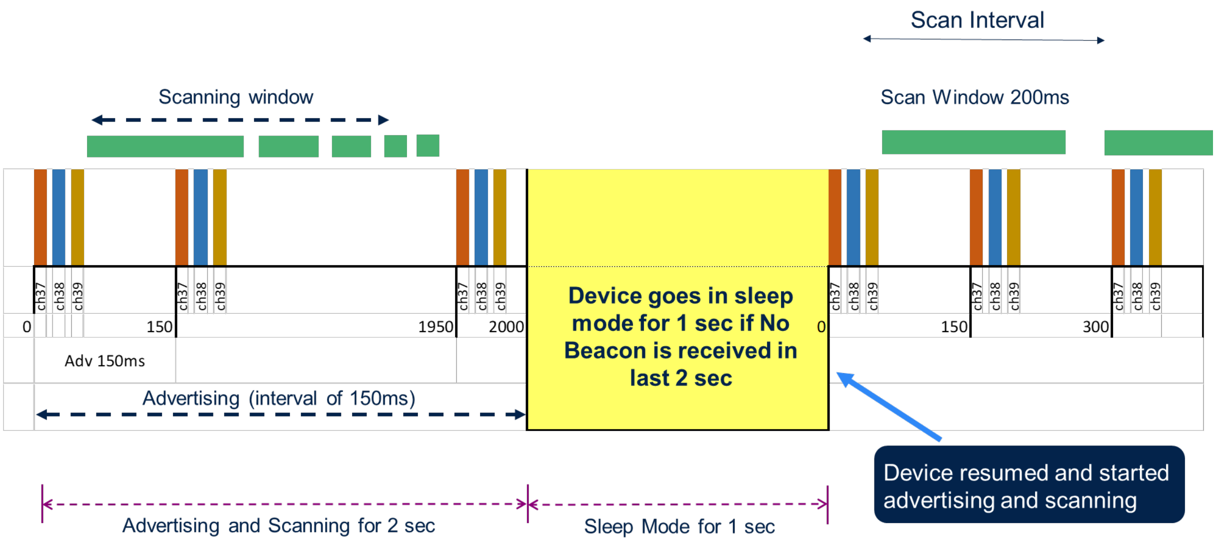
### 3.2.1 No beacon detected

If the device does not detect any beacons in range for `MAX_BEACON_LOSS_COUNT` (2 s) seconds, it enters low power sleep state for `DEVICE_SLEEP_DURATION` (1 s) second. The Observation Procedure is terminated and the device does not transmit any advertising data in connectionless packets.

The device is then placed in Non-discoverable Mode so it cannot enter the INQUIRY_SCAN state, after which the device enters Sleep Mode. Before putting the radio in sleep mode, the device checks that there is no ongoing transmission on the UART terminal.

The device can be woken from its sleep state either by a virtual timer or GPIO event. The virtual timer power management policy keeps the virtual timer running while the device is in sleep mode. After 1 second, the device wakes up and switches to Normal Mode, with the resumption of advertising and scanning activities.

**Figure 9.** **Low Power Mode activation**




### 3.2.2 No movement detected from accelerometer

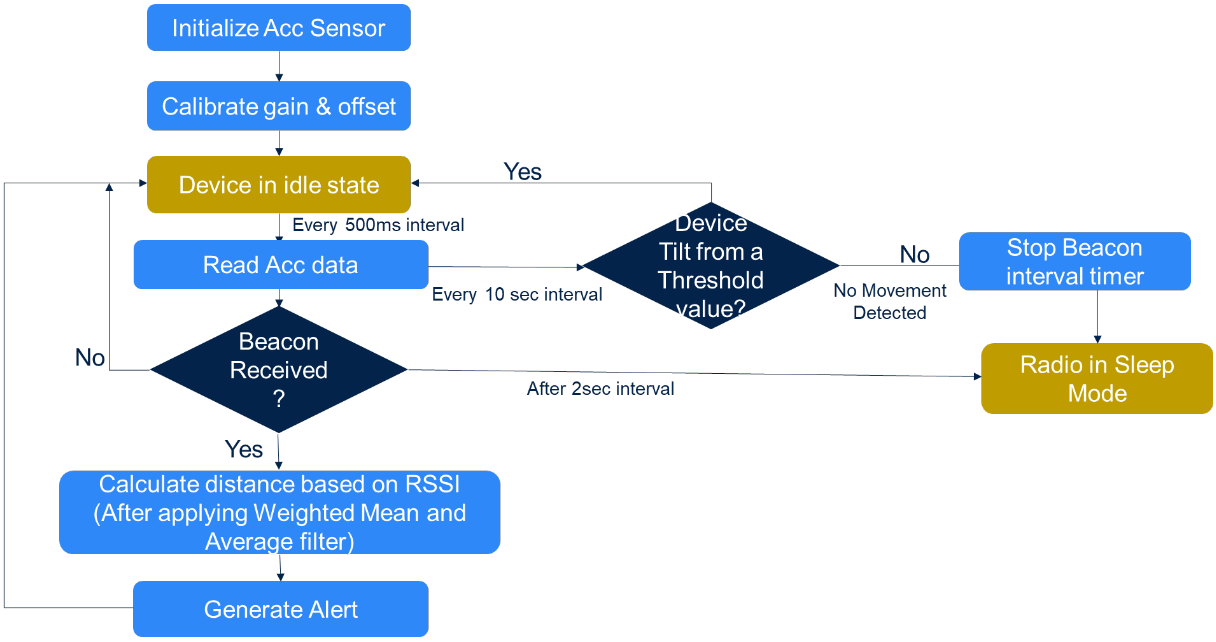
If no movement is detected by the accelerometer for `DEVICE_HALT_TIME` (60 s), it is assumed that user has removed the band and the device is put to sleep. If the device is tilted more that a threshold value, it is woken up and placed in Normal Mode by the GPIO interrupt (GPIO_Pin_13) connected to the accelerometer interrupt line.

The following table shows the accelerometer register configuration to generate the wake-up event using the high-pass digital filter.

**Table 3.** **Accelerometer register configuration for wake-up event**

| Register Name | Address | Value | Comments |
|---|---|---|---|
| CTRL1_XL | 0x10 | 0x60 | Turn on the accelerometer<br>( ODR_XL = 416 Hz, FS_XL = ±2 g) |
| TAP_CFG | 0x58 | 0x10 | Apply high-pass digital filter; latch mode disabled |
| WAKE_UP_DUR | 0x5C | 0x00 | No duration |
| WAKE_UP_THS | 0x5B | 0x02 | Set wake-up threshold |
| MD1_CFG | 0x5E | 0x20 | Wake-up interrupt driven to INT1 pin |

**Figure 10. No movement detection logic**



# 4 Test results

## 4.1 Impact of node rotation along an axis on RSSI

This test observed how tilting the antenna affected RSSI on a receiver, Node B, at a distance of 2 meters from the transmitter Node A.

Using the 3 accelerometer axes for reference, the device is rotated along the Y-axis to tilt the Z-X plane, and along the X-axis to tilt the Z-Y plane. The mean value of multiple RSSI measurements are taken for a series of axis rotations and corresponding tilt angles,and the derived distance is compared with the actual distance to determine the magnitude of error.

**Figure 11.** **Node rotation along an axis**

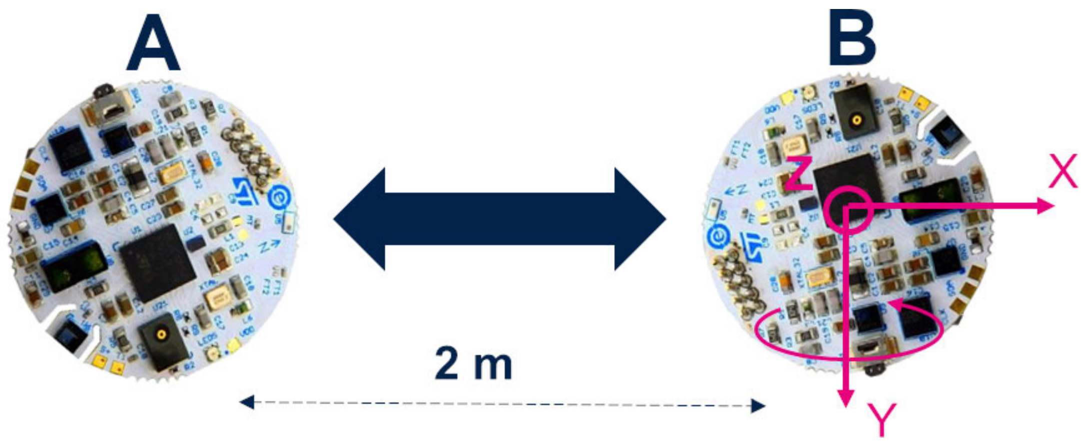



**Table 4.** **Results for tilting the Z-X plane of the receiver**

| Test Scenario | $\theta_X$ deg | $\theta_Y$ deg | $\theta_Z$ deg | RSSI | Distance | % Error |
|---|---|---|---|---|---|---|
| Node B rotate along Y-Axis (STEVAL-BCN002V1) | 0 | 0 | 0 | -70.43 | 2288.22 | 12.59 |
| | 45 | 0 | 45 | -64.78 | 1896.82 | -5.43 |
| | 90 | 0 | 90 | -70.03 | 2259.23 | 11.47 |
| | 45 | 0 | 135 | -68.93 | 2131.34 | 6.16 |
| | 0 | 0 | 180 | -69.11 | 2133.14 | 6.24 |

**Table 5.** **Results for tilting the Z-Y plane of the receiver**

| Test Scenario | $\theta_X$ deg | $\theta_Y$ deg | $\theta_Z$ deg | RSSI | Distance | % Error |
|---|---|---|---|---|---|---|
| Node B rotation along X-Axis (STEVAL-BCN002V1) | 0 | 0 | 0 | -70.43 | 2288.22 | 12.59 |
| | 0 | 45 | 45 | -70.05 | 2251.90 | 11.18 |
| | 0 | 90 | 90 | -59.99 | 1840.60 | -8.66 |
| | 0 | 45 | 135 | -69.70 | 2212.35 | 9.59 |
| | 0 | 0 | 180 | -66.62 | 1852.03 | -7.98 |

## 4.2 Impact of relative node position on RSSI

This test observed how the position of transmitter Node B with respect to the antenna on Node A impacted RSSI at a constant distance of approximately 2 meters.

Several RSSI sample measurements are taken for each position of Node B in a circle around Node A, without any appreciable tilt, and the magnitude of the error between the distances derived from RSSI and actual distances is determined.

**Figure 12. Node positioning with respect to a fixed node**

1. SMD antenna
A. Node A in fixed position
B1-B4. Different positions of Node B around Node A (0°, 90°, 180°, 270°)

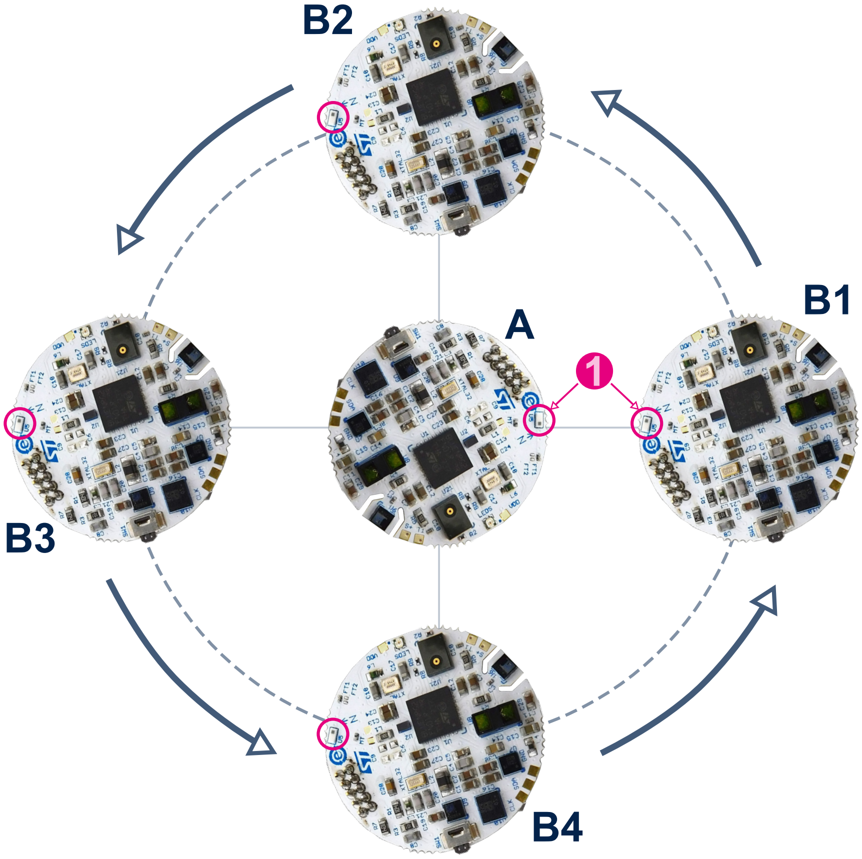



**Table 6. Result for different positions of Node B around Node A**

| Test Scenario | X (0°) | Y (0°) | Z (0°) | RSSI | Distance | Error (%) |
|---|---|---|---|---|---|---|
| Node B at 0° (line of sight) | 0 | 0 | 0 | -69.91 | 2246.73 | 10.98 |
| Node B at 90° | 0 | 0 | 0 | -69.05 | 2138.70 | 6.48 |
| Node B at 180° | 0 | 0 | 0 | -67.04 | 1891.32 | -5.74 |
| Node B at 270° | 0 | 0 | 0 | -69.76 | 2227.58 | 10.21 |

## Revision history

Table 7. **Document revision history**

| Date | Version | Changes |
|---|---|---|
| 01-Jun-2020 | 1 | Initial release. |

# Contents

## List of figures

# List of tables