Getting started with the STM32Cube function pack for ultra-low power context awareness with distributed artificial intelligence (AI)

Introduction

FP-AI-CTXAWARE1 is an STM32Cube function pack featuring examples that let you connect your context awareness node to a smartphone via BLE and use a suitable Android™ or iOS™ application, like the STBLESensor app, to configure the device.

The package enables advanced applications such as human activity recognition (HAR) or acoustic scene classification (ASC), on the basis of outputs generated by the LSM6DSOX machine learning core (MLC) for HAR and the neural networks (NN) for ASC running on the STM32L4R9ZIJ6 MCU. The machine learning for HAR is a decision tree logic algorithm generated by Unico-GUI. The NN are implemented by a multi-network library supporting both floating and fixed point arithmetic, generated by the X-CUBE-AI extension for STM32CubeMX tool. The NN provided in this package are just examples of what can be achieved by combining the output of X-CUBE-AI with connectivity and sensing components from ST.

This package, together with the suggested combination of STM32 and ST sensors, can be used to develop specific wearable AI applications, where ultra-low power consumption is a key requirement, thanks to distributed deep edge AI approach.

The software runs on the STM32 microcontroller and includes all the necessary drivers for the STEVAL-MKSBOX1V1 evaluation board.

1 FP-AI-CTXAWARE1 software description

1.1 Overview

The FP-AI-CTXAWARE1 function pack features:

- Complete firmware to develop a context awareness node with BLE connectivity, digital microphone, environmental and motion sensors, performing real-time monitoring of sensors and audio data
- Machine Learning Core (MLC) featuring real-time human activity recognition (HAR) generated thanks to Unico-GUI and running on LSM6DSOX
- Middleware library generated thanks to STM32CubeMX extension called X-CUBE-AI, featuring example implementation of neural networks for acoustic scene classification (ASC) application
- Multi-network support: concurrent execution of the MLC for HAR and the neural network for ASC
- Ultra-low power implementation based on the use of an RTOS
- Compatible with STBLESensor application for Android/iOS, to perform sensor data reading, audio and motion algorithm feature demo in standalone or combined views, and firmware update over the air (full FOTA)
- Sample implementation available for STEVAL-MKSBOX1V1 evaluation board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software creates the following Bluetooth services:

- hardware characteristics related to MEMS sensor devices:
  - temperature
  - pressure
  - humidity
  - 3D gyroscope, 3D magnetometer, 3D accelerometer
  - microphone dB noise level
  - battery level, voltage and status (charging/discharging/low battery)
- Console service:
  - stdin/stdout for bi-directional communication between client and server
  - stderr for a mono-directional channel
- a service to enable the following expansion hardware features for the LSM6DSOX motion sensor for STEVAL-MKSBOX1V1:
  - free fall detection
  - single tap detection
  - double tap detection
  - wake-up detection
  - tilt detection
  - 3D orientation

The software gathers the temperature, humidity, pressure, audio, magnetometer sensor and motion sensor drivers for the HTS221, LPS22HH, LIS2MDL and LSM6DSOX devices when you use a STEVAL-MKSBOX1V1 evaluation board.

The package is compatible with the STBLESensor Android and iOS (Ver. 4.10.0 or higher) apps, which you can download from the respective stores and use to display information sent via BLE.

The STBLESensor application allows full Over-The-Air firmware update.

1.2 Architecture

The STM32Cube function packs leverage the modularity and interoperability of STM32 Nucleo and X-NUCLEO boards running STM32Cube and X-CUBE software to create functional examples representing some of the most common use cases in certain applications.

The software function packs are designed to fully exploit the underlying STM32 ODE hardware and software components to best satisfy the final user application requirements.

Function packs may include additional libraries and frameworks, not present in the original X-CUBE packages, which enable new functions and create more targeted and usable systems for developers.

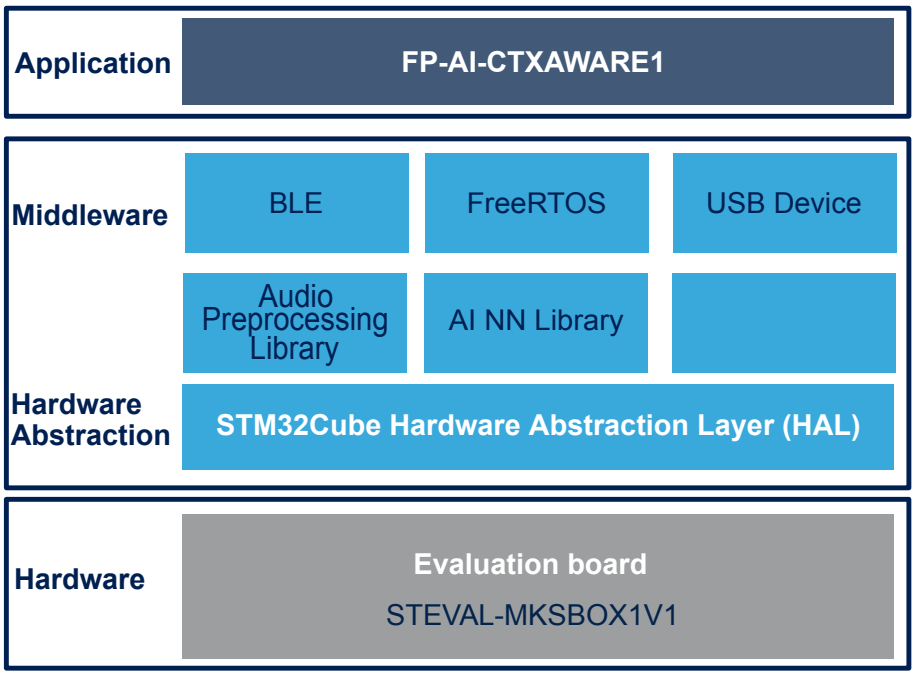
STM32Cube version 1.x includes:

- STM32CubeMX, a graphical software configuration tool that allows the generation of C initialization code using graphical wizards.
- A comprehensive embedded software platform specific to each series (such as the STM32Cube for the STM32 series), which includes:
  - the STM32Cube HAL embedded abstraction-layer software, ensuring maximized portability across the STM32 portfolio
  - a consistent set of middleware components such as RTOS, USB, TCP/IP and graphics
  - all embedded software utilities with a full set of examples

To access and use the sensor expansion board, the application software uses:

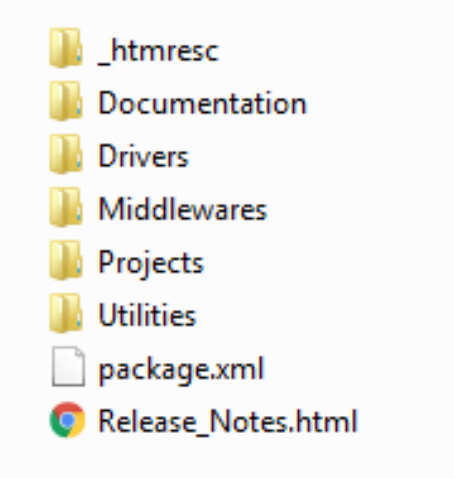
- **STM32Cube HAL layer**: provides a simple, generic and multi-instance set of generic and extension APIs (application programming interfaces) to interact with the upper layer application, libraries and stacks. It is directly based on a generic architecture and allows the layers that are built on it, such as the middleware layer, to implement their functions without requiring the specific hardware configuration for a given microcontroller unit (MCU). This structure improves library code reusability and guarantees easy portability across other devices.
- **Board support package (BSP) layer**: supports the peripherals on the STM32 Nucleo board (except the MCU) with a limited set of APIs providing a programming interface for certain board-specific peripherals like the LED, the user button, etc., and helps determine the specific board version. For the sensor expansion board, it provides the programming interface for various inertial and environmental sensors and support for initializing and reading sensor data.

Figure 1. FP-AI-CTXAWARE1 software architecture



1.3 Folder structure

Figure 2. **FP-AI-CTXAWARE1 package folder structure**



The following folders are included in the software package:

- **Documentation**: contains a compiled HTML file generated from the source code, which details the software components and APIs.
- **Drivers**: contains the HAL drivers, the board-specific drivers for each supported board or hardware platform (including the on-board components), and the CMSIS vendor-independent hardware abstraction layer for the Cortex-M processor series.
- **Middlewares**: contains libraries and protocols for BlueNRG-M2 Bluetooth low energy module (SPBTLE-1S for old batches of STEVAL-MKSBOX1V1), USB Device Library, FreeRTOS real-time operating system, Parson json parser and AI Middleware libraries.
- **Projects**: contains two sample applications used to show the output of the sensors data and the results of the Acoustic Scene Classification or Human Activity Recognition algorithm. One application with Bluetooth low energy protocol provided by STEVAL-MKSBOX1V1 platform and one application to control and visualize the output using a simple Command Line Interface. Both the projects can be compiled using IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM-STR) and STM32CubeIDE.
- **Utilities**: contains the ucf files used to run the human activity recognition on the LSM6DSOX Machine Learning Core.

1.4 Firmware-Over-The-Air (FOTA) update

The FP-AI-CTXAWARE1 firmware may be updated Over-The-Air (FOTA) through the connected Android/iOS device via Bluetooth using the STBLESensor application (ver. 4.1.0 and above) available on their respective application market stores.

The application sends the update and associated CRC (cyclic-redundancy-check) value, which the FP-AI-CTXAWARE1 checks against the hardware cyclic redundancy check calculation unit on the STM32L4R9ZI processor to ensure integrity. If the CRC calculation matches the STBLESensor CRC value, the new firmware is written to the other Flash bank and the FP-AI-CTXAWARE1 will reboot using the new saved firmware.

1.5 APIs

Detailed user-API technical information with full function and parameter descriptions is available in a compiled HTML file in the package “Documentation” folder.

1.6 Algorithms

1.6.1 Acoustic Scene Classification

This algorithm recognizes three different types of acoustic scenes:

- indoor:
  - silent home/office environments
  - home/office environments with people talking and/or TV/radio ON
  - coffee shops/restaurants
  - shopping malls
- outdoor:
  - low noise residential areas
  - city center with car/vehicle traffic
- in-vehicle:
  - in-car with engine ON, radio ON/OFF
  - bus
  - train

Note: Wrong classifications might occur in the following situations:

- *indoor:*
  - *silent environment with background noise (for example, air conditioning, fan electric motors)*
  - *human activities (if the device is worn) can generate air flow on the microphone leading to outdoor classification even indoor*
- *outdoor:*
  - *crowded environments with people talking near the microphone can mislead to indoor classification*

It is possible a run-time monitoring of the ASC model confidence level by enabling the serial in the STBLESensor: you just have to select on the top right option menu [**Show serial console**] to check run-time the model confidence level.

Important:
The training recordings for the Acoustic Scene Classification algorithm have been performed by using the STEVAL-MKSBOX1V1 case and wearing the device on the wrist. Thus, to get accurate acoustic scene predictions, you have to implement the same setup.

1.6.2 Human Activity Recognition

Two different MLC byte code models have been implemented for the Human Activity Recognition (HAR) algorithm using the LSM6DSOX:

1. indoor/outdoor use case, including the following output states:
   - stationary
   - walking
   - jogging
   - biking
2. in-car use case, including the following output states:
   - stationary
   - walking
   - driving

Important:
The training recordings for the Human Activity Recognition algorithm have been performed by using the STEVAL-MKSBOX1V1 case and wearing the device on the wrist. Thus, to get accurate activity detections, you have to implement the same setup.

1.7 Sample applications

The FP-AI-CTXAWARE1 function pack contains two applications that allow control of the running algorithms and visualization of the results: one application uses STBLESensor via Bluetooth and the other application uses a simple command line interface.

1.7.1 BLELowPower Bluetooth application

The BLELowPower application is a low power application that can send the values listed hereafter to the STBLESensor (Android/iOS application):

- temperature/pressure/humidity
- the accelerometer, gyroscope and magnetometer values
- the dB noise measured by the microphone
- the output of the feature recognized by the LSM6DSOX (Free Fall/Tilt/6D Orientation/Wake Up/ Single&Double Tap)
- the output of the Acoustic Scene Classification running on the STM32L4R9ZI processor
- the output of the Activity Recognition running on the LSM6DSOX Machine Learning Core
- a combined view for the output of Audio Scene Classification and the Activity Recognition algorithms

When the Human Activity Recognition (HAR) algorithm is running on the LSM6DSOX, the STM32L4R9ZI is in stop mode reducing the power consumption. The STM32 processor is awakened by the interrupts generated by the LSM6DSOX when it recognizes a new activity.

The output of the Acoustic Scene Classification (ASC) algorithm is used to load the right program on the Machine Learning Core on the basis of the context detected by the algorithm itself.

At the beginning both algorithms are working and the MLC uses the program for indoor/outdoor activity recognition by default.

When the ASC finds a valid output with a good level of accuracy, if it detects that you are in a car, the program for in-car activity recognition is loaded on the MLC, otherwise the program for indoor/outdoor activity recognition continues running.

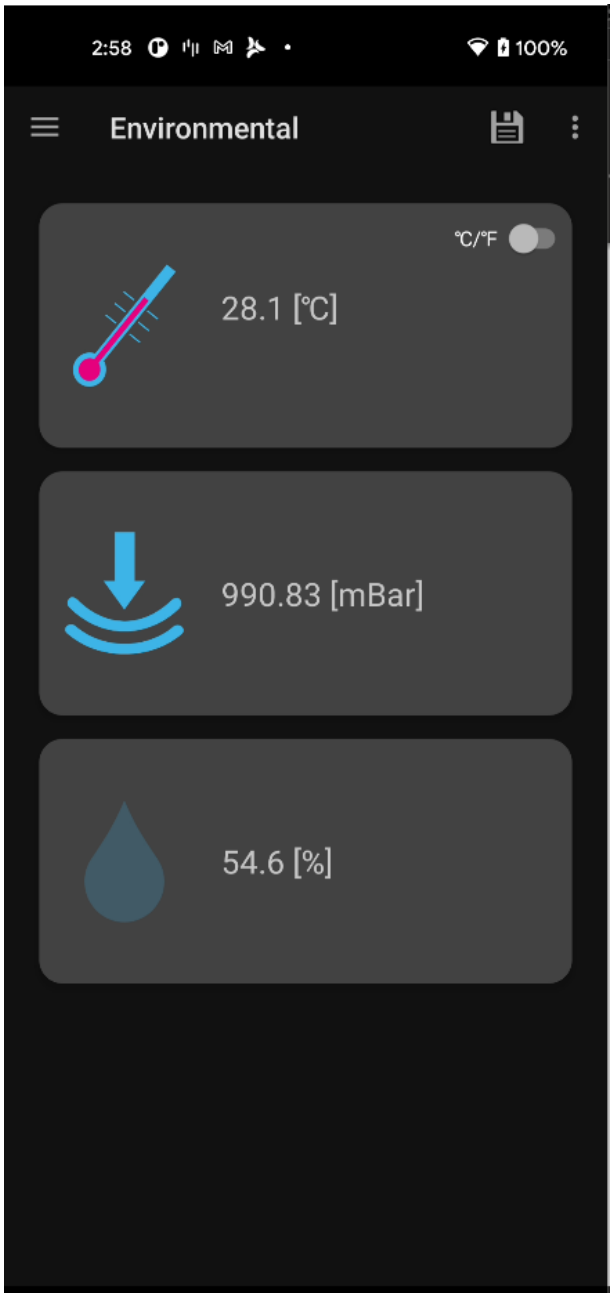
Afterward, the ASC enters sleep mode and puts the STM32L4R9ZI in stop mode.

The ASC is periodically waked up to control whether the board has changed the context from car to indoor/ outdoor to change the MLC running program as well.

The ASC is also waked up by the MLC when it detects a change on the activity recognized that could trigger a new audio scene detection.

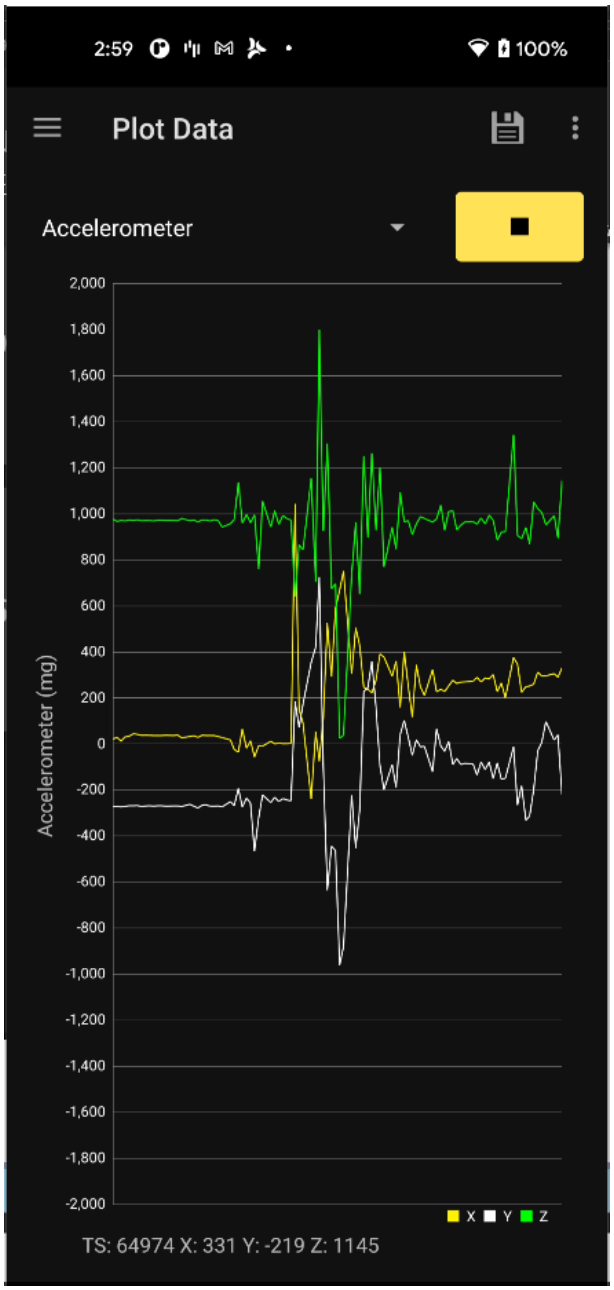
Through the STBLESensor app, the environmental page shows the temperature, pressure and humidity values.

Figure 3. **Environmental demo page**



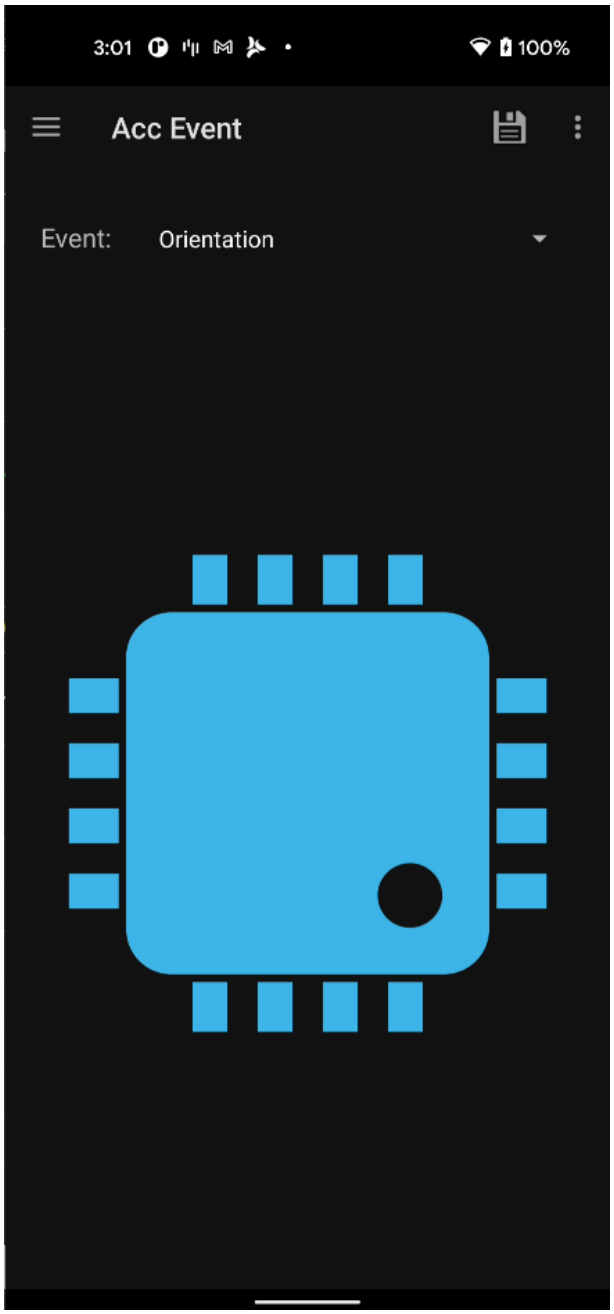
The Plot demo shows plots of all the environmental/inertial sensor values.

Figure 4. **Plot demo page**



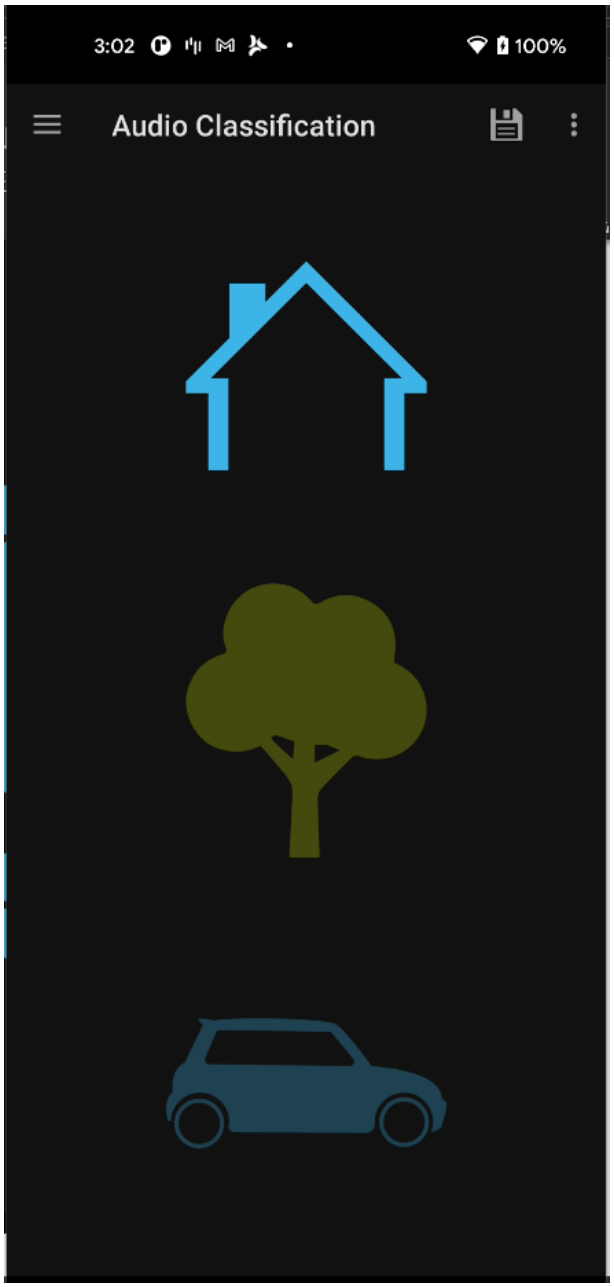
The Acc Event demo shows all the events computed by the LSM6DSOX.

Figure 5. **Acc Event demo page**



The Acoustic Scene Classification demo shows the output of the ASC algorithm running on the STM32L4R9ZI.

Figure 6. **ASC demo page**

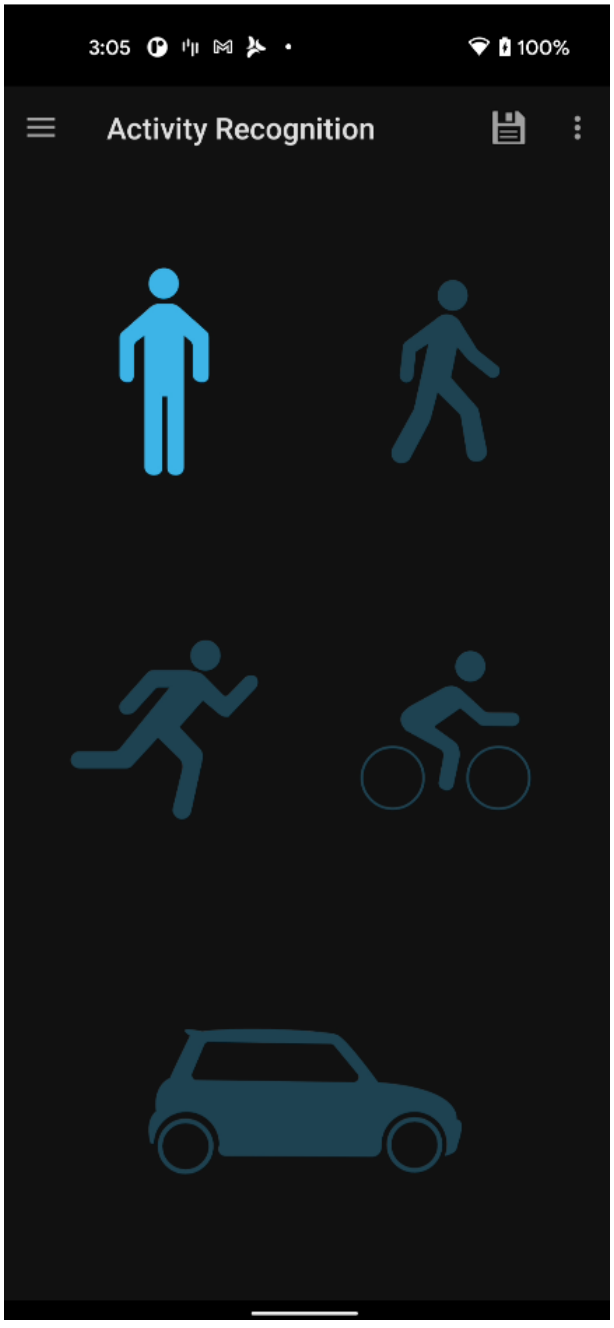


The Activity Recognition demo shows the result of the Human Activity Recognition algorithm running on the LSM6DSOX MLC.

For this page, the indoor/outdoor program has to be loaded on the MLC and, when this demo is running, the STM32L4R9ZI is in stop mode.

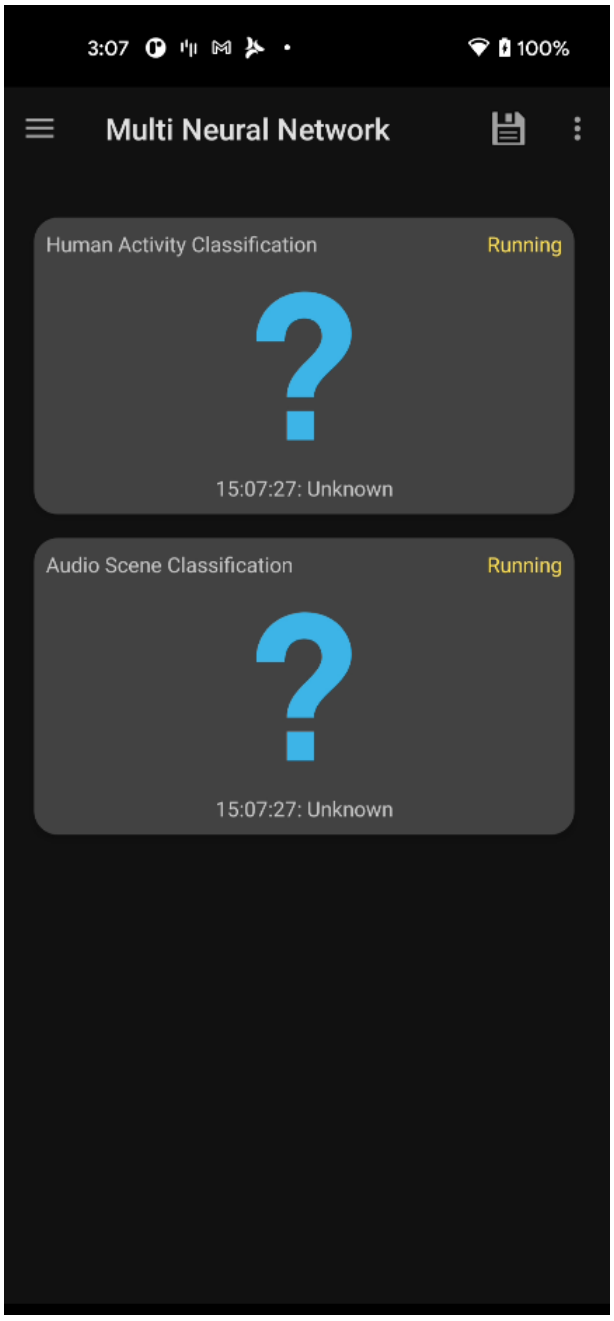
When the MLC recognizes a new activity, it wakes up the STM32L4R9ZI with an interrupt, the STM32 sends the new values to the STBLESensor and then returns in stop mode

Figure 7. **Activity Recognition demo**



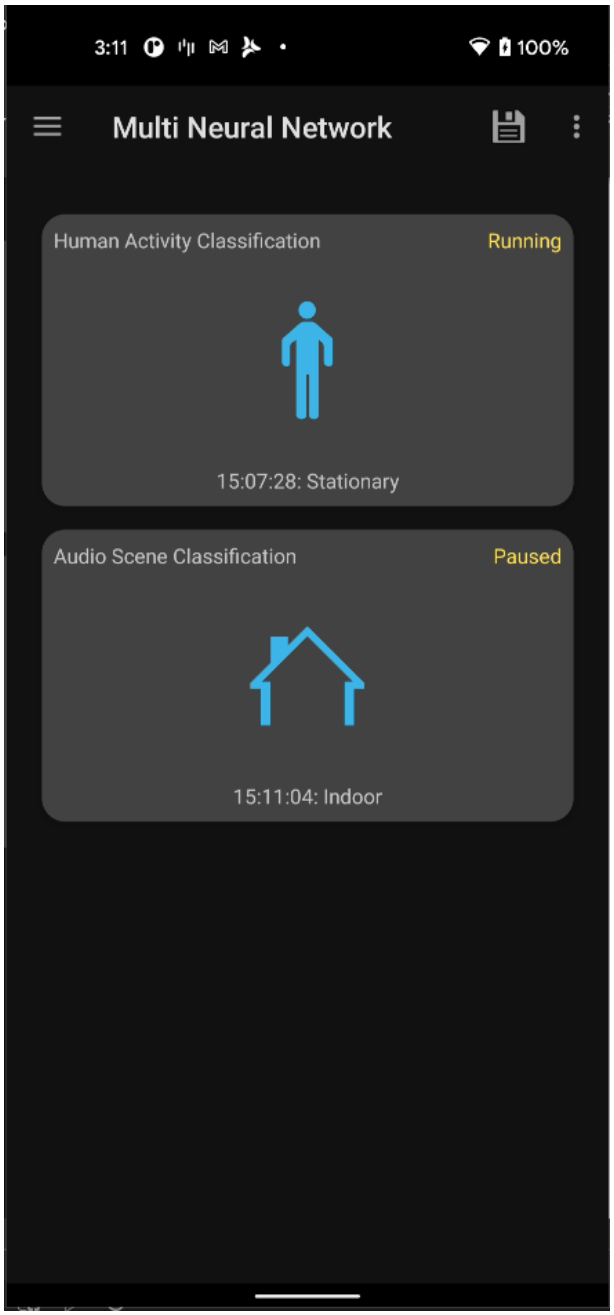
In the Multi Neural Network demo, both algorithms are working: the ASC on the STM32L4R9ZI and the Activity Recognition on the MLC core.

Figure 8. **Multi Neural Network demo when both algorithms are running**



When the ASC finds a good results, the best program for the found scene is loaded on the MLC . The MLC continues running and the ASC goes to sleep mode, putting the STM32L4R9ZI in stop mode to reduce power consumption.

Figure 9. **Multi Neural Network demo when ASC is in sleep mode and the MLC is running**

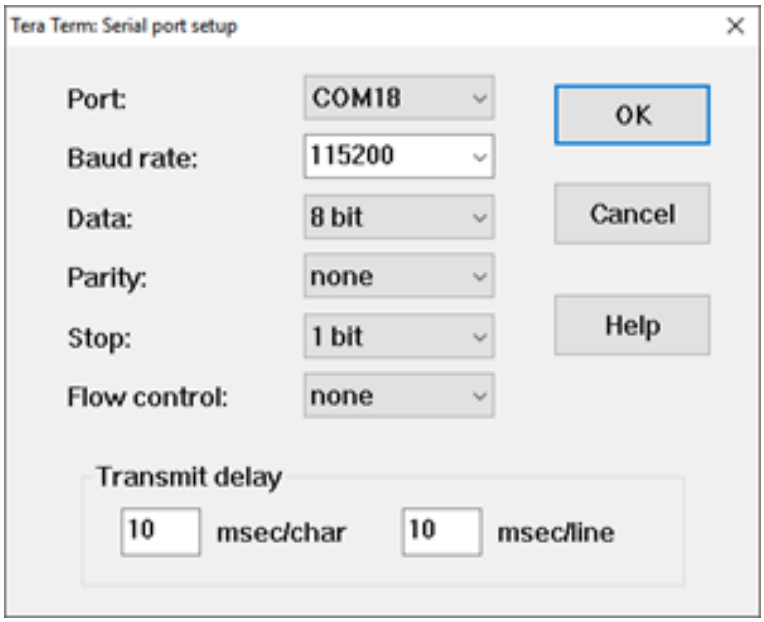


In each Card View, there is a label to show when each algorithm is running or when it is in sleep mode.

1.7.2 CLAi - Command Line Interface application

The Command Line Interface application allows interaction with the STEVAL-MKSBOX1V1 board by just connecting the board to the PC through a USB cable and using a HyperTerminal application like TeraTerm.

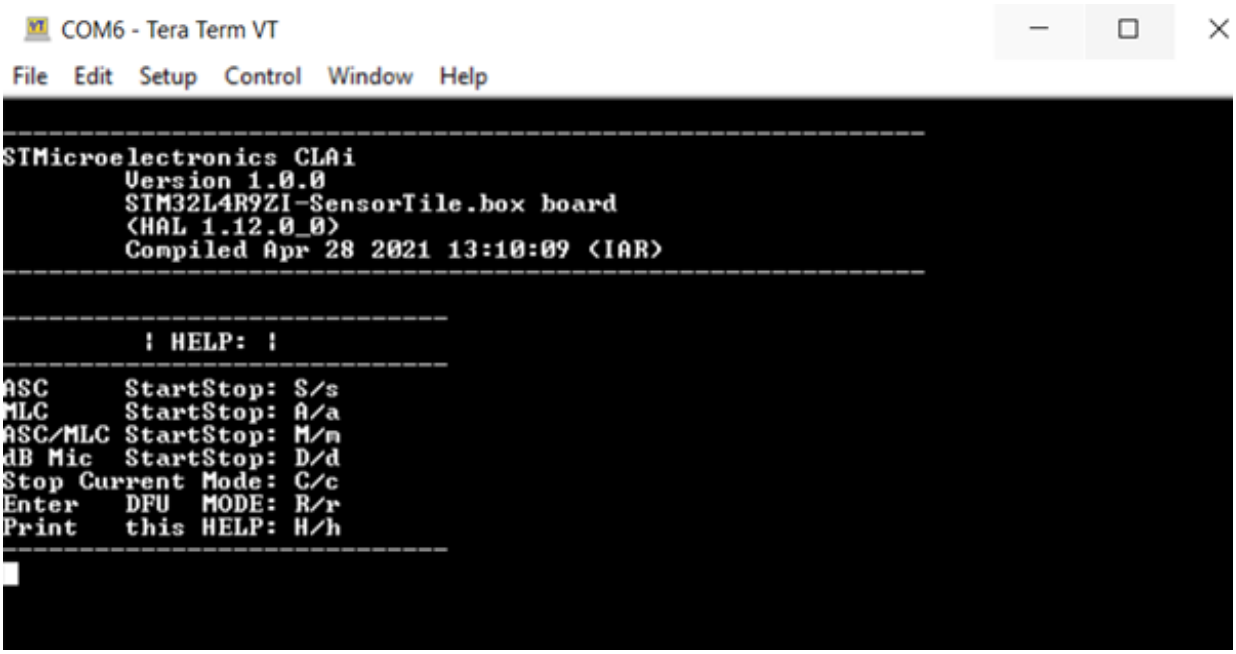
Figure 10. **Serial console settings**



With the CLAi application, you can:

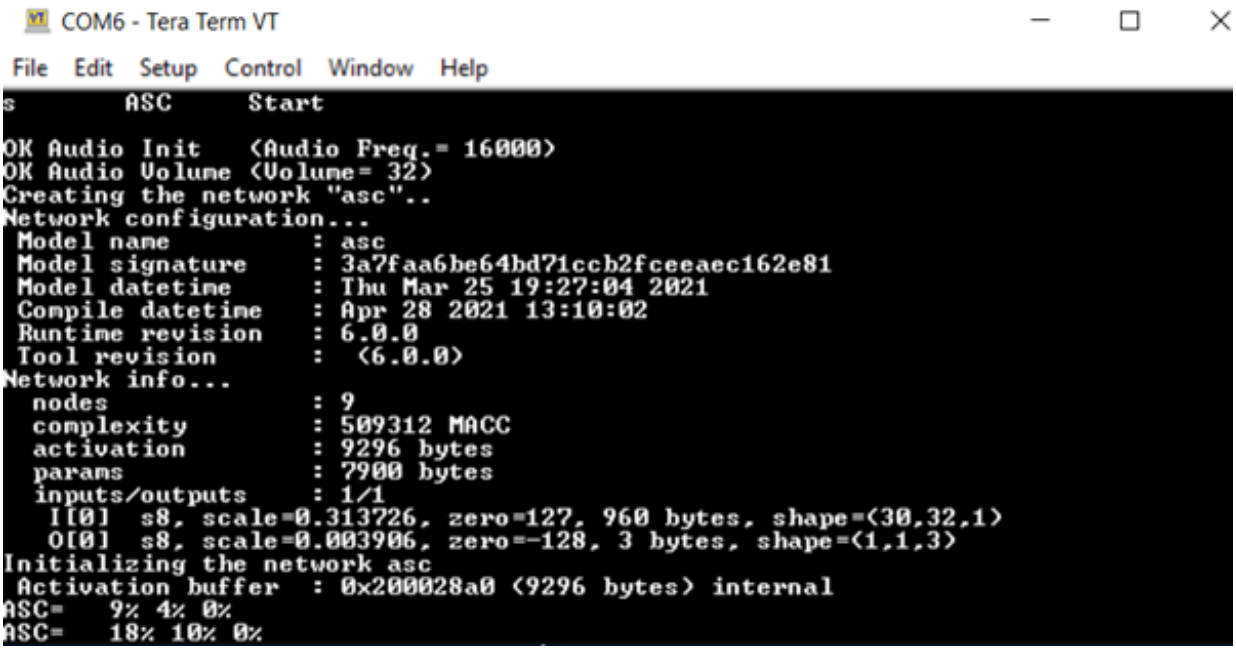
- start/stop the HAR algorithm using the LSM6DSOX MLC in standalone mode
- start/stop the ASC algorithm using the Neural Networks in standalone mode
- start/stop HAR and ASC algorithms in combined mode
- get the dB noise found by the microphone
- enter DFU mode

Figure 11. **CLAi application - list of commands**



To run the ASC algorithm in standalone mode, simply type “S” or “s”: the output on the serial console continuously shows the confidence percentage for every possible ASC classification (indoor, outdoor and in-vehicle) as shown below.

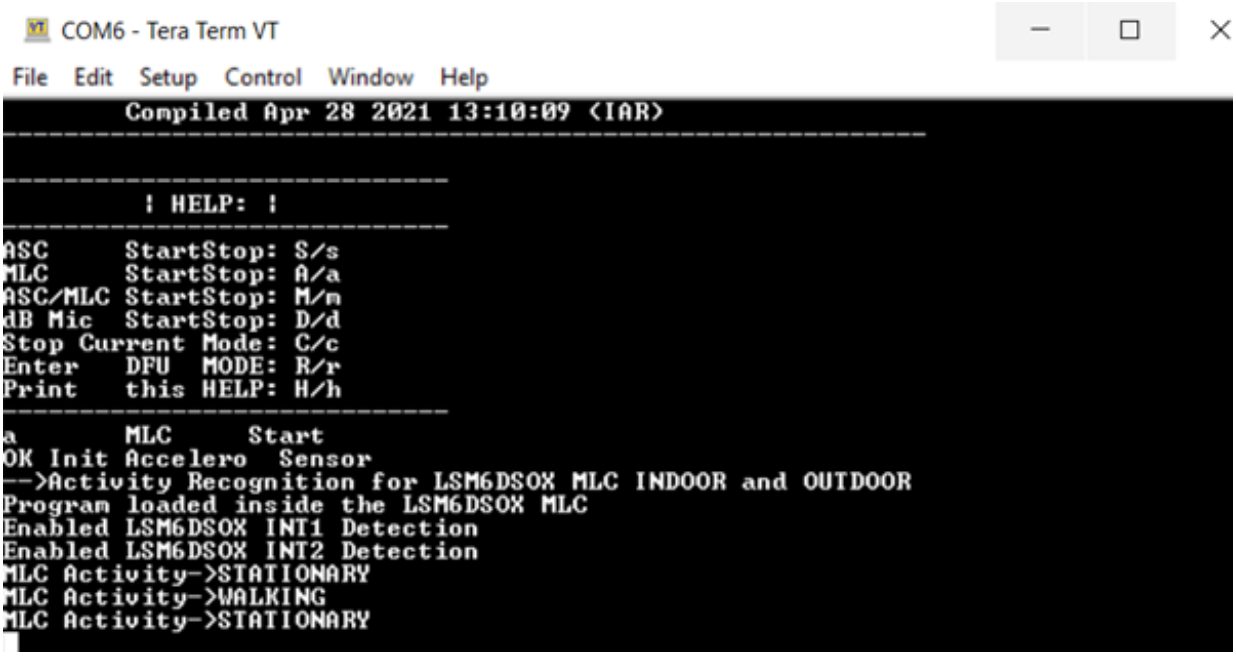
Figure 12. **Output of ASC algorithm in standalone mode**



To stop the ASC algorithm in standalone mode, type “S”, “s”, “C”, or “c” or jump to another mode.

To run the HAR algorithm in standalone mode using the LSM6DSOX MLC, type “A” or “a”. For this use case, the MLC program for Indoor/Outdoor scenario is loaded in the LSM6DSOX. The output on the serial console will show the activity status any time it changes as shown below.

Figure 13. **Output of HAR algorithm in standalone mode**



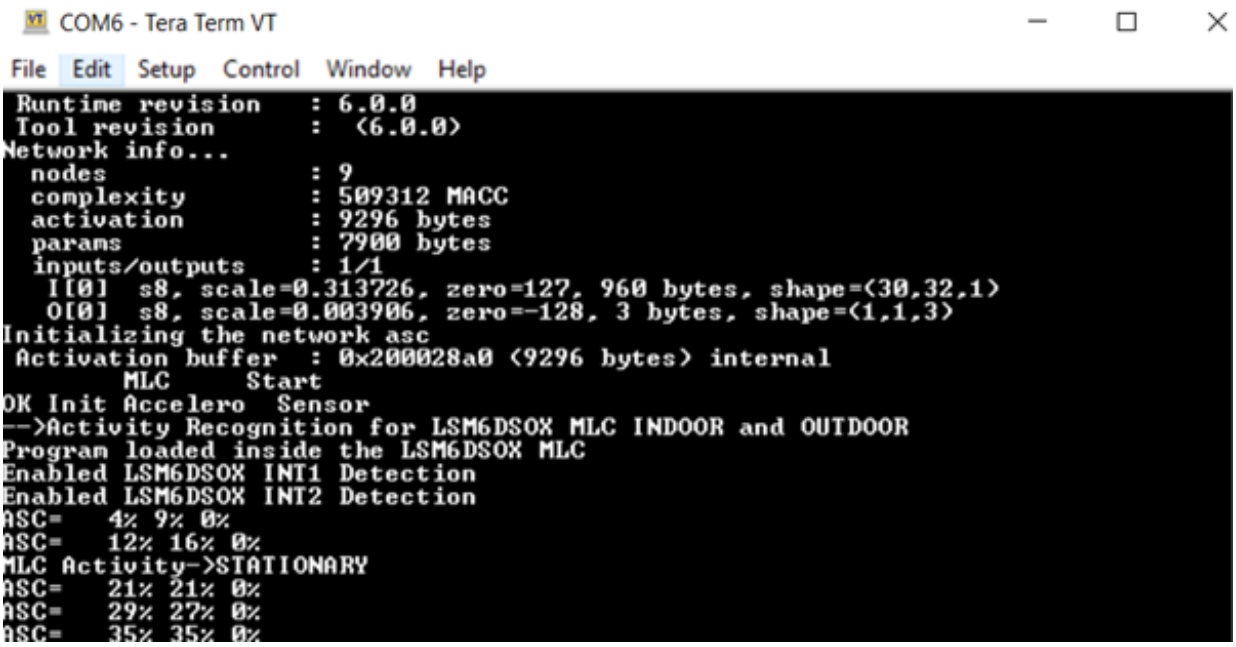
To stop the HAR algorithm in standalone mode, type “A”, “a”, “C” or “c” or jump to another mode.

To run ASC and HAR algorithms in combined mode, type “M” or “m”. At the beginning, the MLC byte code for indoor/outdoor scenario is loaded in the LSM6DSOX. The output on the serial console will show the activity status any time it changes and the confidence percentage for every possible ASC classification until a confidence percentage overcomes the value of 50%.

After this event, according to the output of ASC algorithm, the firmware tries to load the MLC byte code for the most suitable scenario (indoor/outdoor scenario if the output of ASC is indoor or outdoor, car scenario if the output of ASC is in-vehicle) in the LSM6DSOX.

The ASC algorithm is stopped and will be restarted periodically every 30 seconds or any time the HAR output changes.

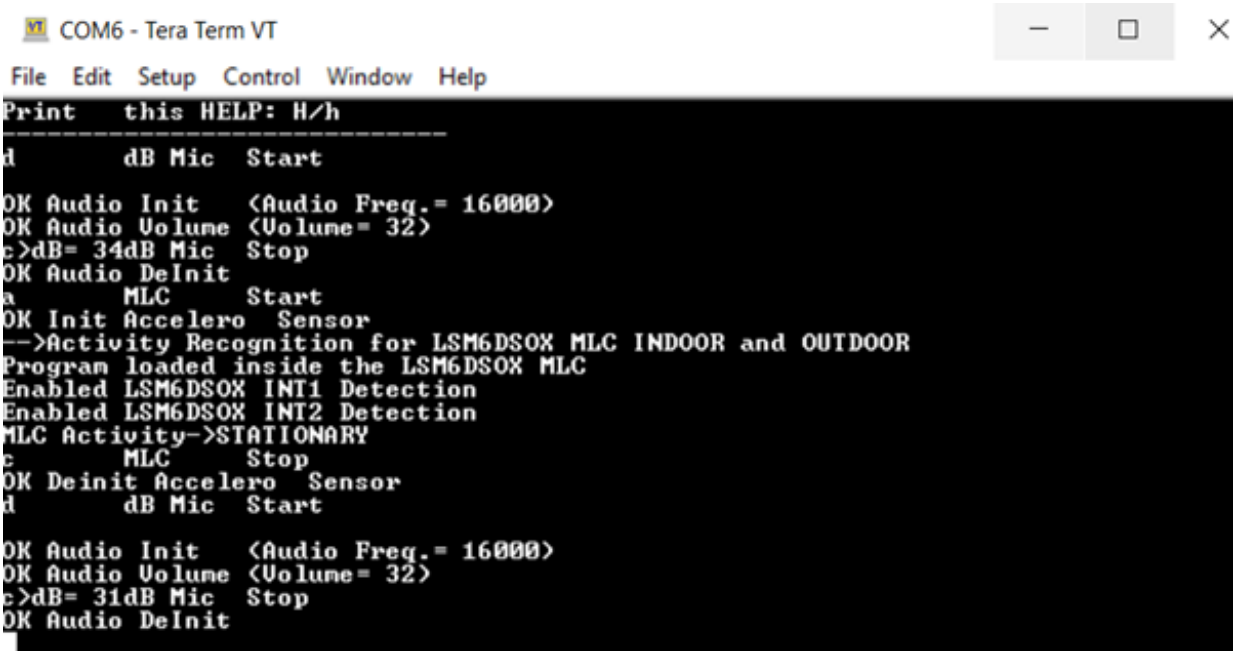
Figure 14. Output of HAR and ASC algorithm in combined mode



To stop the HAR and ASC algorithms in combined mode, type “M”, “m”, “C” or “c” or jump to another mode.

To get the dB noise found by the microphone, type “D” or “d. The output on the serial console will show the current dB noise found by the microphone as shown below.

Figure 15. Output of dB noise found by the microphone



To stop the dB noise display, type “D”, “d”, “C” or “c” or jump to another mode.

2 System setup guide

2.1 Hardware description

2.1.1 STEVAL-MKSBOX1V1 evaluation kit

The STEVAL-MKSBOX1V1 (SensorTile.box) is a ready-to-use box kit with wireless IoT and wearable sensor platform to help you use and develop apps based on remote motion and environmental sensor data, regardless of your level of expertise.

The SensorTile.box board fits into a small plastic box with a long-life rechargeable battery, and the ST BLE Sensor app on your smartphone connects via Bluetooth to the board and allows you to immediately begin using the wide range of default IoT and wearable sensor applications.

In Expert Mode, you can build customs apps from your selection of SensorTile.box sensors, operating parameters, data and output types, and special functions and algorithms available. 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 code development using the STM32 Open Development Environment (STM32 ODE), which includes a sensing AI function pack with neural network libraries.

Figure 16. STEVAL-MKSBOX1V1 evaluation kit

2.2 Software description

The following software components are needed in order to set up a suitable development environment for creating applications for the STEVAL-MKSBOX1V1:

- FP-AI-CTXAWARE1: Bluetooth low energy and sensors software for STM32Cube.
- Development tool-chain and Compiler. The STM32Cube expansion software supports the three following environments:
  - IAR Embedded Workbench for ARM® (EWARM) toolchain + ST-LINK
  - RealView Microcontroller Development Kit (MDK-ARM) toolchain + ST-LINK
  - STM32CubeIDE + ST-LINK

2.3 Hardware requirements

The following hardware components are required:

- the STEVAL-MKSBOX1V1 evaluation board
- the driver for the ST-LINK-V2 debugger/programmer (downloadable at STSW-LINK008 or STSW-LINK009)

- One USB type A to Micro-B USB cable to connect the STEVAL-MKSBOX1V1 to the PC (or plug the battery)
- One USB type A to Mini-B USB cable to connect the ST-LINK/V2 to the PC

2.4 Hardware setup

Step 1. Connect the STEVAL-MKSBOX1V1 to ST-LINK/V2 using the flat cable.

Step 2. Connect the ST-LINK/V2 to the PC using a USB type A to Mini-B USB cable.

Step 3. Power the STEVAL-MKSBOX1V1 by plugging the battery or using a USB type A Micro-B USB cable connected to a PC.

Revision history

Table 1. **Document revision history**

Date	Revision	Changes
10-May-2021	1	Initial release.

Contents

List of figures