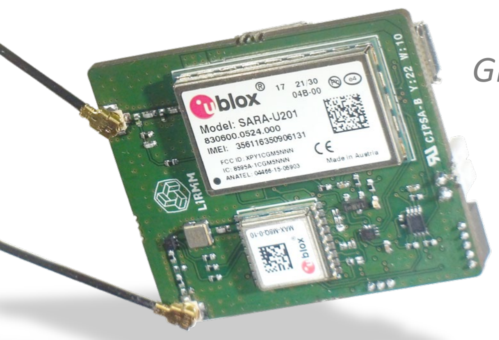
# Approches de conception « low-power » pour l'instrumentation embarquée : Applications à l'enregistrement audio, à l'actimétrie et à la géolocalisation.

Simon Chamaillé-jammes, Gustave Fradin
Laurent Latorre, Jonathan Miquel
Fabian-Robert Stöter

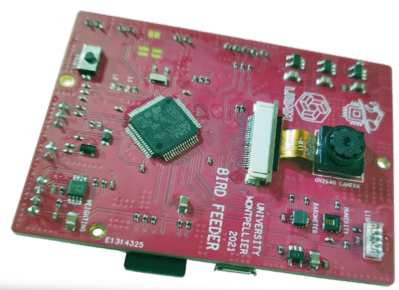
Ateliers Expérimentation et Instrumentation – Montpellier - 29-06-2022

# Context

- Development of "small" dedicated instruments
  - Wildlife monitoring (bio-loggers)
  - Environment monitoring



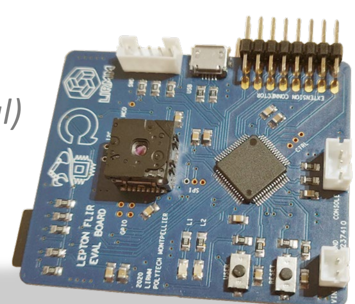
*GNSS Tracker*

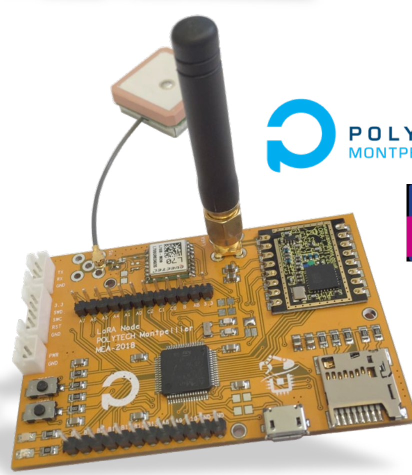


*Bird Feeder*
*Camera trap*
*Weather Station*

*Underwater*
*Light Spectrometer*

*Infrared (thermal)*
*Camera trap*





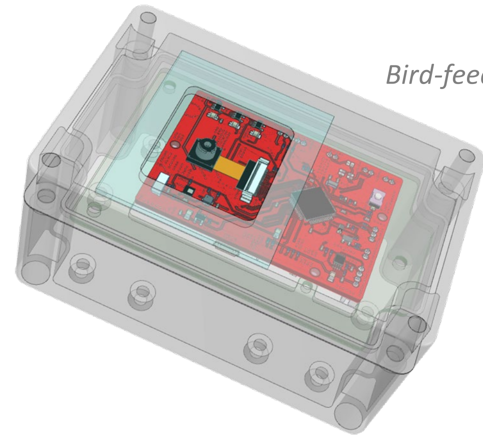
*LoRa sensor node*

*Audio + Inertial*
*Logger*

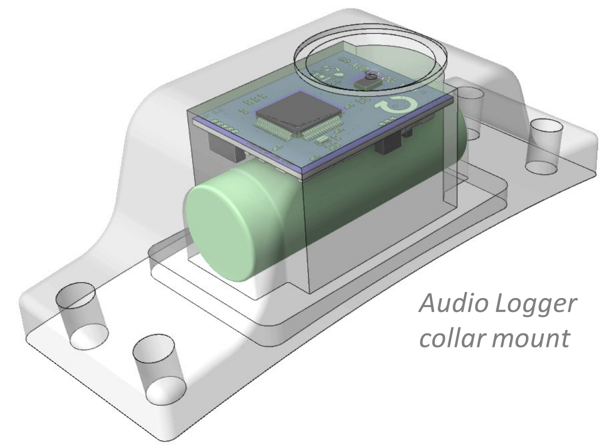
# Context

- Development of dedicated instruments
  - Hardware design (electronics, board, mechanical)
  - Embedded Software development, driven by:
    - Cost
    - Performance
    - Power-Consumption

    } Low-level programming (i.e. close to the hardware)
  - Medium to large-scale manufacturability
    - Electronics/Mechanical integration
    - Environmental constraints (e.g., waterproof, shock-resistant, …)

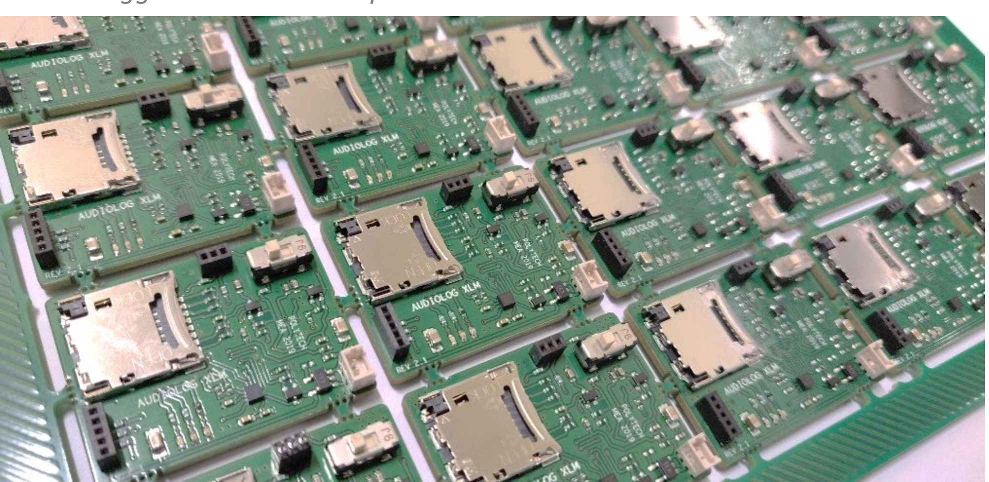


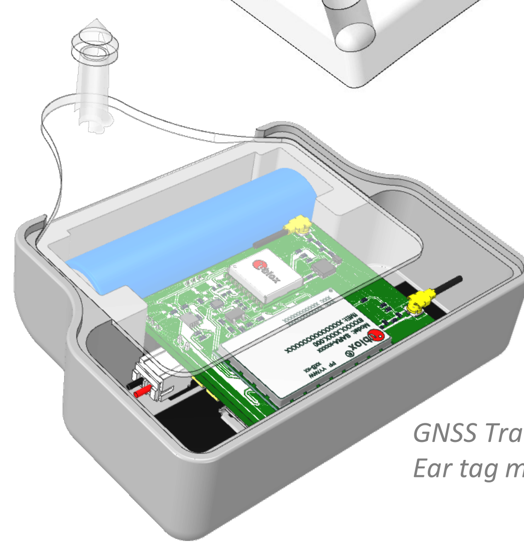
Bird-feeder casing



Audio Logger collar mount

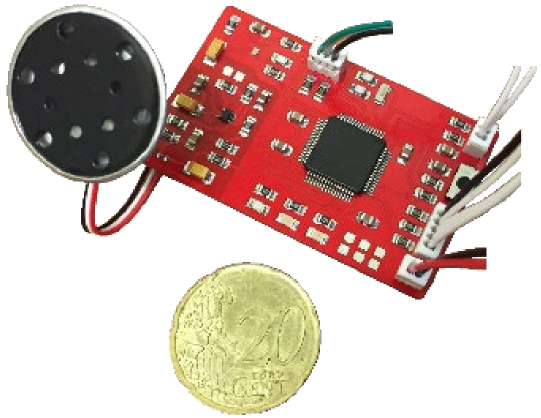
Audio-logger industrial batch production

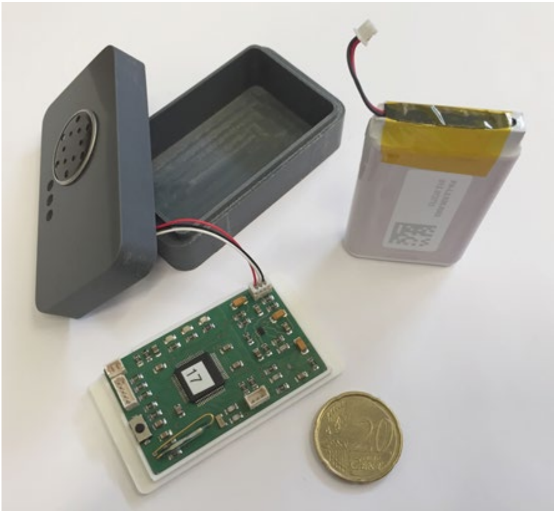




GNSS Tracker
Ear tag mount

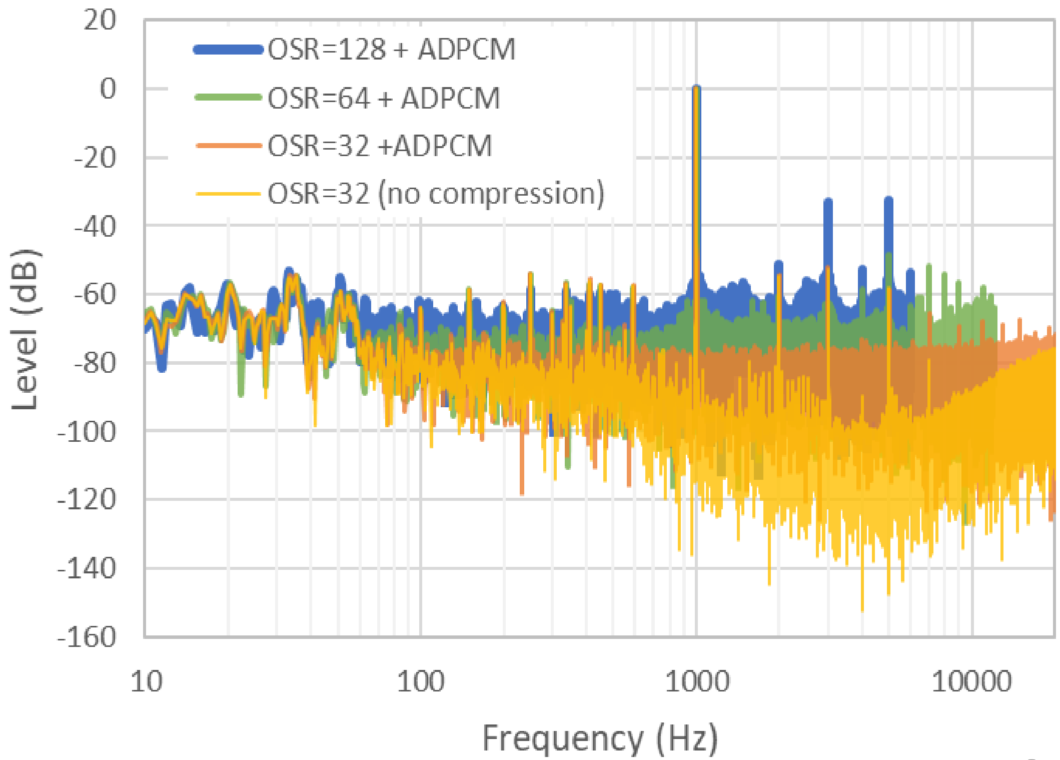
# Audio-logger v1 (2017-2019)



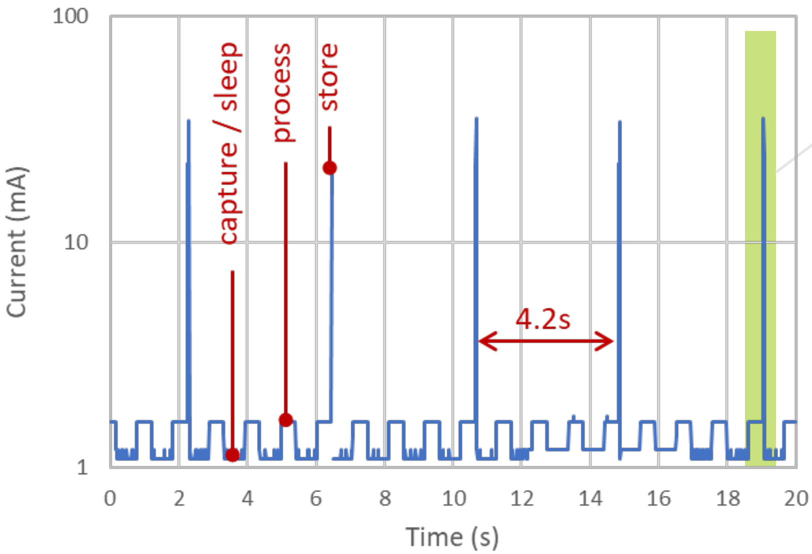


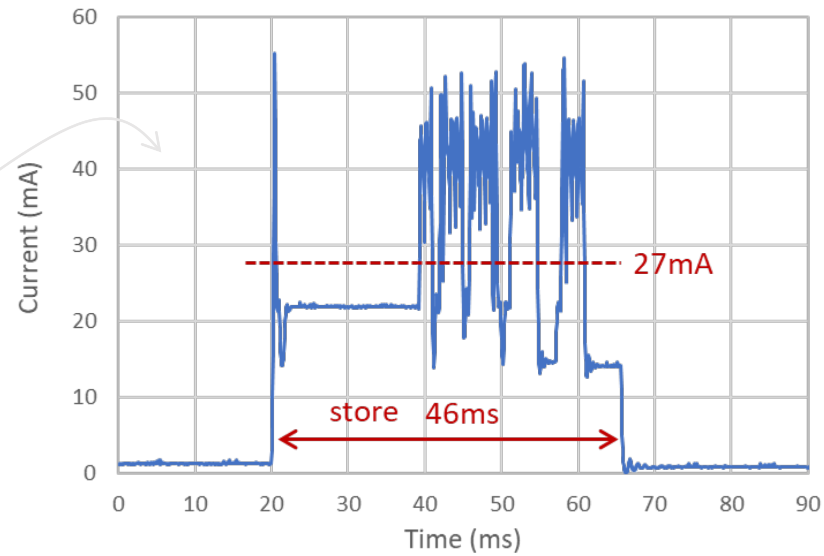
First Generation device:

- Audio only
- Waterproof, analog microphone
- Raw ADPCM in-line encoding



# Audio-logger v1

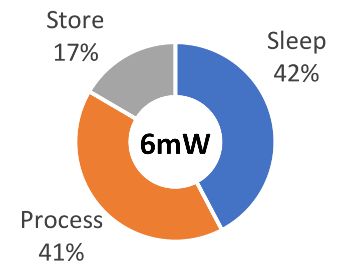


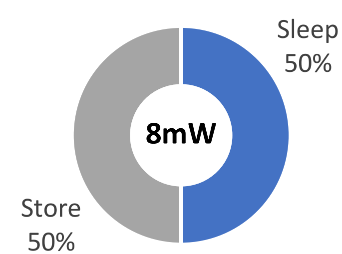




| | |
|---|---|
| CPU clock frequency | 4 MHz |
| SD modulator clock frequency | 2 MHz |
| FOSR | 32 |
| Sinc filter order X | 3 |
| IOSR | 8 |
| Sample Rate | 7.8125kHz |
| Fisrt buffering stage | 8192 (×2) 32-bit = 64kB |
| Second buffering stage | 16kB |

+ Buffering strategy
+ Embedded compression
= Power consumption <6mW

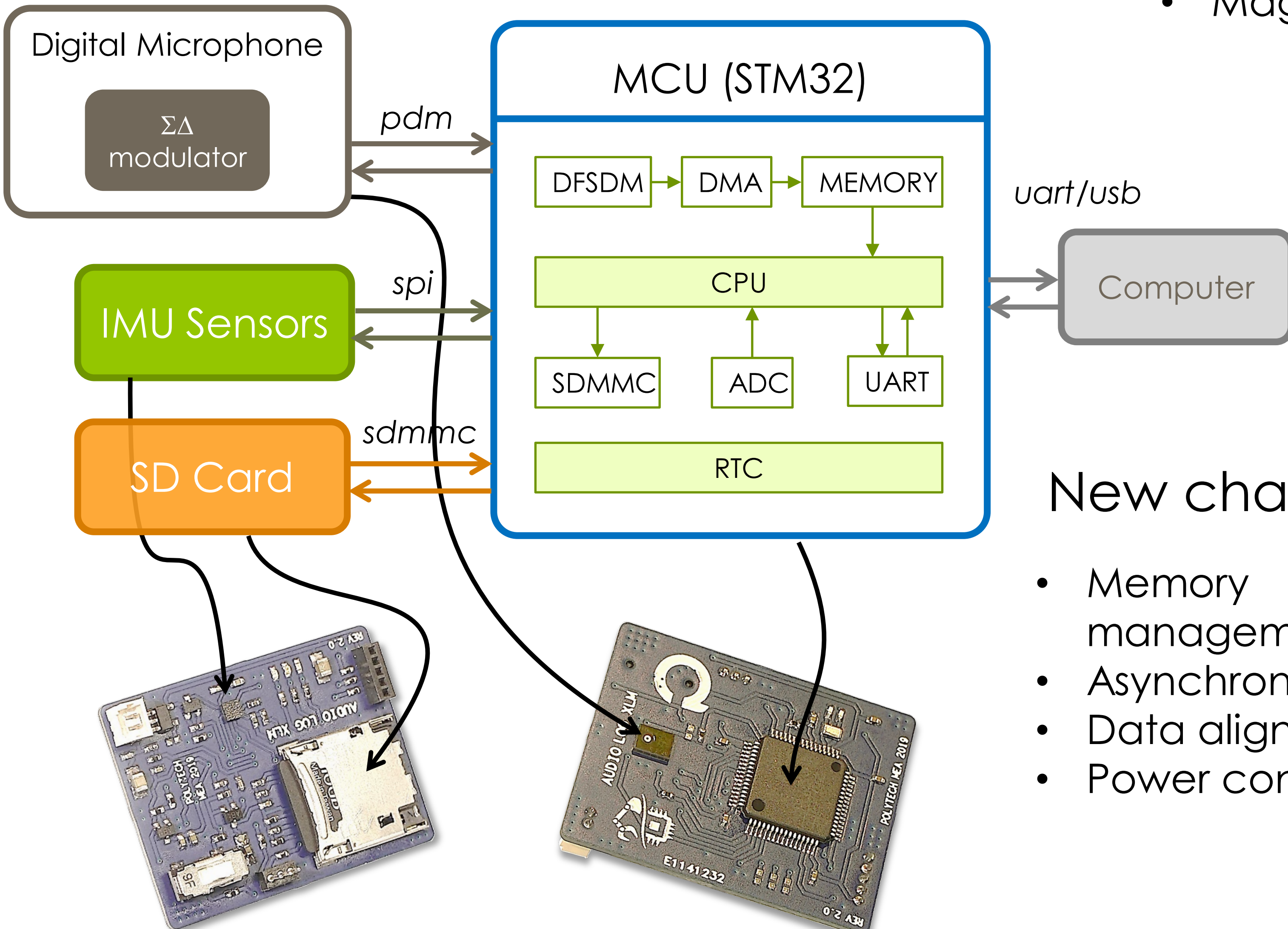
| Vdd = 3.3V | I (mA) | t (ms) | E (mJ) | # |
|---|---|---|---|---|
| **With ADPCM compression** | | | | |
| Sleep/Capture | 1.3 | 610 | 2.62 | 4 |
| Process | 1.8 | 430 | 2.55 | 4 |
| Store | 27 | 46 | 4.10 | 1 |
| Period | | | 4.2s | |
| Average Power | | | **≈ 5.9mW** | |
| **Without ADPCM compression** | | | | |
| Sleep/Capture | 1.3 | 1019 | 4.37 | 1 |
| Process | 0 | 0 | 0 | 0 |
| Store | 27 | 46 | 4.10 | 1 |
| Period | | | 1.05 | |
| Average Power | | | **≈ 8.1mW** | |



# Audio-logger XLM

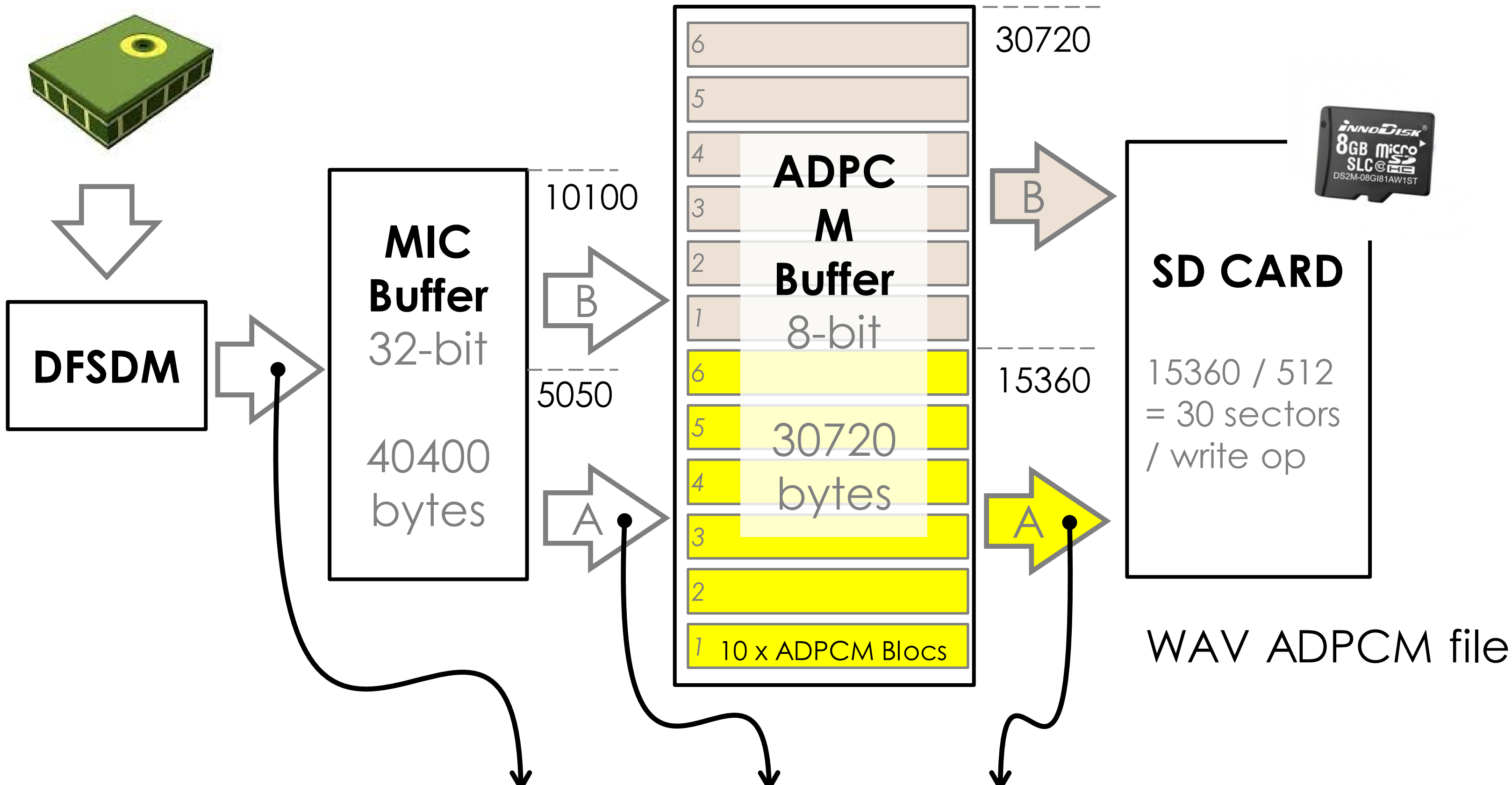
- Audio
- Acceleration
- Magnetic Field




## New challenges

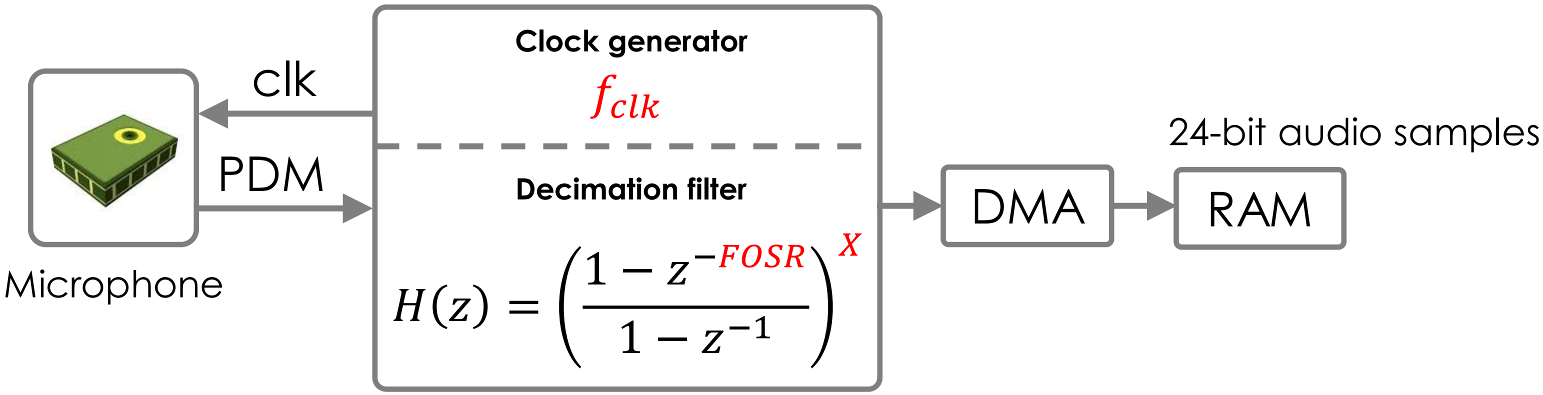
- Memory management
- Asynchronous events
- Data alignment
- Power consumption

# Audio data stream




| f_clk (MHz) | OSR | IOSR | *f_s (Hz)* | *T_wu (ms)* | *T_store (s)* |
|---|---|---|---|---|---|
| 2 | 32 | 8 | **7812,5** | **646.4** | **3.8784** |
| 2 | 64 | 1 | **31250** | **161.6** | **0.96** |

# DFSDM Peripheral (hardware)



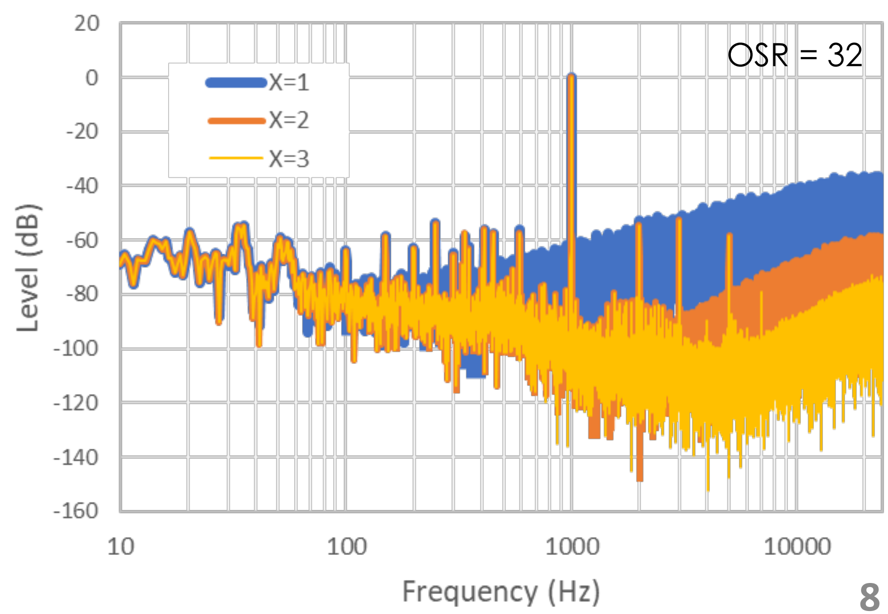

Example with X=2 $Y_n = 2Y_{n-1} - Y_{n-2} + X_n - 2X_{n-FOSR} + X_{n-2FOSR}$

## Audio performance:

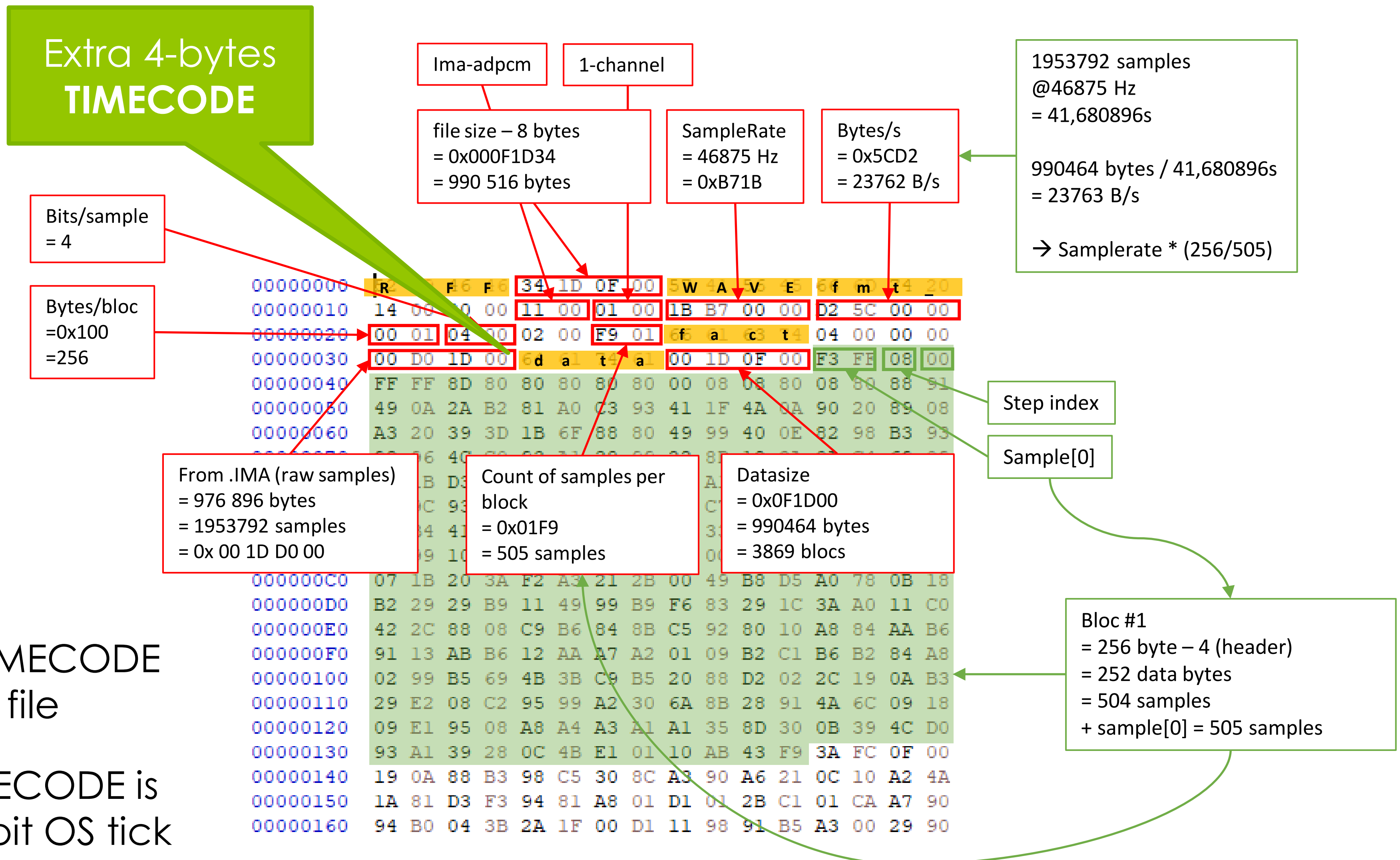
- $f_{clk}, FOSR \rightarrow$ audio bandwidth
- $X \rightarrow$ output dynamic range (FS)

$$Sample\ Rate = \frac{f_{clk}}{FOSR}$$

$$BW = SR/2 \quad FS = \pm FOSR^{X}$$



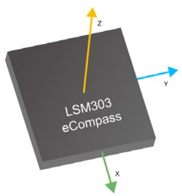
# WAV-ADPCM Header Modification

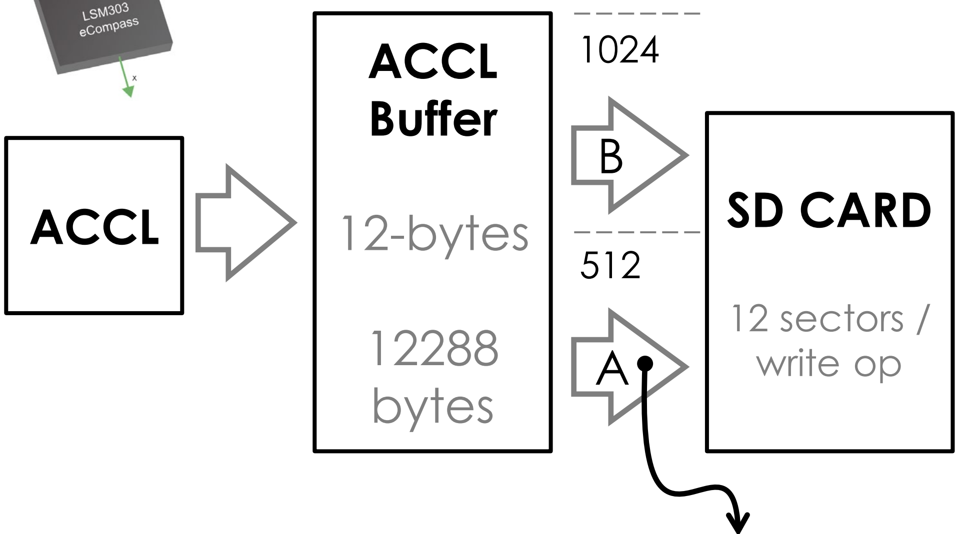



1 TIMECODE per file

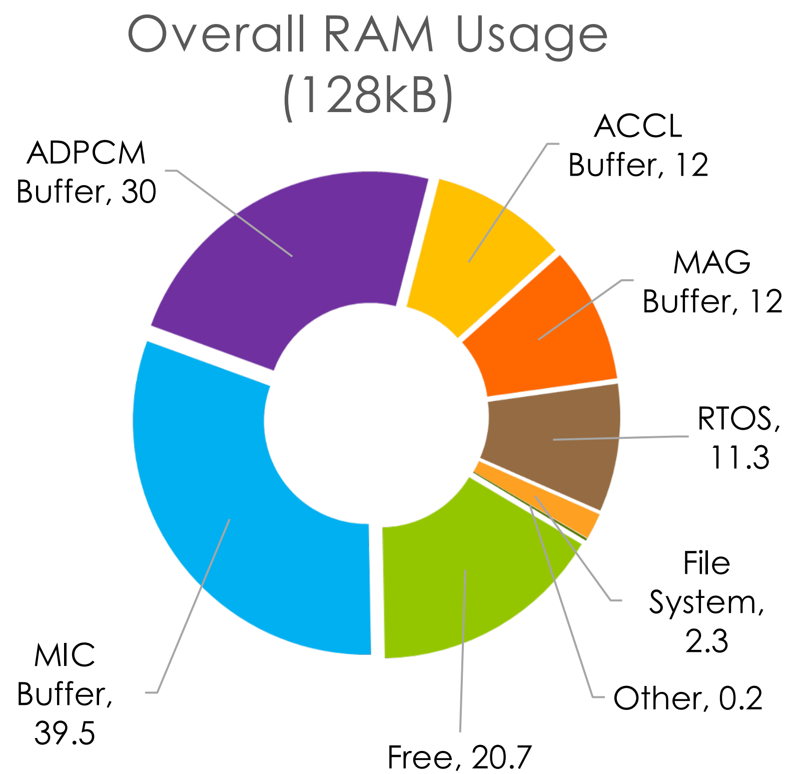
TIMECODE is 32-bit OS tick counter

# IMU data stream




| | ODR | *T_store (s)* |
|---|---|---|
| ACCL | 50 Hz | **10.24** |
| MAG | 10 Hz | **51.20** |




1 IMU data sample = 12 bytes

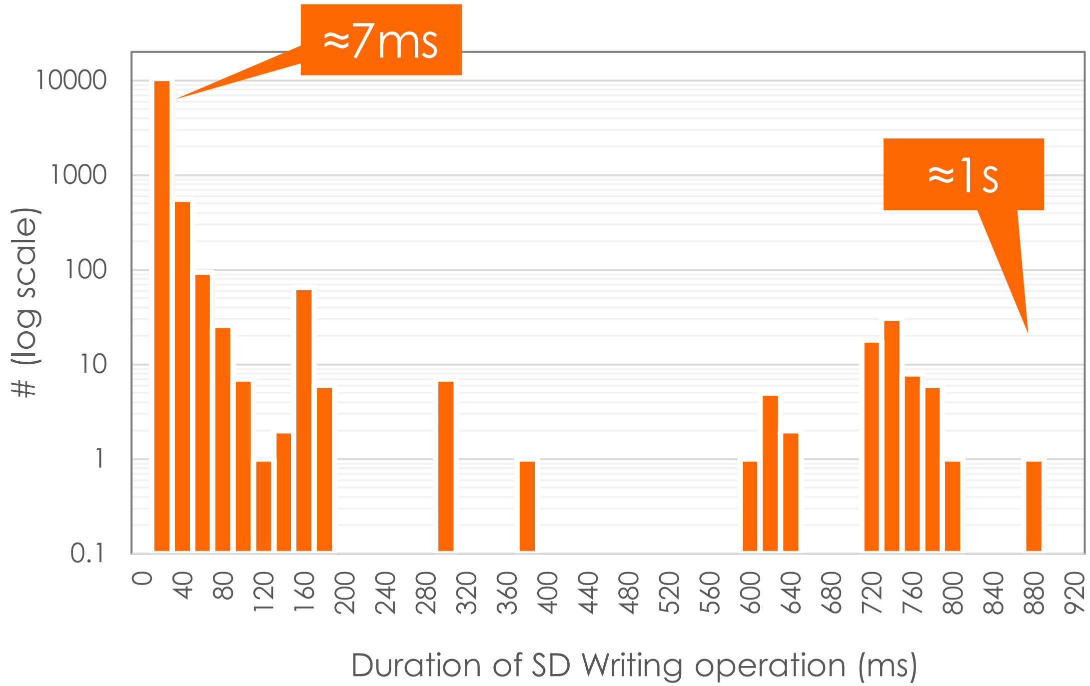
| TIMECODE | Index | z | y | x |
|---|---|---|---|---|
| 4 bytes | 2-bytes | 2-bytes | 2-bytes | 2-bytes |

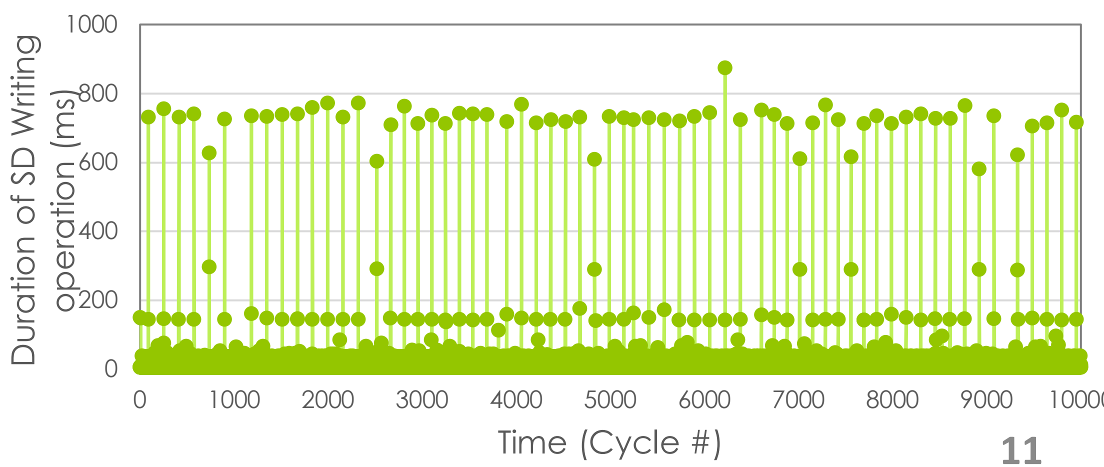
# SD Card Writing Time

**TestBench**

- Minimal firmware (no FS layer)
- Low-level addressing a single SD sector (512 bytes)
- Pre-Formatted SD card
- Timings are measured using MCU crystal OSC
- >10,000 cycles

*Buffering scheme must cope with casually long writing times*



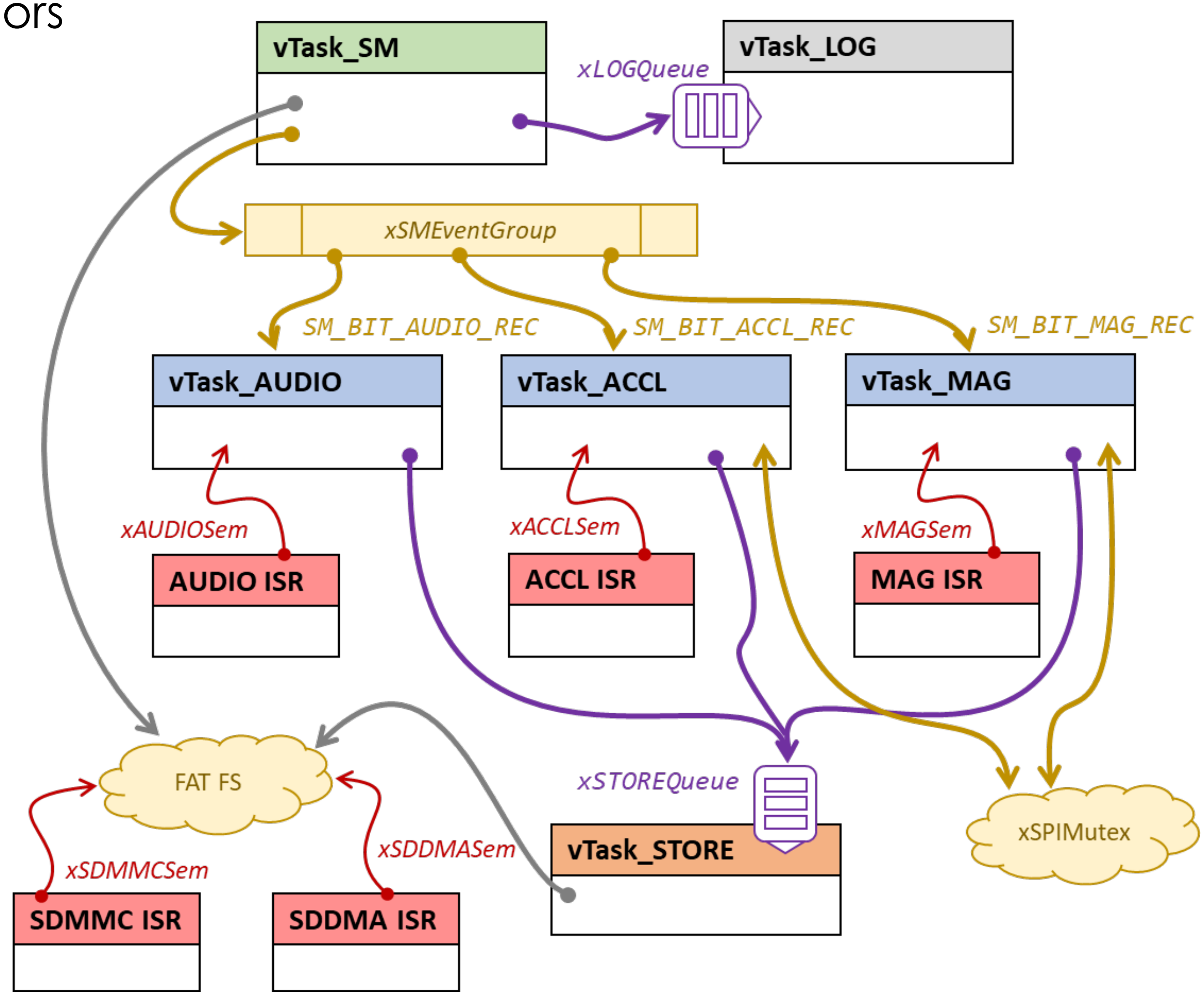
# RTOS Architecture

- Asynchronous availability of 3 sensors data
- Unpredictable SD writing times

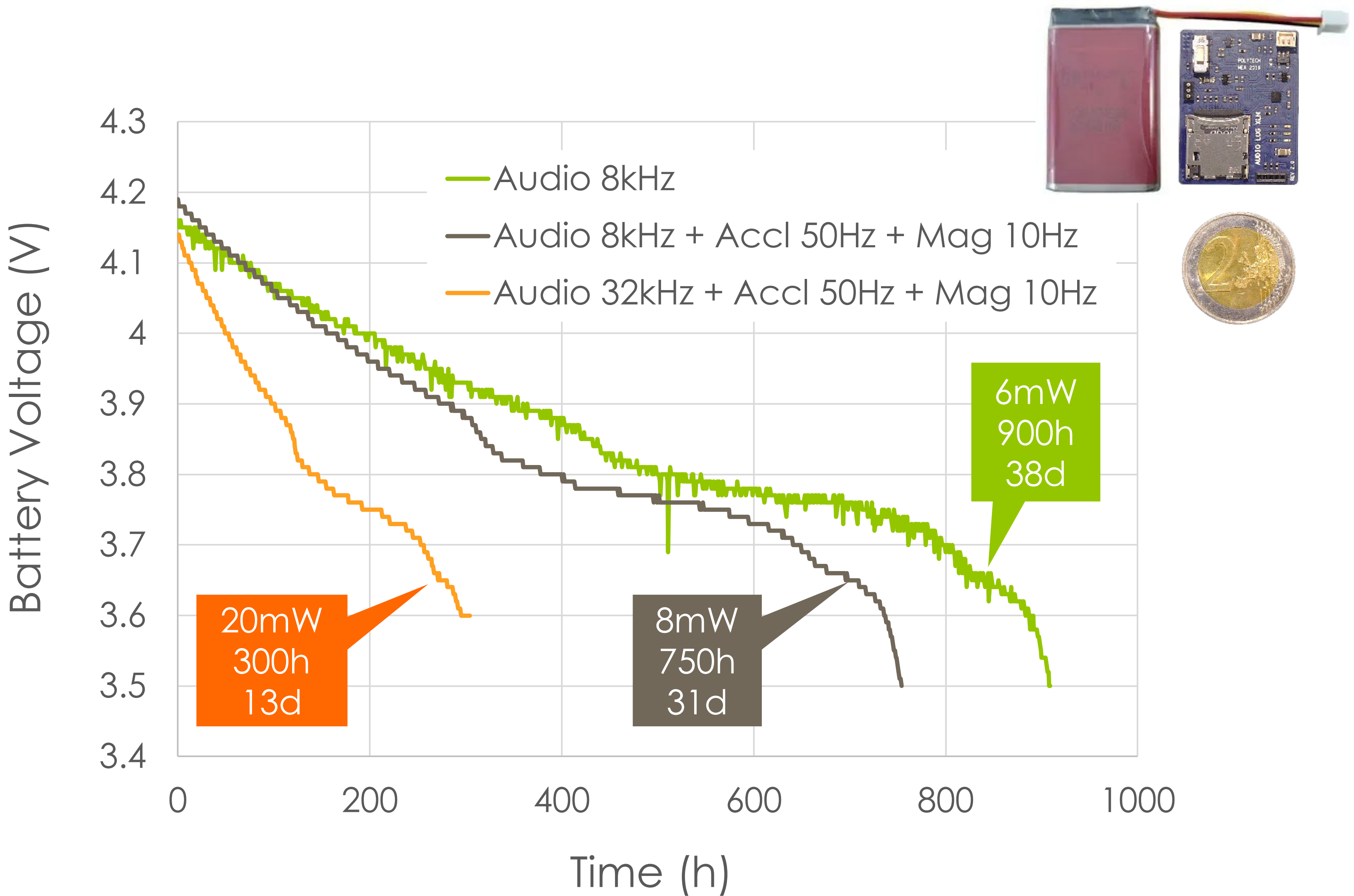
Sharing CPU resource among tasks with good (best?) power efficiency

Firmware maintainability

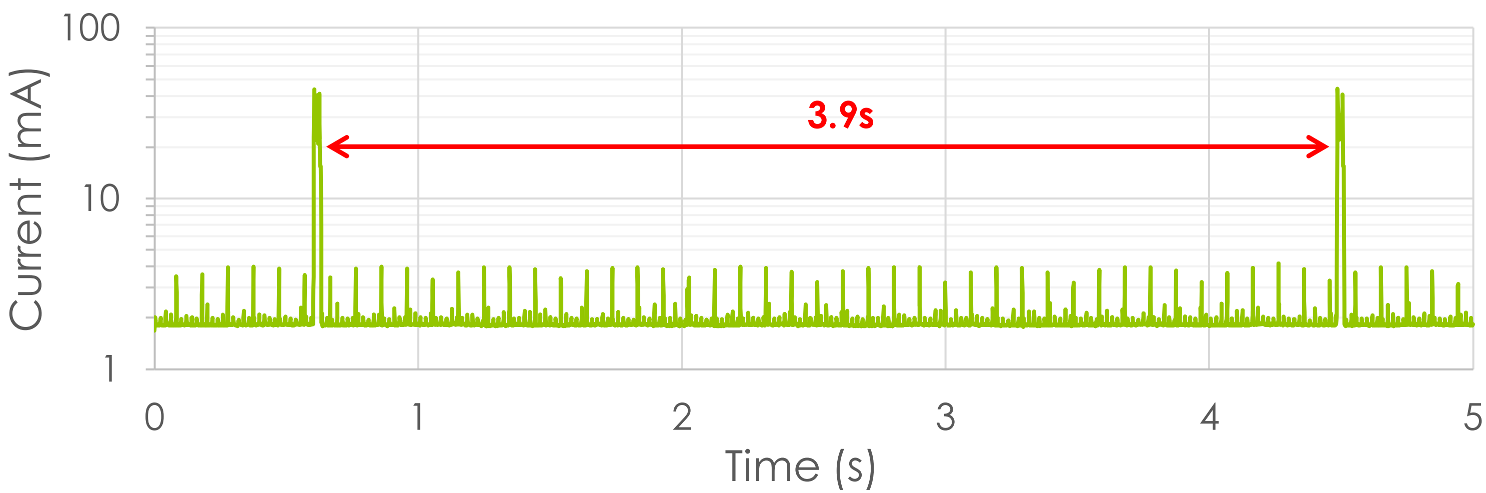
**Embedded RTOS**

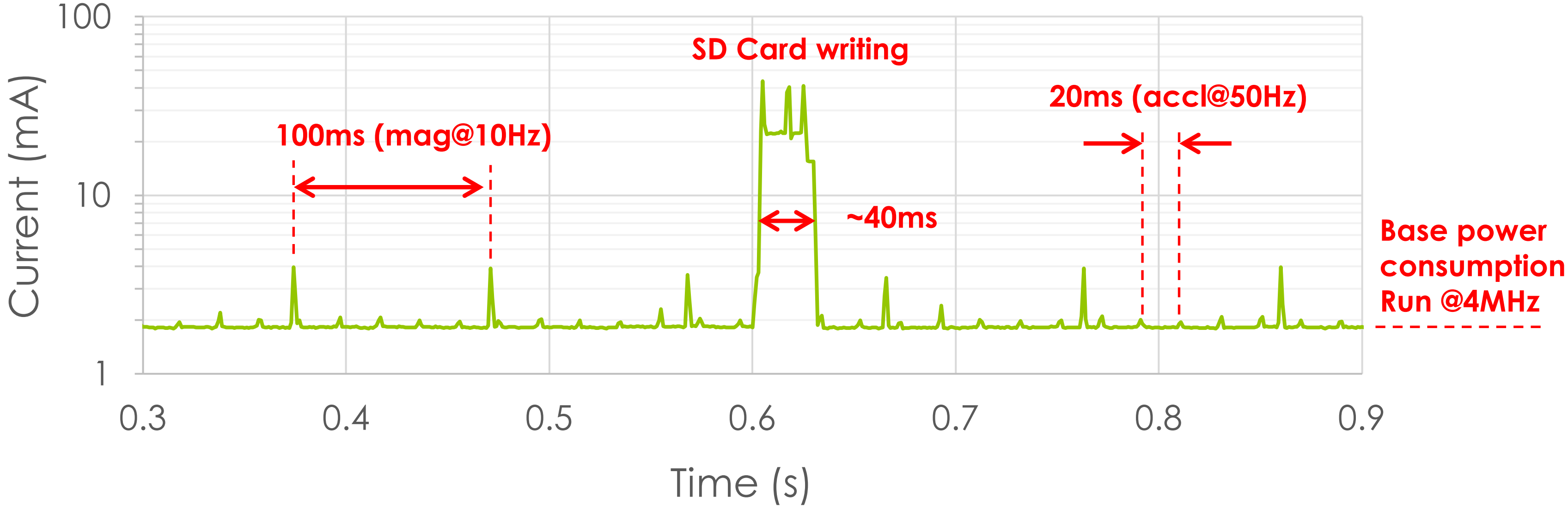


# Power Consumption

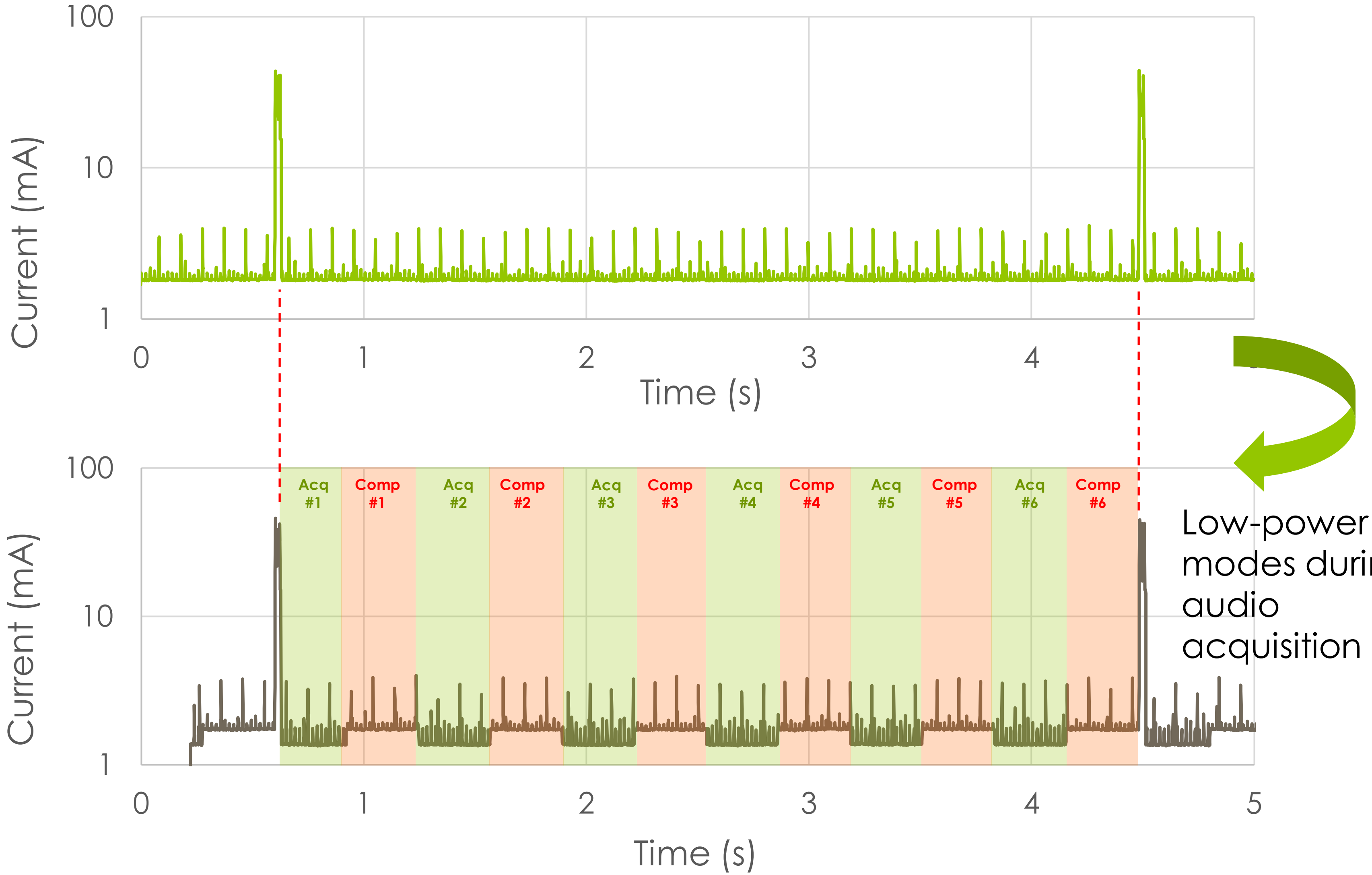


# Power Consumption



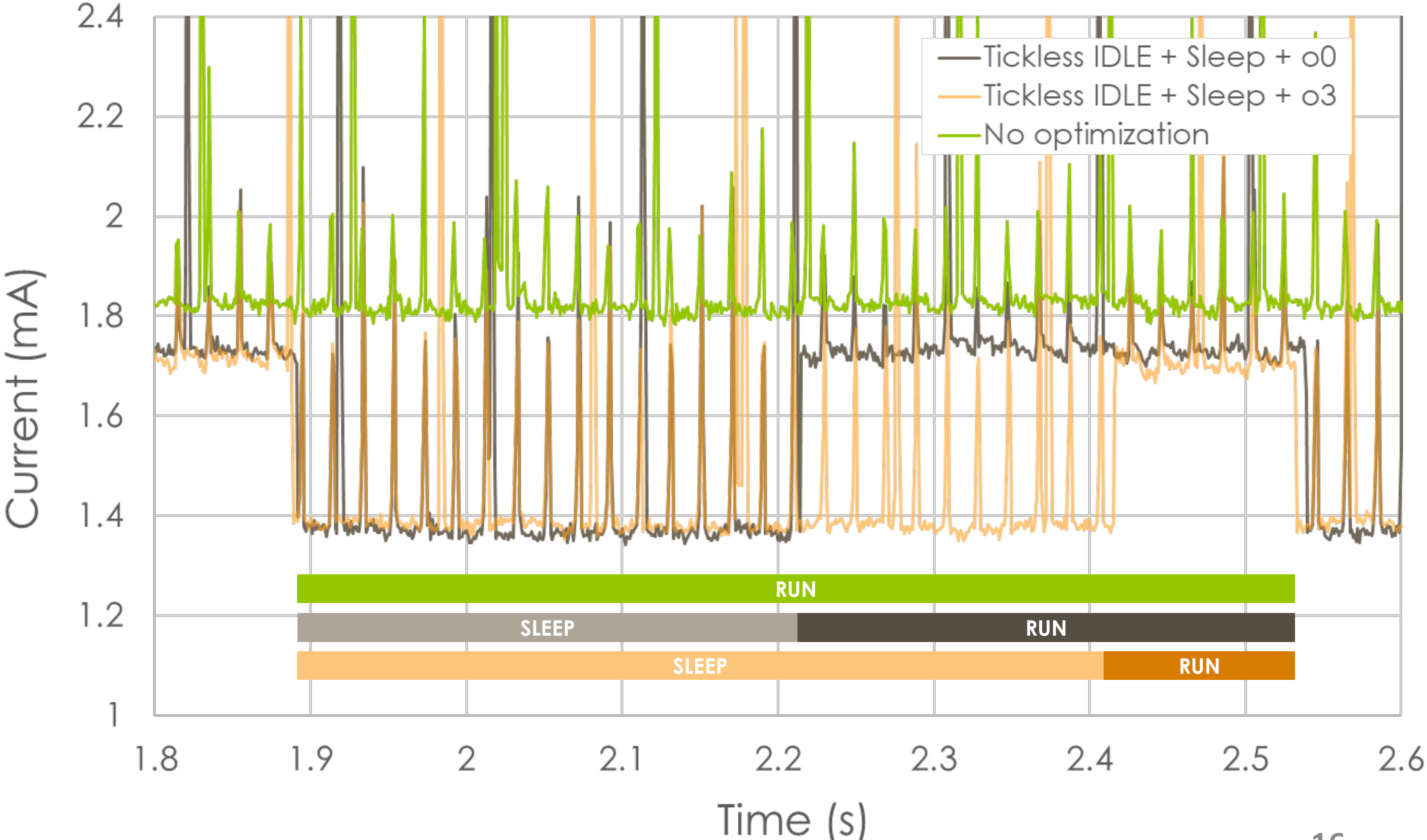
# Power Consumption



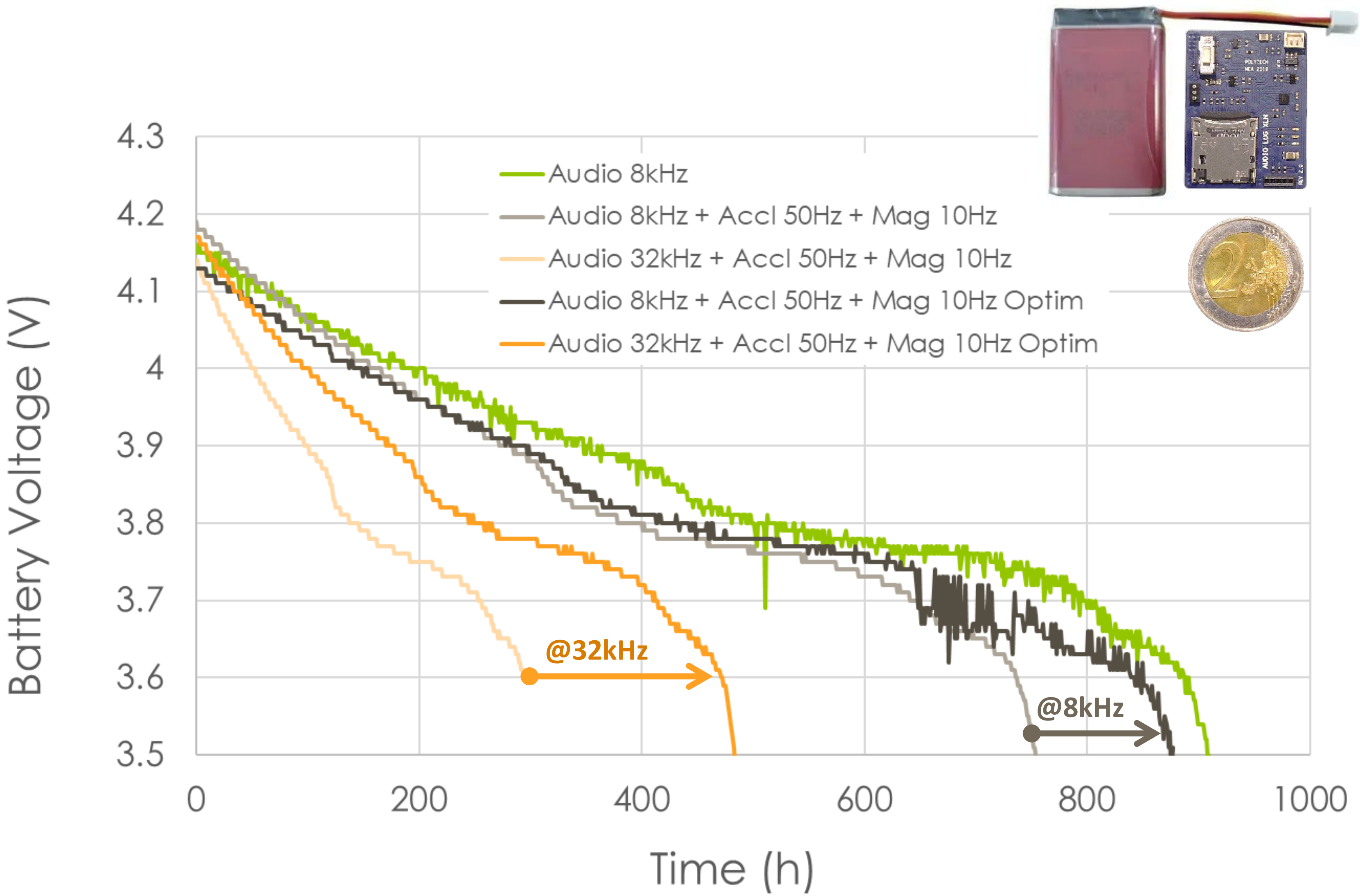

Low-power modes during audio acquisition

# Power Consumption

Compiler local optimizations

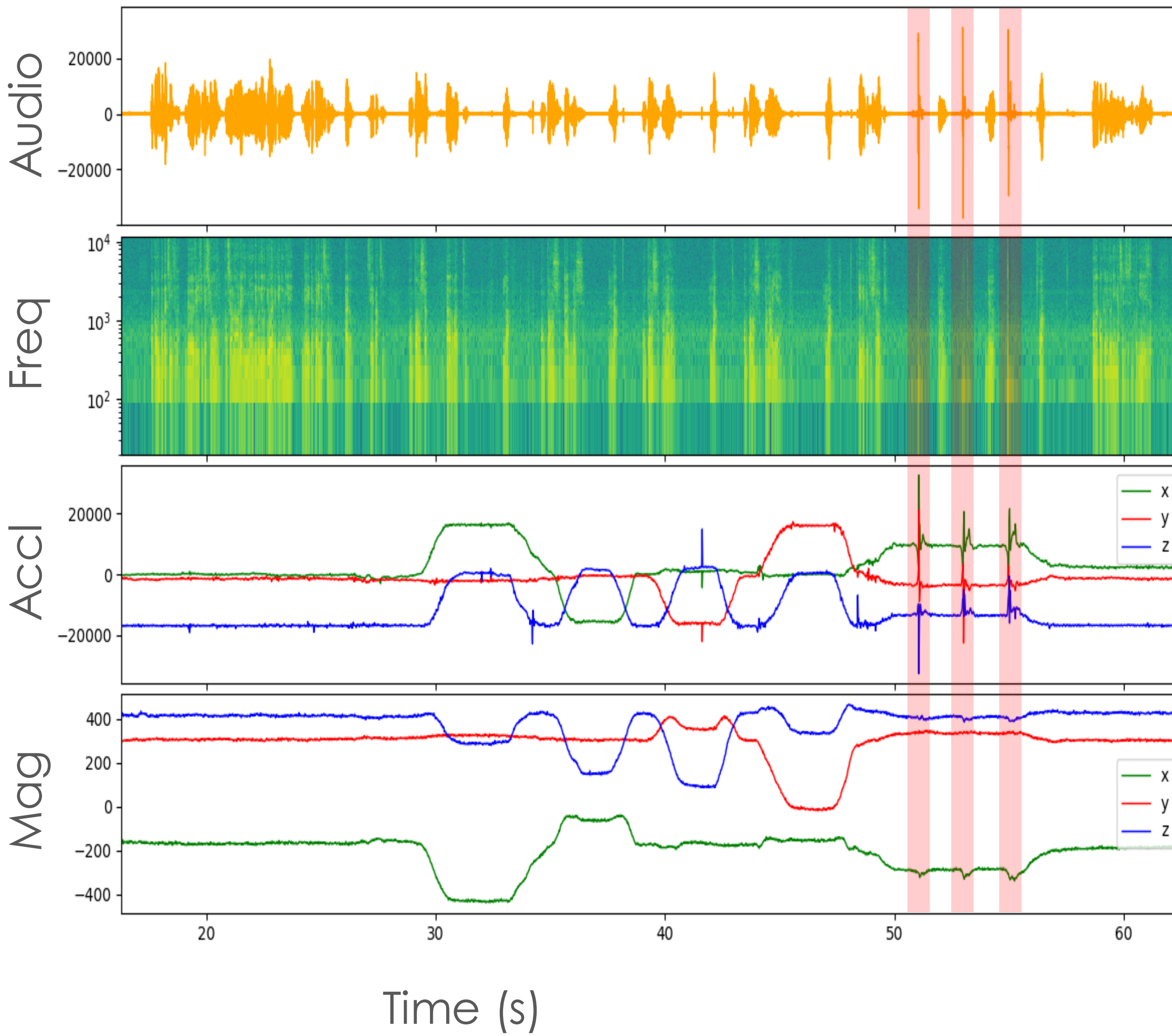


# Power Consumption



# Collected data

Aligned data within few ms accuracy



# Toward Embedded AI ?

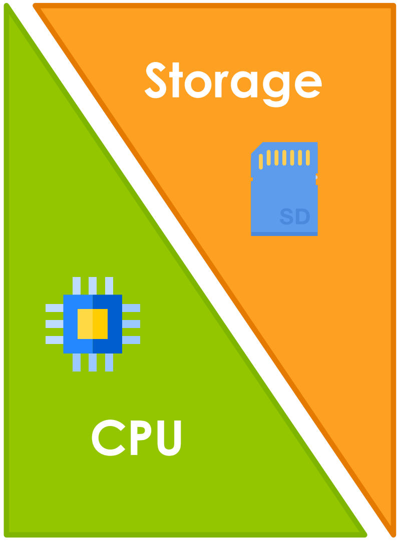
Classification of audio events
- Supervised
- Unsupervised (clustering)

- Offline Model Training

Current Approach
- Embedded raw audio
- Offline feature extraction
- Offline inference

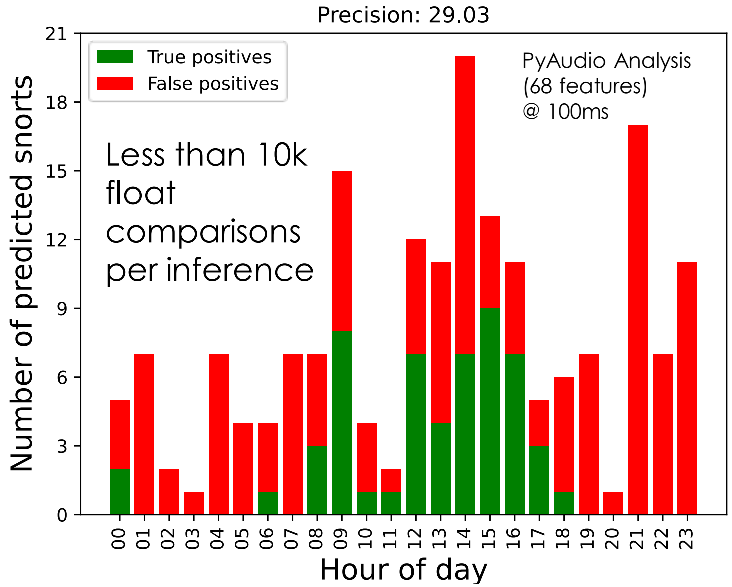
?
- Embedded feature extraction
- Offline inference

?
- Embedded feature extraction
- Embedded inference



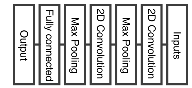
# Toward Embedded AI ?

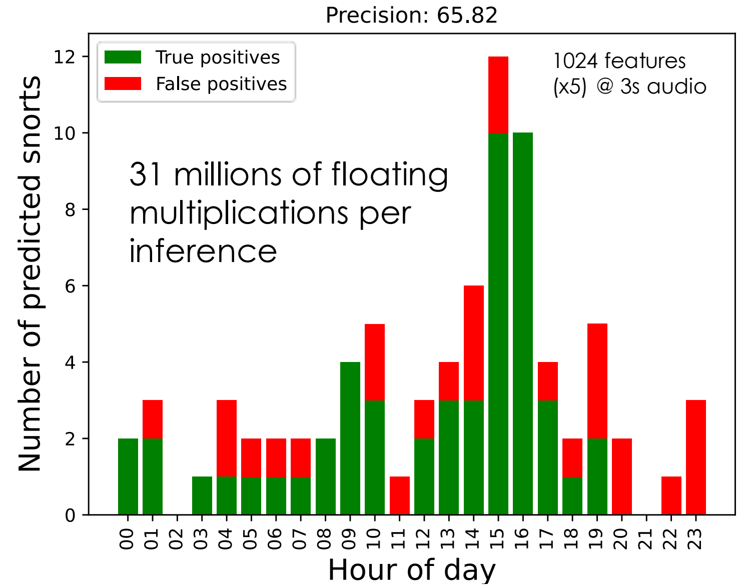
## RandomForest classifier



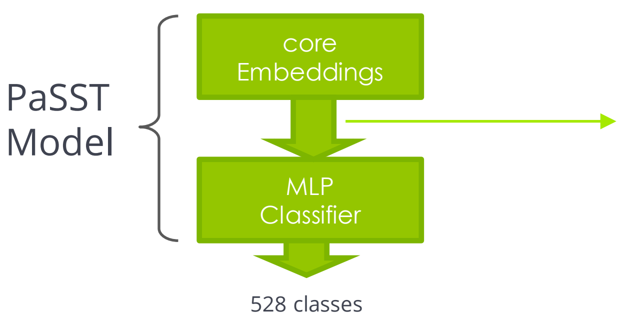

*YAMNET embedding*

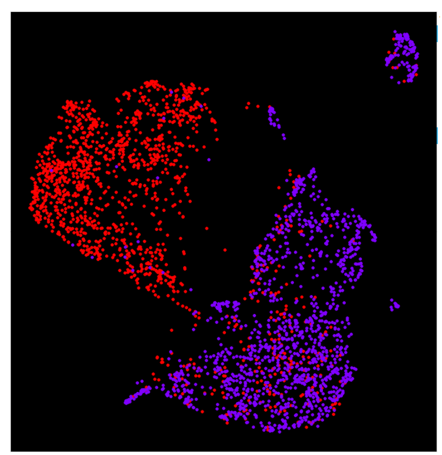
## CNN classifier




Dataset → Audioset (100Mh audio data from YouTube)



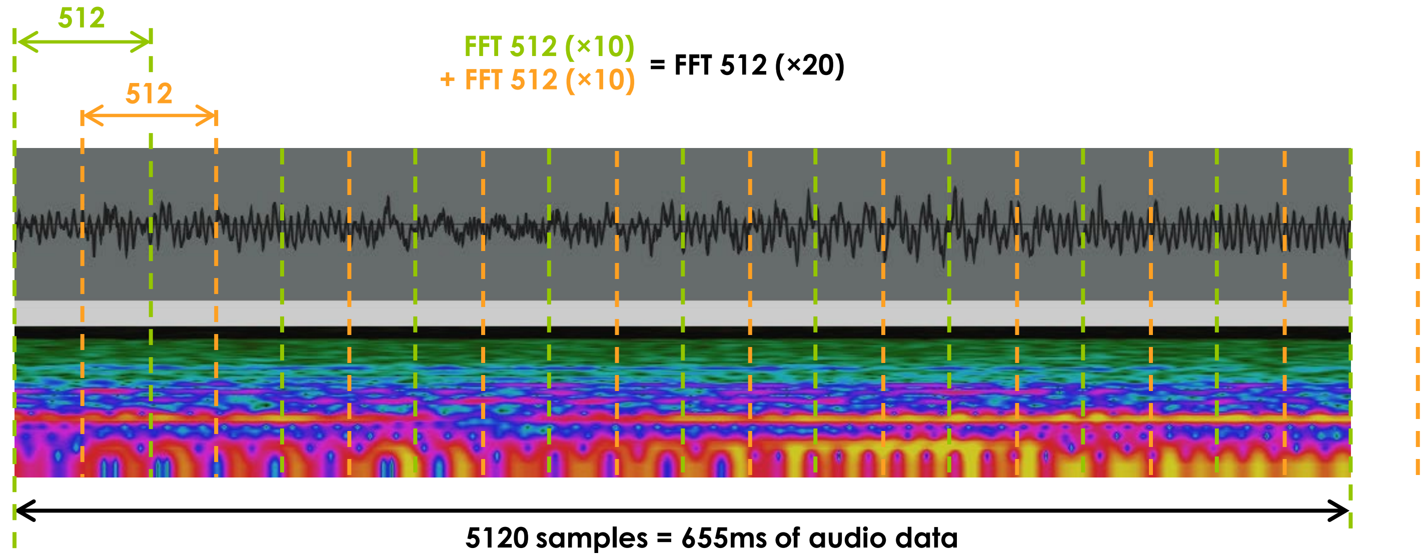
*PaSST Audioset embeddings clustered with umap*

# Toward Embedded AI ?

Audio classification based on spectrograms

Audio bandwidth : 20Hz - 3.9kHz

$f_s$ = 7812,5 Hz >390 samples → FFT 512 (65ms)




20 × FFT512 = 20 × 5ms = 100ms (CPU)
→ Real-time constraint OK
→ Equivalent to ADPCM compression

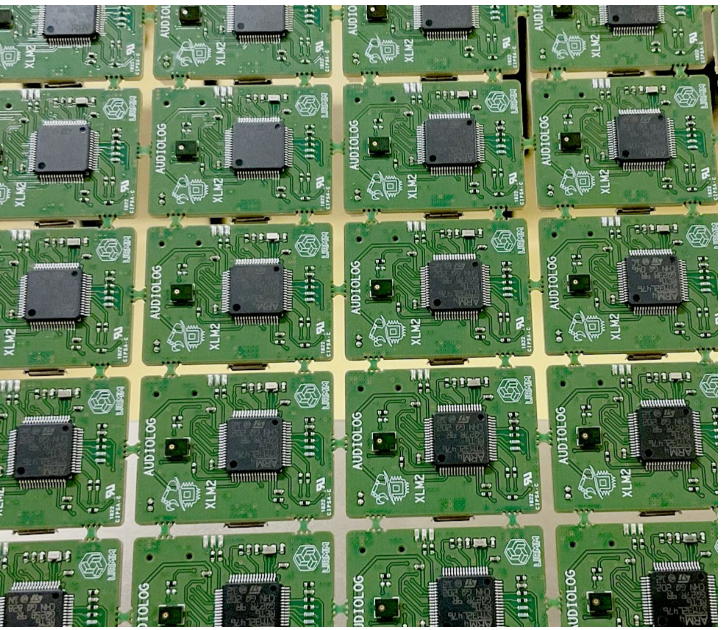
512 audio samples → 256 frq bins
→ Downsampling 1:2 → 128 8-bit frq bins

5120 16-bit audio samples
→ 2560 8-bit spectro samples

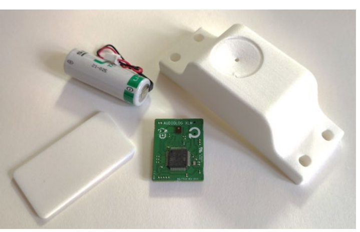
**1:4 compression ratio**

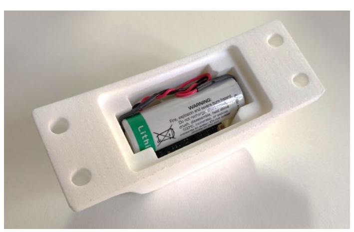
**→ Same as ADPCM**
**→ Loss of audio**

# Conclusion




- Instruments for wildlife/environment monitoring applications
- Hardware/Software development driven by the power consumption
- "Low-Level" software design approach with RTOS
- Still far from fully embedded ML classification considering strong power constraints (i.e. device weight/autonomy ratio)
- Audio-loggers in production with state-of-the-art power performance





*Acknowledgements*

CeMEB

TERRA FORMA

MUSE
MONTPELLIER UNIVERSITÉ D'EXCELLENCE

LabEx NUMEV

POLYTECH MONTPELLIER