

How to use the wireless multi-sensor development kit with customizable app for IoT and wearable sensor applications

Introduction

The **STEVAL-MKSBOX1V1** (SensorTile.box) is a ready-to-use box kit with wireless IoT sensor platform designed to help you build apps that use motion and environmental sensors, regardless of your level of expertise.

The hardware node is a board that fits into a small plastic case (IP54) with a rechargeable battery. You can connect with your smartphone to the board via Bluetooth by using the **ST BLE Sensor** app (available both on Google Play and Apple Store) and immediately build your own apps through a special interface that offers beginner and expert level functionality. This multi-sensor kit therefore allows you to design wireless IoT and wearable sensor applications quickly and easily, without performing any programming.

SensorTile.box includes a firmware programming and debugging interface that allows professional developers to engage in more complex firmware customization using the STM32 Open Development Environment (STM32 ODE), which includes a sensing AI function pack with neural network libraries.

The kit board includes an embedded SPBTLE-1S Bluetooth SMART application processor that is compliant with BT specification v4.2. This transmitter module is FCC (ID:S9NSPBTLE1S) certified and IC (IC:8976-SPBTLE1S) certified.

Note: *SPBTLE-1S has been replaced by the **BlueNRG-M2** module, compliant with BT specification v5.2 in latest production batches.*

Figure 1. STEVAL-MKSBOX1V1 (SensorTile.box) multi-sensor development kit



1 How to set up the hardware

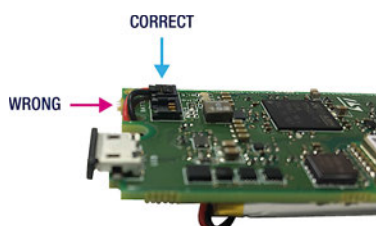
Important:

Before you begin, please check the insert card that comes with the SensorTile.box blister pack. If it doesn't show a procedure for battery connection similar to the steps below, then your device is supplied with the battery already connected to the board. In this case, you only need to connect the device via USB to wake it up the first time.

If the insert card has a similar procedure to the steps below, your device is supplied with the battery disconnected and you should follow this procedure to connect the battery and wake the device up.

- Step 1.** Remove the SensorTile.box contents from its package.
- Step 2.** Unscrew the shroud cover.
You should have the following items:
 - An evaluation board in a plastic shroud
 - A LiPo battery
- Step 3.** Slide the male battery connector vertically into the female connector on the board.
You will hear a light click when the connector is attached correctly.

Figure 2. STEVAL-MKSBOX1V1 battery connection



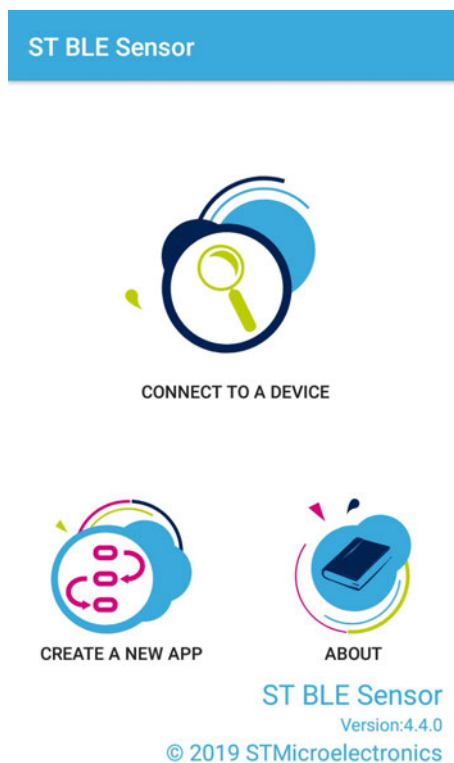
- Step 4.** Re-position the circuit with the battery below it and close the shroud with one of the following types of lid:
 - with flanges
 - without flanges
- Step 5.** If necessary, charge the battery via a USB cable.
The blinking of the red LED indicates the battery charging status.

2 How to use ST BLE Sensor app with SensorTile.box

Before you begin, you need to download and install our [ST BLE Sensor](#) app on your smartphone. The app is available from the [Google](#) and [Apple](#) online stores.

Step 1. Launch the app on your smartphone.

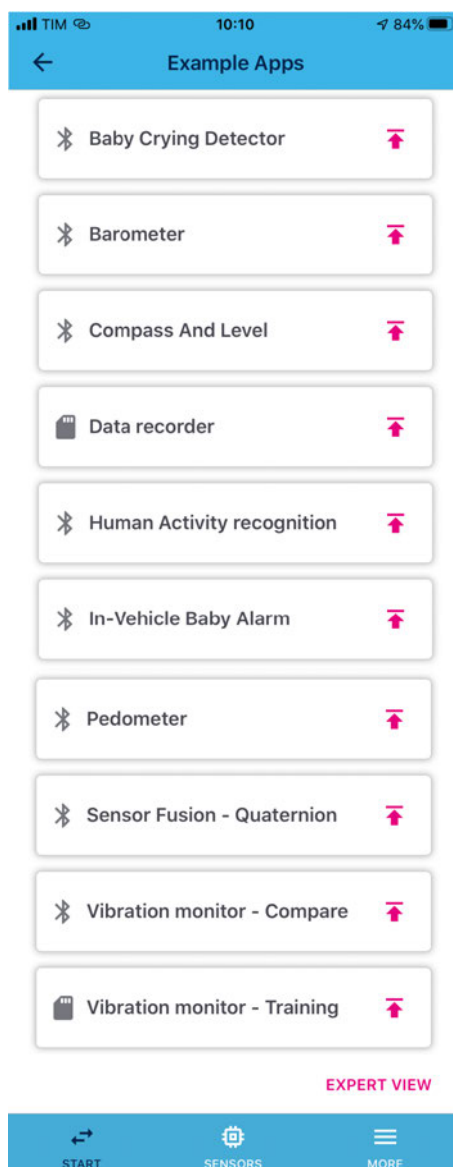
Figure 3. ST BLE Sensor app main screen



Step 2. Select **[CREATE A NEW APP]**.

The Example Apps screen that follows lists the preloaded apps that you can use immediately.

Figure 4. Example Apps screen




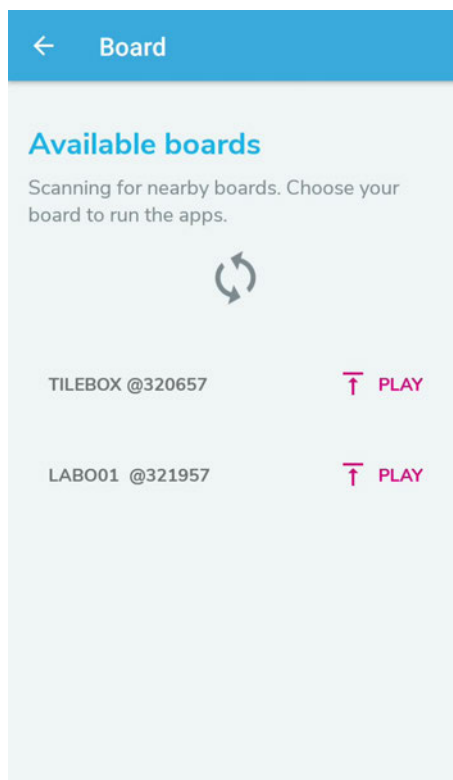
- Step 3.** Select one of the apps with the  icon from the list.
After you select the app, **ST BLE Sensor** will scan for available SensorTile.box devices in range.

Figure 5. Board selection



- Step 4.** Select the appropriate SensorTile.box device from the Board screen.
A blue LED on the SensorTile.box device will flash slowly to confirm Bluetooth pairing.
A pop up message in **ST BLE Sensor** will prompt you to confirm loading the new app in replacement of any previously opened apps.
- Step 5.** Select the appropriate SensorTile.box device from the Device List.
The app will commence monitoring or logging activity and return real time feedback data to the corresponding app screen in **ST BLE Sensor**.

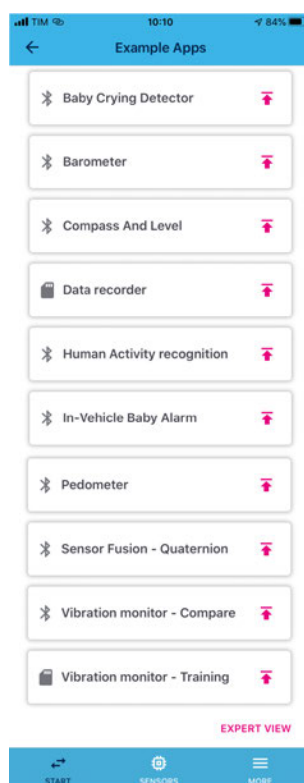
3 Application descriptions

3.1 Mode 1 example apps

The **ST BLE Sensor** bundles the following ready-to-use app scenarios:

- Baby Crying Detector
- Barometer
- Compass and Level
- Data Recorder
- Human Activity recognition
- In-Vehicle Baby Alarm
- Pedometer
- Sensor Fusion - Quaternion
- Vibration Monitor - Training
- Vibration Monitor - Compare

Figure 6. Apps screen



App scenarios with the  icon produce immediate outputs on your smartphone in real time.

App scenarios with the  icon store sample data on the internal micro SD card.

App scenarios with the  icon are reserved for Expert mode.

RELATED LINKS

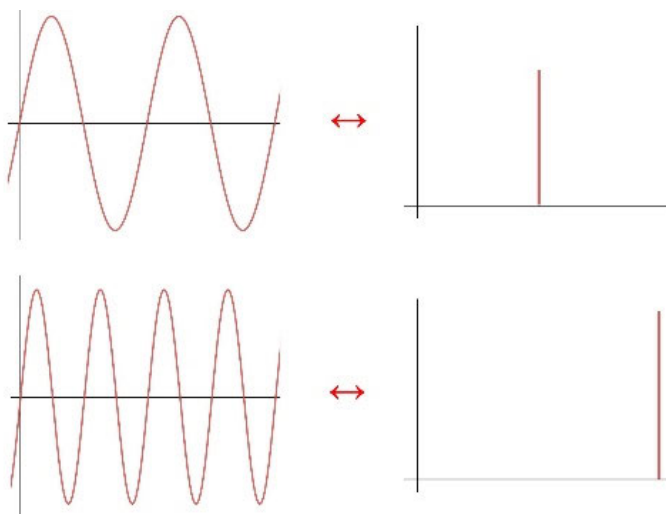
[2 How to use ST BLE Sensor app with SensorTile.box on page 3](#)

3.1.1 Baby crying detector

The baby crying detector (BCD) app implements the Fast Fourier Transform (FFT) and artificial intelligence processing to detect baby crying events using the [Sensortile.box](#) on-board microphone.

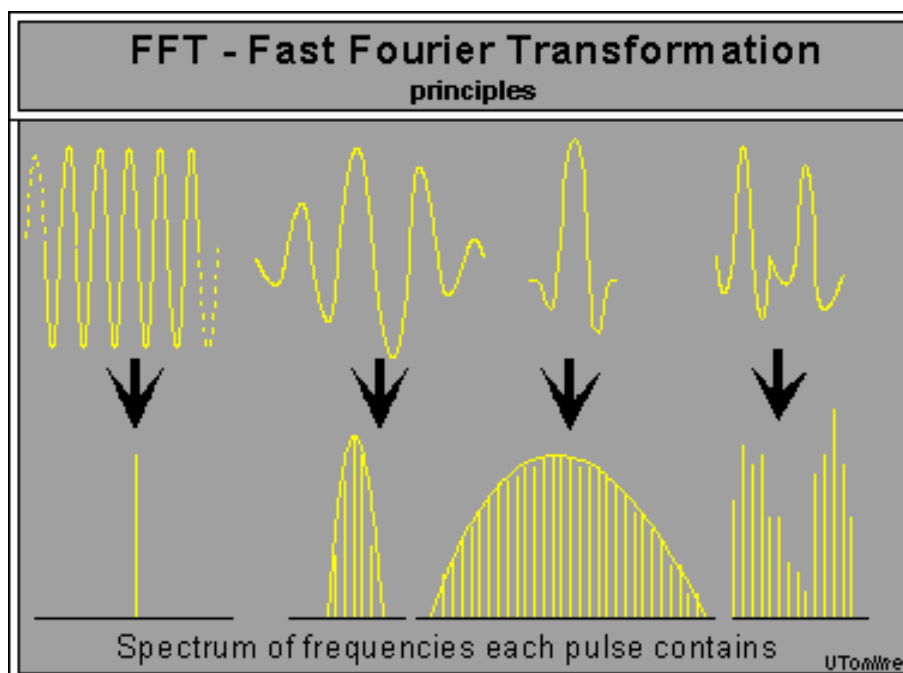
The analysis of the acquired audio is based on the FFT that converts a signal from its original time domain to a representation in the frequency domain.

Figure 7. FFT analysis - from time to frequency domain



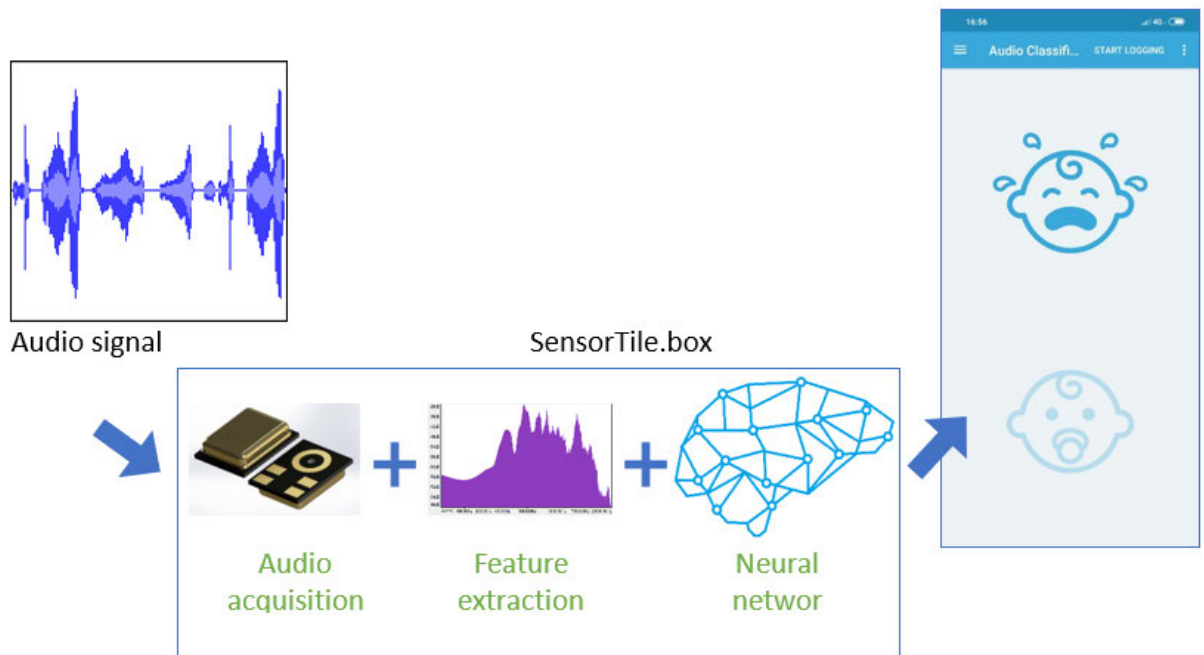
The FFT of the audio signal is the result of all the contributions of each frequency and the related magnitude factor generated by the audio signal.

Figure 8. FFT analysis - principles



The FFT feature extraction of the acquired signal is processed by the STM32 MCU which calculates the MEL FFT and the MFCC (MEL frequency cepstral coefficient) parameters sent to the implemented MCU neural network: if a baby crying event is detected, the green user LED on the [Sensortile.box](#) board lights up and a warning is sent to the smartphone via Bluetooth.

Figure 9. STBLESensor - baby crying detection process

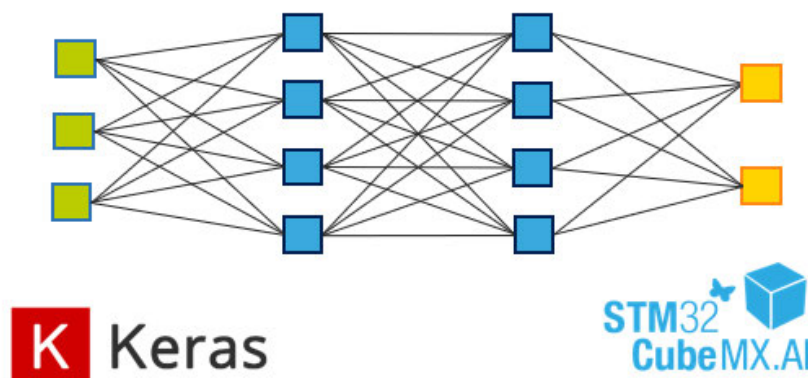


The neural network is classified as a deep feed forward neural network and its structure is composed of 2 hidden nodes of 100 neurons each.

The tool used to develop the neural network is Keras with an open source high level library written in Python.

Optimization and loading of the neural network on the [Sensortile.box](#) has been performed using [STM32CubeMX.AI](#).

Figure 10. STBLESensor - baby crying app neural network



The baby crying app works with the following ST high sensitivity audio sensor and operating parameter settings:

- **APP DATA INPUT:** microphone audio acquisition
- **SENSOR USED:** [MP23ABS1](#) MEMS microphone
- **SENSOR SETTINGS:** 16 KHz sample acquisition
- **APP DATA OUTPUT:** baby crying/not crying icon

RELATED LINKS

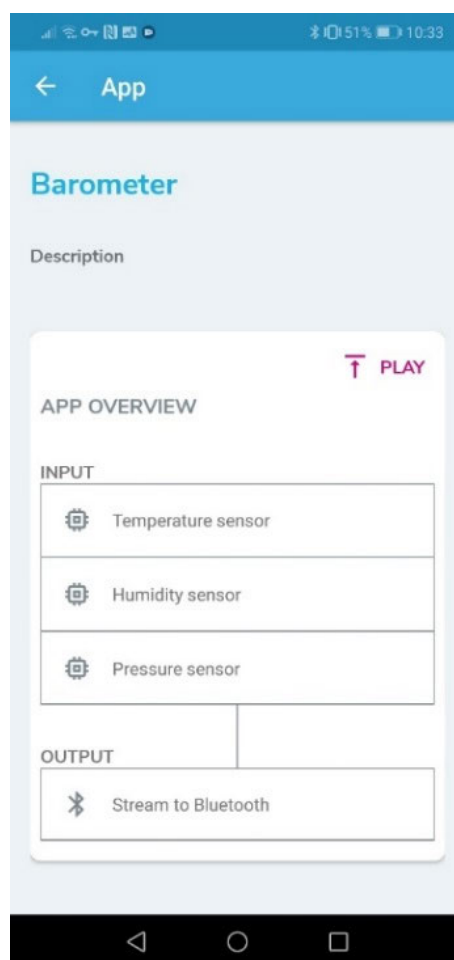
[Appendix A ARMA filter coefficient calculation on page 35](#)

3.1.2

Barometer app

The Barometer app uses the [Sensortile.box](#) on-board environmental sensors ([STTS751](#), [LPS22HH](#) and [HTS221](#)).

Figure 11. Barometer app screen



The Barometer app monitors the environmental information in real-time and shows the data on your smartphone as icons or graph plot.

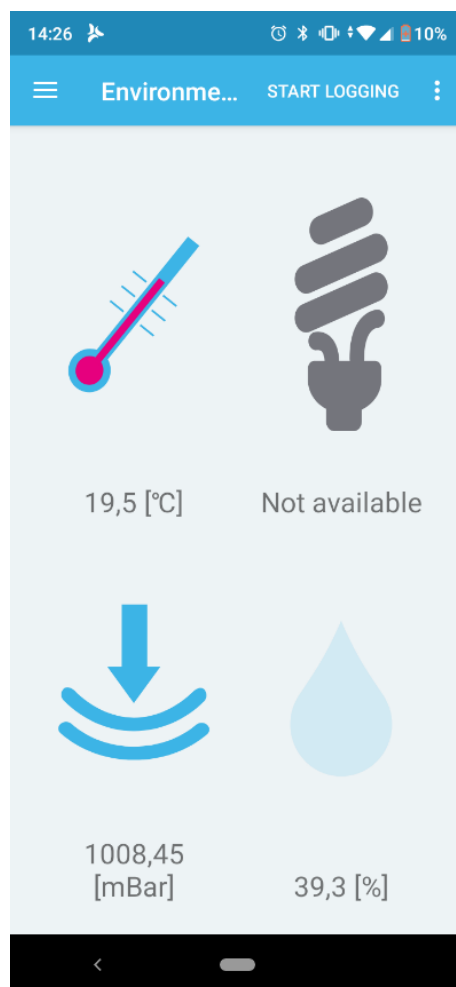
The [LPS22HH](#) pressure sensor embeds another built-in sensor for temperature compensation (0.5 hPa with a range of 260-1260 hPa of absolute pressure).

The [STTS751](#) digital temperature sensor has an operating temperature range of -40/+125 °C, with maximum resolution 0.0625 °C/LSB and precision of ± 0.5 °C (typ.).

The [HTS221](#) digital relative humidity and temperature sensor has a relative humidity range of 0/100%, a sensitivity of 0.004% RH/LSB, a humidity accuracy of ± 3.5% RH, 20-80% RH and an accuracy in temperature of ± 0.5 °C (typ.) in the range of 15/+40 °C.

When you run the Barometer app and connect the [Sensortile.box](#) device, the ST BLE Sensor app shows a monitoring screen for the environmental sensors.

Figure 12. Environmental screen



You can access other output options from the menu icon in the top left of the screen.

Figure 13. Plot Data screen - humidity

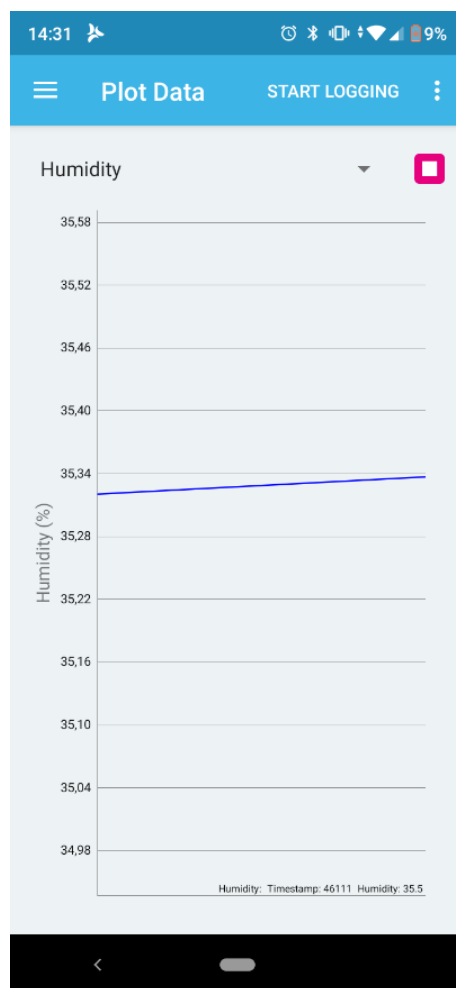
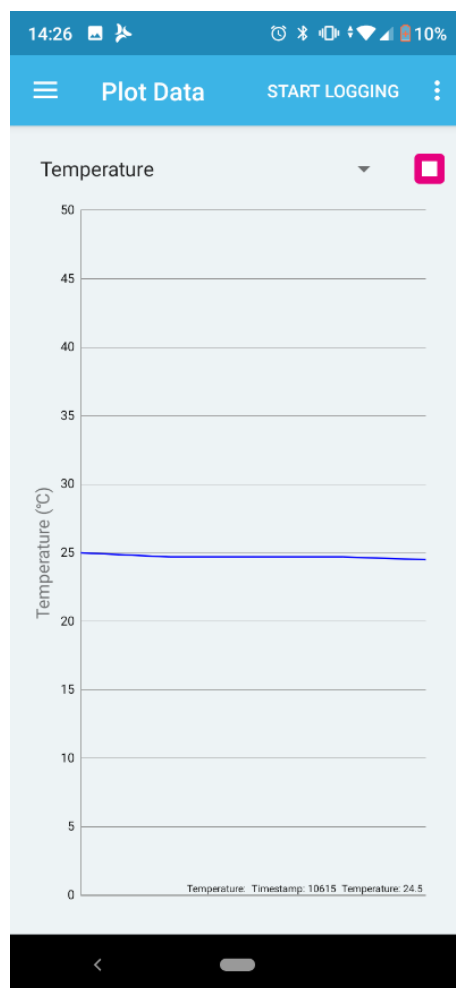


Figure 14. Plot Data screen - temperature



Thanks to the low power sensors, low output data rate and low power MCU, this app is highly suitable for battery-based projects with very low power consumption.

The Barometer app sets the following operating parameter settings for the following ST high accuracy environmental sensors:

- **APP DATA INPUT:** pressure, temperature and humidity values
- **SENSORS USED:**
 - [LPS22HH](#) (absolute pressure MEMS digital sensor)
 - [STTS751](#) (temperature digital sensor)
 - [HTS221](#) (relative humidity and temperature digital sensor)
- **SENSOR SETTINGS:**
 - [LPS22HH](#) settings:
 - Power mode: Low Noise
 - Output Data Rate: 1 Hz
 - Filter: ODR/2
 - [STTS751](#) settings:
 - Low power mode
 - Output data rate: 1 Hz
 - [HTS221](#) settings:
 - Low power mode
 - Output data rate: 1 Hz
- **APP DATA OUTPUT:**
 - Relative humidity (%)
 - Temperature (°C)
 - Absolute Pressure (mBar)
 - Plot collected data

3.1.3

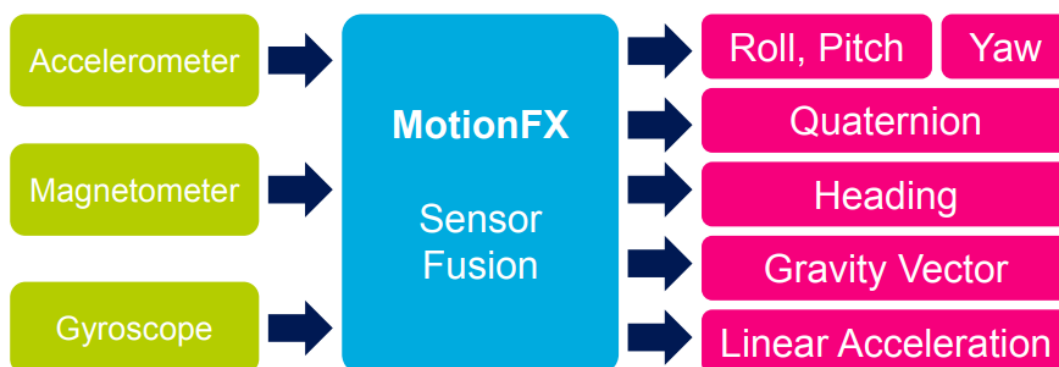
Compass and Level app

The Compass and Level app shows the orientation estimation of the [Sensortile.box](#) in relation to the Earth magnetic North or level indication.

The app is based on the sensor fusion firmware algorithm (MotionFX library) embedded in the [Sensortile.box](#) MCU.

The sensor fusion algorithm is an extended Kalman filter (EKF) that minimizes sensor inaccuracies and includes gyroscope calibration and magnetometer calibration (to compensate the magnetometer offset).

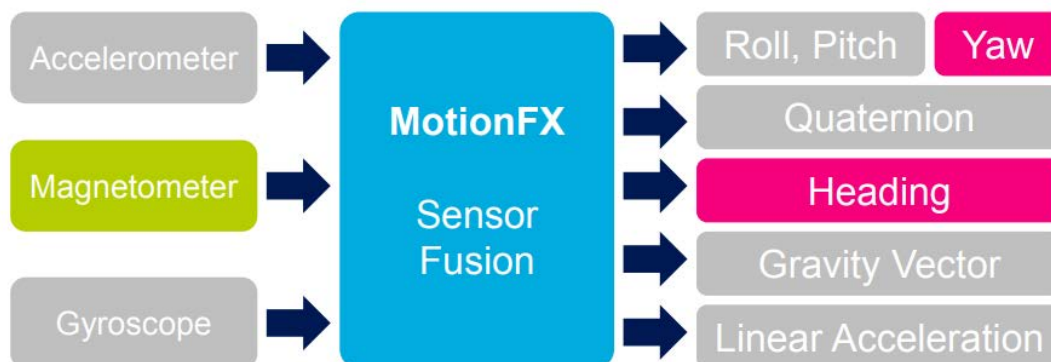
Figure 15. MotionFX algorithm flow



The algorithm uses the [LSM6DSOX](#) iNemo 6-axis accelerometer and gyroscope data and the [LIS2MDL](#) 3-axis compensated magnetometer data as inputs, combining the two sensors in a virtual 9-axis sensor.

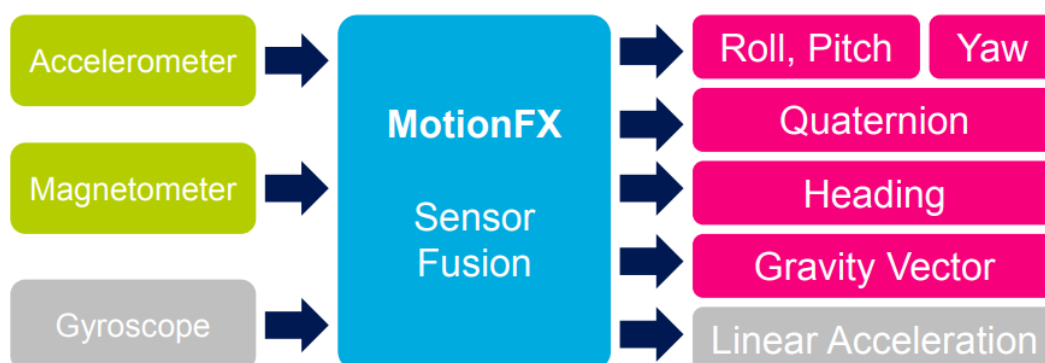
The magnetometer indicates yaw angle and heading, but only if hard-iron offset is compensated and when there is no additional magnetic field around the [Sensortile.box](#) disturbing the measurement.

Figure 16. MotionFX algorithm flow - magnetometer function



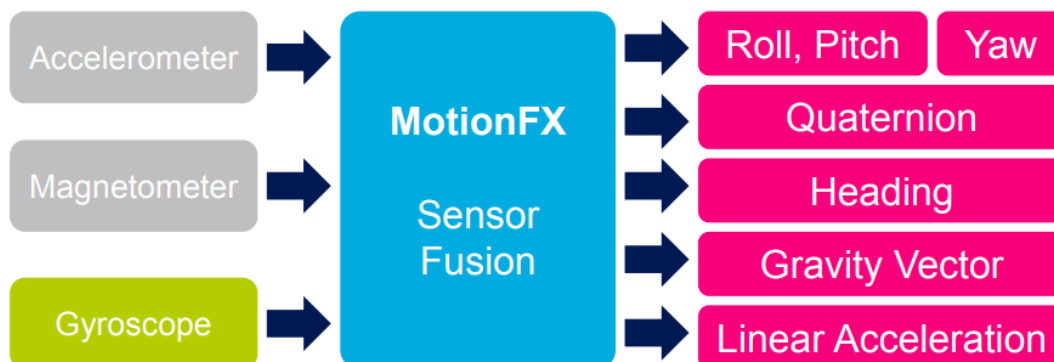
Data regarding yaw and angle heading are also given when tilt is compensated by the accelerometer.

Figure 17. MotionFX algorithm flow - accelerometer tilt compensation



The gyroscope indicates the new orientation based on the previous one, when its bias is compensated by the accelerometer. The gyroscope can detect the static condition of the [Sensortile.box](#),

Figure 18. MotionFX algorithm flow - gyroscope function



To enable the magnetometer calibration, you need to touch the symbol highlighted in the picture below.

Figure 19. STBLESensor Compass and Level app - calibration icon



Then, move the [Sensortile.box](#) in a 8-pattern figure as shown below; the calibration is completed when the icon becomes green.

Figure 20. STBLESensor Compass and Level app - starting the calibration



The MotionFX library provides orientation estimation, magnetometer hard-iron offset compensation, accelerometer vibration rejection and gyroscope bias compensation.

The sensor fusion algorithm calculates the quaternion coefficient and the Euler angles to detect the right orientation of the [Sensortile.box](#) represented by a compass rotation or level indicator.

Figure 21. STBLESensor Compass and Level app - orientation screen



Figure 22. STBLESensor Compass and Level app - offset example 1



Figure 23. STBLESensor Compass and Level app - offset example 2



The Compass and Level app works with the following ST high accuracy motion sensors and operating parameter settings:

- **APP DATA INPUT:** Accelerometer, gyroscope and magnetometer values
- **SENSORS USED:**
 - [LSM6DSOX](#) (acceleration and gyroscope sensor)
 - [LIS2MDL](#) (compensated magnetometer)
- **SENSOR SETTINGS:**
 - [LSM6DSOX](#):
 - Low power mode
 - Output data rate: 52 Hz
 - Low pass filter: 700 Hz
 - Full scale: 2 g for accelerometer, 2000 dps for gyroscope
 - [LIS2MDL](#) settings:
 - Low power mode
 - Output data rate: 50 Hz
 - Full scale: 50 gauss
- **APP DATA OUTPUT:**
 - Compass orientation model
 - Level indication model
 - Quaternion values
 - Heading
 - Euler angles
 - Plot collected data

3.1.4 Data Recorder app

Data recorder can be used to monitor and record movements and/or environmental conditions that parcels or objects are subjected to during movement or shipping.

The data can be used to verify whether a parcel has suffered shocks or undesirable temperatures that could damage the goods, or if a vehicle has been driven according to appropriate speed and safety parameters.

Certain sensors are enabled according to what is being monitored, and data is stored in the internal memory card for later retrieval and analysis. Motion sensors are set to Low Power Mode with a data rate of around 50 to 100 Hz, while a data rate of 1 Hz is appropriate for environmental sensors.

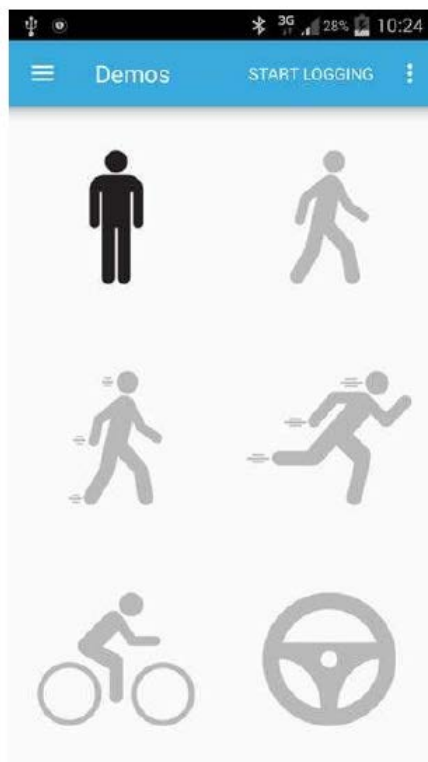
3.1.5 Human Activity Recognition app

The Human Activity Recognition app uses the [Sensortile.box LSM6DSOX](#) MEMS accelerometer and the embedded Machine Learning Core (MLC).

The following activities can be recognized by icons on the screen, independently from the [Sensortile.box](#) orientation:

- Stationary
- Walking
- Fast walking
- Jogging
- Biking
- Driving

Figure 24. Human Activity Recognition app



Four features are used (mean, variance, peak-to-peak, zero-crossing) for MLC code generation.

The algorithm runs at 26 Hz, with a window of 75 samples.

The Human Activity Recognition app works with the following ST high accuracy MEMS acceleration sensor and operating parameter settings:

- **APP DATA INPUT:** Accelerometer values
- **SENSOR USED:** [LSM6DSOX](#) (high bandwidth acceleration sensor)
- **SENSOR SETTINGS:**
 - Low power mode
 - Output data rate: 52 Hz
 - Low pass filter: 700 Hz
 - Full scale: 2 g
- **APP DATA OUTPUT :**
 - Activity recognized icon

3.1.6 In-Vehicle Baby Alarm app

The In-Vehicle Baby Alarm app combines the state of baby crying (see [Section 3.1.1](#)) and the vehicle movement detector.

The sensors used are the [MP23ABS1](#) analog MEMS microphone and the [LSM6DSOX](#) MEMS accelerometer, gyroscope, as well as the embedded Machine Learning Core.

For the MLC code generation, the following features calculated from accelerometer and gyroscope values have been used: MEAN-acc, VAR-acc, PeakToPeak-acc, MAX-acc, MEAN-gyro, VAR-gyro, PeakToPeak-gyro, MAX-gyro, MIN-gyro, ENERGY-gyro.

The app shows:

- whether the adult is in-vehicle or not
- whether the baby is crying or not
- the alarm icon if there is no adult in vehicle and the baby is crying

Figure 25. In-Vehicle Baby Alarm app - baby not crying state

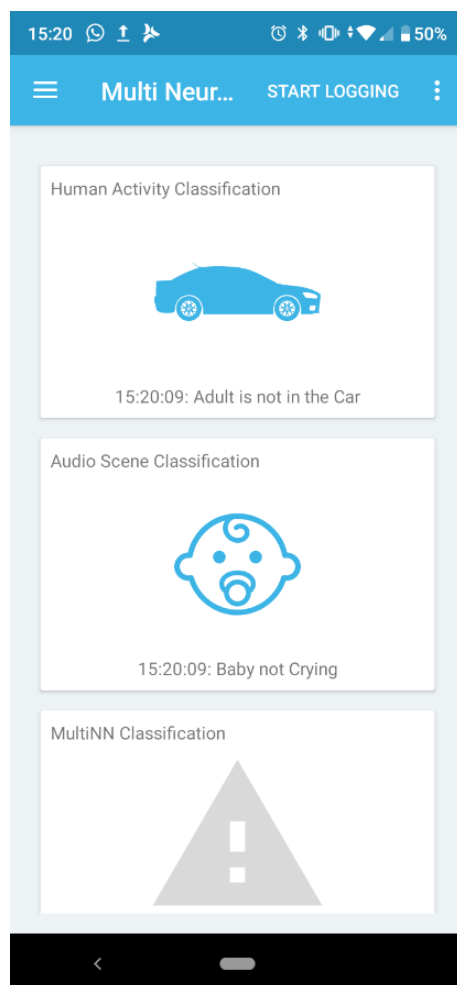
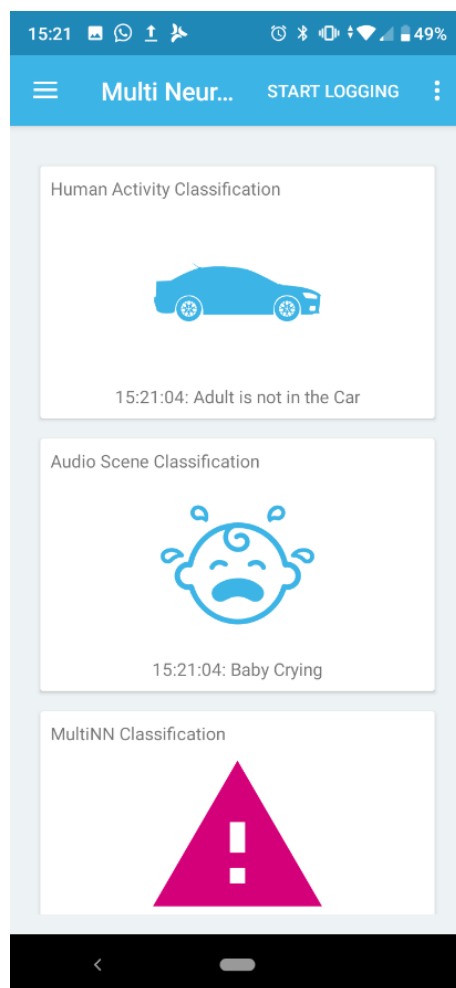


Figure 26. In-Vehicle Baby Alarm app - baby crying state



The In-Vehicle Baby Alarm app works with the following ST high accuracy MEMS acceleration and gyroscope sensor, analog MEMS microphone and the following operating parameter settings:

- **APP DATA INPUT:** accelerometer and gyroscope values, and microphone audio
- **SENSORS USED:**
 - [LSM6DSOX](#) (acceleration/gyroscope sensor)
 - [MP23ABS1](#) (analog MEMS microphone)
- **SENSOR SETTINGS:**
 - LSM6DSOX**
 - Low power mode
 - Output data rate: 52 Hz
 - Low pass filter: 700 Hz
 - Full scale: 2 g for accelerometer, 2000 dps for gyroscope
 - MP23ABS1 settings:**
 - 1600 Hz acquisition sample frequency
- **APP DATA OUTPUT:**
 - Baby crying/ not crying status
 - Adult is in-vehicle or not status
 - Alarm in case of baby-crying and adult is not in vehicle status

3.1.7

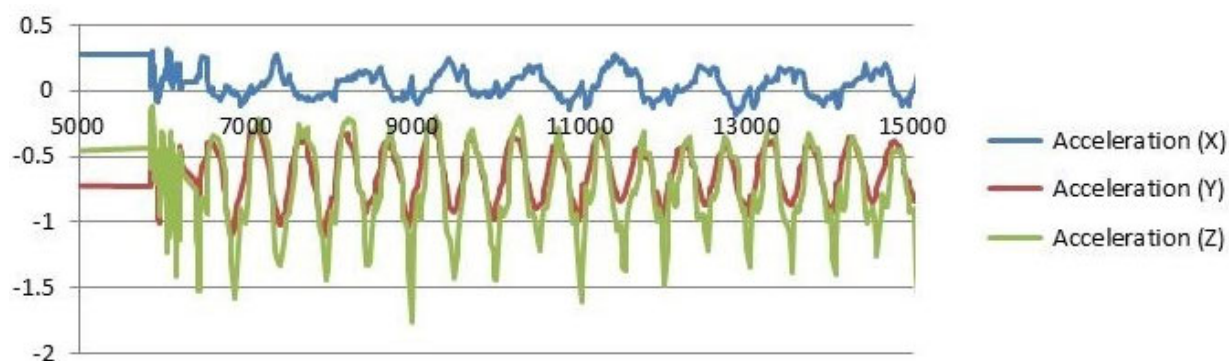
Pedometer app

The Pedometer app uses the pedometer software algorithm (MotionPM library) based on the [Sensortile.box](#) embedded [LSM6DSOX](#) 3-axis MEMS accelerometer data to count the steps and the steps per minute of your walking/running activity and show acquired data.

The 3-axis accelerometer measures the acceleration of your body during the walking.

The walking steps have a specific pattern of acceleration values and peak frequency, different from the body acceleration values pattern of other types of movements.

Figure 27. Raw accelerometer data - user walking with the device



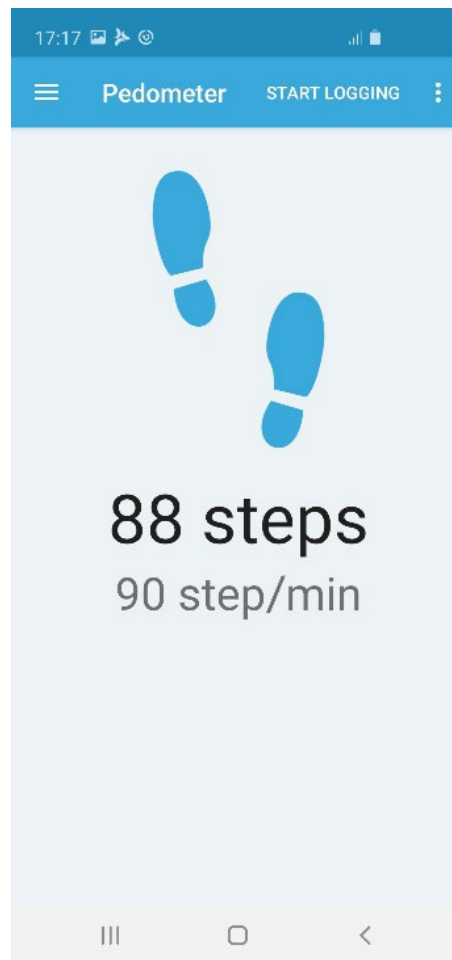
The algorithm is optimized for the [Sensortile.box](#) belt positioning as shown below.

Figure 28. Sensortile.box belt positioning



To avoid counting false positive steps, the step counting starts after 10 seconds of constant walking (debounce time); after this time, the algorithm shows the steps and continues counting from the number of steps already accumulated during the debounce time.

Figure 29. Pedometer app - step count



The accuracy of the Pedometer is Mean Absolute Percentage Error (MAPE)=97.5% with sigma=5 thanks to the high precision of the [LSM6DSOX](#) 3-axis MEMS accelerometer

These parameters are appropriate to capture human movement, filter unwanted noise and save battery energy to extend the potential working time.

The app works with the following ST high accuracy acceleration sensor and operating parameter settings:

- **APP DATA INPUT:** 3-axis acceleration values
- **SENSOR USED:** [LSM6DSOX](#) (accelerometer)
- **SENSOR SETTINGS:**
 - Low power mode accelerometer
 - Power down gyroscope
 - Output data rate: 52 Hz
 - Full scale: 4 g
- **APP DATA OUTPUT:**
 - Number of steps
 - Cadence (number of steps per minute)

3.1.8 Sensor Fusion app

The Sensor Fusion app is based on the sensor fusion firmware algorithm (MotionFX library) embedded in the MCU.

The app shows the orientation estimation of [Sensortile.box](#) in the 3D space.

The algorithm uses the [LSM6DSOX](#) iNemo 6-axis accelerometer and gyroscope data and the [LIS2MDL](#) 3-axis compensated magnetometer data as inputs (9-axis), and calculates the quaternion coefficient and the Euler angles to detect the right orientation of the [Sensortile.box](#) represented by a cube on the smartphone app or by value plots for all sensors and results calculated.

Figure 30. Sensor fusion app - Sensortile.box orientation (cube representation)



Figure 31. Sensor fusion app - Sensortile.box orientation (plot)

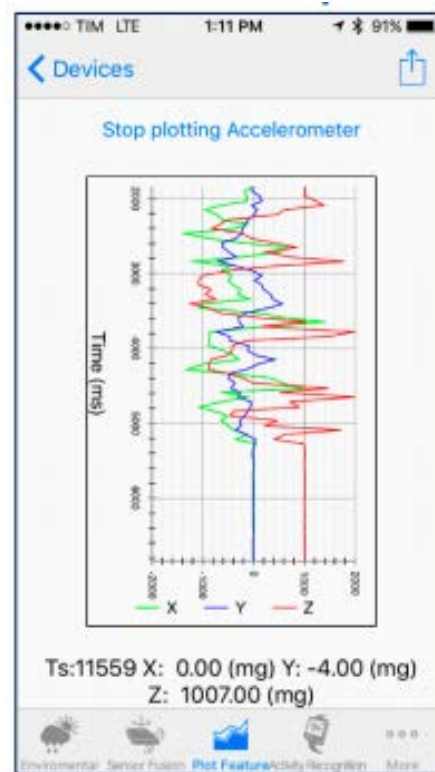
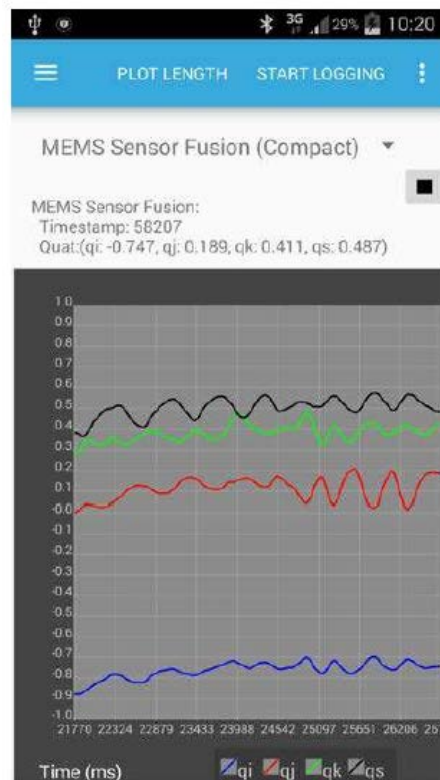


Figure 32. Sensor fusion app - Sensortile.box orientation (results)



To enable magnetometer calibration, you need to touch the symbol shown in the picture below.

Figure 33. Sensor fusion app - enabling magnetometer calibration



Move the board in an 8-pattern shape as shown in the figure below to calibrate the magnetometer: the calibration is completed when the icon turns green.

Figure 34. Sensor fusion app - enabling magnetometer calibration



The Sensor Fusion Motion FX library provides orientation estimation, magnetometer hard-iron offset compensation, accelerometer vibration rejection and gyroscope bias compensation.

The app works with the following ST high accuracy acceleration sensor and magnetometer, and operating parameter settings:

- **APP DATA INPUT:** accelerometer, gyroscope and magnetometer values
- **SENSORS USED:**
 - [LSM6DSOX](#) (high bandwidth acceleration sensor and gyroscope)
 - [LIS2MDL](#) (compensated magnetometer)
- **SENSOR SETTINGS:**
 - LSM6DSOX**
 - Low power mode
 - Output data rate: 52 Hz
 - Low pass filter: 700 Hz
 - Full scale: 2 g for accelerometer, 2000 dps for gyroscope
 - LIS2MDL**
 - Low power mode
 - Output data rate: 50 Hz
 - Full scale: 50 gauss
- **APP DATA OUTPUT:**
 - 3D-Cube orientation model
 - Quaternion values
 - Heading
 - Euler angles
 - Plot collected data

3.1.9 Vibration Monitoring

The Vibration Monitoring app demonstrates how engines, electric motors and the like are monitored to detect potential problems by their mechanical vibrations.

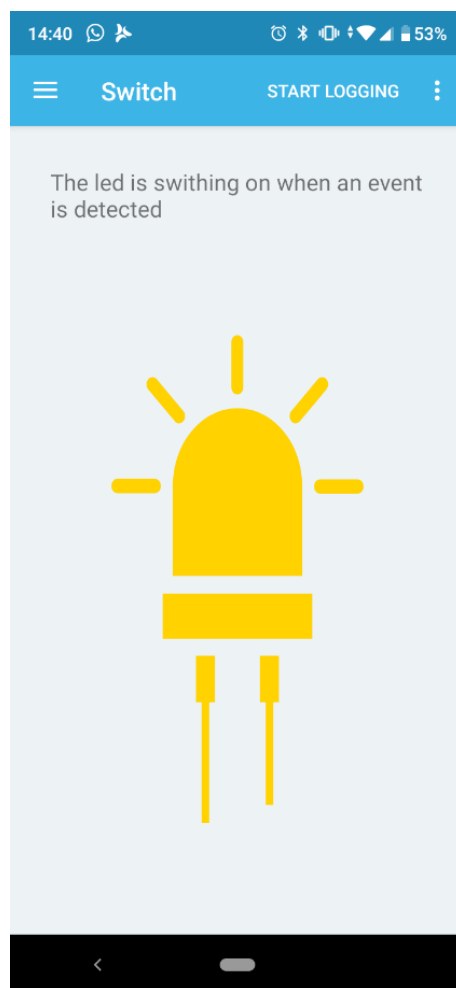
The sensor used is the [LSM6DSOX](#) accelerometer configured in high performance mode, with an output data rate of 6666 Hz, to reach the highest level of bandwidth and best performance.

The app can compare two mechanical vibration data patterns, the vibration under test (Compare-app) with the vibration of a standard scenario previously acquired (Training-app).

- **Vibration Monitor - Training:** is designed to acquire the vibration pattern of new or correctly functioning equipment. The vibration pattern is converted using the Fast Fourier Transform (FFT) function and is stored in the memory card on the [SensorTile.box](#) device.
- **Vibration Monitor - Compare:** is designed to monitor the same equipment and compare the vibration patterns with the original sample captured by Vibration Monitor - Training.

If the difference between the vibration analysis in Vibration Monitor - Training and Vibration Monitor - Compare exceeds a set delta parameter (which can be modified according to equipment age and load conditions), the green user LED on the [Sensortile.box](#) device and the LED icon on the smartphone screen turn on.

Figure 35. Vibration Monitoring app - event detection



The Vibration Monitoring apps work with the following ST high accuracy motion sensor and operating parameter settings:

- **APP DATA INPUT:** accelerometer values for training and compare phases
- **SENSORS USED:** [LSM6DSOX](#) high bandwidth acceleration sensor
- **SENSOR SETTINGS:**
 - High Performance power mode
 - Output data rate: 6666 Hz
 - Filter: none
 - Full scale: 2 g
- **APP DATA OUTPUT:**
 - Green LED on the [SensorTile.box](#)
 - LED icon on the smartphone screen

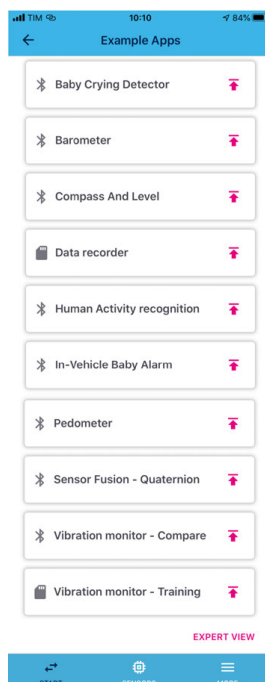
3.2 How to use Expert Mode functionality

The STE BLE Sensor app can help you develop your own app or customize an existing one, which you can then upload and run on the SensorTile.box device.

Step 1. Return to the main screen of the ST BLE Sensor app.

Step 2. Select **[CREATE A NEW APP]**.

Figure 36. Example Apps screen



Step 3. Select **[EXPERT VIEW]**.
A new screen appears with saved apps.

Step 4. Select [+ NEW APP].

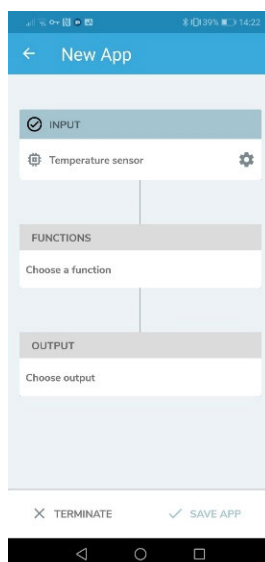
Figure 37. Input sources screen



Step 5. Select one or more of the desired sensor data inputs.
Unselected sensors are put in sleep mode.

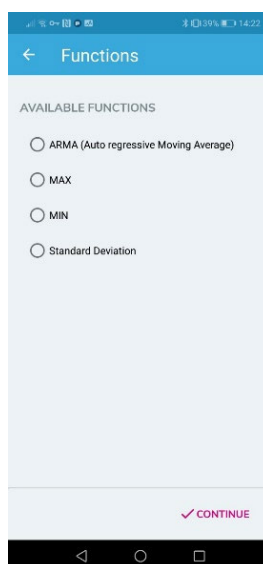
Step 6. Select [SET INPUT] to confirm

Figure 38. Sensor data configuration screen



- Step 7.** Select the gear icon next to each sensor and set the parameters according to your application requirements
- You can set parameters such as full scale, data rate (ODR), Power Mode, Filter, etc., according to device specifications provided in corresponding sensor datasheets.
- Following sensor selection, the function screen lists the available functions for the enabled sensors. For the temperature sensor, for example, the available functions are shown below.

Figure 39. Custom app function screen



- Step 8.** Choose between one of the following output types:
- Via Bluetooth to your smartphone (to view certain data)
 - To the memory card (micro SD)
 - Via USB to a host master (i.e., a PC).
 - To the user LED for logic data types (like the output of a threshold function or comparison).
- The LED option is achieved by selecting [**Save as EXP**] in the output selection screen and enabling the associated output property.
- There are two special output types:
- [**Save as INPUT**]: is a way to concatenate different functions and generate different branches which will be processed one after the other.
 - [**Save as EXP**]: produces an app branch whose output is a digital “true” or “false”. This value can be used in other comparisons or logic functions.
- An app saved as EXP or as INPUT appears in the input selection screen so it can be used more complex app generation.
- Step 9.** Save your app with an appropriate name and optional comment.

RELATED LINKS

[Appendix A ARMA filter coefficient calculation on page 35](#)

3.3 Pro Mode

SensorTile.box is fully compatible with the STM32 Open Development Environment (STM32 ODE) for developers to customize the SensorTile.box firmware. In fact, you can use the STM32Cube function packs [FP-SNS-STBOX1](#), [FP-SNS-ALLMEMS2](#) and [FP-AI-SENSING1](#)

The board is compatible with STLINK-V3 and [STLINK-V3MINI](#) (with UART pins for debugging), and the most recent release of the Sensortile.box includes an adapter board and cable for the [ST-LINK/V2](#) programming and debugging device.

Important:

You must use STLINK-V3 (or [STLINK-V3MINI](#)) and the corresponding level shifter if you require Cube.AI library compatibility with the debugging option.

RELATED LINKS

[Visit the ST website for all the resources you need regarding the STM32 Open Development Environment](#)

Appendix A ARMA filter coefficient calculation

The built-in ARMA filter implemented by SensorTile.box firmware is a general IIR fifth-order polynomial filter described by the equation:

$$y(t) = \frac{ma(0)u(t) + ma(1)u(t-1) + ma(2)u(t-2) + ma(3)u(t-3) + ma(4)u(t-4) + ma(5)u(t-5)}{1 + ar(1)y(t-1) + ar(2)y(t-2) + ar(3)y(t-3) + ar(4)y(t-4) + ar(5)y(t-5)}$$

Where:

y(t) = output of the filter

u(t) = input signal

With this function, low-pass, high-pass, band-pass and band-reject filters can be implemented, and higher filter orders can be obtained by cascading two or more filters, one after the other.

The simplest way to calculate the ma(i) and ar(i) coefficients for the required filter shape is to use a math program like Octave. Octave has a “signal” extension package that can be loaded by typing the command `pkg load signal` at the Octave prompt. Once done, there are few filter calculation options, depending on the type of filter that is requested by the application: Butterworth, Bessel, Chebyshev and elliptic (Cauer) filters can be computed.

RELATED LINKS

[3.1.1 Baby crying detector on page 7](#)

[3.2 How to use Expert Mode functionality on page 31](#)

[Visit this web page for further insight regarding ARMA filters](#)

[GNU Octave home page](#)

A.1

Filter calculation example

The following example illustrates how a second-order Butterworth band-pass filter can be implemented. We will assume that we want to filter our microphone signal with a band-pass filter in the 1 kHz – 3 kHz range.

- Step 1.** We set a microphone sampling rate of 16 kHz.
The maximum signal frequency (or Nyquist frequency) is therefore 16/2 = 8 kHz, according to Nyquist/Shannon theorem.
- Step 2.** Open the Octave command line prompt.
- Step 3.** Type the following command: `>> [MA, AR]=butter(2, [1/8, 3/8])`
This calls the butter function in Octave, where:
- 2 is the filter order
 - 1/8 and 3/8 are the band limits relative to the Nyquist frequency

The program output is:

```
MA =
0.09763    0.00000   -0.19526    0.00000    0.09763
AR =
1.00000   -2.25233    2.27614   -1.23184    0.33333
```

- Step 4.** Set the above values for ma(0) to ma(4), and set ma(5) to zero in the ARMA property screen for the SensorTile.box app.
- Step 5.** Set the above values for ar(0) to ar(4), and set ar(5) to zero in the ARMA property screen for the SensorTile.box app.
Note that ar(0) is always equal to 1, so the ARMA property screen does not require it to be inserted.

RELATED LINKS

[Similar functions can be used for the other type of filters; check Octave documentation for all the options](#)

Appendix B Formal notices required by the U.S. Federal Communications Commission ("FCC")

FCC NOTICE: This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Changes or modifications not expressly approved by the manufacturer could void the user's authority to operate the equipment.

Additional warnings for FCC

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference's by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

B.1 SPBTLE-1S Bluetooth communication module specifications

Operating frequency: 2402 MHz to 2480 MHz

Output power: around 4 dBm (approx.) at ambient temperature

Table 1. SPBTLE-1S Bluetooth module power specifications

Test conditions		Channel		
		Low 2.402 GHz	Middle 2.440 GHz	High 2.480 GHz
Temperature	Voltage	Measured equivalent isotropic power (dBm)		
+25 °C	3.3 V	3.31	4.48	4.30
-40 °C	3.3 V	3.96	5.16	4.99
+85 °C	3.3 V	2.71	3.78	3.58

Revision history

Table 2. Document revision history

Date	Version	Changes
13-May-2019	1	Initial release.
14-Nov-2019	2	<p>Minor text edits</p> <p>Added information regarding ST-LINK/V2 adapter board and cable included in latest version of the kit.</p> <p>Updated Introduction.</p> <p>Updated Section 1 How to set up the hardware.</p> <p>Updated Section 3.1.4 Baby crying app.</p> <p>Added Section Appendix B Formal notices required by the U.S. Federal Communications Commission ("FCC").</p>
02-Dec-2019	3	<p>Minor text edits.</p> <p>Added Section B.1 SPBTLE-1S Bluetooth communication module specifications.</p>
15-Apr-2020	4	<p>Updated Figure 4. Example Apps screen, Section 3.1 Entry level example apps, Section 3.1.1 Baby crying detector, Section 3.1.2 Barometer app, Section 3.1.3 Compass and Level app, Section 3.1.4 Data Recorder app, Section 3.1.7 Pedometer app, Section 3.1.9 Vibration Monitoring and Section 3.3 Pro Mode.</p> <p>Added Section 3.1.5 Human Activity Recognition app, Section 3.1.6 In-Vehicle Baby Alarm app and Section 3.1.8 Sensor Fusion app.</p>
13-May-2020	5	<p>Updated Section 3.1.1 Baby crying detector, Section 3.1.2 Barometer app, Section 3.1.3 Compass and Level app, Section 3.1.5 Human Activity Recognition app, Section 3.1.6 In-Vehicle Baby Alarm app, Section 3.1.7 Pedometer app, Section 3.1.8 Sensor Fusion app and Section 3.1.9 Vibration Monitoring.</p>
06-Apr-2021	6	<p>Updated Introduction.</p> <p>Minor text changes.</p>

Contents

1	How to set up the hardware	2
2	How to use ST BLE Sensor app with SensorTile.box	3
3	Application descriptions	6
3.1	Mode 1 example apps	6
3.1.1	Baby crying detector	7
3.1.2	Barometer app	9
3.1.3	Compass and Level app	13
3.1.4	Data Recorder app	20
3.1.5	Human Activity Recognition app	20
3.1.6	In-Vehicle Baby Alarm app	21
3.1.7	Pedometer app	24
3.1.8	Sensor Fusion app	25
3.1.9	Vibration Monitoring	29
3.2	How to use Expert Mode functionality	31
3.3	Pro Mode	33
Appendix A	ARMA filter coefficient calculation	35
A.1	Filter calculation example	35
Appendix B	Formal notices required by the U.S. Federal Communications Commission ("FCC")	36
B.1	SPBTLE-1S Bluetooth communication module specifications	36
	Revision history	37

List of figures

Figure 1.	STEVAL-MKSBOX1V1 (SensorTile.box) multi-sensor development kit	1
Figure 2.	STEVAL-MKSBOX1V1 battery connection	2
Figure 3.	ST BLE Sensor app main screen	3
Figure 4.	Example Apps screen	4
Figure 5.	Board selection.	5
Figure 6.	Apps screen.	6
Figure 7.	FFT analysis - from time to frequency domain.	7
Figure 8.	FFT analysis - principles	7
Figure 9.	STBLESensor - baby crying detection process	8
Figure 10.	STBLESensor - baby crying app neural network	8
Figure 11.	Barometer app screen	9
Figure 12.	Environmental screen	10
Figure 13.	Plot Data screen - humidity	11
Figure 14.	Plot Data screen - temperature	12
Figure 15.	MotionFX algorithm flow	13
Figure 16.	MotionFX algorithm flow - magnetometer function	14
Figure 17.	MotionFX algorithm flow - accelerometer tilt compensation	14
Figure 18.	MotionFX algorithm flow - gyroscope function.	14
Figure 19.	STBLESensor Compass and Level app - calibration icon	15
Figure 20.	STBLESensor Compass and Level app - starting the calibration	16
Figure 21.	STBLESensor Compass and Level app - orientation screen	17
Figure 22.	STBLESensor Compass and Level app - offset example 1	18
Figure 23.	STBLESensor Compass and Level app - offset example 2	19
Figure 24.	Human Activity Recognition app	21
Figure 25.	In-Vehicle Baby Alarm app - baby not crying state	22
Figure 26.	In-Vehicle Baby Alarm app - baby crying state	23
Figure 27.	Raw accelerometer data - user walking with the device	24
Figure 28.	Sensortile.box belt positioning	24
Figure 29.	Pedometer app - step count	25
Figure 30.	Sensor fusion app - Sensortile.box orientation (cube representation)	26
Figure 31.	Sensor fusion app - Sensortile.box orientation (plot)	27
Figure 32.	Sensor fusion app - Sensortile.box orientation (results)	27
Figure 33.	Sensor fusion app - enabling magnetometer calibration	28
Figure 34.	Sensor fusion app - enabling magnetometer calibration	28
Figure 35.	Vibration Monitoring app - event detection	30
Figure 36.	Example Apps screen	31
Figure 37.	Input sources screen	32
Figure 38.	Sensor data configuration screen	32
Figure 39.	Custom app function screen	33

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2021 STMicroelectronics – All rights reserved