

# Tech Talks LIVE Schedule – Presentation will begin shortly



## Wireless Connectivity Tech Talks



Thursday, July 15<sup>th</sup>

Get to Know OpenThread Resources and Examples

Thursday, August 26<sup>th</sup>

Understand the Benefits of Wi-SUN for Long Range Industrial Applications

Wednesday, September 29<sup>th</sup>

Learn to use Machine Learning for Predictive Maintenance

Recording and slides will be posted to:  
[www.silabs.com/training](http://www.silabs.com/training)

We will begin in

**3:00**



## Speaker



水谷章成 ( Aki Mizutani )

Sr. FAE, Japan



# WELCOME

Learn to add Speech Recognition  
with Machine Learning

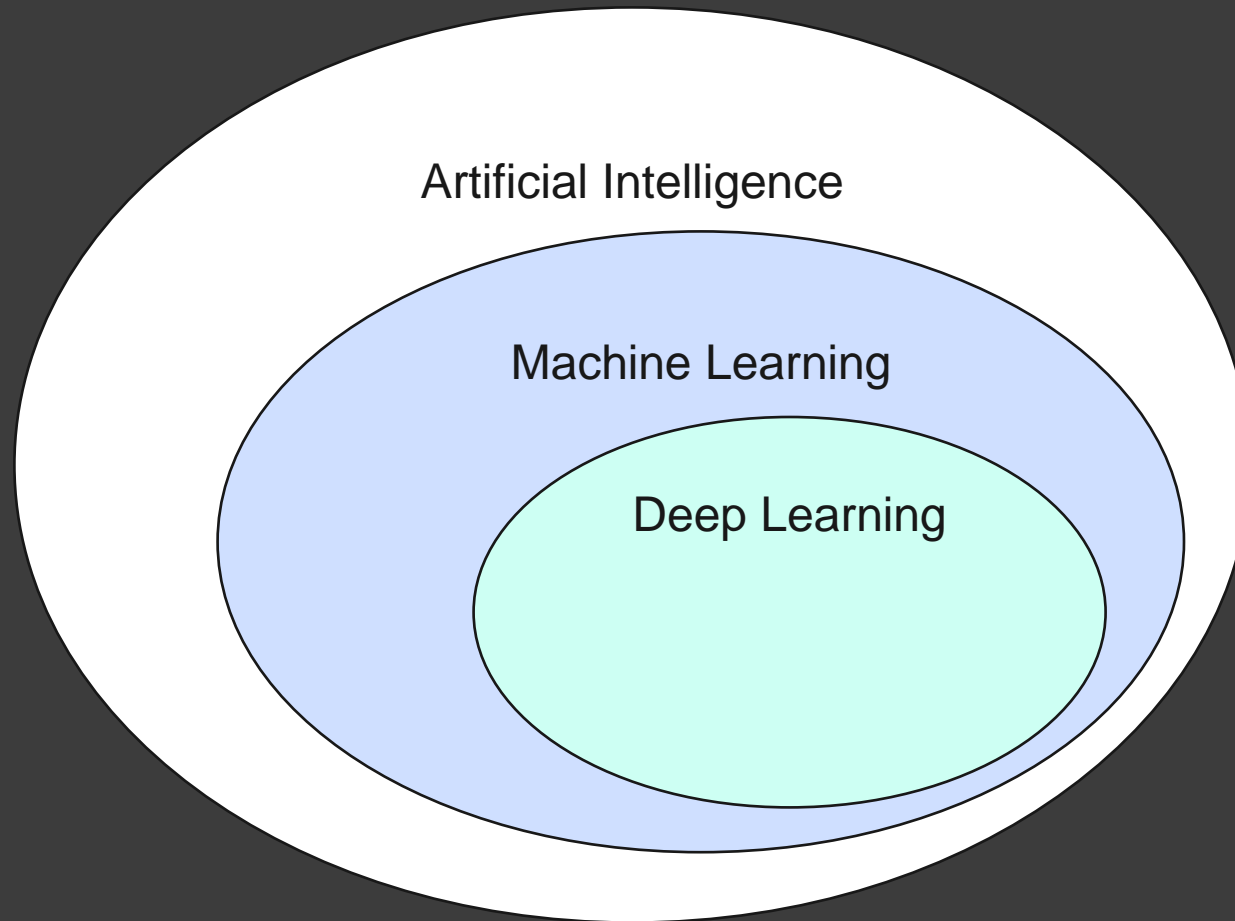
Aki Mizutani



# Agenda

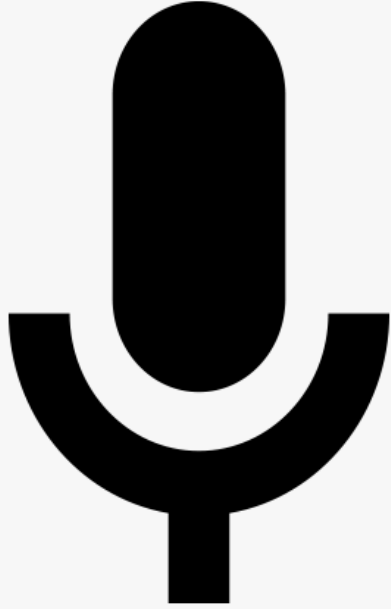
- **A brief introduction to Machine Learning on Embedded Devices**
- **Introduction to Edge Impulse and a high level overview on how to use their platform to create a neural network**
- **Demonstration: “wireless gecko” running on a Thunderboard Sense 2**
- **Wrap-up and Q&A**

# Artificial Intelligence, Machine Learning, and Deep Learning



- **Artificial Intelligence:** Broadly defined as the effort to automate intellectual tasks normally performed by humans
- **Machine Learning:** A system which outputs predictions based on previous observations
- **Deep Learning:** A subset of Machine Learning in which the learning algorithmic architecture is based on approximations of how a human brain might work.

# Embedded Machine Learning Use Cases



AUDIO

**Wake Word / Key Phrase Detection**  
**Glass Breaking**  
**Intrusion**



SENSING

**Motion Sensing**  
**Intelligent Sensor Fusion**

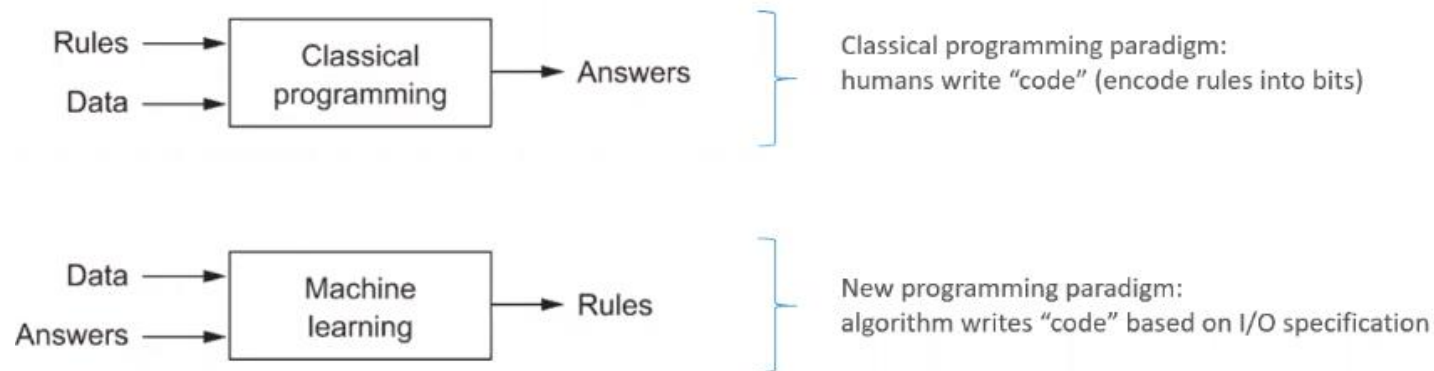


INDUSTRIAL

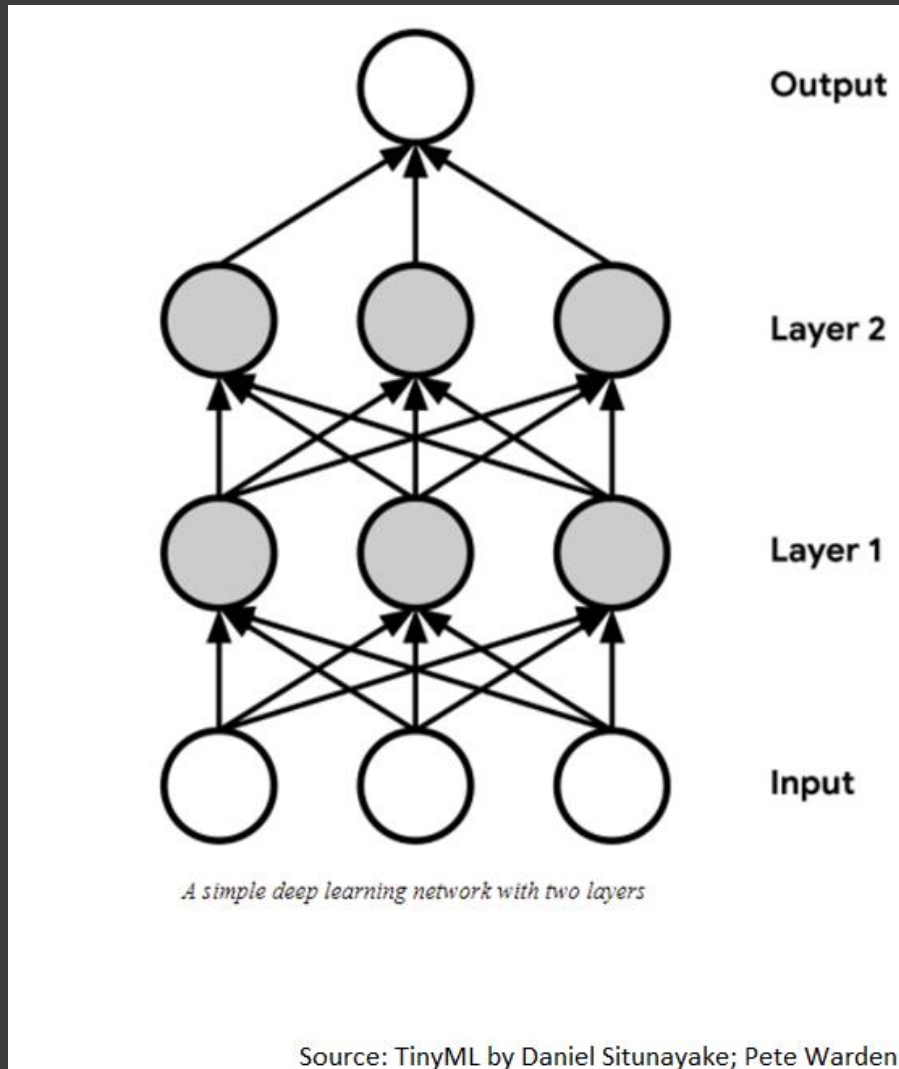
**Predictive Maintenance**

# What is Machine Learning?

- **Definition - Technique for using computers to predict things based on past observations**
- **Collect data – paired input/output**
- **Design Model**
  - Create a computer program that analyzes that data, learns patterns
  - Computer uses the analysis to predict future states from new input
- **Training: Machine Learning model is trained rather than explicitly programmed with rules**
  - Model is presented with many examples of input/output sets
- **Inference: Model finds statistical structure in these examples and produces rules for automating the task**



# Train the Model



## ■ Selecting Data

- Train your model only using information relevant to solving the problem

## ■ Collecting Data

- The more, the more varied, the better!
- 80% for training; 20% for testing

## ■ Labelling Data

- Needed to classify the data

## ■ The idea is to feed training data through a model and make small adjustments

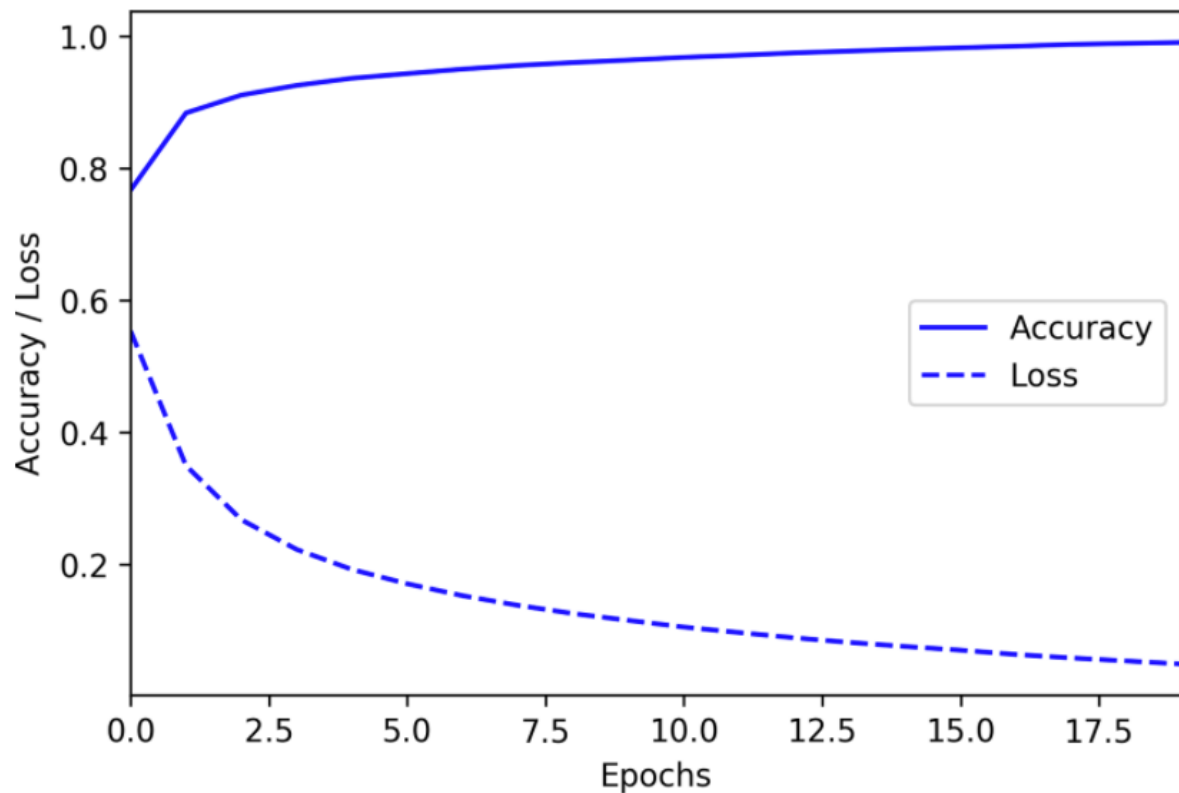
- Weights and biases of each node are initialized with random values

## ■ Training is done in iterations (or epochs) via an algorithm called backpropagation

- Weights and biases are adjusted iteratively
- Stops once the model's performance stops improving: The model "converges"



# Training the Model Cont'd – Accuracy and Loss



*A graph showing model convergence during training*

Source: TinyML by Daniel Situnayake; Pete Warden

- **How to determine if a model has “converged”?**
  - Loss and Accuracy
- **Loss:** gives a numerical estimate of how far the model is from producing the expected answers
- **Accuracy:** percentage of time that the model chooses the correct prediction
- **Perfect Model:** Loss of 0.0 and Accuracy of 100%
- **As training progresses, accuracy increases and loss is reduced until the model no longer improves**

# Training Data Sets

Hot Dog



Not Hot Dog (Unknown)



# Testing and Inferencing



hot dog



NOT hot dog



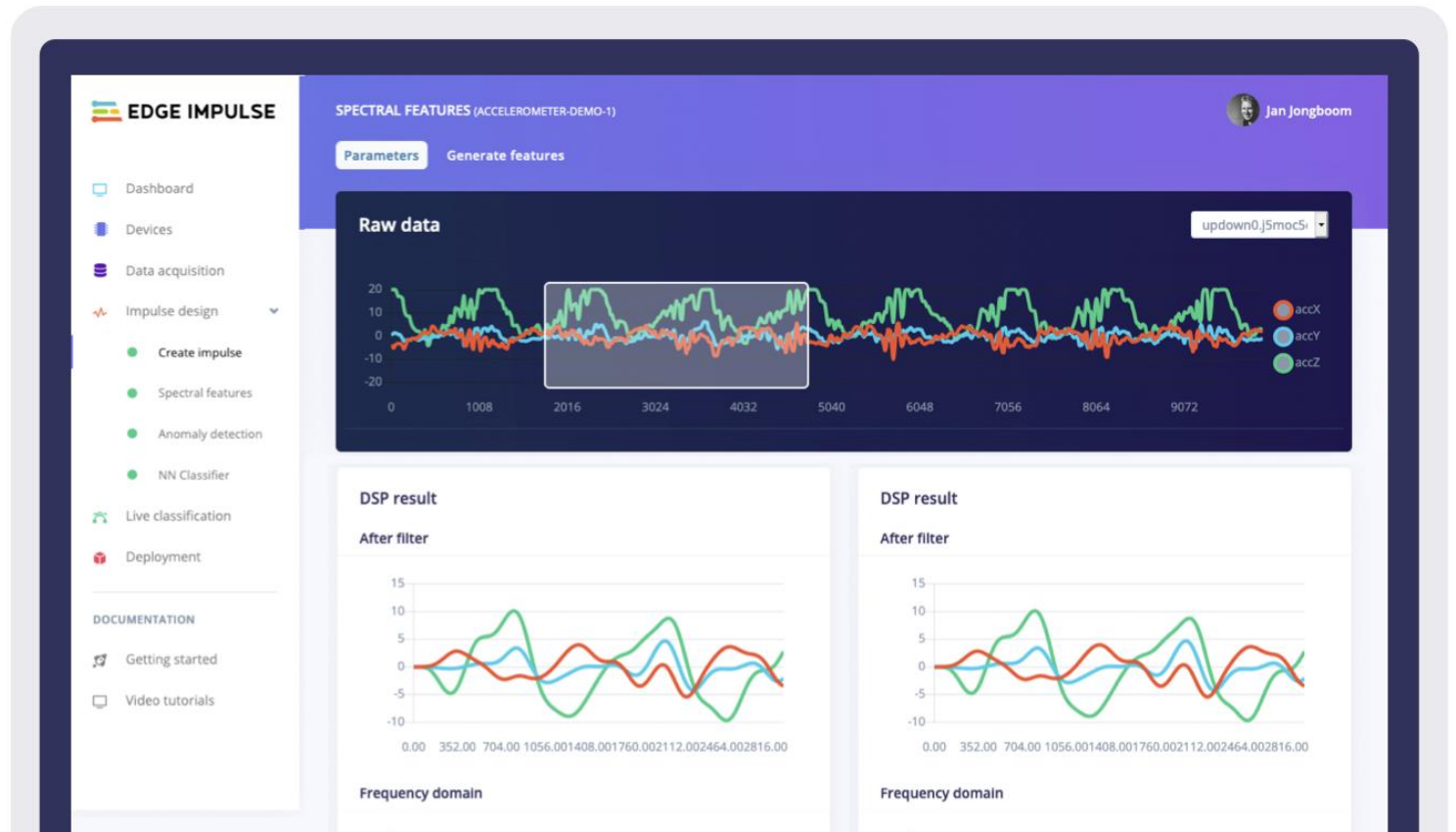
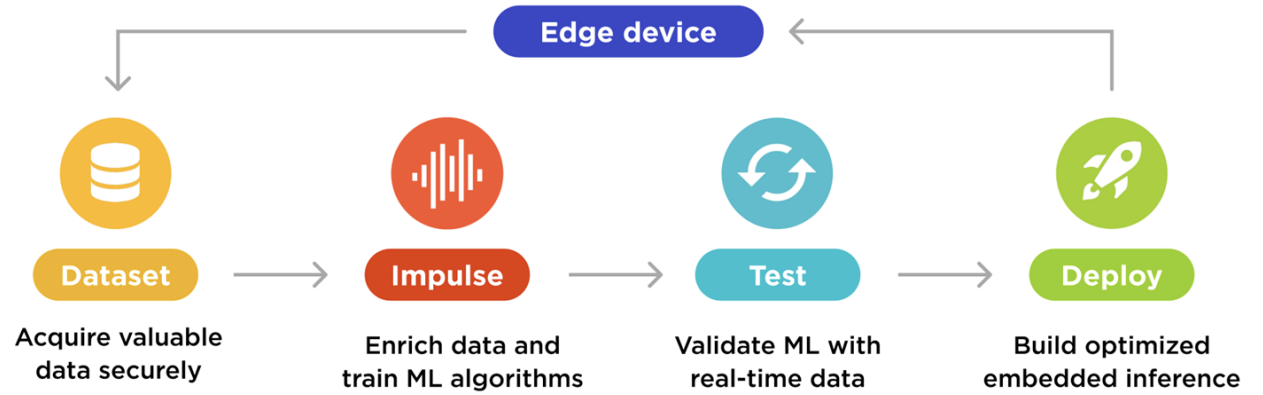
~~hot dog~~

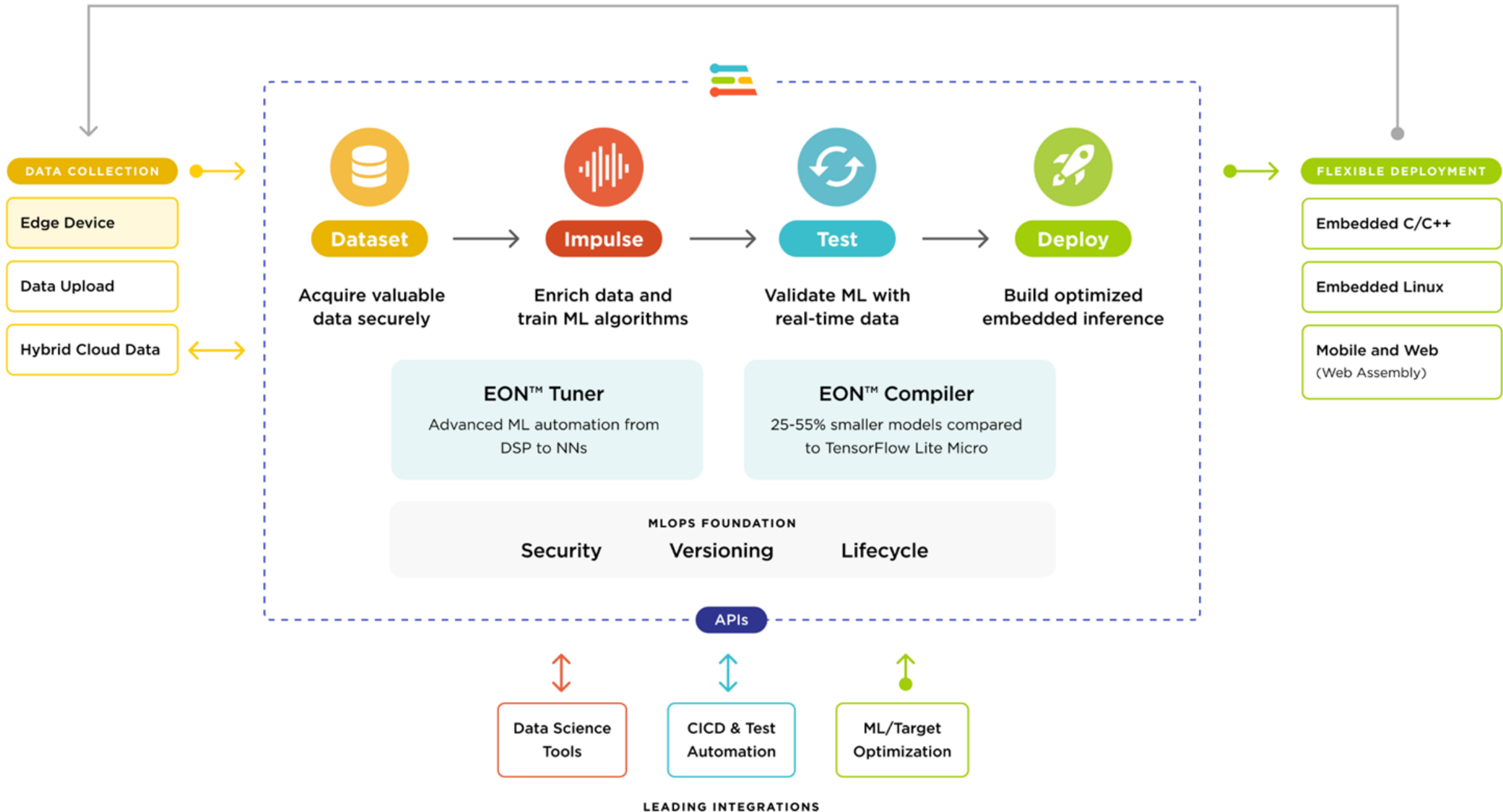
- Reserve 20% of your data set for testing
- A working model will won't be perfect, but should return acceptable results a high percentage of the time
- Poor results?
  - Check the input dataset for miscategorized data
  - Provide more input data
  - 30+ minutes of voice samples not uncommon for typical voice recognition

# The leading embedded ML platform

Learn more at:

<http://edgeimpulse.com>





LEADING INTEGRATIONS

# Closeup: Data Acquisition and Audio Sampling

The screenshot displays the Edge Impulse web interface. On the left is a navigation sidebar with the following items: Dashboard, Devices, Data acquisition, Impulse design, Create impulse, MFCC, NN Classifier, Retrain model, Live classification, Model testing, Versioning, and Deployment. Below these are 'GETTING STARTED' links for Documentation and Forums.

The main content area is titled 'DATA ACQUISITION (WIRELESS GECKO)'. It features two tabs: 'Training data' (selected) and 'Test data'. A notification states: 'Did you know? You can capture data from any device or development board, or upload your existing datasets - Show options'. Two summary cards are present: 'DATA COLLECTED 25m 39s' and 'LABELS 2'. A 'Record new data' button with a 'Connect using WebUSB' option is visible. A message indicates 'No devices connected to the remote management API.'.

The 'Collected data' table lists the following data points:

SAMPLE NAME	LABEL	ADDED	LENGTH
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s
wirelessgecko.299g...	wirelessgecko	Jun 29 2021, 14...	1s

The 'RAW DATA' section shows an audio waveform for 'wirelessgecko.299g1ps5.s5'. The y-axis ranges from -20000 to 30000, and the x-axis shows sample indices from 0 to 932. A play button and a '0:00 / 0:00' timer are at the bottom.

# Closeup: Design the Impulse

The screenshot displays the 'CREATE IMPULSE (WIRELESS GECKO)' interface in the Edge Impulse software. The interface is divided into a left sidebar and a main workspace. The sidebar contains navigation options: Dashboard, Devices, Data acquisition, Impulse design (with sub-items: Create impulse, MFCC, NN Classifier), Retrain model, Live classification, Model testing, Versioning, and Deployment. Below the sidebar is a 'GETTING STARTED' section.

The main workspace is titled 'CREATE IMPULSE (WIRELESS GECKO)' and features a user profile 'Andrew Kr'. A descriptive text box states: 'An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.' The workspace contains four configuration panels:

- Time series data** (red panel):
  - Axis: audio
  - Window size: 1000 ms (with a slider)
  - Window increase: 500 ms (with a slider)
  - Zero-pad data:
- Audio (MFCC)** (white panel):
  - Name: MFCC
  - Input axes:  audio
- Neural Network (Keras)** (purple panel):
  - Name: NN Classifier
  - Input features:  MFCC
  - Output features: 2 (unknown, wirelessgecko)
- Output features** (green panel):
  - 2 (unknown, wirelessgecko)

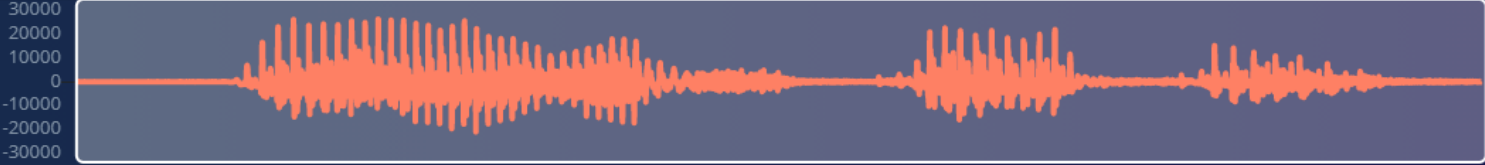
A 'Save Impulse' button is visible on the right side of the workspace.

# Closeup: MFCC and Feature Extraction

MFCC (WIRELESS GECKO) Andrew Kr

Parameters Generate features

**Raw data** 0:00 / 0:00 wirelessgecko.299g1ps5.s5 (wirelessgec)



audio

**Raw features** 📄

16, 10, 17, 26, 28, 31, 32, 8, -4, -2, -13, -23, -34, -38, -31, -41, -39, -46,...

**Parameters**

Mel Frequency Cepstral Coefficients

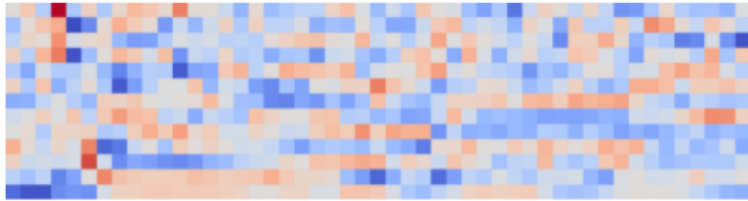
Number of coefficients

Frame length

Frame stride

**DSP result**

Cepstral Coefficients



**Processed features** 📄

-2.3155, -0.3726, 0.7762, 1.5181, 0.3246, 0.0234, -1.2595, 0.7454, -0.5266, 0....



# Closeup: MFCC and Feature Extraction

The screenshot displays the Edge Impulse interface for MFCC (Wireless Gecko). The left sidebar contains navigation options: Dashboard, Devices, Data acquisition, Impulse design, Create impulse, MFCC, NN Classifier, Retrain model, Live classification, Model testing, Versioning, and Deployment. The main area is titled "MFCC (WIRELESS GECKO)" and includes a "Parameters" tab and a "Generate features" button. The "Training set" panel shows the following details:

Parameter	Value
Data in training set	25m 39s
Classes	2 (unknown, wirelessgecko)
Window length	1000 ms.
Window increase	500 ms.
Training windows	1,391

A green "Generate features" button is located at the bottom of the training set panel. The "Feature explorer (1,493 samples)" panel shows a 3D scatter plot with X, Y, and Z axes, each set to "Visualization layer". The plot contains blue dots for "unknown" and orange dots for "wirelessgecko". Below the plot, an audio waveform is shown with a label "Wireless Gecko.wire..." and "Label: wirelessgecko".

# Closeup: Training the NN

## EDGE IMPULSE

- Dashboard
- Devices
- Data acquisition
- Impulse design
  - Create impulse
  - MFCC
  - NN Classifier
- Retrain model
- Live classification
- Model testing
- Versioning
- Deployment

GETTING STARTED

- Documentation
- Forums

### Neural Network settings

#### Training settings

Number of training cycles

Learning rate

Minimum confidence rating

#### Audio training options

Data augmentation

Add noise  None  Low  High

Mask time bands  None  Low  High

Mask frequency bands  None  Low  High

Warp time axis

#### Neural network architecture

Architecture presets  1D Convolutional (Default)  2D Convolutional

- Input layer (650 features)
- Reshape layer (13 columns)
- 1D conv / pool layer (8 neurons, 3 kernel size, 1 layer)
- Dropout (rate 0.25)

### Training output

Model Model version:

Last training performance (validation set)

**ACCURACY** 99.0% **LOSS** 0.03

Confusion matrix (validation set)

	UNKNOWN	WIRELESSGECKO
UNKNOWN	98.8%	1.2%
WIRELESSGECKO	0.8%	99.2%
F1 SCORE	0.99	0.99

Feature explorer (full training set)

- unknown - correct
- wirelessgecko - correct
- unknown - incorrect
- wirelessgecko - incorrect

On-device performance

- INFERRING ... 8 ms.
- PEAK RAM US... 4.4K
- FLASH USAGE 30.0K

# Initial Test Results

- Dashboard
- Devices
- Data acquisition
- Impulse design
  - Create impulse
  - MFCC
  - NN Classifier
- Retrain model
- Live classification
- Model testing
- Versioning
- Deployment

GETTING STARTED

- Documentation
- Forums

This lists all test data. You can manage this data through [Data acquisition](#).

Test data

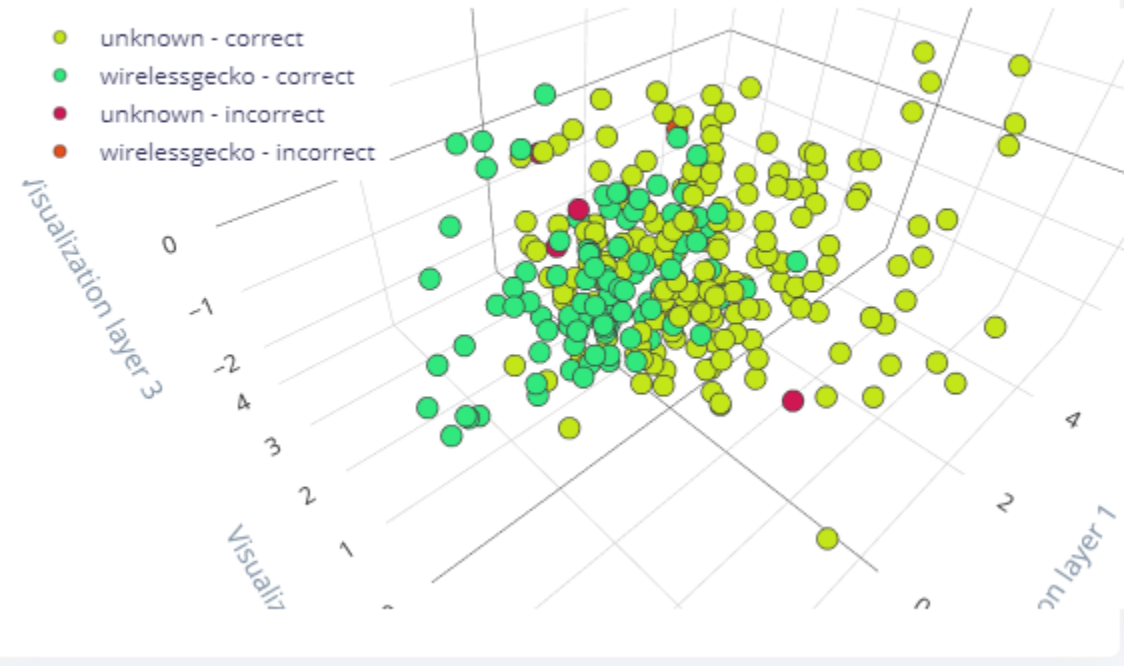
[Classify all](#)

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NA...	EXPECTED OUT...	LENG...	ACCURACY	RESULT
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko
wirelessg...	wirelessgecko	1s	100%	1 wirelessgecko

Feature explorer ?

- unknown - correct
- wirelessgecko - correct
- unknown - incorrect
- wirelessgecko - incorrect



Model te

Files have  
Generating  
Classifyin  
Copying fe  
Copying fe  
Classifyin  
Scheduling  
Job starte  
Classifyin

Job comple

Model te

ACCURACY **97.85%**

	UNKNOWN	WIRELESS...	UNKNOW...	WIRELESS...	UNC
UNKNOWN	97.3%	1.6%	0%	0%	1.1%
WIRELESSGE...	0.6%	55.5%	0%	43.9%	0%
UNKNOWN (...)	-	-	-	-	-
WIRELESSGE...	-	-	-	-	-

# Deployment

## EDGE IMPULSE

Dashboard

Devices

Data acquisition

Impulse design

Create impulse

MFCC

NN Classifier

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

### Select optimizations *(optional)*

Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.



Enable EON™ Compiler

Same accuracy, up to 50% less memory. Open source.



### Available optimizations for NN Classifier

	RAM USAGE	LATENCY	CONFUSION MATRIX		
<b>Quantized (int8)</b> ★ <b>Currently selected</b>	<b>4.4K</b>	<b>8 ms</b>	97.3	1.6	1.1
	FLASH USAGI	ACCURACY	0.8	99.2	0
	<b>30.0K</b>	<b>98.06%</b>			
This optimization is recommended for best performance.					
<b>Unoptimized (float32)</b> <b>Click to select</b>	<b>8.0K</b>	<b>35 ms</b>	97.3	1.6	1.1
	FLASH USAGI	ACCURACY	0.8	99.2	0
	<b>32.4K</b>	<b>98.06%</b>			

Estimate for Cortex-M4F 40MHz (SiLabs Thunderboard Sense 2)

Build

# DEMO

Dashboard - TechTalk\_Demo1 - E X +

studio.edgeimpulse.com/studio/49459

**EDGE IMPULSE**

Project info Keys Export

Aki

# Aki / TechTalk\_Demo1

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

### Creating your first impulse (34% complete)

- Acquire data**  
Every Machine Learning project starts with data. You can capture data from a development board or your phone, or import data you already collected.  
[LET'S COLLECT SOME DATA](#)
- Design an impulse**  
Teach the model to interpret previously unseen data, based on historical data. Use this to categorize new data, or to find anomalies in sensor readings.  
[GETTING STARTED: CONTINUOUS MOTION RECOGNITION](#)  
[GETTING STARTED: RESPONDING TO YOUR VOICE](#)  
[GETTING STARTED: ADDING SIGHT TO YOUR SENSORS](#)
- Deploy**  
Package the complete impulse up, from signal processing code to trained model, and deploy it on your device. This ensures that the impulse runs with

### Sharing

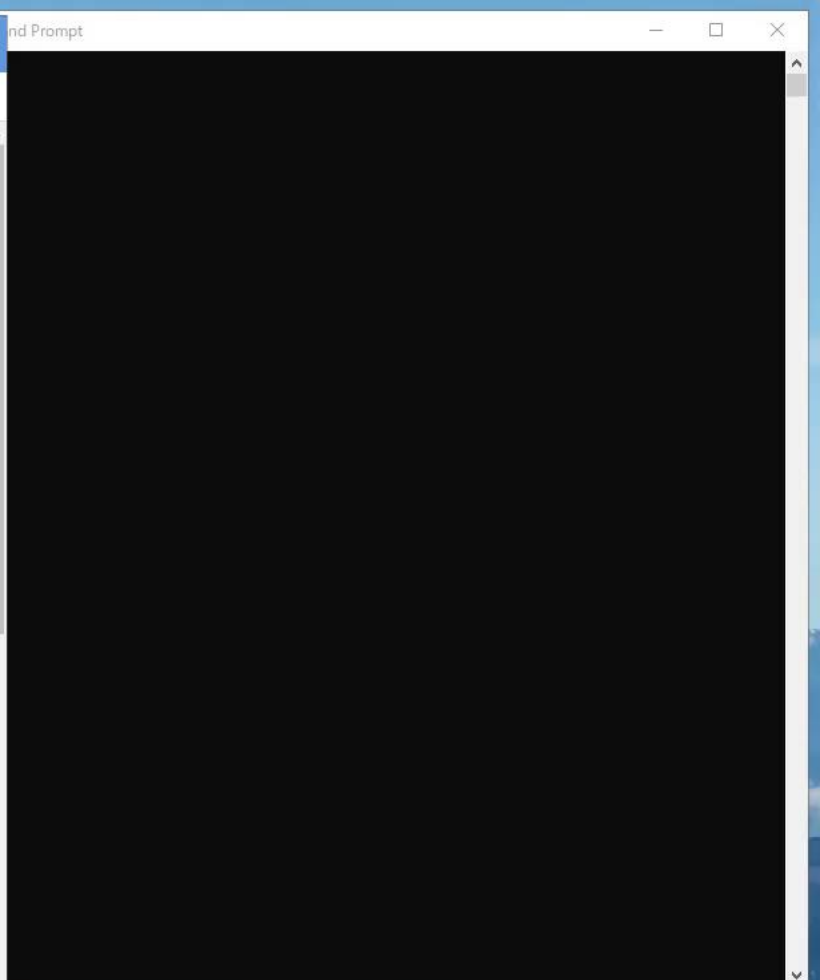
Your project is private.

[Make this project public](#)

### Summary

- DEVICES CONNECTED**  
11
- DATA COLLECTED**  
27m 35s

### Collaborators



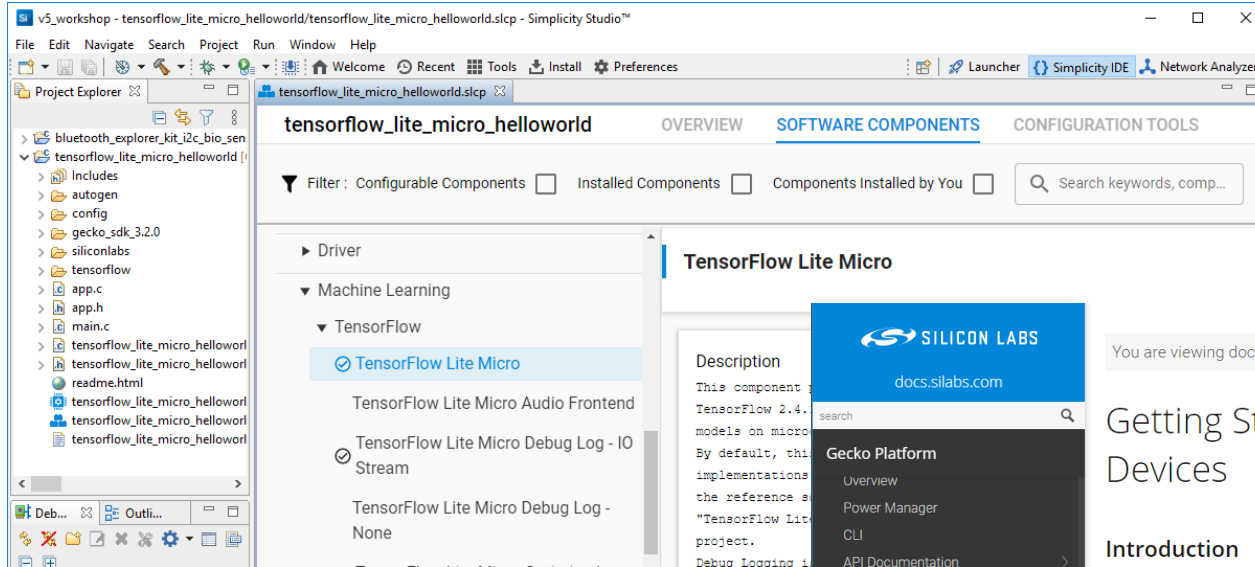
# Testing on Hardware: Thunderboard Sense 2 – SLTB004A



## ■ Thunderboard Sense 2

- **Wireless SoC with multi-protocol radio**
- **Direct support with Edge Impulse**
- **ARM® Cortex® M4 core with 256 kB RAM and 1024 kB Flash**
- **Broad Range of Sensors**
  - 6-axis Inertial Sensor ICM-20648
  - Digital Microphone ICS-43434
  - Pressure Sensor BMP280
  - Indoor Air Quality and Gas Sensor CCS811
  - Relative Humidity and Temperature Sensor Si7021
  - UV and Ambient Light Sensor Si1133
  - Hall-effect Sensor Si7210

# TensorFlow Lite for Microcontrollers



You are viewing documentation for version: 3.2 (latest) | [Version History](#)

## Getting Started with Machine Learning on Silicon Labs Devices

### Introduction

Silicon Labs integrates [TensorFlow](#) as a component within our Gecko SDK and [Project Configurator](#) for our EFX32 series microcontrollers, making it simple to add machine learning capability to any application. This guide covers how to get started using TensorFlow Lite for Microcontrollers on Silicon Labs' EFX32 devices.

### TensorFlow Lite for Microcontrollers

[TensorFlow](#) is a widely used deep learning framework, with capability for developing and executing neural networks across a variety of platforms. [TensorFlow Lite](#) provides an optimized set of tools specifically catered towards machine learning for mobile and embedded devices.

[TensorFlow Lite for Microcontrollers](#) (TFLM) specifically provides a C++ library for running machine learning models in embedded environments with tight memory constraints. Silicon Labs provides tools and support for loading and running pre-trained models that are compatible with this library.

### Gecko SDK TensorFlow Integration

The [Gecko SDK](#) includes TensorFlow as a third-party submodule, allowing for easy integration and testing with Silicon Labs' projects. Note that the included TensorFlow version may differ from the latest release of TensorFlow.

Additionally, [TensorFlow Software Components](#) in the [Project Configurator](#) simplify the process of including the necessary dependencies to use TFLM in a project.

### Developing a Machine Learning Model in TFLM





- **Watch On-demand**

<https://workswith.silabs.com/agenda>



- **Take Survey and Get a FREE Bluetooth Explorer Kit:**





tech **t▶lks**

# Q&A

Facebook



Twitter



Community





# THANK YOU

Recording and slides will be posted to:  
[www.silabs.com/training](http://www.silabs.com/training)

