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

Introduction

The STEVAL-MKBOXPRO (SensorTile.box PRO) 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 with a rechargeable battery. You can connect with your smartphone to the board via Bluetooth by using the ST BLE Sensor Classic app (available on Google Play) and immediately build your own apps through a special interface that offers beginner and expert level functionality. This multisensor kit therefore allows you to design wireless IoT and wearable sensor applications quickly and easily, without performing any programming.

Sensortile.box PRO 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 ultralow power programmable Bluetooth® Low Energy wireless SoC solution BlueNRG-LP 355AC, that is compliant with Bluetooth Low Energy v5.2. This transmitter module is FCC (ID:S9N-MKBOXPRO) certified and IC (IC: 8976C-MKBOXPRO; PMN: STEVAL-MKBOXPRO; HVIN: STEVAL-MKBOXPRO; FVIN: STSW-MKBOXPRO-BL) certified.

Figure 1. STEVAL-MKBOXPRO (Sensortile.box PRO) multisensor development kit

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

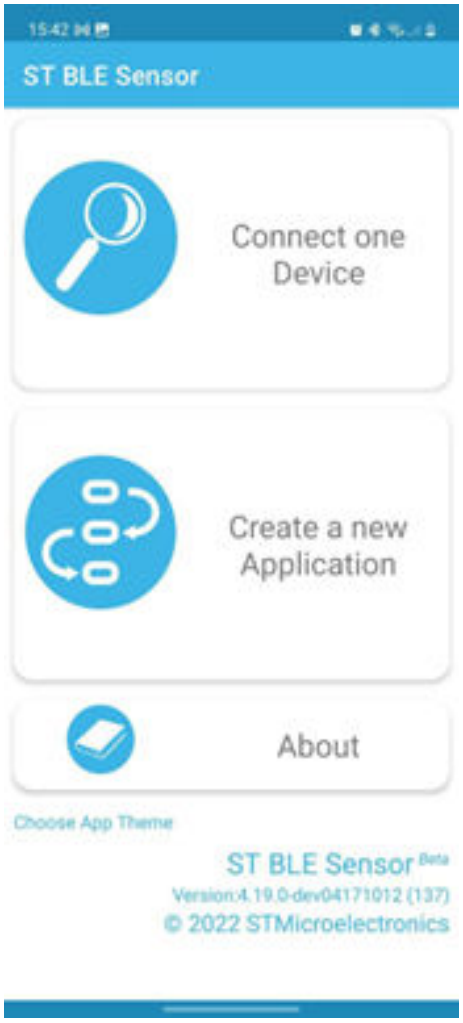
You need to download and install our ST BLE Sensor Classic app on your smartphone. The app is available from the Google and Apple online stores.

The easiest way to use the Sensortile.boxPRO is to upload one of the example apps available in the ST BLE Sensor Classic app.

Here is a quick overview of the operations to be performed.

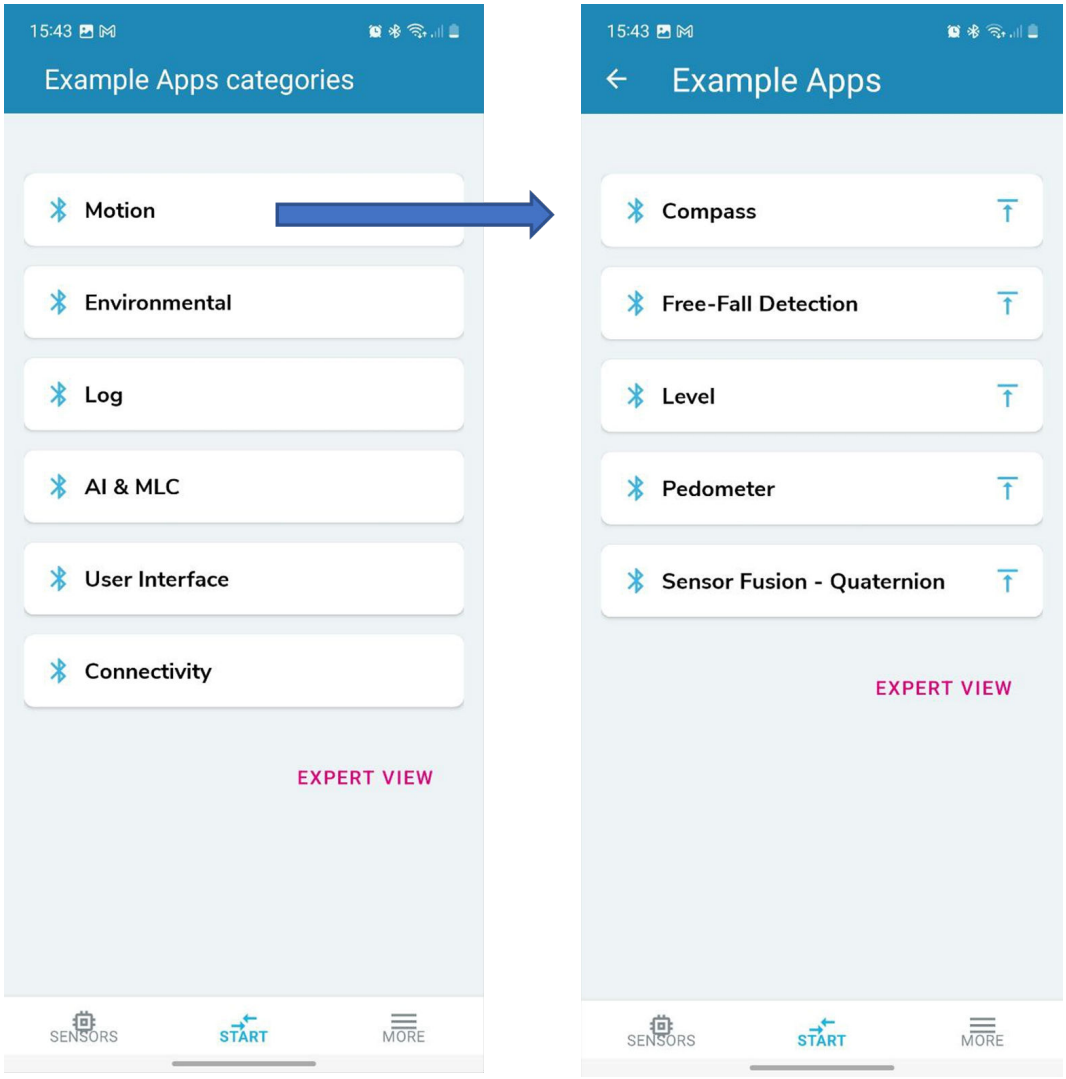
Step 1. Launch the app on your smartphone.

Figure 2. ST BLE Sensor Classic app main screen



Step 2. Select [**CREATE A NEW APP**], then choose the Sensortile.boxPRO as the board you want to program. The Example Apps screen that follows lists the preloaded apps that you can use immediately, organized in different sections.

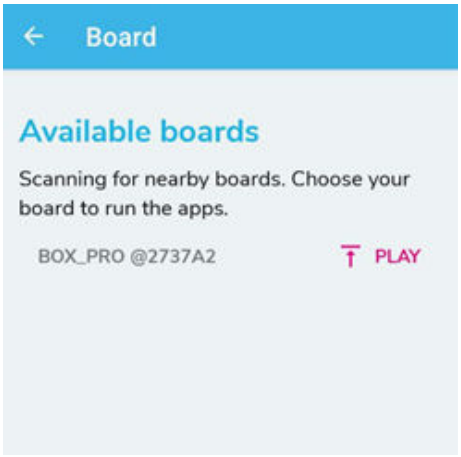
Figure 3. Example Apps screen



Step 3. Select one of the apps with the icon from the list.

After you select the app, ST BLE Sensor Classic will scan for available Sensortile.boxPRO devices in range.

Figure 4. Board selection



Step 4. Select the appropriate Sensortile.boxPRO device from the Board screen.

A blue LED on the SensorTile.box device will turn on to confirm Bluetooth pairing.
A pop up message in ST BLE Sensor Classic will prompt you to confirm loading the new app in replacement of any previously opened apps.

Step 5. Select autoconnect or return to the Device List and select the appropriate Sensortile.boxPRO device.

The app will commence monitoring or logging activity and return real time feedback data to the corresponding app screen in ST BLE Sensor Classic.

2 Entry mode application descriptions

2.1 Example apps overview

The ST BLE Sensor Classic bundles the following ready-to-use application scenarios:

- Motion
  - Compass
  - Level
  - Pedometer
  - Sensor fusion - quaternion
- Environmental
  - Barometer
- Log
  - Data recorder
- AI and MLC
  - Human activity recognition
- User interface
  - Qtouch
- Connectivity
  - NFC tag

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 microSD card.

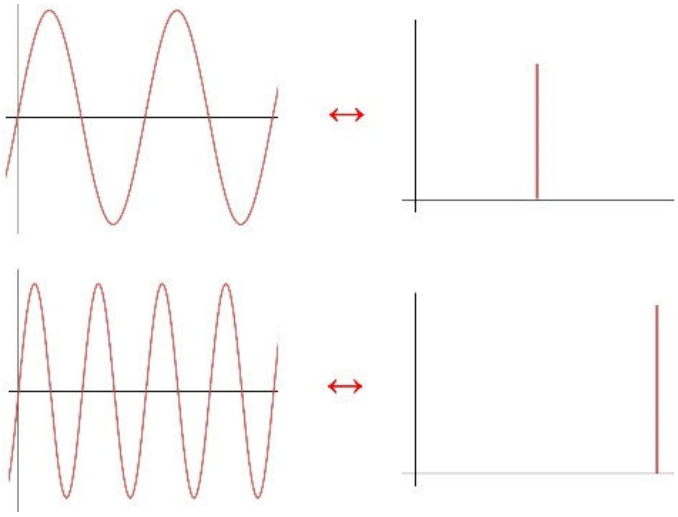
App scenarios with the icon are reserved for Expert mode.

2.1.1 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.boxPRO on-board microphone.

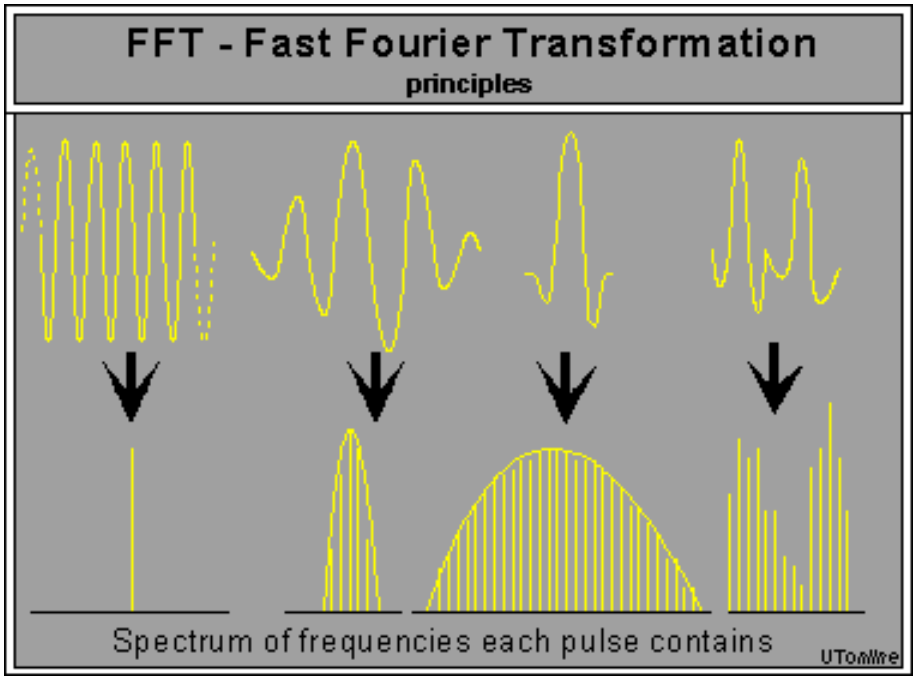
The analysis of the acquired audio is based on the FFT that converts a signal from its original time domain to a representation in the frequency domain.

Figure 5. FFT analysis - from time to frequency domain



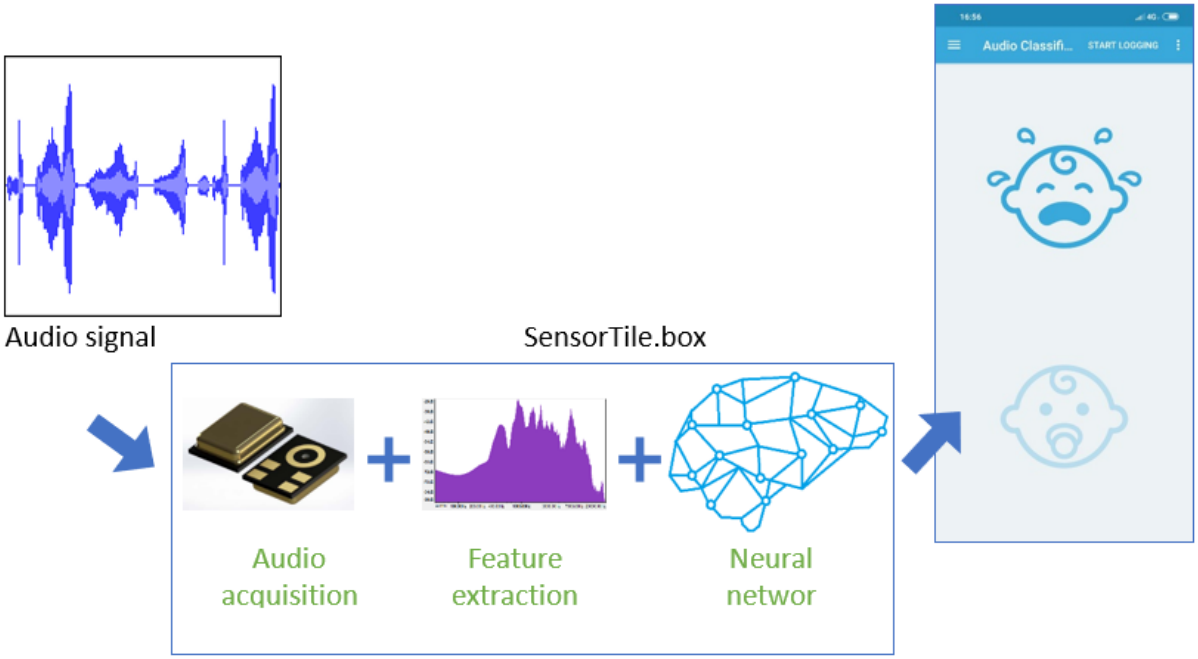
The FFT of the audio signal is the result of all the contributions of each frequency and the related magnitude factor generated by the audio signal.

Figure 6. 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), along with other parameters that are processed by the implemented MCU neural network: if a baby crying event is detected, a warning is sent to the smartphone via Bluetooth.

Figure 7. ST BLE Sensor Classic - baby crying detection process

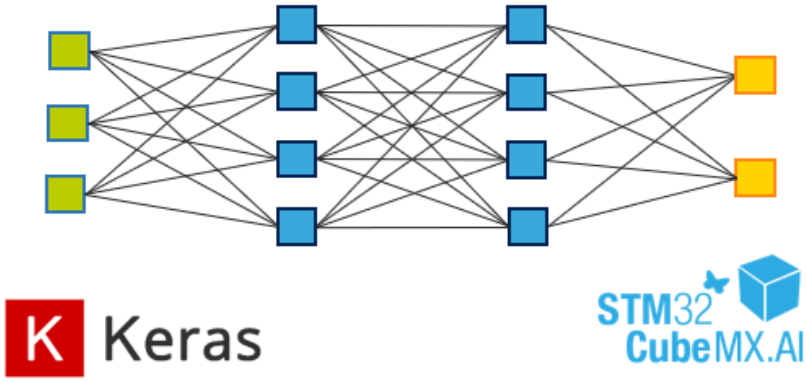


The neural network is classified as a deep feed forward neural network and its structure is composed of 2 hidden nodes of 100 neurons each.

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.boxPRO has been performed using STM32CubeMX.AI.

Figure 8. ST BLE Sensor Classic - baby crying app neural network



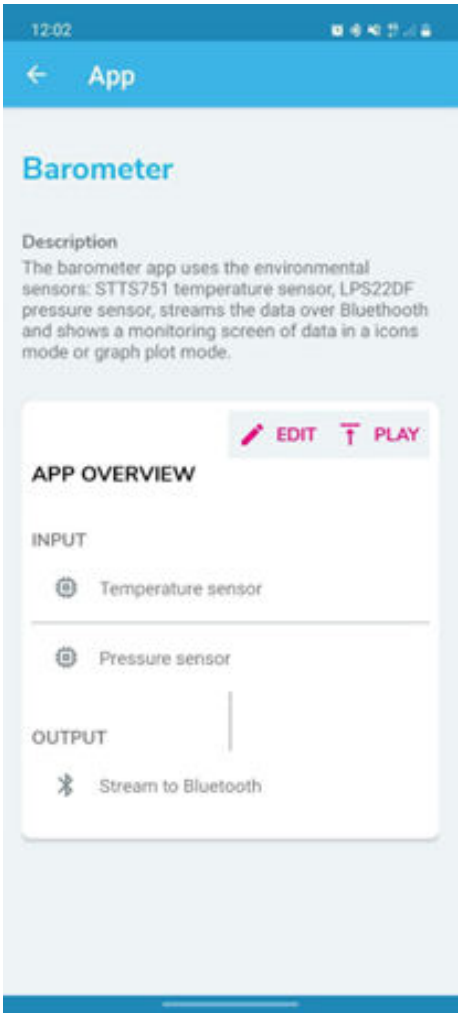
The baby crying app works with the following ST high sensitivity audio sensor and operating parameter settings:

- **APP DATA INPUT**: microphone audio acquisition
- **SENSOR USED**: MP23DB01HP MEMS microphone
- **SENSOR SETTINGS**: 16 KHz sample acquisition
- **APP DATA OUTPUT**: baby crying/not crying icon

2.1.2 Barometer app

The Barometer app uses the SensorTile.boxPRO on-board environmental sensors (STTS22H and LPS22DF).

Figure 9. **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 LPS22DF pressure sensor embeds another built-in sensor for temperature compensation (±0.5 mBar precision with a range of 260-1260 mBar of absolute pressure).

The STTS22H digital temperature sensor has an operating temperature range of -40/+125 °C, with precision of ±0.5 °C (typ.). The temperature shown is the one of the board.

When you run the Barometer app and connect the SensorTile.boxPRO device, the ST BLE Sensor Classic app shows a monitoring screen for the environmental sensors.

Figure 10. **Environmental screen**

It is possible to switch between °C and °F with the toggle in the upper right part of the temperature section.
An offset can be defined for every sensor in the dedicated window. If changed, it will be applied to the data shown.

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

Figure 11. **Plot Data screen - humidity**

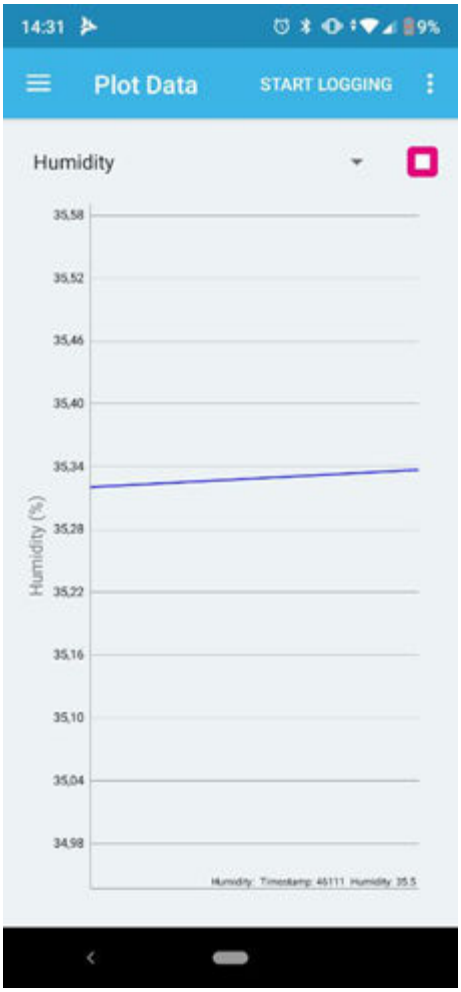
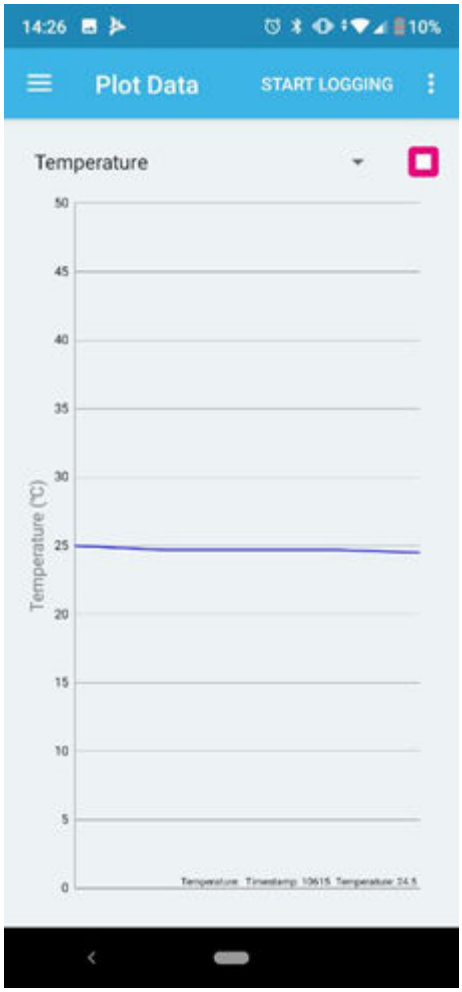


Figure 12. **Plot Data screen - temperature**



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 and temperature
- **SENSORS USED**:
  - LPS22DF (absolute pressure MEMS digital sensor)
  - STTS22H (temperature digital sensor)
- **SENSOR SETTINGS**:
  - LPS22DF settings:
    - Power mode: Low Noise
    - Output Data Rate: 1 Hz
    - Filter: ODR/2
  - STTS22H settings:
    - Low power mode
    - Output data rate: 1 Hz
- **APP DATA OUTPUT**:
  - Temperature (°C)
  - Absolute Pressure (mBar)
  - Plot collected data

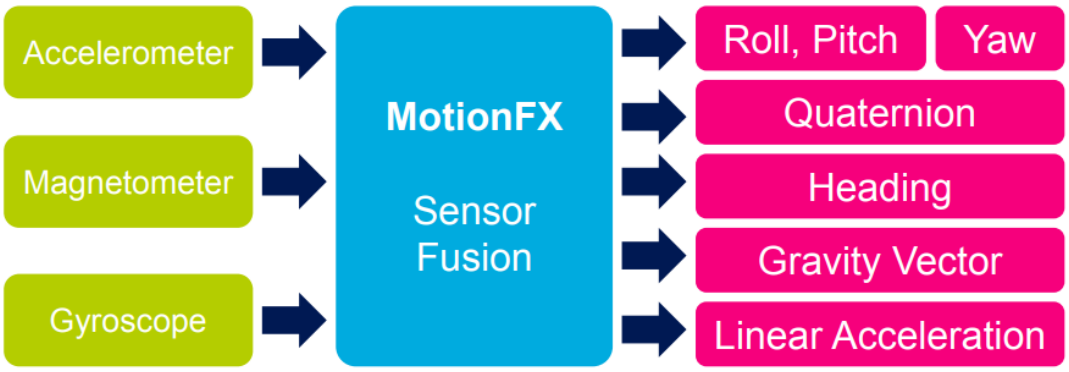
2.1.3 Compass app

The Compass app shows the orientation estimation of the SensorTile.boxPRO in relation to the Earth magnetic North.

The app is based on the sensor fusion firmware algorithm (MotionFX library) embedded in the SensorTile.boxPRO 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).

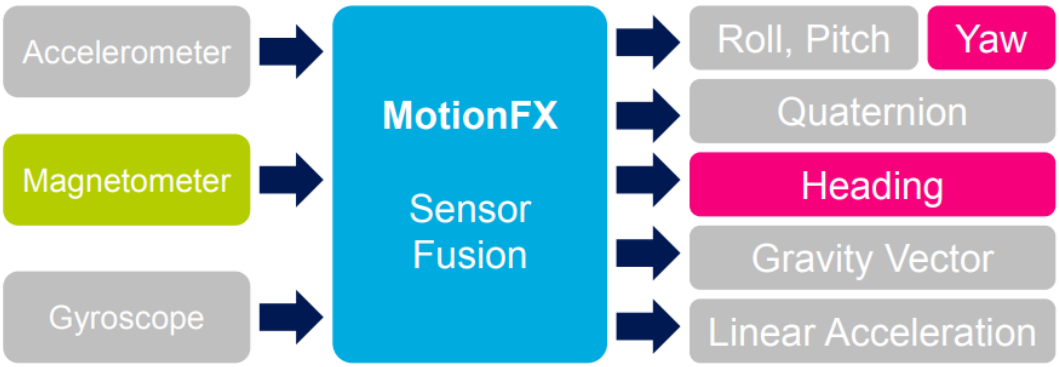
Figure 13. MotionFX algorithm flow



The algorithm uses the LSM6DSV16X 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.

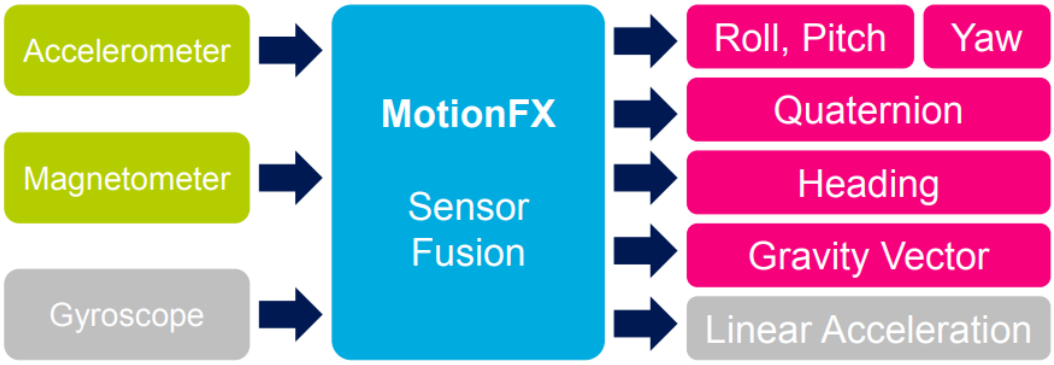
The magnetometer indicates yaw angle and heading, but only if hard-iron offset is compensated and when there is no additional magnetic field around the SensorTile.boxPRO disturbing the measurement.

Figure 14. MotionFX algorithm flow - magnetometer function



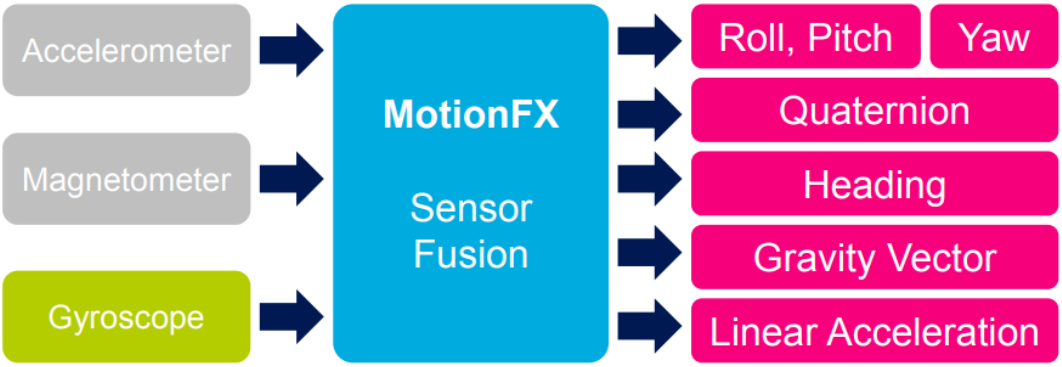
Data regarding yaw and angle heading are also given when tilt is compensated by the accelerometer.

Figure 15. MotionFX algorithm flow - accelerometer tilt compensation



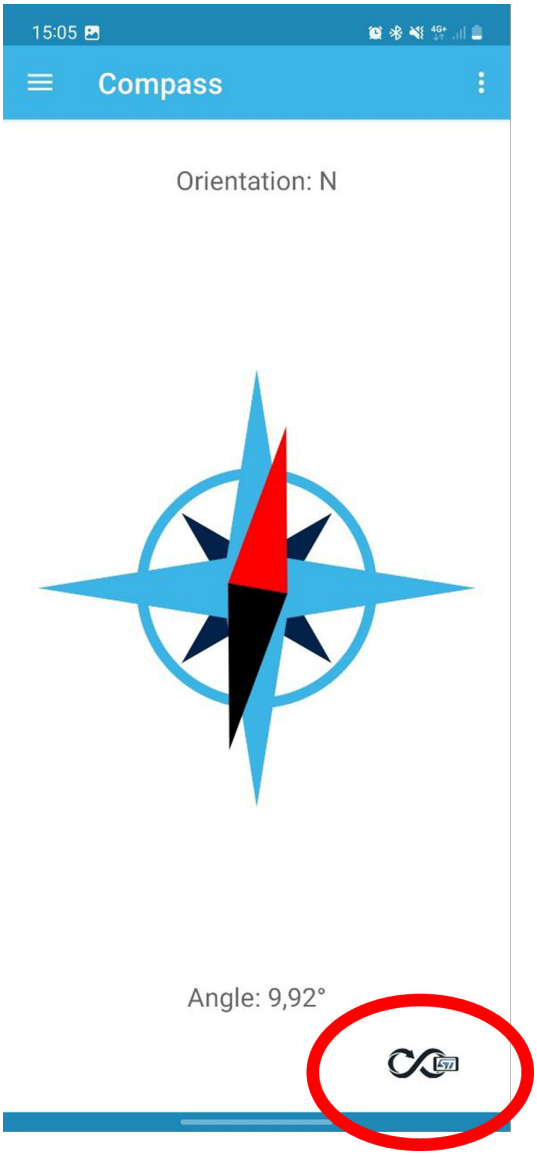
The gyroscope indicates the new orientation based on the previous one, when its bias is compensated by the accelerometer. The gyroscope can detect the static condition of the SensorTile.boxPRO.

Figure 16. MotionFX algorithm flow - gyroscope function



To enable the magnetometer calibration, you need to touch the symbol highlighted in the picture below, in the bottom right corner of the screen.

Figure 17. ST BLE Sensor Classic Compass app - calibration icon



Then, move the SensorTile.boxPRO in a 8-pattern figure as shown below; the calibration is completed when the icon becomes green.

Figure 18. STBLESensor Compass app - starting the calibration

In this screen, pitch and roll can be viewed both together and singularly. The set zero button can be used to set the zero level, to better measure angles on askew surfaces. The reset zero button can be used to return to the default orientation of the board, with the Z axis of the accelerometer pointing up.

The 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**:
  - LSM6DSV16X (acceleration and gyroscope sensor)
- **SENSOR SETTINGS**:
  - LSM6DSV16X:
    - Low power mode
    - Output data rate: 60 Hz
    - Full scale: 2 g for accelerometer
- **APP DATA OUTPUT**:
  - Level orientation model (pitch and roll)

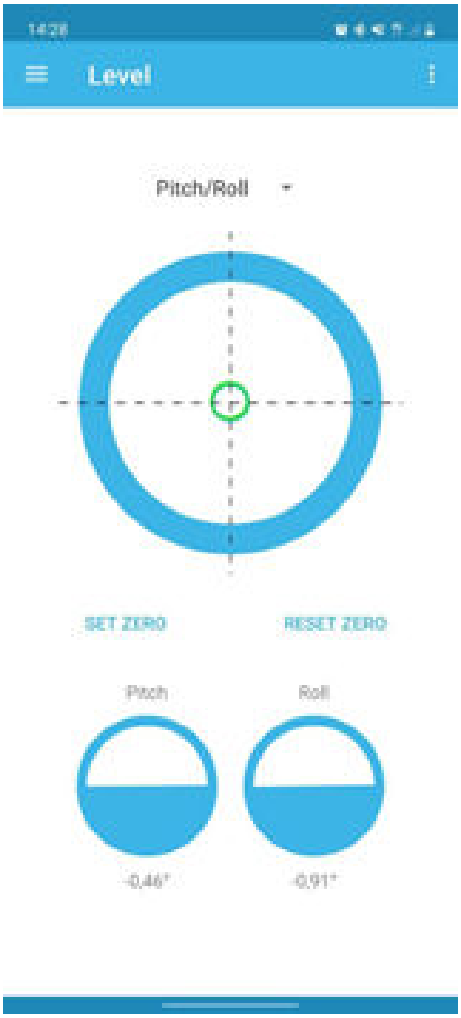
2.1.4 Level app

The Level app shows the level indication estimation of the SensorTile.boxPRO in relation to the Earth surface.

The app is based on the accelerometer sensor inside the LSM6DSV16X, used to extract the pitch and the roll value using simple trigonometry formulas. This demonstrates that the accelerometer is able to detect the level estimation alone, without using the more power consuming Sensor Fusion algorithm.

The picture below shows the main screen of the Level app, that is viewable by sliding right when connecting to the app (the first screen presented to the user is the plot view of the pitch and roll values).

Figure 19. ST BLE Sensor Level app



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**:
  - LSM6DSV16X (acceleration sensor)
- **SENSOR SETTINGS**:
  - LSM6DSV16X:
    - Low power mode
    - Output data rate: 60 Hz
    - Full scale: 2 g for accelerometer
- **APP DATA OUTPUT**:
  - Level orientation model (pitch and roll)

2.1.5 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.

When connecting to the board, a screen appears for a split second before the main screen of this app appears. This screen pops up due to compatibility with other boards and is not important for this application.

Figure 20. SD Logging app main page

In the main screen there is the possibility to start the log with the button in the bottom right corner of the screen.

 is for starting the log, while  is for stopping it.

The application keeps logging even if the board is disconnected from the phone, even if, after disconnecting and reconnecting, the icon shown is the one for starting the log (). In this case the user must click the icon once to synchronize the board with the ST BLE Sensor Classic app, and click again to stop the log.

The log is automatically saved once every minute, to avoid losing too much data if the battery runs out.

The data displayed in the .csv file is:

- RTC(date)
- RTC(time)
- Accelerometer sensor (LSM6DSV16X)
- Gyroscope sensor (LSM6DSV16X)
- Magnetic field sensor (LIS2MDL)
- Temperature sensor (STTS22H)
- Pressure sensor (LPF22DF)

The default name of the output file saved on the SDcard is output.csv.

Date and time are reset at every reset of the board, so it is necessary to synchronize them to have the right timestamp available in the log.

There are two methods: the first is uploading a new app and start it without resetting the board, every time an app is uploaded the timestamp is synchronized, the other is going to the Board Configuration screen (from the menu selectable in the upper left corner) and click on [**Set Time**] and [**Set Date**] to synchronize them.

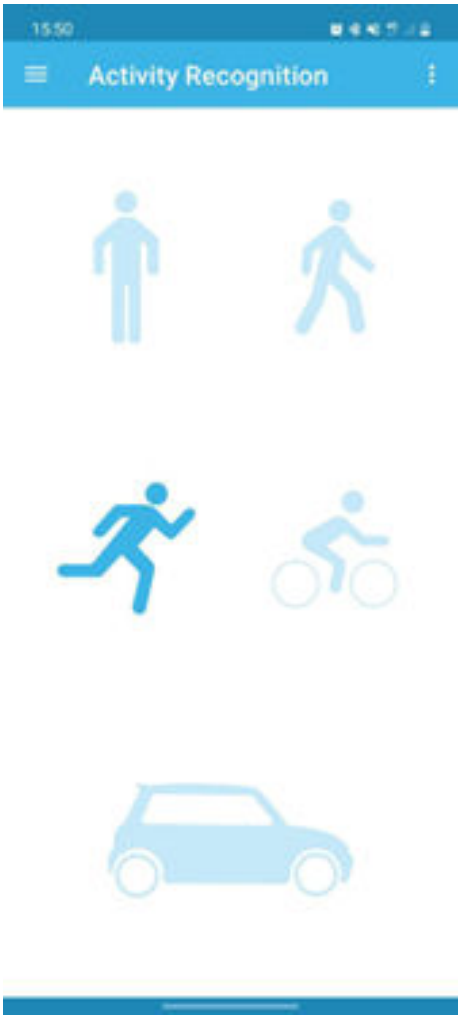
2.1.6 Human Activity Recognition app

The Human Activity Recognition app uses the SensorTile.boxPRO LSM6DSV16X MEMS accelerometer and the embedded machine learning core (MLC).

The following activities can be recognized by icons on the screen, regardless of the orientation of the SensorTile.boxPRO orientation:

- Stationary
- Walking
- Jogging
- Biking
- Driving

Figure 21. **Human Activity Recognition app**



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

The algorithm runs at 30 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**: LSM6DSV16X (high bandwidth acceleration sensor)
- **SENSOR SETTINGS**:
  - Low power mode
  - Output data rate: 60 Hz
  - Full scale: 2 g, 2000 dps
- **APP DATA OUTPUT** :
  - Activity recognized icon

2.1.7 In-Vehicle Baby Alarm app

The In-Vehicle Baby Alarm app combines the state of baby crying (see Section 2.1.1) and the vehicle movement detector.

The sensors used are the MP23DB01HP digital MEMS microphone and the LSM6DSV16X MEMS accelerometer, gyroscope, and 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 22. In-Vehicle Baby Alarm app - baby not crying state

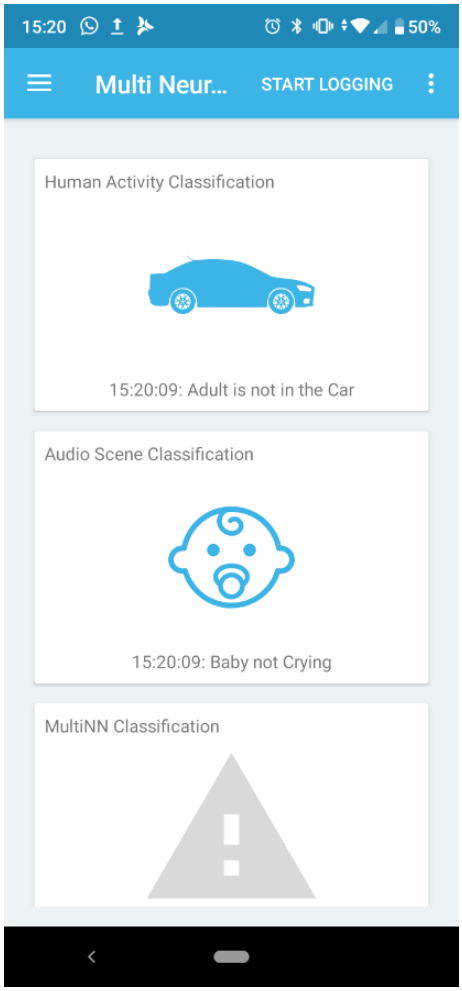
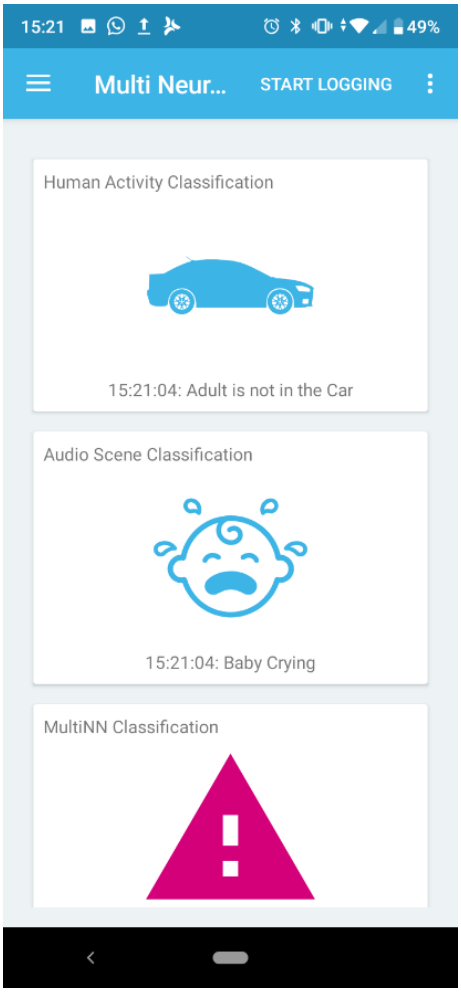


Figure 23. **In-Vehicle Baby Alarm app - baby crying state**



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

- **APP DATA INPUT**: accelerometer and gyroscope values, and microphone audio
- **SENSORS USED**:
  - LSM6DSV16X (acceleration/gyroscope sensor)
  - MP23DB01HP (analog MEMS microphone)
- **SENSOR SETTINGS**:
  LSM6DSV16X
  - Low power mode
  - Output data rate: 60 Hz
  - Full scale: 2 g for accelerometer, 2000 dps for gyroscope

  MP23DB01HP settings:
  - 16 kHz 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

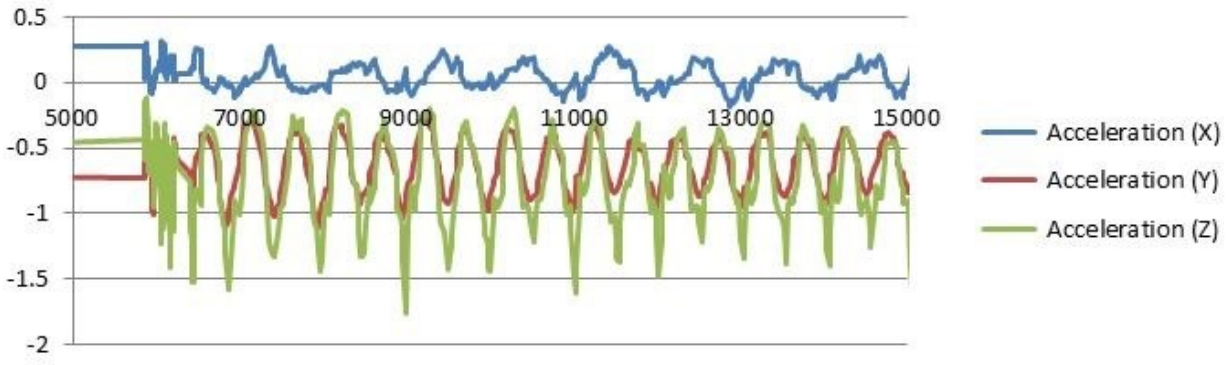
2.1.8 Pedometer app

The Pedometer app uses the pedometer software algorithm (MotionPM library) based on the SensorTile.boxPRO embedded LIS2DU12 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 24. Raw accelerometer data - user walking with the device

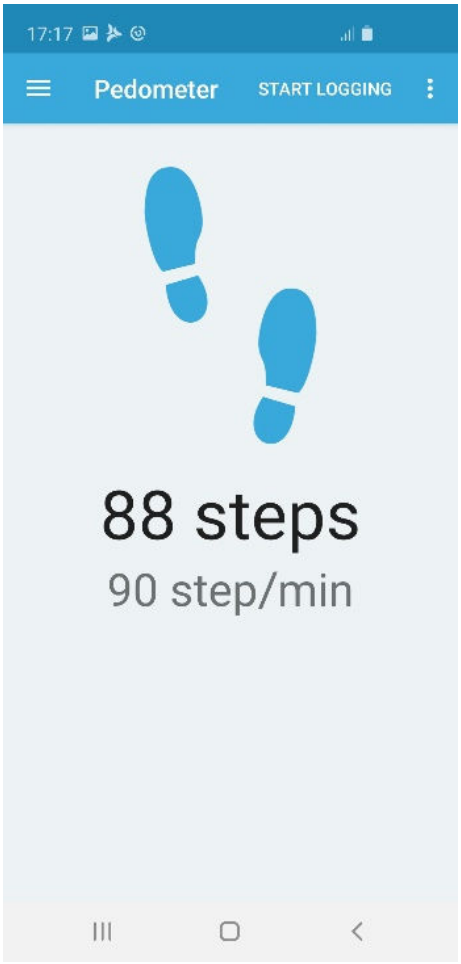


The algorithm is optimized for the SensorTile.boxPRO belt positioning as shown below.

Figure 25. 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 26. **Pedometer app - step count**



The prediction accuracy of the Pedometer app in terms of mean absolute percentage error (MAPE)=97.5% with sigma=5 thanks to the high precision of the LIS2DU12 3-axis MEMS accelerometer

These parameters are appropriate for capturing human movement, filtering unwanted noise, and saving 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**: LIS2DU12 (low power accelerometer)
- **SENSOR SETTINGS**:
  - Power mode: Normal
  - Output data rate: 50 Hz
  - Full scale: 2 g
- **APP DATA OUTPUT**:
  - Number of steps
  - Cadence (number of steps per minute)

2.1.9 Sensor Fusion app

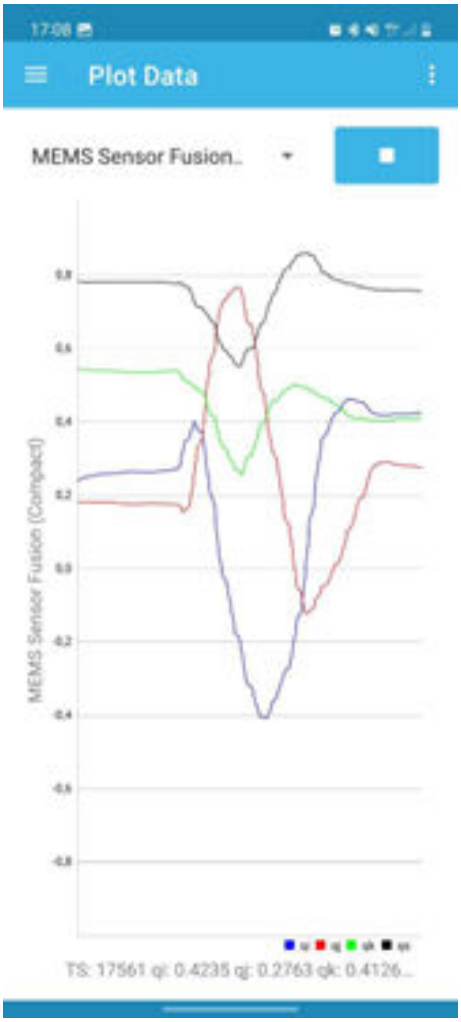
The Sensor Fusion app is based on the sensor fusion firmware algorithm (MotionFX library) embedded in the MCU.

The app shows the orientation estimation of SensorTile.boxPRO in the 3D space.

The algorithm uses the LSM6DSV16X 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 correct orientation of the SensorTile.boxPRO represented by a cube on the smartphone app.

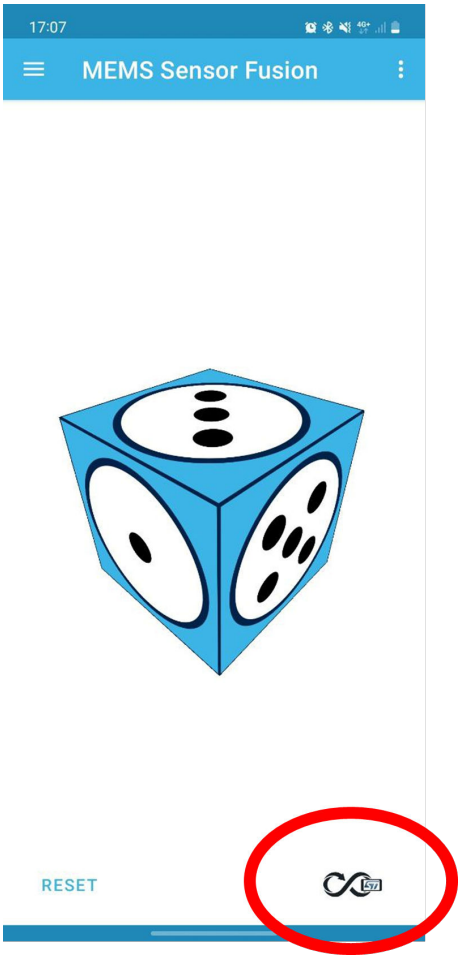
Figure 27. Sensor fusion app - Sensortile.box orientation (cube representation)

Figure 28. **Sensor fusion app - Sensortile.box orientation (plot)**



To enable magnetometer calibration, you need to touch the symbol in the bottom right-hand corner, shown in the figure below.

Figure 29. **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 30. **Sensor fusion app - enabling magnetometer calibration**

The Sensor Fusion Motion FX library provides orientation estimation, which is visible to the user via quaternions data, and magnetometer hard-iron offset compensation, accelerometer vibration rejection, and gyroscope bias compensation as metadata available only in Mode 3.

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**:
  - LSM6DSV16X (high bandwidth acceleration sensor and gyroscope)
  - LIS2MDL (compensated magnetometer)
- **SENSOR SETTINGS**:
  LSM6DSV16X
  - Low power mode
  - Output data rate: 60 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
    - Plot collected data

2.1.10 Free-Fall Detection app

The Free-Fall Detection app is an example of app created in mode 2, that combines the acceleration data of the LSM6DSV16X and the threshold comparison function of the firmware in the STM32U585microcontroller to detect when the board is in a free-fall condition.

The Free-fall condition is triggered when the accelerometer module is under 200 mg, the detected condition is shown to the user both with physical board outputs consisting of a buzzer sound and the red LED that stays on while the condition is true, and with the application that shows the event counter screen (along with the plot visualization).

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

- **APP DATA INPUT**: accelerometer values
- **SENSORS USED**: LSM6DSV16X high bandwidth acceleration sensor and gyroscope
- **SENSOR SETTINGS**:
  - Low power mode
  - Output data rate: 60 Hz
  - Filter: none
  - Full scale: 2 g for accelerometer
- **APP DATA OUTPUT**:
  - event counter
  - Buzzer
  - Red LED

2.1.11 Q-Touch app

The Q-Touch app is an example that uses the electrodes on the side of the board to give the possibility of interacting with the board using a new User Interface that is possible with the embedded on-board Qvar sensor.

This sensor is able to detect the change of static charge in the environment, and it is used in this application in one of the user interface configurations, where the electrodes are designed to be touched by the user to be used as switches.

Thanks to the design of the electrodes, a finite state machine was designed for the LSM6DSV16X that processes Qvar data and is able to detect a touch or a long press of the upper arrow, and when the user slides their finger up and down on the whole electrode.

The two arrows become like switches that can be pressed with a slight touch, just as in a tactile sensor (with a fraction of the power used by a common capacitive or resistive sensor).

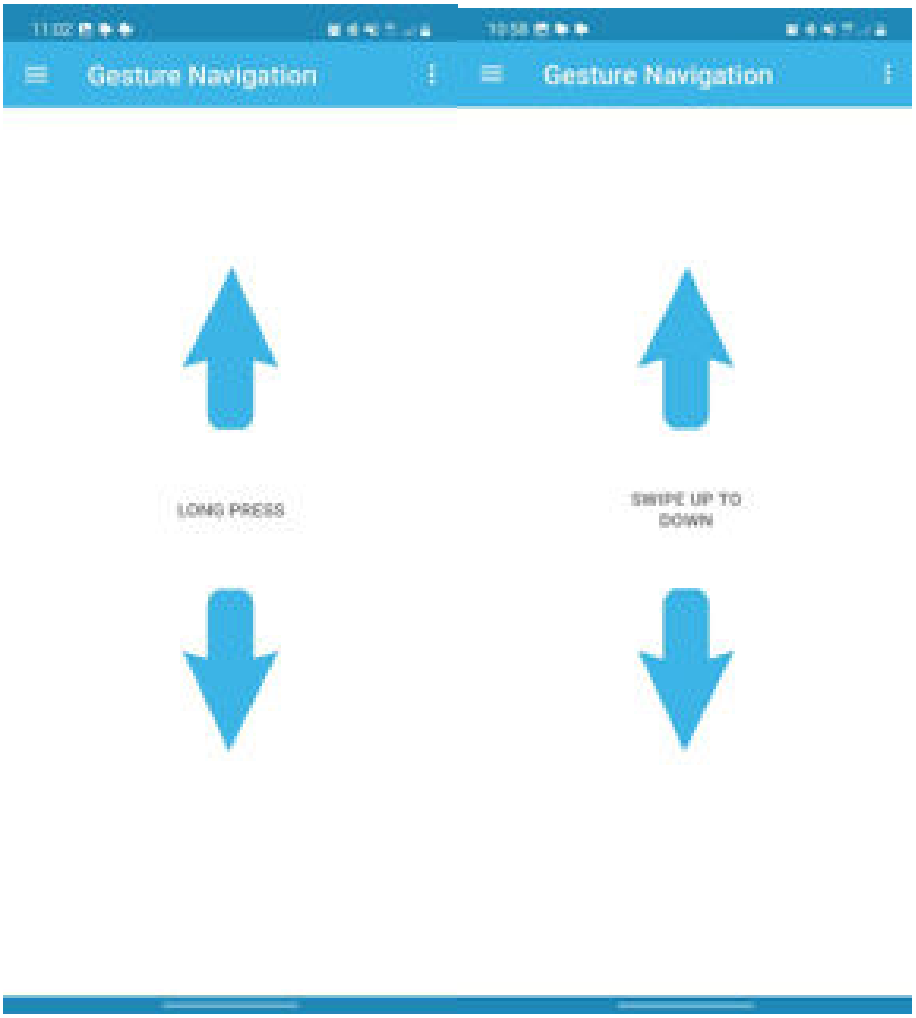
For further information on the Qvar sensor and the applications that use it, please, refer to the dedicated user manual.

Figure 31. Qvar electrodes on the board

It is possible to connect to ST BLE Sensor Classic to view the gesture navigation screen and check when the FSM detects the different outputs available. The buzzer will also beep with different tones depending on whether the slide was downward or upward.

In the following example screens a long press and a downward swipe are tested.

Figure 32. Gesture Navigation screens on ST BLE Sensor Classic App



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

- **APP DATA INPUT**: Qvar data
- **SENSORS USED**: LSM6DSV16X (embedded Qvar sensor)
- **SENSOR SETTINGS**:
  - Low power mode
  - Output data rate: 240 Hz
- **APP DATA OUTPUT**:
  - Gesture navigation screen
  - Buzzer

2.1.12 NFC Writer app

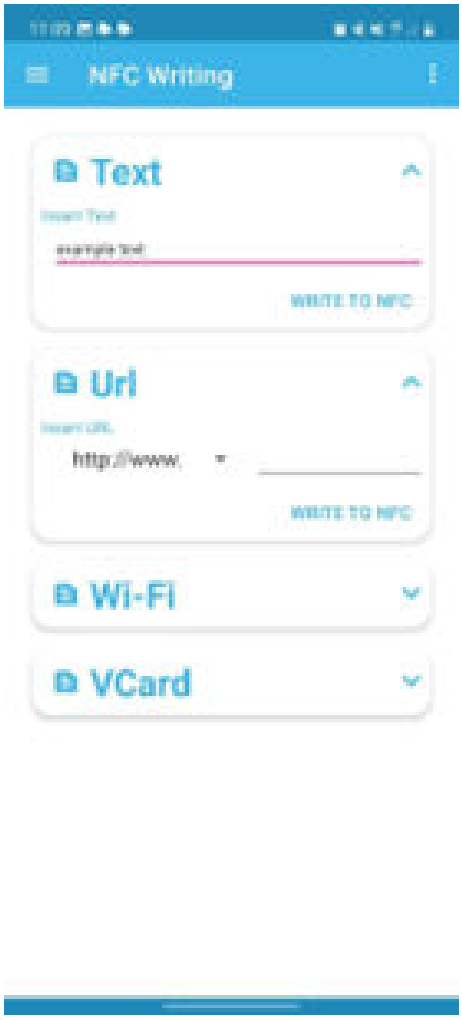
The NFC Writer app makes use of the ST25DV NFC sensor on the board: when uploading the app and connecting to the ST BLE Sensor Classic, the NFC Write screen is shown.

Here the user can write inside the NFC tag whatever he wants, filling the prompt and clicking on *** in the lower right corner of the screen. The data is sent to the NFC tag via BLE, and is written by the internal firmware of the SensorTile.boxPRO.

The user can then check whether the data was written correctly with the same phone (typically smartphones have embedded applications able to read NFC tags, if it is not available on your phone you can download one from the store).

The type of data that can be written on the NFC tag are shown in the figure below.

Figure 33. NFC Writing screen on ST BLE Sensor Classic App



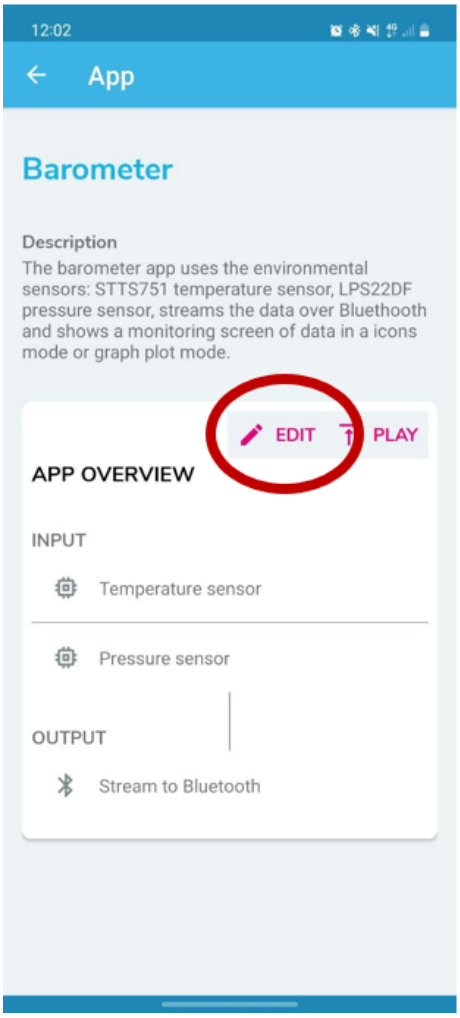
It is possible to connect to ST BLE Sensor Classic to view the gesture navigation screen and check when the FSM detects the different outputs available. The buzzer will also beep with different tones depending on whether the slide was downward or upward.

- **SENSORS USED**: ST25DV (embedded Qvar sensor)
- **APP DATA OUTPUT**:
  - NFC Writer screen

2.1.13 Editing apps

Many example apps are created using the expert mode provided by the ST BLE Sensor App, so they can be modified by changing inputs, outputs and functions involved. These parameters can be changed by going on the app page in the example apps category, and selecting [**edit**] (shown in a red circle in the example figure below).

Figure 34. Example for editing the existing apps



After selecting edit, the expert mode will open. Refer to step 3B and onwards in the next section

3 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.

3.1 How to use Expert Mode functionality

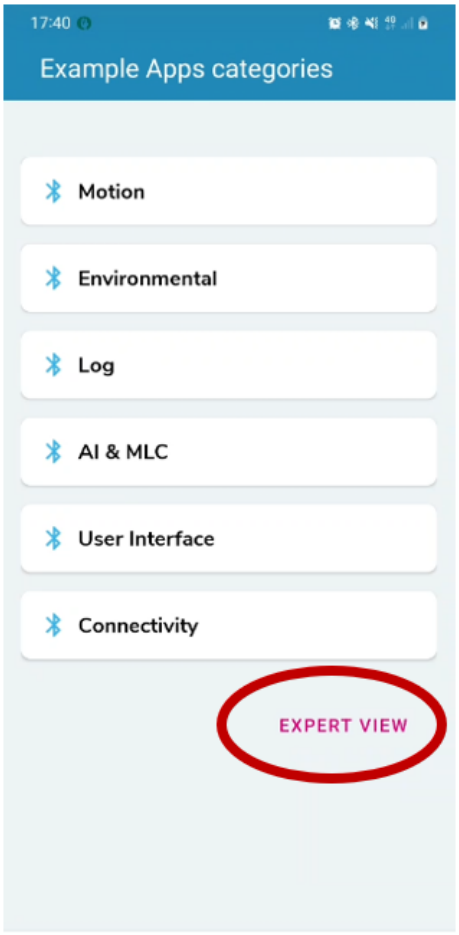
The application (or flow) can be created following the instructions below..

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

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

Step 3. Select [**EXPERT VIEW**] in the bottom right-hand corner.

Figure 35. Expert view selection



A new screen appears with saved apps.

Step 4. Select [**+ NEW APP**].

Step 5. Select one of the example application you want to start from.
For example, the Level app, as shown below.

Step 6. Click on [**EDIT**].

Figure 36. Level app edit

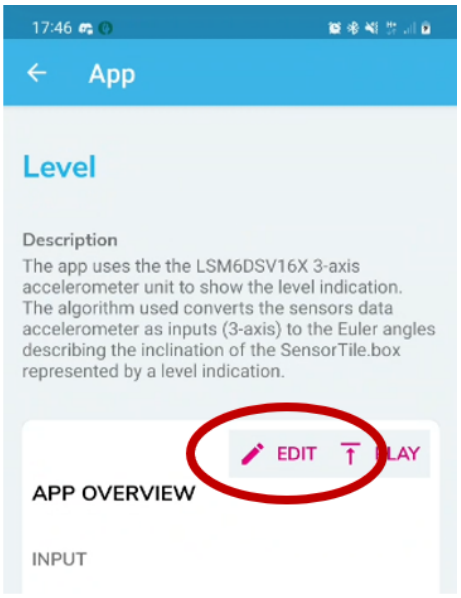
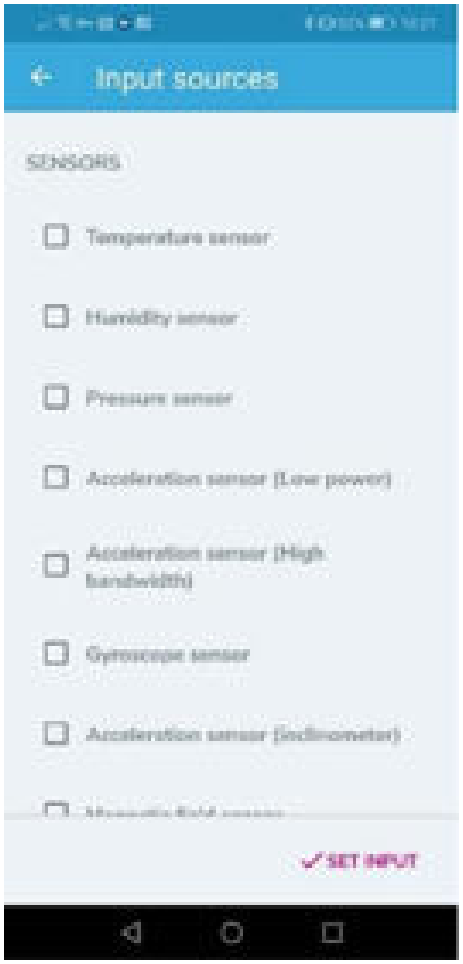


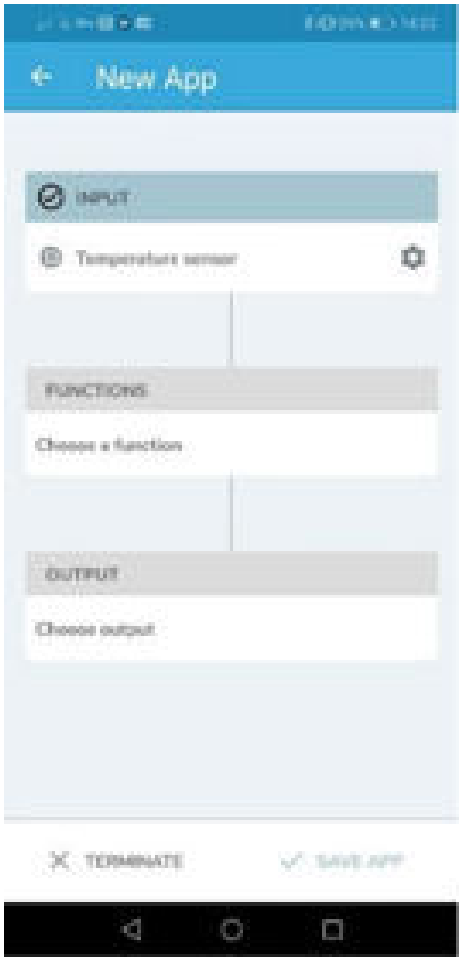
Figure 37. Input sources screen



Step 7. Now you are in the expert mode screen. Select one or more of the desired sensor data inputs and select [**SET INPUT**] to confirm

Unselected sensors are put in sleep mode.

Figure 38. Sensor data configuration screen



Step 8. 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.

Step 9. Under the sensor selection window, the function screen lists the available functions for the enabled sensors.

For the temperature sensor, for example, the available functions are shown below. If no function is selected, the sensor data are transferred directly to the output section.

Figure 39. Custom app function screen

Step 10. Then Choose between one of the following output types:

- File to SD card: To write data into the memory card (micro SD).
- Stream to USB: Via USB cable to be read through PC (VCOM port).
- Stream to Bluetooth: Via Bluetooth to the ST BLE Sensor Classic app on your smartphone (to view certain data).
- Save as Input: (available only when an app has at least one function in the flow). This allows creating a flow that can be used as input to another flow, or in a IF Statement (more info on this function in Section 3.2). it is a way to concatenate different functions and generate different branches that are processed one after the other.
- Save as EXP: (available only when the output is a boolean value, for example, the output of the Threshold Comparison function). This allows to create an Expression, that can be used in an IF Statement, described in detail in the next section. Produces an app branch whose output is a digital “true” or “false”. The options for this output lets the user choose if the red LED and the buzzer should turn on when the condition (the Boolean value in output) is True.

An app saved as EXP or as INPUT appears in the input selection screen so it can be used for more complex app generation.

3.2 Expert IF Statement creation

There is the possibility of using previously created apps to define an IF Statement.

In coding, the if syntax is simply written with open and close braces, followed by the condition the programmer intends to compare or check. The if expression compares whether the condition (or conditions) enclosed in the braces are true or false . If true , the if code block executes.

The general syntax is:

- If (expression == TRUE)
- {
- (statement)
- }

With SensorTile.boxPRO is it possible to choose an expression and a statement previously created to apply this idea.

Select IF from the expert view section to use an if statement. Then a screen opens, where you must select an expression (with [**choose an expression**]) and a statement (with [**choose app to upload**]) and click on play in the lower right corner of the screen.

The app will be uploaded and will be ready to use.

Revision history

Table 1. **Document revision history**

Date	Version	Changes
13-Nov-2023	1	Initial release.

Contents

List of figures