tinyML® Summit

Miniature dreams can come true…

March 28-30, 2022 | San Francisco Bay Area

www.tinyML.org

Attracting Tomorrow TDK

Tiny ML for Tiny Sensors:

Waking smarter for less

Abbas Ataya

March 29, 2022

InvenSense
A TDK Group Company
MEMS Sensors Business Group
Sensor Systems Business Company
[4/4/2022]

Sensors and ML: waking smarter for less

Abstract:

Machine Learning at the Edge – where does the buzz stop and the value begin? If the edge is where the internet stops and the real world begins, sensors define the edge. By application, sensors convert physical phenomena into digital signals: data. What happens next is where ML is driving innovation.

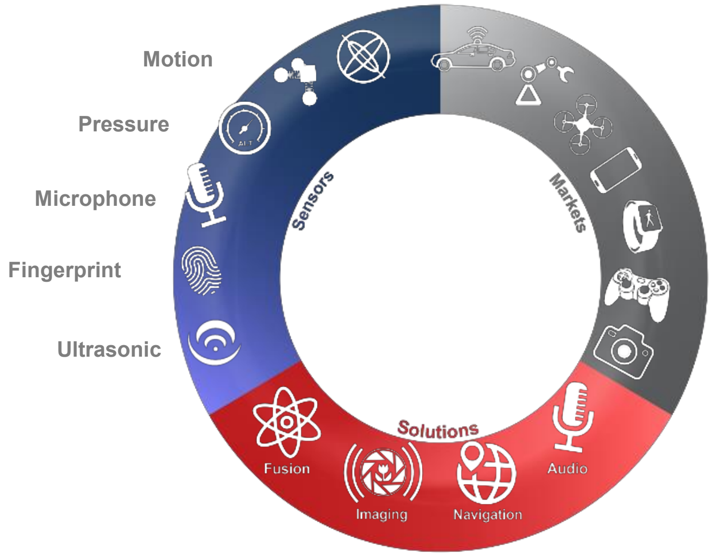
Edge implies the connectivity of a node to a bigger system. The node has a sensor chip, created in a process optimized for cost, size, and power. Transmitting raw sensor data is resource-intensive. Converting the data into information or knowledge before transmission can greatly reduce the overall burden on the system. Machine learning is revolutionizing conversion. Nodes also have a radio chip for connectivity, similarly optimized, though radio chips are made in more advanced process nodes. As such, the radio chip will always have more resources than the sensor chip. The HW/SW node design must balance the conversion giving each chip a clearly defined role to improve accuracy and power conservation in the application.

I will explore how machine learning creates new opportunities to partition far reaching sensor systems and greatly reduce required resources such as dollars, volume, and watts. Using examples of systemic optimization studied in our lab, I will show how phasing the wake up of the nodes in the hierarchy of the system reduces resources while improving response. The analogies we discuss are similar to a person waking up in the night in response to a noise, a smell, a light, or a vibration.

Sensor & TinyML

We're into sensors

- Sensors are the edge – physical converted into digital
  - Actuators convert digital into physical
- What is the job of the sensor?
  - The extremities of this central nervous system that is the internet
  - The sentry, at the edge of the system, trigger the appropriate response
  - Control the wake up, control the power usage
- Tiny is the resources, power, but not importance
  - Sensor has to transform its data into a first level of response,
  - It often serves as a first door to digital for users,
  - Must be accurate for the users and for the power



Sensor Node: An Always-On, Low Power, Sentinel

Attracting Tomorrow TDK

Sensor Node

Communication

Edge Processor

Sensors

Power

Smart Sensor

Always on

Low

Cost

Power

Size

50 to 500µA

Mission

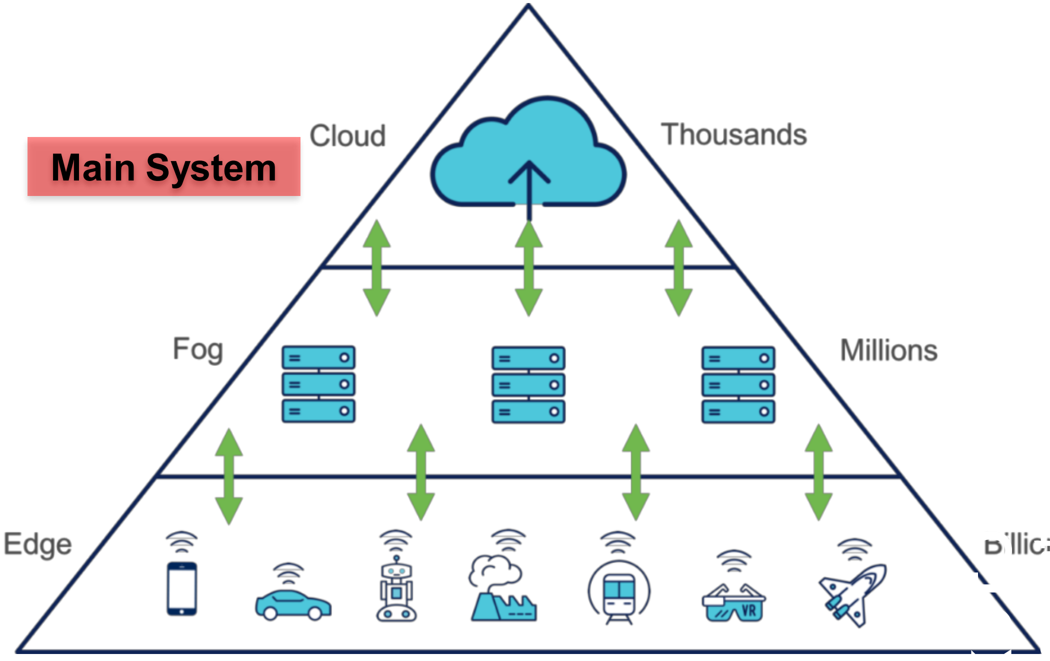
At Very Low Power

Always On

Identifies a pattern

Triggers the first layer of the main system

Main System



5mA to 500mA

High

Cost

Power

Size

10 watts and above

Often off

System phased Wake Up View

Attracting Tomorrow TDK

	Power	Motion	Mic	Computation	Human wakes up in middle of the night	
8	>1 A		Speech to text	Cloud (big iron), learning	Making breakfast, going to work	Plugged in
7	100 mA	Context, accurate Location		AP on, connecting to cloud (system and radio power)	Calling 911, wake up someone else	hours
6	2 mA	Image stabilization, rough location	ANC, CW	Sensor hub on (fusion with other sensors)	Out of bed	day
5	250 µA	Gestures Activity class, GAF	WW	On chip (edge) ML (windowed)	Sit up, lights on	4 days
4	75 µA	Bring2See Pedometer	VAD, AAD	DMP, Target for chip ML	Open eyes	10 days
3	10 µA	Being held/worn	No Audio	RTL, WoM	Listening, did I hear something	2 weeks
2	1 µA				Roll over	months
1	100 nA	No motion		N-Zero program	Grunt	year
0	10 nA				Snoring	years

System phased Wake Up View

Attracting Tomorrow TDK

Level	Power	Motion	Mic	Computation	Human wakes up in middle of the night	
8	>1 A		Speech to text	Cloud (big iron), learning	Making breakfast, going to work	Plugged in
7	100 mA	Context, accurate Location		AP on, connecting to cloud (system and radio power)	Calling 911, wake up someone else	hours
6	2 mA	Image stabilization, rough location	ANC, CW	Sensor hub on (fusion with other sensors)	Out of bed	day
5	250 µA	Gestures Activity class, GAF	WW	On chip (edge) ML (windowed)	Sit up, lights on	4 days
4	75 µA	Bring2See Pedometer	VAD, AAD	DMP, Target for chip ML	Open eyes	10 days
3	10 µA	Being held/worn	No Audio	RTL, WoM	Listening, did I hear something	2 weeks
2	1 µA				Roll over	months
1	100 nA	No motion		N-Zero program	Grunt	year
0	10 nA				Snoring	years

Sensor Domain

System phased Wake Up View

Attracting Tomorrow TDK

Sensor Domain

	Power	Motion	Mic	Computation	**Human wakes up in middle of the night**	
8	>1 A		Speech to text	Cloud (big iron), learning	Making breakfast, going to work	Plugged in
7	100 mA	Context, accurate Location		AP on, connecting to cloud (system and radio power)	Calling 911, wake up someone else	hours
6	2 mA	Image stabilization, rough location	ANC, CW	Sensor hub on (fusion with other sensors)	Out of bed	day
5	250 μA	Gestures Activity class, GAF	WW	On chip (edge) ML (windowed)	Sit up, lights on	4 days
4	75 μA	Bring2See Pedometer	VAD, AAD	DMP, Target for chip ML	Open eyes	10 days
3	10 μA	Being held/worn	No Audio	RTL, WoM	Listening, did I hear something	2 weeks
2	1 μA				Roll over	months
1	100 nA	No motion		N-Zero program	Grunt	year
0	10 nA				Snoring	years

Keep Smallest Size,
Lowest Power,
All While Brining More and More Features and Functionalities

Best (FRR/FAR) are optimizing power

- Low FRR is key for the performance
  - False Rejection is when we miss a Wake Up,
  - In Consumer product, means a user fails his interaction with his device
  - Poor user experience

- Low FAR is key for the power,
  - Example of a Motion to wake up a SmartWatch (Bring to See/B2S)
    - The more B2S False Accepts (detect B2S where it shouldn't),
    - the more power we burn uselessly by waking up the screen (turn on)

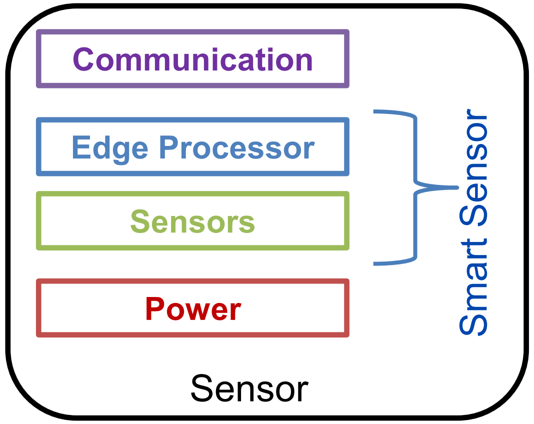
  - Example of a Voice Activity Detection (VAD) on Sensor to trigger a KeyWord Spotting (KWS on MCU) to wake up Automatic Speech Recognition (ASR) on Cloud
    - the more VAD False Accepts (detect Voice Activity when there's none),
    - the more power we burn in useless KWS

Machine Learning with DNN is (best) candidate
to provide optimized FAR/FRR Solution

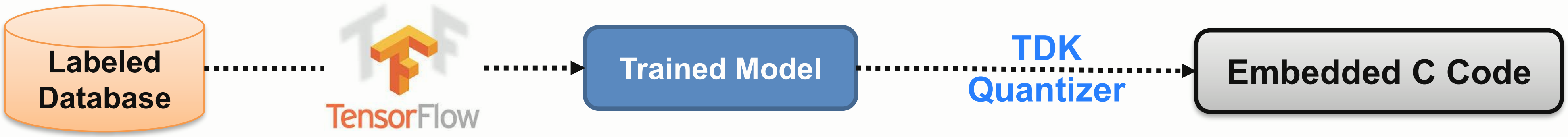
What do we do?

Our Approach

TensorFlow made Tiny & Easy for



~10kB RAM, ROM, 4 MHz
50µA matters



Neural Networks Trained Model is any combination of:

- Dense Layers (DNN)
- Convolutional Layers (CNN)
- Recursive Layers (GRU, time series)
- Activations Layers (Tanh...)
- Decision Layers (Softmax..)

Core SW Tech Automated Embedded code Generation: From TensorFlow to optimized integer C

- All Floats computations converted to integers (Matrix, Tanh.., less die size)
- Windows computations converted to singles frames (less memory, less CPU)
- Layers parameters reduced (Weights quantized to 8 bits, less memory)
- Accuracy tested and kept, reported.
- Code optimized and ready, run smoothly on M0+ ,M4..

AAR example	TF.Lite	TDK Quantizer
Code	303Ko	5.5Ko
Model	6.72ko	1.41ko
Stack	12ko	5.5ko
Accuracy	85.5%	86.95%

```
def create_model_AudioKWS(input_shape,nb_hidden,nb_class): #HiTDK 1.0.0 model

    layersizes=[48, 40, 40, 60]
    dropout=[0.8, 0.8, 0.8, 0.3]
    stride_size=1
    x_input = Input(shape=input_shape)

    # Step 1: CONV layer
    X = Conv1D(layersizes[0], kernel_size=10, strides=stride_size, padding="same")(x_input)
    X = BatchNormalization()(X)  # Batch normalization
    X = Activation('relu')(X)  # ReLu activation
    X = Dropout(dropout[0])(X)  # dropout (use 0.8)

    # Step 2: First GRU Layer
    X = GRU(units=layersizes[1], return_sequ
    X = Dropout(dropout[1])(X)  # dropout (u
    X = BatchNormalization()(X)  # Batch nor

    # Step 3: Second GRU Layer
    X = GRU(units=layersizes[2], return_sequ
    X = Dropout(dropout[2])(X)  # dropout (u
    X = BatchNormalization()(X)  # Batch nor

    # Step 4: final dense layer
    X = Dense(layersizes[3], activation="rel
    X = Dropout(dropout[3])(X)
    X = TimeDistributed(Dense(nb_class, activation="sigmoid"))(X)  # time distributed  (si

    model = Model(inputs=x_input, outputs=X)
    return model
```

TensorFlow

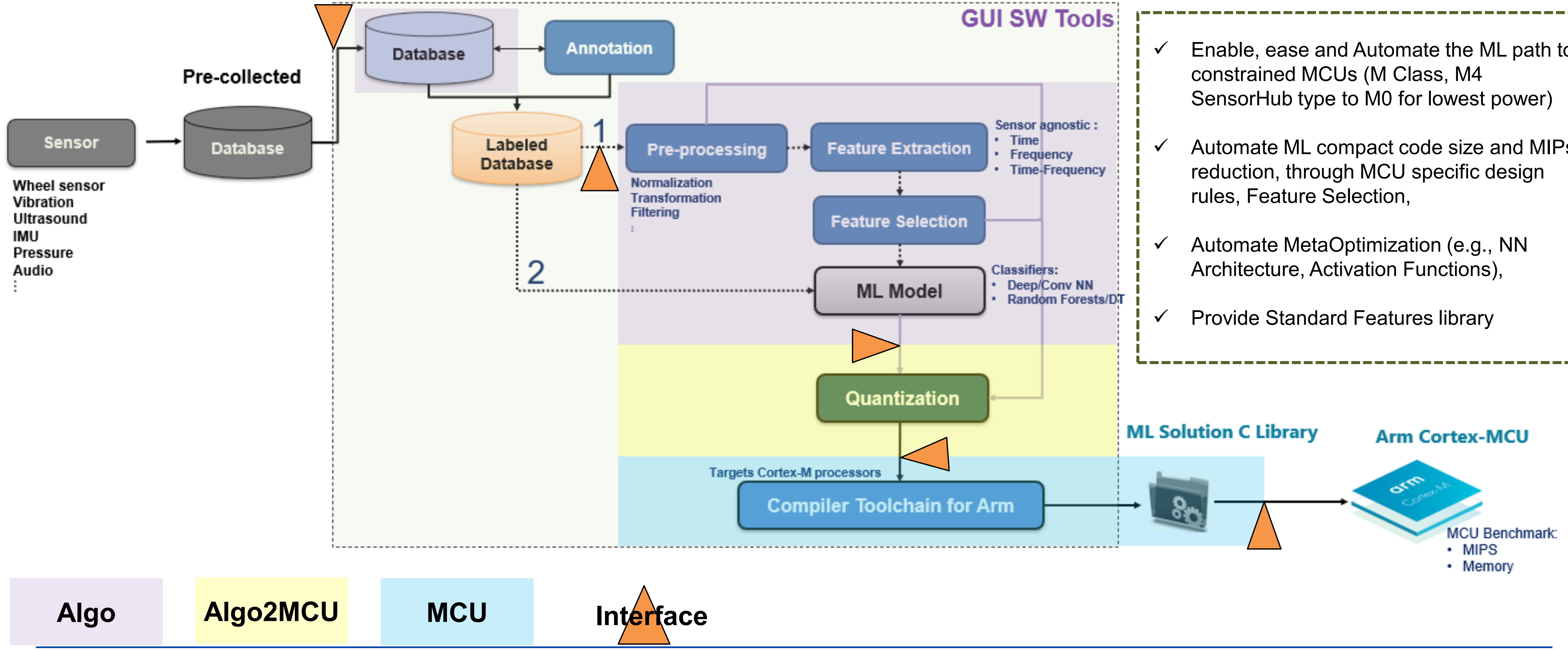
Tiny & Easy

```
int32_t ML2MCU_Predict_DNN(int32_t* pData,int32_t* pred_prob,int32_t* prediction)
{
    Conv1DForward(pData, pWork1, &conv1d_layer_0);
    pData=pWork1;
    BatchNormalization(pData,iBatchScaleQ8_2,iBatchOffsetQ8_2,1,48);
    QuantizedRelu(pData,48);
    GRUForward(pData, pWork2, &gru_layer_4);
    pData=pWork2;
    BatchNormalization(pData,iBatchScaleQ8_5,iBatchOffsetQ8_5,1,40);
    GRUForward(pData, pWork1, &gru_layer_6);
    pData=pWork1;
    BatchNormalization(pData,iBatchScaleQ8_7,iBatchOffsetQ8_7,1,40);
    GlobalGain(pData,iGainQ8_8,40);
    VectorMatrixMult(pData, pWork2, pWeights_8,40,60);
    AddBias(pWork2,pBias_8,60);
    pData=pWork2;
    QuantizedRelu(pData,60);
    GlobalGain(pData,iGainQ8_10,60);
    TimeMultDistributed(pData, pWork1, pWeights_10,60,1,1);
    TimeBiasDistributed(pWork1,pBias_10,1,1);
    pData=pWork1;
    SmoothSigmoidPiecewiseDeg2(pData,1);
    CopyTo(pData,pred_prob,1);
    Threshold(pData, 128,1);
    CopyTo(pData,prediction,1);
    return 0;
}
```

C Integer Code

Machine Learning to Tiny MCU Flow

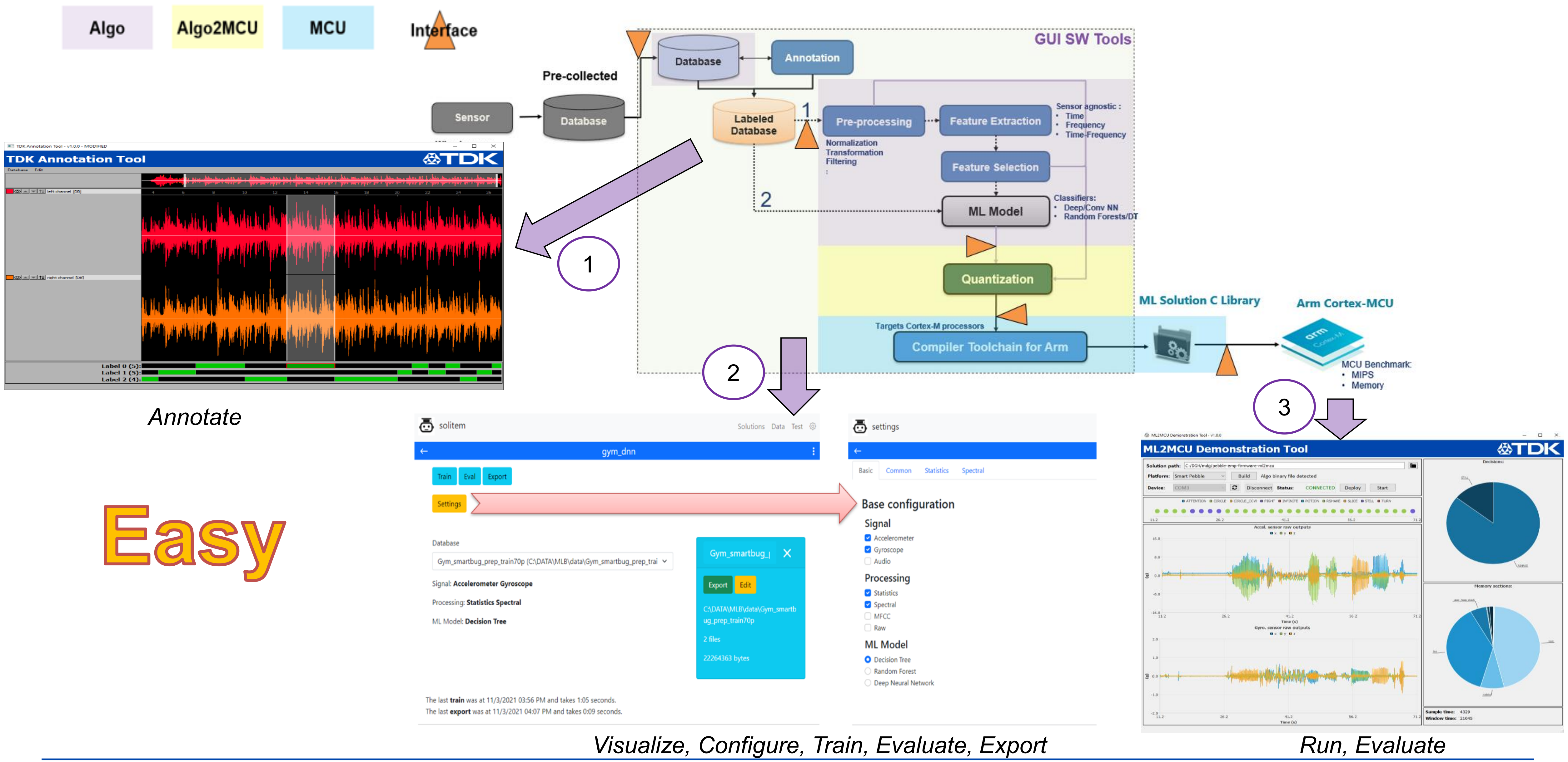
Attracting Tomorrow TDK



- ✓ Enable, ease and Automate the ML path to constrained MCUs (M Class, M4 SensorHub type to M0 for lowest power)
- ✓ Automate ML compact code size and MIPs reduction, through MCU specific design rules, Feature Selection,
- ✓ Automate MetaOptimization (e.g., NN Architecture, Activation Functions),
- ✓ Provide Standard Features library

Automation & Tools

Attracting Tomorrow TDK



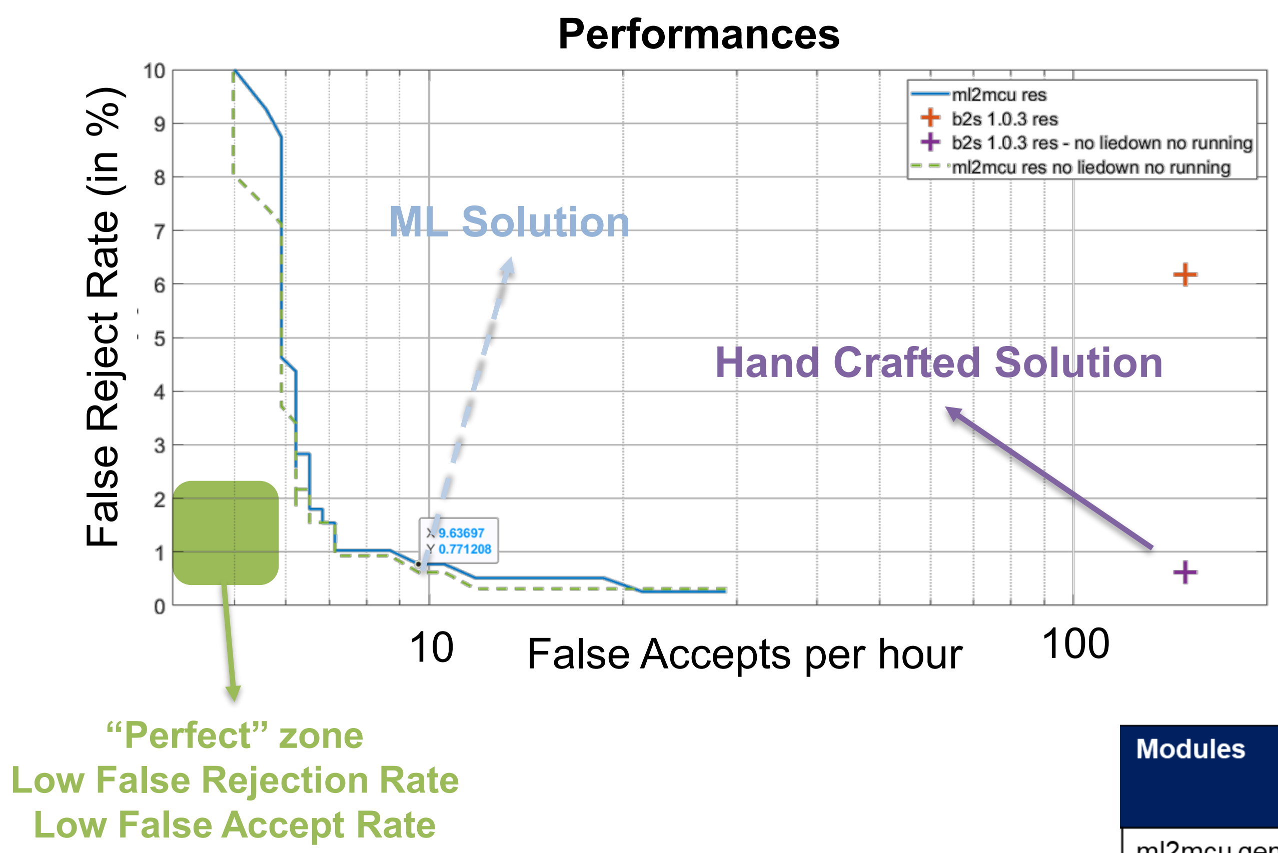
Smart Watch Example

Bring to See

Bring2See for a SmartWatch: FAR/FRR

Attracting Tomorrow

- Bring 2 See is to wakeup the Smart Watch Host MCU



- Uses raw data
- Improved latency of 95ms versus handcrafted

Motion Sensor
32KB/4MHz Clock

Modules	ROM Size (bytes) ro code + ro data	RAM Size (bytes) rw data	Execution time *(eq on M0+ @4Mhz Pebble for 50 ms frames)*
ml2mcu general functions	1202	0	**3.2ms**
b2s model and feature extraction	918	416	
Total:	**2210**	**416**	

Bring2See for a SmartWatch: Power

Attracting Tomorrow TDK

- Assuming 1066mWh battery

Smartwatch	mW
Sleeping	15
Dozing	32
Awake (2%)	310
Average	25.9

https://userpages.umbc.edu/~nroy/courses/fall2018/cmisr/papers/apneaapp_MobiSys15.pdf

- HandCrafted (State Machine) VS Machine Learning False Accept Rate power Impact
  - HandCrafted optimized for User experience
  - Machine Learning point positioned with same FRR (same user experience)

	FA Rate	mW drained due to False Accepts	Battery Life (hours)
ML	10 per hour	42.1	35.3
HandCrafted	100 per hour	30.2	25.3

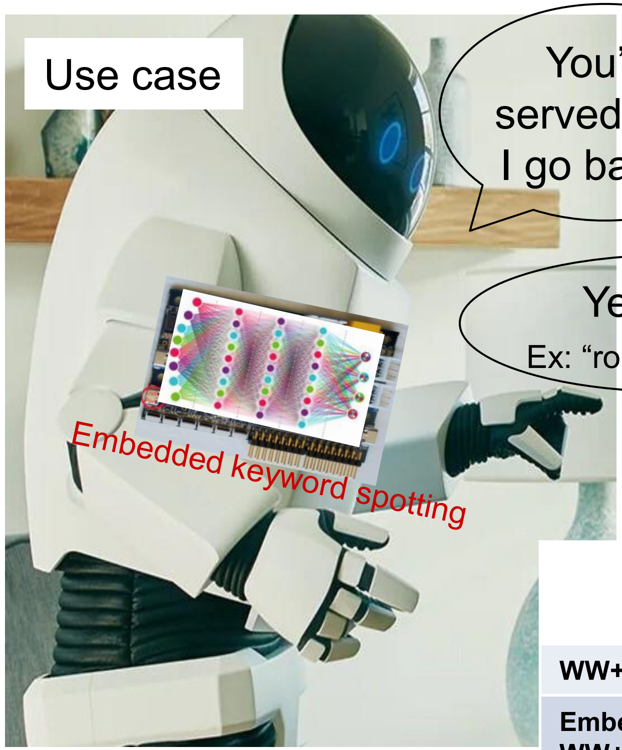
Robotics Example

Audio Wake up and Commands

Low power embedded keyword spotting for robots

Attracting Tomorrow TDK

Use case



You're served, can I go back ?

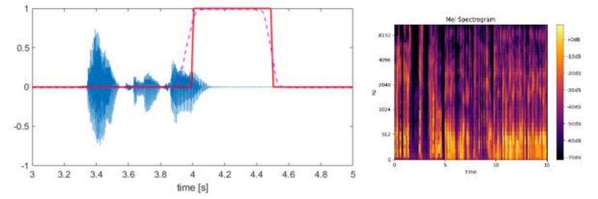
Yes* / no*
Ex: "robot-chan iiyo!"

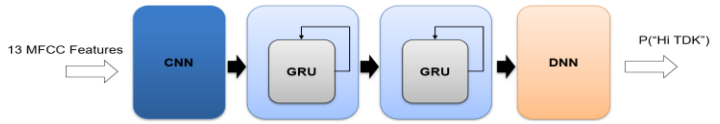
Inputs & built dataset

- 300+ hours of data

Results & performance

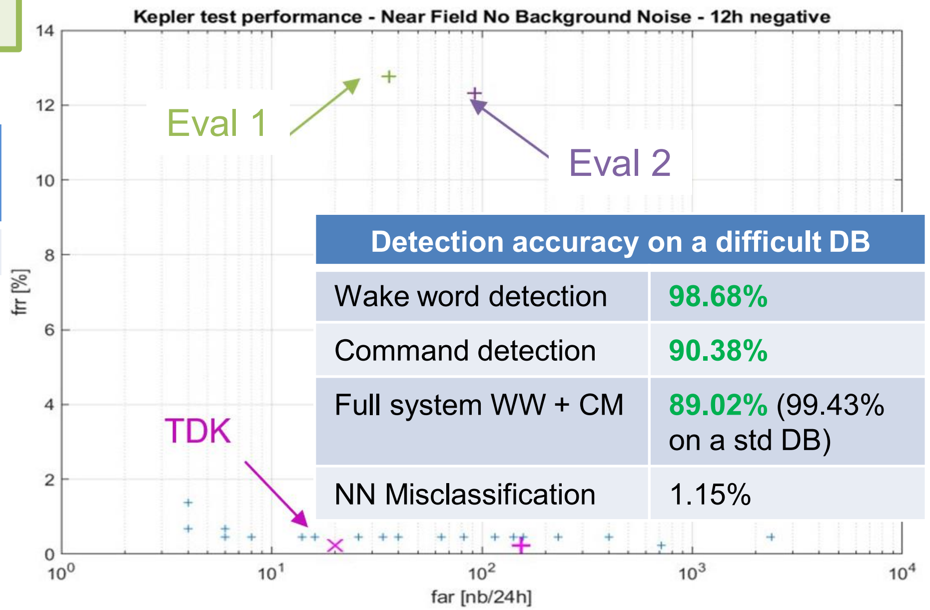
- Fully embedded solution
- 16KHz Mic ODR
- 1 decision every160 ms





MCU-based versus Cloud
- ✓ Improved latency (250x)
- ✓ Reduced Power (80x)

	Memory			Exe time/frame		Extra cloud
	Flash	RAM	Stack			
WW+Cloud	~46KB	~8.5KB	~1KB	0.92ms		2.5 s
Embedded WW+CW	~37.5KB	~9.8KB	~1.3KB	0.48ms	0.5ms	



Detection accuracy on a difficult DB	
Wake word detection	98.68%
Command detection	90.38%
Full system WW + CM	89.02% (99.43% on a std DB)
NN Misclassification	1.15%

Weeks to customize Robokit's KeyWord Spotting
Use Case definition - Audio samples data collection
Audio samples check and post-processing - Data augmentation
Neural Network training - System integration
Performance Tests

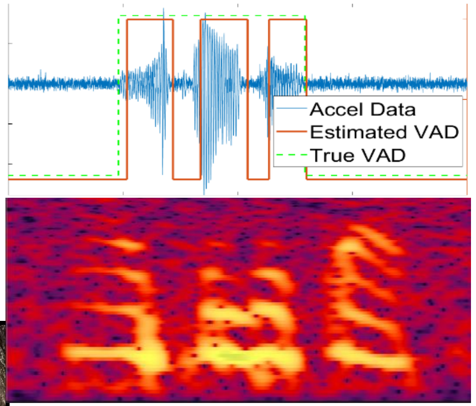
TWS Example

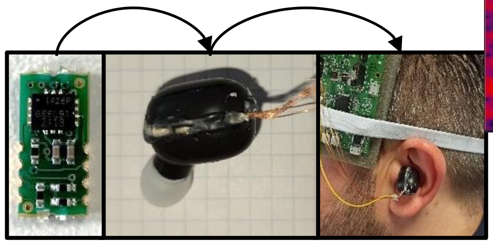
Low power Audio staged detection

Audio and Voice Activity Detector

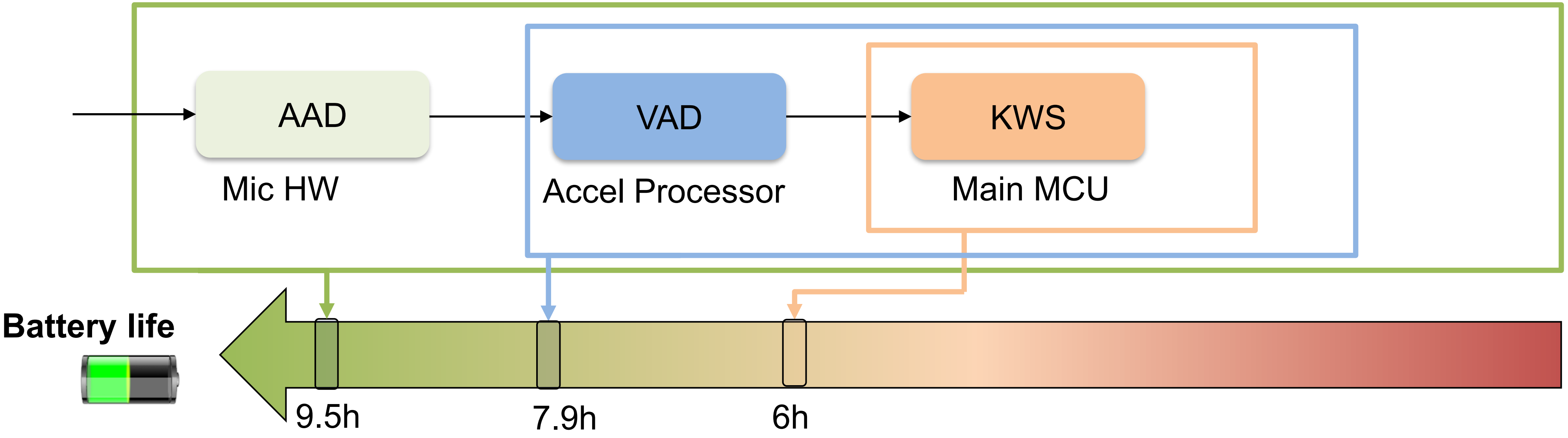
- Audio activity detector is Microphone-based responsible to detect any sound event/activity
  - Runs in HW in extremely low power; First stage for voice detection
- Large band Accelerometer inside TWS or headphone to detect periods of active speech/voice
  - Runs in motion sensor chip MCUSS in very low power; a stage enabling keyword spotting (wake-up word)
- Keyword spotting based on microphone to detect wake-up word
  - Runs on sensor hub, or host MCU;

Recorded Example





TWS application example (battery 100mAh)



Conclusions

- Sensors are the edge to the real world
  - Smallest sentinels linked to bigger components
    - Are ~10kB, Not much MIPs,
    - Our definition of "Tiny"
  - Have a mission to wake up from patterns
    - False Accept cost power,
    - False Rejects cost user experience.
  - Machine Learning can make best use of Data to generate the Wake Up,

- ML brings the promise to save power across all devices without regression of experience
  - We're pushing ML to the "Tiny" level that will improve user experience / and save power

- The Critical Step in this process is the Quantizer / Compress the Network
  - Our Challenge is "How to bring a unified Quantizer",
    - Easy to use (automated)
    - Generates Integer and HW accelerated logic
    - Covers a multitude of NN Architectures

www.invensense.com

TINY ML SUMMIT

tinyML Summit 2022 Sponsors

ANALOG DEVICES AHEAD OF WHAT'S POSSIBLE™
AONdevices
arm
ASPINITY
brainchip The Neuromorphic Computing Company
CEVA®

Deeplite
EDGE IMPULSE
emza visual sense
FotaHub
GREENWAVES TECHNOLOGIES
Grovety Inc.

Himax
HOTG
imagimob
infineon
itemis
KLIKA TECH GLOBAL IOT SOLUTIONS

LatentAI
LATTICE SEMICONDUCTOR
Micro.ai
OmniML
NXP
PXI
Plumerai

PROPHESEE
Qeexo
Qualcomm
Rackner
Reality AI® Engineering Solutions for the Edge
REEXEN technology

RENESAS
SAP
seeed studio The IoT Hardware Enabler
SensiML™
Sony Semiconductor Solutions Corporation
ST life.augmented
SA STREAM ANALYZE

synaptics®
SynSense
SYNTIANT
Tensil.ai
TensorFlow
XMOS

Copyright Notice