

A Breakthrough Innovation in MEMS Sensors

Introducing LSM6DSOX, iNEMO 6DoF inertial measurement unit (IMU), with Machine Learning Core

Thiago Reis



Technology Tour 2019

Toronto, Canada | May 29



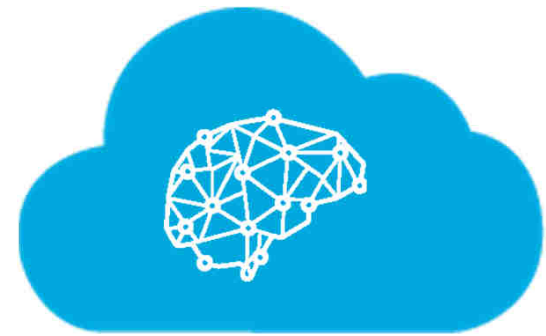
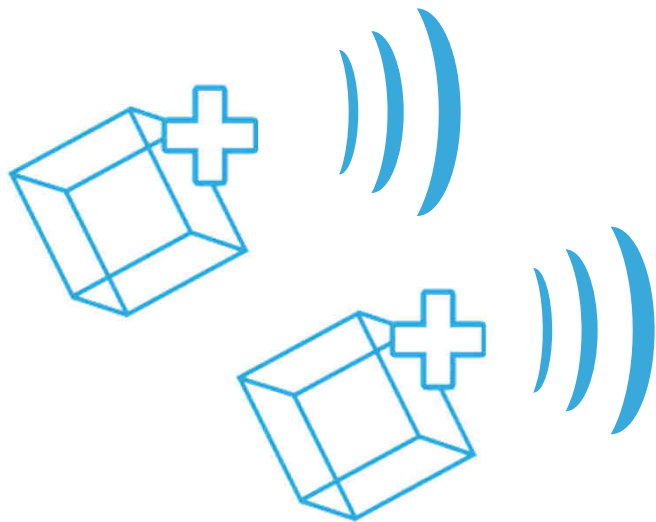
Agenda

2

- Artificial Intelligence in IoT
- ST Smart Sensor: 6-axis IMU with Machine Learning Core
- Why new sensors solutions are attractive?
- Solutions and Comparisons
- Tools
- Conclusions

Artificial Intelligence in IoT

3



Logic Programming

Decision Tree

Machine Learning

Deep Learning

A.I.

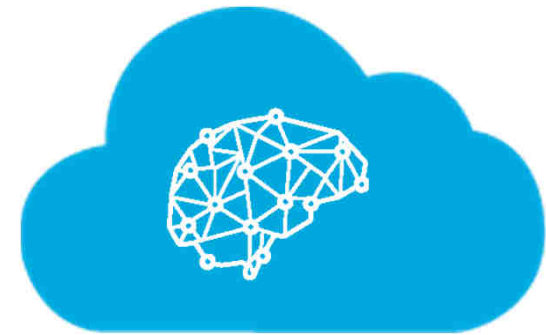
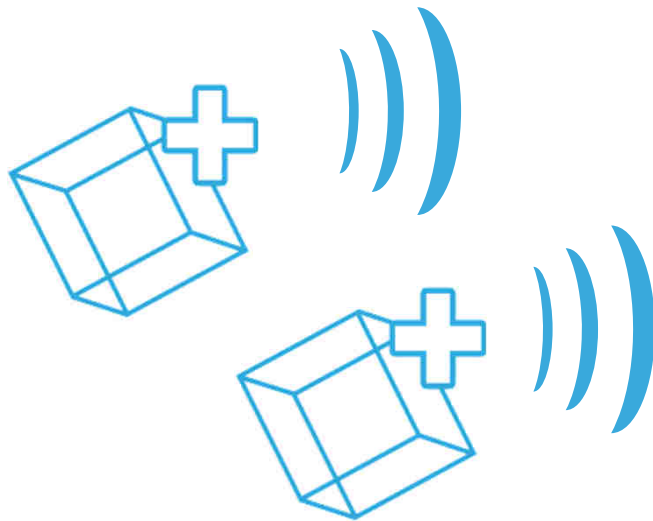
Search Algorithms

Neural Networks

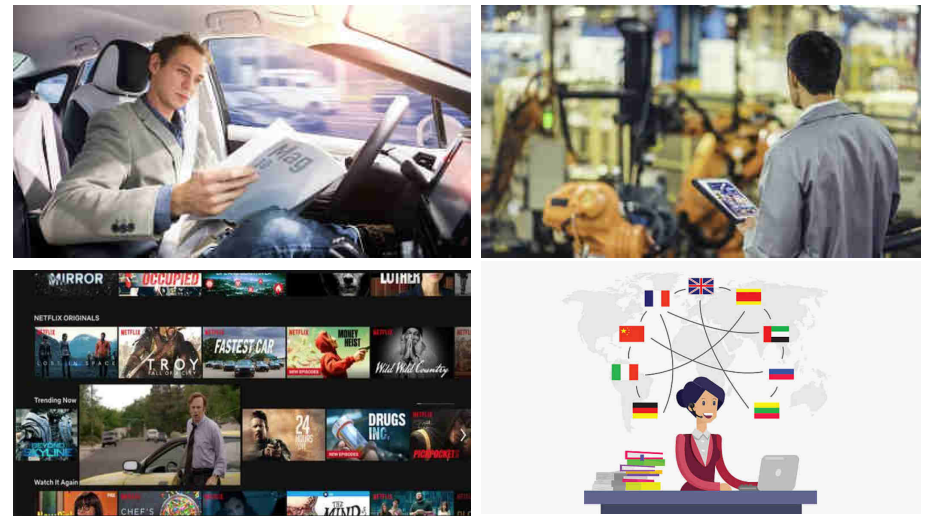
Data Mining

Artificial Intelligence in IoT

4

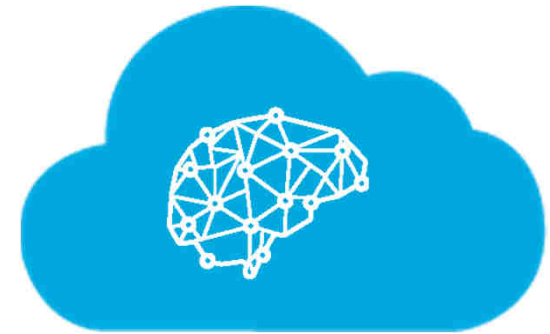
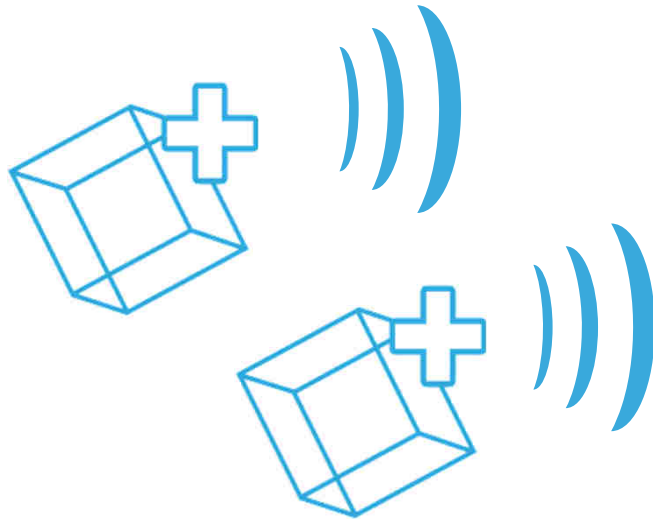


Analysis of vast
amount of information



Artificial Intelligence in IoT

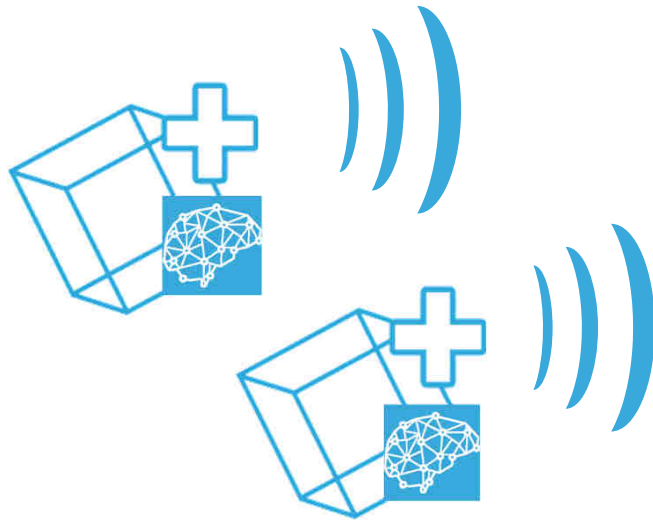
5



Centralized A.I. Architecture

Artificial Intelligence in IoT

6

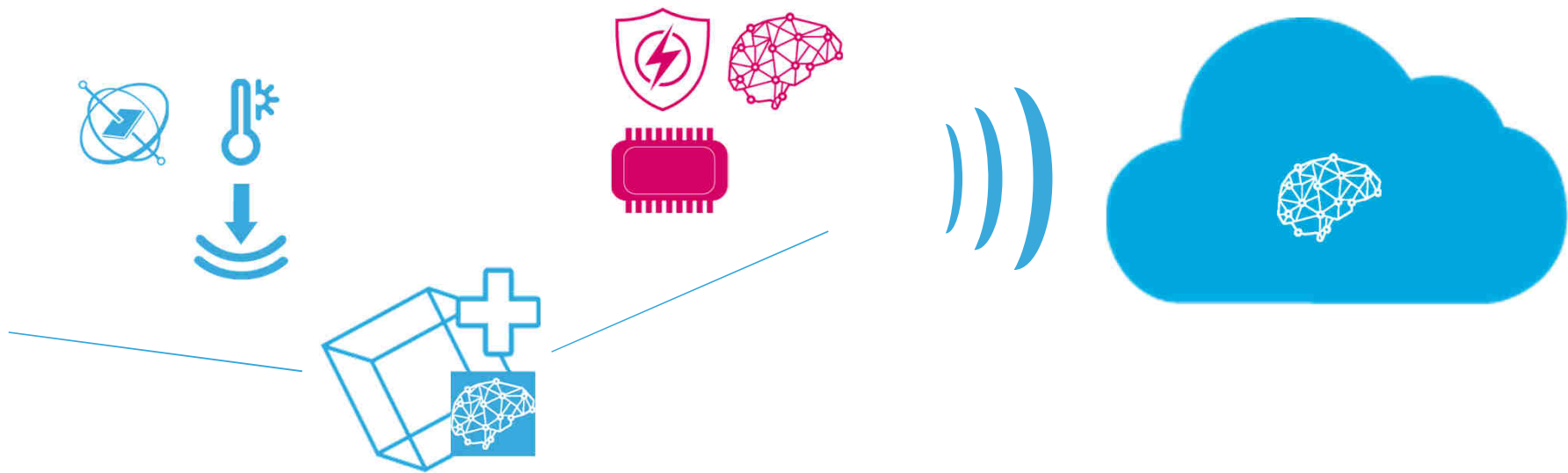


Edge A.I. Architecture



Artificial Intelligence in IoT

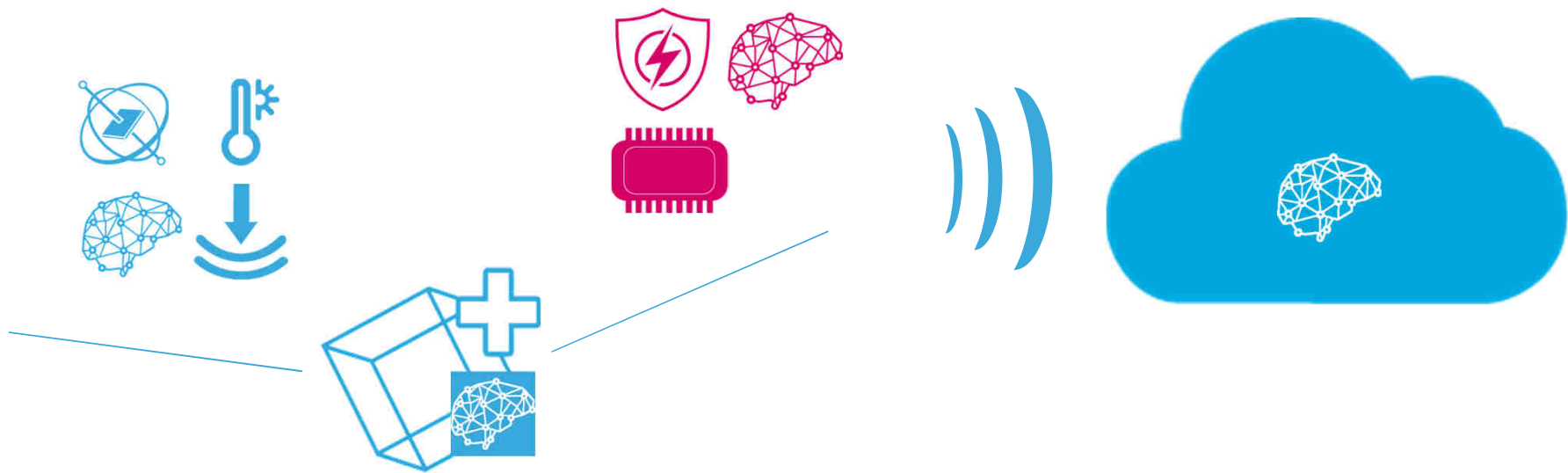
7



Edge A.I. Architecture

Artificial Intelligence in IoT

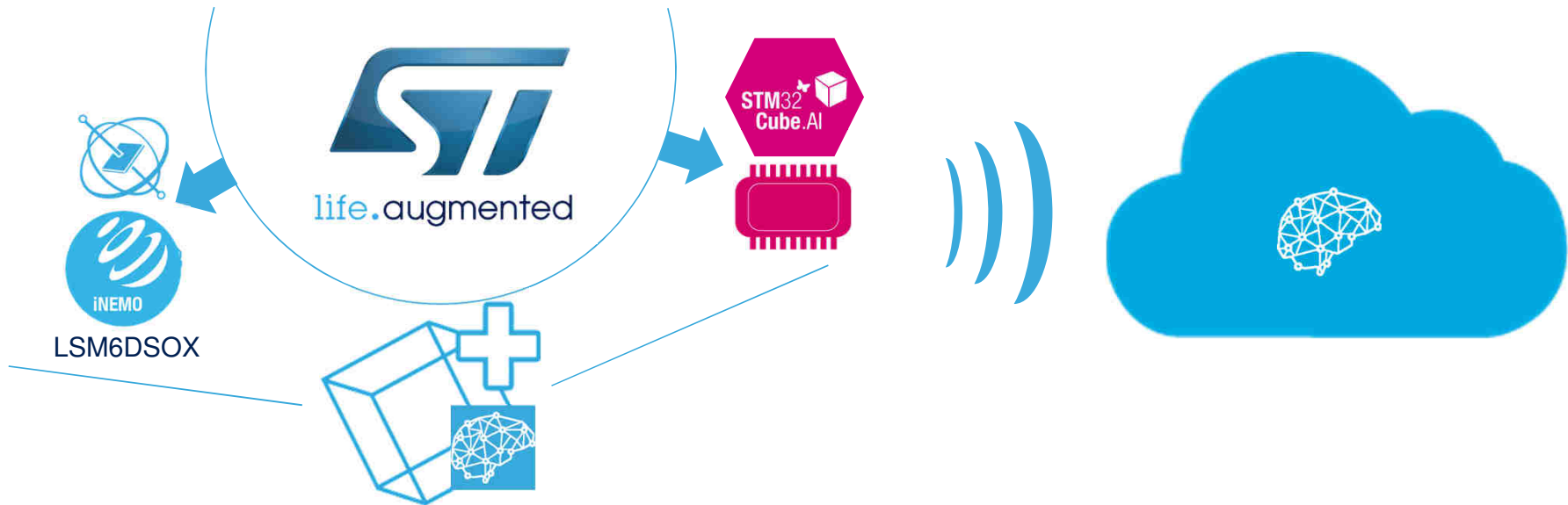
8



Edge A.I. Architecture

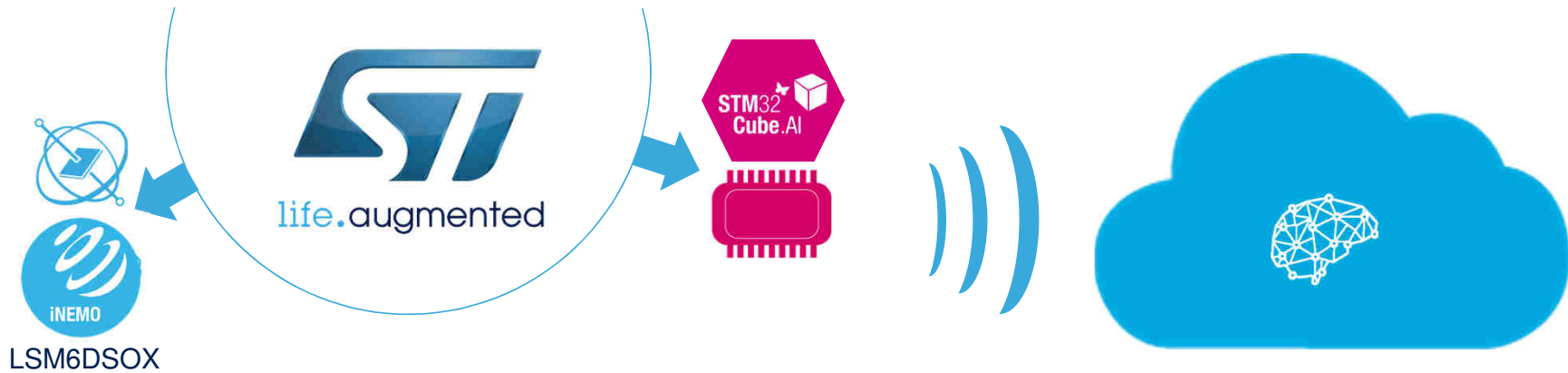
Artificial Intelligence in IoT

9



Artificial Intelligence in IoT

10



Data processing inside the sensor:

- + Local processing
- + Real time analysis
- + Reduced cost of bandwidth
- + Ultra Low Power
- + Intrinsic Security
- = Configurable Logic
- Simple computation (Dec. Tree)

Data processing inside the STM32:

- + Local processing
- + Real time analysis
- + Reduced cost of bandwidth
- = Low Power
- = Improved security
- = Configurable Logic
- = Std Computation (Neural Net.)

Data processing inside the Cloud:

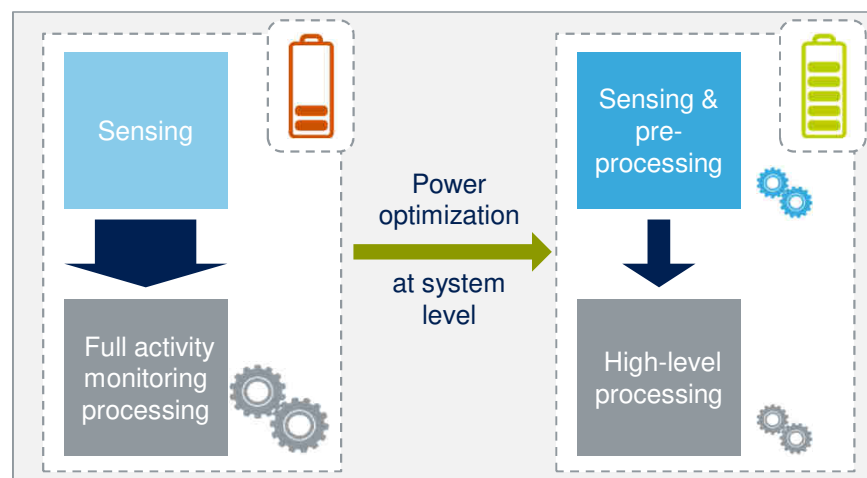
- + Advanced Computation
- + Availability of Wide Amount of Data
- + Continuous Algo Improvement
- = Remote processing
- Data Transfer Latency
- High cost of bandwidth
- Very High current consumption



LSM6DSOX Smart Sensor

11

The First IMU Sensor with a Machine Learning Core

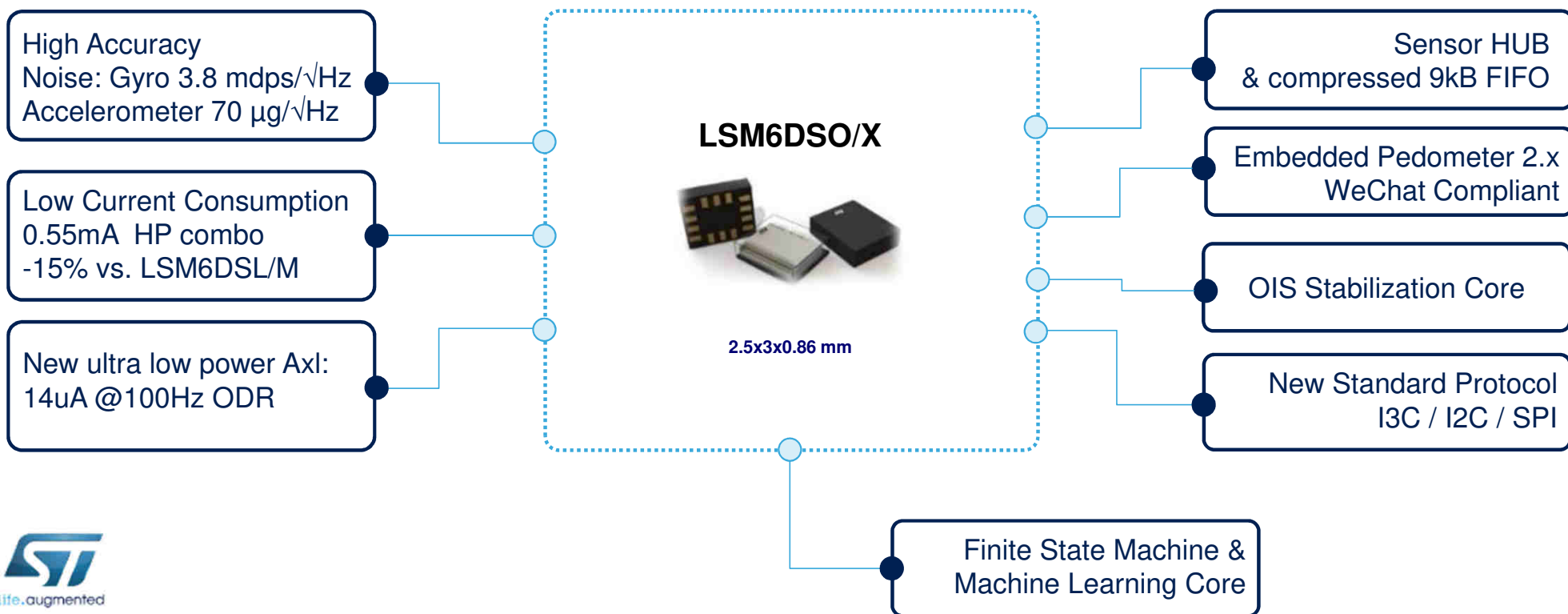




LSM6DSOX Improved Performance

12

Improved Accuracy, Optimized System Power



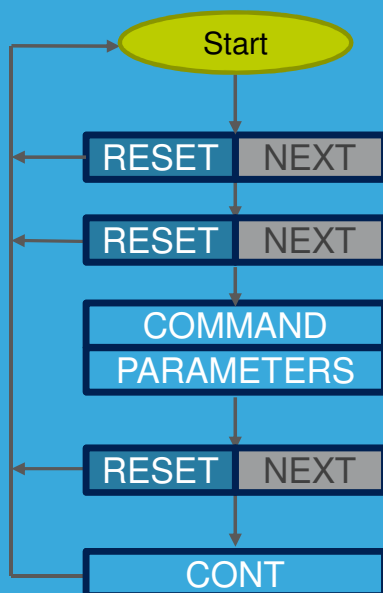


LSM6DSOX Smart Sensor

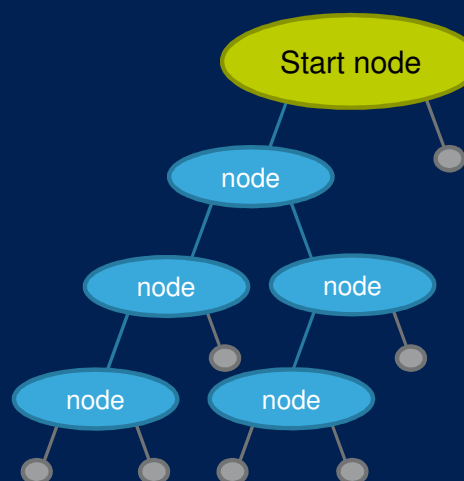
13

The first Smart Sensor embedding:

Finite State Machine



Machine Learning Core



FSM and MLC allows sensors to process data with reduced help of a host Microcontroller

Deductive vs. Inductive algorithms

14

Inertial Algorithms Overview

DEDUCTIVE

Hypotesis



Rules

i.e.
Wake
Up

Finite State Machine

INDUCTIVE

Observation



Model

i.e.
Activity
Rec

Machine Learning Processing

Finite State Machines overview

15

LSM6DSO/X - Finite State Machine

FSM is composed by a series of states, configurable parameters/resources and variables

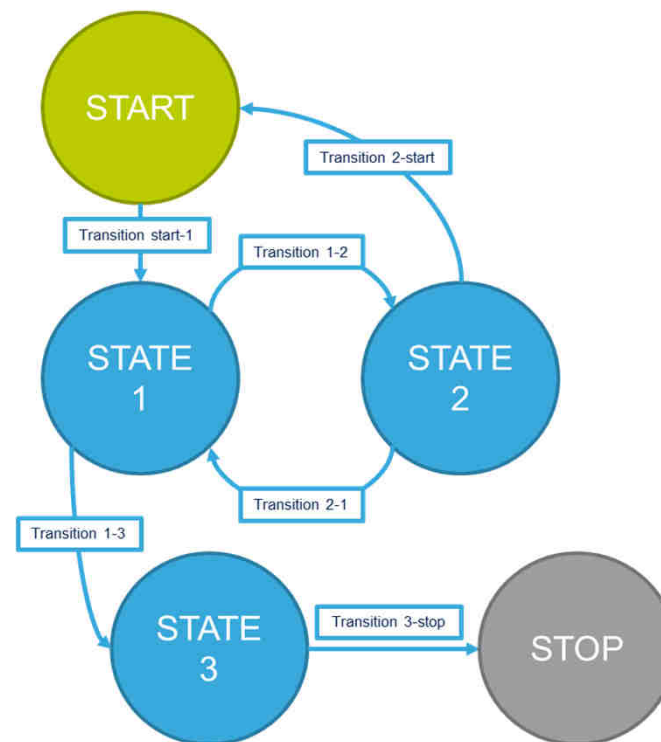
Possibility to implement up to 16 independent FSM

FSM are executed simultaneously or sequentially

FSM can use data from accelerometer, gyro or an external sensor

FSM outputs:

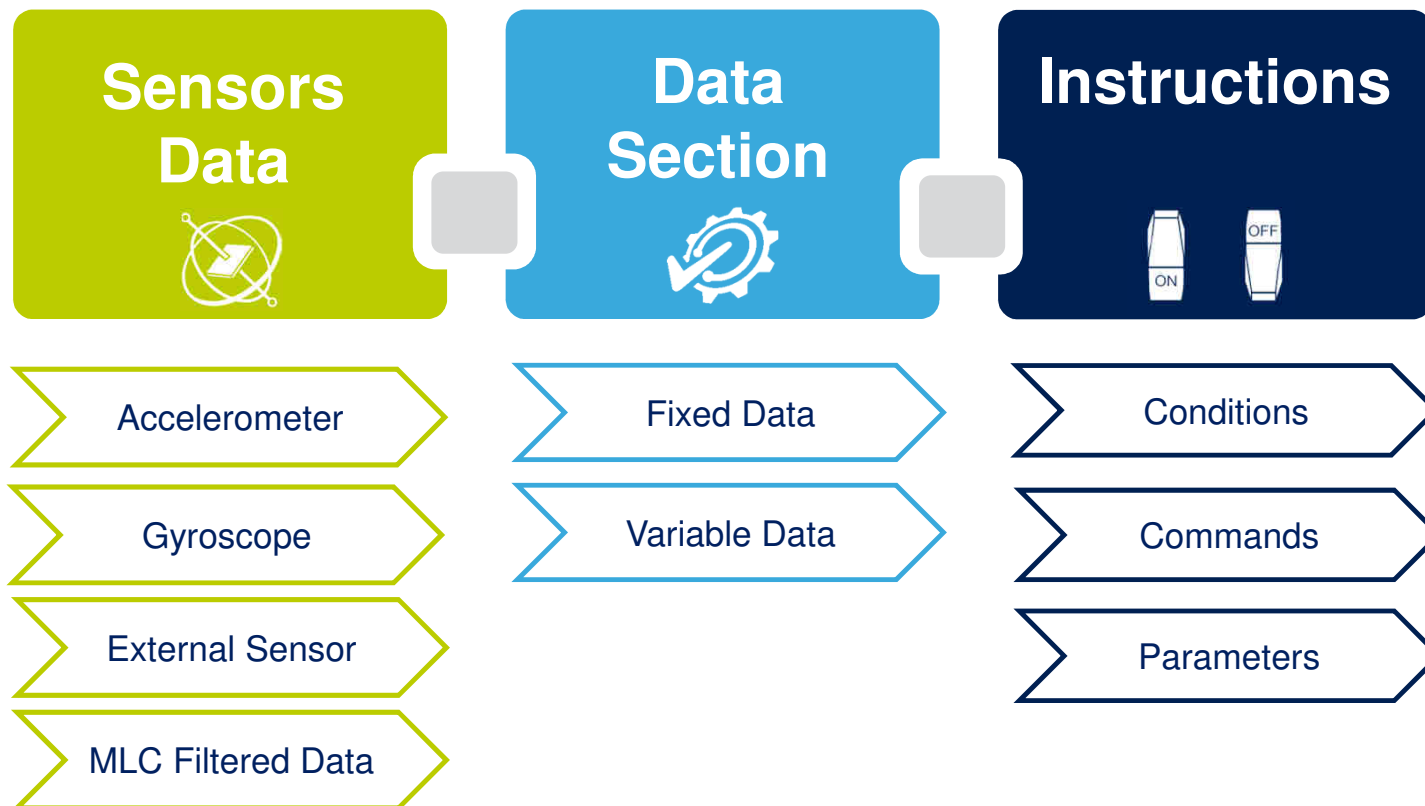
- Interrupt
- Source information



Finite State Machine Structure

In LSM6DSOX

16



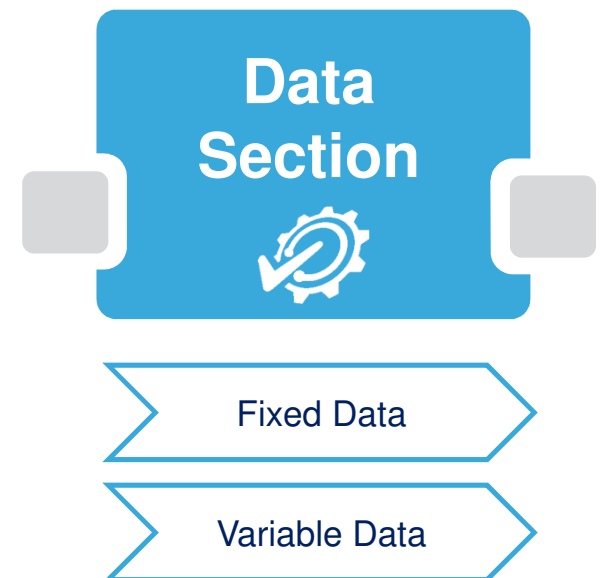
Finite State Machine Structure

In LSM6DSOX

17

There are 2 Data Sections in each FSM program:

- The **Fixed Data Section** of the program is used to specify the resources that are needed in the instructions section. Resources examples:
 - Thresholds
 - Timers
 - Masks
 - Hysteresis
 - Decimation factor
- The **Variable Data Section** is used to define all the values for the used resources.



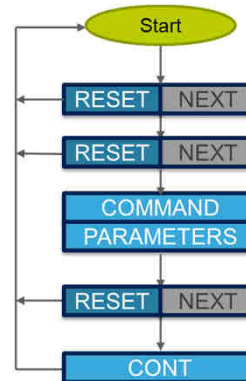
Finite State Machine Structure

In LSM6DSOX

18

In the Instruction Section the user defines the algorithm by using available commands and conditions:

- **Commands** can be followed by parameters. Once executed the program pointer is set to the next command/condition line that is immediately evaluated.
- **Parameter** are application masks, threshold and timers.
- **Conditions** can be:
 - Reset. If true → pointer set to last Reset Point
 - Next. If true → pointer set to Next Line
 - None → interpreter keeps evaluating the same line when next sample occurs



Instructions



Conditions

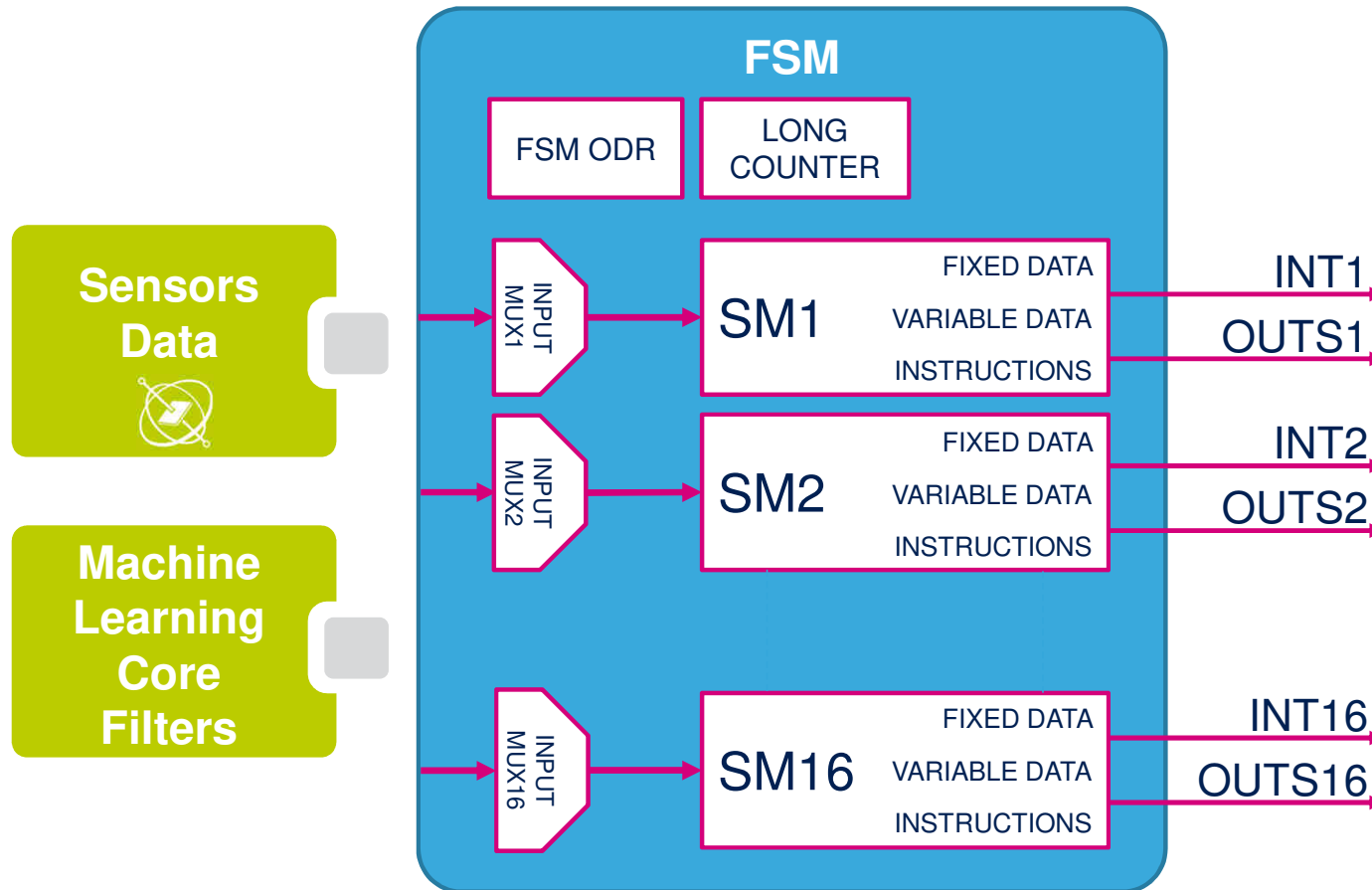
Commands

Parameters

Finite State Machine Blocks

Summary

19



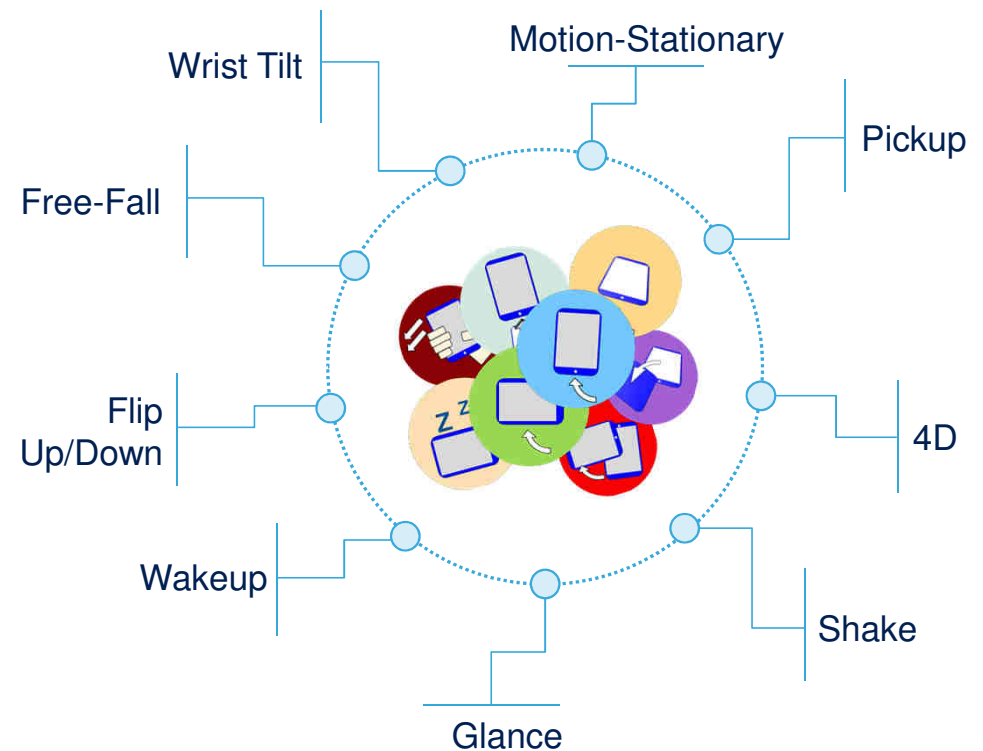
Finite State Machine

Examples of Libraries

20

Exploiting Finite State Machine capabilities

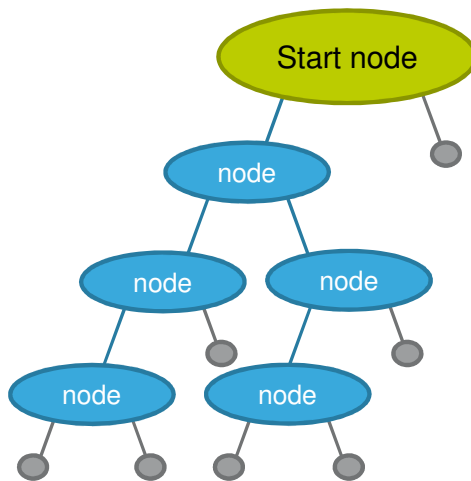
- High Efficiency (Computation Resources, Power)
- High Customization level
- Library of example gestures available:



Machine Learning Core Overview

21

LSM6DSOX - Machine Learning Core



MLC is a programmable logic.

LSM6DSOX embeds a Decision Tree Logic composed by a series of configurable nodes.

Each node is characterized by one "if-then-else" condition.

Up to 8 decision trees can be configured to run simultaneously in LSM6DSOX.

Machine Learning Configuration Flow

22

After the definition of the classes to be recognized: running, walking, car,... Capture data ...

1



User defines
Classes to be
recognized

2



Collect data **Logs** for
each class and **label**
data

3



Select **Features** that
best characterize the
identified classes

4



Machine Learning tools
generate program for
LSM6DSOX based on
Logs and **Features**

5

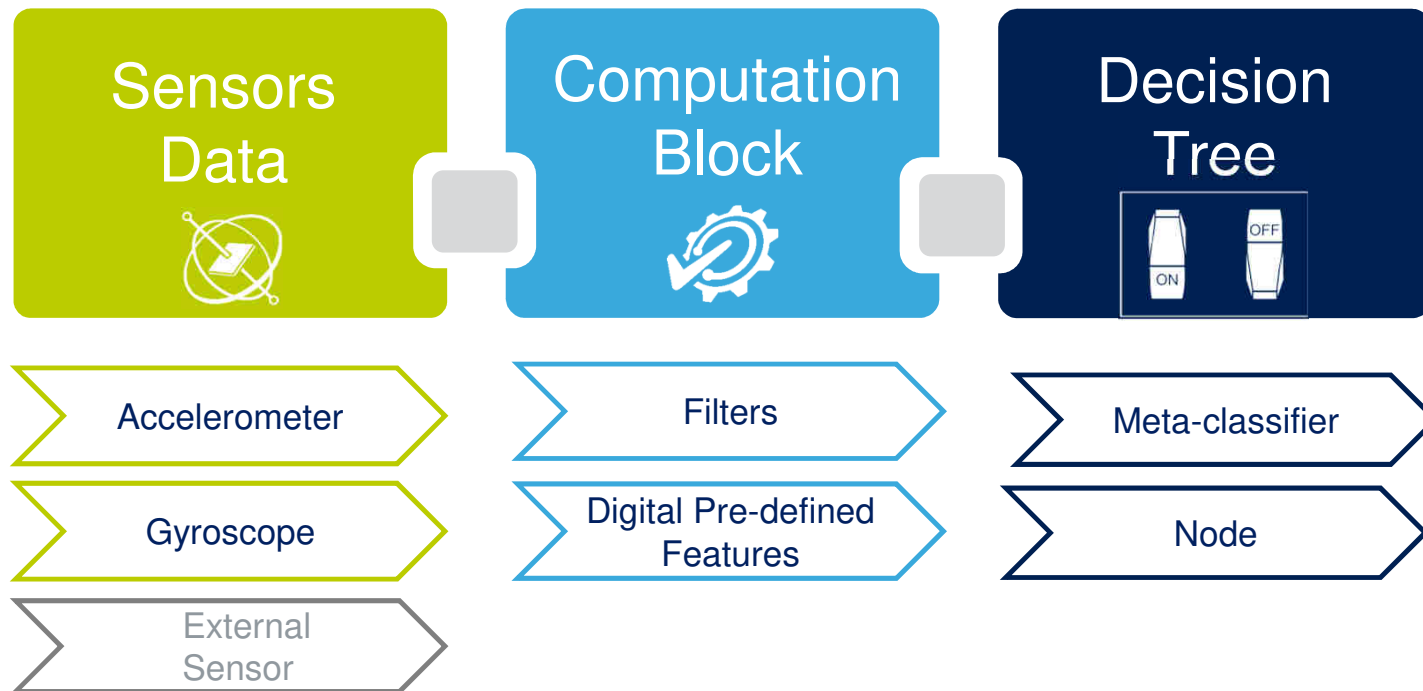


Configure the
LSM6DSOX and
run the application

Machine Learning Core (MLC)

In LSM6DSOX

23

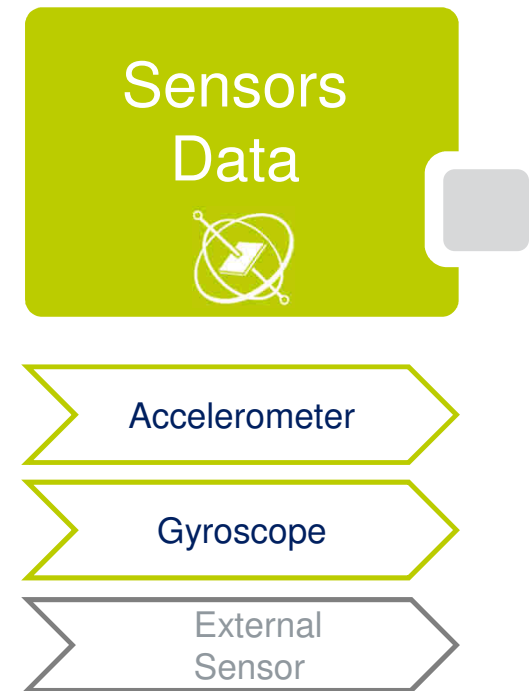


Machine Learning Core (MLC)

In LSM6DSOX

24

- Wide set of Inputs available:
 - **Accelerometer** $\rightarrow [a_x \ a_y \ a_z], [a_v], [a_v^2]$
 - **Gyroscope** $\rightarrow [g_x \ g_y \ g_z], [g_v], [g_v^2]$
 - **External sensor** $\rightarrow [m_x \ m_y \ m_z], [m_v], [m_v^2]$ (*i.e. magnetometer*)
 - **Magnitude** $\rightarrow V = \sqrt{X^2 + Y^2 + Z^2}$

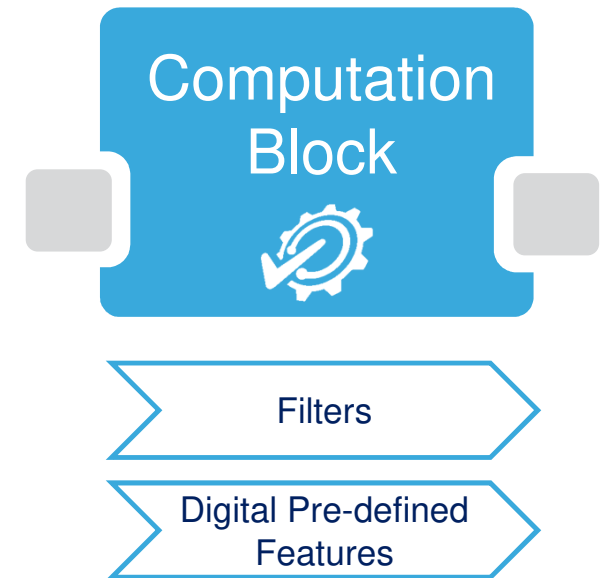


Machine Learning Core (MLC)

In LSM6DSOX

25

- Sensors data can be filtered with a 2nd order IIR **Filter**.
- **Features** are statistical parameter calculated from:
 - Input Data
 - Filtered Input Data
- Examples of features are: Mean, Variance, Energy, Peak to Peak, ...

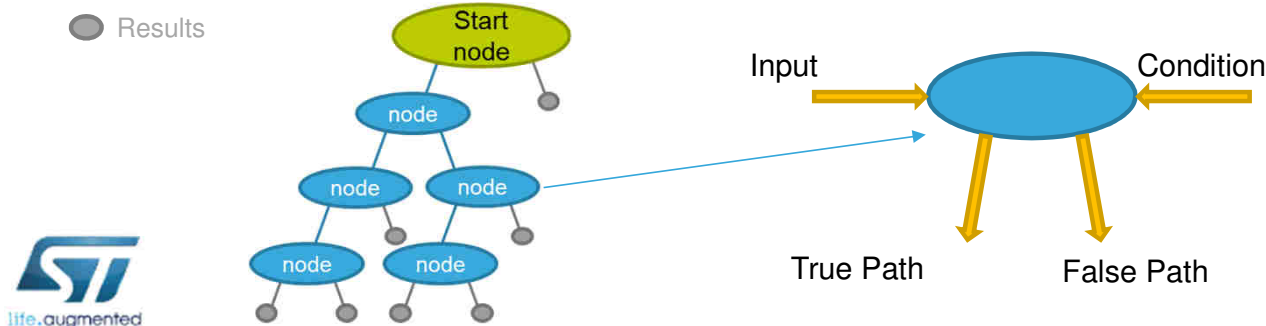
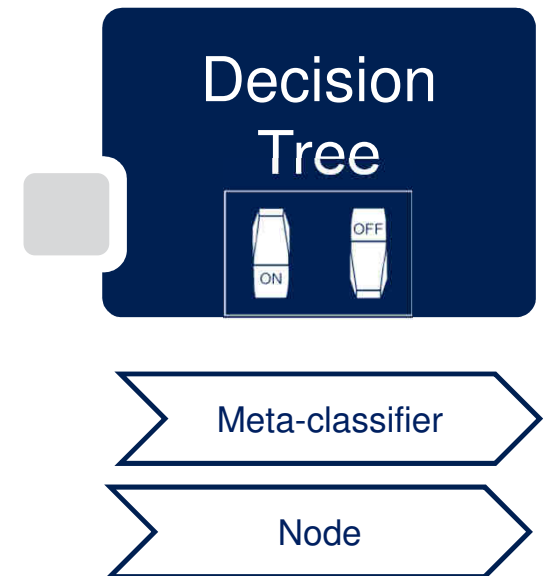


Machine Learning Core (MLC)

In LSM6DSOX

26

- The **Decision Tree** is a predictive model built from training data. The outputs of the computation blocks are the inputs of the decision tree.
- Each **Node** is characterized by one «if-then-else» condition. Some examples of conditions:
 - Mean on Acc_X < 0.5 g
 - Variance on Gyro_Z < 200 dps
- Decision tree can either generate a result at every sample or filter the results with a **Meta-classifier**, to have a more robust output.



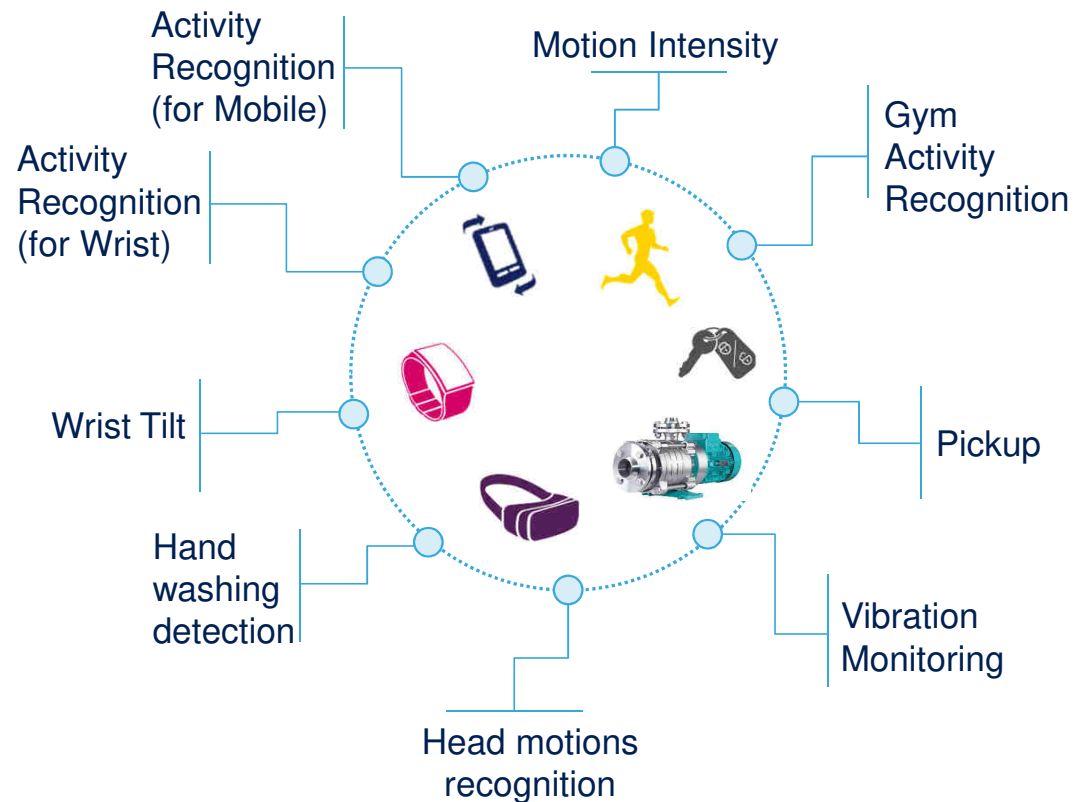
Machine Learning Core

Examples of Libraries

27

Exploiting Machine Learning Core capabilities

- High Efficiency (Computation Resources, Power)
- High Customization level
- Higher Complexity than FSM
- Example Libraries available:





Solutions Comparison

LS6DSO/X Power Consumption

Application Case examples

29

LSM6DSOX Lowest Current Consumption

FSM and MLC are incredibly efficient in current consumption needs:

- + 3 μA for each Finite State Machine

- + 1-15 μA Machine Learning Core

as low power features to recognize customizable conditions and generate interrupts.

Computational Current Consumption

30

Applications has been developed on Machine Learning Core

Algorithm	Sensors used	ODR	Number of decision trees	Number of nodes	MLC additional Current consumption
Vibration Monitoring	A	26 Hz	1	2	1 uA
Motion Intensity	A	12.5 Hz	1	7	1 uA
6D position recognition	A	26 Hz	1	8	2 uA
Activity Recognition for mobile	A	26 Hz	1	126	4 uA

Application Case

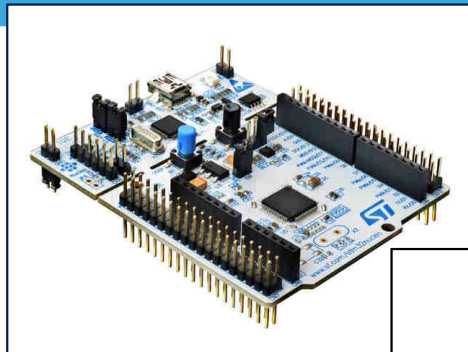
31

Same application has been developed on a low power microcontroller and on Motion Learning Processing

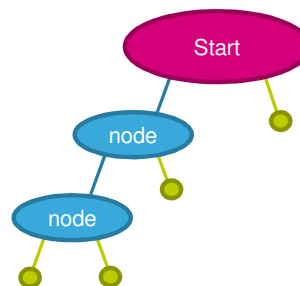
Activity Recognition Algorithm Classes

- Stationary
- Walking
- Fast Walking
- Jogging

...



MLP



Additional Current Consumption Required to Run the Application

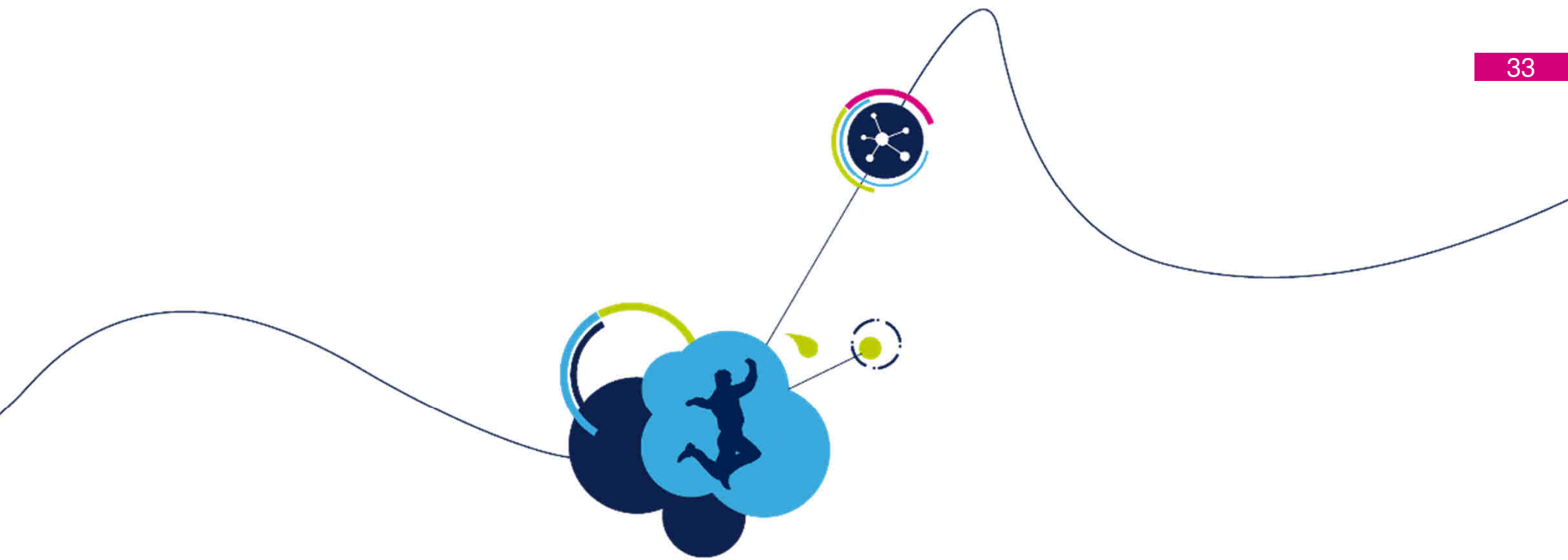
Running on:	[μ A]
Cortex-M3 STM32L152RE@32MHz	240
MLC on LSM6DSOX	7

LSMDSOX Performance

32

Confusion Matrix of Activity Recognition in LSM6DSOX

Detected as ->	Stationary	Walking	Fast Walking	Jogging
Stationary	99.1%	0.9%	0.0%	0.0%
Walking	0.0%	99.4%	0.2%	0.0%
Fast Walking	0.0%	3.7%	95.9%	0.2%
Jogging	0.0%	0.6%	0.7%	98.5%

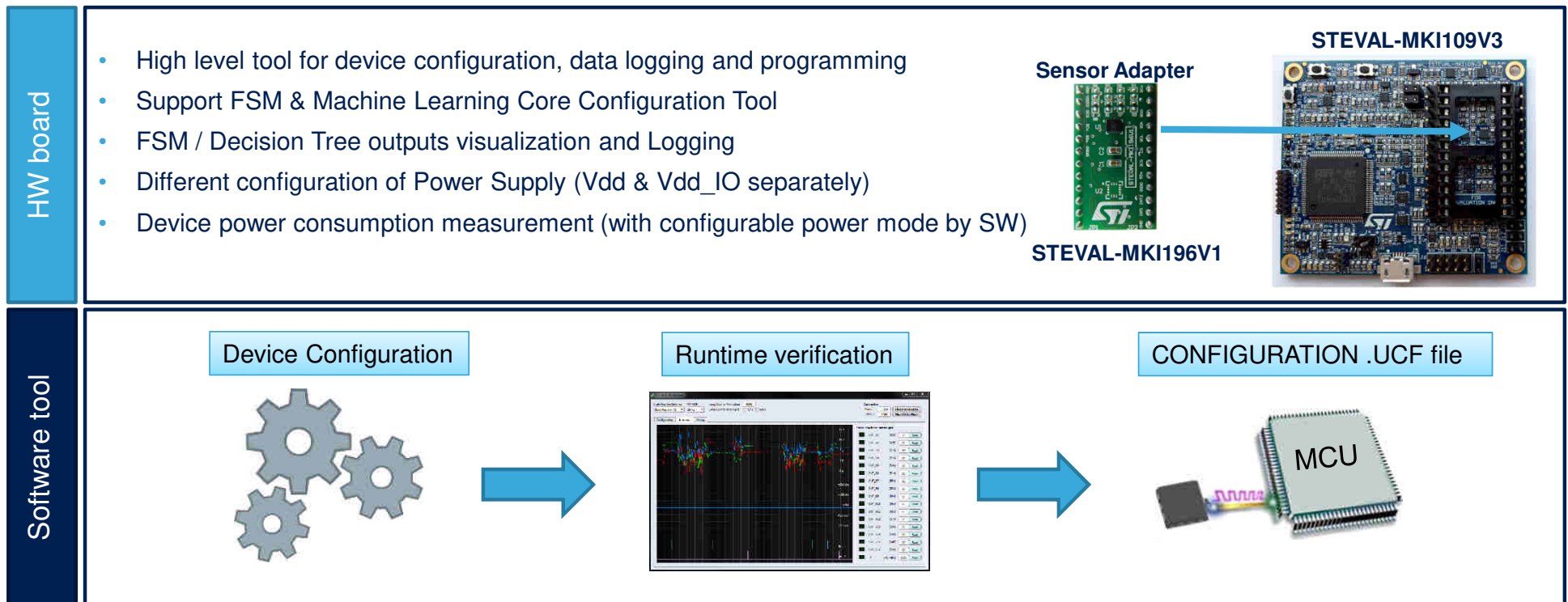


Tools

ST MEMS sensor evaluation kit

34

Professional Unico software Tool and ProfiMEMS



ST Unico GUI software

35

- Unico Lite GUI (or STSW-MEMS034) is an user interface that assists users to graphically and easily:
 - configure registers,
 - see the sensor's behavior on a chart,
 - export settings in a header file to include in the source code of an application.

1

Registers configuration



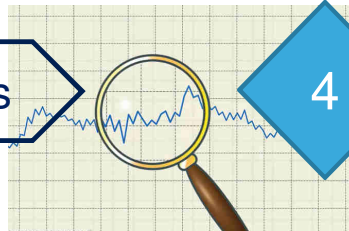
2

Data Collection



3

Pattern analysis



4

FSM Implementation



5

MLC Configuration

FSM examples

36

- ST provides FSM examples to start experimenting and test.

The screenshot shows the ST Finite State Machine (FSM) configuration tool interface. The window is titled "Finite State Machine" and has tabs for "Configuration", "Interrupt", and "Debug".

Configuration Section:

- State Machine Selection: **FSM ODR**
- State Machine #1: **26 Hz**
- Long Counter Max Value: **0000**
- Long Counter Interrupts: ☐ INT1 ☐ INT2

Converter Section:

- Float: **0.000** (Float to Float16)
- Float16: **0000** (Float16 to Float)

SM1 Instructions Section:

Address	Mode	Instruction	Value	Buttons
0x1D	RNC	SELMB	0x 77	+ Add - Remove
0x1E	RNC	STHR1	0x AA	+ Add - Remove
0x1F		THRS LSB Value	0x FA	+ Add - Remove
0x20		THRS MSB Value	0x 3A	+ Add - Remove
0x21	RNC	SINMUX	0x 23	+ Add - Remove
0x22		MUX Value	0x 01	+ Add - Remove
0x23	RNC	LINT1	0x 75	+ Add - Remove
0x24	RNC	SINMUX	0x 23	+ Add - Remove
0x25		MUX Value	0x 07	+ Add - Remove
0x26	RNC	STHR1	0x AA	+ Add - Remove
0x27		THRS LSB Value	0x 2F	+ Add - Remove

SM1 Status Section:

- ☒ Enabled ☐ INT1 ☐ INT2

SM1 Fixed Data Section:

Config A	Config B
Size: 69	Size: 20
RP: 42	RP: 00
Hysteresis: 00	Hysteresis: 00

SM1 Variable Data Section:

Thresh1 (0.872)	Thresh2	Thresh3	Thresh4
3AFA	0019	000A	05

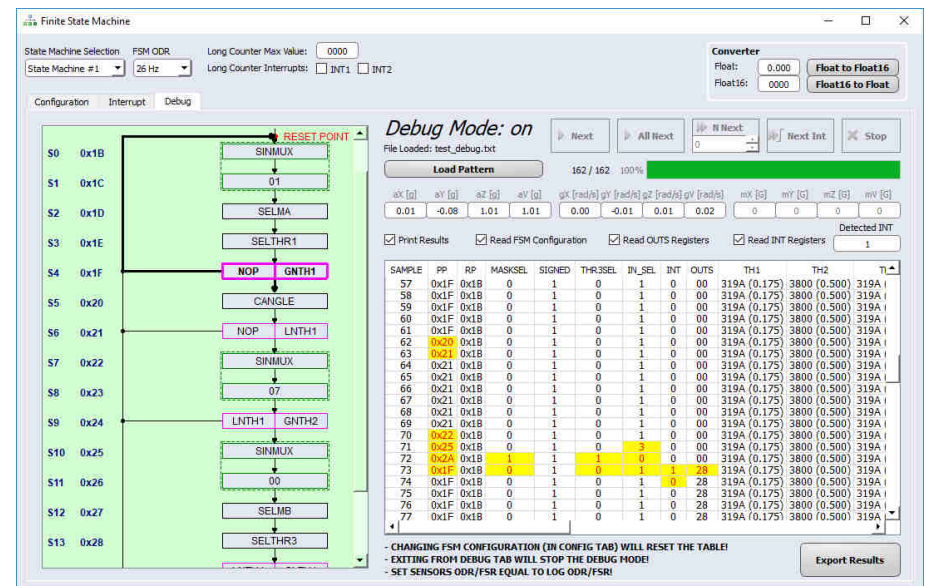
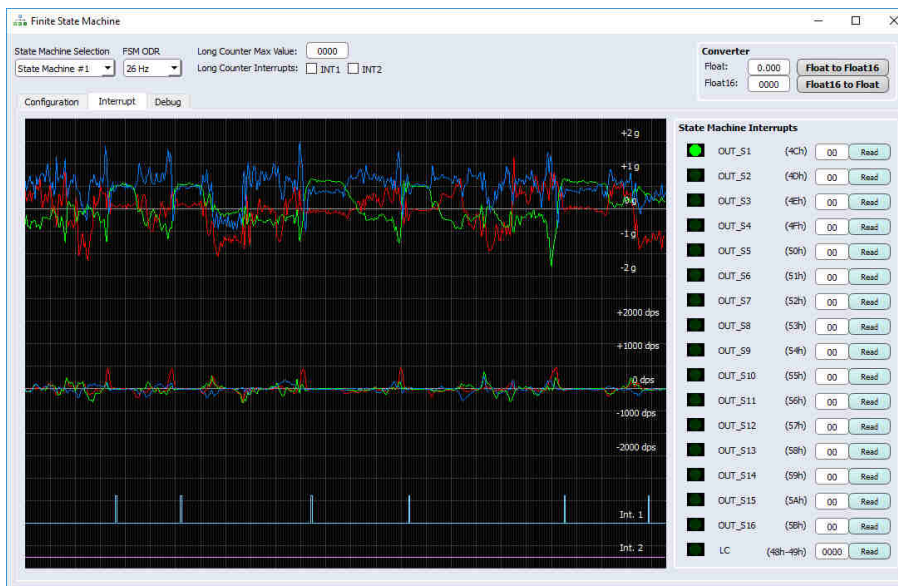
Buttons:

- Import State Machine
- Export State Machine
- Reset State Machine
- Read FSM Configuration
- Write FSM Configuration
- Reset All
- Load Device Configuration
- Save Device Configuration

Fine Tuning

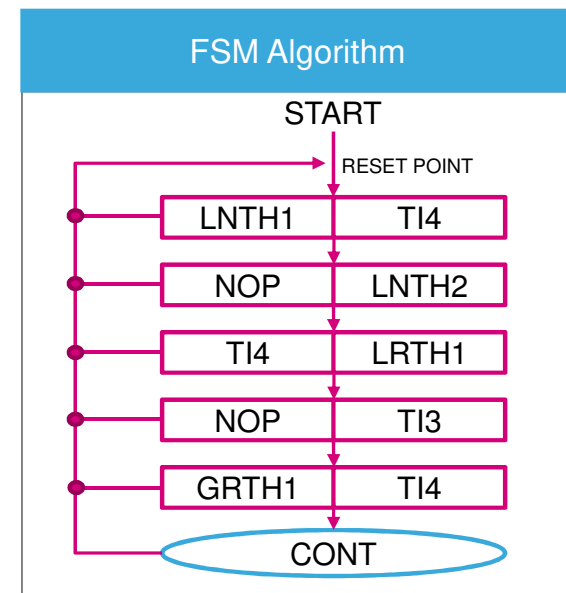
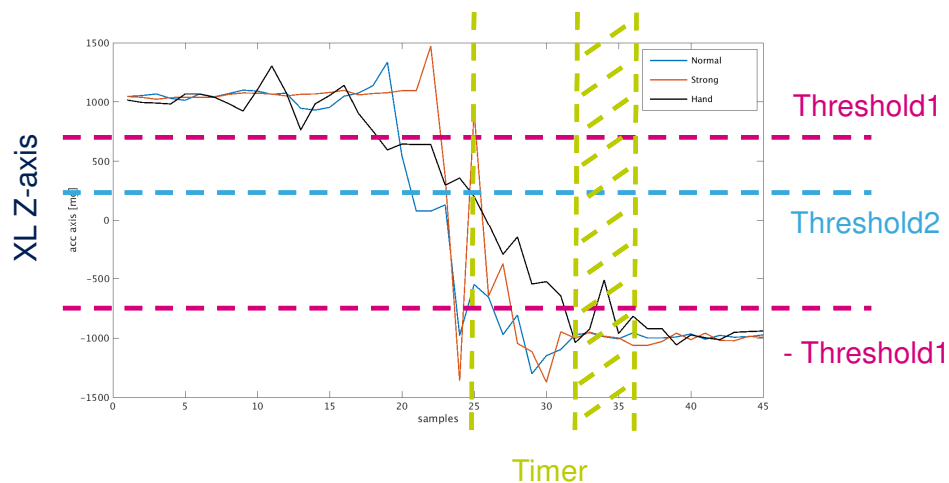
37

It is possible to debug in real time the FSM behavior playing back previously recorded data.



FSM capable of detecting when the user picks up a product

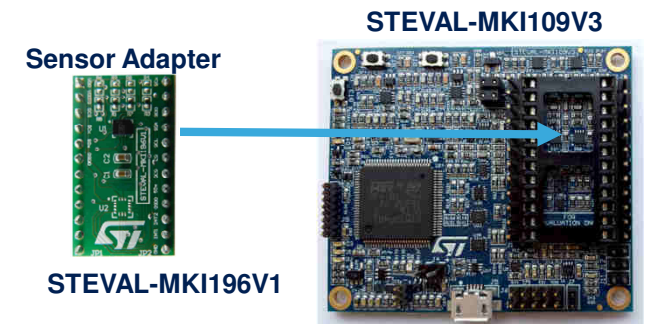
- Option 1: Programmers can load the example, write it onto the sensor, and start experimenting with it right away by changing threshold values or timers, for instance.
- Option 2: You can replay a series of events to see if the FSM would respond appropriately and throw an interrupt at the right moment.



Start Working with ProfiMEMS Board

39

- The best way to start experimenting and prototyping with the LSM6DSO/X is:
 - STEVAL-MKI196V1 daughter board.
 - STEVAL-MKI109V3 motherboard.



- Users need to plug the sensor board into the main board and connect it to a PC using the onboard USB port.
- Unico v8.0 supports FSM & MLC on Windows, Linux, Mac

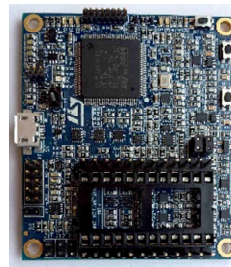
Part Number	Software Version	Marketing Status	Supplier	Download
STSW-MKI109W	8.0.0	Active	ST	Get Software

LSM6DSOX

Reference & Support

40

- LSM6DSOX ST datasheet
 - LSM6DSOX: [LSM6DSOX.html](#)
- Application Notes:
 - AN5272 (LSM6DSOX): [DM00571818.pdf](#)
 - AN5273 (LSM6DSOX Finite State Machine): [DM00572971.pdf](#)
 - AN5259 (LSM6DSOX Machine Learning Core): [DM00563460.pdf](#)
- Video Tutorial
 - Getting Started with LSM6DSOX: [LSM6DSOX.html](#)
- Boards:
 - Professional MEMS tool board - STEVAL-MKI109V3: [mems-motion-sensor-eval-boards/steval-mki109v3.html](#)
 - LSM6DSOX Adapter board - STEVAL-MKI197V1: [mems-motion-sensor-eval-boards/steval-mki197v1.html](#)
- Unico GUI package v8.0 - SW package for boards
 - STSW-MKI109W for Windows OS: [evaluation-tool-software/stsw-mki109w.html](#)
 - STSW-MKI109L for Linux OS: [evaluation-tool-software/stsw-mki109l.html](#)
 - STSW-MKI109M for Mac OS: [evaluation-tool-software/stsw-mki109m.html](#)
- MEMS Community
 - Community: [community.st.com/s/group/CollaborationGroup](#)
 - Q&A: [mems-and-sensors](#)



Takeaways

41

- LSM6DSO/X completely transforms what a sensor can process without the help of a host microcontroller (MCU), thanks to the unique Machine Learning Core.
- Exceptional advantage in terms of Power consumption and Flexibility.
- ST is supporting your projects by reference design, dedicated tools and example libraries.

Thank You!



Thank you

42

ST stands for
life.augmented

Everywhere
microelectronics make a
positive contribution to
people's lives, ST is
there