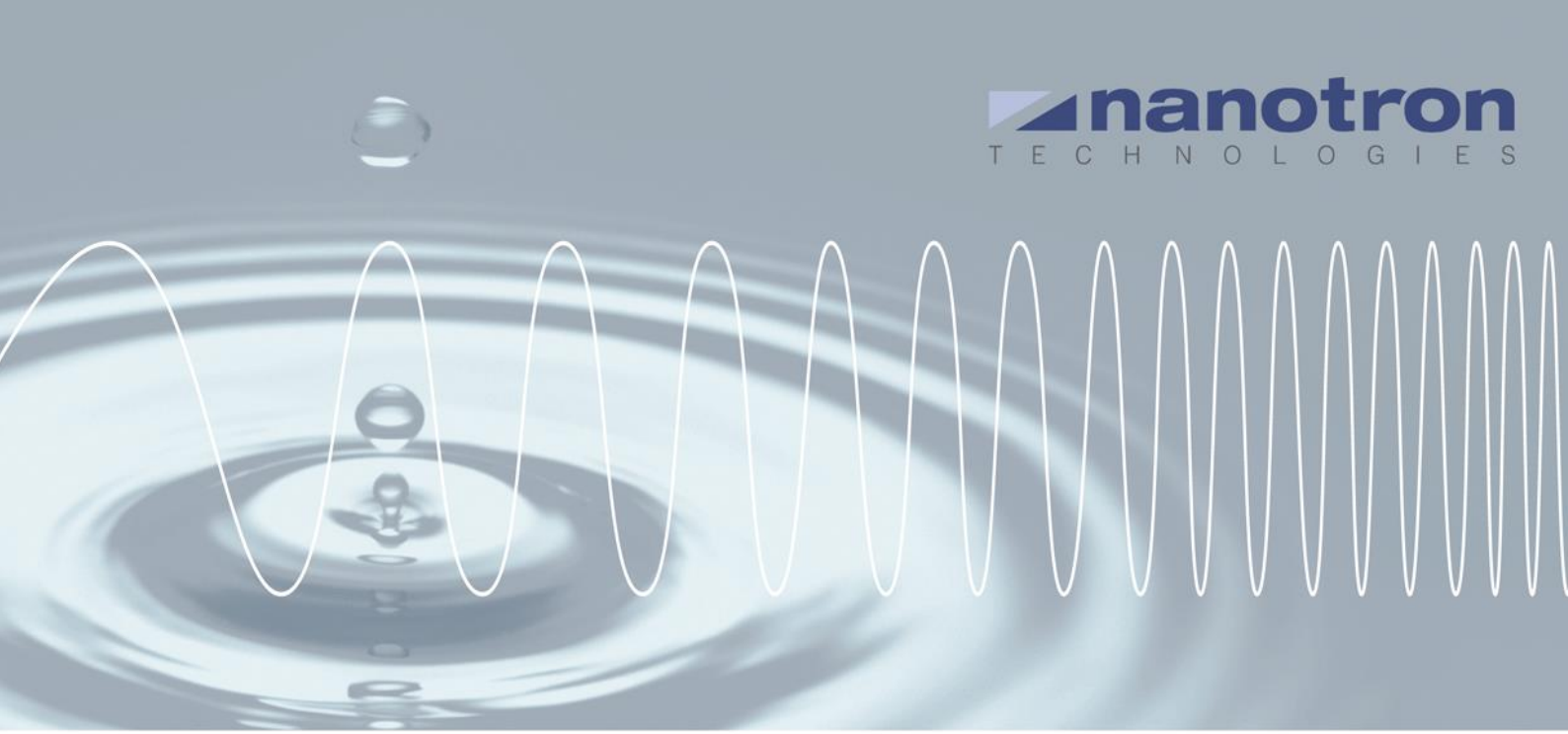

AN0505 Using 3D Acceleration and Temperature Sensor Data from swarm Bee LE

1.1

NA-14-0267-0016-1.1

Document Information

Document Title:	AN0505 Using 3D Acceleration and Temperature Sensor Data from swarm Bee LE
Document Version:	1.1
Current Date:	2016-04-13
Print Date:	2016-04-13
Document ID:	NA-14-0267-0016-1.1
Document Author:	Maria Luisa Arbulo

Contents

1. General introduction

The swarm bee is equiped with a MEMS digital triaxial acceleration sensor which senses tilt, motion and shock vibration in three perpendicular axes. Moreover, it is equiped with a temperature sensor that can be also to compensate the acceleration data.
The MEMS is fully integrated in the system; it can be controlled by some functions provided by the API and its data can be read out as part of the swarm information payload.

This document describes the characteristics of the sensors and how they can be used.

2. Sensors characteristics

The following table describes the main characteristic of the MEM sensor [1]:

Table 2-1: Sensor's parameters

Parameter	Condition	Typ	Units
Acceleration range		±2	g
		±4	g
		±8	g
		±16	g
Device resolution	2g	3.91	mg
Sensitivity	2g, T=25°C	256	LSB/g
	4g, T=25°C	128	LSB/g
	8g, T=25°C	64	LSB/g
	16g, T=25°C	32	LSB/g
Sensitivity temperature drift	2g, -40<T<+85	±0.02	%/K
Zero-g offset	2g, T=25°C	±180	
Zero-g offset, temperature drift	2g, -40<T<+85	±1	
Bandwith		8	Hz
		16	Hz
		31	Hz
		63	Hz
		125	Hz
		250	Hz
		500	Hz

		1000	Hz
Cross axis sensitivity		1	%
Temperature sensor measurement range		-40 .. +87.5	°C
Temperature sensor slop	T=25°C	0.5	LSB/K
Temperature sensor offset	T=25°C	±5	K
Wake-up time		0.8	ms
Start-up time		2	ms
Supply current, suspend mode		139	µA
Supply current, normal mode		0.5	µA
Operating temperature		-40 .. +85	°C

The sensor is power with +2.6V.

3. Sensor data

3.1. Acceleration data

The acceleration data is given as a 2-Byte hexadecimal number per axis; the units are mg.
The obtained values have been filtered before making them available to the swarm. The sampling rate of the filter is twice its bandwidth, which can also be set by the user. Table 3-1 shows the relation between the bandwidth of the filter and its sampling rate:

Table 3-1: Bandwiths and their corresponding update rates

BW (Hz)	7.85	15.63	31.25	62.5	125	250	500	1000
Update time (ms)	64	32	16	8	4	2	1	0

The orientation of the sensor inside the swarm module is described in the following figures.

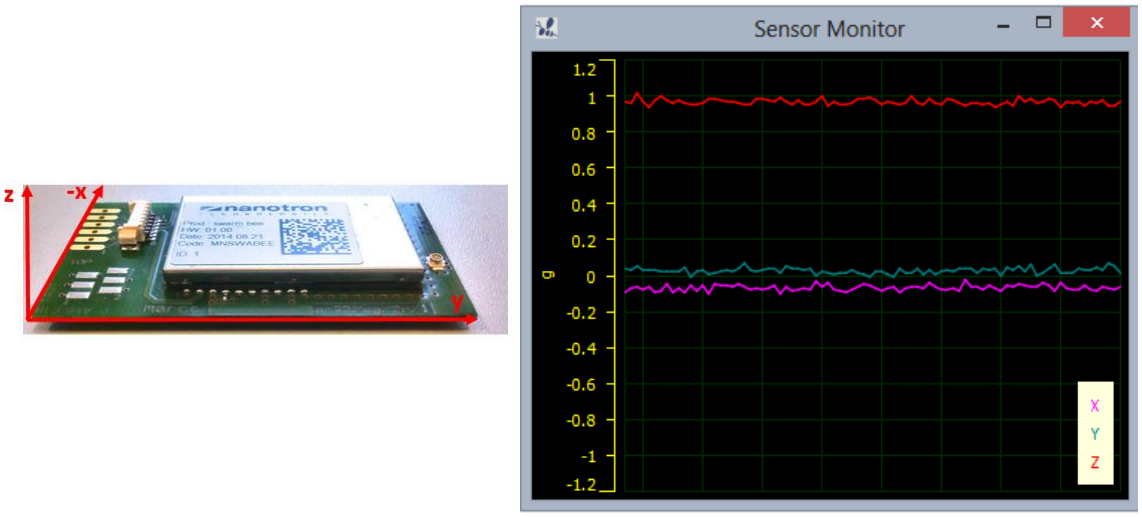

Figure 3-1: 0 g acceleration in x and y coordinates and possitive in the z coordinate

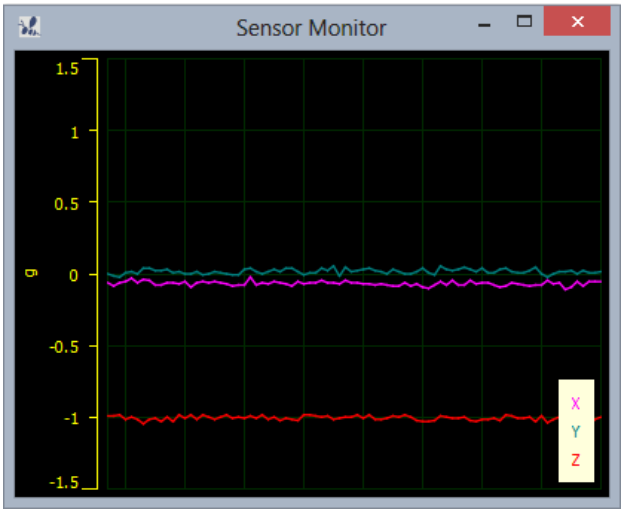

Figure 3-2: 0 g acceleration in x and y coordinates and possitive in the z coordinate

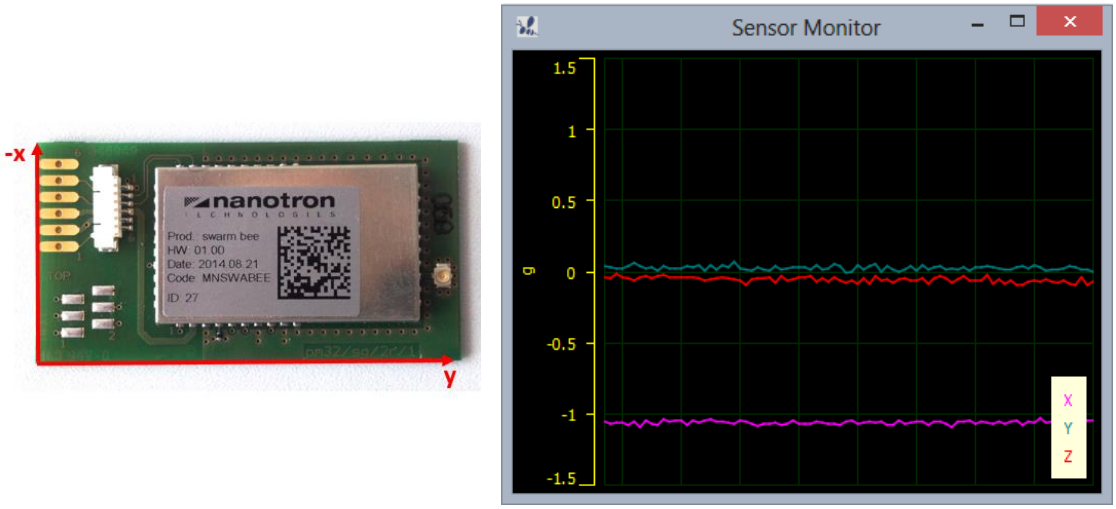

Figure 3-3: 0 g acceleration in y and z coordinates and negative in the x coordinate

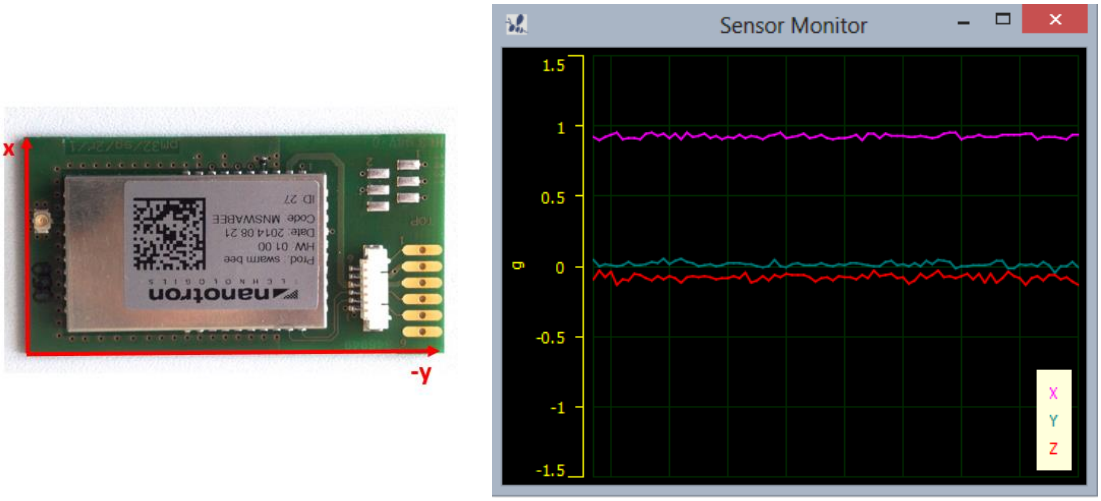

Figure 3-4: 0 g acceleration in y and z coordinates and possitive in the x coordinate

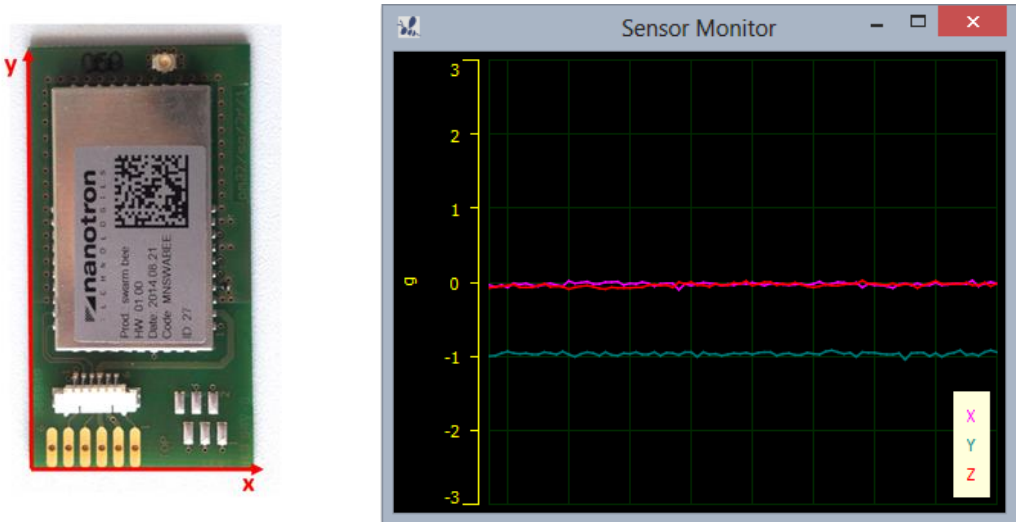

Figure 3-5: 0 g acceleration in x and z coordinates and negative in the y coordinate

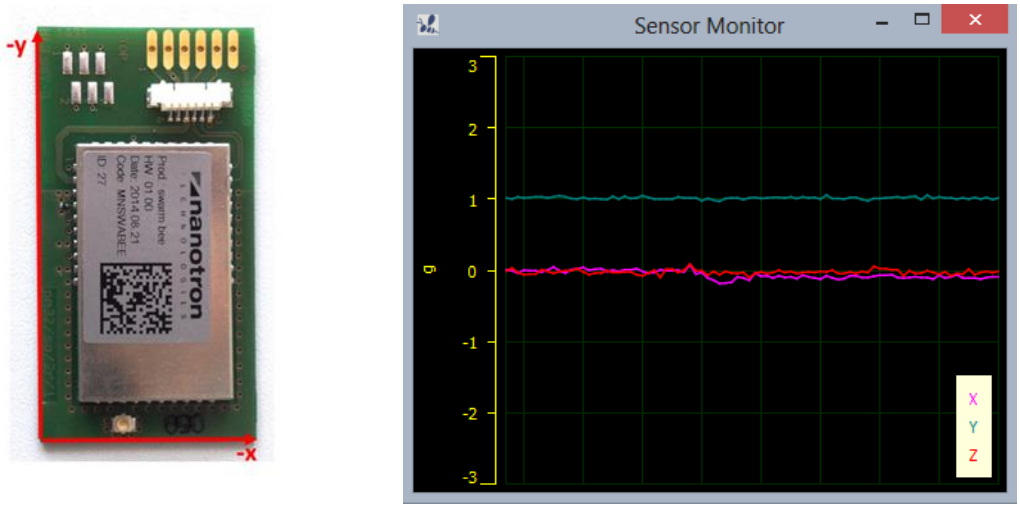

Figure 3-6: 0 g acceleration in x and z coordinates and possitive in the y coordinate

3.2. Temperature data

The temperature data is given in °C and it consist on an hexadecimal value of width 3 bytes. The slope of the temperature sensor is 0.5K/LSB, and its center temperature is 24°C. Therefore, the temperature measurement range is -40°C up to 87.5°C.

4. Sensor initialisation through the API

4.1. API commands

The swarm API [2] provides with a set of commands that allow to initialise and set some parameters of the MEMS sensor.

EMSS <Enable> Enables or disables the MEMS sensor

If <Enable> = 0 the sensor is in suspend mode.

If <Enable> = 1 the sensor is enabled, in normal power mode.

SMRA <Grange> Sets MEMS range, i.e. +/- 2, 4, 8 or 16g.

<Grange> = 1 MEMS g range set to +/- 2g

<Grange> = 2 MEMS g range set to +/- 4g

<Grange> = 3 MEMS g range set to +/- 8g

<Grange> = 4 MEMS g range set to +/- 16g

SMTH <Thershold> Sets the MEMS threshold for the slope interrupt.

The parameter <Thres> is not the threshold itself but a parameter that will be used to estimate the actual threshold. The following formula is used:

if <Grange> = 1 → Threshold = 3.91 mg/LSB * <Thres>

if <Grange> = 2 → Threshold = 7.81 mg/LSB * <Thres>

if <Grange> = 3 → Threshold = 15.62 mg/LSB * <Thres>

if <Grange> = 4 → Threshold = 31.25 mg/LSB * <Thres>

SMBW <BandWidth> Sets MEMs' filter bandwidth of the MEMS.

<BandWidth> = 1 → 7.81 Hz
<BandWidth> = 2 → 15.63 Hz
<BandWidth> = 3 → 31.25 Hz
<BandWidth> = 4 → 62.50 Hz
<BandWidth> = 5 → 125.00 Hz
<BandWidth> = 6 → 250.00 Hz
<BandWidth> = 7 → 500.00 Hz
<BandWidth> = 8 → 1000.00 Hz

SMSL <SleepTime> Sets MEMS sleep time, from0.5 up to 1000ms

<SleepTime> = 1 → 0.5 ms
<SleepTime> = 2 → 1 ms
<SleepTime> = 3 → 2 ms
<SleepTime> = 4 → 4 ms
<SleepTime> = 5 → 6 ms
<SleepTime> = 6 → 10 ms
<SleepTime> = 7 → 25 ms
<SleepTime> = 8 → 50 ms
<SleepTime> = 9 → 100 ms
<SleepTime> = 10 → 500 ms
<SleepTime> = 11 → 1000 ms

During initialization the sensor is set in sleep mode; when this time finishes, the sensor wakes up to check the value of the registers and pass the temperature and acceleration values to the swarm. Then it goes to sleep again.

SMDT Sets MEMS deadtime. This is the minimum time between two possible interrupts.

EBMS <Enable> Enables/disables broadcast MEMS within node ID blink packets.

<Enable> = 0 disabled
<Enable> = 1 enabled

Further details on how to use this commands can be found in [2].

4.2. Default settings

When a swarm is initialised the MEMs is also initialize and enabled. The default values are the following:

MEMS is enabled (EMSS = 1)

Broadcast MEMs data is enabled. (EBMS = 1)

MEMS bandwidth: 250Hz (SMBW = 6)

MEMS dead time: 1000ms (SMDT)

MEMS sleep time: 0.5ms (SMSL = 1)

MEMS g-range: 4g (SMRA = 2)

MEMS threshold: 234.3 mg/LSB (SMTH = 30 and SMRA = 2)

5. How to read out the sensor's data

Every ***swarm*** node is capable of reading its own sensors' data and sending it as payload in its node ID blink. To read out its own sensors they only need to be initialised using the command. To send that data as part of the payload this options should also be enabled (EMSS = 1).
The sensors' data included in the payload can be seen both when the swarm is communicating with other swarms and when it is acting as a tag in a real time location system (RTLS).

Next subchapters describe more in detail the different way sensors' data can be read out.

5.1. Reading out sensor data from the swarm host

5.1.1. Reading sensor data from the own swarm

The API implements two commands to read the data of the sensors integrated in the swarm node connected to the host:

GMYA: gets the acceleration in the three axes.

The return value, in mg, is: =<XACC>,<YACC>,<ZACC>

GMYT: gets the temperature.

The return value, in °C, is: =<TEMPERATURE>

Example:

```
GMYA
=+304,-7,+984

GMYT
=+26
```

Further details on how to use this commands can be found in [2].

5.1.2. Reading sensor information from another swarm radio

When a ***swarm*** radio has the option 'broadcast MEMS within node ID blink' enabled (EBMS = 1), it adds the acceleration and temperature data as payload in every blink it transmits. Other swarms receiving that information will be able of interpreting the payload and pass it to the host. This actions in the receiving swarm is, however, not authomatic. To enable it there are two commands available in the API: EIDN and NCFG.
First the 'node ID notification' should be enabled (EIDN=1). When this is done the swarm sends a notification message every time a new blink is received. This is called 'Node ID notification' (NIN).
Second the Node ID notification mask should be set. swarms can transmit several data in the payload; thus the mask allows the swarm to only pass to the host those pieces of information that are necessary. The mask consist on 10 bits, starting from bit 0 up to bit 9. Setting one bit to 1 means that the data corresponding to that bit will be displayed in the INI. Bit 2 corresponds to the acceleration data and bit 4 to temperature data. It is important to know what mask is being used because the order in which the values are displayed folows the bit order: first bit 0 and last bit 9.
swarm radios in the neigbourhood will receive the blinks with the sensor data. If they are connected to a host and have the 'node ID notification node' option active (EIDN = 1), will send a node ID notification to their host.

Example1:
In this example we want to display only acceleration and temperature data. The mask should then be NCFG = 1010, if we pass it to hexadecimal numbers it is NCFG = A.

```
EIDN 1
=1
NCFG A
=00A
```

The notification received will be:

```
*NIN:00000DCAB039,-39,-46,+937,+28
```

First the acceleration data is given in the order xcc, yacc and zacc and then the temperature.

Example2:
In this example we want to display acceleration, temperature data and other data. The mask should then be NCFG = 11111, if we pass it to hexadecimal numbers it is NCFG = 1F.

```
EIDN 1
=1
NCFG 1F
=01F
```

The notification received will be:

```
*NIN:00000DCAB039,1,+0,-46,+960,-71,+28,0
```
Nodi ID bit 0 bit 1 bit2 bit 3 bit4

The node with ID 00000DCAB039 sent a blink indicating 28°C of temperature and x,y,z acceleration values equal to 0g,-46 mg and 960mg.

5.2. Reading out the sensor data with the RTLS

The use of a swarm as part of a RTLS is based in the fact that the swarm is periodically broadcasting a node ID blink. Nanotron Anchors are capable of detecting that blink and pass it to nanoLES so that it can estimate the absolute position of the swarm. For each blink received, NanoLES gives the estimated position together with data. Among this data, nanoLES includes also the blink payload from the MEMS sensor. As in the previous case, the sensor data is enabled in the blink payload when the option 'broadcast MEMS within node ID blink' is enabled (EBMS = 1). Figure 5-1 shows the nanoLES location data format.

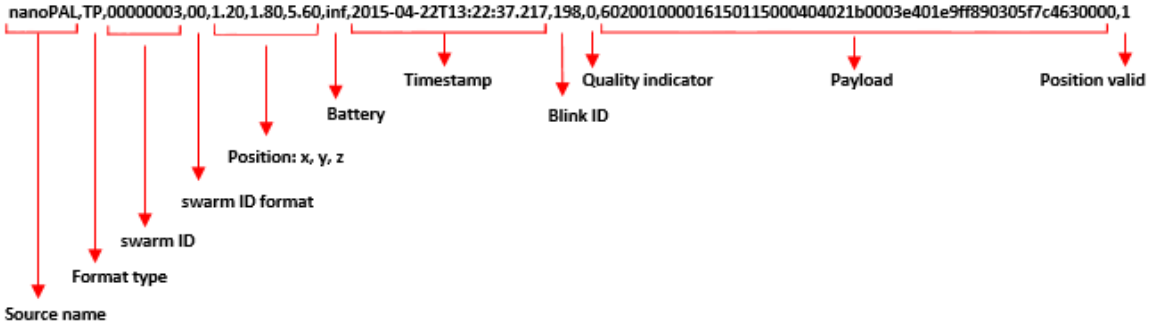

Figure 5-1: swarm location data as output of nanoLES

Each two digit represent 1byte in hexadecimal value. The first 6 bytes represents
SWARM_TYPE – 1 byte. The swarm time is 60
SWARM_VERSION – 1 byte
SWARM_DC – 1 byte
SWARM_SPSA – 1 byte. Power mode of the board
SWARM_WAKEUP_REASON – 1 byte
SWARM_LENGTH – 1 byte. Length, in bytes, of the payload starting after this byte.
SWARM_SENSOR_DATA_LENGTH – 1 byte. Length, in bytes, of the sensors data.

After the sensor data length is stated the actual sensor's data starts. The first byte will be Sensor data type indicatin what kind of data we have inmediatelly after. The possible types are:
Sensor type: 0x00 ESC, indicates the end of the sensors' data.
Sensor type: 0x01 battery status/* 2 byte battery status */
Sensor type: 0x02 temperature
Sensor type: 0x03 acceleration data/* 6 byte mems data */
Sensor type: 0x04 GPIO configuration/* 1 byte gpio pin state */

The sensor data comes inmediatelly after the corresponding sensor type. The length of this fields varies depending on the sensor type and it is represented in two's complement:

Battery: S_DATA_BATTERY_0

S_DATA_BATTERY_1

GPIO: S_TYPE_GPIO
S_DATA_GPIO

Temperature: S_TYPE_TEMP
S_DATA_TEMP_0
S_DATA_TEMP_1

Acceleration: S_TYPE_MEMS
S_DATA_MEMS_X_0
S_DATA_MEMS_X_1
S_DATA_MEMS_Y_0
S_DATA_MEMS_Y_1
S_DATA_MEMS_Z_0
S_DATA_MEMS_Z_1

Example
In Figure 5-1 the payload is:
`602001000016150115000404021b0003e401e9ff890305f7c4630000`

It should be interpretes as follows:

```
60 Type(60=swarm)
20 swarm version
01 swarm class
00 swarm power mode
00 wake-up reason
11 payload length in bytes after this byte (21 bytes)
10 sensar data length in bytes (20 bytes)
01 data type → 01:battery, battery data is given in 2 bytes
1500
04 data type → 04:GPIO, the GPIO configuration is described with 1 byte
04 the setting of the GPIO is 00000100
02 data type → 02:temperature, it is given in 2 bytes
1b
00 temperature equal to 0x001b → 27°C
03 data type → 03:acceleration data, it is give in 2 bytes for each direction
e4
01 x acceleration 0x01e4 → 0000000111100100 (two's complement) → 484mg
e9
ff y acceleration 0xffe9 → 1111111111101001 (two's complement) → -23mg
89
03 z acceleration 0x0389 → 0000001110001001 (two's complement) → 905mg
05 data type → 05:timestamp, 4 bytes LSB
f7
c4
63
00 timestamp 0x0063c4f7 → 6538487 ms
00 data type → ESC, indicates the end of the sensors data
```

In case some other data (not sensor data) is transmitted in the payload it would go after the "ESC" indicator.

References

[1] BMA250 Data Sheet, BTS- BMA250-DS002-04. Bosch SensorTec GmbH, 1 November 2011
[2] Swarm API 2.1, NA-13-0267-0003-2.1 Nanotron Technologies GmbH, 16 Mai 2015

Document History

Date	Author	Version	Description

Life Support Policy

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nanotron Technologies GmbH customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify nanotron Technologies GmbH for any damages resulting from such improper use or sale.

About Nanotron Technologies GmbH

Today nanotron's *embedded location platform* delivers location-awareness for safety and productivity solutions across industrial and consumer markets. The platform consists of chips, modules and software that enable precise real-time positioning and concurrent wireless communication. The ubiquitous proliferation of interoperable location platforms is creating the location-aware Internet of Things.

Further Information

For more information about products from nanotron Technologies GmbH, contact a sales representative at the following address:

nanotron Technologies GmbH
Alt-Moabit 60
10555 Berlin, Germany
Phone: +49 30 399 954 – 0
Fax: +49 30 399 954 – 188
Email: sales@nanotron.com
Internet: www.nanotron.com