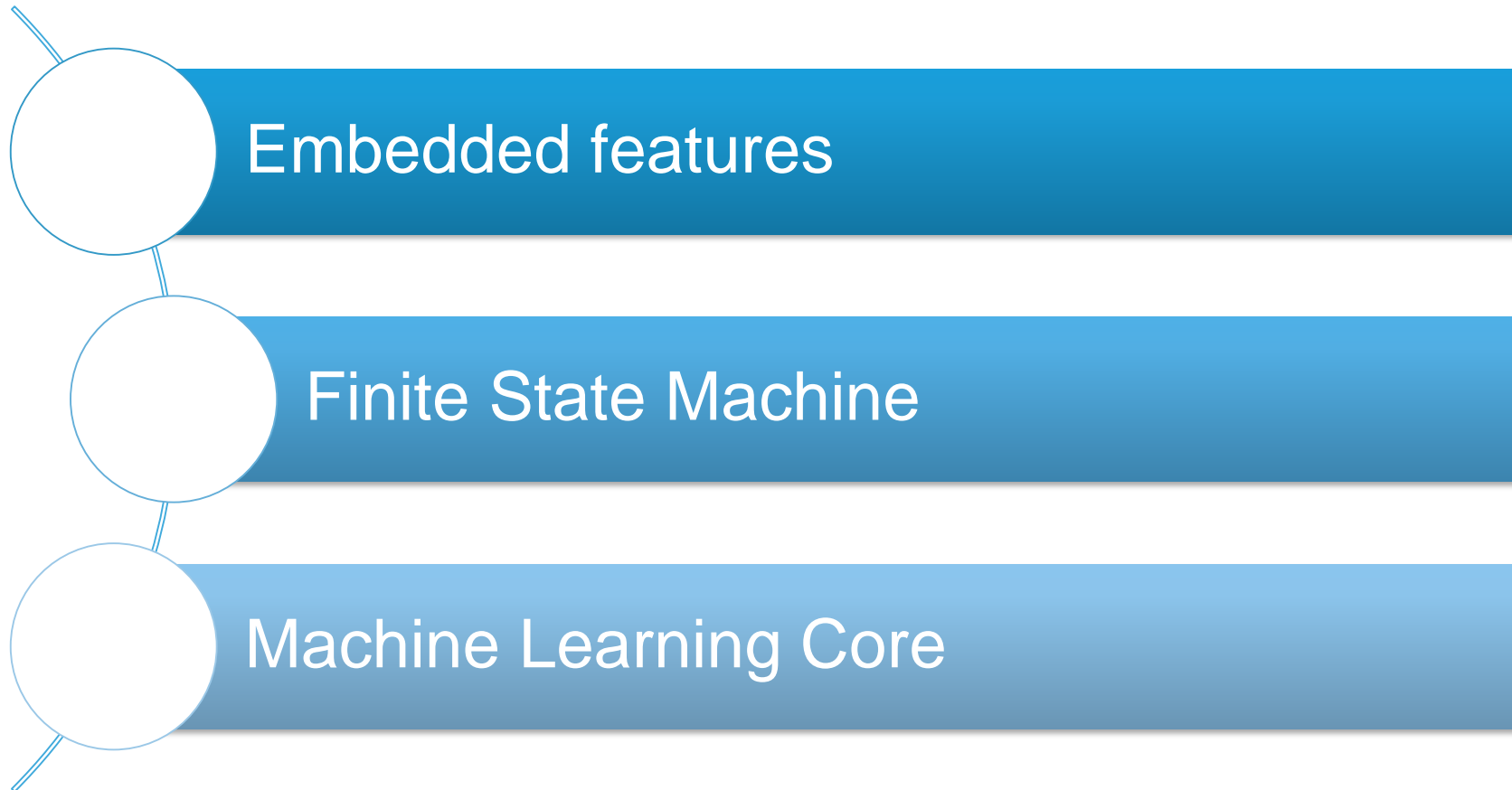


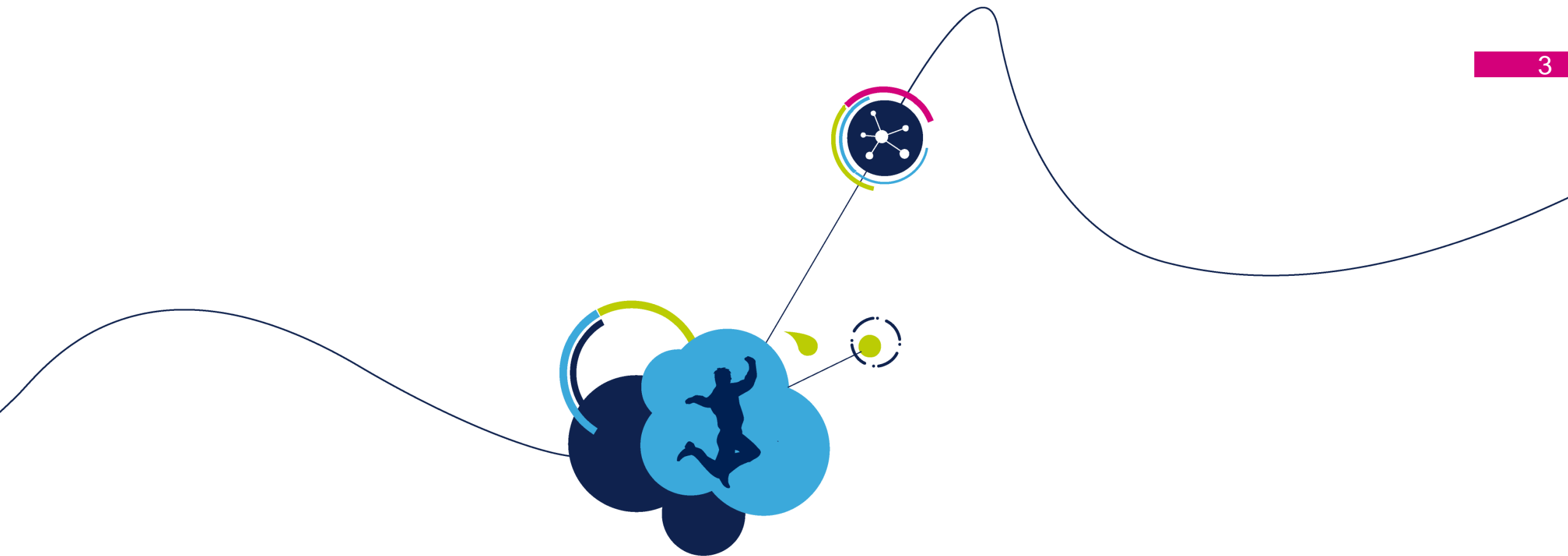
Features embedded in ST MEMS sensors

ST MEMS workshop, Italy

November 2019





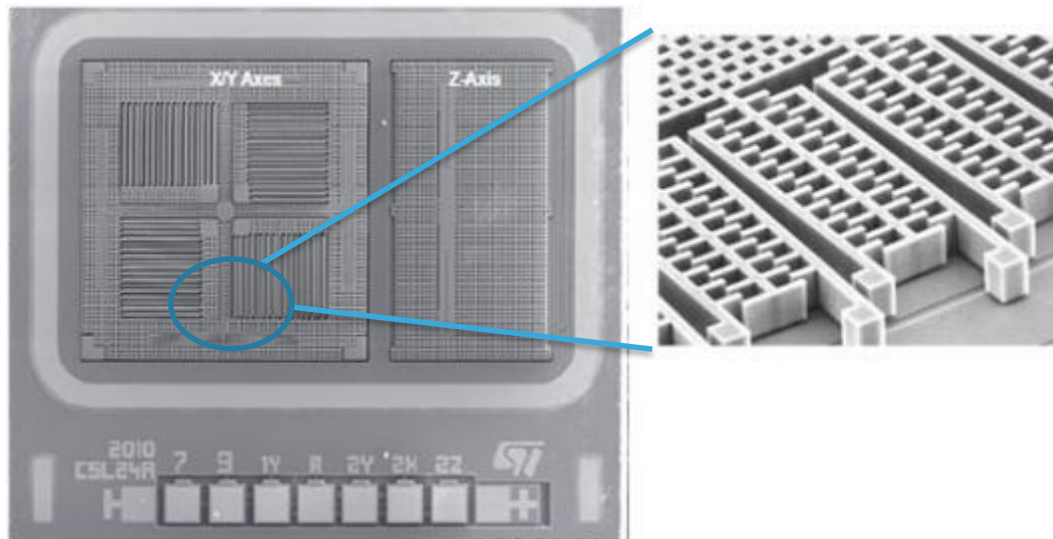


Embedded features

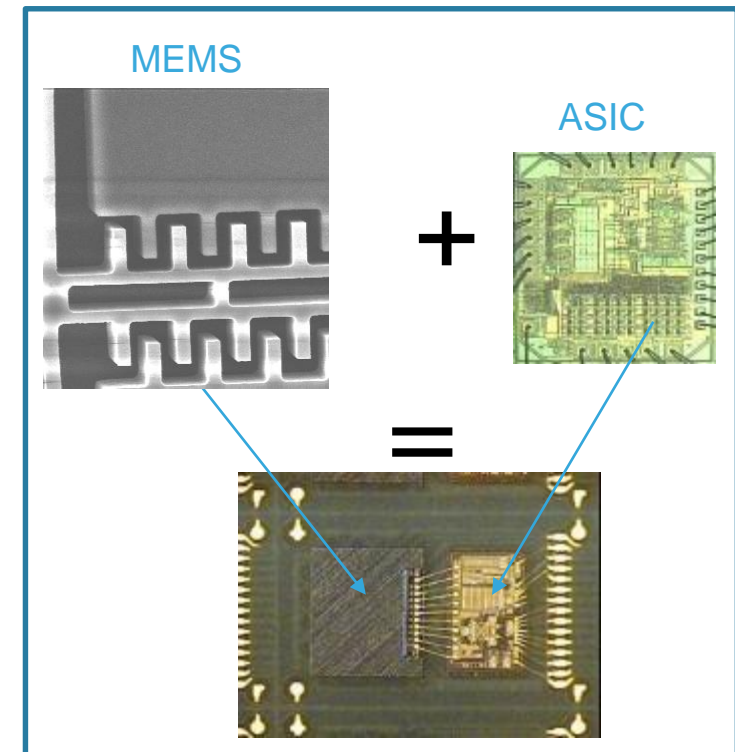


What is a MEMS?

- MEMS means **M**icro **E**lectro **M**echanical **S**ystem MEMS contain movable 3-D structures
- The Structure moves accordingly to external displacement
- In MEMS not only electrons are moving!

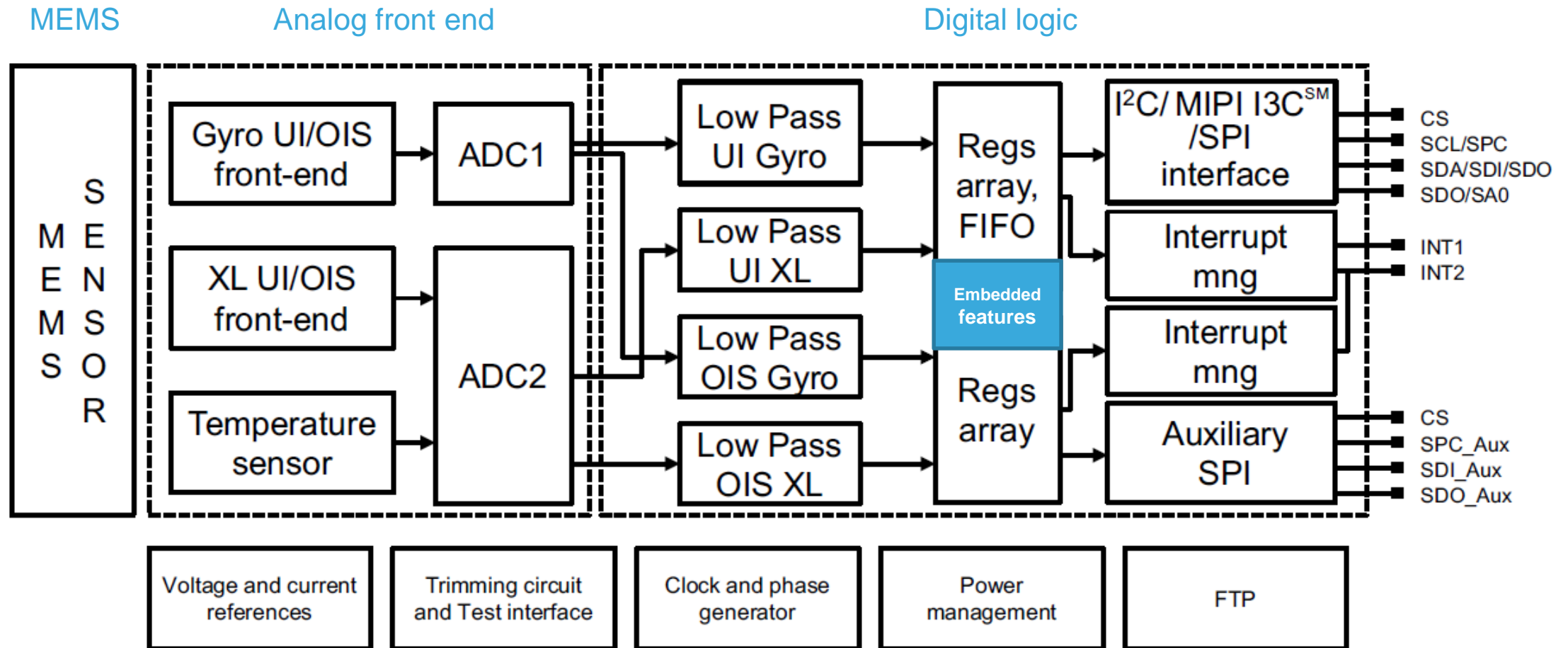


Scanning Electron Microscope pictures of a capacitive micro-machined structure manufactured with THELMA process



Typical MEMS Sensor Block Diagram

5



LSM6DSO block diagram



Movement recognition features embedded in LSM6DSx

6

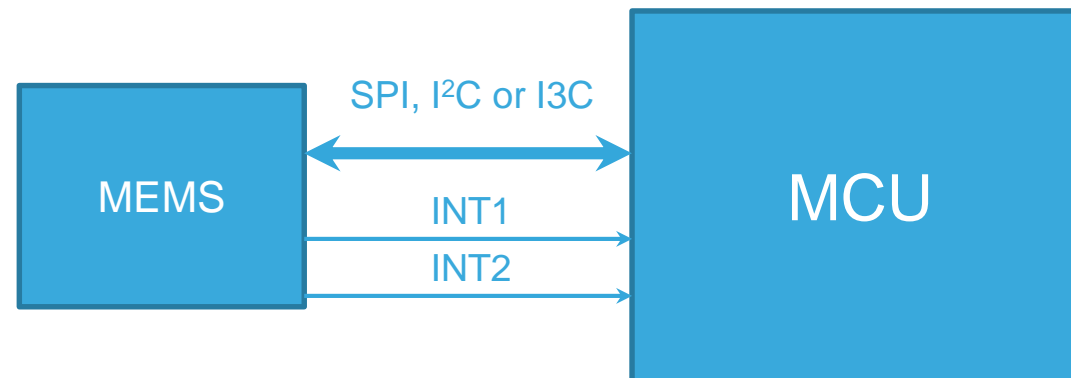
- Basic (accelerometer based)
 - Wake up (threshold recognition)
 - Free-fall
 - 6D orientation
 - Single/Double tap
 - Activity/Inactivity (threshold + ODR switching)
- Advanced (accelerometer based)
 - Pedometer
 - Significant motion
 - Relative tilt
- Enhanced (accelerometer, gyroscope, external sensor based)
 - Finite state machine
 - Machine learning core



Schematic considerations

7

- Power supply
 - Separated Vdd (for measurement chain) and Vdd_IO lines (for IO pins)
 - Supply voltage range: depends on device
 - LIS2DH12: 1.71 to 3.6V for both, LSM6DSO: 1.71-3.6V for Vdd, 1.62-3.6V for VDD_IO
- Sensors are capable to communicate over **SPI and I²C** digital serial interfaces, but always only one at a time. New devices support also **I3C** interface.
- Device setup and data acquisition is done by **accessing registers** of the device
- INT1 and INT2 interrupts pins have programmable functionality

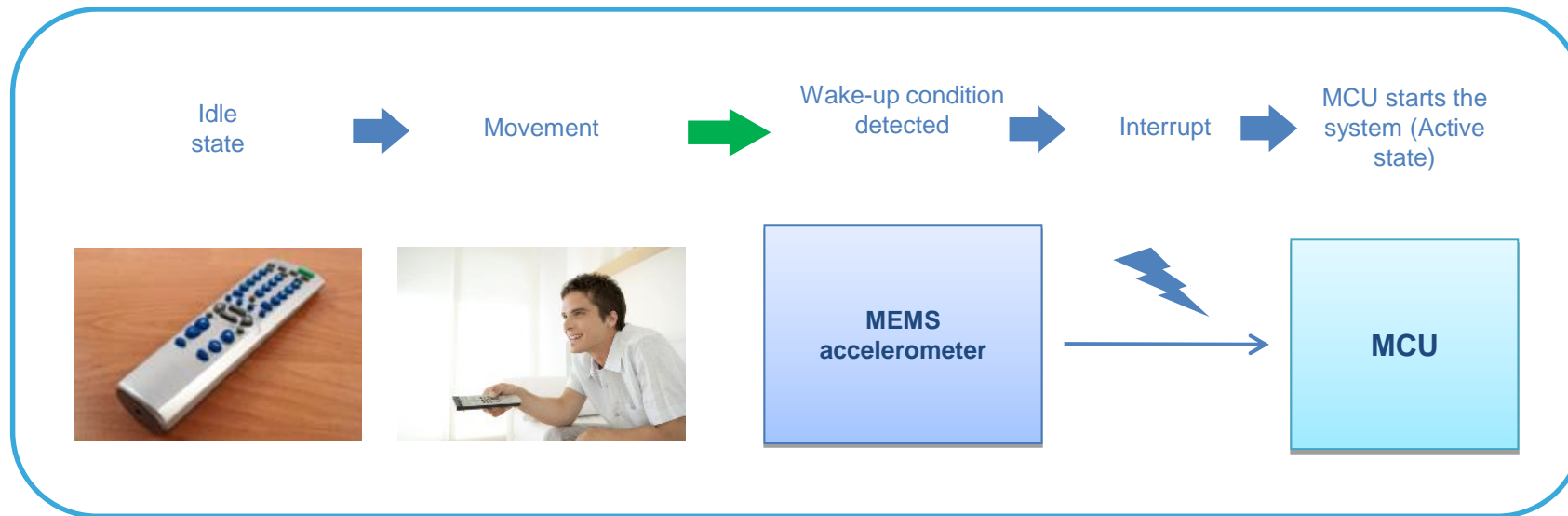




Wakeup Description

8

- Detects when the device changes its state from a stable position to a movement

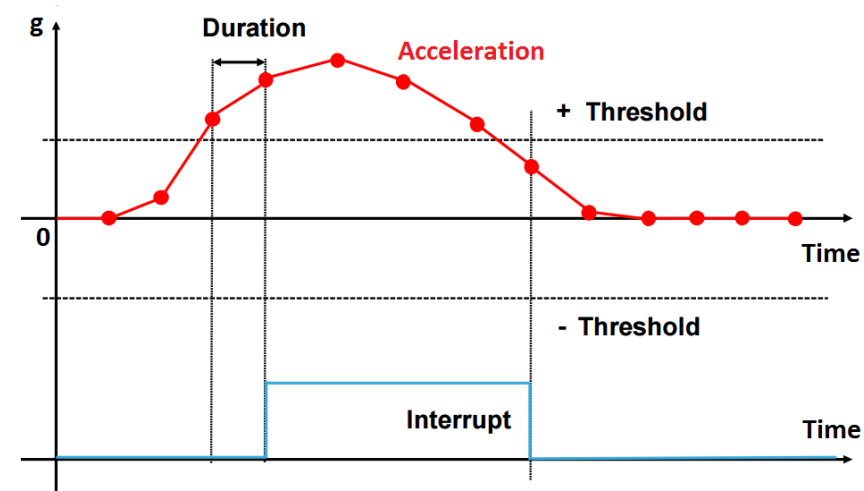
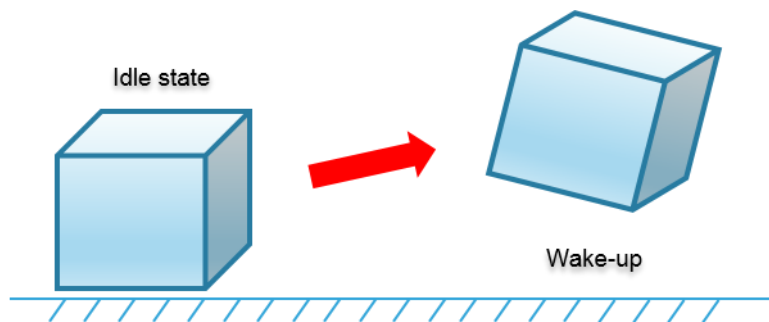




Wakeup Parameters

9

- Threshold
 - defines amplitude which at least one of the axis has to exceed to be assumed as wakeup
- Duration
 - defines minimum time period during which acceleration must exceeds the threshold to generate wakeup interrupt

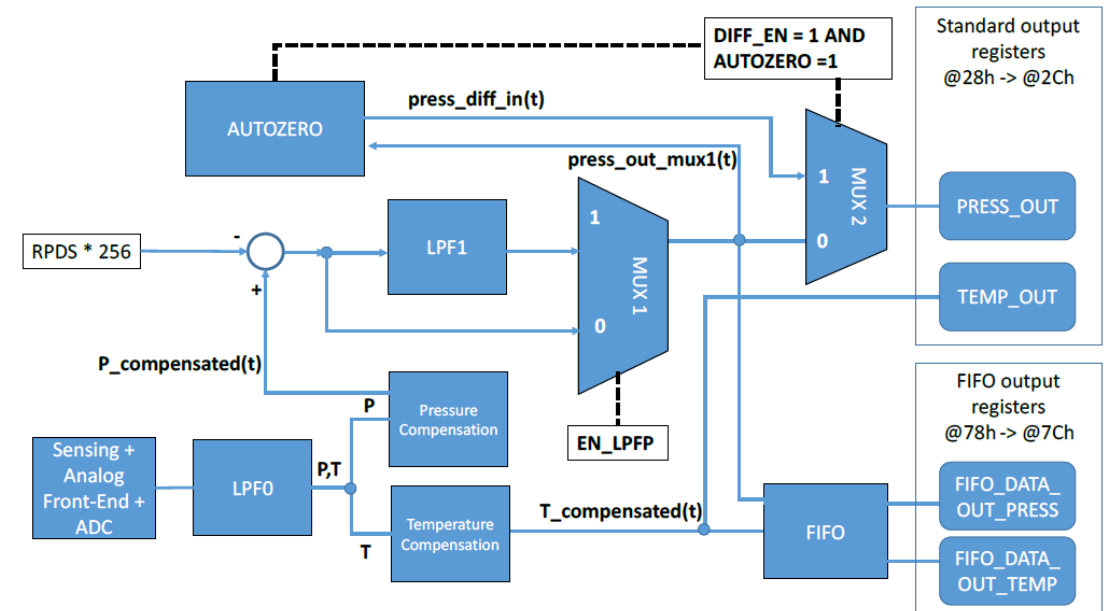


Pressure sensors

Output registers

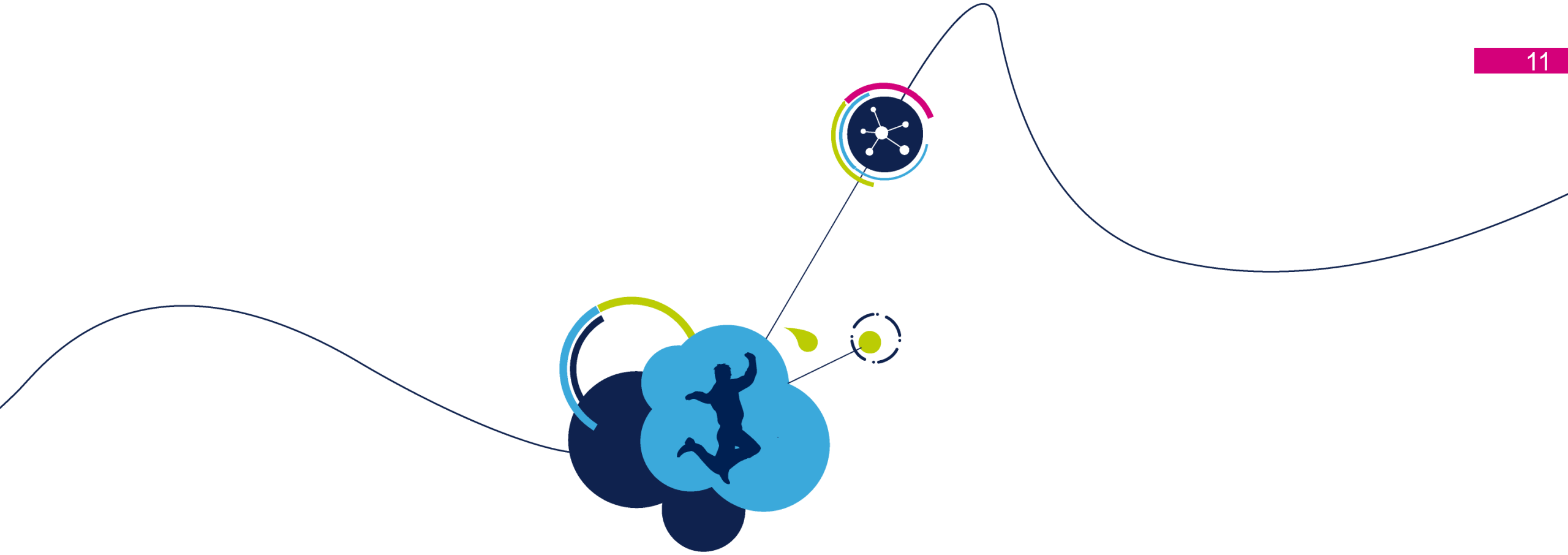
10

Name	Address	Description
PRESS_OUT	2Ah & 29h & 28h	24-bit absolute pressure data or this contains the difference in pressure between REF_P and PRESS_OUT when auto-zero mode is used
TEMP_OUT	2Ch & 2Bh	16-bit temperature data
REF_P	16h & 15h	16-bit pressure data subtracted from sensor output measurement in auto-zero mode
RPDS	19h & 18h	16-bit pressure offset value can be used to implement One-Point-Calibration (OPC) after soldering the device



LPS22HH data path





Finite State Machine and Machine learning core



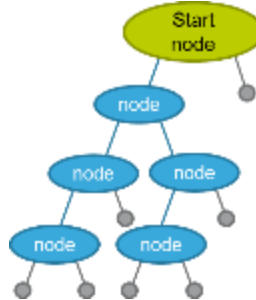
When using MLC or FSM

12

Activity Tracking with Machine Learning Core

Programmable logic able to identify if a **pattern** matches an **activity** in a user defined set of classes: e.g. running, walking, driving, etc...

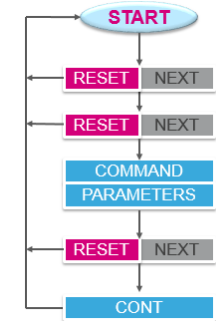
- ✓ Decision Tree logic embedded
- ✓ Up to 8 decision trees can be configured to run simultaneously
- ✓ Machine Learning approach



Gesture Recognition with Finite State Machine

The **Finite State Machine** is a series of state parameters and variables that allows the **running of low power programs**

- ✓ Run up to 16 independent programs simultaneously
- ✓ FSM can process Axl, Gyro data and external sensor like Magnetometer
- ✓ Interface with MLC



Main Differences

Labeled Sensor data with features	INPUT	Sensor Samples data
Machine Learning based logic	LOGIC	Events/Triggers based logic using thresholds / timers
Pattern classification using a Decision Tree	OUTPUT	Event detection using commands and conditions

MLC → Motion patterns detection
(Activity classes)

FSM → Well defined gestures / movements



Finite State Machine (FSM)

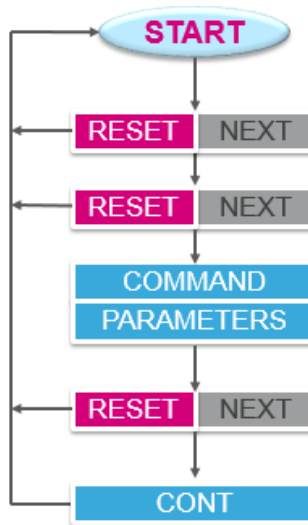
13

Innovative Embedded
Solution

Inertial Sensor
LSM6DSO / DSR

FSM

Up to 16



1x FSM

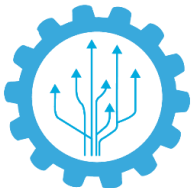
Each FSM is intended to detect:
single specific gesture

- Wrist Tilt
- Free Fall
- Pick Up
- Wake-Up
- Shake / Jiggle
- Glance
- Motion /Stationary
- Etc...

Pros

- ✓ Ultra Low Power
- ✓ Parallel processing of many Algos
- ✓ Lower interaction with MCU
- ✓ Interrupts
- ✓ Flexible Configurability
- ✓ Evaluation of multiple sensors





Machine Learning Core (MLC)

14

Enhanced Innovative
Embedded Solution

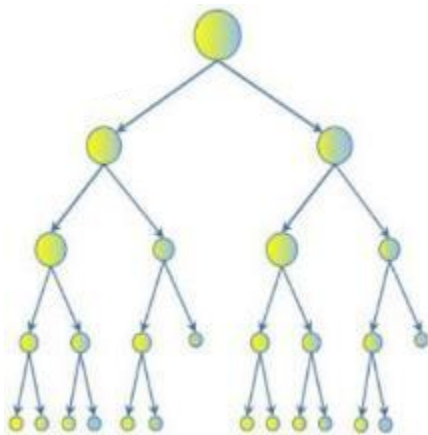
Inertial Sensor
LSM6DSOX / DSRX

FSM

Up to 16

MLC

Up to 8



1x Decision tree

Each Application is intended to detect:
User contexts

- Activity recognition
- Carry position
- Fitness activities
- Context awareness
- Motion intensity
- False positive rejection
- Vibration intensity
- Etc...

Pros

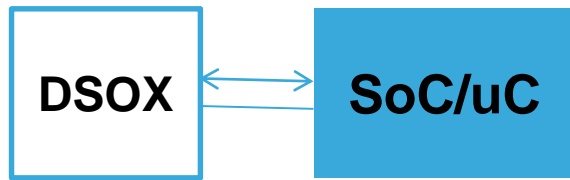
- ✓ Those of FSM, plus
- ✓ Handles complex algorithms
- ✓ Dramatically decreases MCU load and current consumption
- ✓ BOM reduction (no DSP needed)



System Level Benefits using LSMDSOX

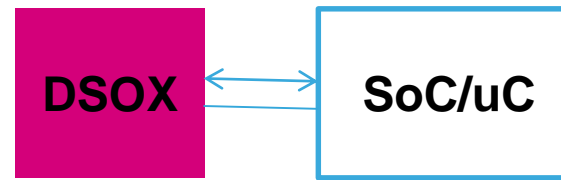
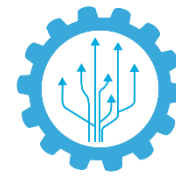
15

Activity Recognition in SW



Real Time Data

Activity Recognition in DSOX



Interrupt

Parameter

Benefit in using MLC

DSOX current



Similar DSOX consumption

MIPS uC/SoC



Offload SoC / uC

Traffic



Less traffic on Bus

Battery usage



Reduction of System current



Current consumption

- FSM and MLC are incredibly efficient in current consumption needs
 - ~ 3 μA for each Finite State Machine (@ODR = 12.5Hz)
 - 1 μA to 15 μA for Machine Learning Core

MLC algorithm	Sensor used	ODR	Number of decision trees	Number of nodes	MLC current consumption
Vibration Monitoring	Accelerometer	26 Hz	1	2	1 μA
Motion Intensity	Accelerometer	12.5 Hz	1	7	1 μA
6D orientation recognition	Accelerometer	26 Hz	1	8	2 μA
Activity recognition for mobile	Accelerometer	26 Hz	1	126	4 μA



Current consumption improvement

17

10 to 100 times better current consumption with MLC!

Activity recognition library (MotionAR)
running **in SW in MCU**

LSM6DSOX Sensor	Sensor Current consumption
Core	15 μ A
MLC – not used	0 μ A

MCU	Wake-up rate	MCU Current consumption
STM32F401RE	1/16 = 63ms	91 μ A
STM32L152RE	1/16 = 63ms	82 μ A
STM32L476RG	1/16 = 63ms	51 μ A

Activity recognition algorithm
running **inside LSM6DSOX**

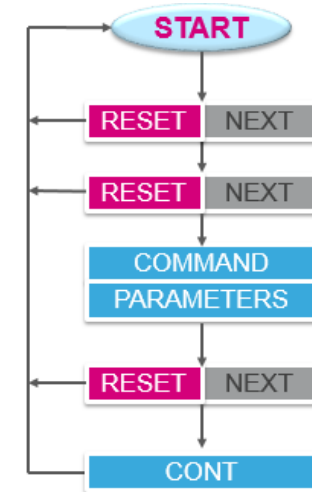
LSM6DSOX Sensor	Sensor Current consumption
Core	15 μ A
MLC	4 μ A

MCU	Wake-up rate	MCU Current consumption
STM32F401RE	1 s	9.27 μ A
	30 s	3.02 μ A
	100 s	2.8 μ A
STM32L152RE	1 s	3.24 μ A
	30 s	1.46 μ A
	100 s	1.4 μ A
STM32L476RG	1 s	2.8 μ A
	30 s	0.65 μ A
	100 s	0.59 μ A

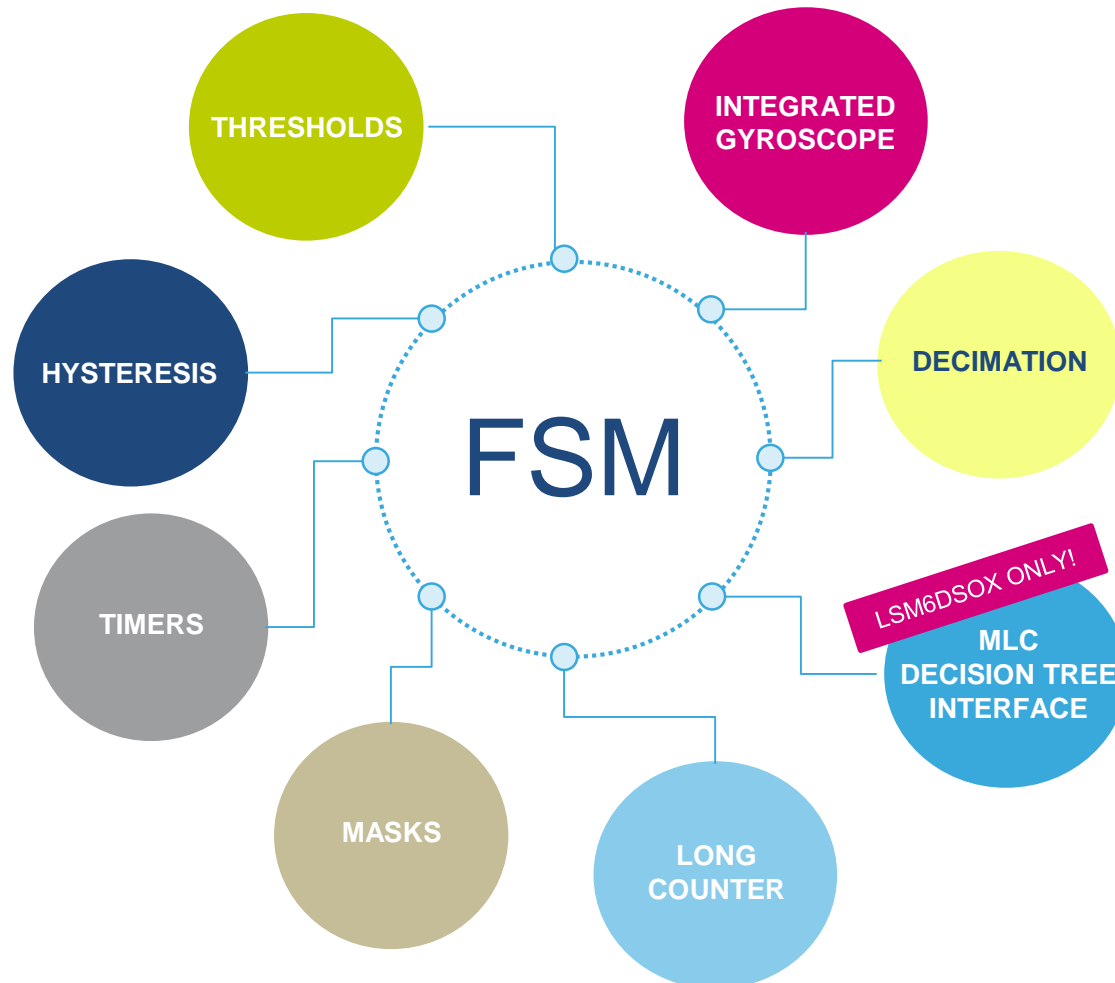




Finite State Machine



Resources allow FSM to analyze input data.



Resources

- Up to 3 different thresholds
- Hysteresis value
- Up to 4 different timers
- Up to 3 different masks
 - Axis selection in runtime
- Long Counter
 - Count events
- Decimation factor
- Input selector
 - XL
 - G
 - Mag
 - Etc...
- MLC Decision Tree interface



Easy FSM programs development, from algorithm definition to validation!

The image displays three overlapping windows of the FSM Tool software, illustrating the workflow from configuration to validation.

Top Window (Configuration): Shows the 'Finite State Machine' configuration. It includes a 'State Machine Selection' dropdown set to 'FSM ODR', a 'Long Counter Max Value' of 0000, and a 'Converter' section with 'Float' set to 0.175 and 'Float16' set to 319A. The 'SM1 Instructions Section' lists 10 instructions (S0 to S9) with their respective commands (RNC, CMD) and values. The 'SM1 Status' section shows 'Enabled' and 'INT1' selected. The 'SM1 Fixed Data Section' shows 'Config A' and 'Config B' values.

Middle Window (Status): Shows the 'Finite State Machine' status. It includes a 'State Machine Selection' dropdown set to 'FSM ODR', a 'Long Counter Max Value' of 0000, and a 'Converter' section with 'Float' set to 0.000 and 'Float16' set to 0000. The 'State Machine Interrupts' section shows 'OUT_S1', 'OUT_S2', and 'OUT_S3' with their respective values and 'Read' buttons.

Bottom Window (Debug Mode): Shows the 'Finite State Machine' in 'Debug Mode: on'. It includes a 'File Loaded: test_debug.txt' and a 'Load Pattern' button. The 'Print Results' section shows a table of results for 162 samples. The 'Detected INT' section shows '1'. The 'Export Results' button is visible.

Left Window (Waveform): Shows a waveform plot with multiple channels (blue, green, red, black) over time. The plot is labeled with 'S0 0x1B', 'S1 0x1C', 'S2 0x1D', 'S3 0x1E', 'S4 0x1F', 'S5 0x20', 'S6 0x21', 'S7 0x22', 'S8 0x23', 'S9 0x24', 'S10 0x25', 'S11 0x26', 'S12 0x27', and 'S13 0x28'.

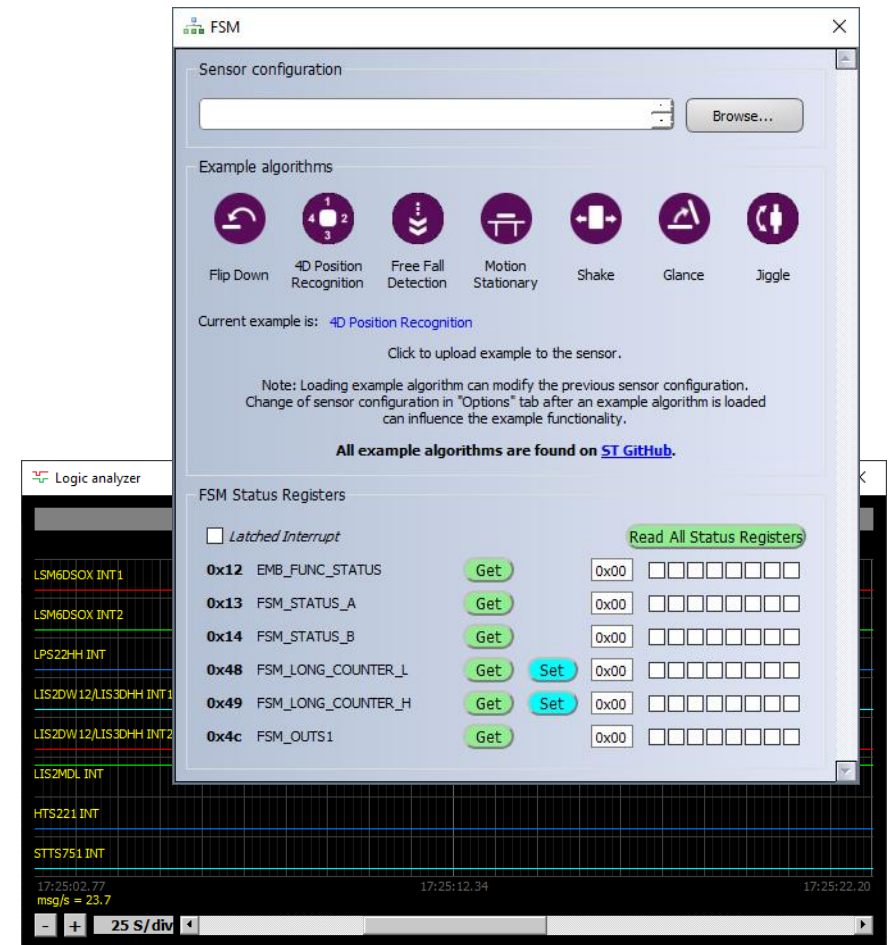


Finite State Machine Tools

21

- Application note [AN5273](#)
- Development / Debugging, Demonstration
 - [Unico](#)
 - offline mode option for development
 - STEVAL-MKI109V3 + LSM6DSO(X) / LSM6DSR(X) adapter
- Demonstration:
 - [Unicleo-GUI](#)
 - STM32Nucleo + X-NUCLEO-IKS01A3 (no adapter needed)
- Gestures database
 - Available on Github:
github.com/STMicroelectronics/STMems_Finite_State_Machine

FSM examples in Unicleo-GUI

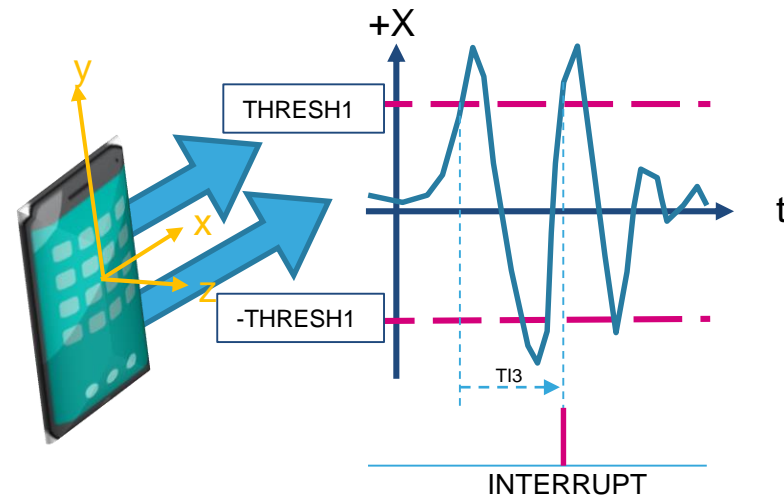
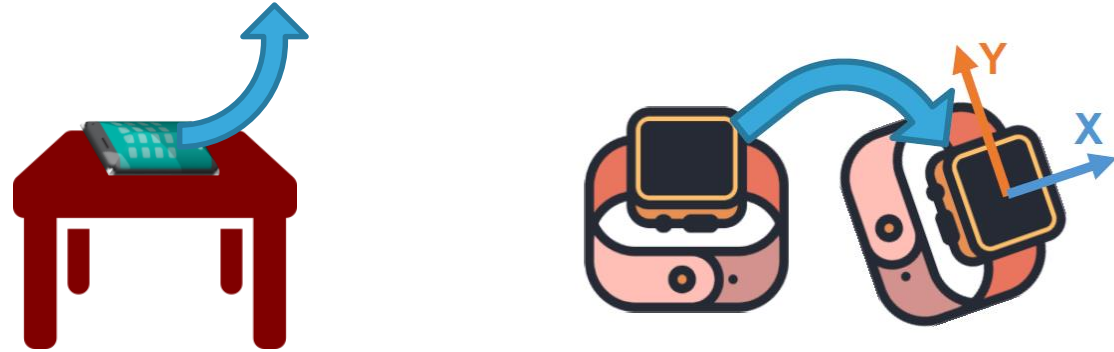


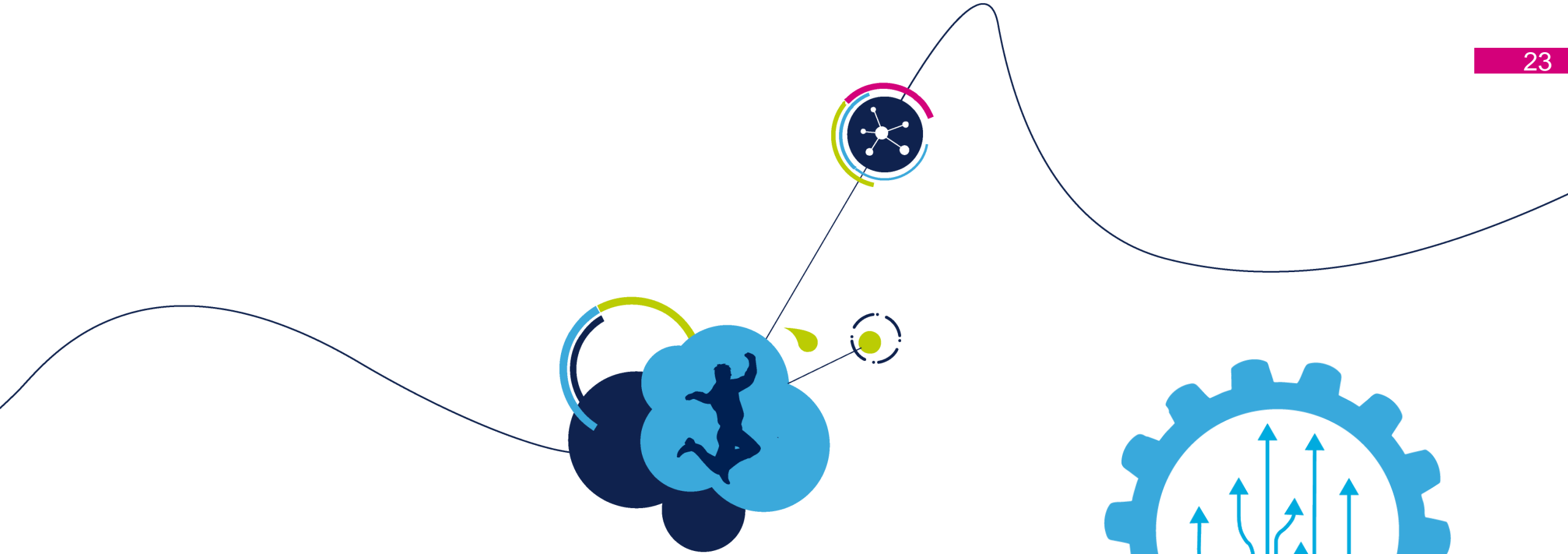
Finite State Machine

Gestures Database

22

- Wrist-Tilt
- Free Fall / Shock Detection
- Pick-Up
- Wake-Up
- Twist
- Jiggle / Shake
- Glance
- Motion / Stationary
- 4D / 6D
- Flip-Up / Down
- ... and others!





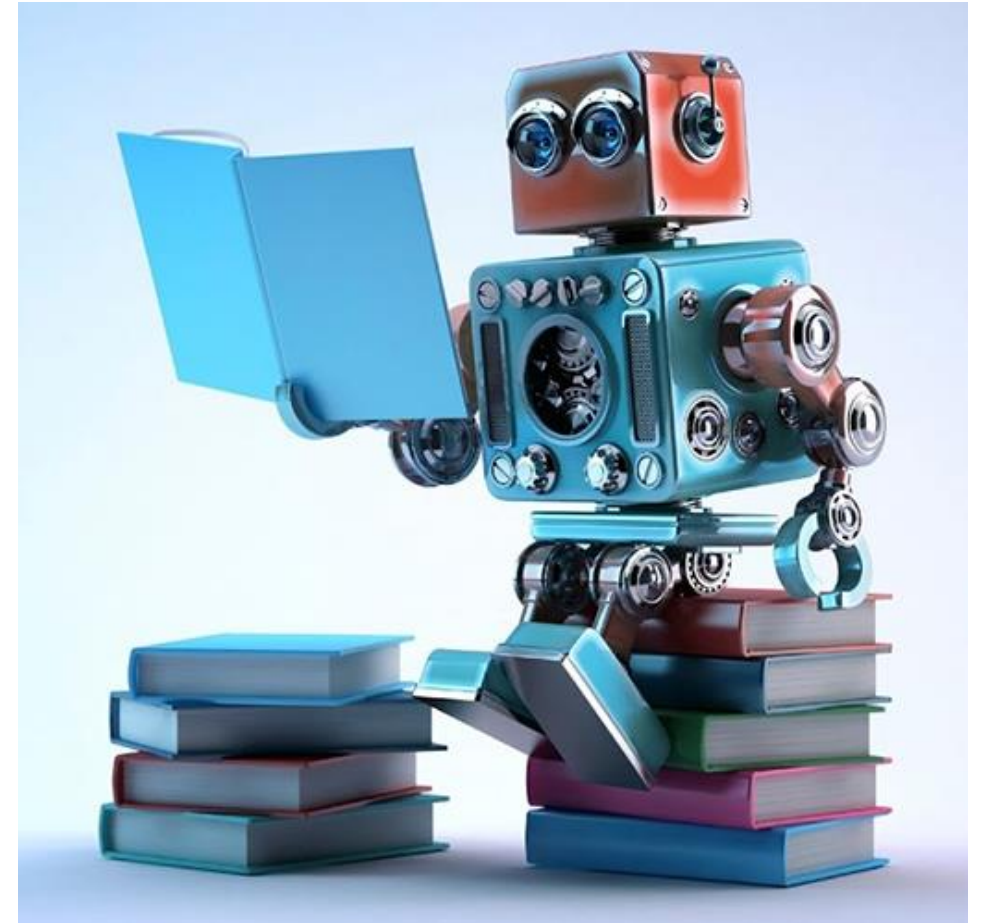
Machine Learning Core



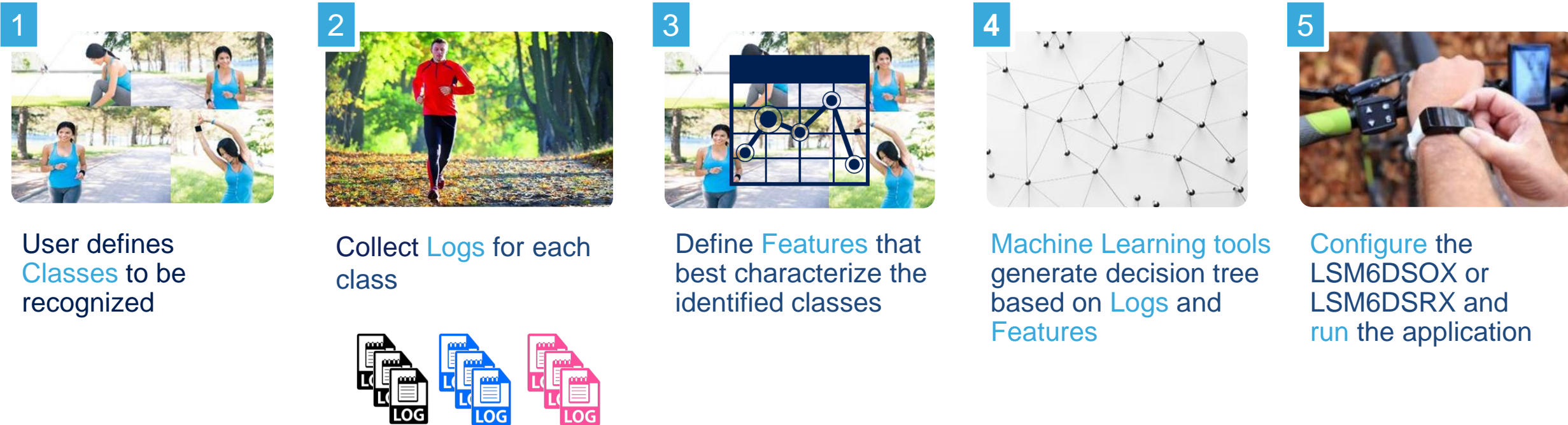
Machine Learning (ML)

24

- ML is a sub-branch of Artificial Intelligence
- The field of computer science that gives computers the ability to learn without being explicitly programmed.
- Consists of algorithms that can learn and make predictions on data:
 - Such algorithms train on previous examples to build and estimate the models;
 - Machine learning is usually employed where traditional programming is infeasible;
 - If trained properly should work for new cases.

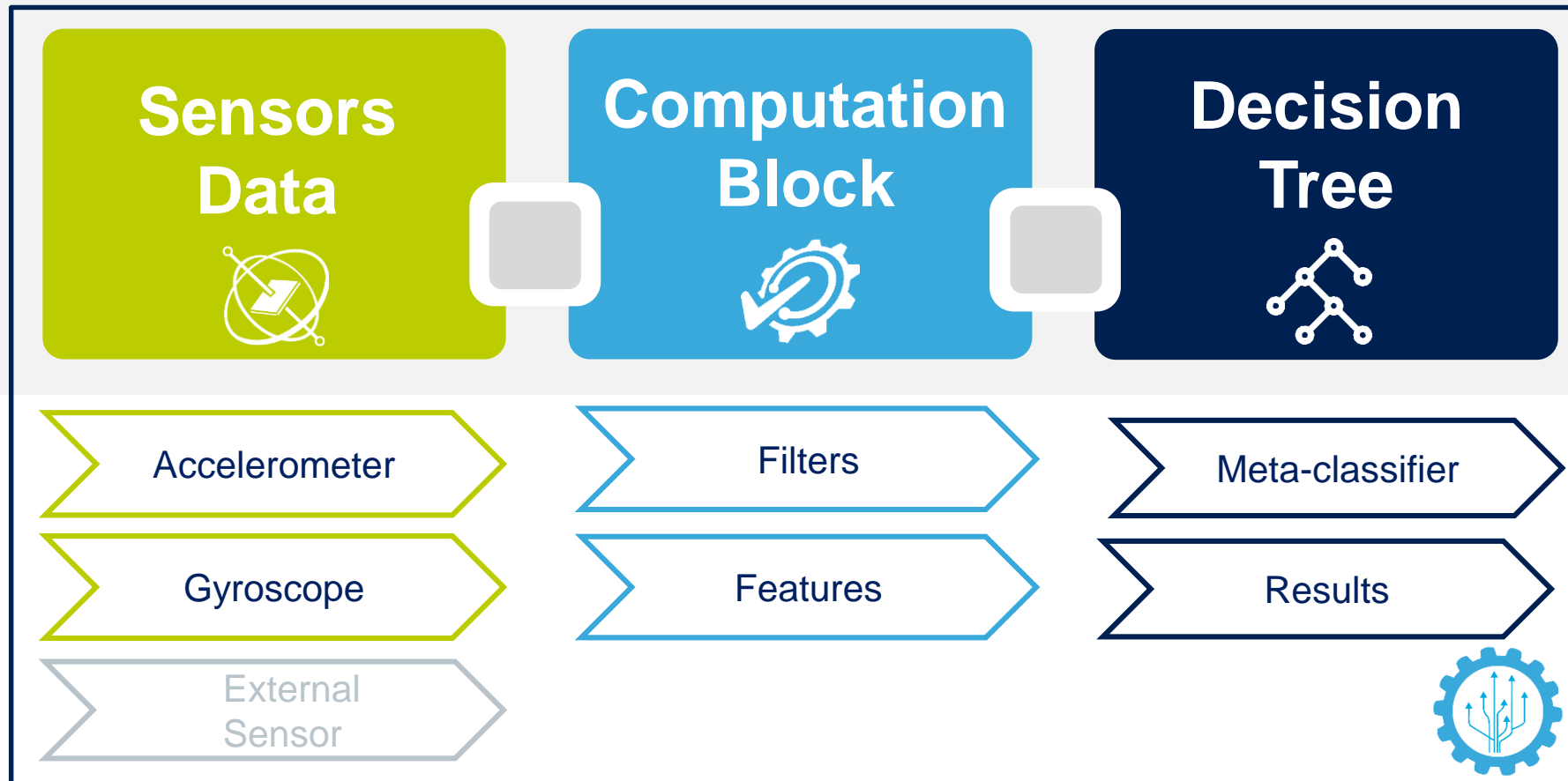


MLC uses Supervised Machine Learning



5 simple steps for 10 to 1,000 times power saving

Three main blocks

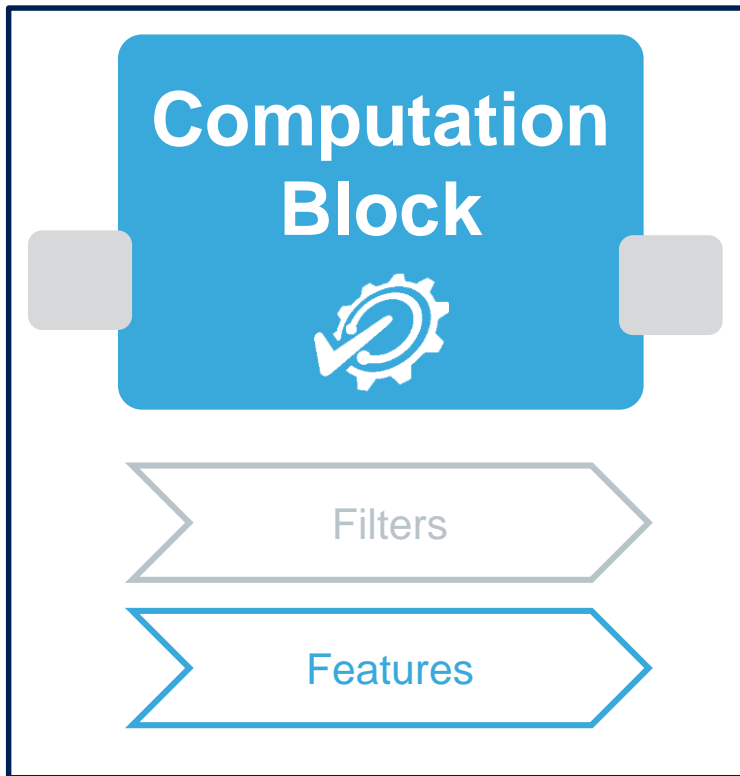
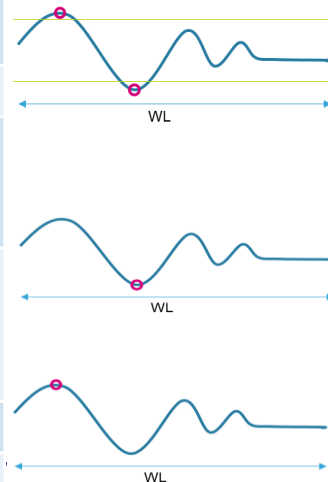


Features computation every WL samples

- Features:** statistical parameters calculated on

- Input data (e.g. *Acc_X*, *Acc_Y*, *Acc_Z*, *Acc_V*, *Acc_V²*, *Gyro_X*, *Gyro_Y*, etc...)
- Input data filtered (e.g. *high-pass on Acc_Z*, *band-pass on Acc_V²*, etc...)
- Note: the window for features computation (WL) is configurable (from 1 to 255 samples). It is not a moving window*

Feature Name	Feature Description
MEAN	Computes the average of the selected input in the defined time window $Mean = \frac{1}{WL} \sum_{k=0}^{WL-1} I_k$
VARIANCE	Computes the variance of the selected input in the defined time window $Variance = \left[\left(\frac{\sum_{WL} i^2}{WL} \right) - \left(\frac{\sum_{WL} i}{WL} \right)^2 \right]$
ENERGY	Computes the energy of the selected input in the defined time window $Energy = \sum_{k=0}^{WL} input^2$
PEAK TO PEAK	Computes the maximum peak to peak value of the selected input in the defined time window
ZERO CROSSING	Computes the number of times the selected input crosses a certain threshold . This internal threshold is defined as the sum between the average value computed in the previous window (feature "Mean") and an hysteresis defined by the user. This feature is also available as: <ul style="list-style-type: none"> - Positive zero crossing → Only transitions with positive slope - Negative zero crossing → Only transitions with negative slope
PEAK DETECTOR	Counts the number of peaks of the selected input in the defined time window. A threshold has to be defined by the user for this feature, and a buffer of three values is considered for the evaluation. If the second value of the three values buffer is higher (or lower) than the other two values of a selected threshold, the number of peaks is increased. This feature is also available as: <ul style="list-style-type: none"> - Positive peak detector → Only positive peaks - Negative peak detector → Only negative peaks
MINIMUM	Minimal value of the selected input in the defined time window
MAXIMUM	Maximum value of the selected input in the defined time window



Decision Tree: A predictive model built from the training data

Decision Tree

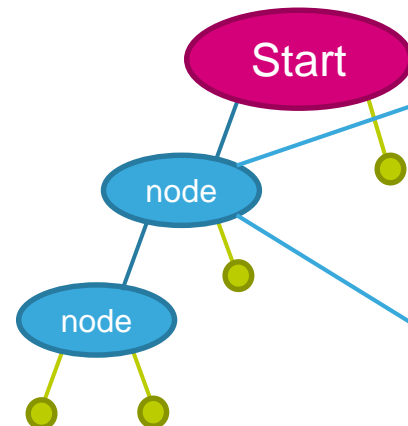


Meta-classifier

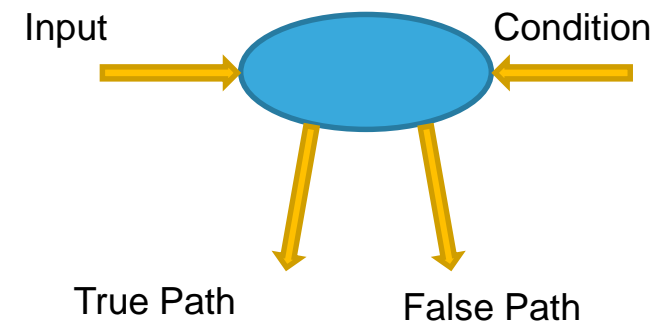
Results

- The outputs of the computation blocks (filters and features) are the inputs of the decision tree
- The decision tree is composed by nodes, it is a binary tree
- Each node of the decision tree contains a condition (if / else)
 - Some examples of conditions:
 - Mean on Acc_X < 0.5 g
 - Variance on Gyro_Z < 200 dps
- The decision tree generates a new result every WL (window for features computation)
- The decision tree is automatically built by a dedicated ML tool

Tree Example



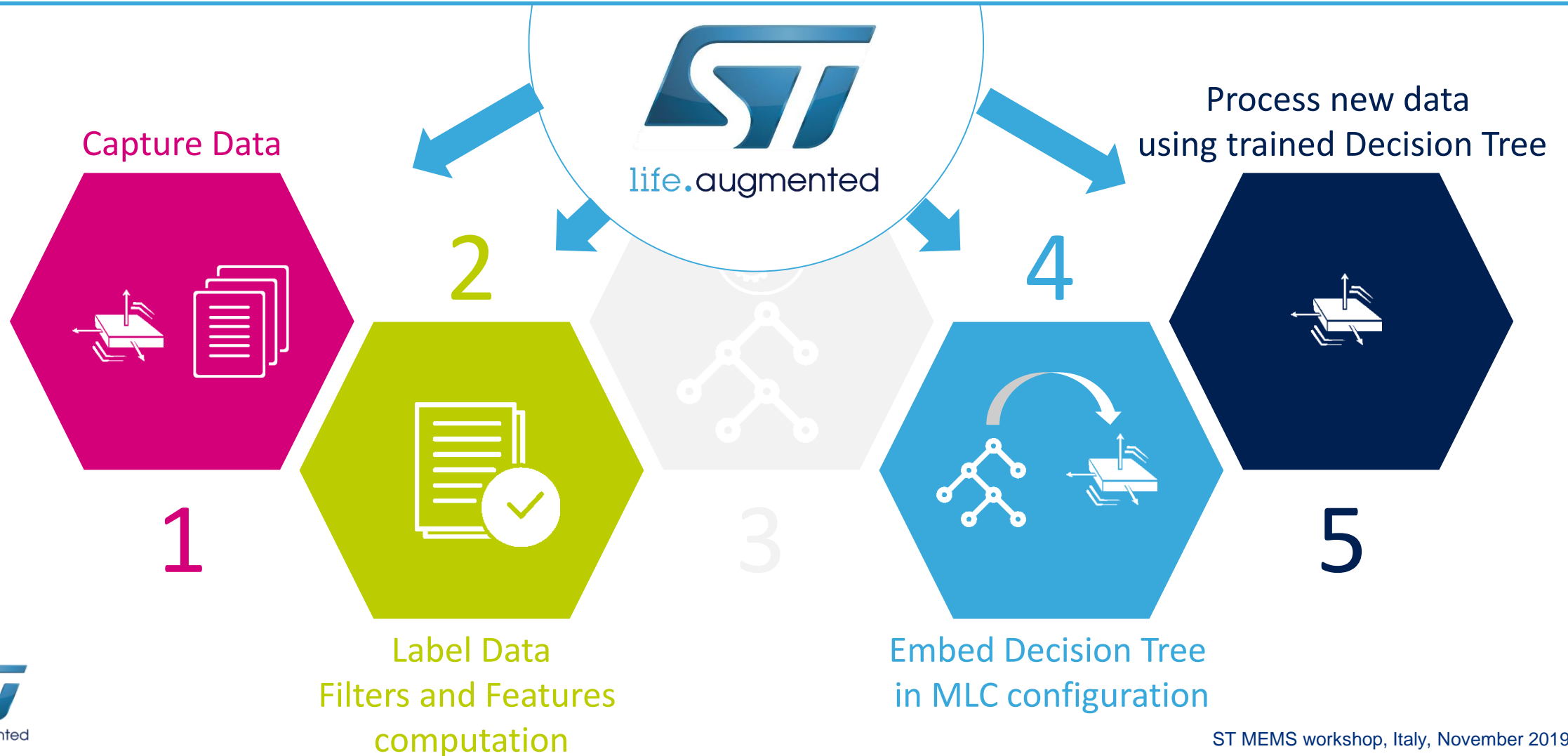
Typical node



ST Tools for Machine Learning Core

29

ST's Unico GUI SW for PC supports steps 1 / 2 / 4 / 5



Build the Decision Tree through ML tools

30

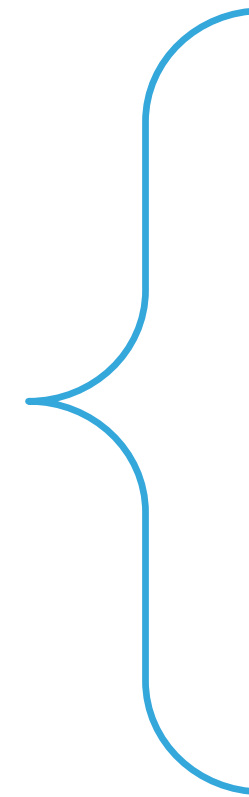
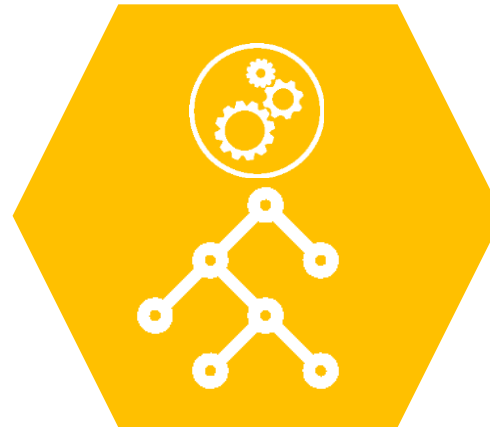
External tools for decision tree generation

- The decision tree can be generated by a dedicated Machine Learning tools:

WEKA

- Attributes selection (from .ARFF file)
- Data filtering
- Decision tree generation
- Decision tree performance evaluation (e.g. number of nodes, accuracy, confusion matrix, etc...)

Build Decision Tree



- Or other alternative tools:

- RapidMiner → Example available in AN5259 rev2

- Matlab → Scripts available at https://github.com/STMicroelectronics/STMems_Machine_Learning_Core/tree/master/tools/matlab

- Python → Scripts available at https://github.com/STMicroelectronics/STMems_Machine_Learning_Core/tree/master/tools/python

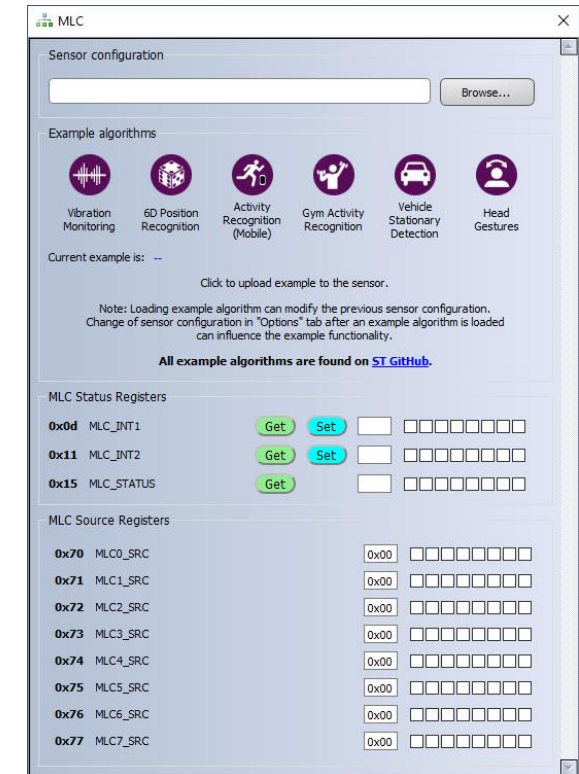


Machine Learning Core Tools

31

- Application note [AN5259](#)
- Development and Demonstration:
 - [Unico](#)
 - offline mode option for development
 - WEKA – 3rd party tool for decision tree generation
 - STEVAL-MKI109V3 + LSM6DSOX / LSM6DSRX adapter
- Demonstration:
 - [Unicleo-GUI](#)
 - STM32Nucleo + X-NUCLEO-IKS01A3 + LSM6DSOX / LSM6DSRX adapter
- Video available: [Intro to LSM6DSOX: Machine Learning Core \(11:47\)](#)
 - It explains how to utilize the MLC of the sensor.

MLC examples in Unicleo-GUI



Machine Learning Core

Algorithms Database

32

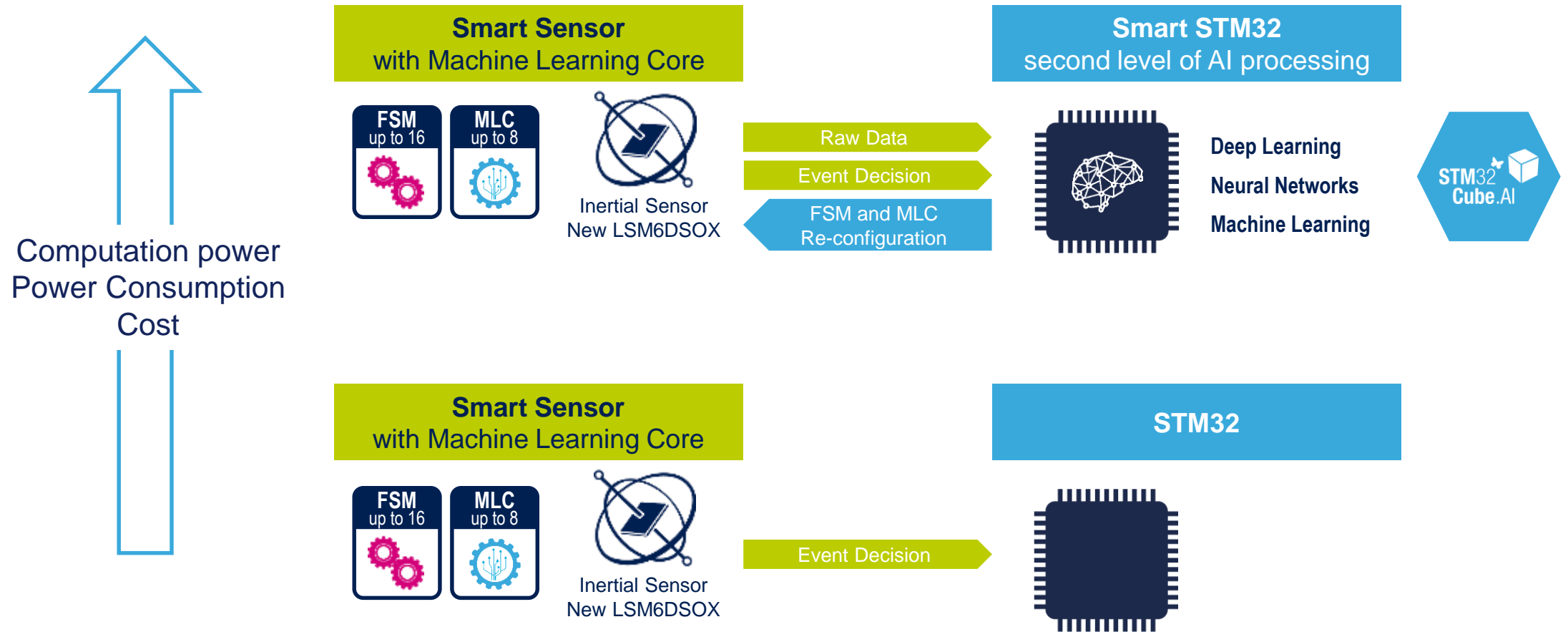
- Vibration monitoring
- Motion intensity
- 6D position recognition
- Activity recognition
- Gym activity recognition
- Carry position for mobile
- Vehicle stationary detection
- Head gestures

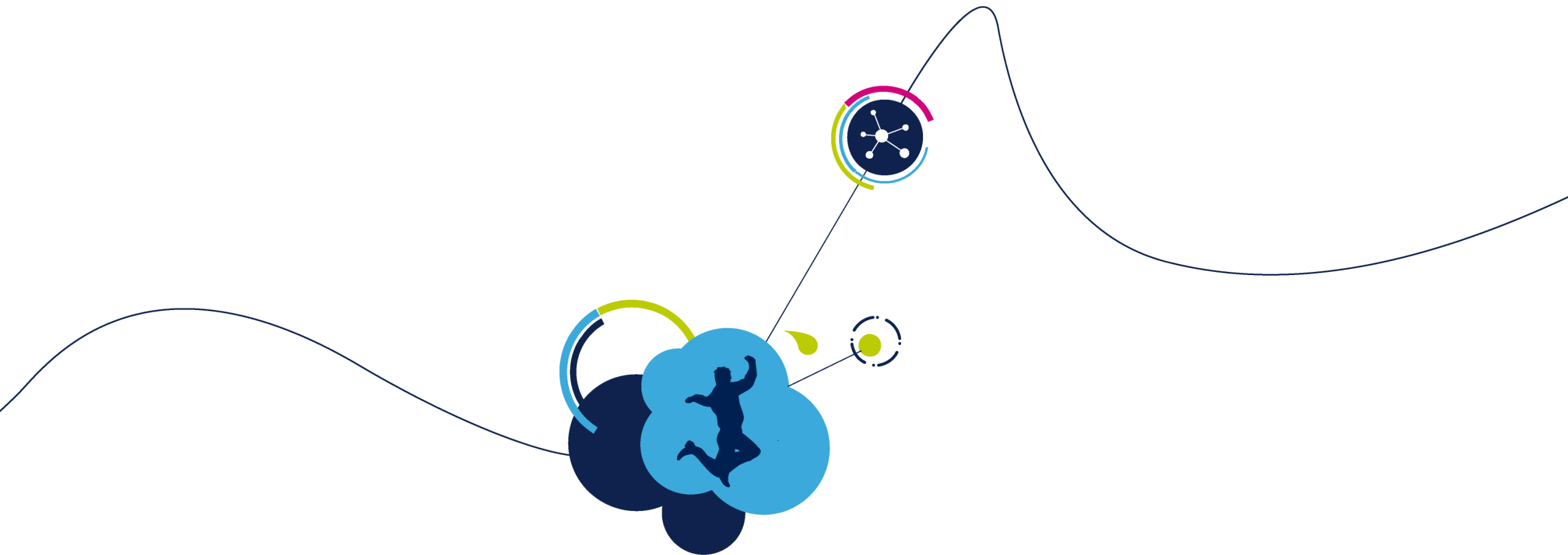


Artificial intelligence with ST devices

33

- Machine Learning Core (MLC) enables a real Edge computing by enabling system flexibility





Thank You!

