

# eSense BLE Specifications

Commands, and Data Structure

---

Author	Pervasive Systems
Organization	Nokia Bell Labs, Cambridge
Last Update	June 12, 2019

---

# Contents

1	Overview.....	3
2	List of BLE Characteristics.....	5
2.1	General Command Structure .....	5
2.2	Start/Stop IMU Data Sampling.....	6
2.3	Change Device Name .....	7
2.4	Get Device Name .....	7
2.5	Change Advertising and Connection Interval .....	7
2.6	Get Advertising and Connection Interval .....	9
2.7	Get Battery Voltage.....	10
2.8	Get Accelerometer Factory Offset.....	10
2.9	Change/Get IMU Full Scale Range.....	11
2.10	Change/get IMU Low Pass Filter Configuration.....	13
3	List of BLE Notifications Sent from eSense.....	16
3.1	IMU Sensor Data .....	16
3.2	Push Button Status .....	17
4	Orientation of IMU Axes.....	18

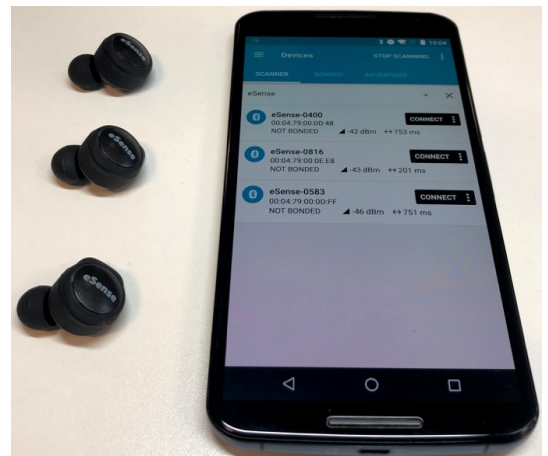
# 1 Overview

This document describes the characteristics and the commands used to interact with eSense through its BLE interface (for the left earbud). Any host device (phone, laptop, raspberry pi, ...) capable of connecting to a BLE peripheral is capable of interacting with eSense by reading and writing the characteristics detailed below.

**All changes to the configuration of eSense are permanent and preserved even after the device is turned off or reset.**

All left earbuds have a BLE interface which can be used to configure various aspects of the IMU sensor and collect accelerometer and gyroscope data. The left earbuds also transmit periodic BLE advertisement packets (with an interval between 625ms and 750ms, by default), even during a BLE connection, and as such can be used for proximity detection. The right earbuds do not have a BLE interface.

Each left earbud has a unique device name in the format *eSense-`<four digits number>`*. The easiest way to discover the name of an earbud is using a BLE scanner application on a host device (e.g. smartphone). We recommend *Nordic nRF Connect* for iOS and Android or *BLE Scanner* for iOS. As you can see in the image once a scanning filter has been set up with the name *eSense* the app discovers the three earbuds and shows useful information (MAC address, RSSI and advertisement interval). The same apps can also be used to connect to the earbuds and explore the available BLE characteristics.



Notice that it might take several seconds before the apps show the device name of the earbuds, during this initial period the apps might show only the MAC address and the other info but not the name.

The advertisement packets contain the Complete List of 16-bit Service Class UUIDs and the Complete Local Name of the device as defined by the Generic Access Profile (<https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile/>).

There is no need to pair the left earbud with a host device to use its BLE interface. The pairing of the Bluetooth Classic interface is required only to stream music to the earbud or record audio from its microphone. Only one BLE connection at the time is supported, if multiple hosts try to connect to the earbud's BLE interface at the same time, the earbud might become unstable and stop working.

## 2 List of BLE Characteristics

All the characteristics are contained inside the BLE service 0xFF06, unless otherwise specified. For the characteristics that support it, notification can be enabled by writing on the descriptor 0x2902.

UUID	Purpose	Read/Write	Notification
0xFF07	Start/stop IMU sampling and configure advertisement and connection intervals	R/W	No
0xFF08	Get sensor data when IMU sampling is enabled	R	Yes
0xFF09	Get push button status	R	Yes
0xFF0A	Get battery voltage	R	No
0xFF0B	Get advertisement and connection intervals	R	No
0xFF0C	Change device name	W	No
0xFF0D	Get accelerometer factory offset	R	No
0xFF0E	Get/set accelerometer and gyroscope configuration (range and low pass filter)	R/W	No
0x2A00	Get device name (inside service 0x1800)	R	No

### 2.1 General Command Structure

Some characteristics perform different tasks based on the value (i.e. command) that is written. Commands follow a common structure as described below:

Cmd Header	Checksum	DataSize	Data0	Data1	...	DataN
(1byte)	(1byte)	(1byte)	(1byte)	(1byte)		(1byte)

**Checksum = Data Size + Data0 + Data1 + ....+ DataN (if longer than 1 byte use only least significant byte).**

Notice: if the command has no data, then Data Size = 0 and there is no Data bytes. So this command total 3 bytes.

## 2.2 Start/Stop IMU Data Sampling

This command starts/stops the sampling of the IMU on the eSense device specifying the desired sampling rate. Once the sampling has started the IMU data is streamed to the connected host device using BLE notifications (see [Section 3.1](#) for more details).

Command to write to characteristic 0xFF07:

CmdHead (1 byte)	Checksum (1 byte)	DataSize (1 byte)	Data0 (1 byte)	Data1 (1 byte)
0x53		0x02	ON/OFF	Sample Rate

Checksum = DataSize + Data0 + Data1 (if longer than 1 byte use only least significant byte)

ON/OFF = 0x01 (Starts the data sampling)

ON/OFF = 0x00 (Stops the data sampling)

Sample Rate (Hz) = 1 – 100 (in HEX)

### Examples:

Start sampling at 50Hz: 0x53 0x35 0x02 0x01 0x32

Start sampling at 20Hz: 0x53 0x17 0x02 0x01 0x14

Stop sampling: 0x53 0x02 0x02 0x00 0x00

To check if the value has been set correctly read back the same characteristic and check that the value returned is the same as the value written.

Given that the data is streamed through a BLE connection it might be necessary to lower the connection interval (see [Section 2.5](#)) to achieve high data rates. Also, consider that the number of operations performed by the earbud affects the achievable data rate. For example, if the IMU and

the microphone are enabled on the same earbud while the user is also listening to music, it might not be possible to achieve the desired data rate and the music might be interrupted.

### 2.3 Change Device Name

Write new device name in ASCII to characteristic 0xFF0C.

Data max length: 22 bytes

**Example:**

New device name is: ABCD

The data to write in the characteristic 0xFF0C is: 0x41 0x42 0x43 0x44

### 2.4 Get Device Name

To read the device name you need to read the characteristic 0x2A00 contained in the service 0x1800. This is the standard BLE characteristic for the device name.

### 2.5 Change Advertising and Connection Interval

This command changes the advertising and connection intervals on eSense. Both values are specified with a minimum and maximum value which the device will use to set the actual intervals.

The advertisement interval determines the frequency of BLE beacons transmitted from eSense. eSense transmits periodic BLE advertisement packets (with an interval between 625ms and 750ms, by default) even during a BLE connection. Lowering the advertisement interval increases the frequency of transmissions but increases the power consumption. The advertisement interval needs to be written on eSense in units of 0.625ms. For example, if an interval of 1000ms is required, the value to be written is  $1000/0.625 = 1600$ .

The configured connection interval is suggested from eSense during a connection with a host device (e.g. phone) but it is the host device which selects the actual interval. Lowering the connection interval increases the rate at which data is received from eSense but also increases the power consumption as more frequent radio operations will take place.

The connection intervals need to be written on eSense in units of 1.25ms. For example, if an interval of 1000ms is required, the value to be written is  $1000/1.25 = 800$ . Setting the min interval to 20ms and the max interval to 40ms allows to achieve the highest IMU data rate of 100Hz (on average, without music playing or audio recording).

For more details regarding advertisement interval and connection interval, please refer to the BLE core specification 4.2 (<https://www.bluetooth.com/specifications/archived-specifications/>).

**Notice:** these values become effective after the host disconnects from the earbud and re-connects again.

Command to write to characteristic 0xFF07:

CmdHead (1byte)	Checksum (1byte)	DataSize (1byte)	Data0 (1byte)	Data1 (1byte)	...	Data7 (1byte)
0x57		0x08				

Checksum = DataSize + Data0 + Data1 + ... + Data7 (if longer than 1 byte use only least significant byte)

Data0 : adv\_interval\_min Most Significant Byte

Data1 : adv\_interval\_min Least Significant Byte

Data2 : adv\_interval\_max Most Significant Byte

Data3 : adv\_interval\_max Least Significant Byte

Data4 : connection\_interval\_min Most Significant Byte

Data5 : connection\_interval\_min Least Significant Byte

Data6 : connection\_interval\_max Most Significant Byte

Data7 : connection\_interval\_max Least Significant Byte

The octets pairs (i.e. Data0-Data1, Data2-Data3, ...) are **unsigned 16-bit integers** and therefore need to be stored in appropriate data structures.

**Limits:**

Adv Interval Min >= 100



Adv Interval Max <= 5000

Adv Interval Max >= Adv Interval Min

Conn Interval Min >= 20

Conn Interval Max <= 2000

Conn Interval Max >= Conn Interval Min

Conn Interval Max – Conn Interval Min >= 20

### Examples:

Adv. Interval between 625ms and 750ms – Conn. interval between 90ms and 110ms:

0x57 0x47 0x08 0x03 0xe8 0x04 0xb0 0x00 0x48 0x00 0x58

These are the default values on eSense.

Adv. Interval between 125ms and 187ms – Conn. interval between 90ms and 110ms:

0x57 0x9d 0x08 0x00 0xc8 0x01 0x2c 0x00 0x48 0x00 0x58

## 2.6 Get Advertising and Connection Interval

Read characteristic 0xFF0B and the format will be the same as the command used to write it (see [Section 2.5](#)). To convert the High and Low bytes and obtain the values in milliseconds use the following equations:

Adv Interval min = (Data0\*256 + Data1) \* 0.625;

Adv Interval max = (Data2\*256 + Data3) \* 0.625;

Conn Interval min = (Data4\*256 + Data5) \* 1.25;

Conn Interval max = (Data6\*256 + Data7) \* 1.25;

## 2.7 Get Battery Voltage

Read characteristic 0xFF0A to get the battery voltage of the eSense device. The format of the received data is the following:

<b>CmdHead (1byte)</b>	<b>Checksum (1byte)</b>	<b>DataSize (1byte)</b>	<b>Data0 (1byte)</b>	<b>Data1 (1byte)</b>	<b>Data2 (1byte)</b>
0x56		0x03	BattVol_H	battVol_L	ChargeStatus

Checksum = DataSize + Data0 + Data1 + Data2 (only least significant byte)

Battery voltage = (BattVol\_H\*256 + BattVol\_L) / 1000 (unit: V)

ChargeStatus = 0 not charging

ChargeStatus = 1 is charging

## 2.8 Get Accelerometer Factory Offset

The accelerometer inside the IMU is factory calibrated and the trim values are stored inside registers in the IMU. By reading the characteristic 0xFF0D its possible to read the content of these registers. The data read from the characteristic has the following format.

<b>CmdHead (1byte)</b>	<b>Checksum (1byte)</b>	<b>DataSize (1byte)</b>	<b>Data0 (1byte)</b>	<b>...</b>	<b>Data11 (1byte)</b>
0x58		0x0C			

Checksum = DataSize + Data0 + ... + Data11 (only least significant byte)

Data0, Data1 = Most Significant and Least Significant Byte of Gyro X-axis calibration

Data2, Data3 = Most Significant and Least Significant Byte of Gyro Y-axis calibration

Data4, Data5 = Most Significant and Least Significant Byte of Gyro Z-axis calibration

Data6, Data7 = Most Significant and Least Significant Byte of Accelerometer X-axis calibration

Data8, Data9 = Most Significant and Least Significant Byte of Accelerometer Y-axis calibration

Data10, Data11 = Most Significant and Least Significant Byte of Accelerometer Z-axis calibration

The octets pairs (i.e. Data0-Data1, Data2-Data3, ...) are **signed 16-bit integers** and therefore need to be stored in appropriate data structures. The calibration values for the gyroscope will always be zero because the gyroscope is not factory calibrated. The values for the accelerometer are in  $\pm 16G$  format, therefore  $1g = 2048$  read from the registers.

## 2.9 Change/Get IMU Full Scale Range

The IMU on eSense allows to configure the maximum range of the gyroscope and accelerometer. The supported ranges are  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  and  $\pm 16g$  for the accelerometer and  $\pm 250\text{deg/s}$ ,  $\pm 500\text{deg/s}$ ,  $\pm 1000\text{deg/s}$  and  $\pm 2000\text{deg/s}$  for the gyroscope. The default configuration is  $\pm 4g$  for the accelerometer and  $\pm 500\text{deg/s}$  for the gyroscope. See [Section 4](#) for details about the axes orientation.

By writing on the characteristic 0xFF0E it is possible to select these ranges, while reading the characteristic allows to find the current configuration. The format of the data read from the characteristic is the following.

<b>CmdHead</b> <b>(1byte)</b>	<b>CheckSum</b> <b>(1byte)</b>	<b>DataSize</b> <b>(1byte)</b>	<b>Data0</b> <b>(1byte)</b>	<b>Data1</b> <b>(1byte)</b>	<b>Data2</b> <b>(1byte)</b>	<b>Data3</b> <b>(1byte)</b>
0x59		0x04				

CheckSum = DataSize + Data0 + Data1 + Data2 + Data3 (only least significant byte)

Data0 = General configuration register (default value is 0x06).

Data1 = Gyro Configuration register (default value is 0x08).

Data2 = Accelerometer Configuration register (default value is 0x08).

Data3 = Accelerometer Configuration register 2 (default value is 0x06).

The gyro range can be selected by changing the bits 4 and 3 in Data1 as indicated in the table below.

<b>Data1 – Gyro Configuration register</b>			
Bits [7:5]	Bits [4:3]	Bit 2	Bits [1:0]
Reserved	Gyro range bits 00 = $\pm 250$ deg/s 01 = $\pm 500$ deg/s 10 = $\pm 1000$ deg/s 11 = $\pm 2000$ deg/s	Reserved	Gyro LPF Bypass

The accelerometer range can be selected by changing the bits 4 and 3 in Data2 as indicated in the table below.

<b>Data2 – Accelerometer Configuration register</b>		
Bits [7:5]	Bits [4:3]	Bits [2:0]
Reserved	Accelerometer range bits 00 = $\pm 2$ g 01 = $\pm 4$ g 10 = $\pm 8$ g 11 = $\pm 16$ g	Reserved

**When changing the range of accelerometer or gyroscope make sure to change only the bits specified above leaving all the others at the same value (also for Data0 and Data3 bytes).**

The table below reports the sensitivity scale factors for the various ranges. These are necessary to convert the ADC values read from the IMU to acceleration in g or rotational speed in deg/s, for example (see [Section 3.1](#) for more details).

Accelerometer		Gyroscope	
Range	Scale Factor	Range	Scale Factor
± 2 g	16384 LSB/g	± 250 deg/s	131 LSB/(deg/s)
± 4 g	8192 LSB/g	± 500 deg/s	65.5 LSB/(deg/s)
± 8 g	4096 LSB/g	± 1000 deg/s	32.8 LSB/(deg/s)
± 16 g	2048 LSB/g	± 2000 deg/s	16.4 LSB/(deg/s)

## 2.10 Change/get IMU Low Pass Filter Configuration

The IMU on eSense allows to configure a low pass filter (LPF) for the accelerometer and one for the gyroscope. The configuration can be read/modified from the characteristic 0xFF0E (see [Section 2.9](#) for its format).

The options available for the gyroscope are listed in the table below.

Gyro LPF Bypass (Bits [1:0] in Data1)	Gyro LPF Conf (Bits [2:0] in Data0)	Bandwidth (Hz)
1	x	Highest (8800)
0	0	250
0	1	184
0	2	92
0	3	41
0	4	20
0	5	10
0	6	5
0	7	3600

Bits 0 and 1 in Data1 (characteristic 0xFF0E) control if the gyro LPF is bypassed or not as defined below:

<b>Data1 – Gyro Configuration register</b>			
Bits [7:5]	Bits [4:3]	Bit 2	Bits [1:0]
Reserved	Gyro range bits	Reserved	Gyro LPF Bypass 1 = LPF bypassed 0 = LPF bypassed

While the bits 0, 1 and 2 in Data0 select the configuration of the filter when it is not bypassed:

<b>Data0 – General Configuration register</b>	
Bits [7:3]	Bits [2:0]
Reserved	Gyro LPF Conf Value between 0 and 7

The options available for the accelerometer are listed in the table below.

<b>Accelerometer LPF Bypass</b> (Bit 3 in Data3)	<b>Accelerometer LPF Conf</b> (Bits [2:0] in Data3)	<b>Bandwidth (Hz)</b>
1	x	Highest (1.13k)
0	0	460
0	1	184
0	2	92
0	3	41
0	4	20
0	5	10
0	6	5

Bit 3 in Data3 controls if the accelerometer LPF is bypassed or not while bits 0, 1 and 2 select the configuration of the filter when it is not bypassed:

<b>Data3 - Accelerometer Configuration register 2</b>		
Bits [7:4]	Bit 3	Bits [2:0]
Reserved	Acc LPF Bypass 1 = LPF bypassed 0 = LPF bypassed	Acc LPF Conf Value between 0 and 6

**When changing the LPF configuration for accelerometer or gyroscope make sure to change only the bits specified above leaving all the others at the same value.**

The default configuration enables a low pass filter with 5Hz bandwidth for the accelerometer and a low pass filter with 5Hz bandwidth for the gyroscope.

## 3 List of BLE Notifications Sent from eSense

### 3.1 IMU Sensor Data

Subscribe to the BLE notifications from characteristic 0xFF08 to get new sensor data when available (if IMU sampling is enabled). This characteristic can also be read (without notification). The format of the received data is the following:

<b>CmdHead</b> <b>(1byte)</b>	<b>PacketIndex</b> <b>(1byte)</b>	<b>Checksum</b> <b>(1byte)</b>	<b>DataSize</b> <b>(1byte)</b>	<b>Data0</b> <b>(1byte)</b>	<b>....</b>	<b>Data11</b> <b>(1byte)</b>
0x55	0~255		0x0C			

PacketIndex = sequential number from 0 to 255

Checksum = DataSize + Data0 + Data1 + ... + Data11 (only least significant byte)

Data0, Data1 = Most Significant and Least Significant Byte of Gyro X-axis

Data2, Data3 = Most Significant and Least Significant Byte of Gyro Y-axis

Data4, Data5 = Most Significant and Least Significant Byte of Gyro Z-axis

Data6, Data7 = Most Significant and Least Significant Byte of Accelerometer X-axis

Data8, Data9 = Most Significant and Least Significant Byte of Accelerometer X-axis

Data10, Data11 = Most Significant and Least Significant Byte of Accelerometer X-axis

The octets pairs (i.e. Data0-Data1, Data2-Data3, ...) are **signed 16-bit integers** and therefore need to be stored in appropriate data structures. The data read for each axis is in ADC format as read directly from the sensor and can be converted to different units as shown below using the scale factors described in [Section 2.9](#).

#### Accelerometer data conversion to m/s<sup>2</sup>

Float value in m/s<sup>2</sup> = (Acc value / Acc scale factor) \* 9.80665



### Accelerometer data conversion to g

Float value in g = Acc value / Acc scale factor

### Gyroscope data conversion to deg/s

Float value in deg/s = Gyro value / Gyro scale factor

## 3.2 Push Button Status

Subscribe to the BLE notifications from characteristic 0xFF09 to get updates about the push button. This characteristic can also be read (without notification). The format of the received data is the following:

<b>CmdHead (1byte)</b>	<b>Checksum (1byte)</b>	<b>DataSize (1byte)</b>	<b>Data0 (1byte)</b>
0x54		0x01	

Checksum = DataSize + Data0 (only least significant byte)

Data0:

bit0 = 1, button is pressed

bit0 = 0, button is not pressed

# 4 Orientation of IMU Axes

The figure below shows the orientation of the IMU axes and the polarity of rotation in relation to the earbud.

