AN5195

Application note

Power profiling of the FP-SNS-ALLMEMS2 function pack

Introduction

This document describes how to carry out power measurement on FP-SNS-ALLMEMS2 function pack and check its power improvement over FP-SNS-ALLMEMS1.

FP-SNS-ALLMEMS2 is a firmware evolution of FP-SNS-ALLMEMS1 function pack that features FreeRTOS support and ultra-low power optimization which, for example, extends the SensorTile battery life.

Although FP-SNS-ALLMEMS1 and FP-SNS-ALLMEMS2 support different hardware platforms, only the NUCLEO-L476RG development board has been taken into consideration for this study, mainly for STM32L4 power efficiency.

1 Hardware and software requirements

To carry out power measurement on FP-SNS-ALLMEMS2, you need the following resources:

- Hardware
  - a NUCLEO-L476RG development board
  - an X-NUCLEO-IDB05A1 BLE expansion board
  - an X-NUCLEO-IKS01A2 expansion board for MEMS sensor devices (HTS221, LSP22HB, LSM6DSL, LSM303AGR)
  - an X-NUCLEO-CCA02M1 digital MEMS microphone expansion board
  - an X-NUCLEO-LPM01A expansion board for power consumption measurement
- Software
  - FP-SNS-ALLMEMS1 3.4.0
  - FP-SNS-ALLMEMS2 1.0.0
- Tools
  - IAR V8.20.2
  - STM32 ST-LINK utility
  - ST-LINK, ST-LINK/V2, ST-LINK/V2-1 USB driver signed for Windows7, Windows8, Windows10
  - STM32CubeMonPwr

2 Hardware configuration

Step 1. Configure the NUCLEO-L476RG as follows:
- remove JP5 (PWR)
- open SB2

Step 2. Configure the X-NUCLEO-CCA02M1 as follows (refer to FP-SNS-ALLMEMS1.chm or FP-SNS-ALLMEMS2.chm):
- close the SB12, SB16 solder bridges
- open the solder bridges SB7, SB15 and SB17

Step 3. Configure the X-NUCLEO-LPM01A as follows (refer to UM2243 on www.st.com):
- select the ARD pin on the Arduino nano connector as power output
- close JP9 AREF_ARD jumper
- open JP10 3V3_ARD jumper
- close JP4 additional decoupling capacitor jumper

Step 4. Stack the boards as follows (bottom-up):

a. NUCLEO-L476RG

b. X-NUCLEO-CCA02M1

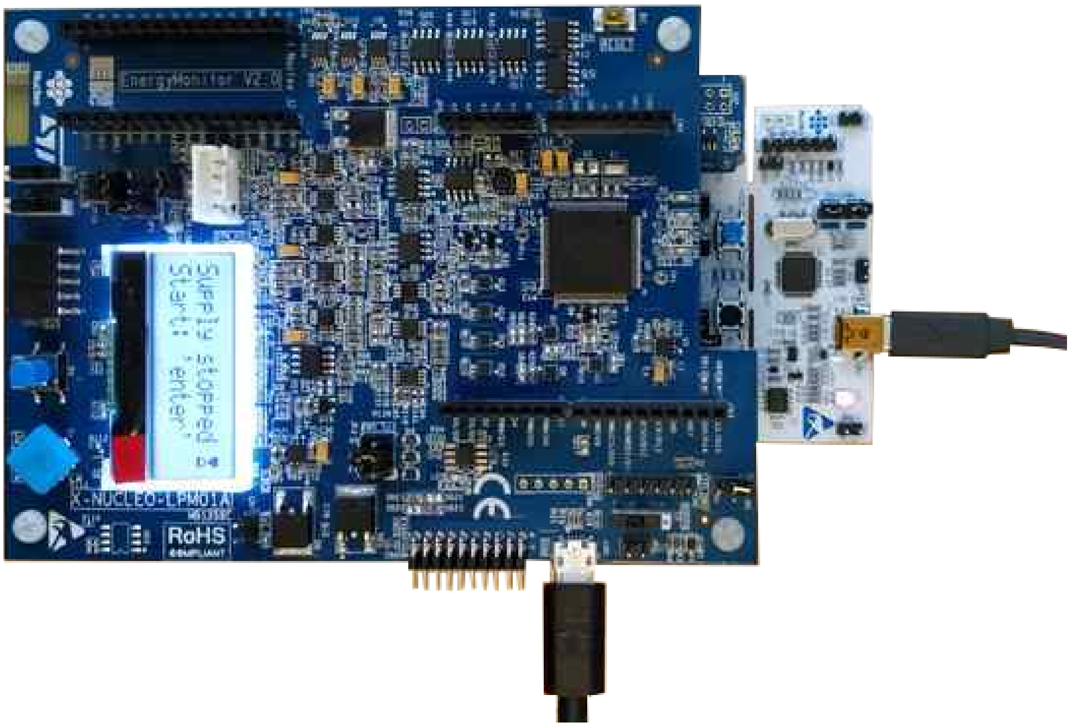
c. X-NUCLEO-IDB05A1

d. X-NUCLEO-IKS01A2

e. X-NUCLEO-LPM01A

Step 5. Connect the STM32 Nucleo (NUCLEO-L476RG) and the power shield (X-NUCLEO-LPM01A) to a PC USB port

Figure 1. NUCLEO-L476RG and X-NUCLEO-LPM01A PC connection



3 How to flash the STM32 Nucleo board

3.1 FP-SNS-ALLMEMS1 firmware

Step 1. Download the FP-SNS-ALLMEMS1 firmware.

Step 2. Power the STM32 Nucleo board by pressing enter on the power shield.

Step 3. Start the STM32 ST-LINK utility.

Step 4. Connect the Nucleo target [**Target**]>[**Connect**])

Step 5. Program and verify (CTRL+P) with the file STM32CubeFunctionPack_ALLMEMS1_V3.4.0\Projects \Multi\Applications\ALLMEMS1\Binary\STM32L476RG-Nucleo\ ALLMEMS1_IKS01A2_NucleoL476_BL.bin

Step 6. Click [**Start**] on the pop-up window.

The power shield gives approximately 14 mA at 3 V.

3.2 FP-SNS-ALLMEMS2 firmware

Step 1. Download the FP-SNS-ALLMEMS2 firmware.

Step 2. Power the STM32 Nucleo board by pressing enter on the power shield.

Step 3. Start the STM32 ST-LINK utility.

Step 4. Connect the Nucleo target [**Target**]>[**Connect**])

Step 5. Program and verify (CTRL+P) with the file STM32CubeFunctionPack_ALLMEMS2_V1.0.0\Projects \Multi\Applications\ALLMEMS2\Binary\STM32L476RG-Nucleo\ ALLMEMS2_NucleoL476.bin

Step 6. Click [**Start**] on the pop-up window.

The power shield gives approximately 4 mA at 3 V.

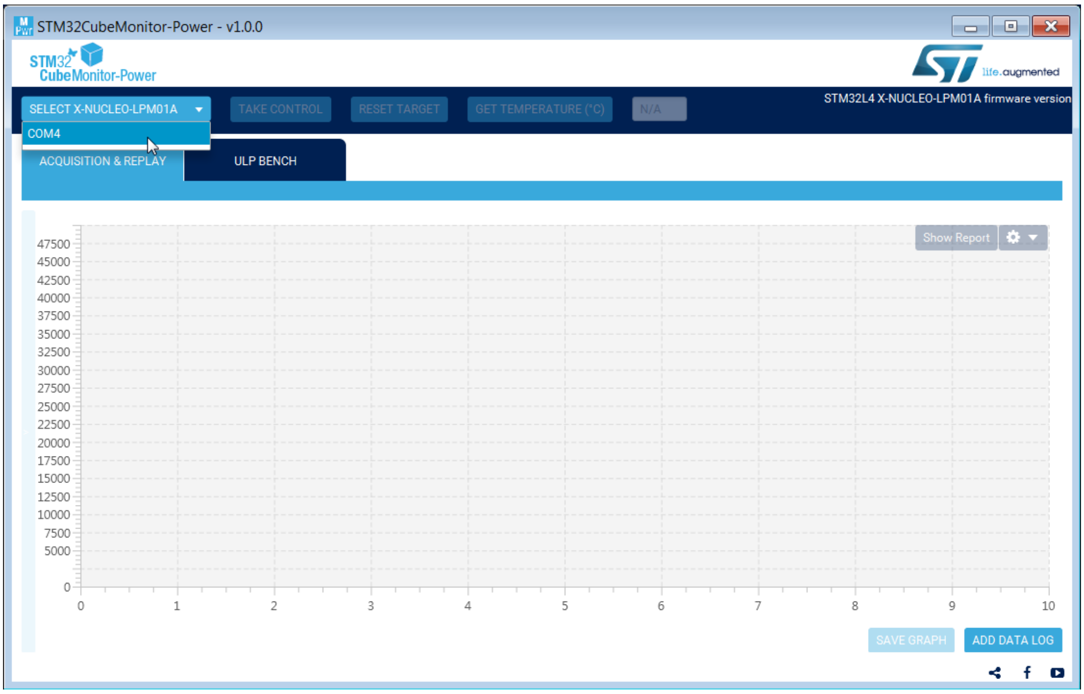
4 Using the power shield for dynamic power profiling

Step 1. Download STM32CubeMonPwr and install it on your PC.

Step 2. Launch the application.

Step 3. Select the COM port where the power shield is connected.

Figure 2. **STM32CubeMonPwr: COM port selection**



Step 4. Click on [**Show Report**].

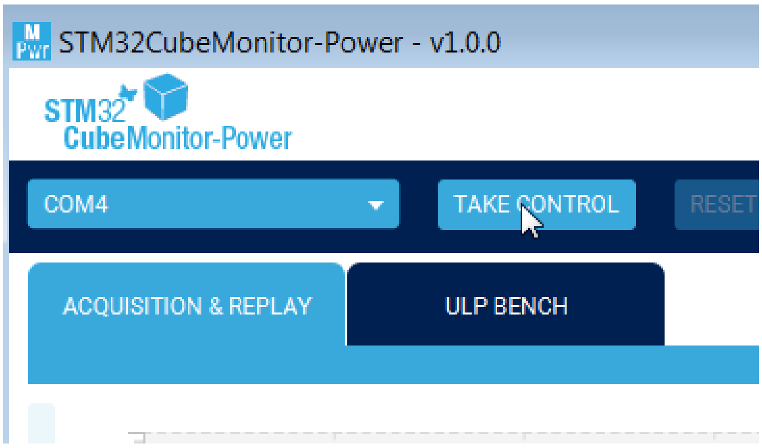
Figure 3. **STM32CubeMonPwr: show report**

The following report frame appears:

Figure 4. STM32CubeMonPwr: report frame

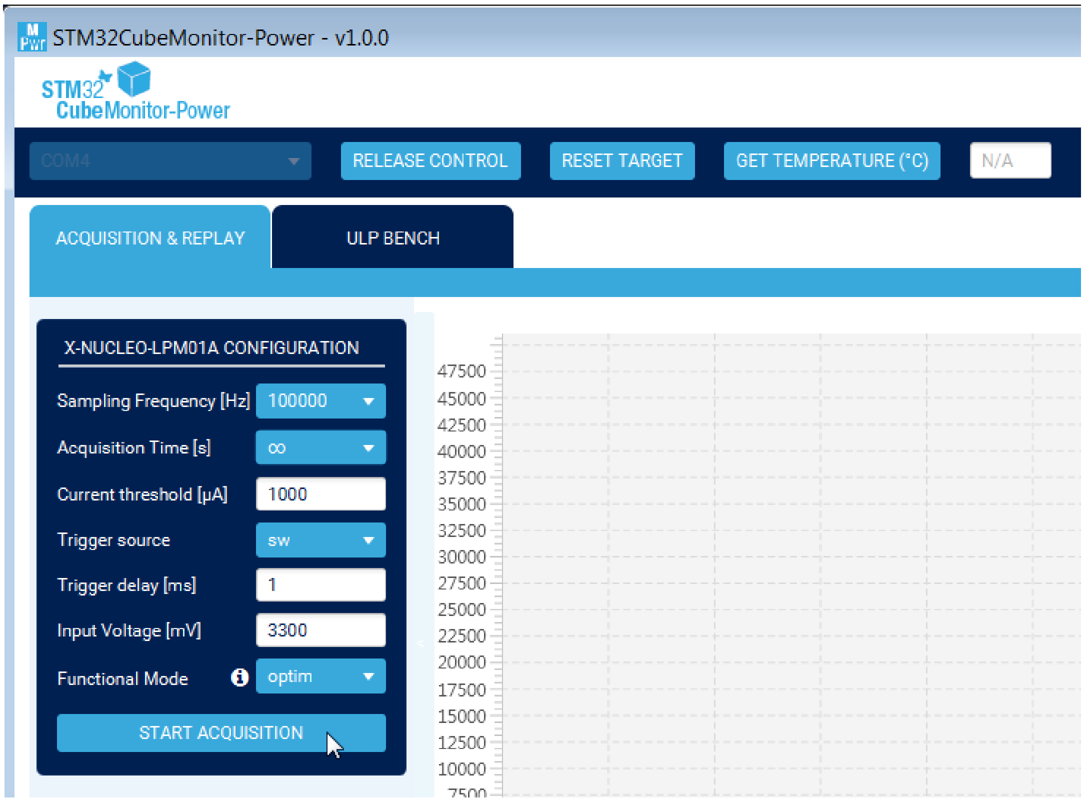
Step 5. Click on [**Take control**]

Figure 5. STM32CubeMonPwr: control



Step 6. Configure the power shield using the following settings, overriding the default ones:

Figure 6. STM32CubeMonPwr: acquisition settings



Step 7. Click on [**Start acquisition**] and [**Stop acquisition**] anytime

Figure 7. STM32CubeMonPwr: first plot

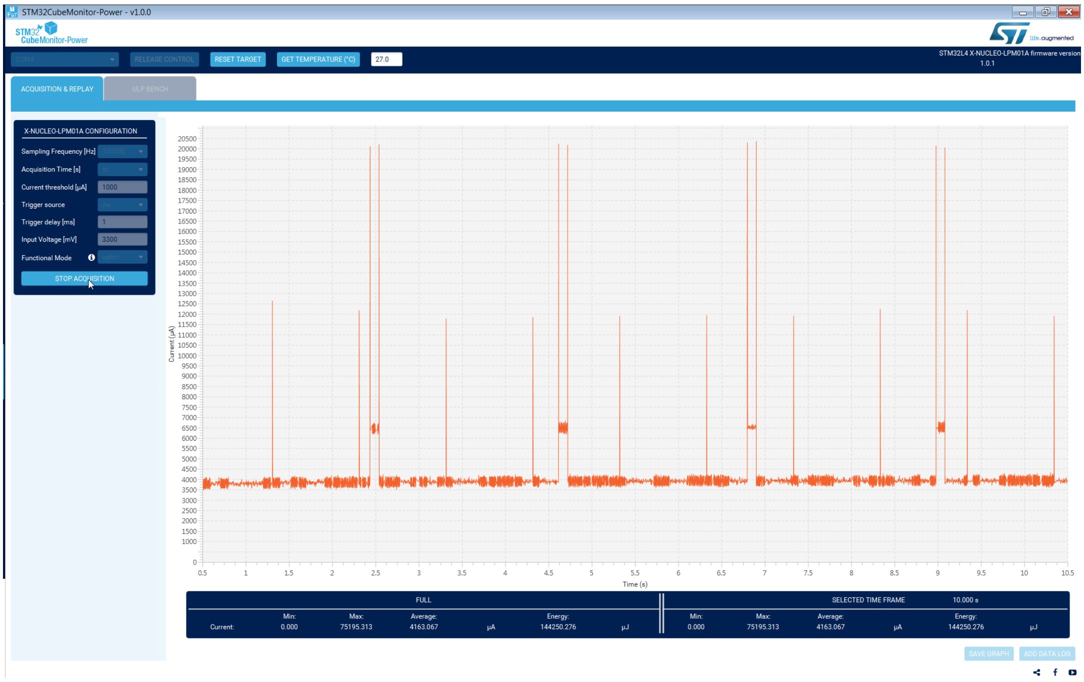
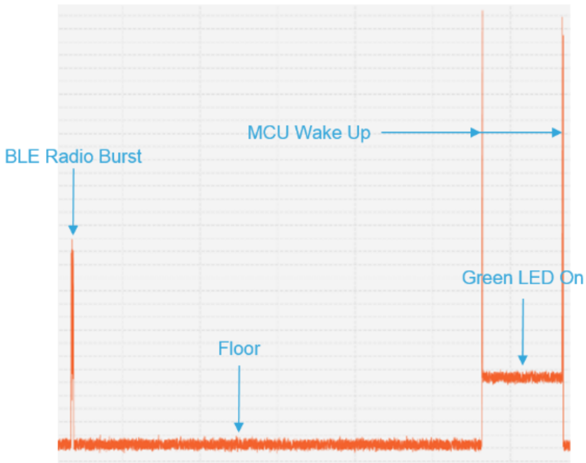


Figure 8. STM32CubeMonPwr: first plot (zoom)



5 Current floor breakdown

The measured current floor is 3.9 mA (the minimum current that can be measured in this configuration).

In this hardware setup, two voltage domains are at stake:

- 3V3 for MCU, audio and Bluetooth
- 1V8 for MEMS sensors

The voltage conversion is necessary through LDO and level shifters.

A more product-oriented design usually avoids this overhead and aligns all voltages to 1V8 to save voltage conversion loss and consume less power with a lower voltage.

Likewise, ST-LINK circuitry on the STM32 Nucleo, which allows download and debug, also adds power overhead.

The measurements have been performed on:

- NUCLEO-L476RG: 0.8 mA (ST-LINK circuitry)
- X-NUCLEO-IKS01A2: 3 mA (voltage conversion circuitry)

Important:
The result is systematically 3.8 mA applying to all the setup, as a fair relative comparison among the options taken into account in this application note.

6 Use cases

The ST BlueMS application available for Android and iOS shows the data exported by a BLE device using the BlueST protocol (for example, FP-SNS-ALLMEMS1 STM32Cube function pack).

The app shows different panels based on the data types exported by the firmware, including: environmental data, MEMS sensor fusion, plot, activity recognition, carry position, acceleration event, BlueVoice, pedometer, switch, motion intensity, compass, cloud logging, node status.

A sequence of use cases has been defined from the power on to the last application panel (grossly, one application panel is a use case).

The following sections build a sequence chosen to reflect the variety of execution models available.

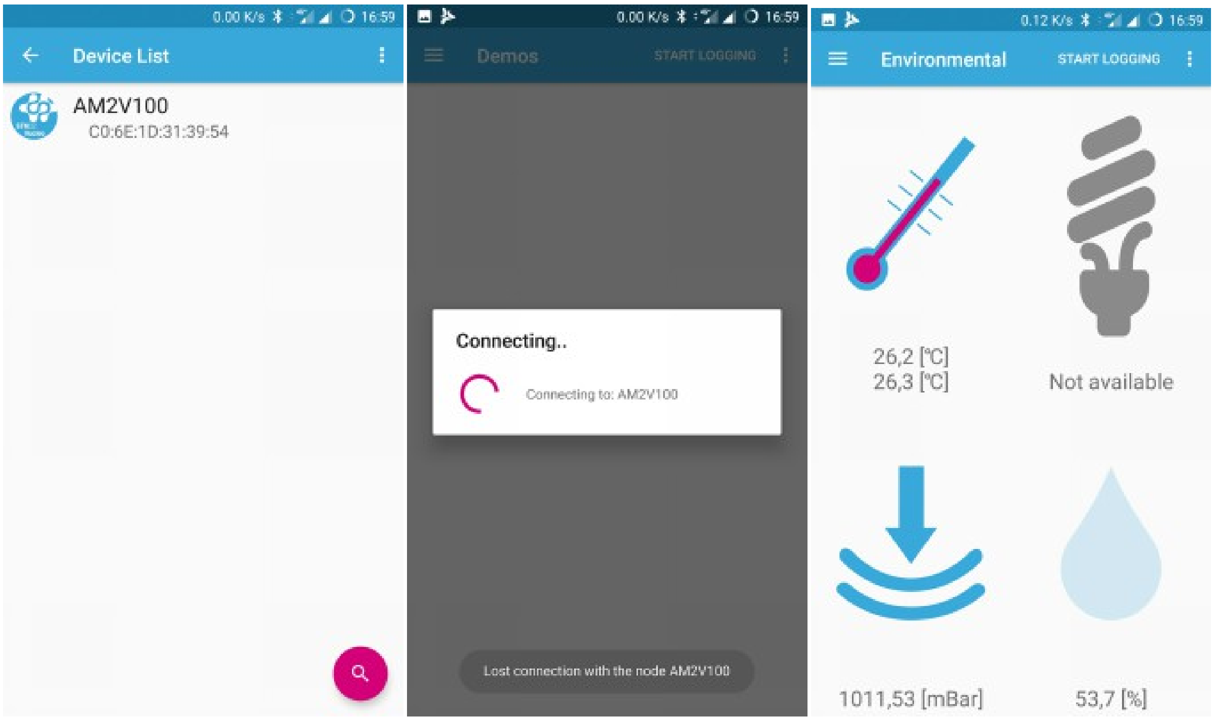
6.1 Connectable

The node waits for the Android host to connect and advertise itself on air (BLE).

Figure 9. BlueMS: scanning frame

The next use case needs a BLE connection to the host. For this reason, you must tap on **Start scanning** to look for nodes. After selecting the node, the device is connected in few seconds.

Figure 10. BlueMS: connection to device



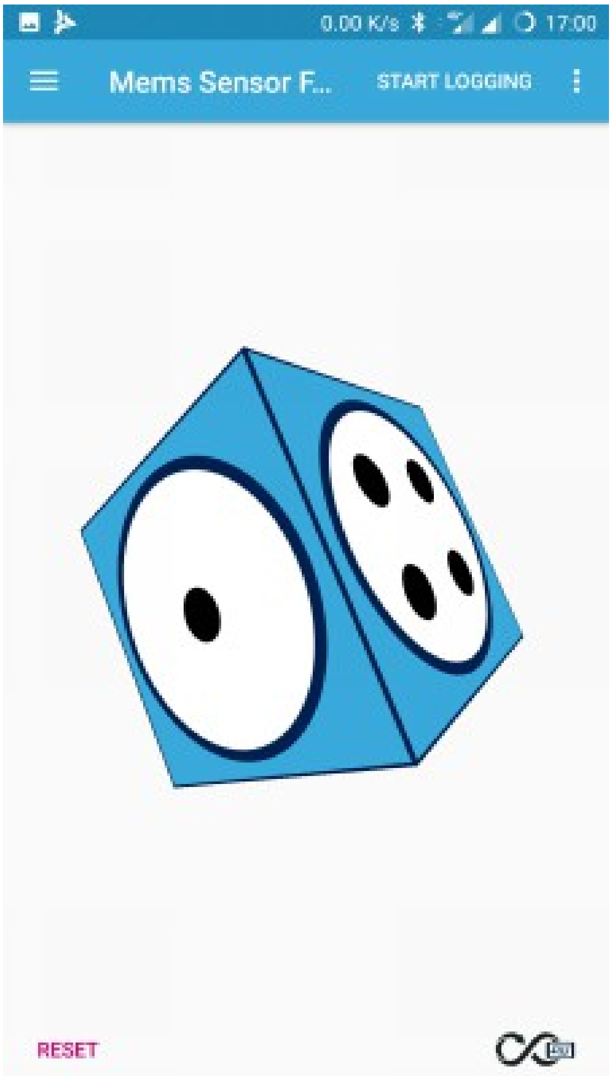
6.2 Environmental

The node regularly polls for environmental parameters (temperature, pressure and humidity) and reports them to the host application via BLE.

6.3 Sensor fusion

Data is collected from various motion sensors (accelerometer, magnetometer and gyroscope) and processed altogether (fusion). The application accurately builds an avatar of the device.

Figure 11. BlueMS: sensor fusion panel



6.4 Plot sensor data

The application plots the selected data (acceleration or microphone data) in graphic form in real-time.

Figure 12. **BlueMS: plot accelerometer panel**

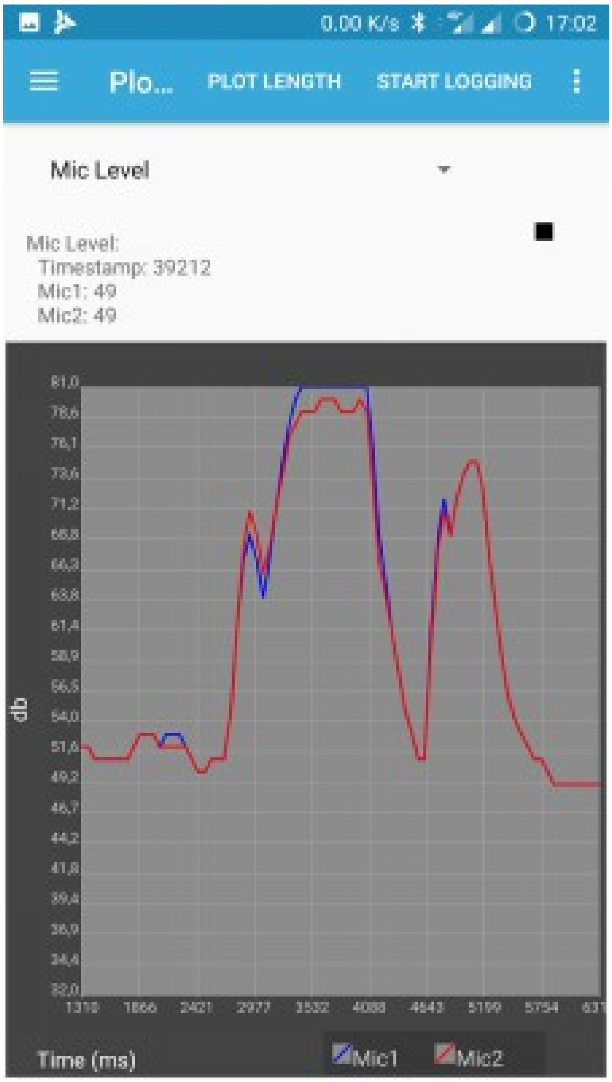
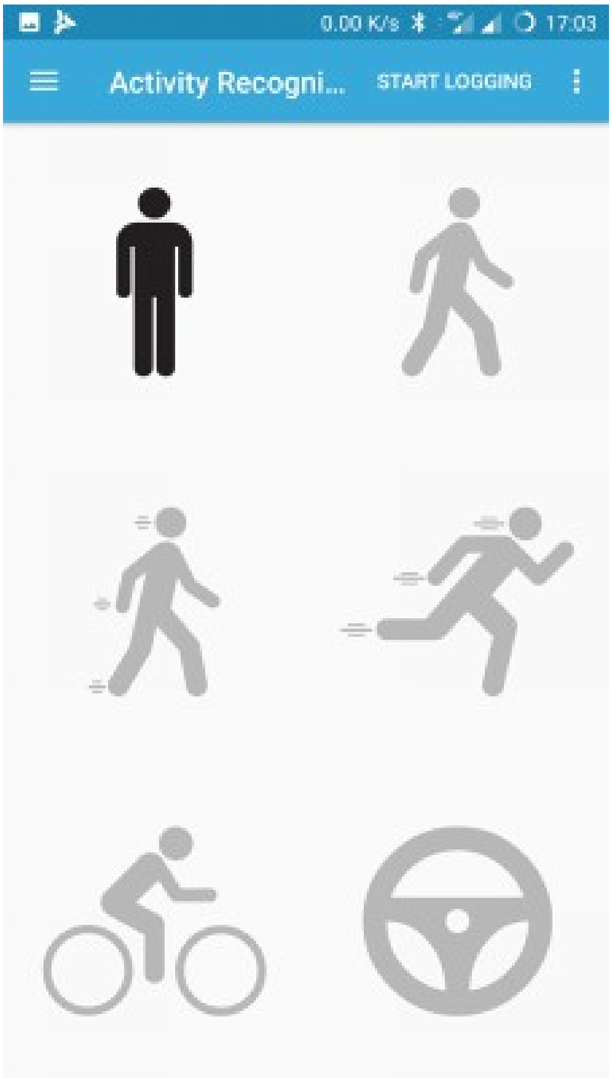


Figure 13. BlueMS: plot microphone level panel

6.5 Activity recognition

According to sensed data, the node displays one activity among a set of six (stand up, walking, running, etc.).

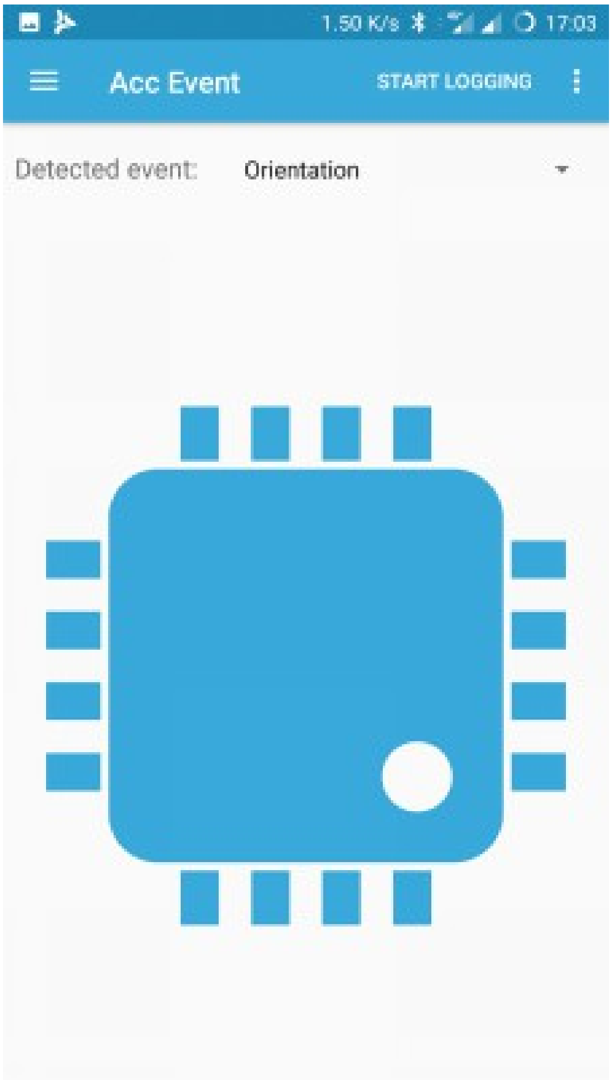
Figure 14. **BlueMS: activity recognition panel**



6.6 Hardware event detection

The node reports events from the sensors.

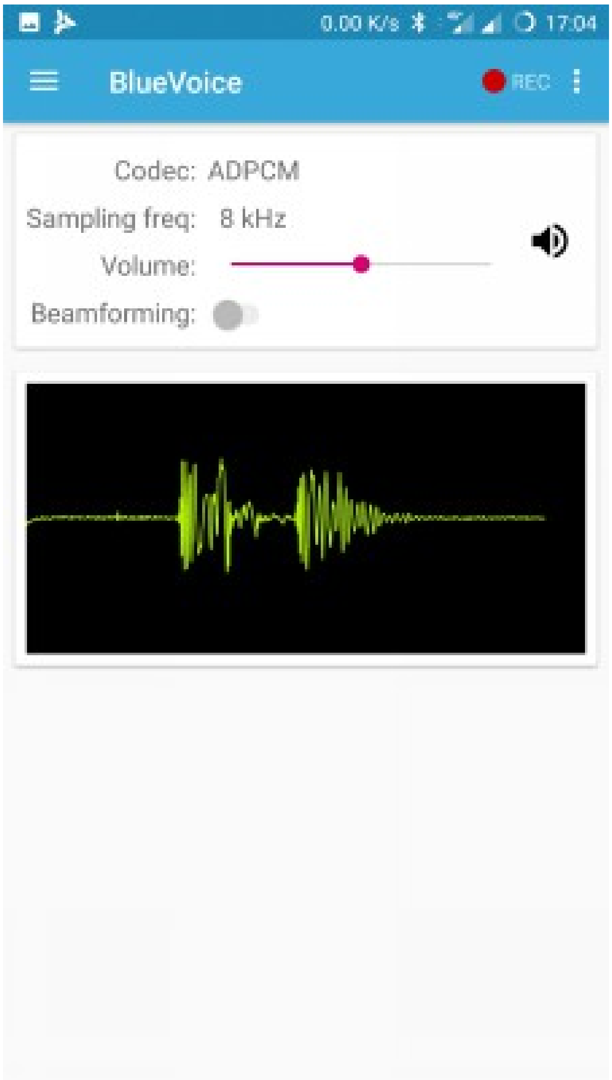
Figure 15. BlueMS hardware event detection panel



6.7 BlueVoice

The node transmits audio samples on a BLE link.

Figure 16. BlueMS BlueVoice panel



6.8 Compass

The node indicates the north.

Figure 17. BlueMS compass panel

6.9 Node monitor

The node indicates the Rssi and the battery level when available (only for Sensortile and BlueCoin).

Figure 18. **BlueMS Rssi and battery panel**

7 Dynamic power profiling

For both FP-SNS-ALLMEMS1 and FP-SNS-ALLMEMS2 firmware packages, a dynamic profiling is carried out through the following use case sequence:

1. connectable
2. environmental
3. sensor fusion
4. plot acceleration
5. plot microphone level
6. activity recognition
7. hardware event detection
8. BlueVoice
9. compass
10. node monitor

An experimental version of FP-SNS-ALLMEMS2 (which adapts the BLE link at the expense of a degraded transition time between use cases), called FP-SNS-ALLMEMS2+, can be generated via the following function `#define BLE_LINK_ADAPT` in targetFeature.h where you can rebuild the project.

8 Connectable use case

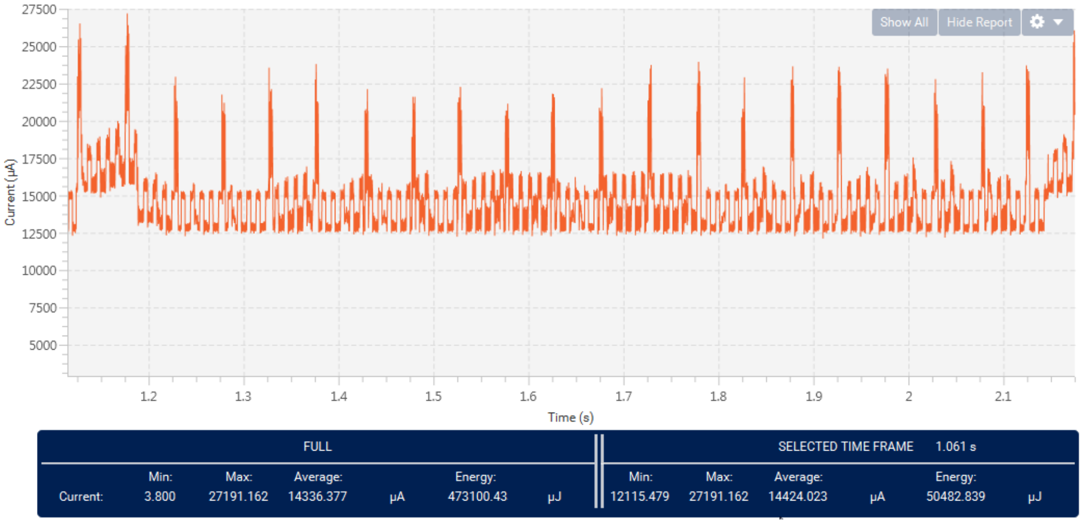
The node awaits a Bluetooth connection with a host and advertises itself on the air.

In the FP-SNS-ALLMEMS1 version, all peripherals are left on even if they are not part of the use case.

In this configuration, the MCU is in WFI state and wakes up every second with the LED blinking for 200 ms.

The plot shows this activity and the spikes occurring every 50 ms corresponding to the BLE radio burst for advertising, together with a 10 ms periodic activity of motion sensing.

Figure 19. FP-SNS-ALLMEMS1 connectable plot

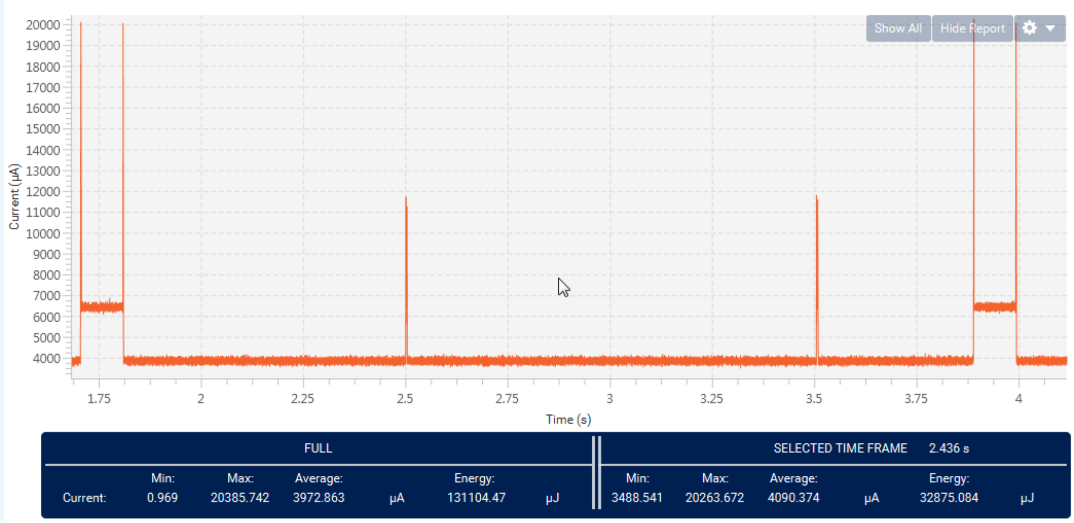


By zooming the plot, you can see the 1 ms tick timer spikes overlap and the MCU wakes up only to increment a counter (this can be optimized as shown further on).

Figure 20. FP-SNS-ALLMEMS1 connectable plot (zoom)

In the FP-SNS-ALLMEMS2 connectable plot, the current waveform is much more readable, all the activities that constitute the use case can be well identified.

Figure 21. FP-SNS-ALLMEMS2 connectable plot



Every two seconds (instead of 1 second in the previous case), the MCU wakes up from stop mode (instead of WFI in the previous case) to put the LED on and then goes back to stop mode again. The MCU wakes up again after 100 ms (instead of 200 ms in the previous case). On the graph, you can see a step of 2500 µA corresponding to LED current draw.

Note: The BLE radio burst is the same as the original version, but the duration has been extended 20 times, from 50 ms to 1000 ms; as a result, a 1-second latency has been added for connecting.

Taking into account a 3800 µA floor current as discussed earlier, the average current drops from 10536 µA to 172 µA gaining 98%.

This gain has been achieved by powering down unused hardware components, exploiting STM32L4 lower power mode, adjusting connection parameters and LED duty cycle and suppressing system tick.

This use case can last very long as a gateway to all others, that is the user can forget the device in its pocket 20 times longer than the original version.

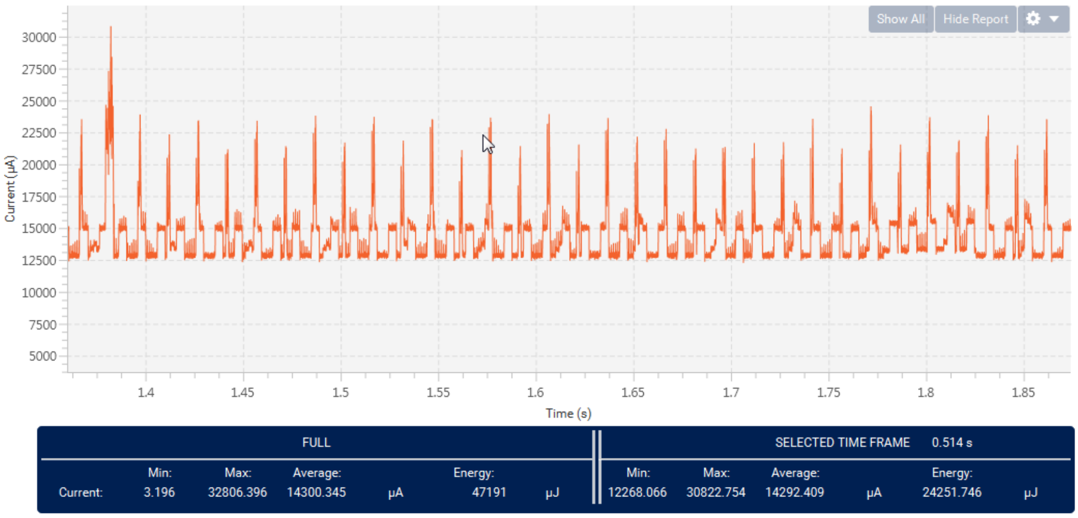
Note: All the observations for the FP-SNS-ALLMEMS2 also apply to FP-SNS-ALLMEMS2+.

9 Environmental use case

This use case reports environmental sensor data to the host.

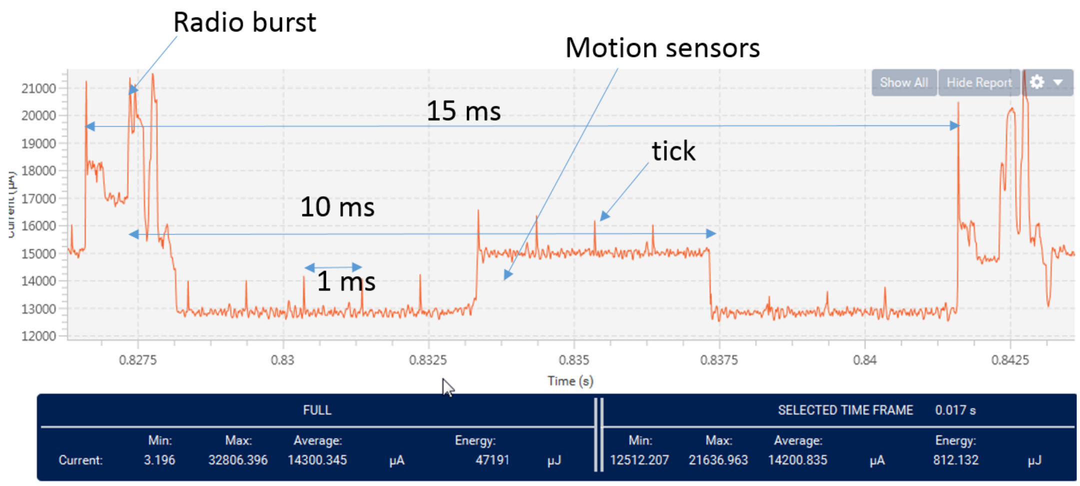
For the FP-SNS-ALLMEMS1, the current floor is on 12500 µA, encompassing the 3800 µA, MCU in WFI (most of the time) BLE and all sensors.

Figure 22. **FP-SNS-ALLMEMS1 environmental plot**



The estimated time is 10 ms periodic for motion sensor activity, 1 ms for system tick, 15 ms for BLE reporting (see the figure below for details).

Figure 23. **FP-SNS-ALLMEMS1 environmental plot (zoom)**

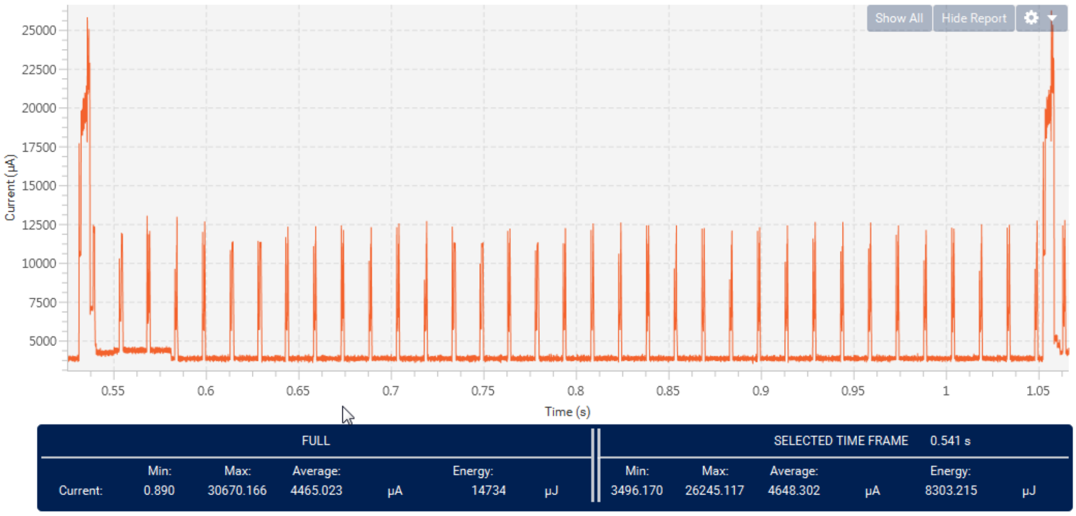


In the FP-SNS-ALLMEMS2, all motion sensors have been powered down, tick suppressed and MCU lower power mode used, so only environmental and BLE activity is left.

Using the single-shot mode for sensors, the current floor is greatly lowered to a bare minimum, with superimposition of the environmental waveform (big spikes every 500 ms) and the BLE radio burst at 15 ms periodic.

The average current drops from 10500 µA to 665 µA, so a 94 % gain at system level.

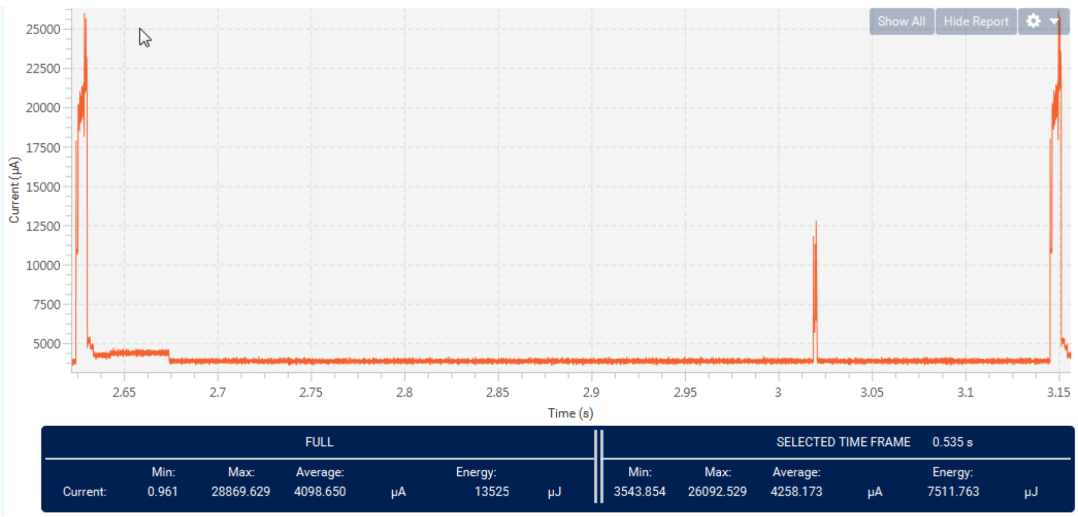
Figure 24. **FP-SNS-ALLMEMS2 environmental plot**



The FP-SNS-ALLMEMS2+ adapts the BLE link periodicity (bandwidth) and reduces the BLE activity to 500 ms, which corresponds to the actual need for evacuating the measures of sensors towards the host. As a result, on a 500 ms period, the plot shows only two bursts (BLE burst and sensor access).

Compared to the FP-SNS-ALLMEMS2, there is an extra 55% of power consumption gain.

Figure 25. **FP-SNS-ALLMEMS2+ environmental plot**



10 Sensor fusion use case

This use case implies frequent access to all motion sensors as well as significant MCU processing for carrying out the fusion of the sensors.

For the FP-SNS-ALLMEMS1, in addition to the usual spikes (ticks, environmental sensor activity, etc.) the estimation is 10 ms patterned activity where CPU remains active 50 % of the time. When the CPU is not processing, it is set in WFI, with a current floor of 12500 µA.

Figure 26. **FP-SNS-ALLMEMS1 sensor fusion plot**

For the FP-SNS-ALLMEMS2, once spurious activities have been cleared out, the waveform reveals the essential activities (processing, sensor access, sensor processing, MCU processing, Bluetooth). This generates a 41% power gain which is remarkable for an intense use case where the system is active most of time.

Figure 27. **FP-SNS-ALLMEMS2 sensor fusion plot**

Note: All the observations for the FP-SNS-ALLMEMS2 also apply to FP-SNS-ALLMEMS2+.

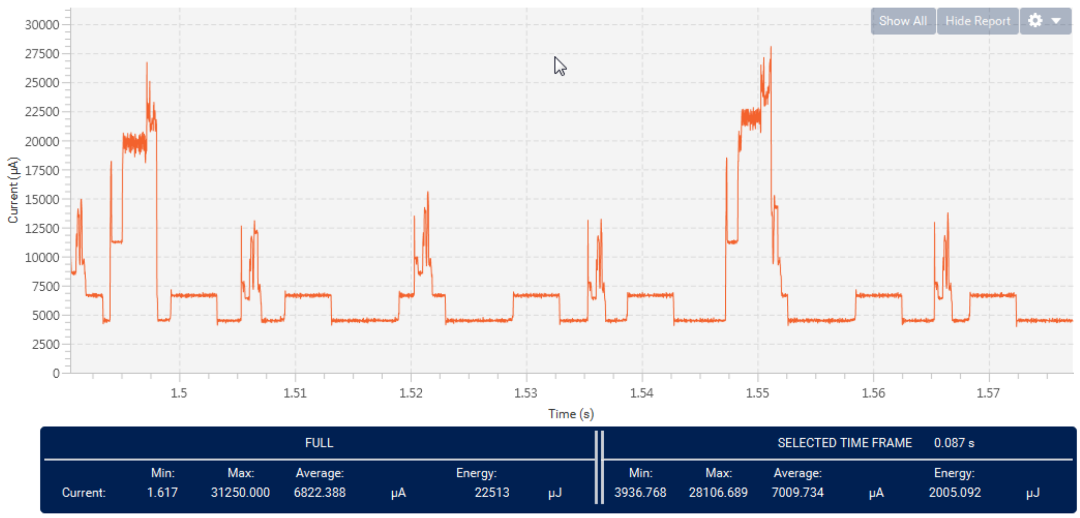
11 Plot acceleration use case

In this use case, only the accelerometer data are accessed and measurements reported through BLE link. Therefore, the MCU processing phases are missing.

Figure 28. FP-SNS-ALLMEMS1 acceleration plot

For the FP-SNS-ALLMEMS2, after power tuning, the 10 ms motion sensor activity reveals itself as well as BLE 15 ms periodic reporting. A 72% gain is observed besides a high frequency polling and reporting.

Figure 29. FP-SNS-ALLMEMS2 acceleration plot

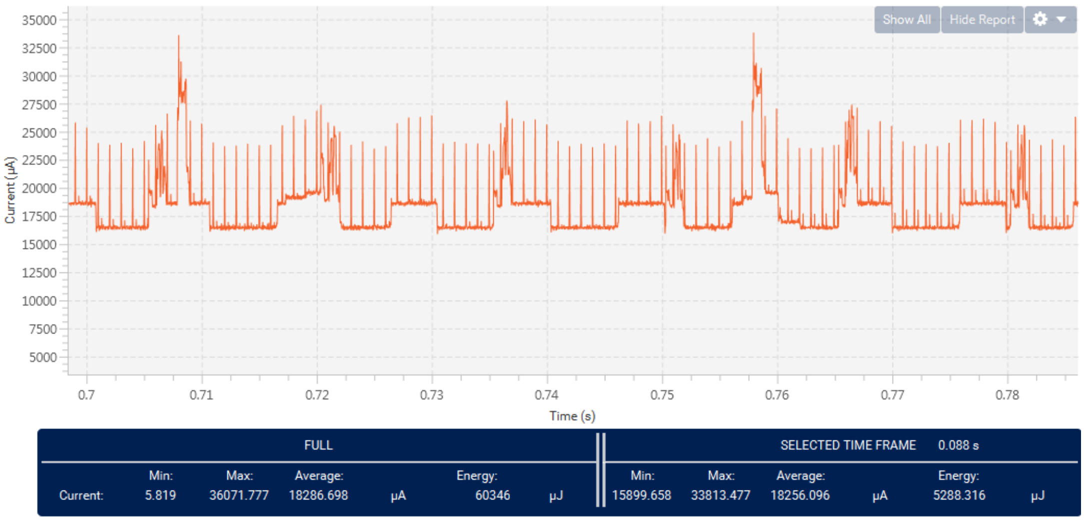


Note: All the observations for the FP-SNS-ALLMEMS2 also apply to FP-SNS-ALLMEMS2+.

12 Plot microphone level use case

This use case is the first audio (that is, using microphones) of the series, explaining the extra power consumption observed. The patterns are the same but the current floor is higher.

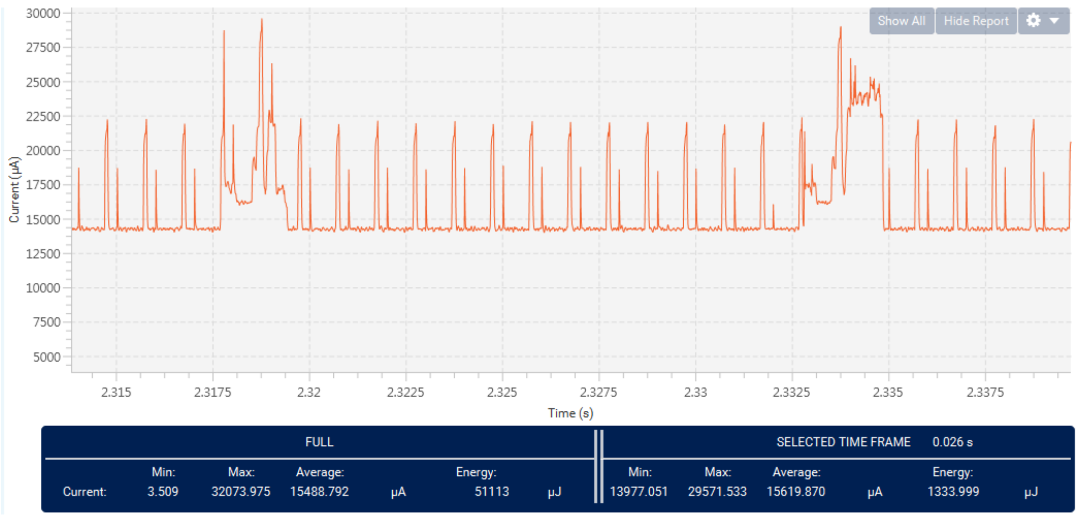
Figure 30. FP-SNS-ALLMEMS1 microphone plot



For the FP-SNS-ALLMEMS2, when spurious (MEMS) activity has been removed, the 1 ms, that so far were mostly suppressed, are still in the picture due to the very short idle periods induced by the use case. An additional 1 ms periodic wave form, due to microphone streaming, is also noticeable.

Even with this use case involving continuous activity, a 19% power gain is demonstrated.

Figure 31. FP-SNS-ALLMEMS2 microphone plot



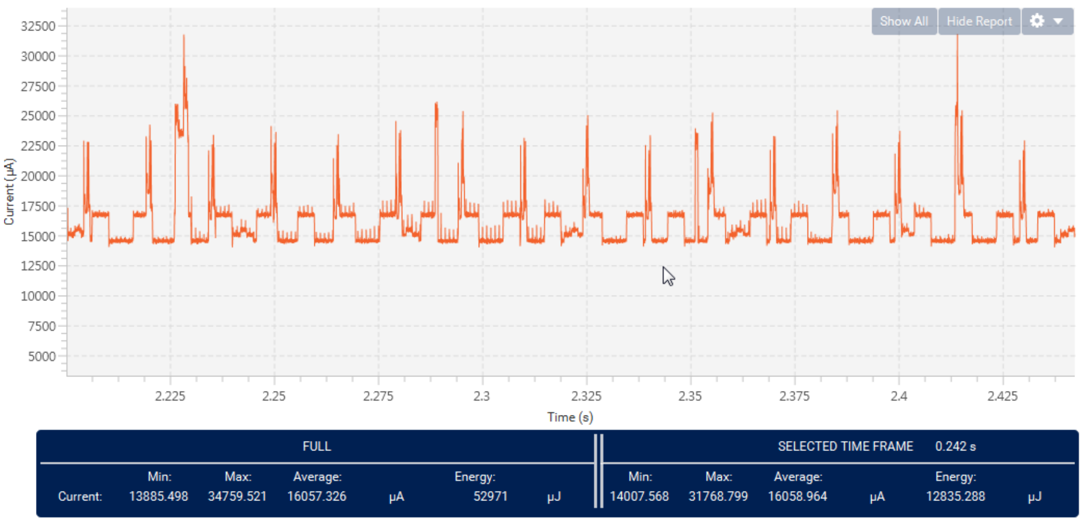
Note: All the observations for the FP-SNS-ALLMEMS2 also apply to FP-SNS-ALLMEMS2+.

13 Activity recognition use case

This use case uses accelerometer data every 60 ms to produce an estimation of current human activity of the user holding the device.

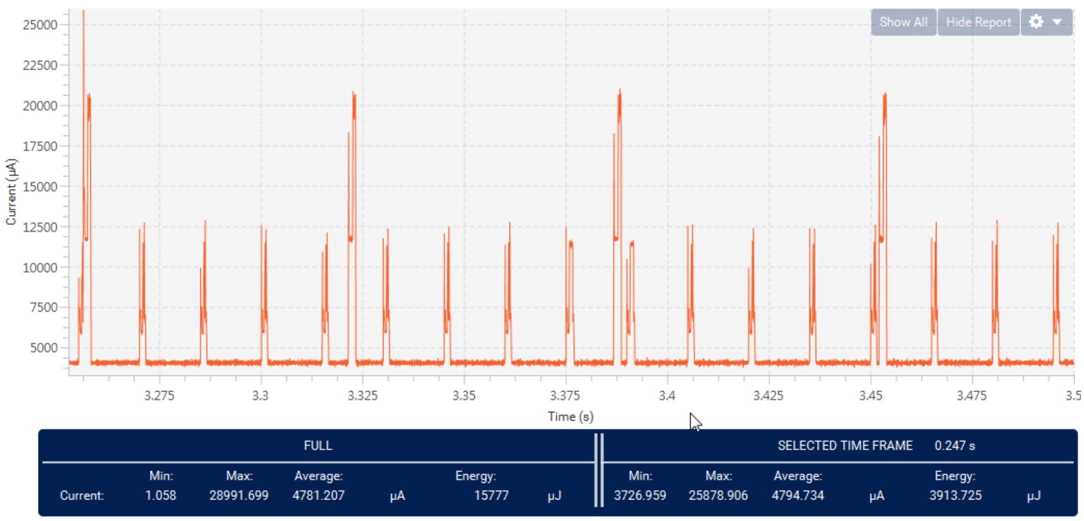
In the FP-SNS-ALLMEMS1, all sensors are active and the MCU is left in WFI mode. The 1 ms tick spikes are visible.

Figure 32. **FP-SNS-ALLMEMS1 activity recognition plot**



The FP-SNS-ALLMEMS2 keeps only the accelerometer activity and BLE that are clearly detached on the plot. The activity processing on MCU (every 60 ms) is light, unlike sensor fusion processing. The gain of FP-SNS-ALLMEMS2 compared to FP-SNS-ALLMEMS1 is 92%.

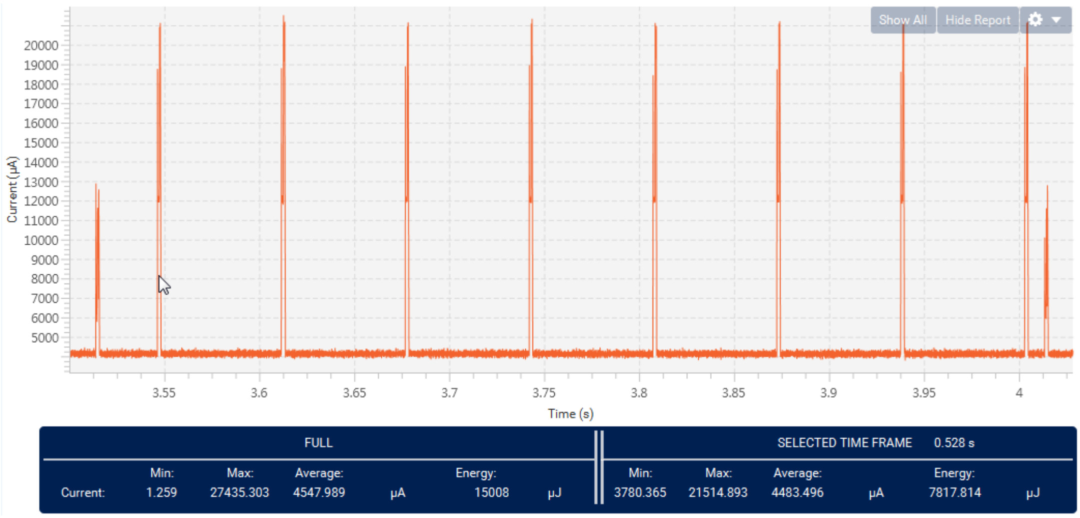
Figure 33. **FP-SNS-ALLMEMS2 activity recognition plot**



The FP-SNS-ALLMEMS2+, with BLE link adaptation, extends the reporting period to the host to 500 ms. This is hardly noticed considering the algorithm latencies.

As a result, an additional 24% gain is measured, making it a 94% gain versus FP-SNS-ALLMEMS1.

Figure 34. FP-SNS-ALLMEMS2+ activity recognition plot

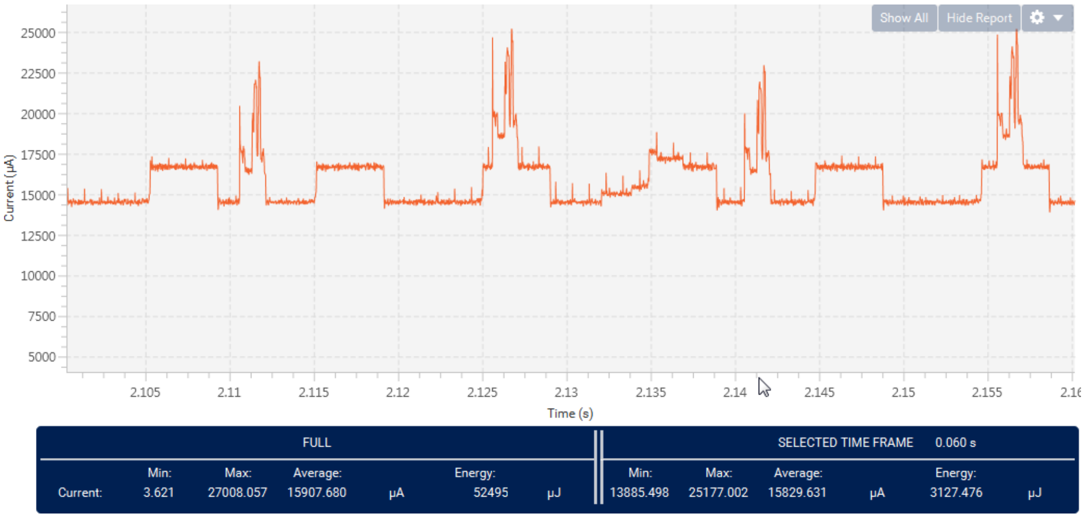


14 Hardware event detection use case

In this use case the node reports accelerometer hardware event through an interrupt mechanism, so the system is almost always idling.

The FP-SNS-ALLMEMS1 waveform represents only the overhead.

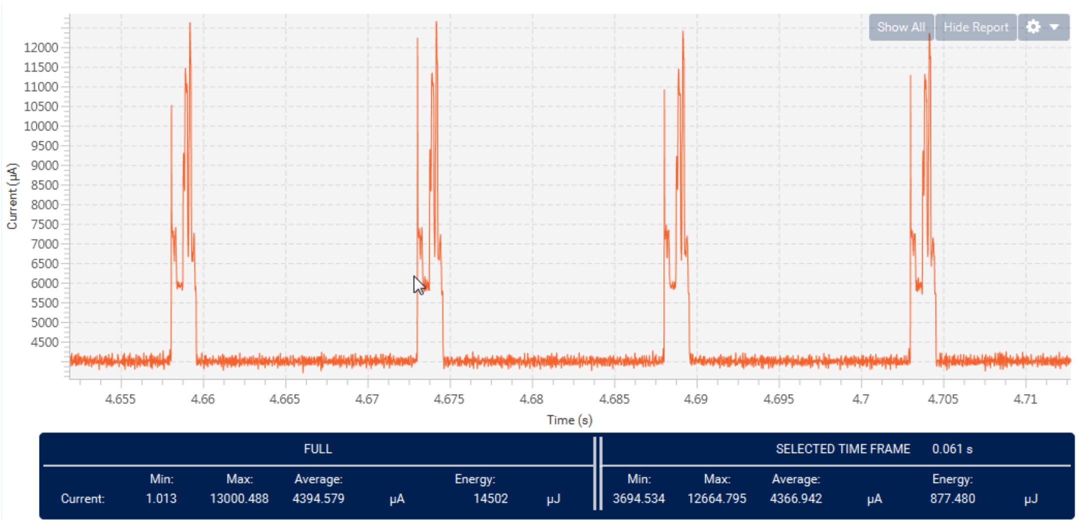
Figure 35. FP-SNS-ALLMEMS1 hardware event detection plot



After optimizing power, the waveform barely reveals the BLE activity, accelerometer power consumption is included in the current floor.

The gain in power is 95 %.

Figure 36. FP-SNS-ALLMEMS2 hardware event detection plot



As for the previous use case, the reporting period is hardly noticed, so it has been extended to 500 ms in the “+” version, with no impact for the user experience.

It gives an extra 64% of power gain, so that the overall gain (vs original firmware) sums up to 98%.

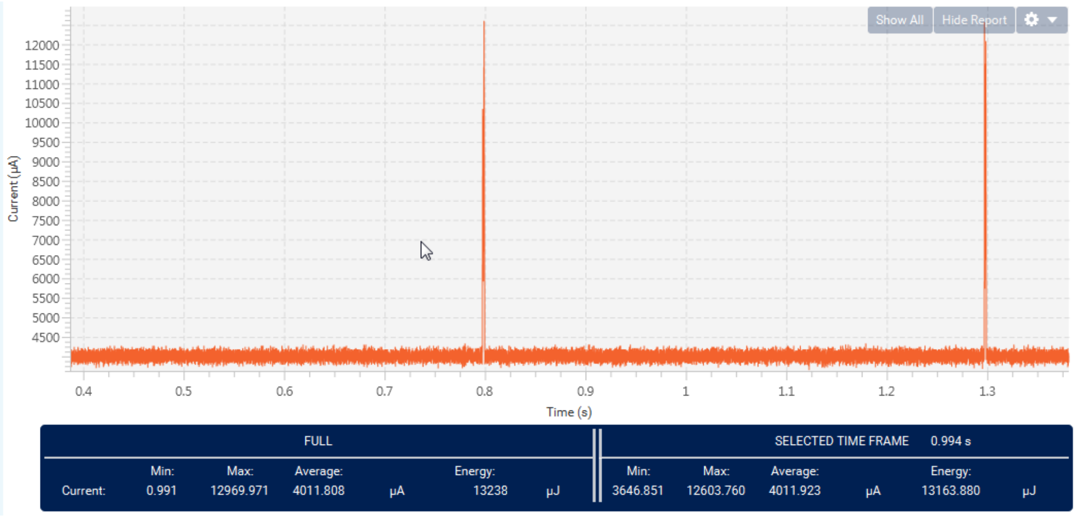


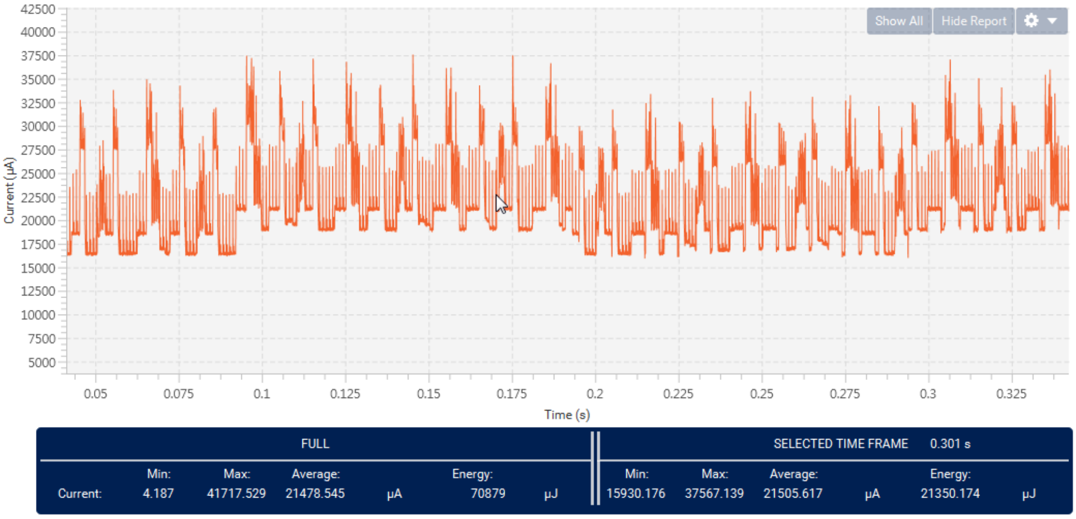
Figure 37. FP-SNS-ALLMEMS2+ hardware event detection plot

15 BlueVoice use case

BlueVoice streams audio samples out from microphones, compress and send them on a high-speed synchronous BLE link to the host in real-time.

It is the most power consuming use case (above 17 000 µA). A lot of MCU processing is entailed as well as a huge amount of data on air, which results in a big BLE activity.

Figure 38. FP-SNS-ALLMEMS1 BlueVoice plot



In the FP-SNS-ALLMEMS2, power resource management has been optimized, that is powering down unused hardware (such as sensors).

The gain is 16%.

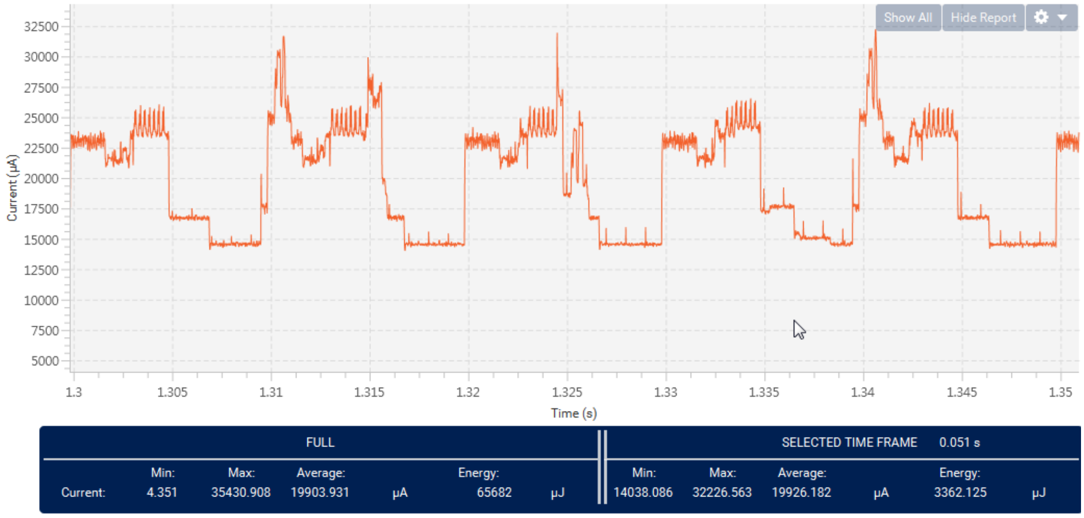
Figure 39. FP-SNS-ALLMEMS2 BlueVoice plot

Note: All the observations for the FP-SNS-ALLMEMS2 also apply to FP-SNS-ALLMEMS2+.

16 Compass use case

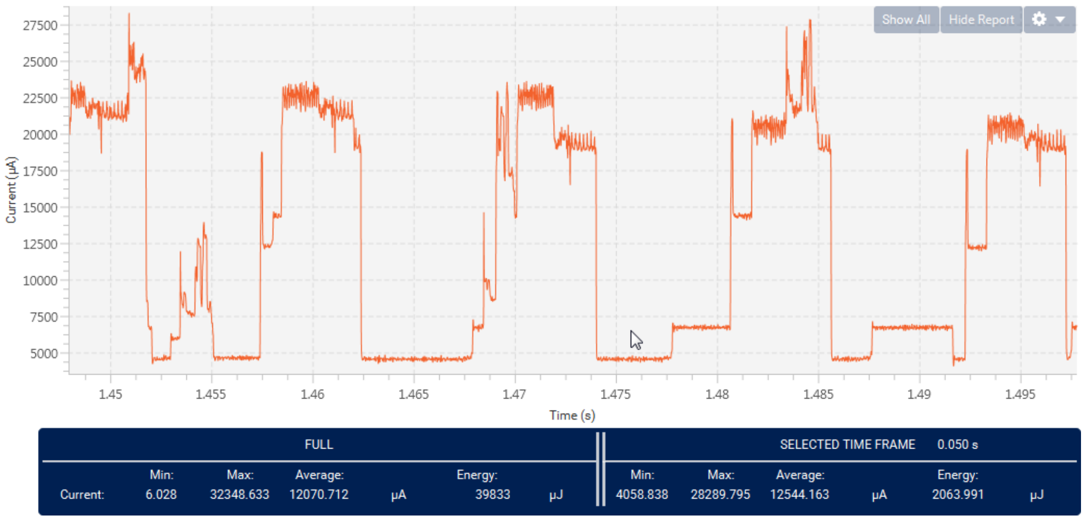
This use case is very similar to sensor fusions in terms of scheduling and processing, even though the algorithms are different.

Figure 40. FP-SNS-ALLMEMS1 compass plot



In the FP-SNS-ALLMEMS2, a gain of 49 % is measured (as in the sensor fusion).

Figure 41. FP-SNS-ALLMEMS2 compass plot

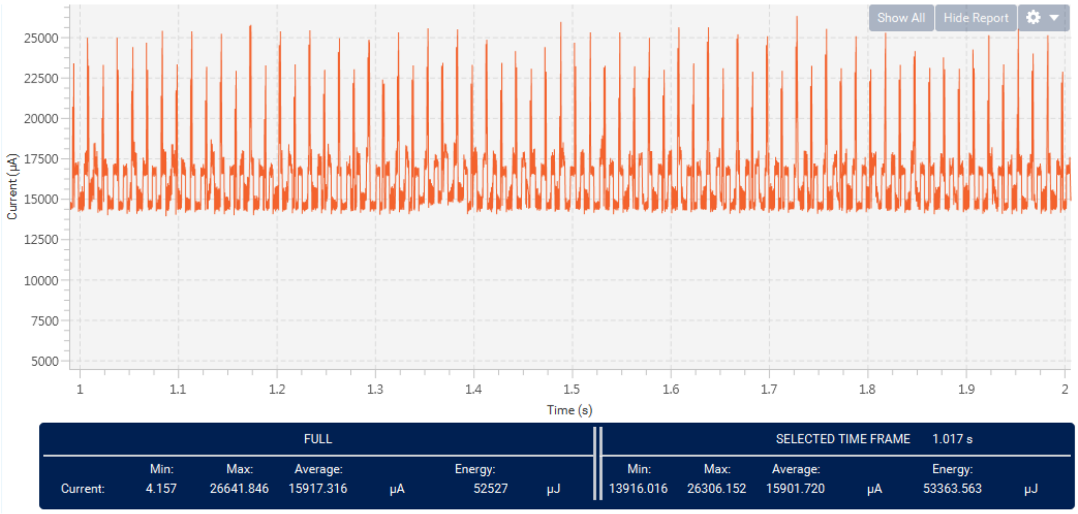


Note: All the observations for the FP-SNS-ALLMEMS2 also apply to FP-SNS-ALLMEMS2+.

17 Node monitor use case

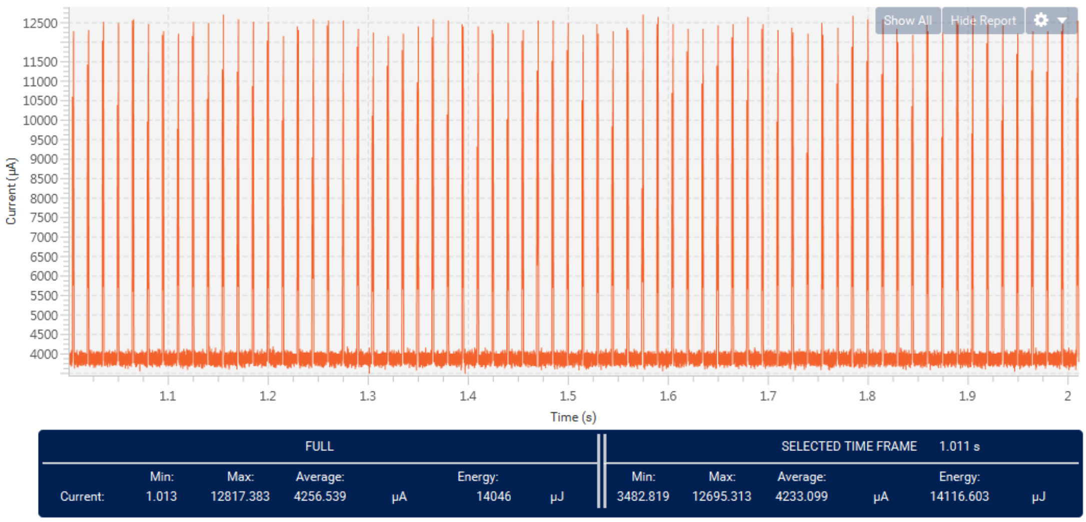
In this use case the node reports only its battery level, which makes it highly optimizable.
The level of current looks similar to low activity use cases (for instance, hardware event).

Figure 42. FP-SNS-ALLMEMS1 node monitor plot



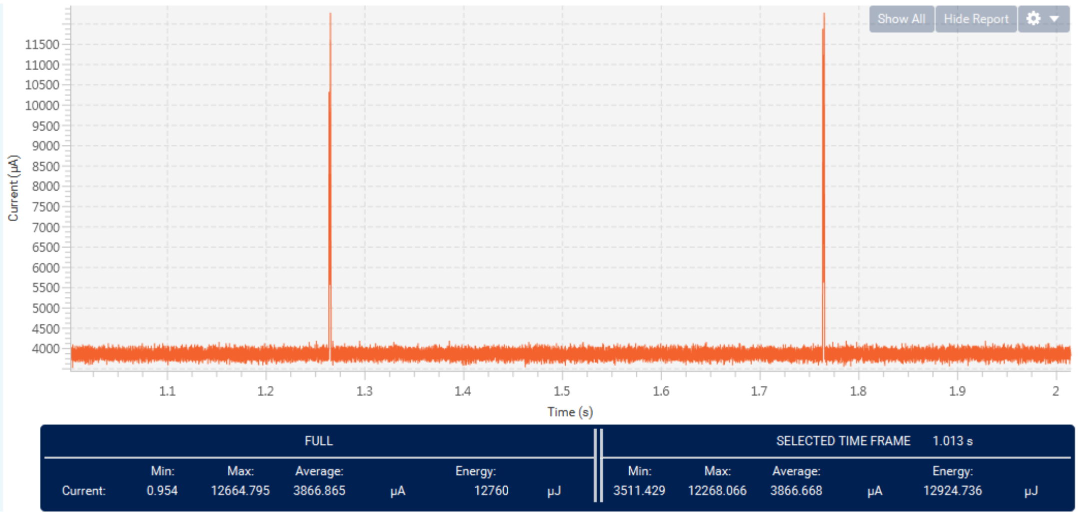
In the FP-SNS-ALLMEMS2, once unnecessary activities have been optimized, BLE radio burst remains at a too high frequency considering battery charge/discharge time constants.
A sensible gain of 96% is achieved.

Figure 43. FP-SNS-ALLMEMS2 node monitor plot



In the FP-SNS-ALLMEMS2+, a 500 ms reporting period is applied, the battery monitoring activity is not even noticeable in terms of power consumption. The 500 ms could even be extended to few seconds without a big inconvenience, but this would add a visible delay at host side when connecting/disconnecting the charger (USB plug).

Figure 44. **FP-SNS-ALLMEMS2+ node monitor plot**



18 Synthesis of gains in power

Table 1. **Power gain comparison**

Comparison table	Average current (µA)			Gain		
	FP-SNS-ALLMEMS1	FP-SNS-ALLMEMS2	FP-SNS-ALLMEMS2+	FP-SNS-ALLMEMS2 vs. FP-SNS-ALLMEMS1	FP-SNS-ALLMEMS2+ vs. FP-SNS-ALLMEMS1	FP-SNS-ALLMEMS2+ vs. FP-SNS-ALLMEMS2
Connectable	10536	172		98%		
Environmental	10500	665	298	94%	97%	55%
Sensor fusion	14918	8817		41%		
Plot acceleration	10975	3022		72%		
Plot microphone level	14486	11688		19%		
Activity recognition	12257	981	747	92%	94%	24%
Hardware event detection	12107	594	211	95%	98%	64%
BlueVoice	17678	14870		16%		
Compass	16103	8270		49%		
Node monitor	12117	456	66	96%	99%	86%

All use cases are taking advantage from the reworking of the original FP-SNS-ALLMEMS1 firmware package, even those which are particularly unfit for power optimization like BlueVoice.

Some use cases are particularly interesting, such as connectable, which is the least consuming one.

Sensor fusion and compass are also very interesting as activity intensive use cases and still yielding a bit less than 50% of power gain.

19 Conclusions

In this application note, FP-SNS-ALLMEMS2 has been introduced, with a detailed observation of the power optimization for one of the supported platform, the NUCLEO-L476RG development board, equipped with some expansion boards.

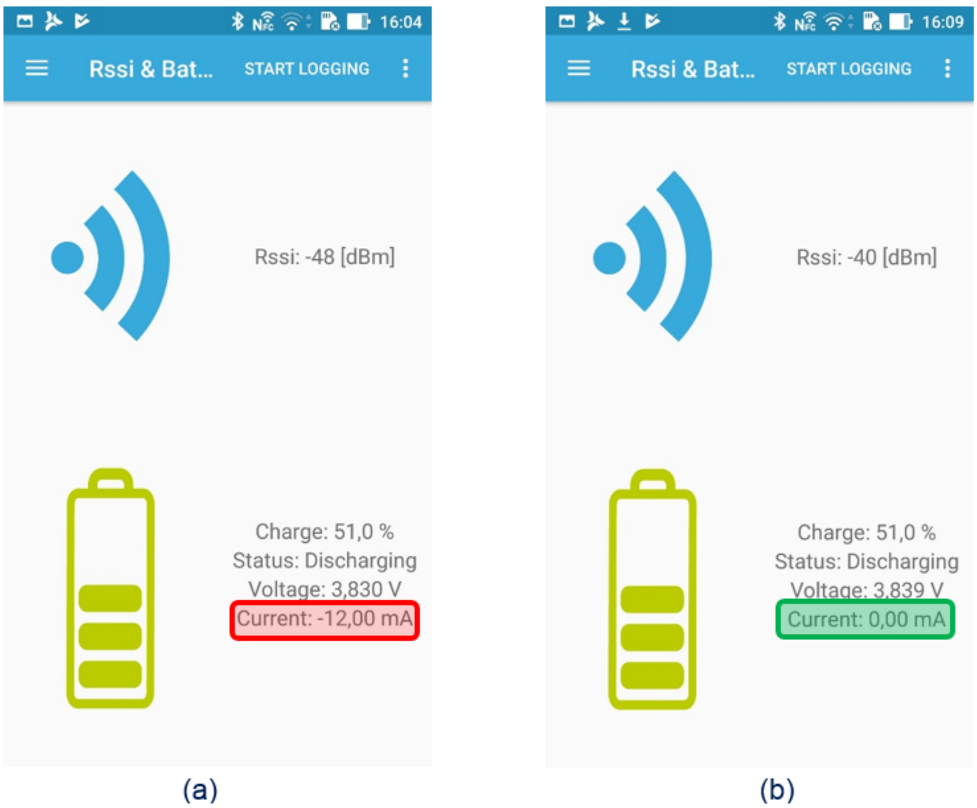
The ST power shield (X-NUCLEO-LPM01A) proved to be very handy to profile and fine-tune the firmware. Depending on use cases, gains in power have been accurately measured up to 99%.

This application note does not consider the other hardware platforms supported by the FP-SNS-ALLMEMS2 function pack.

For further optimization trials, you can exploit the features of the SensorTile, a battery operated, form factor device (MCU at 1V8) that demonstrates that all the benefits of power optimizations are directly translated in operation time increase.

You can try to flash the SensorTile, as a first step with the FP-SNS-ALLMEMS1 firmware, then go to the node monitor panel in the Rssi & Battery page of the ST BlueMS application (version 3.2.0 or higher). Then, as a second step, you can do the same with the FP-SNS-ALLMEMS2 firmware. The next figure outlines the result in both cases.

Figure 45. ST BlueMS app Rssi & Battery for FP-SNS-ALLMEMS1 (a) and FP-SNS-ALLMEMS2 (b)



From a drown current of 12 mA, which is consistent with our measures and floor current assumption when using an STM32 Nucleo board, with the original version of the firmware, we then fall close to 0 mA with the new version.

Revision history

Table 2. **Document revision history**

Date	Version	Changes
13-Jul-2018	1	Initial release.

Contents

List of tables

List of figures