UM2524

User manual

Getting started with the STM32Cube function pack for ultra-low power IoT nodes for AI applications based on audio and motion sensing

Introduction

FP-AI-SENSING1 is an STM32Cube function pack featuring examples that let you connect your IoT node to a smartphone via BLE and use an Android™ or iOS™ application such as the STBLESensor app to configure the device. A Command Line Interface through UART is provided for developer convenience.

The package enables advanced applications such as human activity recognition and audio scene classification based on outputs generated by neural networks (NN). The four NN are implemented in a library generated by the X-CUBE-AI extension for STM32Cube tool. As such, the implementation provided in this package is just an example of what can be achieved by combining the output of X-CUBE-AI with connectivity and sensing components from ST.

The package comes with an AI utility for data logging and annotation on SD cards. You can record data from sensors and define the classes or events to record. You can then use the annotated data to train your own Neural Network on your PC/GPU/Cloud, obtain the model, use the X-CUBE-AI extension for STM32CubeMX tool for conversion and then run it on the STM32 platform.

This package and the suggested combination of STM32 and ST devices can be used to develop specific wearable AI applications, industrial predictive maintenance applications, smart things and buildings applications in general, where ultra-low power consumption is a key requirement.

The software runs on the STM32 microcontroller and includes all the necessary drivers for the STM32 Nucleo development board and expansion boards, as well as for the STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1 evaluation boards and the B-L475E-IOT01A STM32L4 Discovery kit IoT node.

1 FP-AI-SENSING1 software description

1.1 Overview

The FP-AI-SENSING1 function pack has the following features:

- Complete firmware to develop an IoT node with BLE connectivity, digital microphone, environmental and motion sensors, and perform real-time monitoring of sensors and audio data
- Middleware library generated thanks to STM32CubeMX extension called X-CUBE-AI, featuring example implementation of neural networks for real-time human activity recognition (HAR) and acoustic scene classification (ASC) applications
- Multi-network support: concurrent execution of several neural networks
- AI utility for data logging and annotation on SD card or QSPI Flash memory
- 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, and firmware update over the air (full and partial FOTA)
- Sample implementation available for STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1 evaluation boards, for B-L475E-IOT01A and for X-NUCLEO-CCA02M1, X-NUCLEO-IKS01A2 and X-NUCLEO-IDB05A1 connected to a NUCLEO-L476RG board
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

This software creates the Bluetooth services listed below:

1. 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) for STEVAL-MKSBOX1V1 and STEVAL-STLKT01V1
2. Software characteristics:
   - Data logging (audio and MEMS data) using Generic FAT File System middleware (for STEVAL-STLKT01V1, STEVAL-MKSBOX1V1 and B-L475E-IOT01A)
3. Console service:
   - stdin/stdout for bi-directional communication between client and server
   - stderr for a mono-directional channel from the STM32 Nucleo board to an Android/iOS device
4. A service to enable the following expansion hardware features for LSM6DSL on X-NUCLEO-IKS01A2 expansion board for STM32 Nucleo L4, for LSM6DSM motion sensor STEVAL-STLKT01V1 and for LSM6DSO motion sensor for STEVAL-MKSBOX1V1:
   - pedometer
   - free fall detection
   - single tap detection
   - double tap detection
   - wake-up detection
   - tilt detection
   - 3D orientation
   - multi-events detection (3D orientation, pedometer, single tap, double tap, free fall and tilt detection)

This software gathers:

- the temperature, humidity, pressure, audio and motion sensor drivers for the HTS221, LPS22HB, MP34DT01-M, LSM6DSL and LSM303AGR devices when you use an X-NUCLEO-IKS01A2 expansion board mounted on an STM32 Nucleo platform

- the temperature, pressure, audio, motion sensor and gas gauge IC drivers for the LPS22HB, MP34DT04, LSM6DSM, LSM303AGR and STC3115 devices for the STEVAL-STLKT01V1 evaluation board
- the temperature, humidity, pressure, audio, magnetometer sensor and motion sensor drivers for the HTS221, LPS22HB, MP34DT01-M, LIS3MDL and LSM6DSL devices when you use a B-L475E-IOT01A STM32L4 Discovery kit IoT node
- the temperature, humidity, pressure, audio, magnetometer sensor and motion sensor drivers for the HTS221, LPS22HH, LIS2MDL and LSM6DSO devices when you use a STEVAL-MKSBOX1V1 evaluation board

The package is compatible with the STBLESensor Android and iOS (Ver. 4.1.0 or higher) apps, which you can download from the respective stores and use to display information sent via BLE.

The STBLESensor application allows full and partial 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.

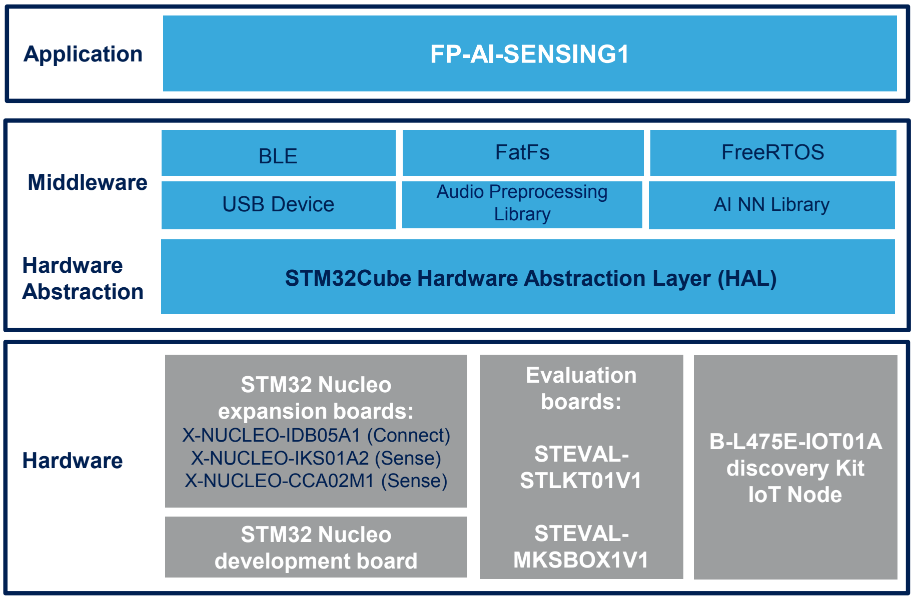
STM32Cube version 1.x includes:

- 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:

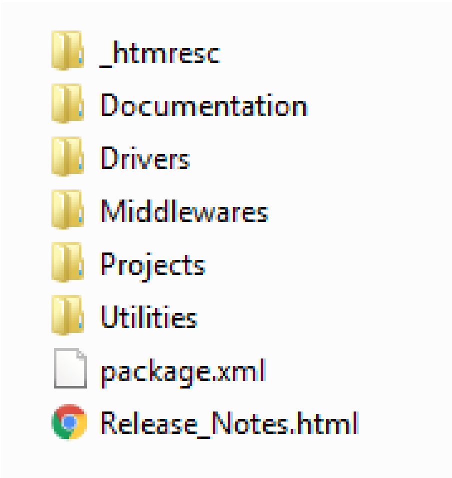
- **STM32Cube HAL layer**: provides a simple, generic and multi-instance set of generic and extension APIs (application programming interfaces) to interact with the upper layer application, libraries and stacks. It is directly based on a generic architecture and allows the layers that are built on it, 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-SENSING1 software architecture**



1.3 Folder structure

Figure 2. **FP-AI-SENSING1 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 Bluetooth low energy, USB Device Library, Generic FAT File System Module (FatFs), FreeRTOS real time operating system, Meta Data Manager for saving meta data on Flash and AI Middleware libraries.

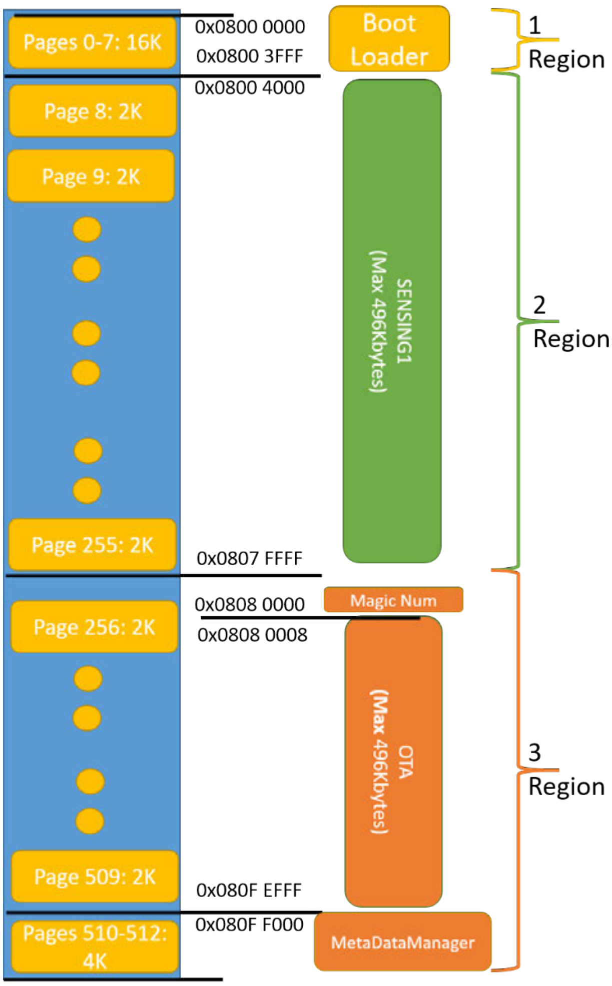
- **Projects**: contains a sample application used for transmitting the output of the sensors data and the results of the Audio Scene Classification or Activity Recognition algorithm over Bluetooth low energy protocol provided for the NUCLEO-L476RG, STEVAL-STLKT01V1, B-L475E-IOT01A and STEVAL-MKSBOX1V1 platforms through the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit (MDK-ARM) and System Workbench for STM32 development environments.
  - **Utilities**: contains the boot loader binaries ready to be flashed for the NUCLEO-L476RG, STEVAL-STLKT01V1 and B-L475E-IOT01A or to the STEVAL-MKSBOX1V1 boards and the Trace Recorder Library for FreeRTOS.

1.4 Flash management

Apart from storing code, FP-AI-SENSING1 uses the Flash memory for Firmware-Over-The-Air updates. It is divided into the following regions:

- the first region contains a custom boot loader
- the second region contains the FP-AI-SENSING1 firmware
- the third region is used for storing the FOTA before the update
- the Meta Data Manager is placed at the end of the Flash

Figure 3. FP-AI-SENSING1 Flash structure



RELATED LINKS

RM0351 Reference manual STM32L4x6 advanced ARM®-based 32-bit MCUs

1.5 The boot process

The FP-AI-SENSING1 cannot not be flashed at the beginning of the Flash (address 0x08000000), and is therefore compiled to run from the beginning of the second flash region, at 0x08004000.

To enable this behavior, we set the vector table offset in Src/system_stm32l4xx.c (for STM32L476): `#define VECT_TAB_OFFSET 0x4000`.

The FP-AI-SENSING1 function pack features several boards based on two different STM32 processors:

- STM32L476RG processor, with 1 MByte of Flash memory, for STEVAL-STLKT01V1 and for B-L475E-IOT01A STM32L4 Discovery kit IoT node
- STM32L4R9ZI processor, with 2 MBytes of Flash memory, for STEVAL-MKSBOX1V1 evaluation board

Important:
Due to the different Flash memory size, the maximum code size usable for STM32L4R9ZI is the double of the one usable for STM32L476RG processor. Thus, this section takes into account only the example of the modification and the boot process for the STM32L476RG processor.

We also changed the linker script. For example, for IAR Embedded Workbench for ARM, the script is:

```
define symbol   ICFEDIT_intvec_start = 0x08004000;
/*-Memory Regions-*/
define symbol   ICFEDIT_region_ROM_start = 0x08004000; define symbol   ICFEDIT_region_ROM_end
 = 0x0807FFFF;
```

Using the above linker script, the maximum usable code size is fixed at 496 KB.

You must Flash the appropriate bootloader binary for STM32L476RG in the Utilities\BootLoader folder to the first FLASH region (address 0x08000000).

On any board reset:

- the bootloader checks the presence of a FOTA magic number in the Flash memory which indicates FOTA (full or partial) is present in the third Flash region:
  - if the FOTA magic number is present, the bootloader retrieves the size and destination address of the FOTA from the Flash region 3 and copies the FOTA at the specified address. The FOTA magic number is cleared and the board is restarted

Note: *In case of full FOTA the size is the maximum and the destination address is the region 2 start address.*

  - otherwise, the boot loader jumps to the FP-AI-SENSING1 firmware.

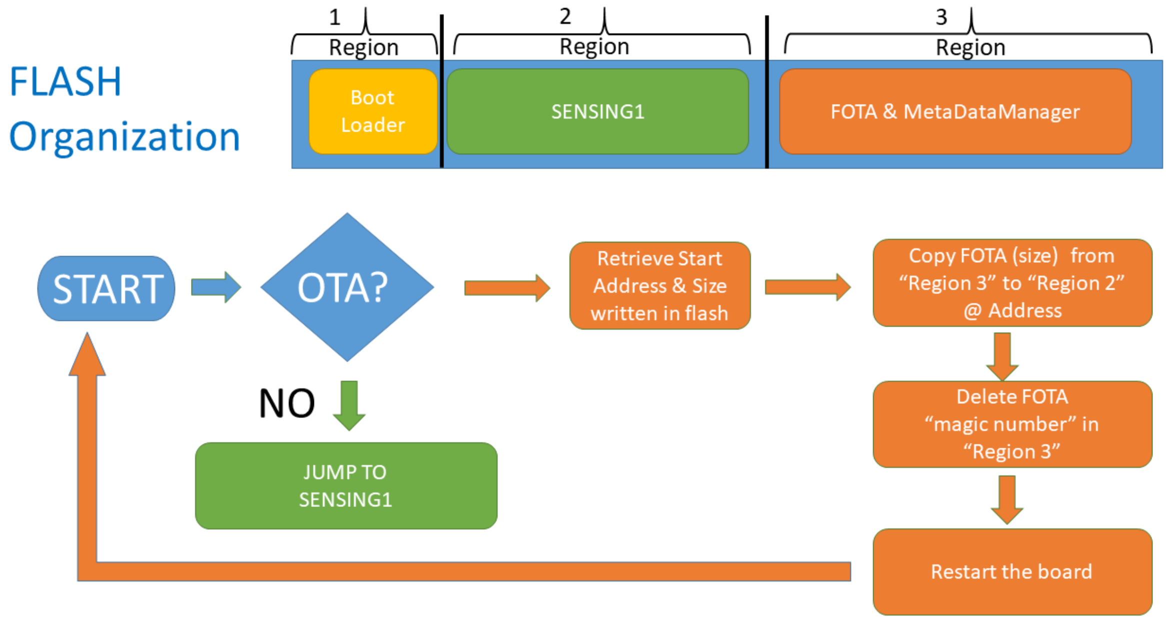


Figure 4. **FP-AI-SENSING1 boot sequence**

1.6 The installation process

Each platform (NUCLEO-L476RG, STEVAL-STLKT01V1, B-L475E-IOT01A and STEVAL-MKSBOX1V1) SENSING1 application contains a Binary directory, including:

- **SENSING1.bin**: pre-compiled application binaries to be programmed to the correct memory address, (0x08004000) using ST-LINK Utility

Note: These pre-compiled binaries are compatible with the FOTA update procedure.

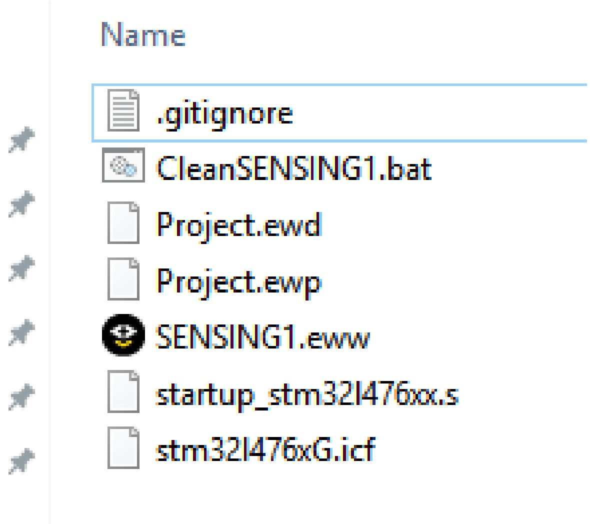
- **SENSING1_BL.bin**: pre-compiled binaries combining both the bootloader and the application. They can be programmed directly using ST-LINK Utility or drag-and-drop (for STM32 Nucleo and IoT node boards only)

Note: These pre-compiled binary are not compatible with the FOTA update procedure.

To flash modified SENSING1 firmware, simply flash the compiled firmware to the correct address (0x08004000).

The **CleanSENSING1.bat** script is provided to simplify this operation by saving the firmware and the BootLoader to the right position; it is available for each platform (NUCLEO-L476RG, STEVAL-STLKT01V1, B-L475E-IOT01A and STEVAL-MKSBOX1V1) and for each IDE (IAR/RealView/System Workbench).

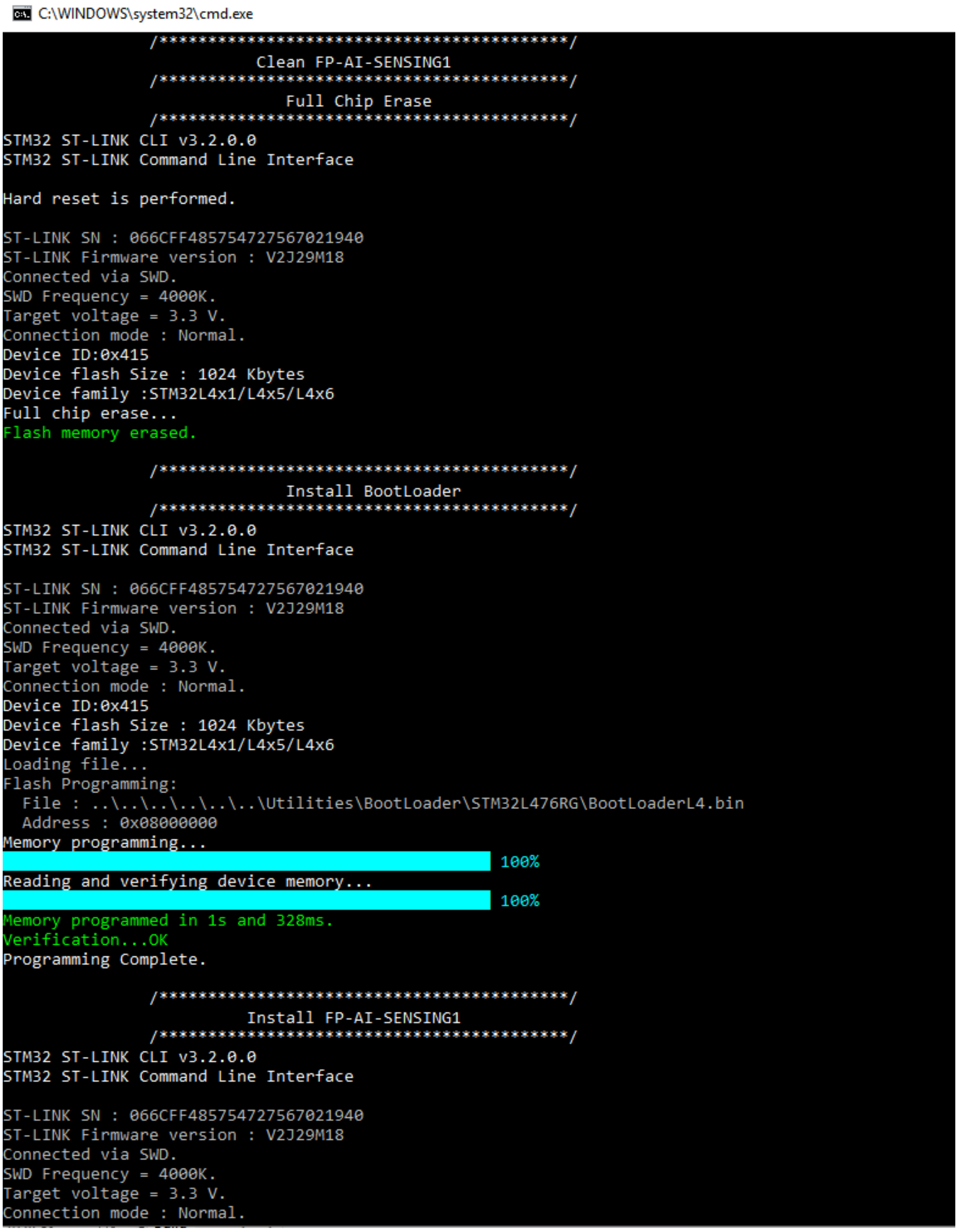
Figure 5. **Content of a project folder**



The script performs the following operations:

1. Performs a full Flash erase to start from a clean system.
2. Flashes the BootLoader to the correct position 0x08000000.
3. Flashes the firmware to the correct position 0x08004000.

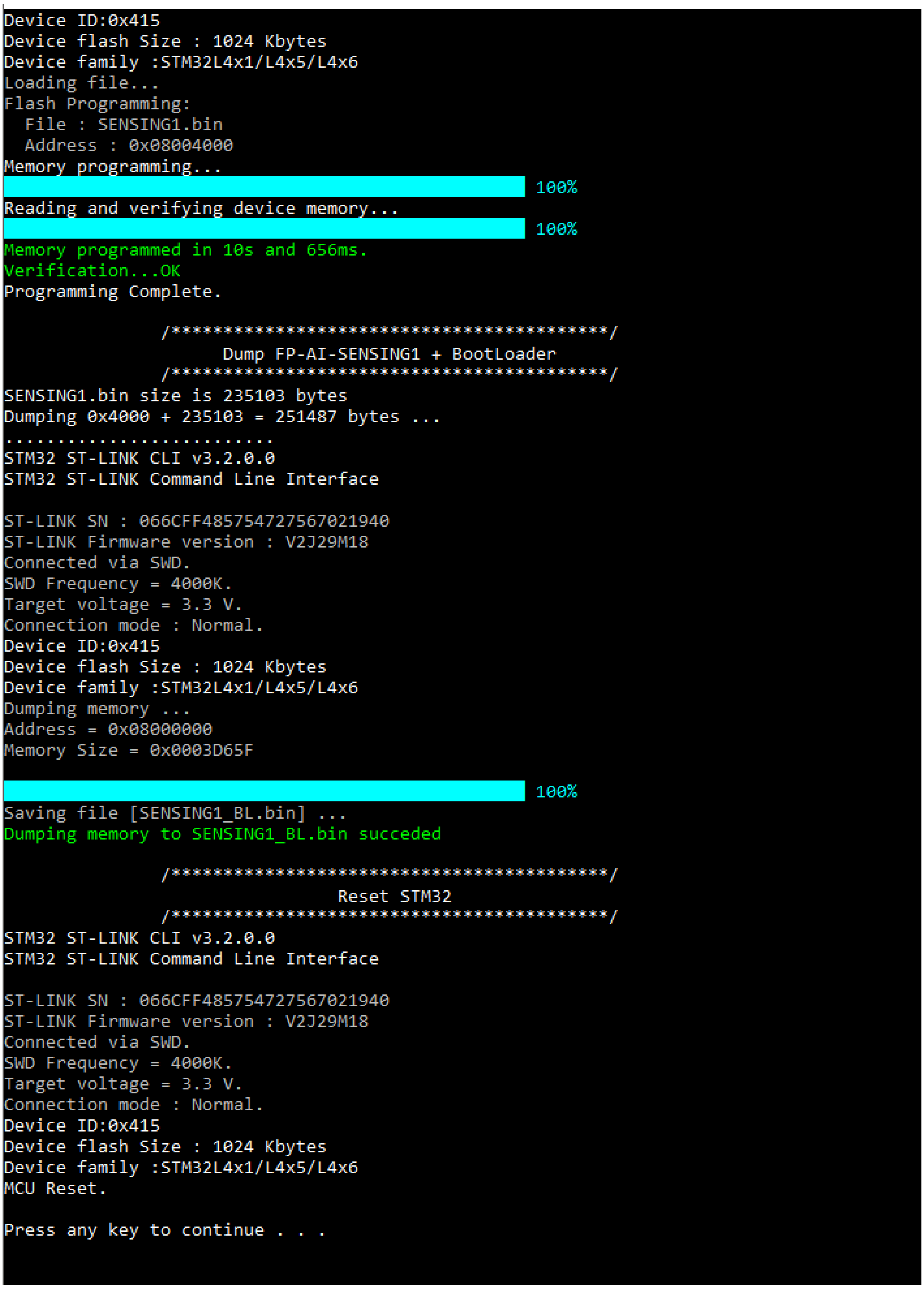
Figure 6. BootLoader and SENSING1 installation



```
C:\WINDOWS\system32\cmd.exe
                /*********************************************/
                           Clean FP-AI-SENSING1
                /*********************************************/
                              Full Chip Erase
                /*********************************************/
STM32 ST-LINK CLI v3.2.0.0
STM32 ST-LINK Command Line Interface

Hard reset is performed.

ST-LINK SN : 066CFF485754727567021940
ST-LINK Firmware version : V2J29M18
Connected via SWD.
SWD Frequency = 4000K.
Target voltage = 3.3 V.
Connection mode : Normal.
Device ID:0x415
Device flash Size : 1024 Kbytes
Device family :STM32L4x1/L4x5/L4x6
Full chip erase...
Flash memory erased.

                /*********************************************/
                              Install BootLoader
                /*********************************************/
STM32 ST-LINK CLI v3.2.0.0
STM32 ST-LINK Command Line Interface

ST-LINK SN : 066CFF485754727567021940
ST-LINK Firmware version : V2J29M18
Connected via SWD.
SWD Frequency = 4000K.
Target voltage = 3.3 V.
Connection mode : Normal.
Device ID:0x415
Device flash Size : 1024 Kbytes
Device family :STM32L4x1/L4x5/L4x6
Loading file...
Flash Programming:
  File : ..\..\..\..\..\..\Utilities\BootLoader\STM32L476RG\BootLoaderL4.bin
  Address : 0x08000000
Memory programming...
                                                  100%
Reading and verifying device memory...
                                                  100%
Memory programmed in 1s and 328ms.
Verification...OK
Programming Complete.

                /*********************************************/
                          Install FP-AI-SENSING1
                /*********************************************/
STM32 ST-LINK CLI v3.2.0.0
STM32 ST-LINK Command Line Interface

ST-LINK SN : 066CFF485754727567021940
ST-LINK Firmware version : V2J29M18
Connected via SWD.
SWD Frequency = 4000K.
Target voltage = 3.3 V.
Connection mode : Normal.
```

The script also dumps an image containing the BootLoader and the firmware, which can be directly flashed to the beginning of the Flash memory in the same way as the image provided in the Binary folder.

Figure 7. **SENSING1 dump process**



The **CleanSENSING1.sh** script for Linux or macOS operating systems uses OpenOCD instead of the ST-LINK command line; it is only included in the System Workbench IDE.

To function, the script must be modified with:

- The installation path for OpenOCD
- The installation path for STM32 OpenOCD scritps
- And the Library path for OpenOCD

Below is the section of the OpenOCD script to be edited:

```
# 1) Set the Installation path for OpenOCD
# example:
#OpenOCD_DIR="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.externaltools.openocd.win3 2_1.17.0.2
01801120948/tools/openocd/"
OpenOCD_DIR=""
# 2) Set the installation path for stm32 OpenOCD scritps
# example:
#OpenOCD_CFC="C:/Ac6/SystemWorkbench/plugins/fr.ac6.mcu.debug_2.1.4.201801120948/resources/op
enocd/scripts" OpenOCD_CFC=""
# 3) Only for Linux/macOS add openocd library path to _LIBRARY_PATH:
# For macOS example:
#export DYLD_LIBRARY_PATH=${DYLD_LIBRARY_PATH}:${OpenOCD_DIR}"lib/"
# For Linux example:
#export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${OpenOCD_DIR}"lib/"
```

1.7 Generation of a partial update binary file for FOTA

The Cube.AI tool is provided with a a Command Line Interface that generates a binary file containing a descriptive header, compatible with partial FOTA process along with the updated neural network weights described by its model.

The command to be issued is

```
stm32ai.exe generate --model model.h5 --compress 4 --binary --generate.fota True
```

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

The FP-AI-SENSING1 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-SENSING1 checks against the hardware cyclic redundancy check calculation unit on the STM32L476 processor to ensure integrity. If the CRC calculation matches the STBLESensor CRC value, the new firmware or the updated neural network weights are written to the beginning of the third Flash region. A "magic number" setting signals the boot loader that a Firmware update has been received and checked, and is ready to replace the current FP-AI-SENSING1 firmware or neural network weights (see Firmware-Over-The-Air update with STBLESensor).

1.9 APIs

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

1.9.1 Sample application description

The SENSING1 sample application can be used with:

- the X-NUCLEO-IKS01A2, X-NUCLEO-CCA02M1 and X-NUCLEO-IDB05A1 expansion boards with the NUCLEO-L476RG board
- the STEVAL-STLKT01V1 evaluation board
- the B-L475E-IOT01A STM32L4 Discovery kit IoT node
- the STEVAL-MKSBOX1V1 evaluation board

The sample application features three Human Activity Recognition (HAR) algorithms using the motion sensors and one Acoustic Scene Classification (ASC) algorithm.

A sample application project is available for each board supporting the following IDEs: IAR 8.32, Keil 5.27 and SW4STM32 2.9.

1.9.2 Human activity recognition (HAR)

The FP-AI-SENSING1 software features the following activity recognition algorithms based on neural networks:

- **HAR_GMP**: ST proprietary design trained on an ST proprietary data set

- **HAR_IGN**: ST simplified design taken from *Andrey Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks", Applied Soft Computing 62 (2018), pp 915-922* trained on an ST proprietary data set
- **HAR_IGN_WSDM**: same network topology as HAR_IGN but trained on the public Wireless Sensor Data Mining (WSDM) dataset in *Jennifer R. Kwapisz, Gary M. Weiss and Samuel A. Moore. "Activity Recognition using Cell Phone Accelerometers" in ACM SIGKDD Exploration Newsletter, volume 12 issue 2, December 2010, pp 74-82*.

The HAR_GMP or HAR_IGN configurations can signal one of the following recognized activities:

- stationary
- walking
- jogging
- biking
- driving

The HAR_IGN_WSDM configuration can signal one of the following recognized activities:

- stationary
- walking
- jogging
- stairs

When the HAR_IGN_WSDM configuration is selected, you can also use a Playback Mode with test vectors instead of live motion sensor data. Playback Mode can be enabled using `#define TEST_IGN_WSDM` in the file SENSING1.h:

```
/* For HAR_IGN_WSDM playback mode (use test vectors instead of sensors) */
#ifdef NN_IGN_WSDM
  // #define TEST_IGN_WSDM
#endif /* NN_IGN_WSDM */
```

A preprocessing step is applied to the raw sensor data before being processed by the neural network. The step is specific to the type of class being considered and cannot be generalized for any scenario based on motion MEMS.

For human activity recognition, a high pass filter (4th order, cutoff frequency around 1 Hz) is used to separate the gravity component from the dynamic (oscillating) part of the acceleration. The dynamic component of the gravity is rotated to always points in the same direction regardless of the sensor's orientation.

Rodrigues' rotation formula is used to rotate the dynamic part.

For HAR_GMP, 24x3 data matrix (24 x-axis, 24 y-axis, 24 z-axis) is given as input every 0.62 ms to the neural network.

For HAR_IGN, it is every 0.92 s due to the fact that there is no overlapping.

A training script for HAR is provided in the `Utilities\AI_Resources\Training Scripts\HAR` folder along with a Jupyter Notebook to explain all the steps taken.

Note: Only a dummy data set is provided.

RELATED LINKS

Rodrigues' rotation formula at wikipedia

1.9.3 Acoustic scene classification (ASC)

The FP-AI-SENSING1 software features one Acoustic Scene Classification (ASC) algorithm based on neural networks:

- **ASC**: ST simplified design taken from *Valenti, M., Diment, A., Parascandolo, G., Squartini, S., & Virtanen, T", "DCASE 2016 acoustic scene classification using convolutional neural networks. In Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2016), Budapest, Hungary* trained on an ST proprietary data set

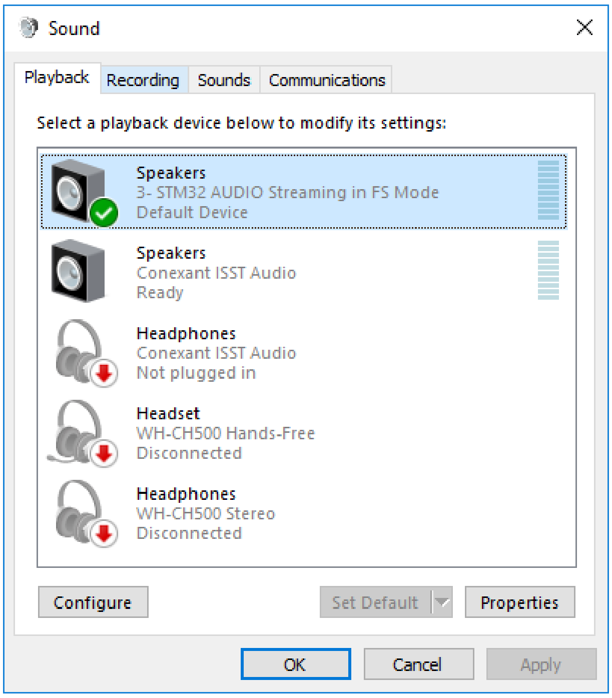
Running the ASC configuration enables the [**Audio Classification**] menu in the STBLESensor smartphone client application. The classification result can then be monitored using the smartphone client application to signal one of the following recognized audio scenes:

- Indoor
- Outdoor
- In-Vehicle

By default, the ASC configuration uses the on-board microphone on the STEVAL-STLKT01V1, the X-NUCLEO-CCA02M1 expansion board, the B-L475E-IOT01A or the STEVAL-MKSBOX1V1 for the audio input signal. A Playback mode is also available in the sample application.

On both the B-L475E-IOT01A and the X-NUCLEO-CCA02M1, the `#define SENSING1_USE_USB_AUDIO` can be enabled in the file SENSING1_config.h to play audio using a host PC by selecting the STM32 Audio Streaming in FS Mode device, as shown in the following figure.

Figure 8. STM32 Audio device selection in Windows



You can then use your PC to play the sample audio file provided with the AI middleware (FP-AI-SENSING1/ Firmware/Middlewares/ST/STM32_AI_Library/Src/bus.wav)

*Note: The USB Audio device is only instantiated after the Audio Classification algorithm is initialized; i.e., when you open the [**Audio Classification**] menu in the STBLESensor smartphone client application.*

```
/* For using USB Audio instead of Microphones */
#define SENSING1_USE_USB_AUDIO
```

The ASC configuration captures audio at a 16 kHz (16-bit, 1 channel) rate using the on-board MEMS microphone. Every millisecond, a DMA interrupt is received with the last 16 PCM audio samples. These samples are then accumulated in a sliding window consisting of 1024 samples with a 50% overlap. For every 512 samples (i.e., 32 ms), the buffer is injected into the ASC preprocessing for feature extraction.

The ASC preprocessing extracts audio features into a LogMel (30x32) spectrogram. For computational efficiency and memory management optimization, the step is divided into two routines:

1. The first part computes one of the 32 spectrogram columns from the time domain input signal into the Mel scale using FFT and Filter bank applications (30 mel bands).
2. The second part, when all 32 columns have been calculated (i.e., after 1024 ms), a log scaling is applied to the mel scaled spectrogram, creating the input feature for the ASC convolutional neural network.

Every 1024ms, the (30x32) LogMel spectrogram is fed to the ASC convolutional neural network input, which can then classify the output labels: indoor, outdoor and in-vehicle.

A training script for HAR is provided in the `Utilities\AI_Resources\Training Scripts\HAR` folder along with a Jupyter Notebook to explain all the steps taken.

Note: Only a dummy data set is provided.

1.9.3.1 Quantization

The floating-point ASC model has been post-quantized to integer (weight and activations) using the optimizer from Tensor Flow Lite converter: dynamic range of activations is measured with the injection of a representative dataset.

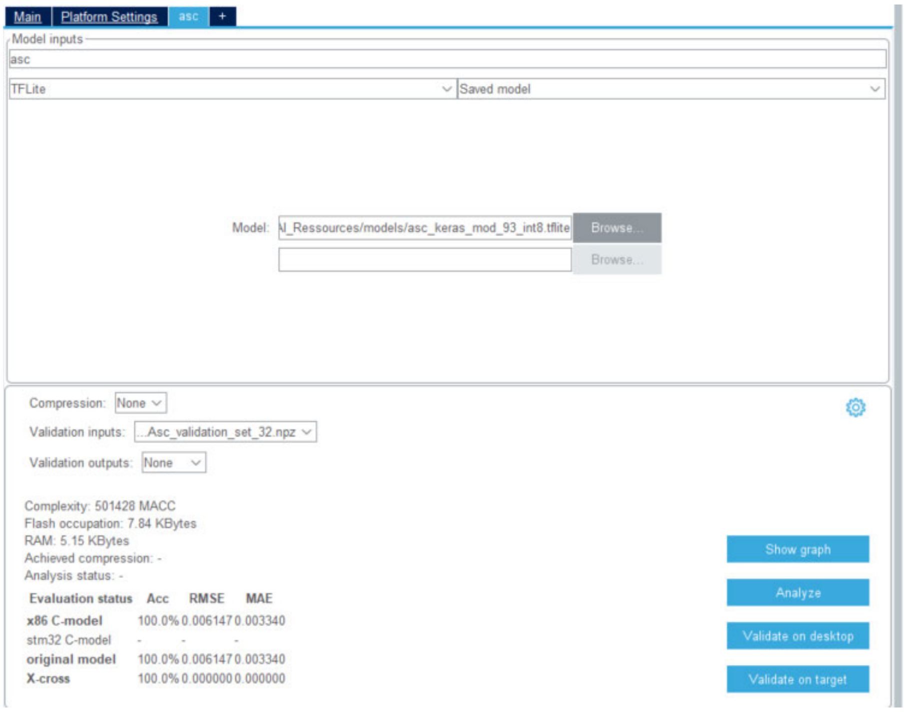
The script that converts the original keras model can be found under `Utilities\AI_Ressources\TFlite\ASC\asc_keras_to_tflite_full_int8.py`

Note: The quantization process requires a sample dataset that is not included in this package. A dummy dataset is provided instead so that the script can be executed properly.

The quantization process output can be found under `Utilities\AI_Ressources\models\asc_keras_mod_93_int8.tflite`.

X-CUBE-AI v4.1.0 uses these files to generate the STM32 fixed-point implementation below.

Figure 9. X-CUBE-AI: STM32 fixed-point implementation



The table below shows the tradeoffs between the original floating point implementation of the ASC mode and the fixed-point (quantized) implementation.

Table 1. Float vs. quantized implementation

Implementation	RO data (Bytes)	RW data (Bytes)	Inference time (ms)	Accuracy
Single precision float	30,812	21,260	82,600	100%
Quantized	7,840	5,150	42,373	100%

Note: The report accuracy was obtained using a test dataset of 12 samples and compared against ground truth values.

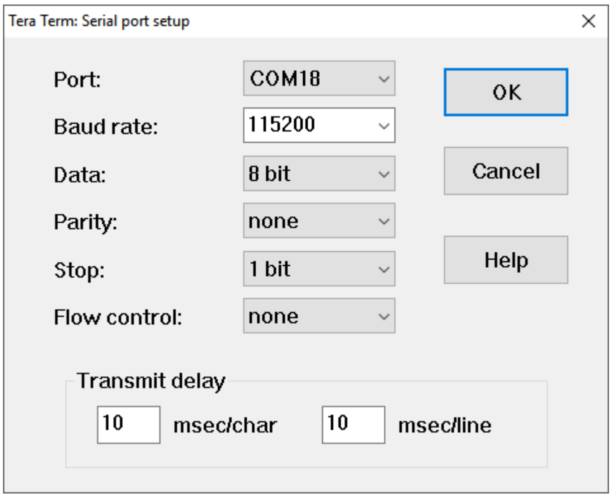
RELATED LINKS

For the original floating-point ASC model, refer to UM2526, "Getting started with X-CUBE-AI Expansion Package for Artificial Intelligence (AI)"

1.9.4 Setting up the terminal window

With the NUCLEO-L476RG and B-L475E-IOT01A boards, you can set up a terminal window for the appropriate UART communication port to control the initialization phase.

Figure 10. Terminal setting



The same feature is available for the STEVAL-STLKT01V1 evaluation board when connecting the micro-USB port to a PC. However, as it is necessary to register the USB device, this is only possible when the STEVAL-STLKT01V1 starts initializing. A 10-second delay is added to the initialization phase to give you time to follow the process.

You must modify the SENSING1_config.h file by enabling the `/*#define SENSING1_USE_PRINTF*/` feature for the STEVAL-STLKT01V1, as it is disabled by default.

```
/**
 * @brief Enable printf
 *        When enabled on the SensorTile and SensorTile.box, printf is
 *        redirected to USB CDC. It will introduce a delay of 10 s before
 *        starting the application for having time to open the Terminal.
 *        On other platforms, printf is redirected to a UART interface.
 */
```

Note: For the STEVAL-MKSBOX1V1 this feature is not available at the moment.

1.9.5 Sample application startup description

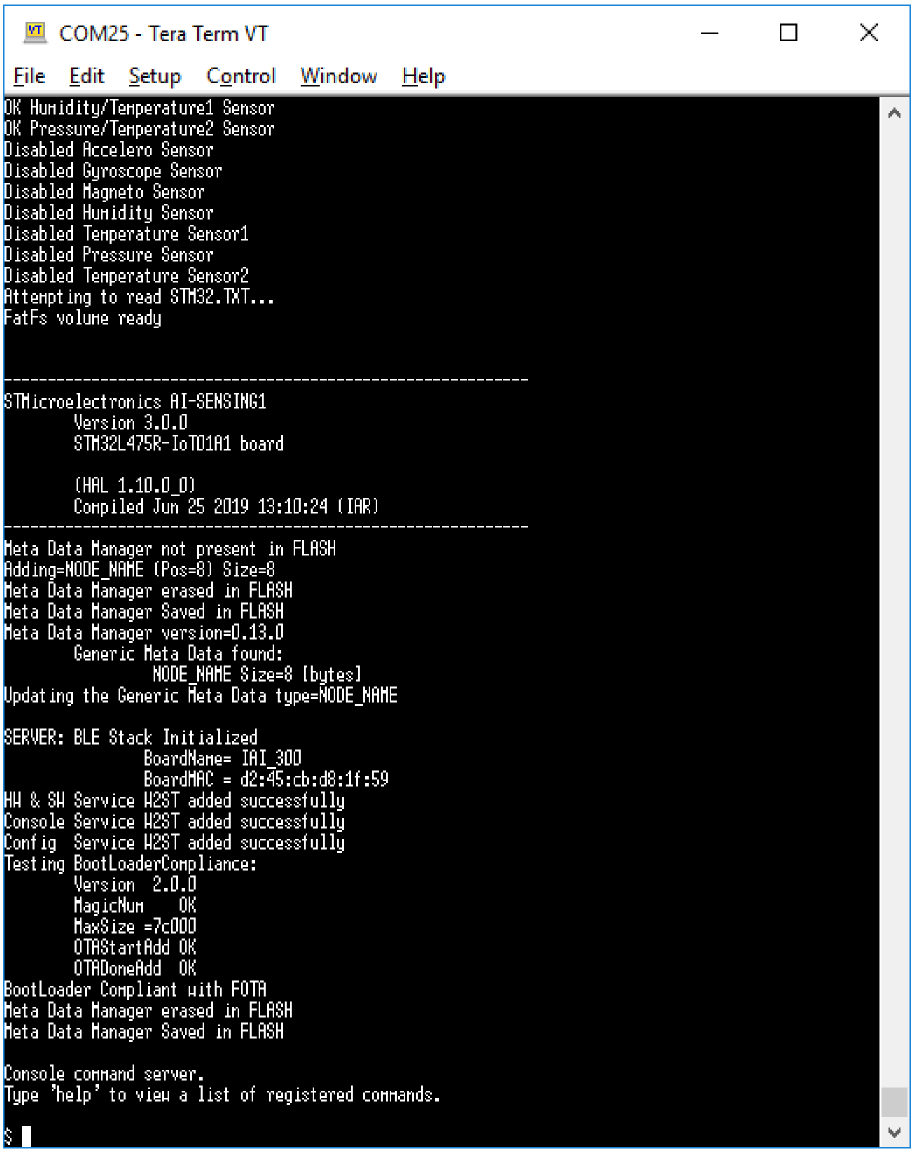
When you first press the reset button, the application:

- starts initializing the UART, I²C and SPI interfaces
- checks whether all the sensors are present and working
- checks that the BlueNRG expansion board is connected to the STM32 Nucleo board, and reads the hardware and firmware version information
- creates a random BLE MAC address
- initializes the BLE hardware service (adding the temperature, humidity, pressure, 3D gyroscope, 3D magnetometer, 3D accelerometer, microphone and Gas Gauge IC characteristics) and the BLE software service

- initializes the BLE console service adding the stdin/stdout and stderr characteristics
- Initializes the BLE configuration service to enable the hardware features for LSM6DSL mounted on the X-NUCLEO-IKS01A2 expansion board (for NUCLEO-L476RG only).

It can generate an interrupt due to free fall, tilt, wake up, single tap, double tap, 6D position or pedometer events, and transmit it via Bluetooth to the connected Android or iOS device.

Figure 11. Initialization phase



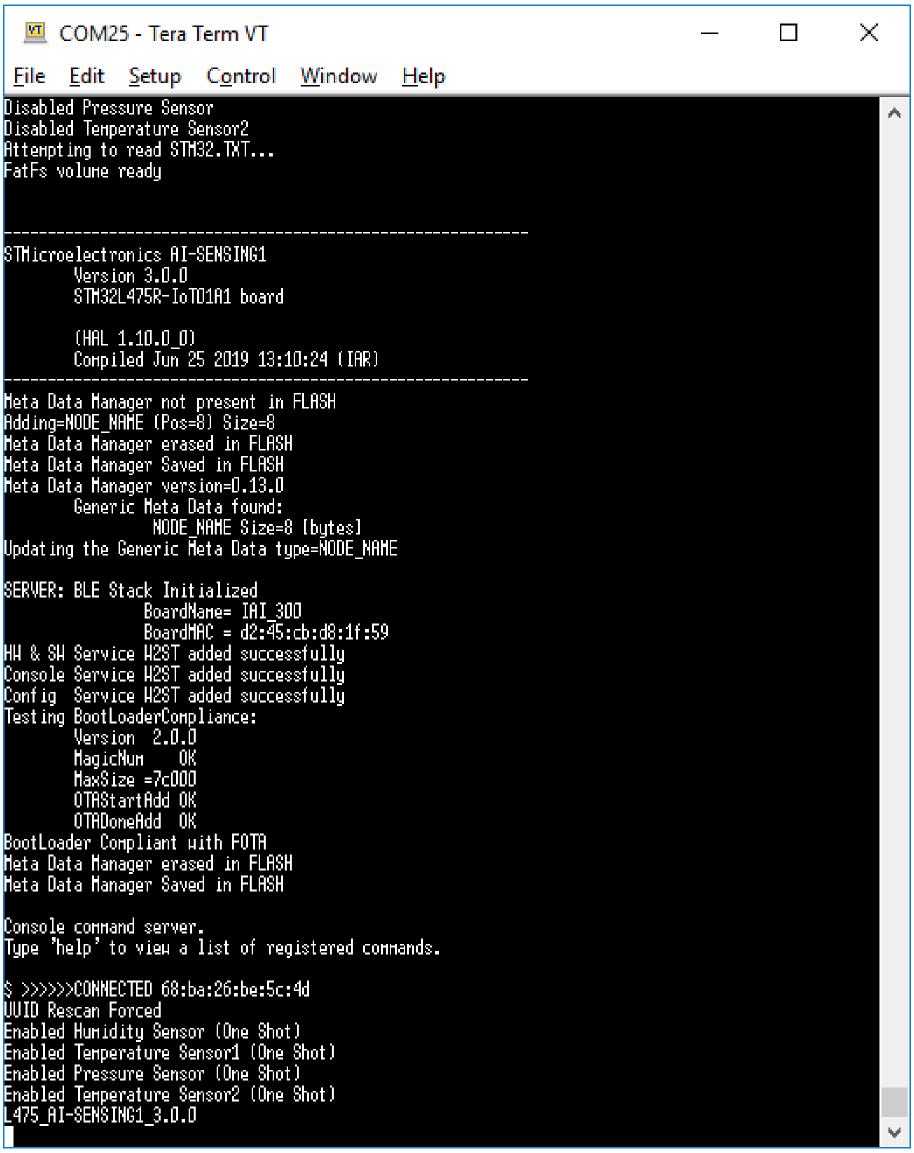
As shown in the console output above, the application sends:

- temperature/humidity/pressure data every 500 ms
- 3D accelerometer, 3D gyroscope and 3D magnetometer data every 50 ms
- signal noise microphone levels every 50 ms

The FatFs library provides access to the storage devices for sensor data logging (feature available for STEVAL-STLCS01V1, STEVAL-MKSBOX1V1 and for B-L475E-IOT01A STM32L4 Discovery kit IoT node).

When an Android/iOS device is connected to one supported board (if the define `#define SENSING1_USE_PRINTF` is enabled), you can control data transmitted via the board.

Figure 12. UART console output when a device is connected to the board



1.10 Android and iOS sample client application

Developed to reduce power consumption, the FP-AI-SENSING1 software for STM32Cube is compatible with the STBLESensor Android/iOS applications (ver. 4.1.0 or higher) available at the respective Play/iOS stores.

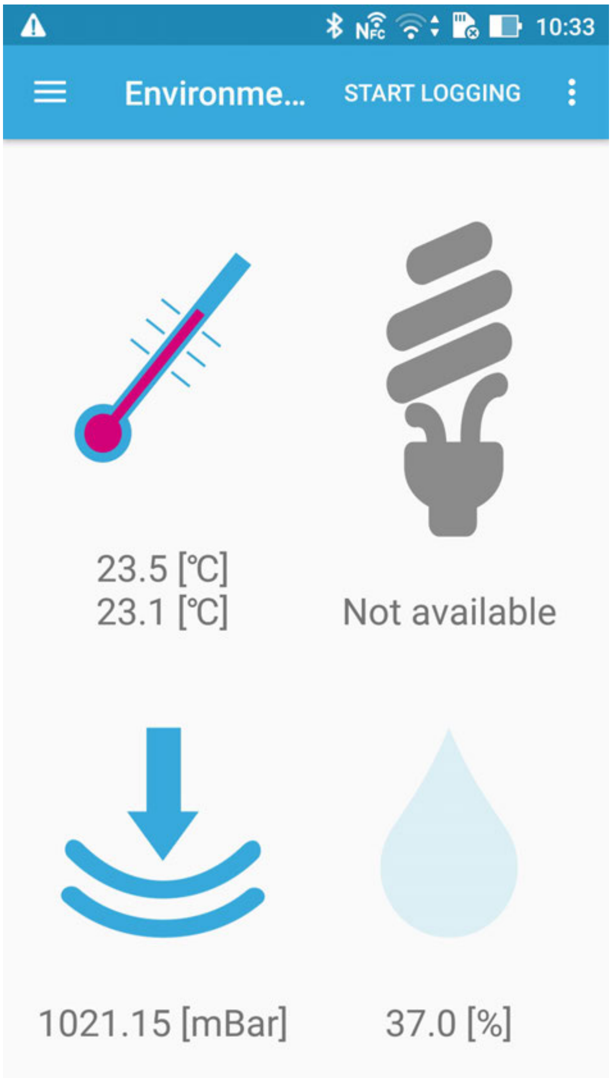
The STBLESensor Android/iOS application allows Over-The-Air firmware updates and version 3.2.0 or higher is required to display battery information (remaining charge, voltage and charge status) for the STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1 evaluation boards.

We will use the Android application for this demonstration.

1.10.1 Main page

Following connection, STBLESensor opens the main page with temperature, pressure and humidity readings.

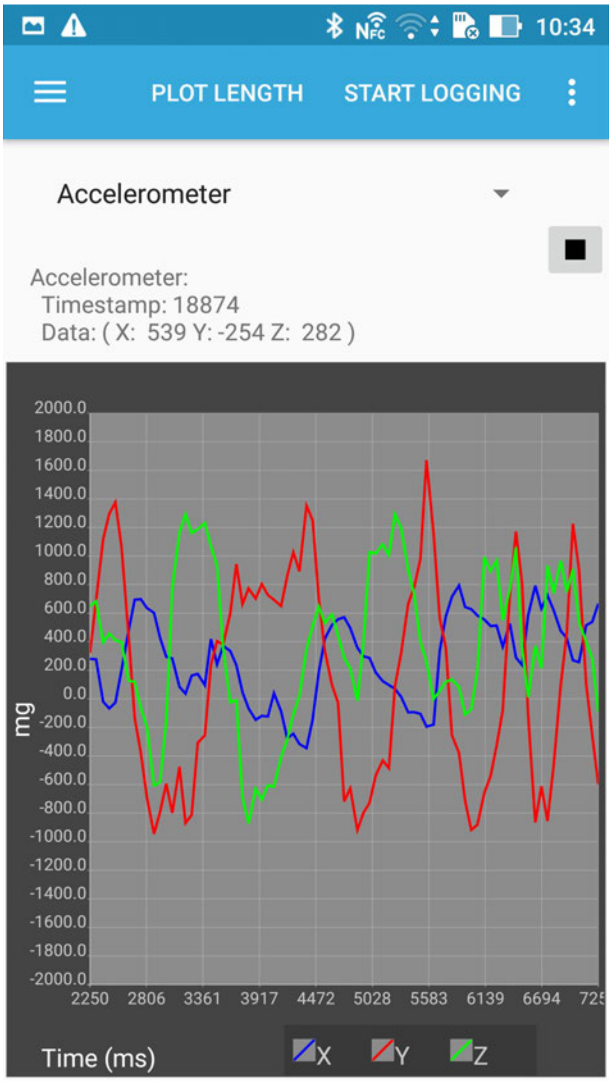
Figure 13. STBLESensor (Android version) main page following BLE connection



1.10.2 Plot data

On the next page to the left, you can plot any value from the sensor expansion boards.

Figure 14. STBLESensor (Android version) accelerometer plot

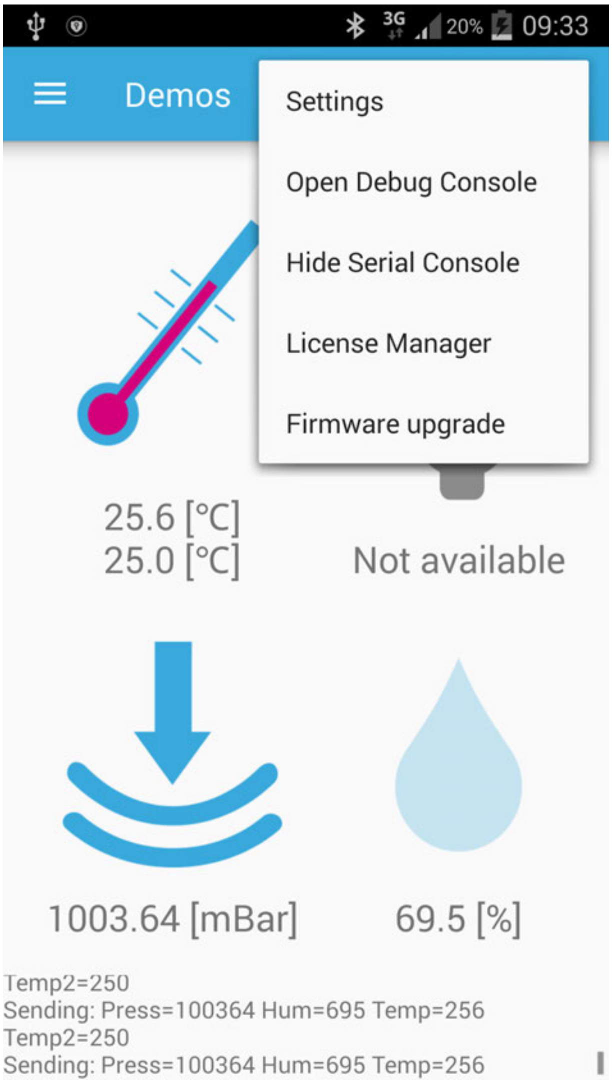


1.10.3 Settings, serial and debug console

In the option menu below, you can open:

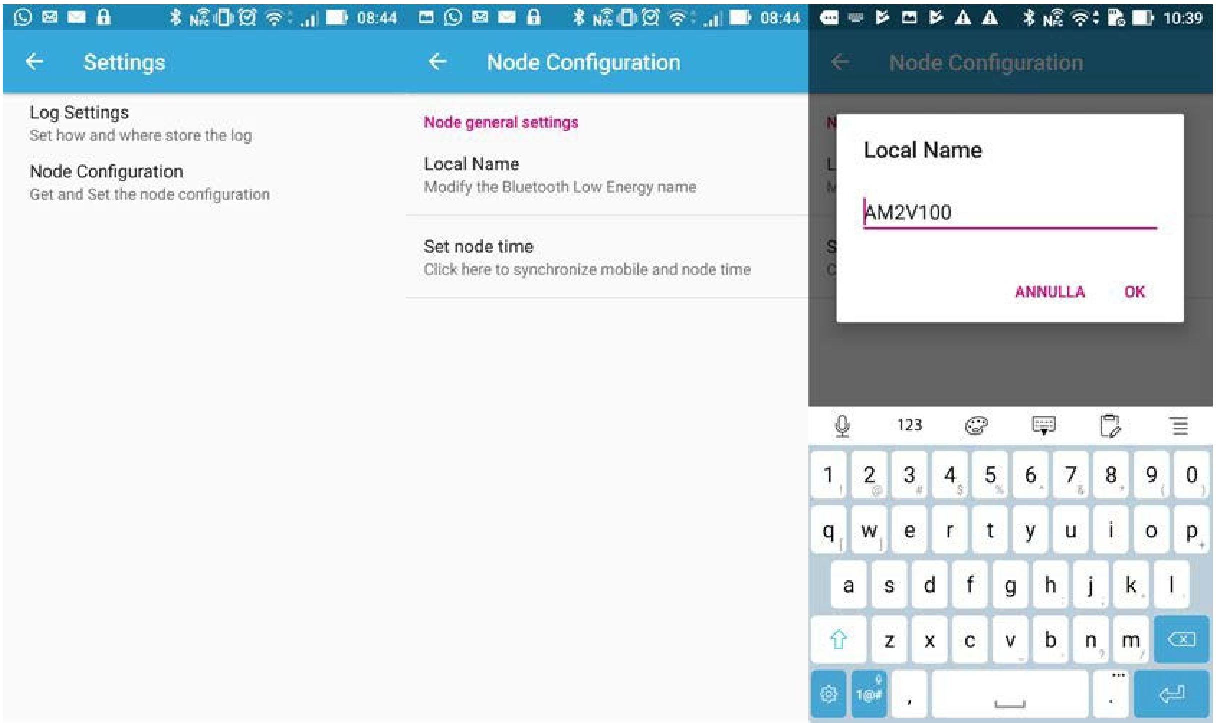
- Settings
- Serial or Debug (with stdin) console
- Firmware upgrade

Figure 15. STBLESensor (Android version) menu selection



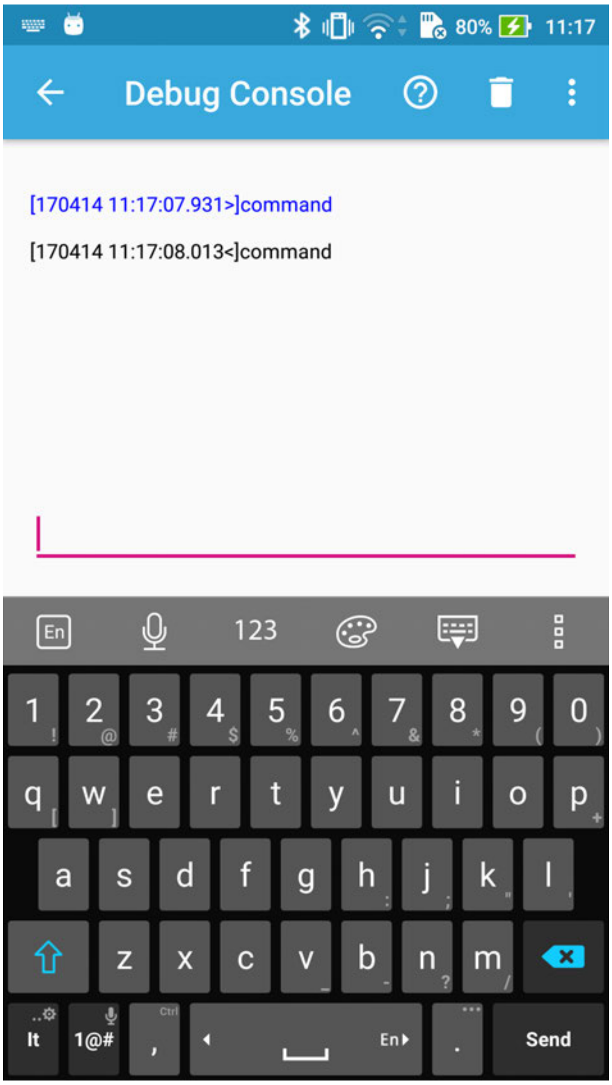
You can change the node name in [**Settings**]>[**Node Configuration**]>[**Local Name**]:

Figure 16. STBLESensor (Android version) Local Name change



If the Serial console is enabled, stdout/stderr is displayed, as shown below.

Figure 17. STBLESensor (Android version) Debug console (stdin/stdout/stderr)

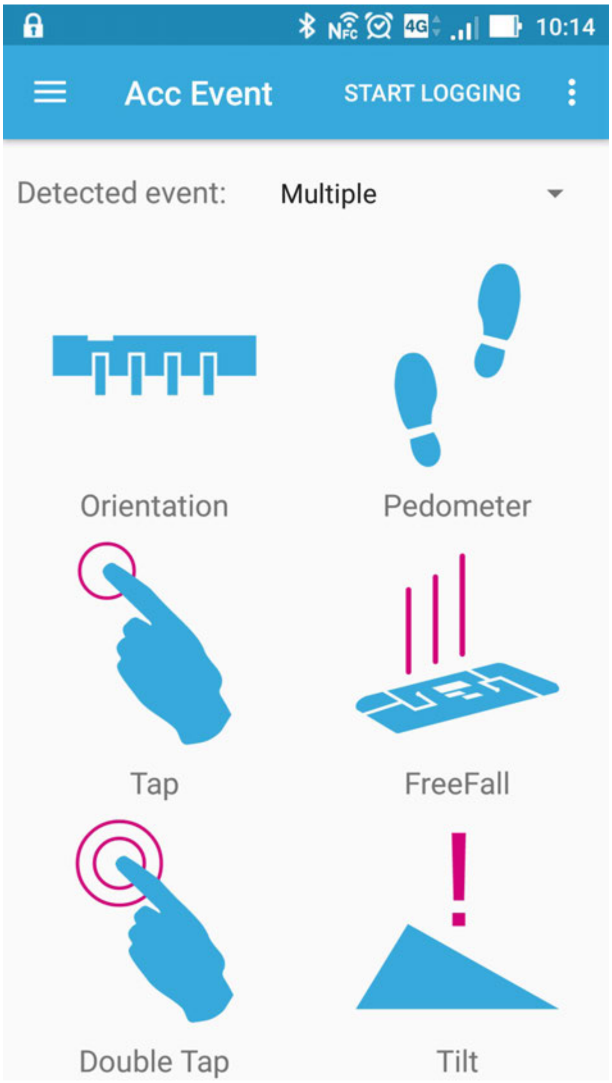


1.10.4 Enable hardware features

Another page in the app lets you choose which hardware feature to enable (one at a time) and view the readings on the same page from:

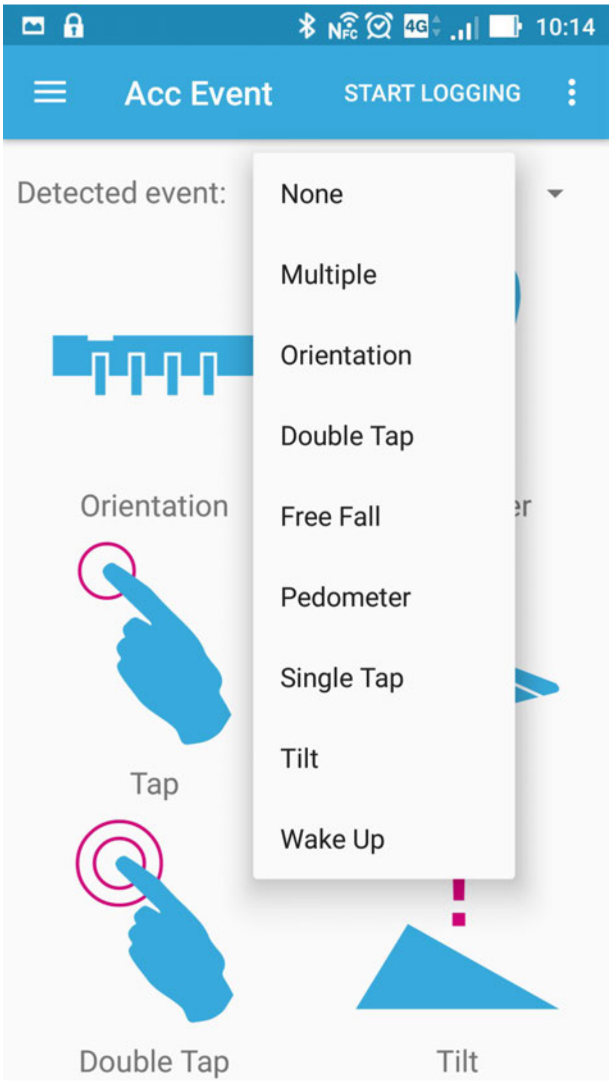
- LSM6DSL on X-NUCLEO-IKS01A2 expansion board for NUCLEO-L476RG board only
- LSM6DSM for STEVAL-BCNKT01V1 and STEVAL-STLKT01V1 boards. The multiple hardware feature is the default setting
- LSM6DS0 for STEVAL-MKSBOX1V1 evaluation board

Figure 18. STBLESensor (Android version) multiple hardware feature



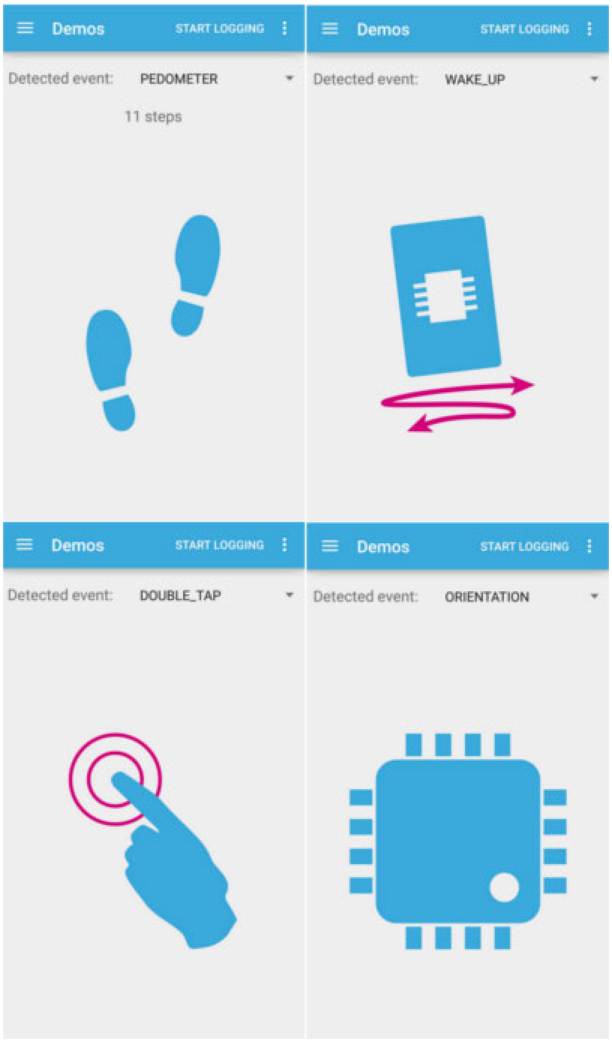
You can select one feature at a time from the [**Acc Event**] menu.

Figure 19. STBLESensor (Android version) hardware feature menu



Below are some examples of readings for selected accelerometer events.

Figure 20. STBLESensor (Android version) hardware feature examples



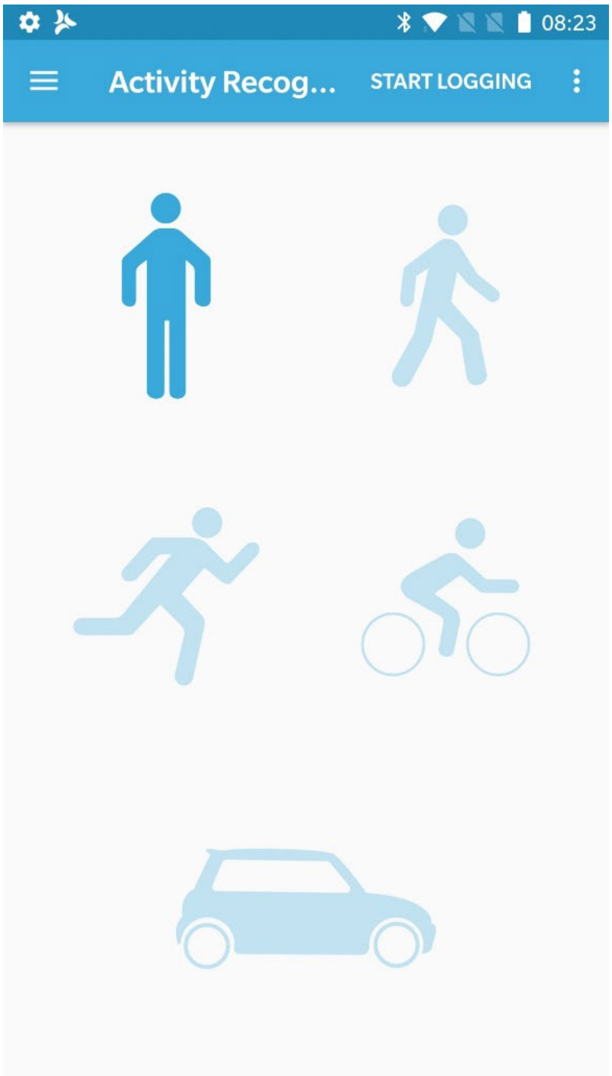
1.10.5 Activity recognition

The Activity Recognition page can be used to monitor the AI neural network classification results from one of three HAR algorithms provided with FP-AI-SENSING1. Depending on which HAR configuration has been selected, the page may differ.

If HAR_GMP or HAR_IGN configuration is selected, the page shows one of the following recognized activities:

- stationary
- walking
- jogging
- biking
- driving

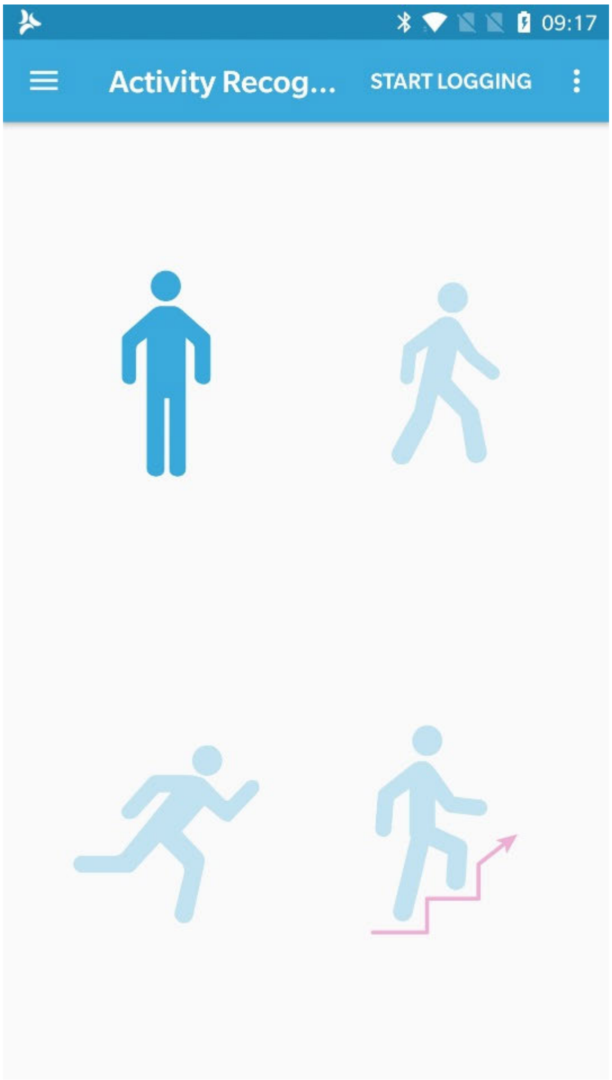
Figure 21. STBLESensor (Android version) HAR_GMP or HAR_IGN activity recognition page



If the HAR_IGN_WSDM configuration is selected, the page will begin signaling one of the following recognized activities:

- stationary
- walking
- jogging
- stairs

Figure 22. STBLESensor (Android version) HAR_IGN_WSDM activity recognition page



As the algorithm has to collect data before recognizing any activity, all the images are shown in grey for few seconds after the demo starts.

1.10.6 Audio Classification

The Audio Classification page can be used to monitor the AI neural network classification results from the ASC algorithm provided with FP-AI-SENSING1, signaling one of the following recognized audio scenes:

- Indoor
- Outdoor
- In-Vehicle

Figure 23. STBLESensor (Android version) ASC Audio Classification page

As the algorithm has to collect data before recognizing any audio scene, all the images are shown in grey for few seconds after the demo starts.

1.10.7 AI Multi Network

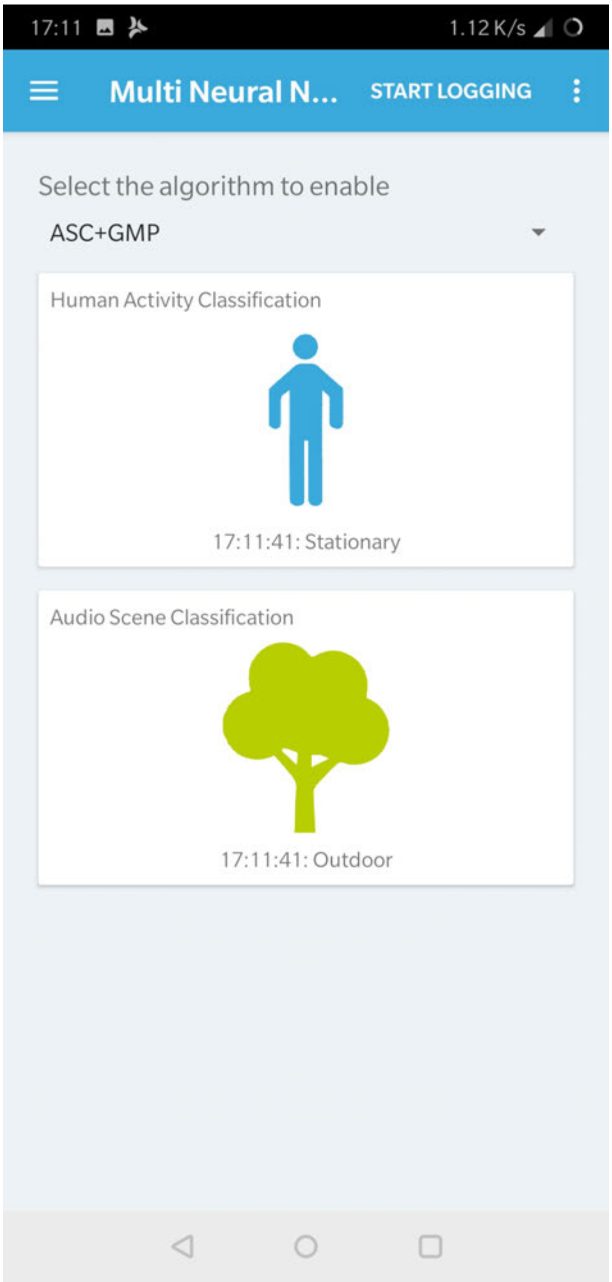
This page allows showing the execution of several AI neural networks in parallel.

You can select which networks run in concurrence.

The current firmware features 4 NNs: one ASC NN and three HAR NN.

The drop down list allows any combination of the ASC and one HAR.

Figure 24. STBLESensor (Android version) Multi Network page



Depending on the configuration selected, the following contexts are classified:

- acoustic scene
  - indoor
  - outdoor
  - driving
- human activity
  - driving
  - stationary
  - walking
  - jogging
  - biking
  - stairs

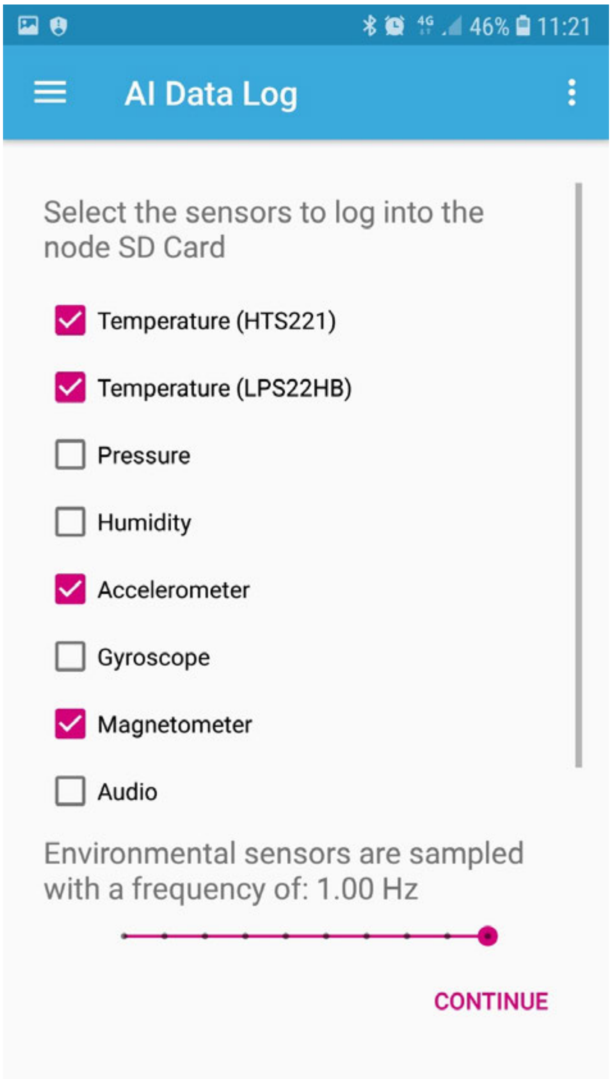
1.10.8 AI Data Logging

This page shows the AI Data Log settings available for the STEVAL-STLKT01V1, for the B-L475E-IOT01A and STEVAL-MKSBOX1V1 boards.

Sensor data can be onto the SENSING1 FAT filesystem medium, that is:

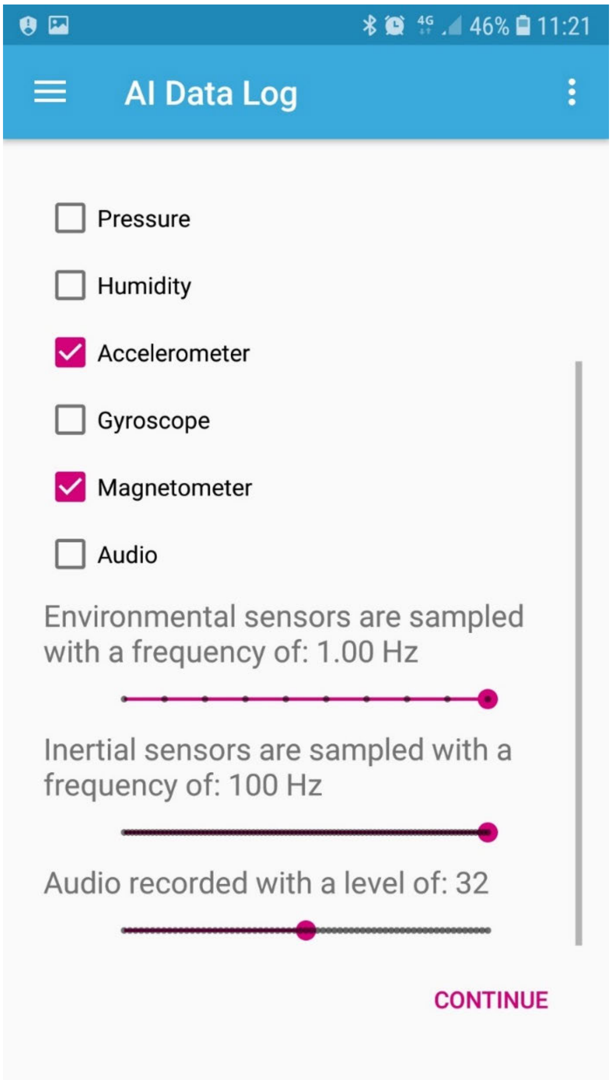
- an SD card for the STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1
- the on-board 64 Mbit QSPI Flash memory for the B-L475E-IOT01A

Figure 25. STBLESensor (Android version) AI Data Log



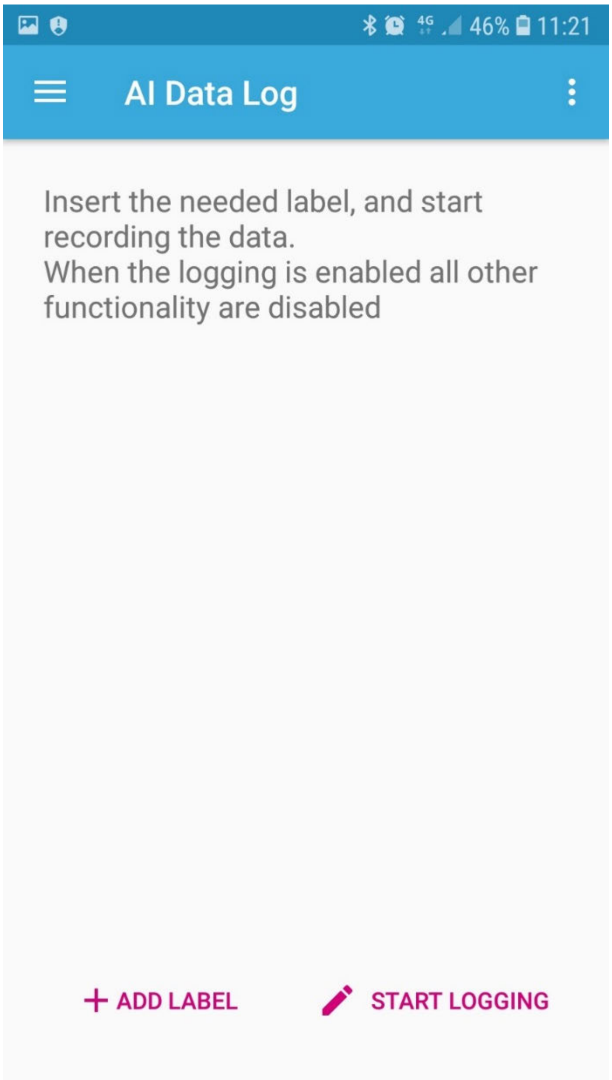
You can select any combination of sensors in the AI Data Log page that can be logged on the SD card. You can also select the sample rate for environmental sensors (temperature, pressure and humidity) and inertial sensors (accelerometer, gyroscope and magnetometer), and you can adjust the microphone sensitivity.

Figure 26. **STBLESensor (Android version) AI Data Log – sample rates**



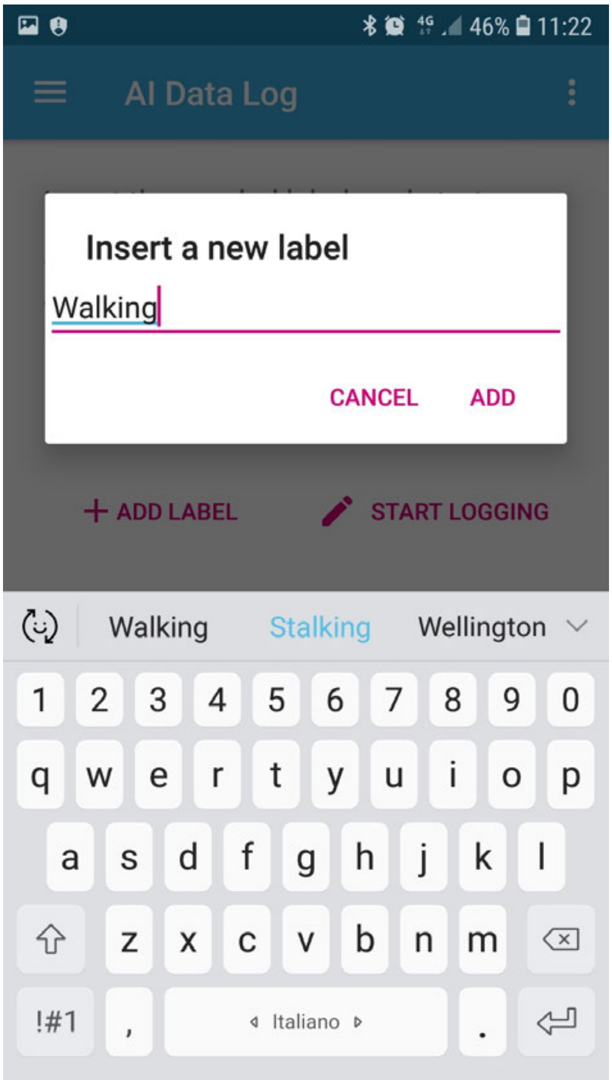
After selecting the sensors and the sample rates, you can start the data log and add a label to attach to the data.

Figure 27. STBLESensor (Android version) AI Data Log – label and start



Select [**+ADD LABEL**] to insert a new label; you can add as many labels as you like. The label remains inside the application, so you don't need to add them again before you begin logging.

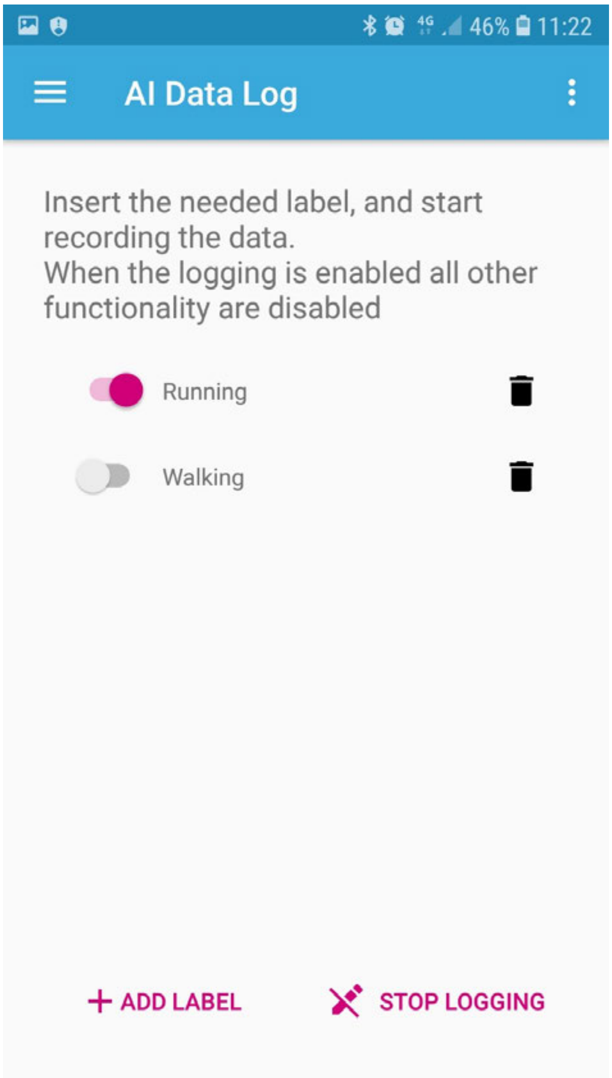
Figure 28. STBLESensor (Android version) AI Data Log – adding a label



After you have inserted a label, you can press the [**START LOGGING**] button to begin recording data. You can toggle labels while sensor data is being logged.

Press [**STOP LOGGING**] to stop recording data for that particular session.

Figure 29. STBLESensor (Android version) select a label that is logging data

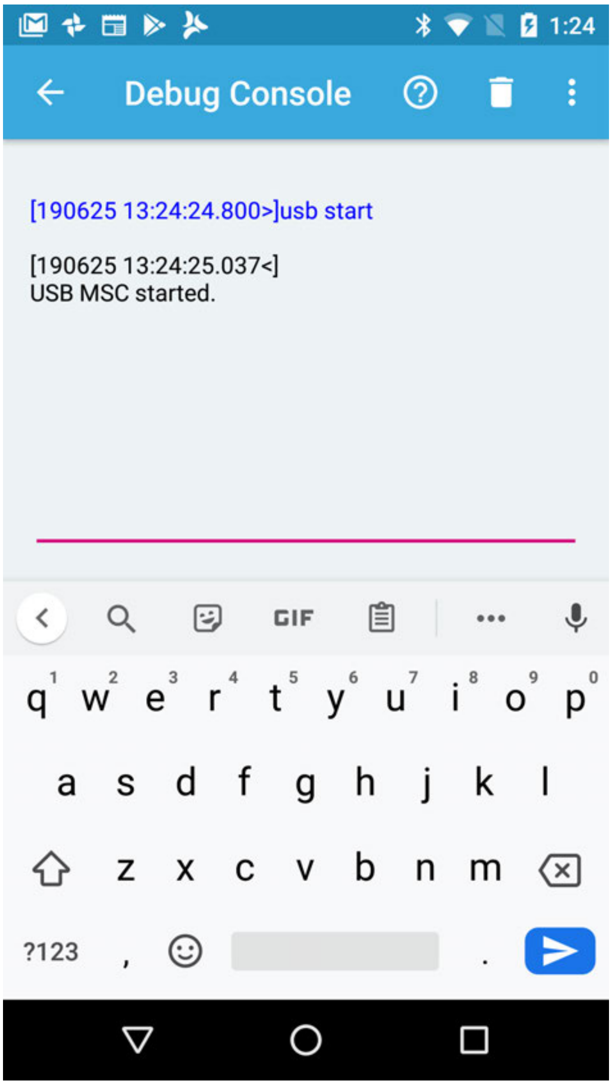


Sensor data is captured as a .csv file for environmental and motion sensors and as a .wav file for audio.

The captured data can then be copied onto a desktop for AI training.

When using the B-L475E-IOT0A board, a USB Mass Storage Device feature has been implemented to facilitate data recovery. The feature can be enabled with the Debug Console (see Figure 17. STBLESensor (Android version) Debug console (stdin/stdout/stderr)) using the usb start command. Once enabled, you can connect a standard USB micro-B cable to the USB OTG connector and the QSPI Flash memory will then appear as a new mass storage device on your computer.

Figure 30. USB Mass Storage Device Start command



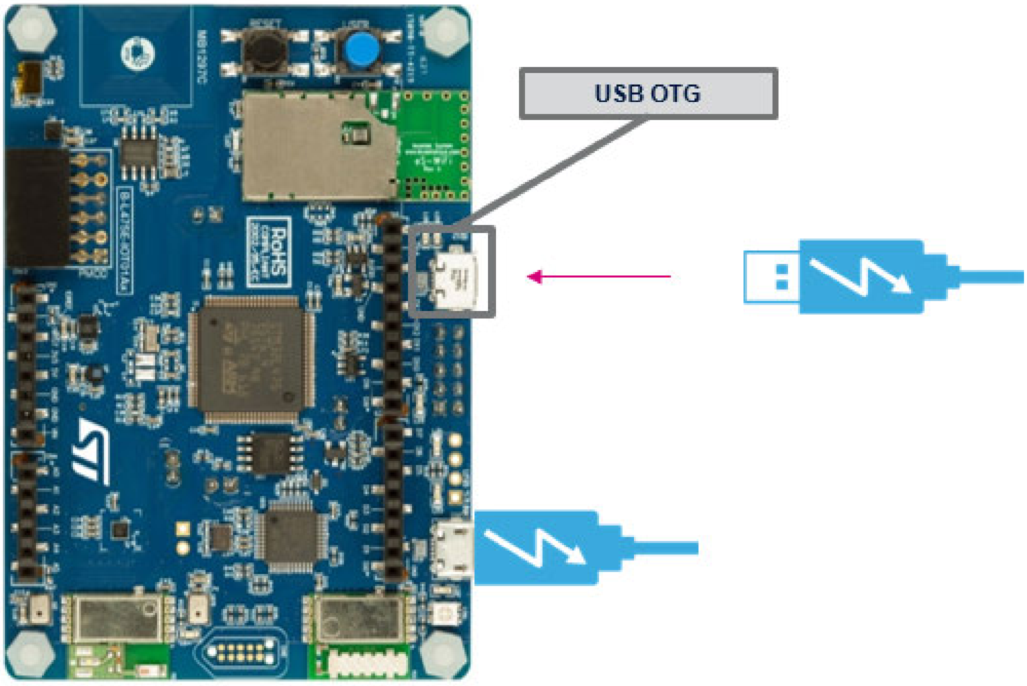


Figure 31. **STM32L4 IoT Discovery node USB Mass Storage Device connection**

The STEVAL-STLKT01V1 and STEVAL-MKSBOX1V1 have been tested to work with the following SD cards:

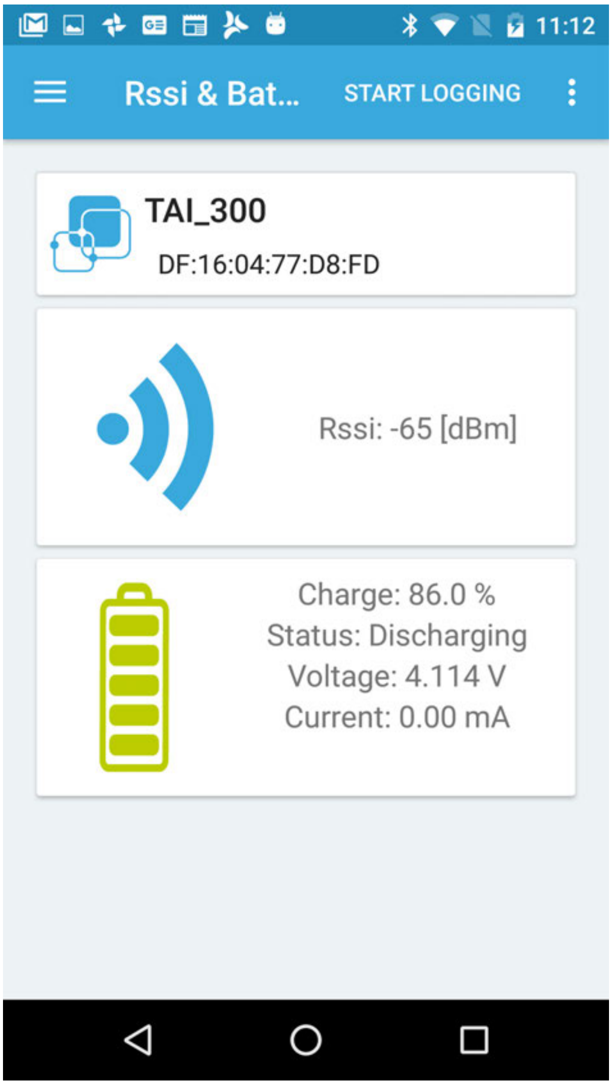
- Netac 8GB micro SD Class 6
- Netac 2GB micro SD Class 6
- Samsung 2GB
- SanDisk Ultra 32GB micro SDHC Class 10 UHS-1

1.10.9 RSSI and battery

This page shows the RSSI of the Bluetooth signal strength and charge percentage level, measured voltage and battery status (charging/discharging/low battery) when using the STEVAL-STLKT01V1, if the battery is connected.

For the STEVAL-MKSBOX1V1 evaulation board, only the measured battery voltage is displayed.

Figure 32. STBLESensor (Android version) Battery and RSSI information



The RSSI value is updated every 0.5 seconds.

1.11 Command line interface (CLI)

When enabled, the command line interface (CLI) allows the user to interact with the STM32 firmware without a BLE connection.

The CLI is designed for development purposes to quickly test functionalities or AI algorithms in a console like experience. Some of the features implemented in CLI are:

- running AI algorithms (HAR, ASC or multi)
- system information management (info, date, bdaddr, uid and sdname)
- starting an AI Datalog
- file system management (ls, cat, rm and format)
- toggling USB Mass Storage Class (for IoT only)

The CLI is implemented using the FreeRTOS+CLI framework. Command functions are defined, mapped and registered to the FreeRTOS+CLI interpreter in `cli_commands.c`.

The UARTConsoleThread manages the FreeRTOS+CLI command interpreter interface.

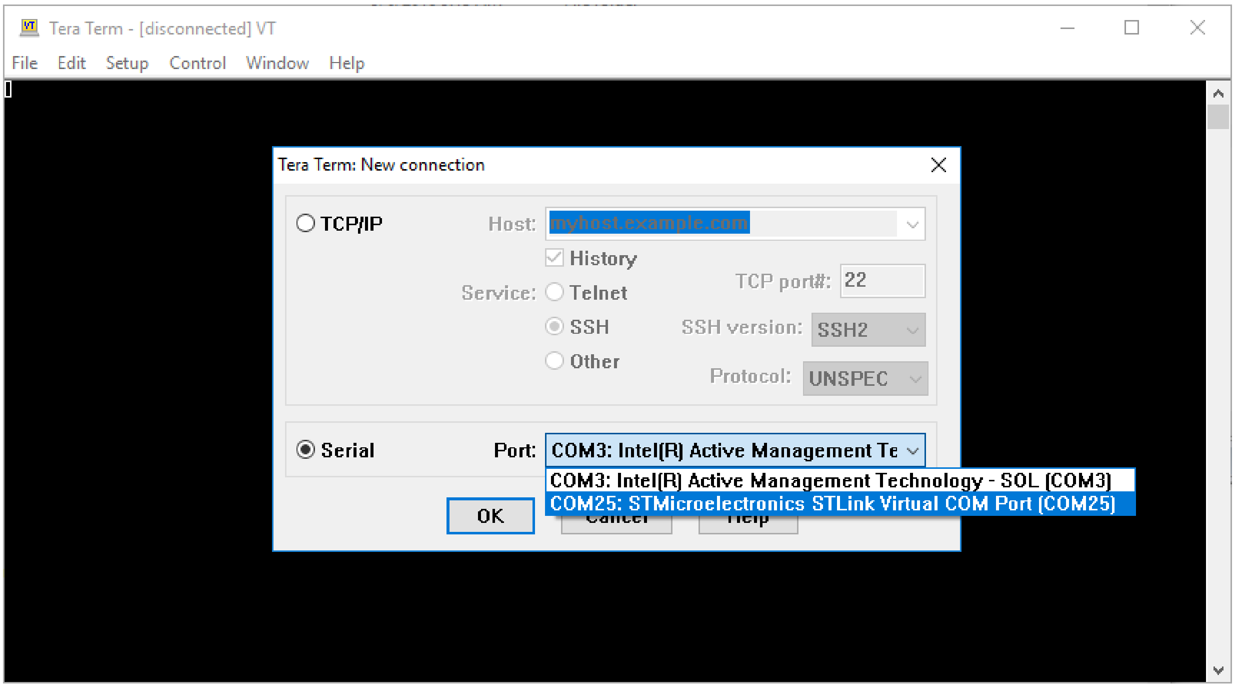
1.11.1 Configuration and setup

To enable the CLI, the firmware must be compiled with `SENSING1_USE_PRINTF` enabled (1) and `SENSING1_USE_PWR_MGNT` disabled (0). Both options are defined in `SENSING1_Config.h`.

```
/**
 * @brief Enable printf
 *        When enabled on the SensorTile and SensorTile.box, printf is
 *        redirected to USB CDC. It will introduce a delay of 10 s before
 *        starting the application for having time to open the Terminal.
 *        On other platforms, printf is redirected to a UART interface.
 */
#define SENSING1_USE_PRINTF       1

/**
 * @brief Enable printf
 *        The Datalog feature enables a local filesystem. Volume media is either
 *        an SD card for the SensorTile and SensorTile.box or the external
 *        on-board QSPI Flash memory for the IoT node.
 */
#define SENSING1_USE_DATALOG      1

/**
 * @brief Enable Power Management
 */
#define SENSING1_USE_PWR_MGNT     0
```

For boards with a UART interface (STM32 Nucleo and IoT node platforms), the CLI is implemented using the UART interface. The serial terminal application has to be connected to the STMicroelectronics STLink Virtual COM Port as shown below.

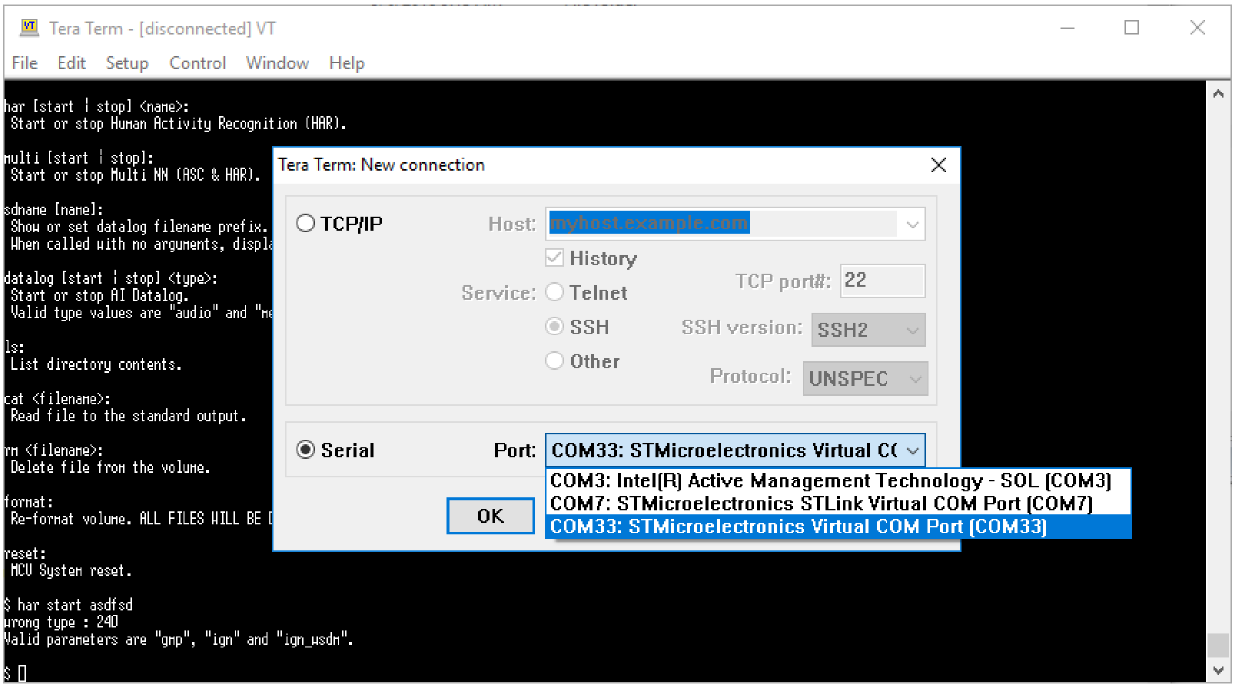
Figure 33. **STMicroelectronics STLink Virtual COM Port**



On other boards with no UART interface (SensorTile and SensorTile.box), the CLI is implemented using a USB Communications Device Class (CDC) interface. In this case, it is required to connected an additional USB cable (USB OTG connector).

The serial terminal application has to be connected to the STMicroelectronics Virtual COM Port.

Figure 34. STMicroelectronics STLink Virtual COM Port new connection (COM33)



To set up the CLI, the serial terminal application (e.g. Tera Term) must be configured as shown below (COM port number may differ).

Figure 35. Serial port setup

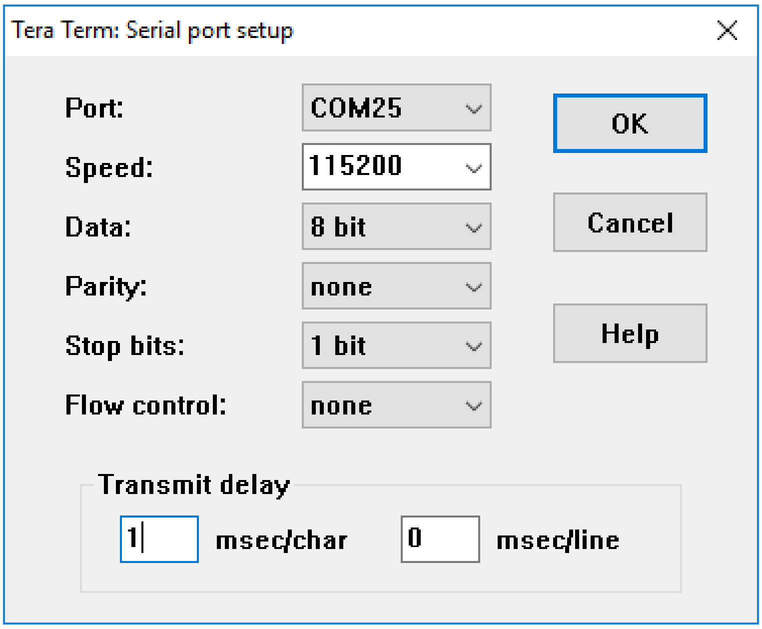
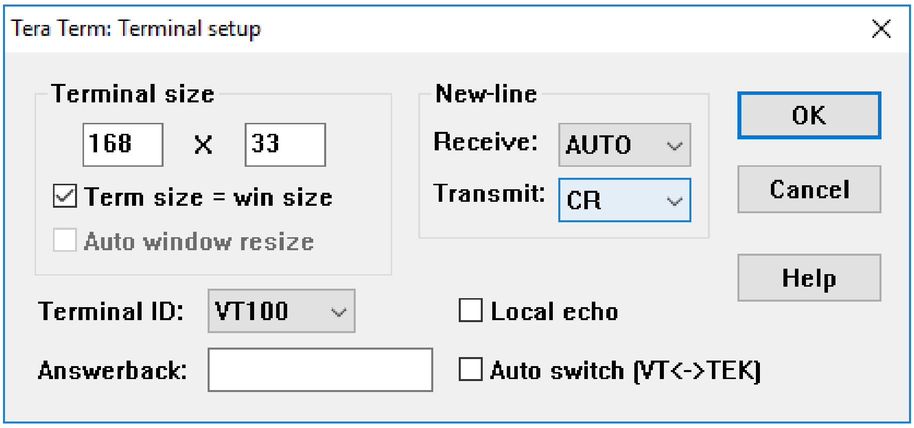


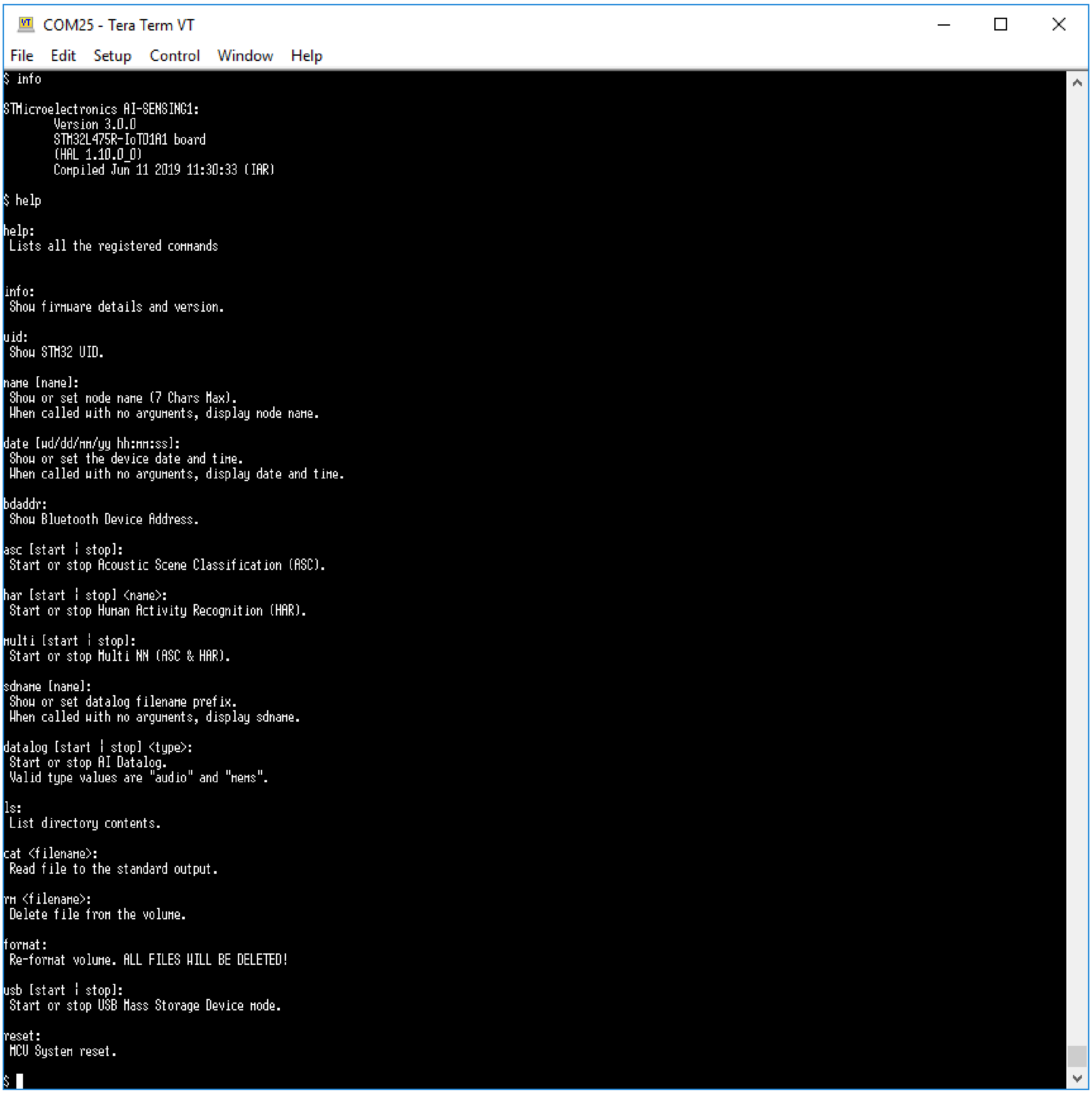
Figure 36. Terminal setup



1.11.2 CLI usage

After the FP-AI-SENSING1 firmware initialization, the CLI can be accessed using the serial command application. A prompt symbol (i.e. $) is displayed. Commands can then be typed in and executed when the return key is pressed.

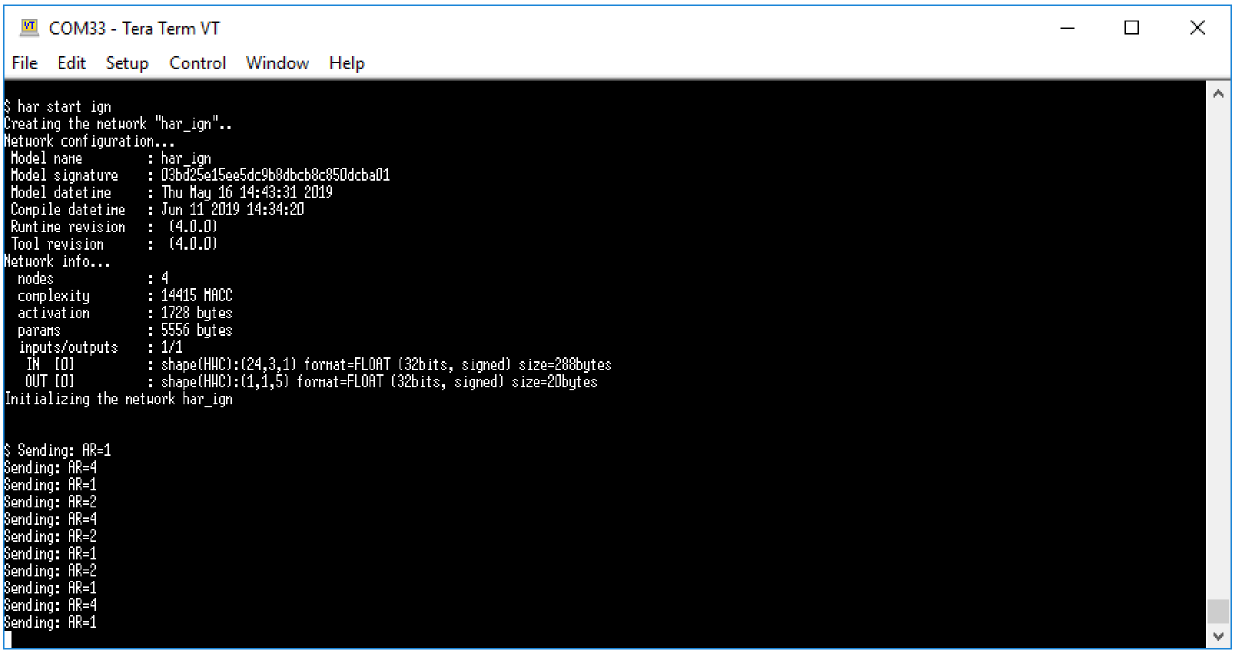
Figure 37. CLI usage example with *info* and *help* commands



1.11.2.1 *Usage example 1: selecting and running the HAR GMP NN model*

To run a specific human activity recognition (HAR) model (e.g., GMP), the `$ har start gmp` command has to be used as shown below.

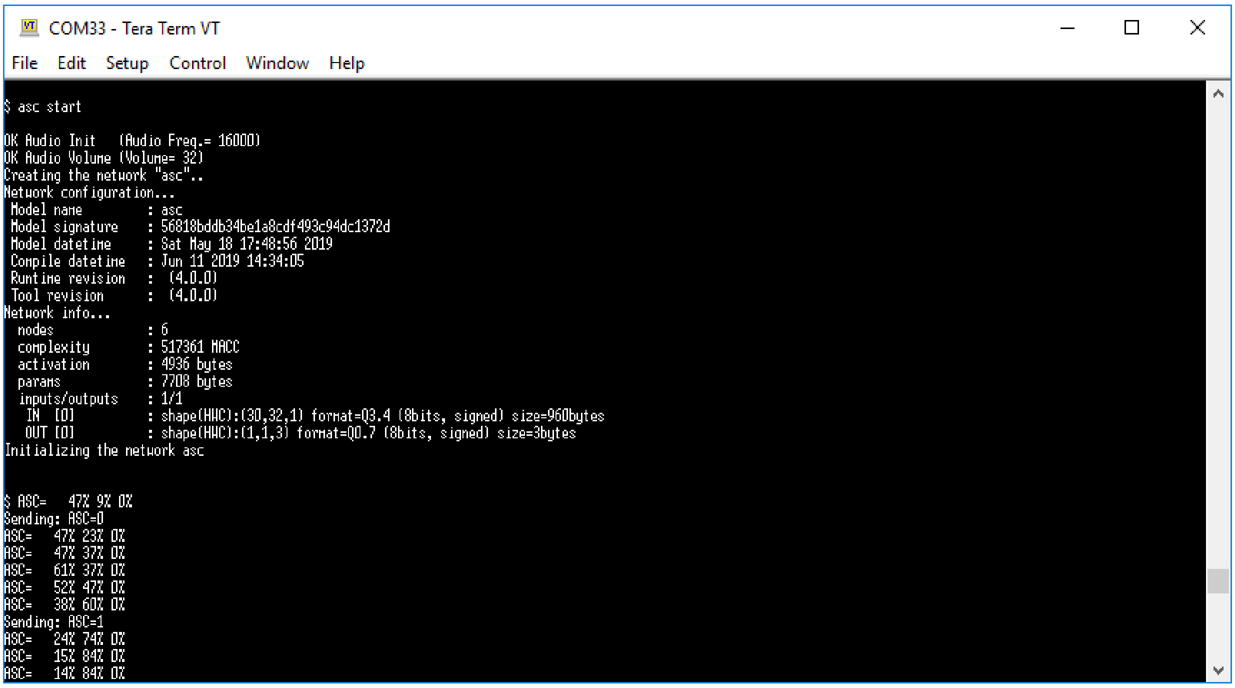
Figure 38. **CLI usage example 1**



1.11.2.2 *Usage example 2: running the ASC NN model*

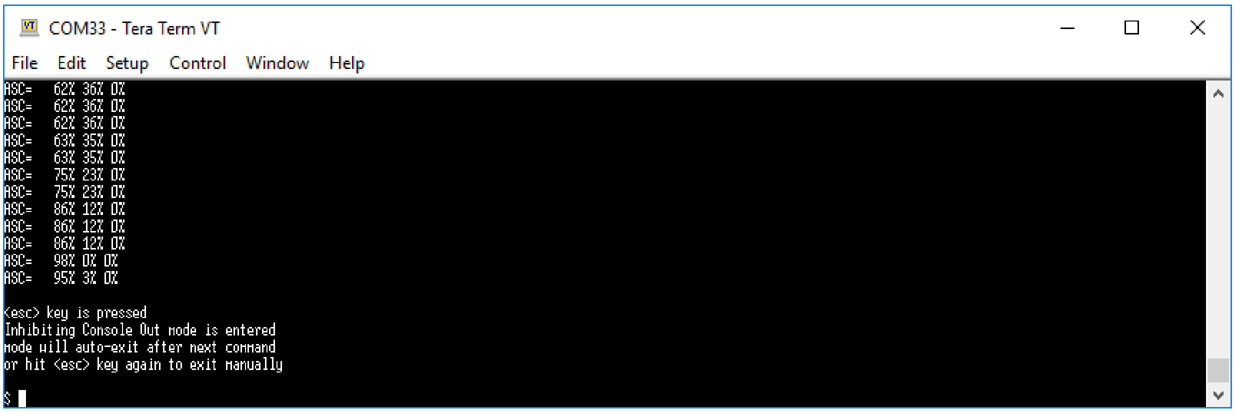
The acoustic scene classification (ASC) process can be started using the `$ asc start` command as shown below.

Figure 39. **CLI usage example 2**



*Note: To facilitate the typing of new commands when a long process is running, the console output can be temporarily disabled by pressing [**ESC**] (see the figure below). The output can be resumed by pressing [**ESC**] a second time.*

Figure 40. **Console output disabled by pressing [ESC]**



1.11.2.3 *Usage example 3: running the the HAR and ASC models in parallel*

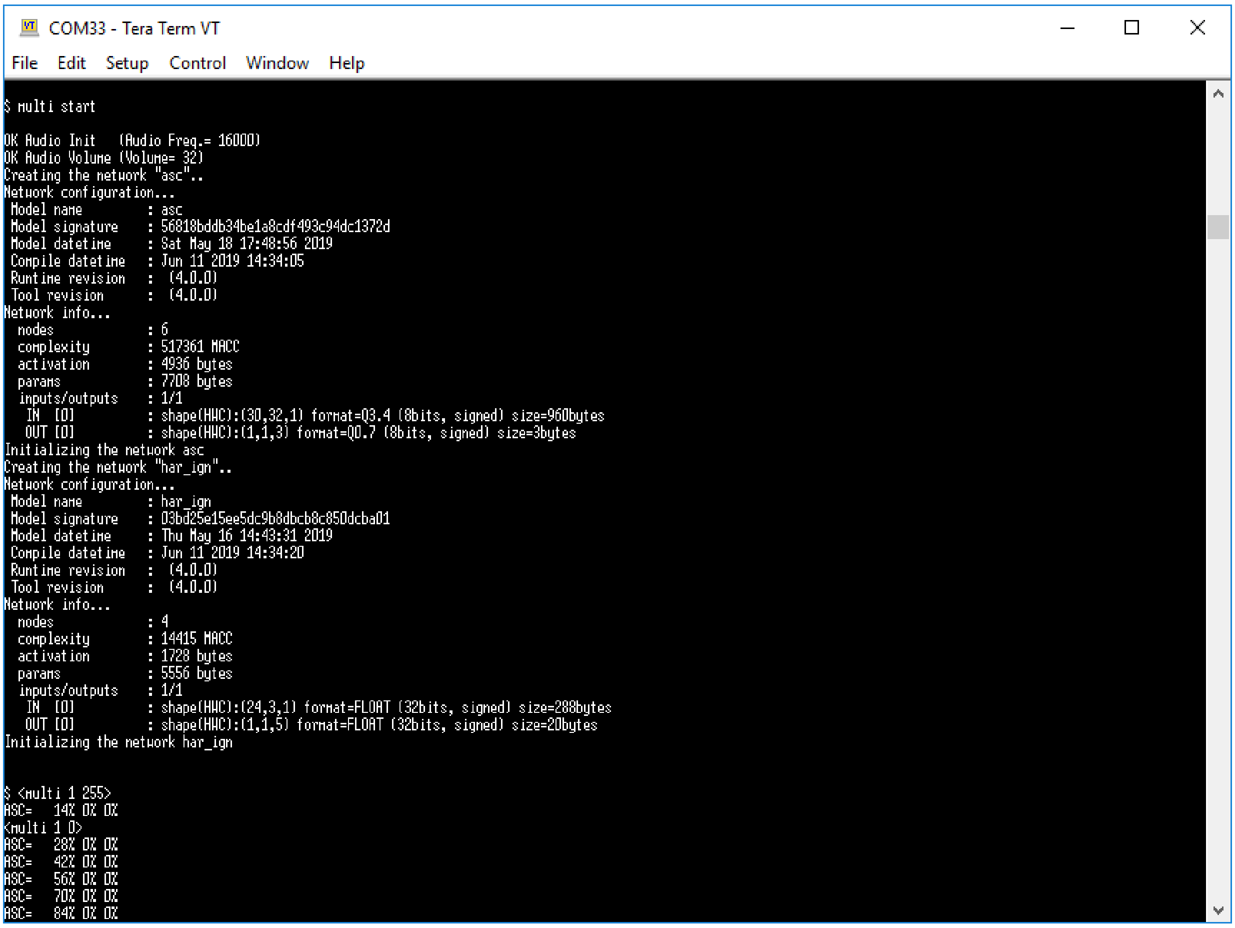
The *multi* command can be used to run the HAR and ASC algorithms at the same time.

Multi NN can be started using the `$ multi start` command.

The multi NN output is displayed each time there is an output classification change in one of the networks.

Results are display as `<multi x y>`, where x is the HAR argmax and y is the ASC argmax.

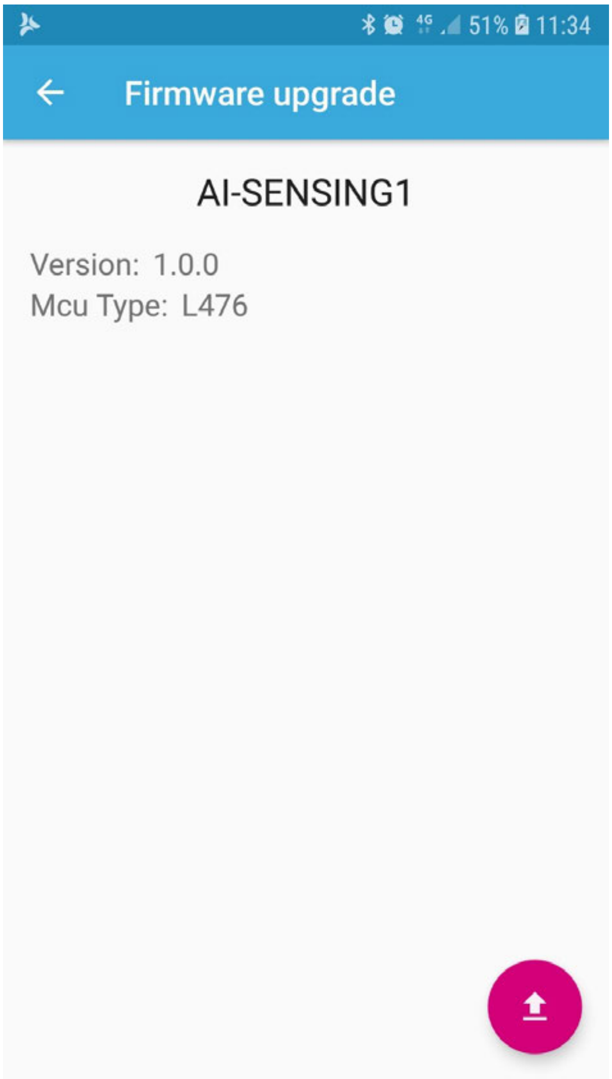
Figure 41. **CLI usage example 3**



1.12 Firmware-Over-The-Air update with STBLESensor

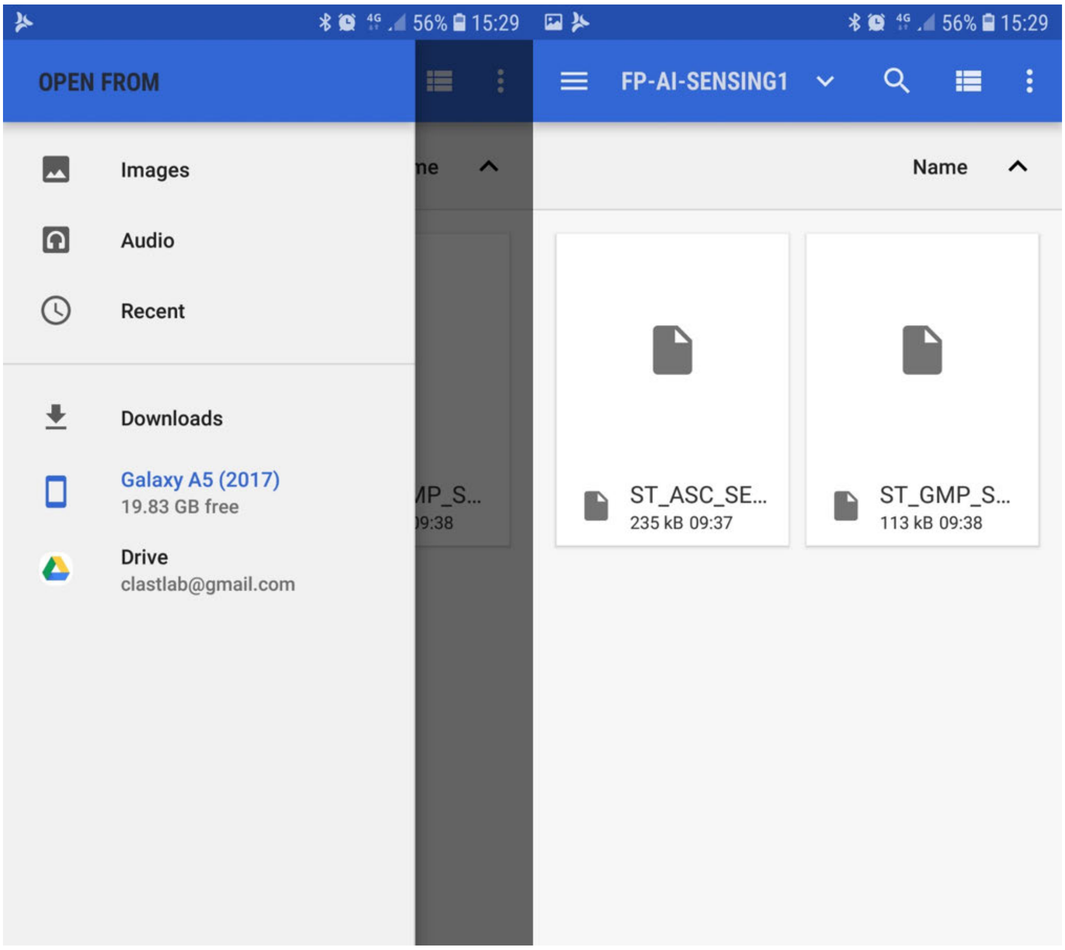
If the [**Firmware upgrade**] menu option is selected, the following page appears.

Figure 42. **STBLESensor (Android version) firmware upgrade page**



The STBLESensor application shows which FP-AI-SENSING1 software version is running and the board type. To apply an update, press the red button and choose the appropriate file.

Figure 43. STBLESensor (Android version) firmware update file selection

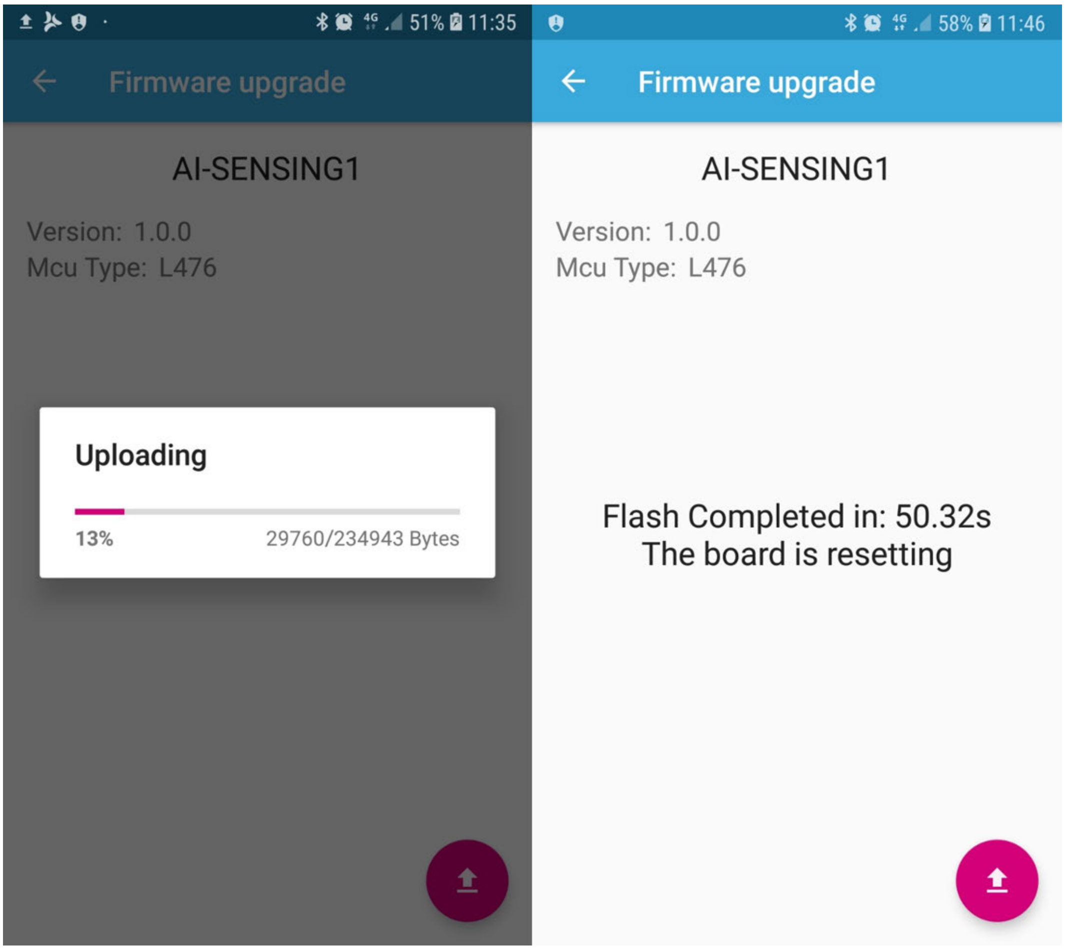


The STBLESensor app sends the FP-AI-SENSING1 software a command communicating that it is going to send an update of a certain byte size and corresponding CRC value.

If you use a UART connection and open a terminal window to monitor the behaviour of the FP-AI-SENSING1 software, you can view the debug information returned during FOTA.

The STBLESensor displays a progress bar during the FOTA procedure and the total upload time on completion.

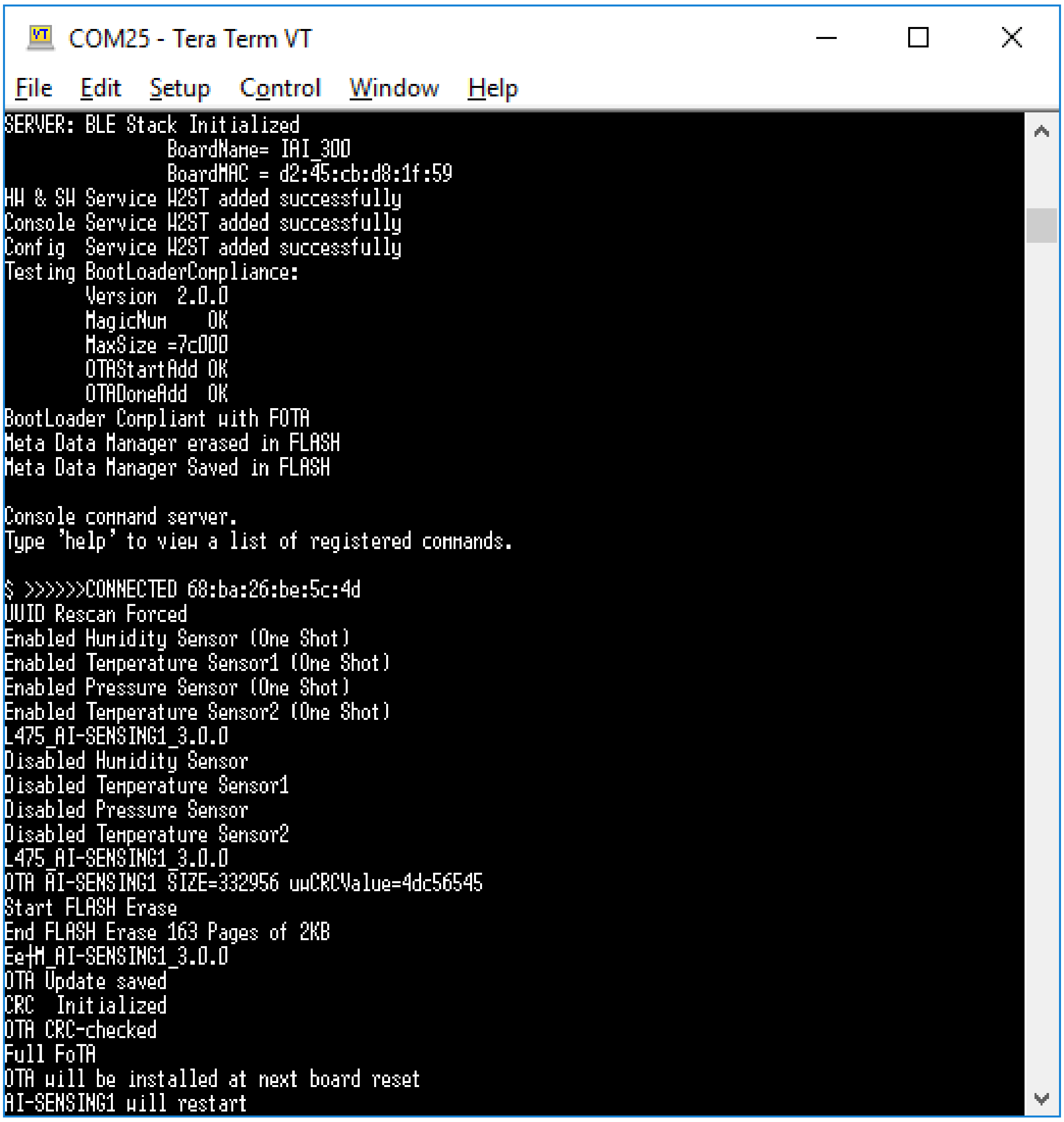
Figure 44. **STBLESensor (Android application) feedback during and after FOTA transmission**



On completion of FOTA transmission, the STM32 uses the CRC hardware unit to calculate the CRC value for the FOTA received. If this CRC matches the expected CRC previously sent by the STBLESensor application, the FP-AI-SENSING1 software writes a code number to signal the BootLoader that an OTA is ready to be applied.

The BootLoader applies the OTA at the next board reboot and executes the new FP-AI-SENSING1 firmware.

Figure 45. STBLESensor (Android application) UART console after FOTA upload, before installation



1.13 Generation of AI libraries with X-CUBE-AI

X-CUBE-AI is an artificial intelligence software package extension for STM32CubeMX. It provides an automatic and advanced neural network mapping tool to generate an optimized and robust implementation of a pre-trained neural network (also called pre-trained model). Supported deep learning frameworks (also called DL tool-box) and layers are documented in the extension data brief.

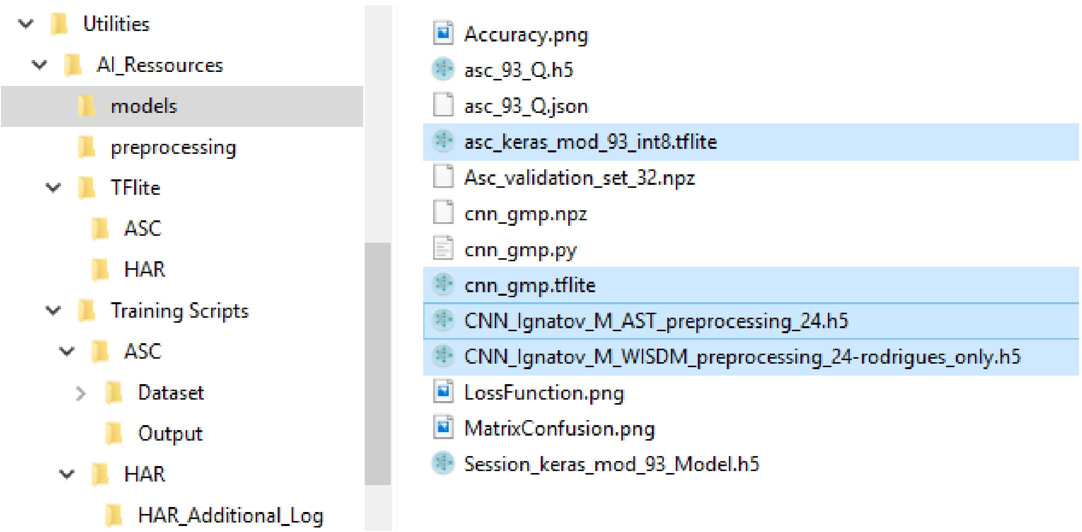
FP-AI-SENSING1 contains four sample AI designs generated using STM32CubeMX with the X-CUBE-AI plug-in:

- HAR_GMP
- HAR_IGN
- HAR_IGN_WSDM
- ASC

For HAR_IGN_WSDM, a compression factor of 4 was applied.

The pre-trained models can be found in the model folders.

Figure 46. Pre-trained models



The Keras Deep Learning Toolbox was used to generated all models. The model for HAR_GMP has been converted to TFLite (float) and ASC has been converted to TFlite (int8) as described in Section 1.9.3 Acoustic scene classification (ASC).

The conversion scripts are in the `Utilities\AI_Ressources\TFlite` folder.

You can find a Python script used to preprocess sensor data for HAR Deep Learning process:

```
Utilities\AI_Ressources\preprocessing\har_Preprocessing.py
```

Which is functionally equivalent to c code used at runtime, located in the following location:

```
Projects\[BOARD NAME]\Applications\SENSING1\Src\har_Preprocessing.c
```

Similarly, you can find a Python script used to preprocess audio signals for ASC Deep Learning process:

```
Utilities\AI_Ressources\preprocessing\asc_Preprocessing.py
```

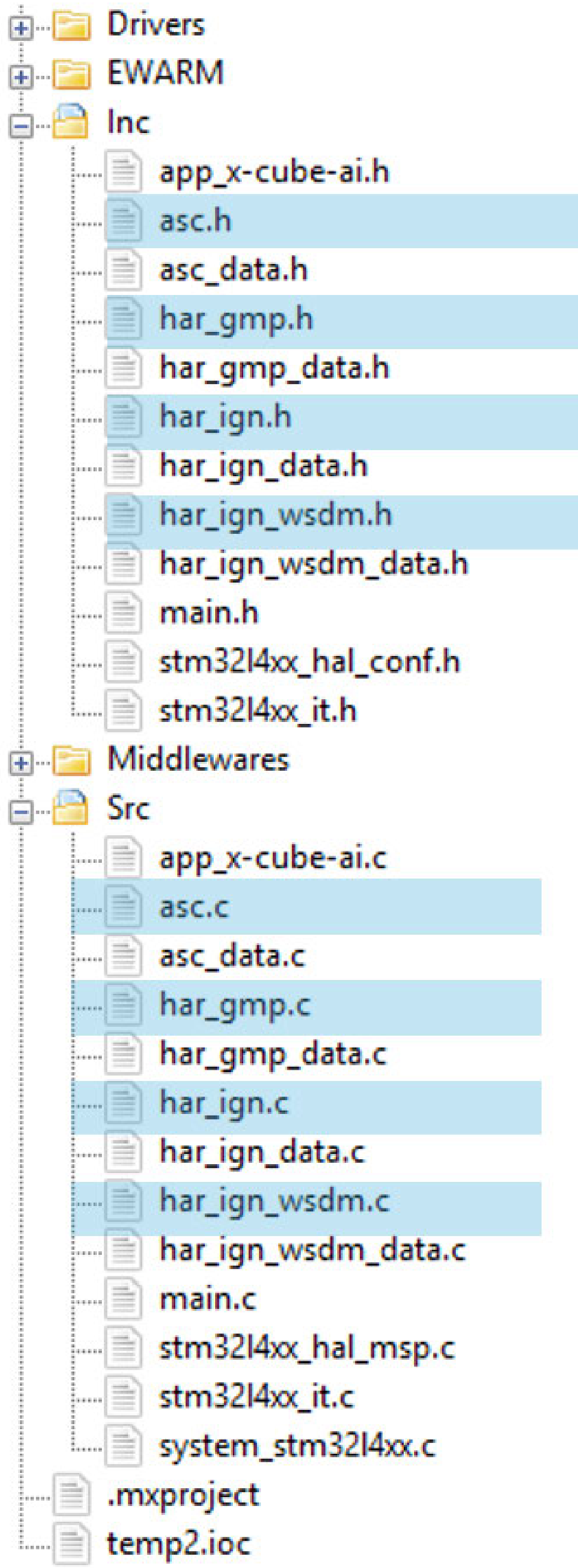
Which is functionally equivalent to c code used at runtime, located in the following location:

```
Projects\[BOARD NAME]\Applications\SENSING1\Src\asc_preprocessing.c
```

The AI library is generated for a specific IDE project (IAR, SW4STM32 or KEIL). As the generic part (\Firmware \Middlewares\ST\STM32_AI_Library\lib) is already located in the Function Pack, only the network-specific part needs to be copied:

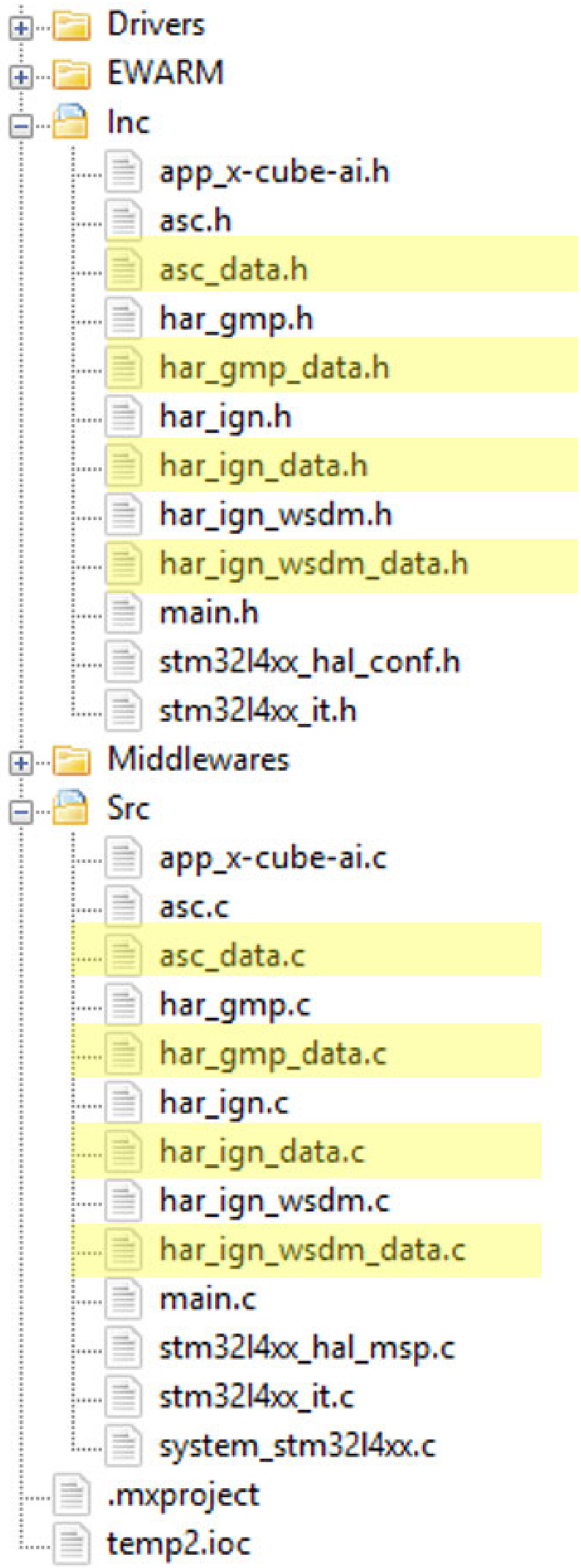
You can find the neural network source code in the following folders:

Figure 47. Neural network file locations in include and src folders



You can find the neural network weights in the following locations:

Figure 48. Neural network weights



You need to copy generated files and overwrite existing files:

- <network_name>.h and <network_name>_data.h from Inc\ to Projects\[BOARD NAME]\Applications\SENSING1\Inc\
- <network_name>.c and <network_name>_data.c from Src\ to Projects\[BOARD NAME]\Applications\SENSING1\Src\

Note: There is no need to update the STM32_AI_Library middleware.

1.14 Power profiling

Note: No specific power profiling for the ASC use case has been carried out. The figures are thus given for your reference only.

Figure 49. HAR GMP power profiling measured on sensor tile

Input voltage = 1.8 V
Average = 1,026 µA

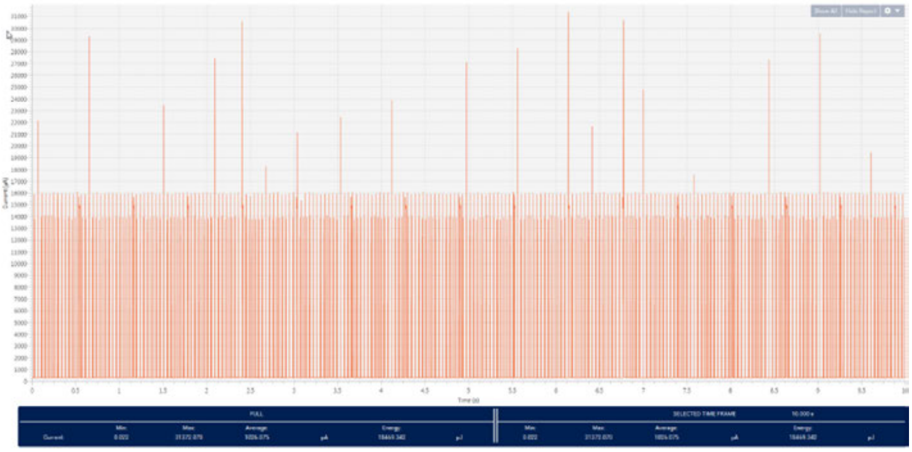
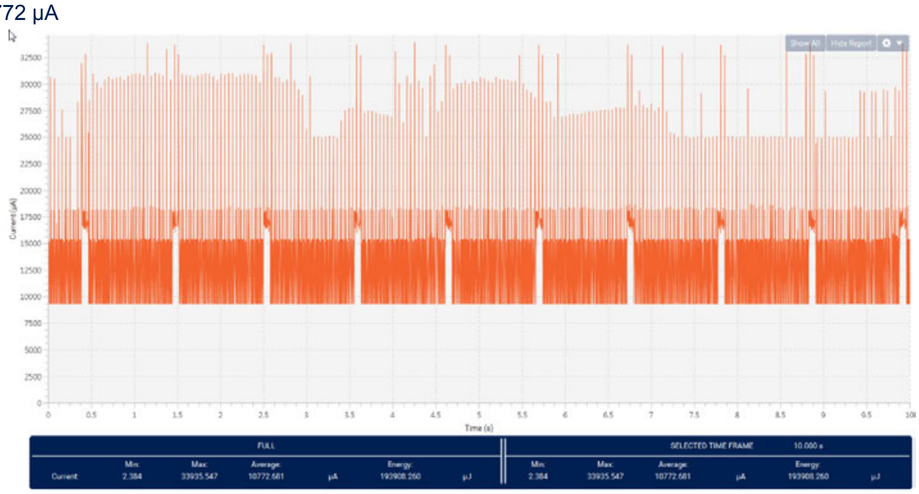


Figure 50. ASC power profiling measured on sensor tile

Input voltage = 1.8 V
Average = 10,772 µA



RELATED LINKS

AN5195: Power profiling of the FP-SNS-ALLMEMS2 function pack

2 System setup guide

2.1 Hardware description

2.1.1 STM32 Nucleo

STM32 Nucleo development boards provide an affordable and flexible way for users to test solutions and build prototypes with any STM32 microcontroller line.

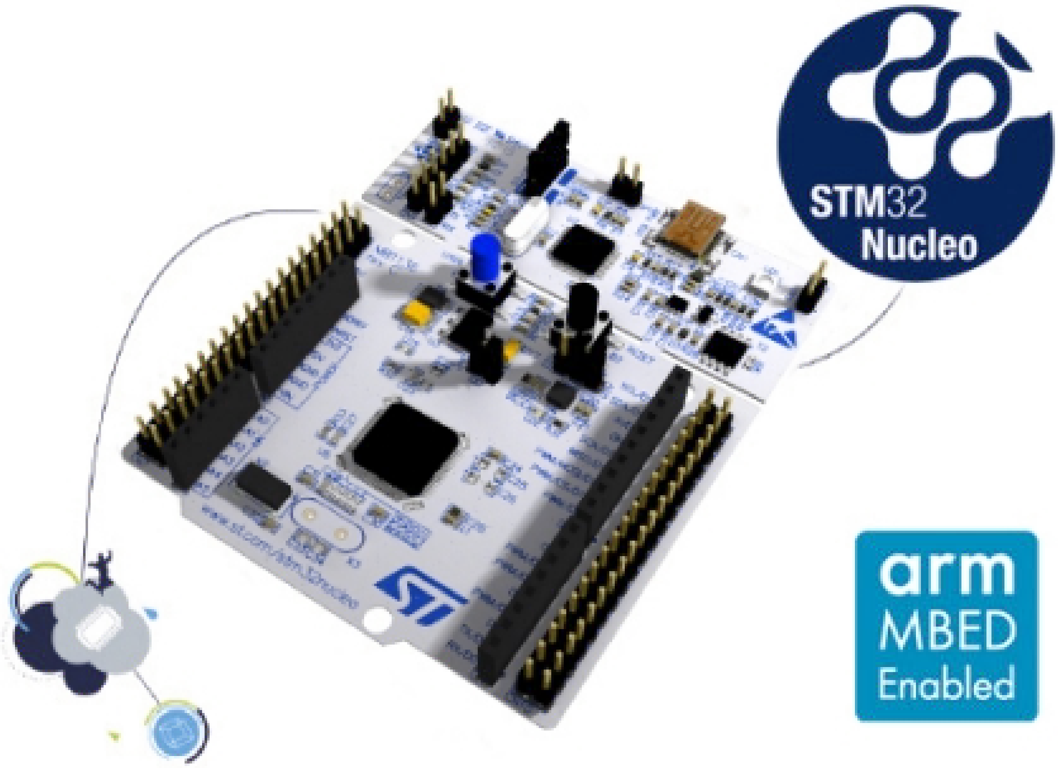
The Arduino™ connectivity support and ST morpho connectors make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide range of specialized expansion boards to choose from.

The STM32 Nucleo board does not require separate probes as it integrates the ST-LINK/V2-1 debugger/ programmer.

The STM32 Nucleo board comes with the comprehensive STM32 software HAL library together with various packaged software examples for different IDEs (IAR EWARM, Keil MDK-ARM, STM32CubeIDE, mbed and GCC/ LLVM).

All STM32 Nucleo users have free access to the mbed online resources (compiler, C/C++ SDK and developer community) at www.mbed.org to easily build complete applications.

Figure 51. STM32 Nucleo board



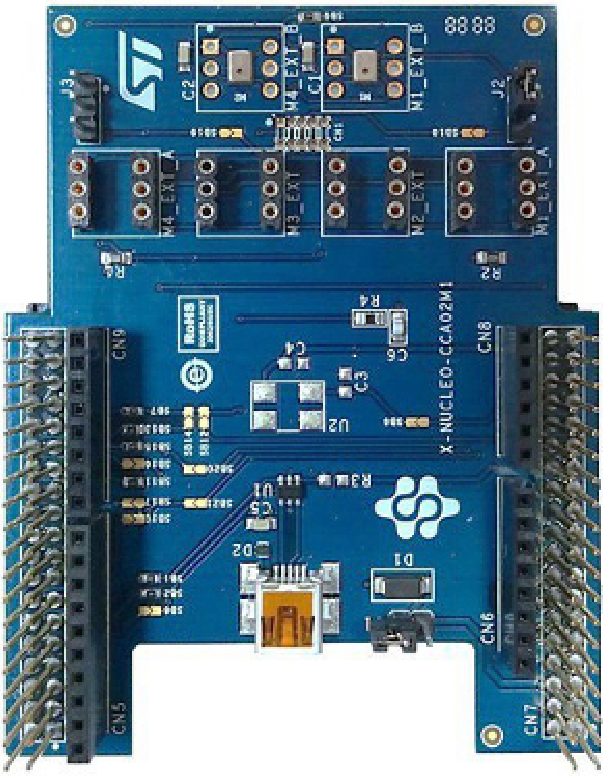
Information regarding the STM32 Nucleo board is available at www.st.com/stm32nucleo

2.1.2 X-NUCLEO-CCA02M1 expansion board

The X-NUCLEO-CCA02M1 is an expansion board based on digital MEMS microphones. It is compatible with the morpho connector layout, and is designed around STMicroelectronics MP34DT01-M digital microphones. There are two microphones soldered onto the board and you can plug in additional microphones using MP32DT01 (or MP34DT01-M) based coupon evaluation board STEVAL-MKI129V3 (or STEVAL-MKI155V3).

The X-NUCLEO-CCA02M1 allows the acquisition of up to two microphones using the I²S bus and up to four coupon microphones using I²S and SPI together. In addition, it offers a USB output for the STM32 Nucleo board. It represents a fast and easy solution for the development of microphone-based applications as well as a starting point for audio algorithm implementations.

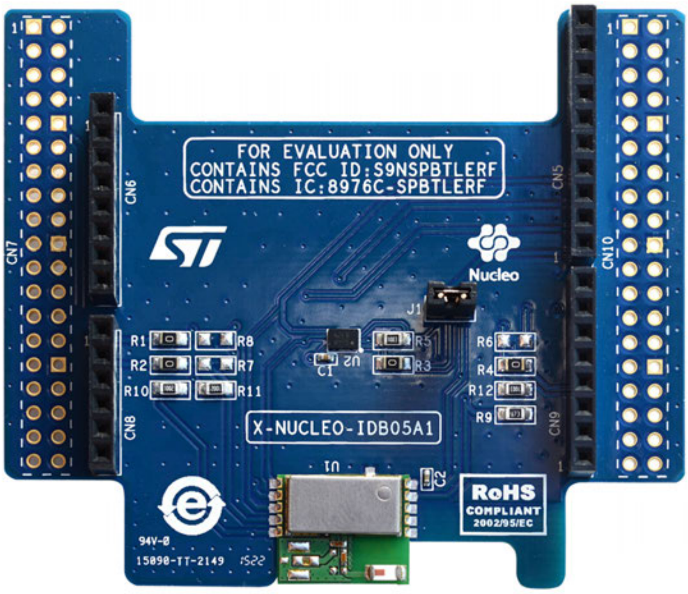
Figure 52. **X-NUCLEO-CCA02M1 expansion board**



2.1.3 X-NUCLEO-IDB05A1 expansion board

The X-NUCLEO-IDB05A1 is a Bluetooth low energy expansion board based on the SPBTLE-RF RF module, built around the BlueNRG-MS network processor, to allow expansion of the STM32 Nucleo boards. The SPBTLE-RF module is FCC (FCC ID: S9NSPBTLERF) and IC certified (IC: 8976C-SPBTLERF). The BlueNRG-MS is a very low power Bluetooth low energy (BLE) single-mode network processor, compliant with Bluetooth specification v4.2. X-NUCLEO-IDB05A1 is compatible with the ST morpho and Arduino™ UNO R3 connector layout. This expansion board can be plugged into the Arduino UNO R3 connectors of any STM32 Nucleo board.

Figure 53. **X-NUCLEO-IDB05A1 expansion board**



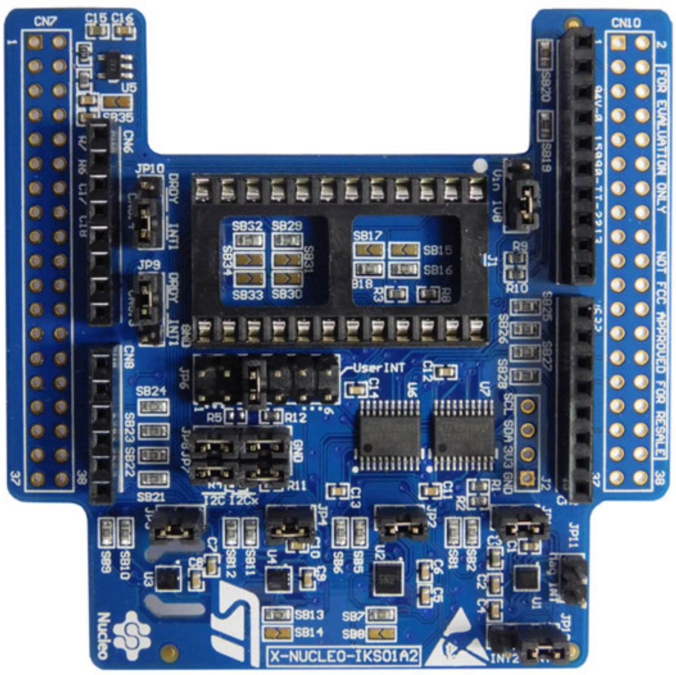
2.1.4 X-NUCLEO-IKS01A2 expansion board

The X-NUCLEO-IKS01A2 is a motion MEMS and environmental sensor expansion board for STM32 Nucleo.

It is compatible with the Arduino UNO R3 connector layout, and is designed around the LSM6DSL 3D accelerometer and 3D gyroscope, the LSM303AGR 3D accelerometer and 3D magnetometer, the HTS221 humidity and temperature sensor and the LPS22HB pressure sensor.

The X-NUCLEO-IKS01A2 interfaces with the STM32 microcontroller via the I²C pin, and it is possible to change the default I²C port.

Figure 54. **X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board**



2.1.5 STEVAL-STLKT01V1 development kit

The STEVAL-STLKT01V1 is a comprehensive development kit designed to support and expand the capabilities of the SensorTile and comes with a set of cradle boards enabling hardware scalability.

The development kit simplifies prototyping, evaluation and development of innovative solutions. It is complemented with software, firmware libraries and tools, including a dedicated mobile App.

The SensorTile is a tiny, square-shaped IoT module that packs powerful processing capabilities leveraging an 80 MHz STM32L476JGY microcontroller and Bluetooth low energy connectivity based on BlueNRG-MS network processor as well as a wide spectrum of motion and environmental MEMS sensors, including a digital microphone.

SensorTile can fit snugly in your IoT hub or sensor network node and become the core of your solution.

To upload new firmware onto the SensorTile, an external SWD debugger (not included in the kit) is needed. It is recommended to use ST-LINK/V2-1 found on any STM32 Nucleo-64 development board.

2.1.5.1 Features

- Included in the development kit package:
  - SensorTile module (STEVAL-STLCS01V1) with STM32L476JG, LSM6DSM, LSM303AGR, LPS22HB, MP34DT05-A, BlueNRG-MS, BALF-NRG-02D3 and LD39115J18R
  - SensorTile expansion Cradle board equipped with audio DAC, USB port, STM32 Nucleo, Arduino UNO R3 and SWD connector
  - SensorTile Cradle with battery charger, humidity and temperature sensor, SD memory card slot, USB port and breakaway SWD connector
  - 100 mAh Li-Ion battery
  - Plastic box
  - SWD programming cable
- Software libraries and tools
  - STSW-STLKT01: SensorTile firmware package that supports sensors raw data streaming via USB, data logging on SDCard, audio acquisition and audio streaming.
  - FP-SNS-ALLMEMS1 and FP-AI-SENSING1: STM32Cube function packs
  - STBLESensor: iOS and Android demo Apps
  - BlueST-SDK: iOS and Android Software Development Kit
- CE certified
- RoHS and China RoHS compliant
- WEEE compliant
- FCC (ID: S9NSTILE01) certified
- IC (IC: 8976C-STILE01) certified with PMN: STEVAL-STLKT01V1; HVIN: STEVAL-STLCS01V1; HMN: STEVAL-STLCX01V1; FVIN: bluenrg_7_1_e_Mode_2-32MHz-XO32K_4M.img
- TYPE certified (006-000482)

2.1.5.2 Boards included in the kit

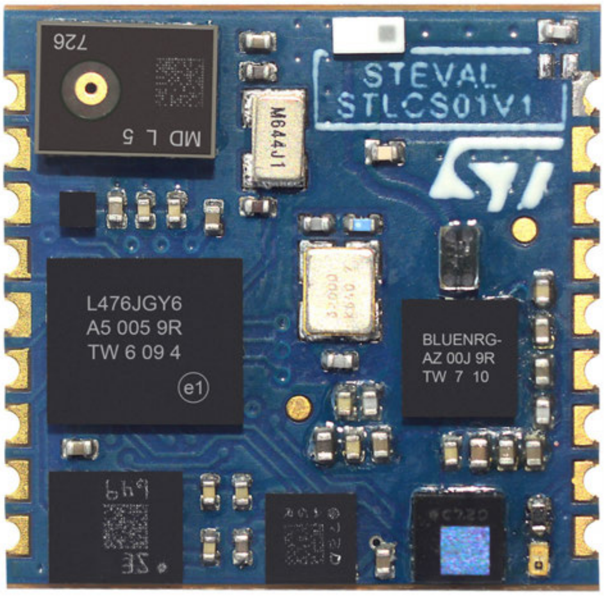


Figure 55. STLCS01V1 board photo

STLCS01V1 SensorTile component board features

- Very compact module for motion, audio, environmental sensing and Bluetooth® low energy connectivity with a complete set of firmware examples
- Mobile connectivity via the STBLESensor app, available for iOS™ and Android™
- Main components:
  - STM32L476JG – 32-bit ultra-low-power MCU with Cortex®M4F
  - LSM6DSM – iNEMO inertial module: 3D accelerometer and 3D gyroscope
  - LSM303AGR – Ultra-compact high-performance eCompass module: ultra-low power 3D accelerometer and 3D magnetometer
  - LPS22HB – MEMS nano pressure sensor: 260-1260 hPa absolute digital output barometer
  - MP34DT05-A – 64 dB SNR digital MEMS microphone
  - BlueNRG-MS – Bluetooth low energy network processor
  - BALF-NRG-02D3 – 50 Ω balun with integrated harmonics filter
  - LD39115J18R – 150 mA low quiescent current low noise LDO 1.8 V
- 2 V - 5.5 V power supply range
- External interfaces: UART, SPI, SAI (serial audio interface), I²C, DFSDM, USB OTG, ADC, GPIOs
- Pluggable or solderable interface
- SWD interface for debugging and programming capability
- CE certified
- RoHS and China RoHS compliant
- WEEE compliant
- FCC certified
- IC certified
- TYPE certified

STLCS01V1 SensorTile component board description

The STEVAL-STLCS01V1 (SensorTile) is a highly integrated reference design that can be plugged into form-factor prototypes to add sensing and connectivity capabilities to new designs through a smart hub solution. It can also easily support development of monitoring and tracking applications as standalone sensor node connected to iOS/Android smartphone applications.

The SensorTile comes in a very small square shape 13.5 x 13.5 mm. All the electronic components are on the top side of the pcb, while the bottom side has a small connector through which it is possible to easily plug and unplug it from a motherboard. The connector pinout is also replicated on 18 pcb pads that render the SensorTile a solderable system on module as well.

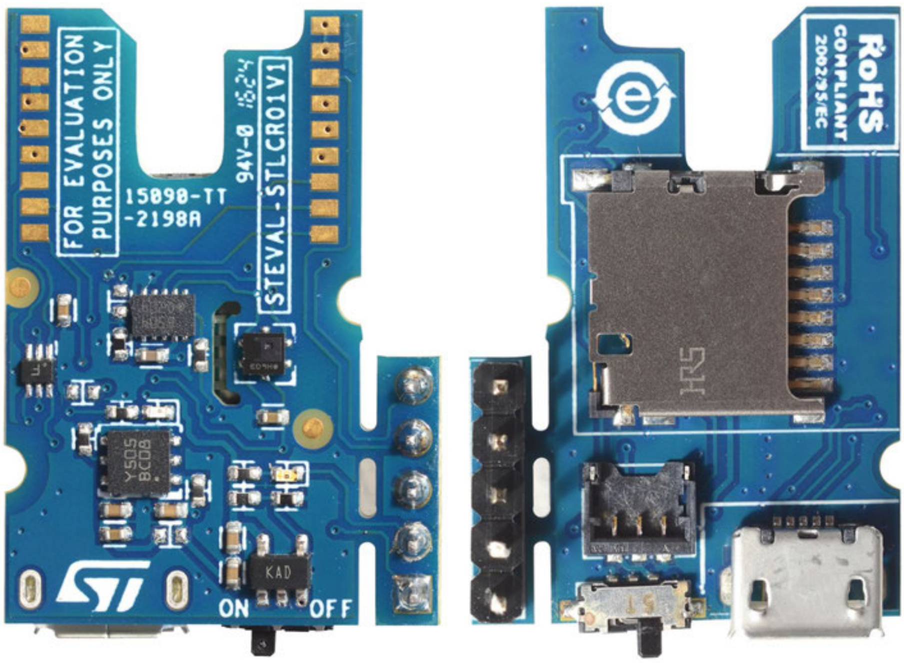
The module comes with pre-loaded FP-SNS-ALLMEMS1 (former BLUEMICROSYSTEM2) software that initializes all the sensors and the Bluetooth low energy radio. The STBLESensor app, available free of charge on the respective Google and Apple stores, is the easiest and fastest way to start using the SensorTile board and to experience a real activity monitoring system.

The SensorTile firmware package STSW-STLKT01, built on the STM32Cube software technology, includes all the low level drivers to manage the on-board devices and system-level interfaces. It has been designed in order to be easily extended and personalized as starting point for development and customization of new dedicated applications.

All the firmware packages are freely available on www.st.com.

The Bluetooth radio power output is set by default at 0 dBm. The FCC and IC certifications refer to this operating value. The power output can be changed up to 8 dBm by reprogramming the device firmware, but the change of this operating value will require an update of the FCC and IC certifications, with additional radio emission tests to be performed.

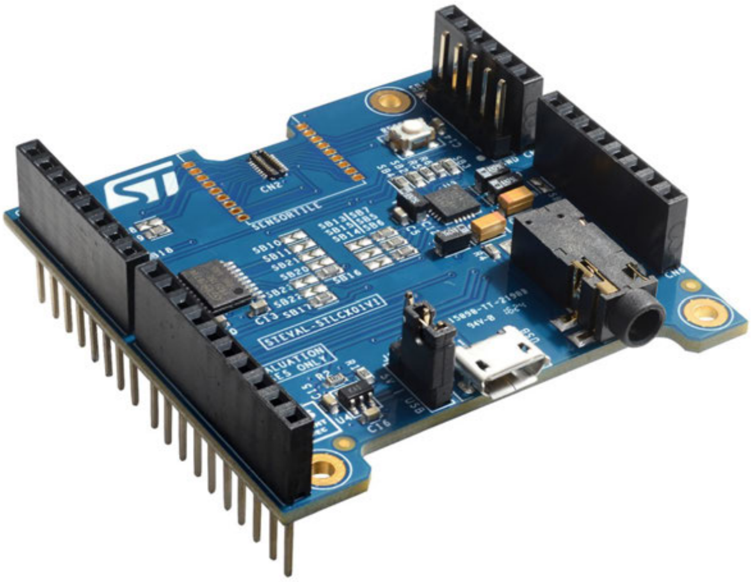
Figure 56. STLCR01V1 board photo



STLCR01V1 SensorTile component board features

- Sensortile Cradle board with SensorTile footprint (solderable)
- STBC08PMR – 800 mA standalone linear Li-Ion battery charger
- HTS221 – capacitive digital sensor for relative humidity and temperature
- LDK120M-R – 200 mA low quiescent current very low noise LDO
- STC3115 – Gas gauge IC
- USBLC6-2P6 – very low capacitance ESD protection
- USB type A to Mini-B USB connector for power supply and communication
- microSD card socket
- SWD connector for programming and debugging

Figure 57. STLCX01V1 board photo



STLCX01V1 SensorTile component board features

- Sensortile Cradle expansion board with SensorTile plug connector
- Compatible with STM32 Nucleo boards through Arduino UNO R3 connector
- LDK120M-R – 200 mA low quiescent current very low noise LDO
- ST2378ETTR – 8-bit dual supply 1.71 to 5.5 V level translator
- USBLC6-2P6 – very low capacitance ESD protection
- 16-Bit, low-power stereo audio DAC
- Micro-USB connector for power supply and communication
- Reset button
- SWD connector for programming and debugging

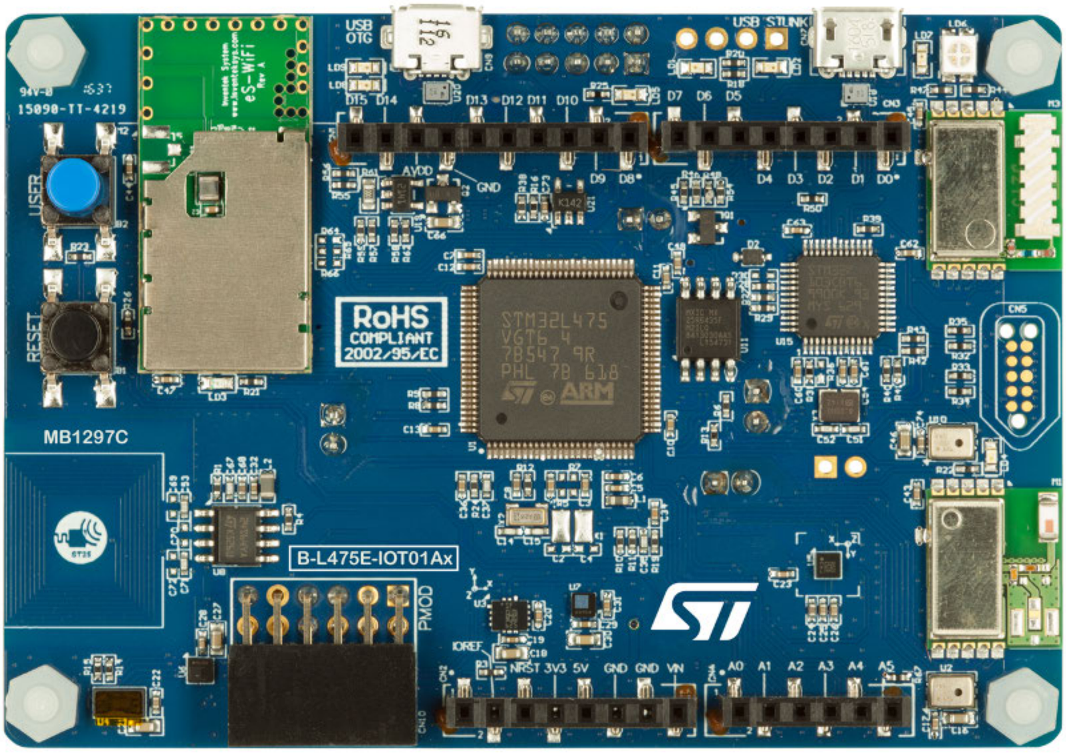
2.1.6 STM32L4 Discovery kit IoT node

The B-L475E-IOT01A Discovery kit for IoT node allows you to develop applications to directly connect to cloud servers.

The Discovery kit enables a wide variety of applications by exploiting low-power communication, multi-way sensing and ARM® Cortex® -M4 core-based STM32L4 series features.

It supports Arduino Uno R3 and PMOD connectivity providing unlimited expansion capabilities with a large choice of dedicated add-on boards.

Figure 58. B-L475E-IOT01A Discovery kit



2.1.7 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 59. 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 STM32 Nucleo equipped with the sensors, microphones and Bluetooth low energy expansion boards and for the STEVAL-STLKT01V1, B-L475E-IOT01A1 and the STEVAL-MKSBOX1V1:

- FP-AI-SENSING1: 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
  - System Workbench for STM32 + ST-LINK

2.3 Hardware and software setup

2.3.1 Hardware setup

The following hardware components are required:

- for STM32 Nucleo platforms:
  - One STM32 Nucleo board (order code NUCLEO-L476RG)
  - One microphone expansion board (order code: X-NUCLEO-CCA02M1)
  - One sensor expansion board (order code X-NUCLEO-IKS01A2)
  - One BlueNRG Bluetooth low energy expansion board (order code: X-NUCLEO-IDB05A1)
  - One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
- For the STEVAL-STLKT01V1 kits:
  - The STEVAL-STLKT01V1 development kit
  - The ST-LINK/V2-1 debugger/programmer integrated on the STM32 Nucleo board
  - One USB type A to Mini-B USB cable to connect the STM32 Nucleo to the PC
  - One USB type A to Micro-B USB cable to connect the STEVAL-STLKT01V1 to the PC
- For the B-L475E-IOT01 STM32L4 Discovery kit IoT node:
  - the B-L475E-IOT01 STM32L4 Discovery kit IoT node
  - One USB type A to Micro-B USB cable to connect the B-L475E-IOT01 to the PC
- For the STEVAL-MKSBOX1V1 evaluation board:
  - the STEVAL-MKSBOX1V1 evaluation board
  - the ST-LINK-V2 debugger/programmer
  - 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

2.3.2 Software setup

This section describes how to set up different hardware parts before writing and executing an application:

- on the STM32 Nucleo board with the expansion boards
- on the STEVAL-STLKT01V1 evaluation board
- on the B-L475E-IOT01 STM32L4 Discovery kit IoT node
- on the STEVAL-MKSBOX1V1 evaluation board

2.3.2.1 Development tool-chains and compilers

Select one of the Integrated Development Environments supported by the STM32Cube expansion software and follow the system requirements and setup information provided by the selected IDE provider.

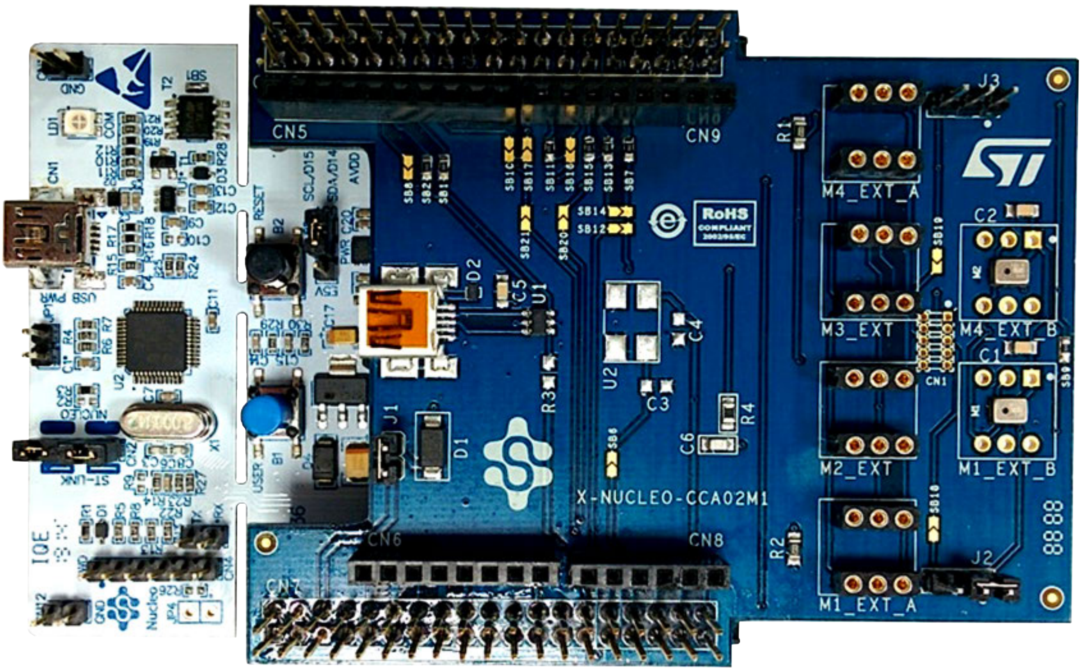
2.3.3 System setup guide

2.3.3.1 *STM32 Nucleo and expansion board setup*

The STM32 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. You can download the relevant version of the ST-LINK/V2-1 USB driver at STSW-LINK008 or STSW-LINK009.

The X-NUCLEO-CCA02M1 sensor board is easily connected to the STM32 Nucleo board through the morpho connector, as shown below.

Figure 60. STM32 Nucleo plus X-NUCLEO-CCA02M1 boards



The X-NUCLEO-IDB05A1 expansion board is connected to the X-NUCLEO-CCA02M1 board through the Arduino UNO R3 extension connector, as shown below.

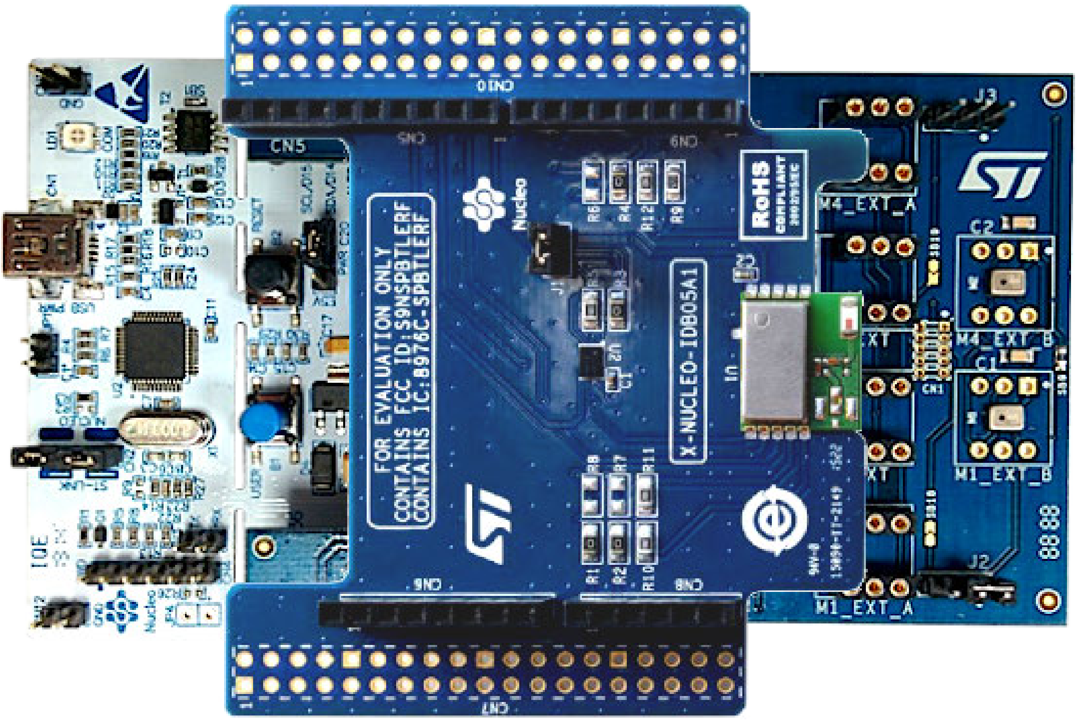
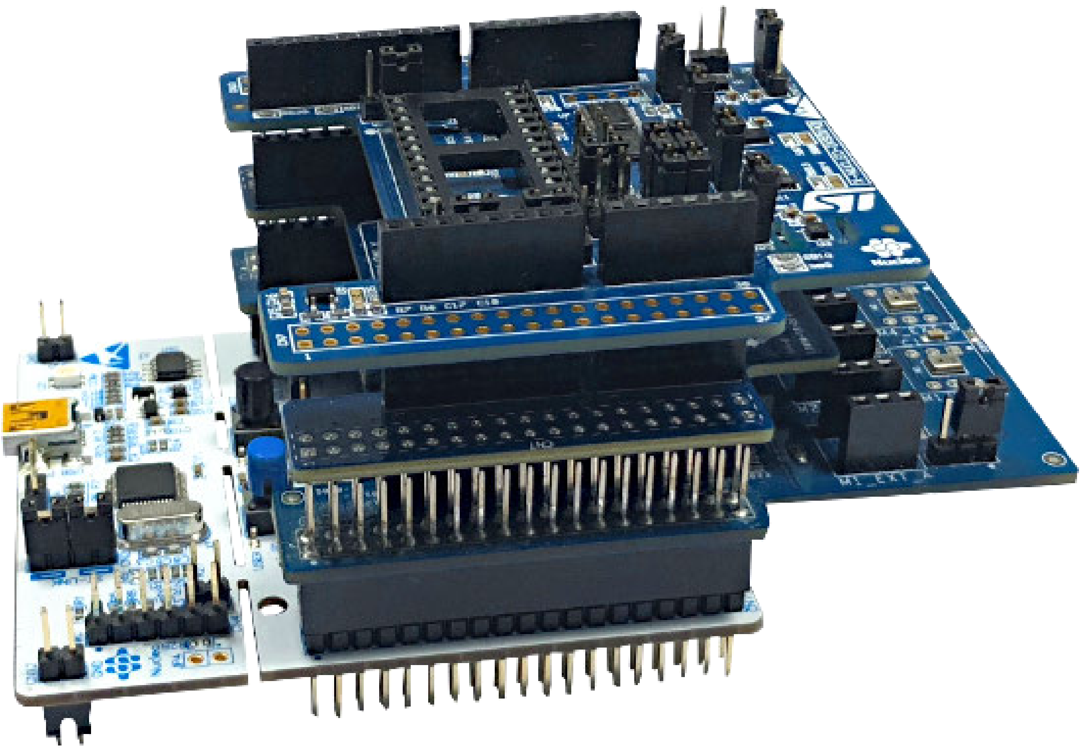


Figure 61. STM32 Nucleo plus X-NUCLEO-CCA02M1 plus X-NUCLEO-IDB05A1 boards

The X-NUCLEO-IKS01A2 or X-NUCLEO-IKS01A3 sensor board is then connected to the X-NUCLEO-IDB05A1 expansion board through the Arduino UNO R3 extension connector, as shown below.

Figure 62. STM32 Nucleo plus X-NUCLEO-CCA02M1 plus X-NUCLEO-IDB05A1 plus X-NUCLEO-IKS01A2 boards



Note: The stacking sequence indicated above helps optimize the performance of the SPBTLE-RF module on the X-NUCLEO-IDB05A1 expansion board, and reduce interference from its antenna.

2.3.3.2 STEVAL-STLKT01V1 setup

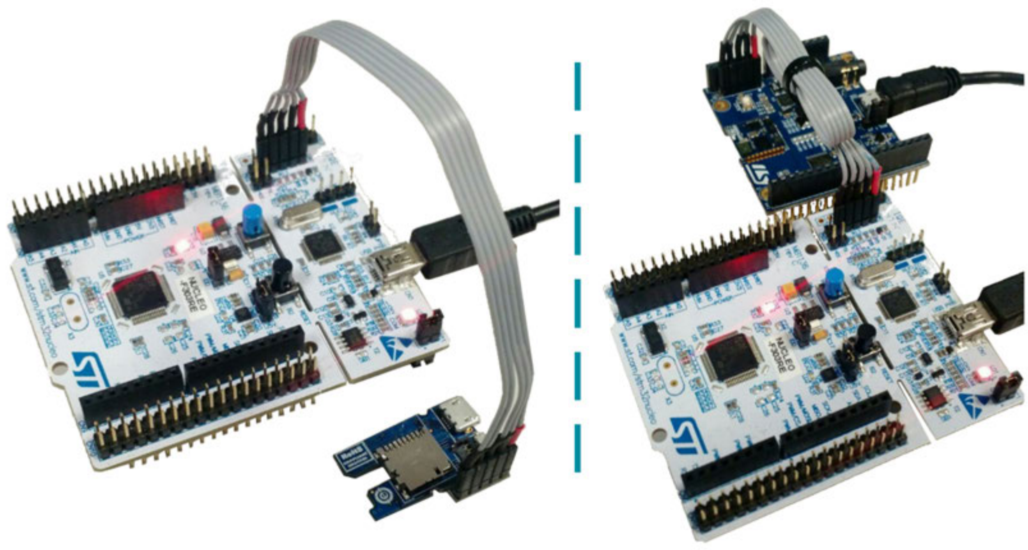
The ST-LINK/V2-1 debugger/programmer integrated on STM32 Nucleo board must be used to program the STEVAL-STLCS01V1 (SensorTile). The developer can download the relevant version of the ST-LINK/V2-1 USB driver at STSW-LINK008 or STSW-LINK009.

Connect STEVAL-STLCS01V1 (SensorTile) on the STEVAL-STLCR01V SensorTile Cradle board or on the STEVAL-STLCX01V1 Sensortile Cradle Expansion board.

Use the SWD connector to connect the Sensortile Cradle board to ST-LINK/V2-1 debugger/programmer integrated on the STM32 Nucleo board for programming.

Be sure that CN2 Jumpers are OFF and connect your STM32 Nucleo board to the SensorTile Cradle through the provided cable paying attention to the polarity of the connectors. Pin 1 can be identified by a little circle on the pcb silkscreen (STM32 Nucleo board and SensorTile Cradle Expansion) or by the square shape of the soldering pad of the connector (SensorTile Cradle).

Figure 63. SensorTile Cradle expansion board and Sensor Tile Cradle board connected to ST-LINK/V2-1



2.3.3.3 B-L475E-IOT01A setup

The IoT Discovery kit node board integrates the ST-LINK/V2-1 debugger/programmer. You can download the relevant version of the ST-LINK/V2-1 USB driver at STSW-LINK008 or STSW-LINK009.

The board can be used in its default factory configuration state by connecting a USB Micro-B cable to the CN7 ST-LINK connector to program and debug the STM32L475VGT6 microcontroller.

2.3.3.4 STEVAL-MKSBOX1V1 setup

To program the STEVAL-MKSBOX1V1 (SensorTile.box) you need an ST-LINK/V2 debugger/programmer (integrated in the B-L475E-IOT01AIoT Discovery kit node board). You can download the relevant version of the ST-LINK/V2-1 USB driver at STSW-LINK008 or STSW-LINK009.

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 one USB type A to Mini-B USB cable

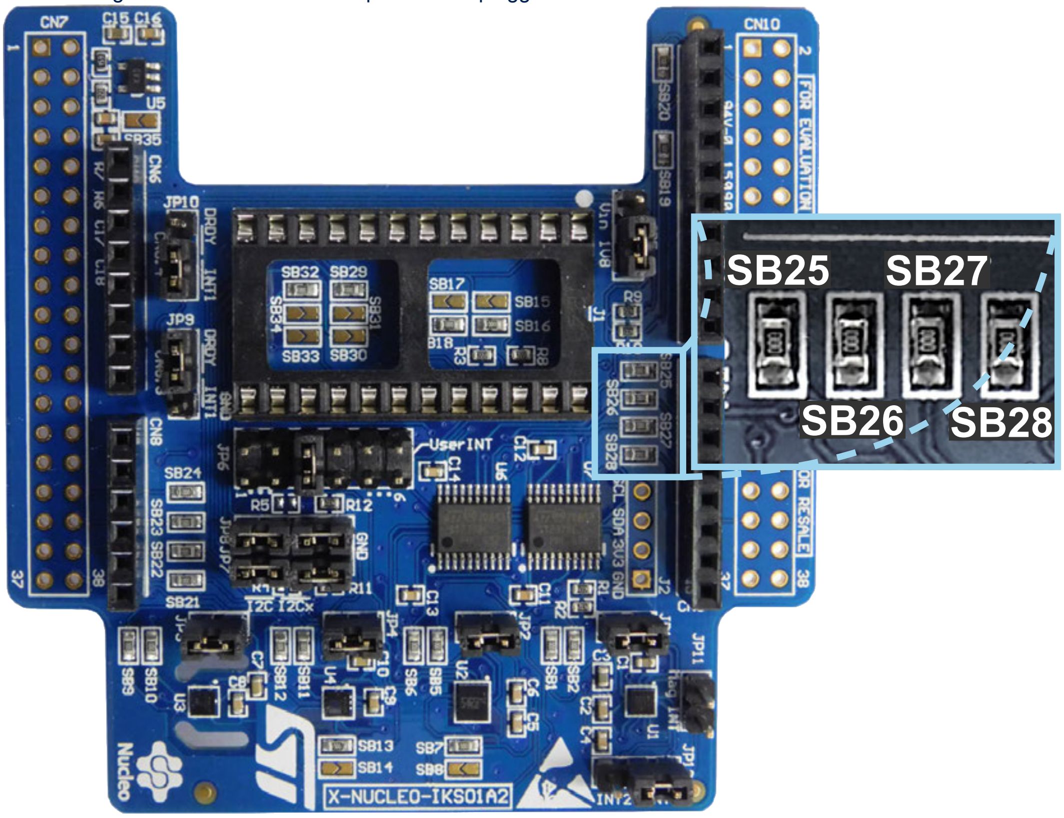
Step 3. Power the STEVAL-MKSBOX1V1 by plugging the battery or using a USB type A Micro-B USB cable connected to the PC

2.3.3.5 Important additional hardware information

For the STM32 Nucleo board: before connecting the X-NUCLEO-IKS01A2 to the X-NUCLEO-CCAM02M1 expansion board through the Arduino UNO R3 extension connector, remove these 0-Ω resistors on the X-NUCLEO-IKS01A2 board

Figure 64. X-NUCLEO-CCAM02M1 solder bridge configuration

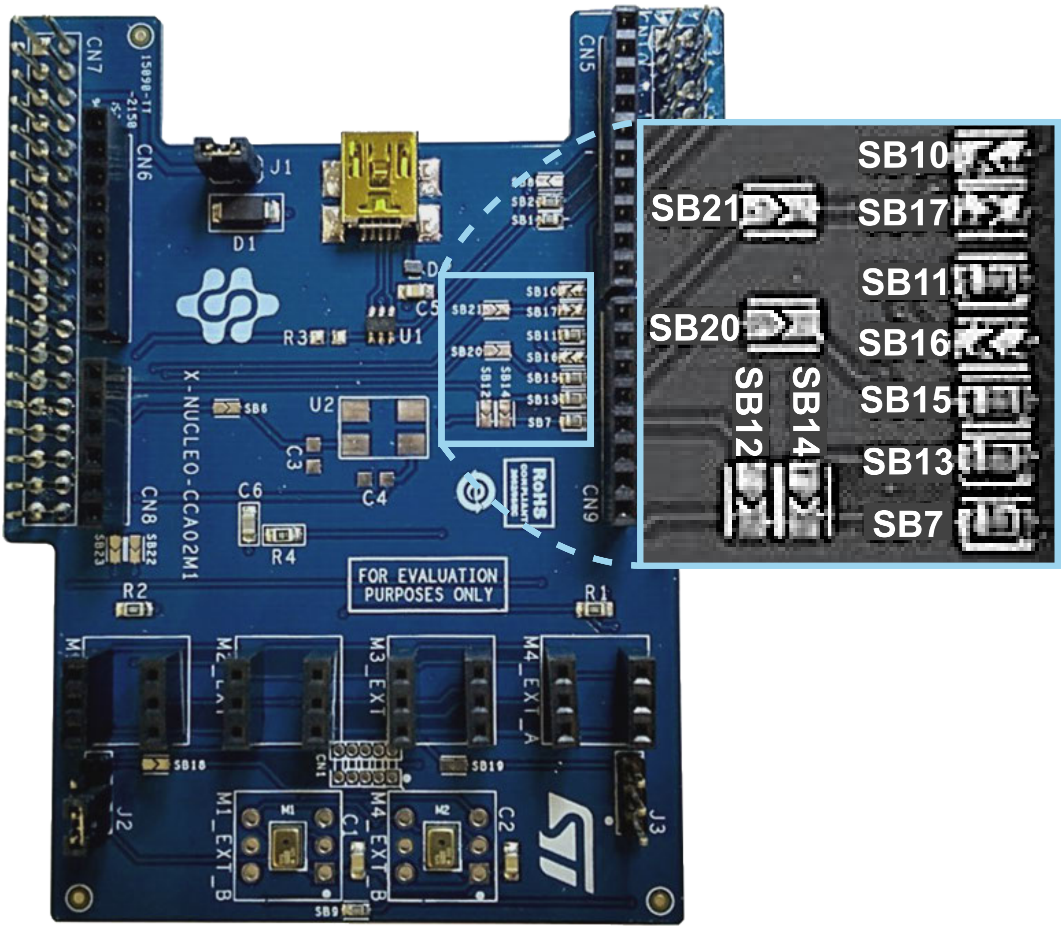
Remove solder bridge SB25 if additional microphones are plugged onto the X-NUCLEO-CCAM02M1 board.



Before connecting the X-NUCLEO-CCAM02M1 board to the STM32 Nucleo L4-series development board, on the X-NUCLEO-CCAM02M1 board:

Figure 65. X-NUCLEO-CCA02M1 solder bridge configuration for the NUCLEO-L476RG board

close solder bridges SB12 and SB16
open solder bridges SB7, SB15 and SB17



3 References

1. Valenti, M., Diment, A., Parascandolo, G., Squartini, S., & Virtanen, T. (2016). DCASE 2016 acoustic scene classification using convolutional neural networks. IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE2016), Budapest, Hungary.
2. Han, Y., Park, J., & Lee, K. (2017). Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification. the Detection and Classification of Acoustic Scenes and Events (DCASE), 1-5.
3. Efficient Light Harvesting for Accurate Neural Classification of Human Activities, Proceedings of 2018 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, January 12-14, 2018
4. Embedded Real-Time Fall Detection with Deep Learning on Wearable Devices; Euromicro DSD/SEAA 2018, August 29 – 31, 2018, Prague | Czech Republic
5. Andrey Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks", Applied Soft Computing 62 (2018), pp 915-922

Revision history

Table 2. Document revision history

Date	Version	Changes
13-Dec-2018	1	Initial release.
15-Apr-2019	2	Updated Introduction, Section 1.1 Overview, Section 1.3 Folder structure, Figure 1. FP-AI-SENSING1 software architecture, Section 1.6 The installation process, Section 1.8.1 Sample application description, Section 1.9.7 AI Data Logging, Section 2.3.1 Hardware setup, Section 2.3.2 Software setup. Added Section 2.1.6 STM32L4 Discovery kit IoT node and Section 2.3.3.3 B-L475E-IOT01A setup.
02-May-2019	3	Updated Introduction, Section 1.1 Overview, Figure 1. FP-AI-SENSING1 software architecture and Section 1.5 The boot process. Added Section 2.1.7 STEVAL-MKSBOX1V1 evaluation kit and Section 2.3.3.4 STEVALMKSBOX1V1 setup. Added STEVAL-MKSBOX1V1 compatibility information.
01-Jul-2019	4	Updated Introduction, Section 1.1 Overview, Figure 1. FP-AI-SENSING1 software architecture, Section 1.5 The boot process, Section 1.6 The installation process, Section 1.9.1 Sample application description, Section 1.9.3 Acoustic scene classification (ASC), Section 1.9.4 Setting up the terminal window, Figure 11. Initialization phase, Figure 12. UART console output when a device is connected to the board, Section 1.10.6 Audio Classification, Figure 29. USB Mass Storage Device Start command, Figure 31. STBLESensor (Android version) Battery and RSSI information, Figure 44. STBLESensor (Android application) UART console after FOTA upload, before installation and Section 1.13 Generation of AI libraries with X-CUBE-AI. Added Section 1.7 Generation of a partial update binary file for FOTA, Section 1.9.3.1 Quantization, Section 1.11 Command line interface (CLI), Section 1.11.1 Configuration and setup, Section 1.11.2 CLI usage, Section 1.11.2.1 Usage example 1: selecting and running the HAR GMP NN model, Section 1.11.2.2 Usage example 2: running the ASC NN model and Section 1.11.2.3 Usage example 3: running the the HAR and ASC models in parallel.
25-Oct-2019	5	Updated Section 1.9.2 Human activity recognition (HAR), Section 1.9.3 Acoustic scene classification (ASC), Section 1.9.3.1 Quantization and Section 1.13 Generation of AI libraries with X-CUBE-AI. Added Section 1.10.7 AI Multi Network.
05-May-2020	6	Updated Section 1.13 Generation of AI libraries with X-CUBE-AI.

Contents

1 FP-AI-SENSING1 software description . . . 2

1.1 Overview . . . 2
1.2 Architecture . . . 3
1.3 Folder structure . . . 4
1.4 Flash management . . . 5
1.5 The boot process . . . 6
1.6 The installation process . . . 7
1.7 Generation of a partial update binary file for FOTA . . . 11
1.8 Firmware-Over-The-Air (FOTA) update . . . 11
1.9 APIs . . . 11
1.9.1 Sample application description . . . 11
1.9.2 Human activity recognition (HAR) . . . 11
1.9.3 Acoustic scene classification (ASC) . . . 12
1.9.4 Setting up the terminal window . . . 15
1.9.5 Sample application startup description . . . 15
1.10 Android and iOS sample client application . . . 17
1.10.1 Main page . . . 18
1.10.2 Plot data . . . 19
1.10.3 Settings, serial and debug console . . . 20
1.10.4 Enable hardware features . . . 23
1.10.5 Activity recognition . . . 26
1.10.6 Audio Classification . . . 28
1.10.7 AI Multi Network . . . 28
1.10.8 AI Data Logging . . . 31
1.10.9 RSSI and battery . . . 38
1.11 Command line interface (CLI) . . . 38
1.11.1 Configuration and setup . . . 39
1.11.2 CLI usage . . . 41
1.12 Firmware-Over-The-Air update with STBLESensor . . . 45
1.13 Generation of AI libraries with X-CUBE-AI . . . 49

1.14 Power profiling 53
2 **System setup guide 54**
2.1 Hardware description 54
2.1.1 STM32 Nucleo 54
2.1.2 X-NUCLEO-CCA02M1 expansion board 54
2.1.3 X-NUCLEO-IDB05A1 expansion board 55
2.1.4 X-NUCLEO-IKS01A2 expansion board 56
2.1.5 STEVAL-STLKT01V1 development kit 57
2.1.6 STM32L4 Discovery kit IoT node 60
2.1.7 STEVAL-MKSBOX1V1 evaluation kit 61
2.2 Software description 62
2.3 Hardware and software setup 63
2.3.1 Hardware setup 63
2.3.2 Software setup 63
2.3.3 System setup guide 64
3 **References 69**
Revision history 70

List of figures

Figure 1. FP-AI-SENSING1 software architecture . . . 4
Figure 2. FP-AI-SENSING1 package folder structure . . . 4
Figure 3. FP-AI-SENSING1 Flash structure . . . 5
Figure 4. FP-AI-SENSING1 boot sequence . . . 7
Figure 5. Content of a project folder . . . 8
Figure 6. BootLoader and SENSING1 installation . . . 9
Figure 7. SENSING1 dump process . . . 10
Figure 8. STM32 Audio device selection in Windows . . . 13
Figure 9. X-CUBE-AI: STM32 fixed-point implementation . . . 14
Figure 10. Terminal setting . . . 15
Figure 11. Initialization phase . . . 16
Figure 12. UART console output when a device is connected to the board . . . 17
Figure 13. STBLESensor (Android version) main page following BLE connection . . . 18
Figure 14. STBLESensor (Android version) accelerometer plot . . . 19
Figure 15. STBLESensor (Android version) menu selection . . . 20
Figure 16. STBLESensor (Android version) Local Name change . . . 21
Figure 17. STBLESensor (Android version) Debug console (stdin/stdout/stderr) . . . 22
Figure 18. STBLESensor (Android version) multiple hardware feature . . . 23
Figure 19. STBLESensor (Android version) hardware feature menu . . . 24
Figure 20. STBLESensor (Android version) hardware feature examples . . . 25
Figure 21. STBLESensor (Android version) HAR_GMP or HAR_IGN activity recognition page . . . 26
Figure 22. STBLESensor (Android version) HAR_IGN_WSDM activity recognition page . . . 27
Figure 23. STBLESensor (Android version) ASC Audio Classification page . . . 28
Figure 24. STBLESensor (Android version) Multi Network page . . . 29
Figure 25. STBLESensor (Android version) AI Data Log . . . 31
Figure 26. STBLESensor (Android version) AI Data Log – sample rates . . . 32
Figure 27. STBLESensor (Android version) AI Data Log – label and start . . . 33
Figure 28. STBLESensor (Android version) AI Data Log – adding a label . . . 34
Figure 29. STBLESensor (Android version) select a label that is logging data . . . 35
Figure 30. USB Mass Storage Device Start command . . . 36
Figure 31. STM32L4 IoT Discovery node USB Mass Storage Device connection . . . 37
Figure 32. STBLESensor (Android version) Battery and RSSI information . . . 38
Figure 33. STMicroelectronics STLink Virtual COM Port . . . 39
Figure 34. STMicroelectronics STLink Virtual COM Port new connection (COM33) . . . 40
Figure 35. Serial port setup . . . 40
Figure 36. Terminal setup . . . 41
Figure 37. CLI usage example with *info* and *help* commands . . . 42
Figure 38. CLI usage example 1 . . . 43
Figure 39. CLI usage example 2 . . . 43
Figure 40. Console output disabled by pressing [**ESC**] . . . 44
Figure 41. CLI usage example 3 . . . 44
Figure 42. STBLESensor (Android version) firmware upgrade page . . . 45
Figure 43. STBLESensor (Android version) firmware update file selection . . . 46
Figure 44. STBLESensor (Android application) feedback during and after FOTA transmission . . . 47
Figure 45. STBLESensor (Android application) UART console after FOTA upload, before installation . . . 48
Figure 46. Pre-trained models . . . 49
Figure 47. Neural network file locations in include and src folders . . . 50
Figure 48. Neural network weights . . . 51
Figure 49. HAR GMP power profiling measured on sensor tile . . . 53
Figure 50. ASC power profiling measured on sensor tile . . . 53
Figure 51. STM32 Nucleo board . . . 54
Figure 52. X-NUCLEO-CCA02M1 expansion board . . . 55

Figure 53. X-NUCLEO-IDB05A1 expansion board 56
Figure 54. X-NUCLEO-IKS01A2 MEMS and environmental sensor expansion board 56
Figure 55. STLCS01V1 board photo 58
Figure 56. STLCR01V1 board photo 59
Figure 57. STLCX01V1 board photo 60
Figure 58. B-L475E-IOT01A Discovery kit 61
Figure 59. STEVAL-MKSBOX1V1 evaluation kit 62
Figure 60. STM32 Nucleo plus X-NUCLEO-CCA02M1 boards 64
Figure 61. STM32 Nucleo plus X-NUCLEO-CCA02M1 plus X-NUCLEO-IDB05A1 boards 65
Figure 62. STM32 Nucleo plus X-NUCLEO-CCA02M1 plus X-NUCLEO-IDB05A1 plus X-NUCLEO-IKS01A2 boards 65
Figure 63. SensorTile Cradle expansion board and Sensor Tile Cradle board connected to ST-LINK/V2-1 66
Figure 64. X-NUCLEO-CCAM02M1 solder bridge configuration 67
Figure 65. X-NUCLEO-CCA02M1 solder bridge configuration for the NUCLEO-L476RG board 68