How to implement Distributed Deep Edge AI with STMicroelectronics?
1. ST strategy for TinyML
2. The Context Awareness use case & function pack
3. Human Activity Recognition in Smart Sensor LSM6DSOX
4. Acoustic Scene Classification on STM32L4R9ZI6J6 MCU
5. Context Awareness Demo
ST strategy for TinyML
Moving closer to data source

Deep Edge AI
Data are processed in the smart sensors, smart nodes and smart gateways

Pros
- Responsiveness (fast)
- Bandwidth (low)
- Improved privacy (data stored locally)
- Energy saving

Cons
- Distribute the processing over multiple devices

* Optional
AI in endpoints and gateways

**Solutions**

- **STM32 32-bit Microprocessors** with ARM® Cortex® A+M cores, embedded memory and peripherals
- **STM32 32-bit Microcontrollers** with ARM Cortex-M cores, embedded memory and peripherals
- **Advanced sensors** that can be used alone or in combination with an MCU/MPU to further improve power savings

**Tools**

- **TensorFlowLite & X-Linux-AI + STM32Cube.AI**
- **STM32Cube.AI tool** to automatically optimize Artificial Neural Networks for STM32 MCUs and MPUs
- **Current generation of Smart Sensors with MLC and FSM to enabling ML processing**
Context awareness
use case & function pack
Context awareness: where are we? what are we doing?

FP-AI-CXTAWARE1: the best power system saving solution

Sensor
Human
Activity
Recognition

Audio
Scene
Classification

Data collection
Decision Trees generation
and upload in sensor

Sensor with Machine Learning Core

Sensor Data
Computation Block
Decision Tree

Accelerometer
Filters
Motion classifier

External sensors
Digital pre-defined features
Features

UNICO-GUI

Audio data recording
Pre-processing
DB Preparation of NN Training and Test

Neural Network Topology
Definition, Training and Test
Using existing DL Frameworks

Optimized Neural Network code automatically generated for STM32

Upload Neural Network code on STM32 MCU

STM32 Cube.AI

Open Framework
Context awareness activity recognition: What do you need?

STEVAL-MKSBOX1V1 evaluation board

Multi-sensor kit with portable sensor box and smart sensor app

- Compact casing, IP54-compliant
  57 x 38 x 20 mm (L x W x h)
  Additional cases (with flanges or hinges) available

- 500 mA-h Li-Po battery

- 8 GB µSD card as mass memory
  extendable to 64 GB

- Compatible with “ST BLE Sensor” app
  Available on Google Play and App Store
Where can you find the SensorTile.box (STEVAL-MKSBOX1V1)?

Evaluation kit available in ST eStore and Distributors: https://estore.st.com/en/steval-mksbox1v1-cpn.html

Contents:
• STEVAL-MKSBOX1V1 with 2 different cases
• SD-Card and rechargeable battery
• JTAG20 to STDC14 adapter
• STDC14 cable
### What’s inside the SensorTile.box (STEVAL-MKSBOX1V1)?

<table>
<thead>
<tr>
<th>Motion Sensors</th>
<th>Environmental Sensors</th>
<th>Processing</th>
<th>Connectivity</th>
<th>Power management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-power 6-axis IMU,</td>
<td>Altimeter / Pressure sensor</td>
<td>STM32L4 low-power MCU</td>
<td>Bluetooth Low Energy Module</td>
<td>Battery charger</td>
</tr>
<tr>
<td>embedding Machine</td>
<td>LPS22HH</td>
<td>STM32L4R9ZI-J6</td>
<td>SPBTLE-1S</td>
<td>STBC02</td>
</tr>
<tr>
<td>Learning LSM6DSOX</td>
<td>Accurate temperature sensor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-performance and</td>
<td>STTS751</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>low power accelerometers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIS3DHH &amp; LIS2DW12</td>
<td>Humidity sensor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magnetometer</td>
<td>HTS221</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIS2MDL</td>
<td>Analog wide-band microphone</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MP23ABS1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*STM32L4*: Low-power Microcontroller
*LIS3DHH*: High-performance Accelerometer
*LIS2DW12*: Low-power Accelerometer
*LIS2MDL*: Magnetometer
*LPS22HH*: Altimeter
*STTS751*: Temperature sensor
*HTS221*: Humidity sensor
*MP23ABS1*: Analog wide-band microphone
*STM32L4*: Low-power Microcontroller
*SPBTLE*: Bluetooth Low Energy Module
*STBC02*: Battery charger

---

*Note:* The information is based on visual components and typical names of components commonly found in sensor tiles.
Pre-integrated application example

The SensorTile.box is supported by several STM32Cube function packs

<table>
<thead>
<tr>
<th>Product Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP-SNS-STBOX1</td>
<td>For building custom applications using the Pro Mode</td>
</tr>
<tr>
<td>FP-ATR-BLE1</td>
<td>For asset tracking using BLE connectivity</td>
</tr>
<tr>
<td>FP-SNS-ALLMEMS</td>
<td>Ultra-low power IoT node with BLE connectivity, digital microphone, environmental and motion sensors</td>
</tr>
<tr>
<td>FP-AI-SENSING1</td>
<td>Ultra-low power IoT node with artificial intelligence (AI) application based on audio and motion sensing</td>
</tr>
<tr>
<td>FP-AI-CXTAWARE1</td>
<td>Ultra-low power context awareness with distributed artificial intelligence (AI): acoustic analysis with NN on MCU and motion analysis with ML on IMU</td>
</tr>
</tbody>
</table>
What’s inside the FP-AI-CTXAWARE1?

Complete firmware
Context awareness 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 X-CUBE-AI, featuring example implementation of neural networks for real time acoustic scene classification (ASC) application.

Concurrent execution ML & NN
With concurrent execution of the MLC for HAR and neural network for ASC.

Features
Ultra-low power implementation based on the use of an RTOS Compatible with ST BLE Sensor application for Android/iOS Free, user-friendly license terms

Overall Software Architecture

<table>
<thead>
<tr>
<th>Application</th>
<th>FP-AI-CTXAWARE1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middleware</td>
<td>BLE</td>
</tr>
<tr>
<td></td>
<td>FreeRTOS</td>
</tr>
<tr>
<td></td>
<td>USB Device</td>
</tr>
<tr>
<td>Hardware Abstraction</td>
<td>STM32Cube Hardware Abstraction Layer (HAL)</td>
</tr>
<tr>
<td>Hardware</td>
<td>Evaluation boards</td>
</tr>
<tr>
<td></td>
<td>STEVAL-MKS80X1V1</td>
</tr>
</tbody>
</table>

Latest info available at www.st.com

FP-AI-CTXAWARE1
Setup & application examples

Hardware prerequisites

- 1x STEVAL-MKSBOX1V1 evaluation board
- Laptop/PC with Windows 7, 8 or 10
- 1 x microUSB cable
- 1x USB type A to Mini-B USB cable
- ST-Link/V2 in-circuit debugger/programmer for STM8 and STM32
Setup & application examples
Software and other prerequisites

• **STM32 ST-Link Utility**
  - Download and install STSW-LINK004 from www.st.com

• **FP-AI-CTXAWARE1**
  Copy the .zip file content into a folder on your PC. The package will contain source code example (Keil, IAR, STM32Cube IDE) based on STEVAL-MKSBOX1V1.

• **ST BLE Sensor Application** for Android or iOS, to download from Google Store / iTunes

• **Serial line monitor**, e.g. TeraTerm (https://ttssh2.osdn.jp/)


FP-AI-CTXAWARE1 setup overview

1. www.st.com/stm32ode-fp

2. Android™/iOS™ smartphone and ST BLE Sensor application or Serial Line Monitor

3. Download & unpack

4. FP-AI-CTXAWARE1 package structure
   - Docs
   - BSP, HAL drivers
   - BlueNRG-1, FreeRTOS, AI-Library
   - Applications
   - MLC programs

5. Projects\STM32L4R9ZI-SensorTile.box\Applications\BLELowPower\EWARM
   Projects\STM32L4R9ZI-SensorTile.box\Applications\CLAi\EWARM

6. Select Function Pack: FP-AI-CTXAWARE1
BLELowPower (FP-AI-CTXAWARE1)
BLE version

Environmental

Plot

HW features

Audio scene classification

Activity recognition
BLELowPower (FP-AI-CTXAWARE1)
BLE version

Human Activity Recognition (HAR) running on MLC provides indication about activity status ("Stationary", "Walking", "Jogging", "Biking") while Acoustic Scene Classification (ASC) is used for context awareness ("Indoor", "Outdoor", "In Vehicle")

The Human Activity Recognition (HAR) is always running on MLC (LSM6DSOX) without consuming power on STM32

The Acoustic Scene Classification (ASC) runs periodically for controlling the context, and changing the program on MLC if it is necessary

<table>
<thead>
<tr>
<th>Indoor/Outdoor</th>
<th>In Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stationary</td>
<td>Stationary</td>
</tr>
<tr>
<td>Walking</td>
<td>Walking</td>
</tr>
<tr>
<td>Jogging</td>
<td>Driving</td>
</tr>
<tr>
<td>Biking</td>
<td></td>
</tr>
</tbody>
</table>
Command Line AI Application (CLAi) (FP-AI-CTXWARE1)

Command Line version

Same functionalities as the Bluetooth version

Serial Line Monitor Configuration

At the beginning the Board will print out the usage:

Using the Command Line interface, it’s possible to:

• Run the Audio Scene Classification
• Run the Activity Recognition
• Run the Activity Recognition and the Audio Scene Classification in a combined mode
• Print out the dB measured by microphone
FP-AI-CTXAWARE1

- **DB4483**: STM32Cube function pack for ultra-low power context awareness with distributed artificial intelligence (AI): acoustic analysis with NN on MCU and motion analysis with ML on IMU – *databrief*
- **UM2870**: Getting started with the STM32Cube function pack for ultra-low power context awareness node with artificial intelligence (AI) application based on audio and motion sensing – *user manual*
- **Software setup file**

STEVAL-MKSBOX1V1

**Gerber files, BOM, Schematic**
**DB3903**: SensorTile.box wireless multi sensor development kit with user friendly app for IoT and wearable sensor applications – *databrief*
**UM2580**: How to use the wireless multi sensor development kit with customizable app for IoT and wearable sensor applications – *user manual*

All documents are available in the Design tab of the related products webpage

Consult www.st.com for the complete list
Human activity recognition in smart sensor LSM6DSOX
Machine learning embedded in ST sensors

Classical machine learning

Unsupervised
- Dimension reduction
- Clustering
- Pattern search

Supervised
- Regression
- Classification
- Decision trees
  - Naïve Bayes
  - SVM
  - k-NN
  - Logistic regression

New products with engine embedded:
- 6x IMUs with sensor hub
- Inclinometer
What is a Machine Learning Core (MLC)?

MLC is an in-sensor classification engine based on Decision Tree logic. It is able to increase accuracy with better context detectability, offloading the main processor while the built-in sensors identify motion data.
ML solutions in accelerometers and gyroscopes

Personal Electronics
- Activity recognition
- Gym activity recognition
- Airplane mode detection
- Virtual Reality
- Sensor Fusion
- Vehicle stationary detection

Industrial IoT
- Smart antennas
- Structural health monitoring
- Leveling instruments
- Equipment installation and monitoring

Devices:
- LSM6DSOX
- LSM6DSRX
- LSM6DSO32X
- ISM330DHCX
- IIS2ICLX
Machine Learning solutions in sensors: new developer model approach

Shorter development time and better accuracy with Machine Learning techniques (Decision Trees)

How it works in 5 simple steps and with an intuitive use case:

1. User defines **Classes** to be recognized
2. Label data and select **filters and features**
3. Build the **decision tree** based on a wide range of SW tools.
4. Program the decision tree into the MLC enabled Sensor
5. Run the MLC model and process incoming data in real time

**UNICO-GUI**
Machine Learning solutions in sensors: ecosystem

A complete suite to create ML applications in sensors

- Function packs for quick prototyping
- Getting start with ST development kit and GUI
- Examples for motion recognition and context recognition
- Videos, training material, in products campaign available
- MEMS & Sensor community: MEMS Machine Learning & AI

st.com/mlc

FP-AI-CXTAWARE1

GitHub

Videos, training material, in products campaign available

MEMS & Sensor community: MEMS Machine Learning & AI
Acoustic scene classification on STM32L4R9ZIJ6 MCU
Enhance sensor technologies with AI on MCU

From sensor data to added-value services

- RGB camera
- IR camera
- Time of Flight
- Audio events and voice
- Inertial Motion Sensors
- QVAR
- Bio-signals
- TMOS
Convert models from any AI framework

Input your framework-dependent, pre-trained Neural Network into the STM32Cube.AI conversion tool

Automatic and fast generation of an STM32-optimized library

STM32Cube.AI offers interoperability with state-of-the-art Deep Learning design frameworks

Any framework that can export models in ONNX open format can be imported

Train NN Model

Convert NN into optimized code for MCU

Process & analyze new data using trained NN
STM32Cube.AI, an STM32CubeMX expansion pack
STM32Cube.AI main features

STM32Cube.AI is available both as a graphical and command line interface

- Quickly assess model footprint requirements
- Select and configure MCU in STM32CubeMX
- Review model layers in STM32Cube.AI

- Generate C-code for pre-trained model
- Support quantized models to reduce RAM, flash and latency with minimal loss of accuracy
- Use light run-time libraries
- Optimize for performance

- Optimize memory allocation
- Fine control of weight mapping
- Split between internal and external memory
- Update model without full FW update

And quickly iterate thanks to on-target validation
Audio scene classification using STM32Cube.AI

Model Accuracy (f32)

- Quantized ASC int8 model on STM32Cube.AI
- Performance on STM32L476 @ 80MHz

Dataset Composition

<table>
<thead>
<tr>
<th>Class</th>
<th>Training Set</th>
<th>Validation Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor</td>
<td>6:02:50</td>
<td>1:21:12</td>
<td>7:19:49</td>
</tr>
<tr>
<td>Outdoor</td>
<td>6:03:22</td>
<td>1:19:28</td>
<td>1:46:53</td>
</tr>
<tr>
<td>In-vehicle</td>
<td>6:03:29</td>
<td>1:20:51</td>
<td>3:58:13</td>
</tr>
<tr>
<td>Total</td>
<td>18:09:41</td>
<td>4:01:31</td>
<td>13:04:55</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg. Acc. (%)</th>
<th>NVM (KB)</th>
<th>RAM (KB)</th>
<th>Inference Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASC CNN int8</td>
<td>89.17</td>
<td>7.71</td>
<td>10.02</td>
<td>30.631</td>
</tr>
</tbody>
</table>
Context awareness demo
Want to learn more?

www.st.com/ai
Our technology starts with You
Premier Sponsor
Automated TinyML

Zero-code SaaS solution

Create tiny models, ready for embedding, in just a few clicks!

Compare the benchmarks of our compact models to those of TensorFlow and other leading neural network frameworks.

Executive Sponsors
Arm: The Software and Hardware Foundation for tinyML

- Connect to high-level frameworks
- Supported by end-to-end tooling
- Connect to Runtime

Application

Optimized models for embedded

Runtime
(e.g. TensorFlow Lite Micro)

Optimized low-level NN libraries
(i.e. CMSIS-NN)

RTOS such as Mbed OS

Arm Cortex-M CPUs and microNPUs

Stay Connected

@ArmSoftwareDevelopers

@ArmSoftwareDev

Resources: developer.arm.com/solutions/machine-learning-on-arm
TinyML for all developers

**Dataset**
- Acquire valuable training data securely
- Enrich data and train ML algorithms

**Edge Device**
- Real sensors in real time
- Open source SDK
- Embedded and edge compute deployment options

**Impulse**
- Test impulse with real-time device data flows

**Test**
- www.edgeimpulse.com
Advancing AI research to make efficient AI ubiquitous

A platform to scale AI across the industry

Power efficiency
- Model design, compression, quantization, algorithms, efficient hardware, software tool

Personalization
- Continuous learning, contextual, always-on, privacy-preserved, distributed learning

Efficient learning
- Robust learning through minimal data, unsupervised learning, on-device learning

Perception
- Object detection, speech recognition, contextual fusion

Reasoning
- Scene understanding, language understanding, behavior prediction

Action
- Reinforcement learning for decision making

Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.
Syntiant Corp. is moving artificial intelligence and machine learning from the cloud to edge devices. Syntiant’s chip solutions merge deep learning with semiconductor design to produce ultra-low-power, high performance, deep neural network processors. These network processors enable always-on applications in battery-powered devices, such as smartphones, smart speakers, earbuds, hearing aids, and laptops. Syntiant's Neural Decision Processors™ offer wake word, command word, and event detection in a chip for always-on voice and sensor applications.

Founded in 2017 and headquartered in Irvine, California, the company is backed by Amazon, Applied Materials, Atlantic Bridge Capital, Bosch, Intel Capital, Microsoft, Motorola, and others. Syntiant was recently named a CES® 2021 Best of Innovation Awards Honoree, shipped over 10M units worldwide, and unveiled the NDP120 part of the NDP10x family of inference engines for low-power applications.

www.syntiant.com

@Syntiantcorp
Platinum Sponsors
Part of your life. Part of tomorrow.

www.infineon.com
Add Advanced Sensing to your Product with Edge AI / TinyML

Pre-built Edge AI sensing modules, plus tools to build your own

Reality AI solutions
- Pre-built sound recognition models for indoor and outdoor use cases
- Solution for industrial anomaly detection
- Pre-built automotive solution that lets cars "see with sound"

Reality AI Tools® software
- Build prototypes, then turn them into real products
- Explain ML models and relate the function to the physics
- Optimize the hardware, including sensor selection and placement
Gold Sponsors
Adaptive AI for the Intelligent Edge

LatentAI

Latentai.com
Build Smart IoT Sensor Devices From Data

SensiML pioneered TinyML software tools that auto generate AI code for the intelligent edge.

- End-to-end AI workflow
- Multi-user auto-labeling of time-series data
- Code transparency and customization at each step in the pipeline

We enable the creation of production-grade smart sensor devices.

densiml.com
Silver Sponsors
Copyright Notice

The presentation(s) in this publication comprise the proceedings of tinyML® EMEA Technical Forum 2021. The content reflects the opinion of the authors and their respective companies. This version of the presentation may differ from the version that was presented at tinyML EMEA. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyML.org