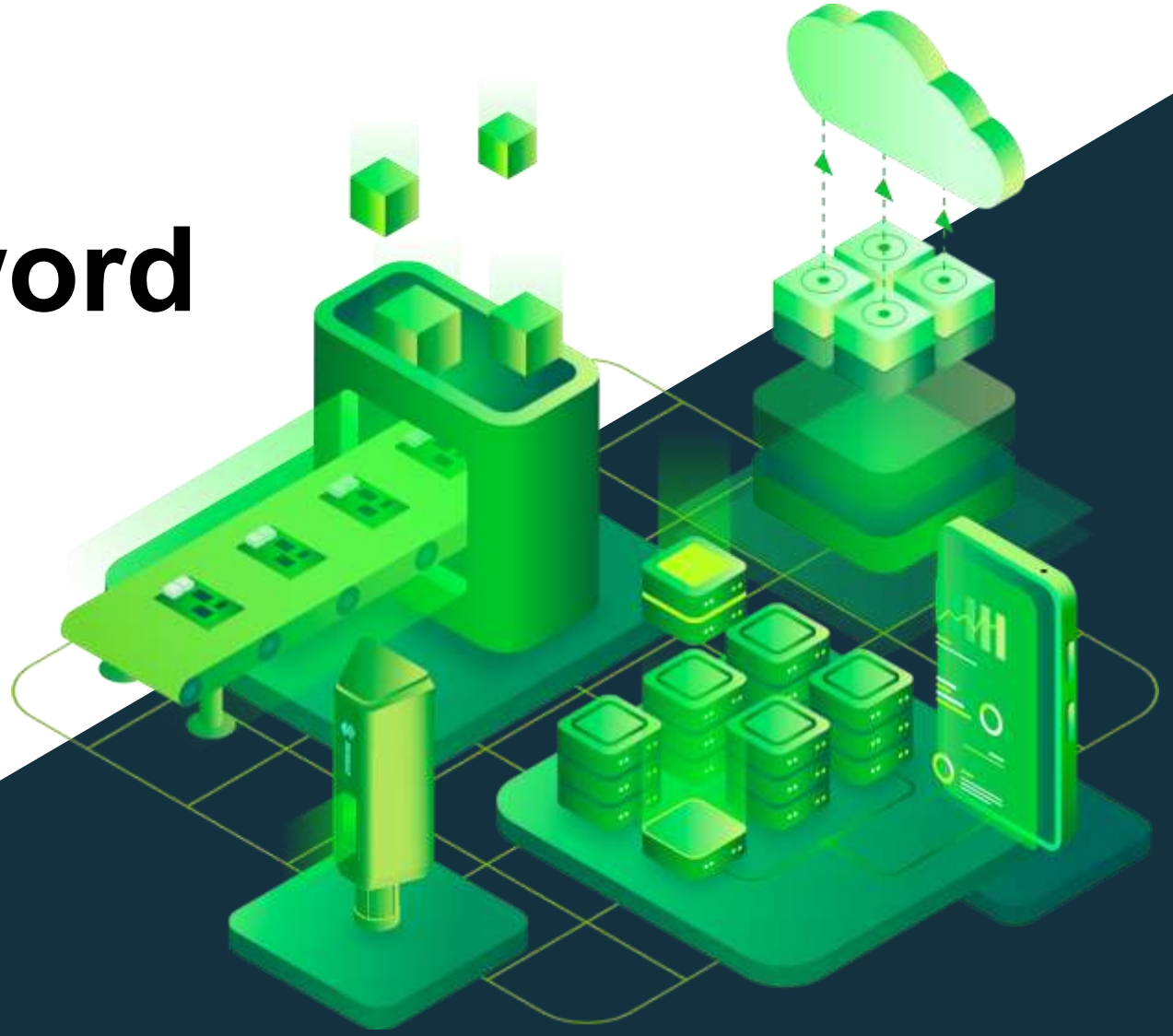
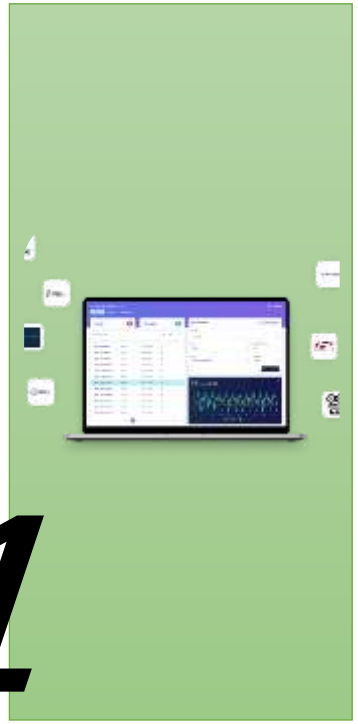
seeed studio

# Hands-On Keyword Spotting

ICTP-UNU Workshop on TinyML
for Sustainable Development

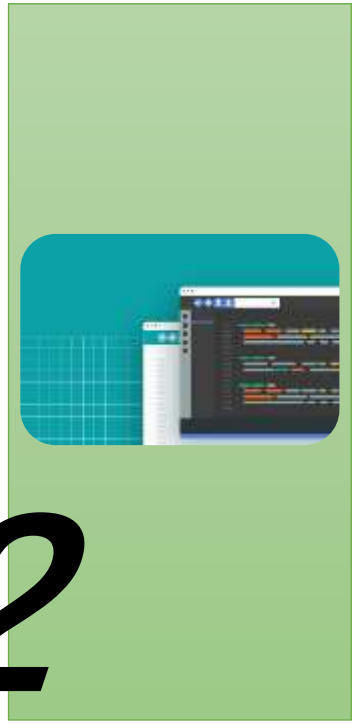


# Main Content



1

Edge Impulse & XIAO ESP32S3 Sense Introduction



2

Recording Audio with XIAO ESP32S3 Sense

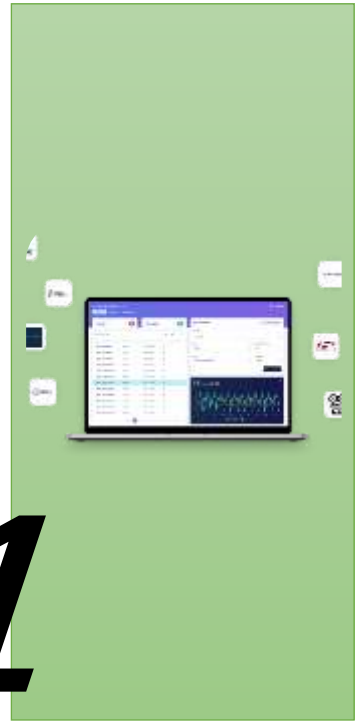

3

Training Exported Models with Edge Impulse

4

Deploying models to XIAO ESP32S3 Sense

# Main Content



1

Edge Impulse &
XIAO ESP32S3 Sense
Introduction

# Edge Impulse & XIAO ESP32S3 Sense Introduction


Edge Impulse is a platform for developing machine learning models specifically designed for edge devices and embedded systems. It provides a comprehensive set of tools and services that enable developers to quickly create, train, and deploy machine learning models without requiring deep expertise in machine learning.

# Edge Impulse & XIAO ESP32S3 Sense Introduction

## *Seeed Studio XIAO ESP32S3 Sense*

Ultra-small ESP32-S3 development board with OV2640 camera, a risging star in the fields of TinyML and Smart Home

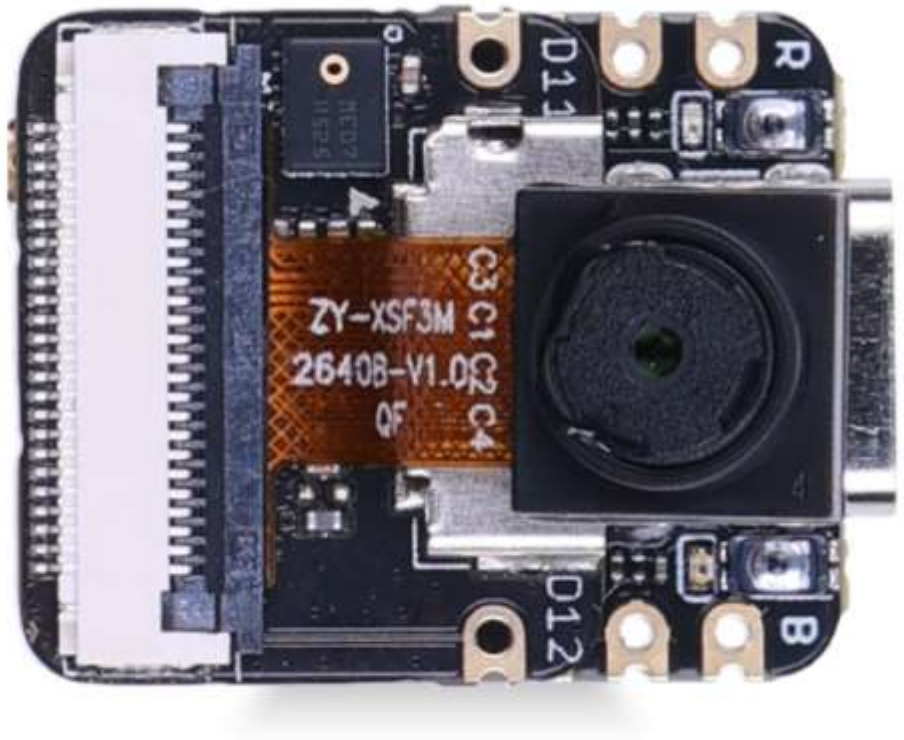
**HIGH-PERFORMANCE**
240MHz Xtensa 32-bit LX7 dual-core processor

**MEMORY**
8MB PSRAM + 8MB FLASH

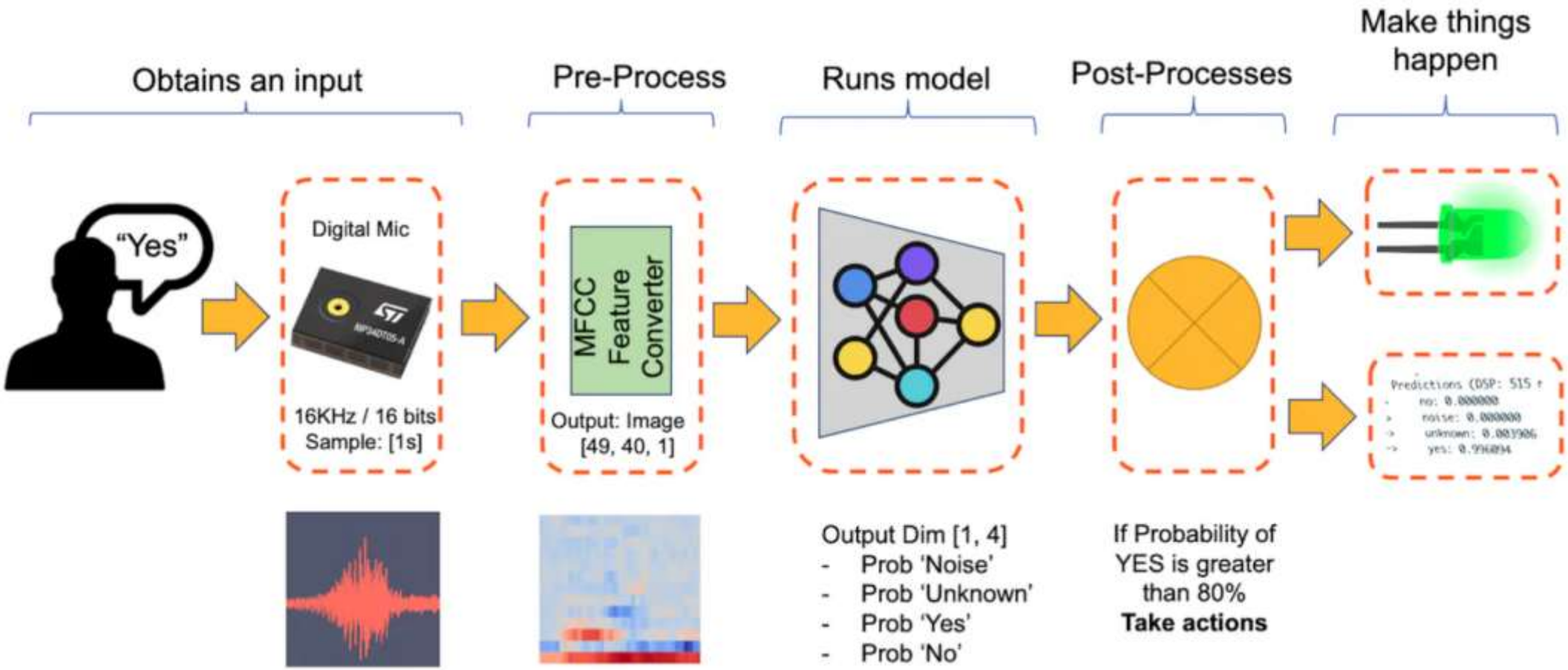
**MULTI-FUNCTIONAL**
Microphone/SD card slot/Detachable OV2640

**WIRELESS**
2.4GHz WiFi and BLE 5

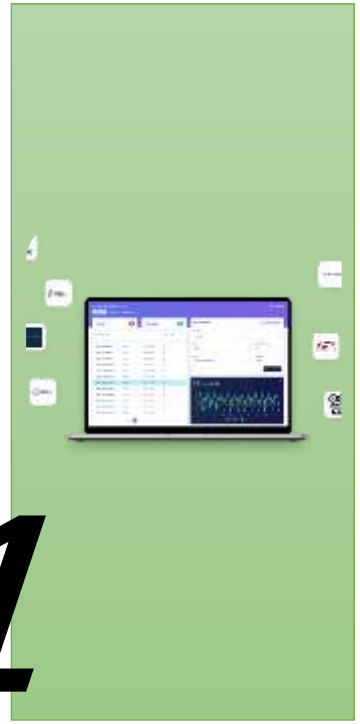
**TINYML-SUPPORTED**
Image Processing/Speech Recognition



# Edge Impulse & XIAO ESP32S3 Sense Introduction



# Main Content



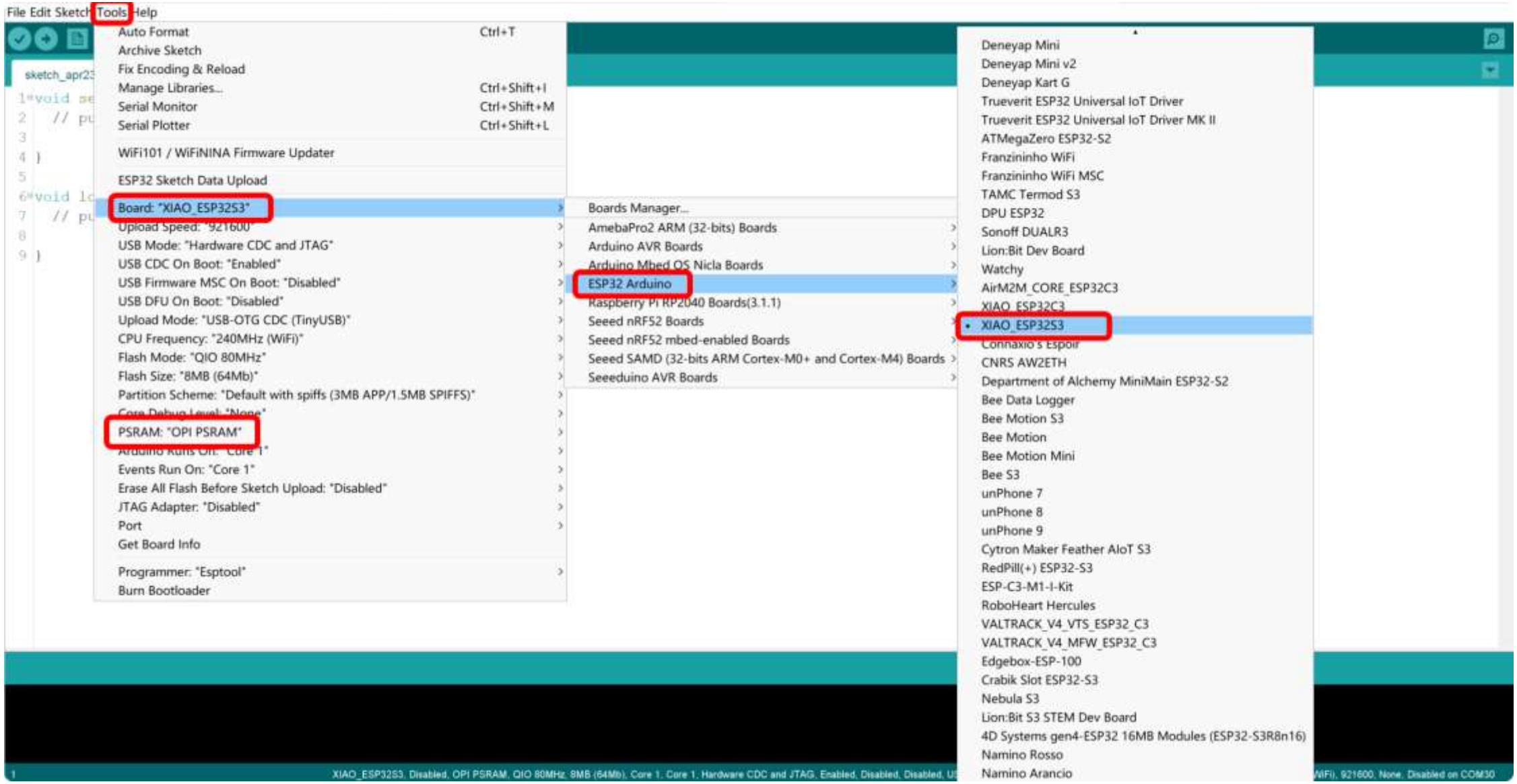
1

Edge Impulse &
XIAO ESP32S3 Sense
Introduction

2

Recording Audio
with XIAO ESP32S3
Sense

# Recording Audio with XIAO ESP32S3 Sense

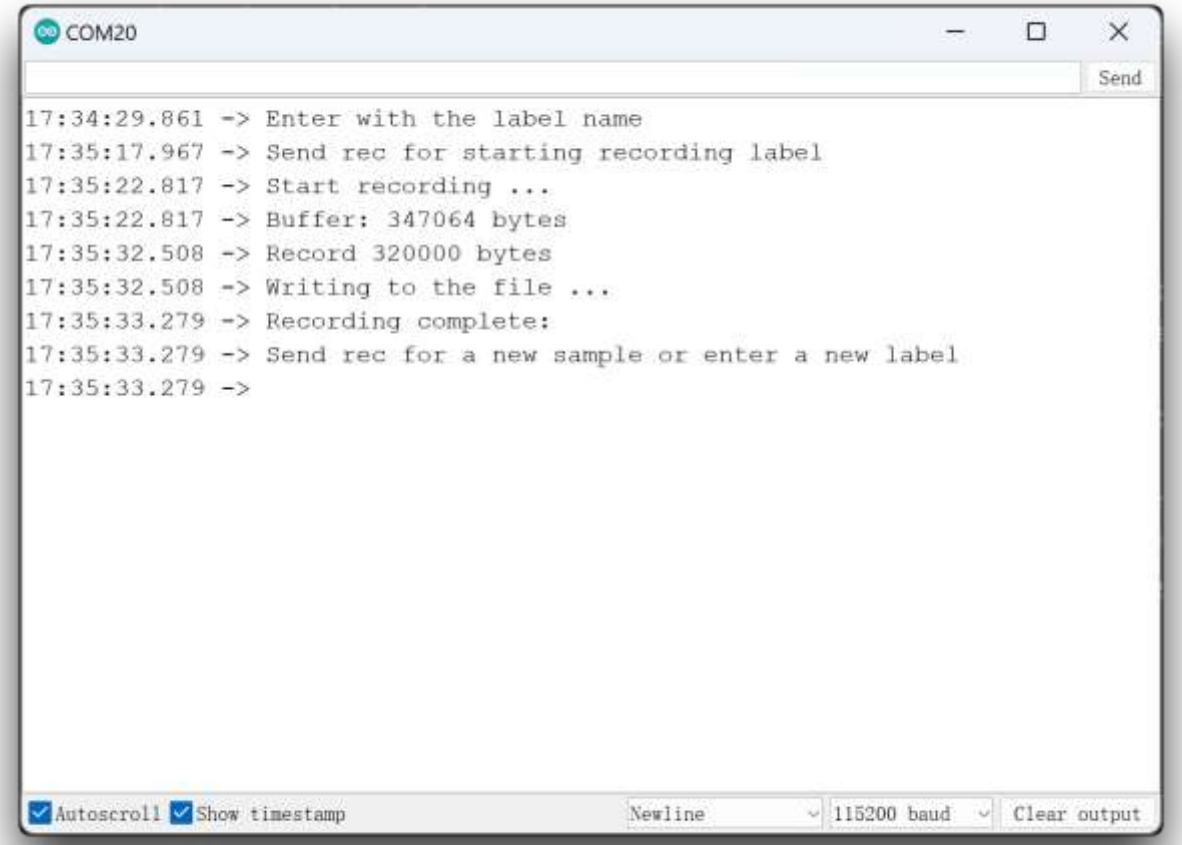


# Recording Audio with XIAO ESP32S3 Sense


Insert the microSD card into the microSD card slot. Please note the direction of insertion, the side with the gold finger should face inward.

# Recording Audio with XIAO ESP32S3 Sense

https://wiki.seeedstudio.com/tinyml_course_Key_Word_Spotting/#step-1-save-recorded-sound-samples-as-wav-audio-files-to-a-microsd-card




Now, upload the code to the XIAO and get samples from the keywords (hello and stop). You can also capture noise and other words. The Serial monitor will prompt you to receive the label to be recorded.

Send the label (for example, hello). The program will wait for another command: rec.

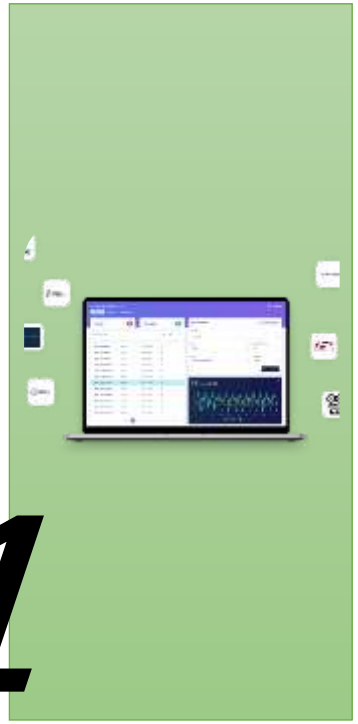
And the program will start recording new samples every time a command rec is sent. The files will be saved as hello.1.wav, hello.2.wav, hello.3.wav, etc. until a new label (for example, stop) is sent. In this case, you should send the command rec for each new sample, which will be saved as stop.1.wav, stop.2.wav, stop.3.wav, etc.

Ultimately, we will get the saved files on the SD card.

# Recording Audio with XIAO ESP32S3 Sense

Use a card reader to save all the sound samples stored inside the SD card to your computer.

# Main Content



1

Edge Impulse & XIAO ESP32S3 Sense Introduction

2

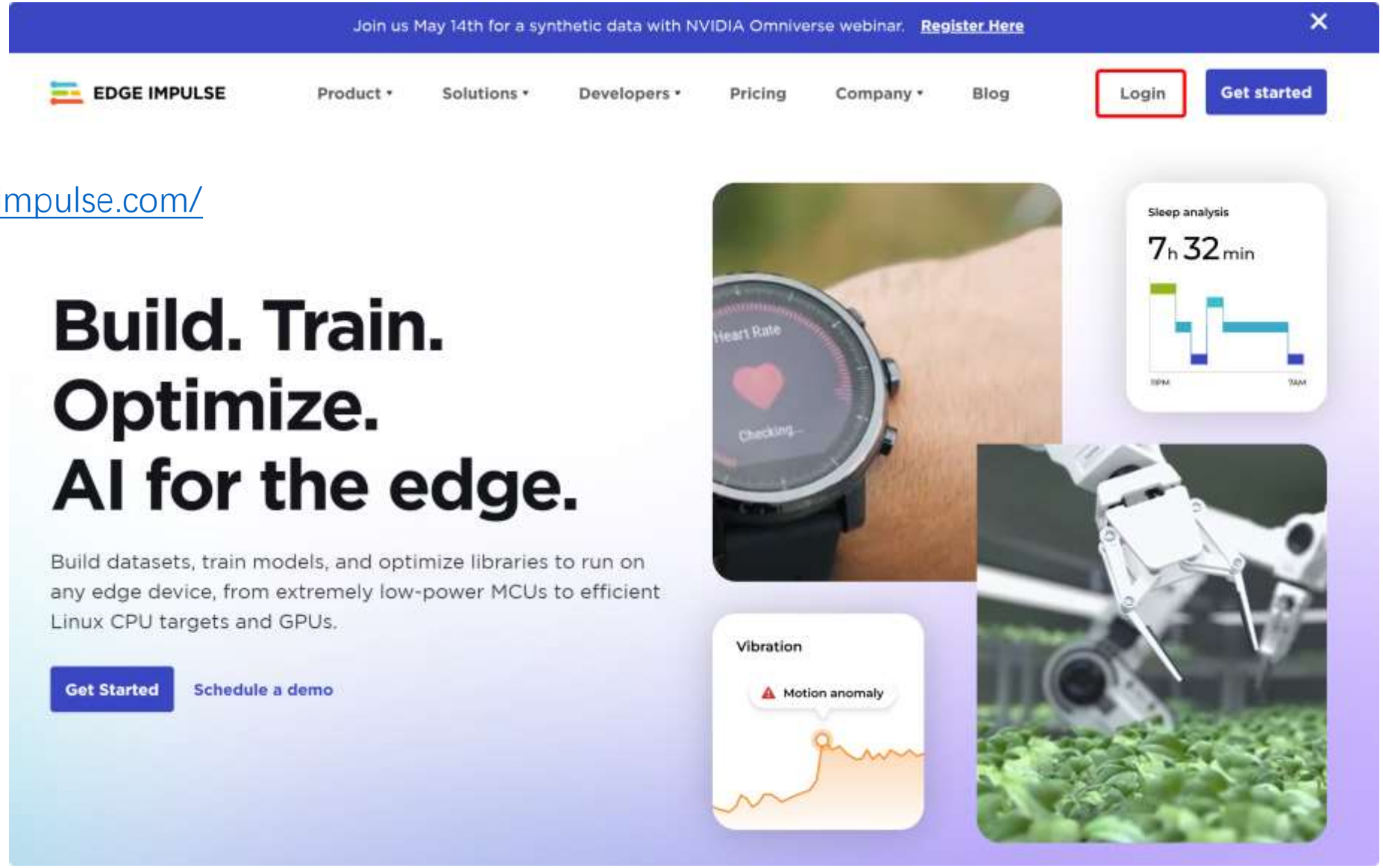
Recording Audio with XIAO ESP32S3 Sense


3

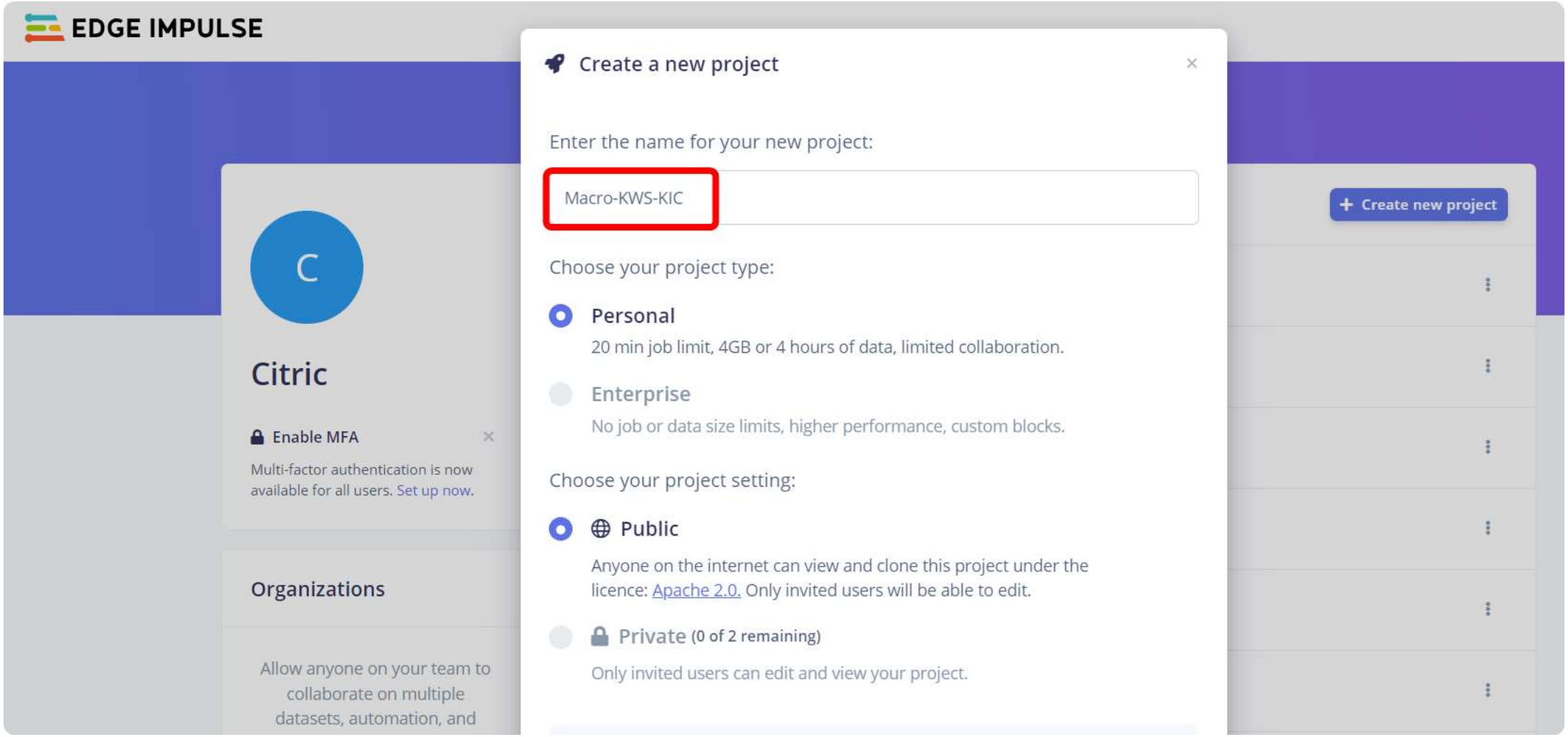
Training Exported Models with Edge Impulse

# Training Exported Models with Edge Impulse

https://edgeimpulse.com/

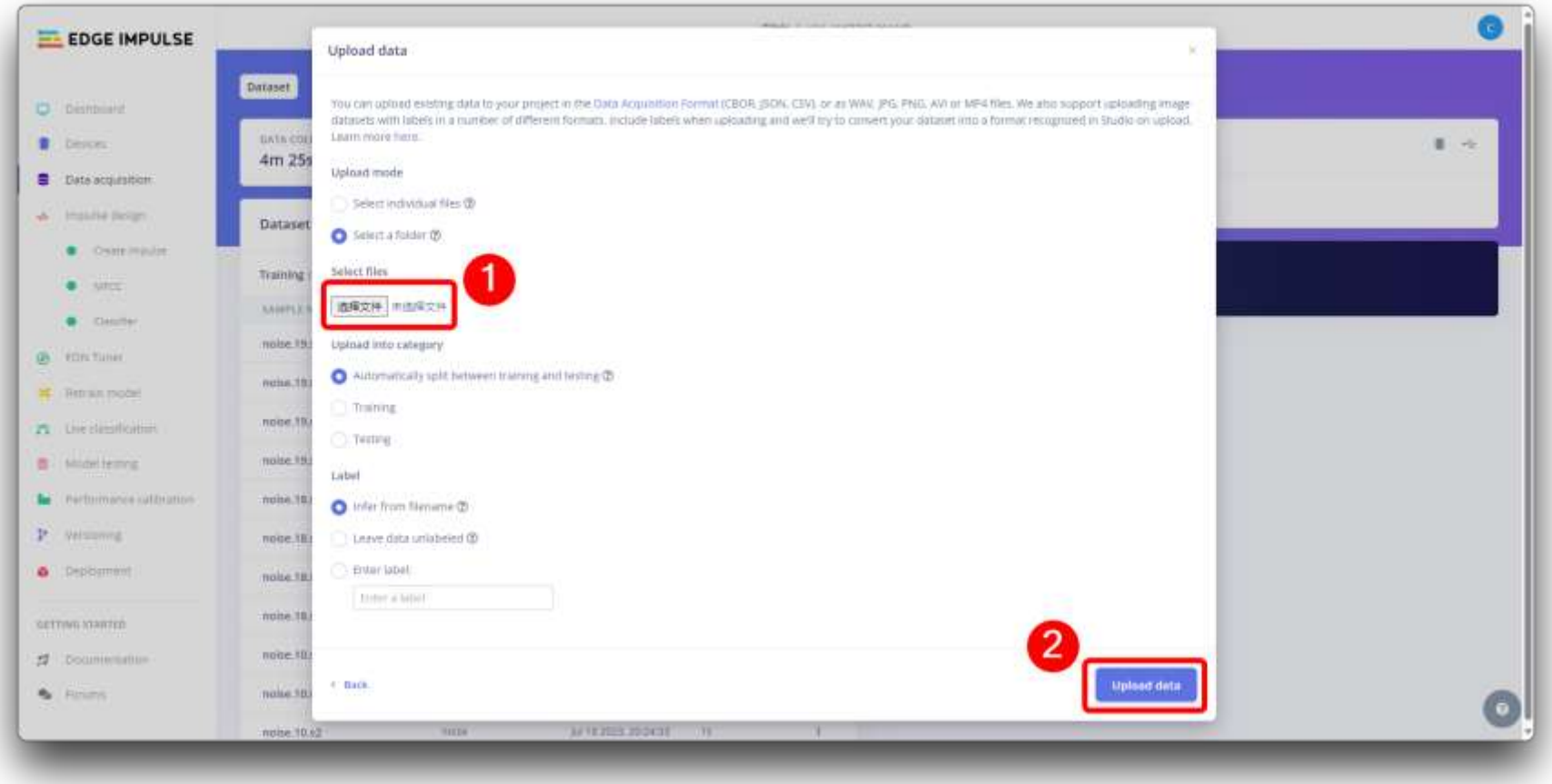


# Training Exported Models with Edge Impulse



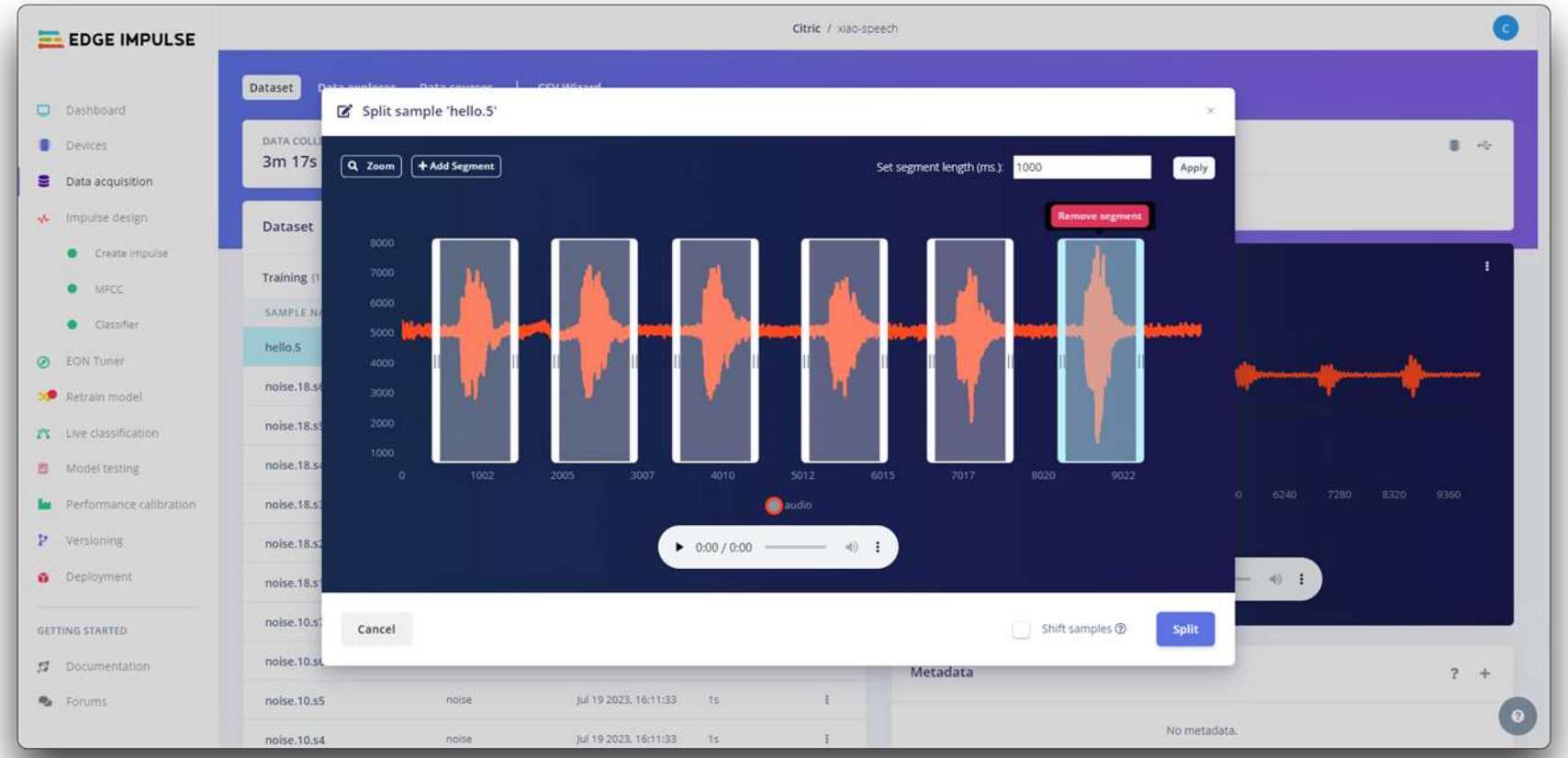

We should initiate a new project at Edge Impulse and give it the same name “Macro-KWS-KIC”.

# Training Exported Models with Edge Impulse




Once the project is created, select the **Upload Data** tool in the **Data Acquisition** section. Choose the files to be uploaded.

# Training Exported Models with Edge Impulse

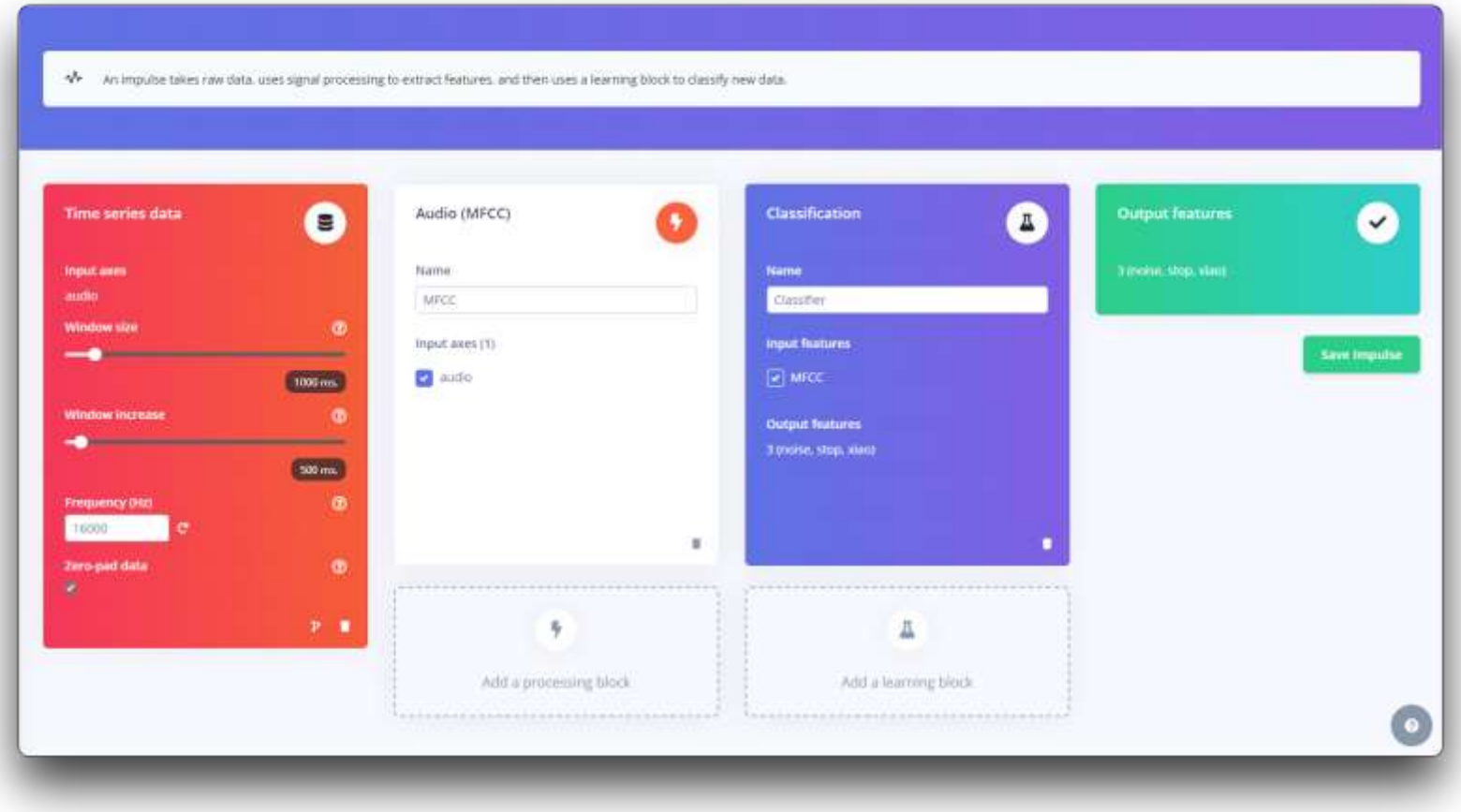



All data on dataset have a 1s length, but the samples recorded in the previous section have 10s and must be split into 1s samples to be compatible. Click on three dots after the sample name and select **Split sample**.

Once inside de tool, split the data into 1-second records. If necessary, add or remove segments.

This procedure should be repeated for all samples.

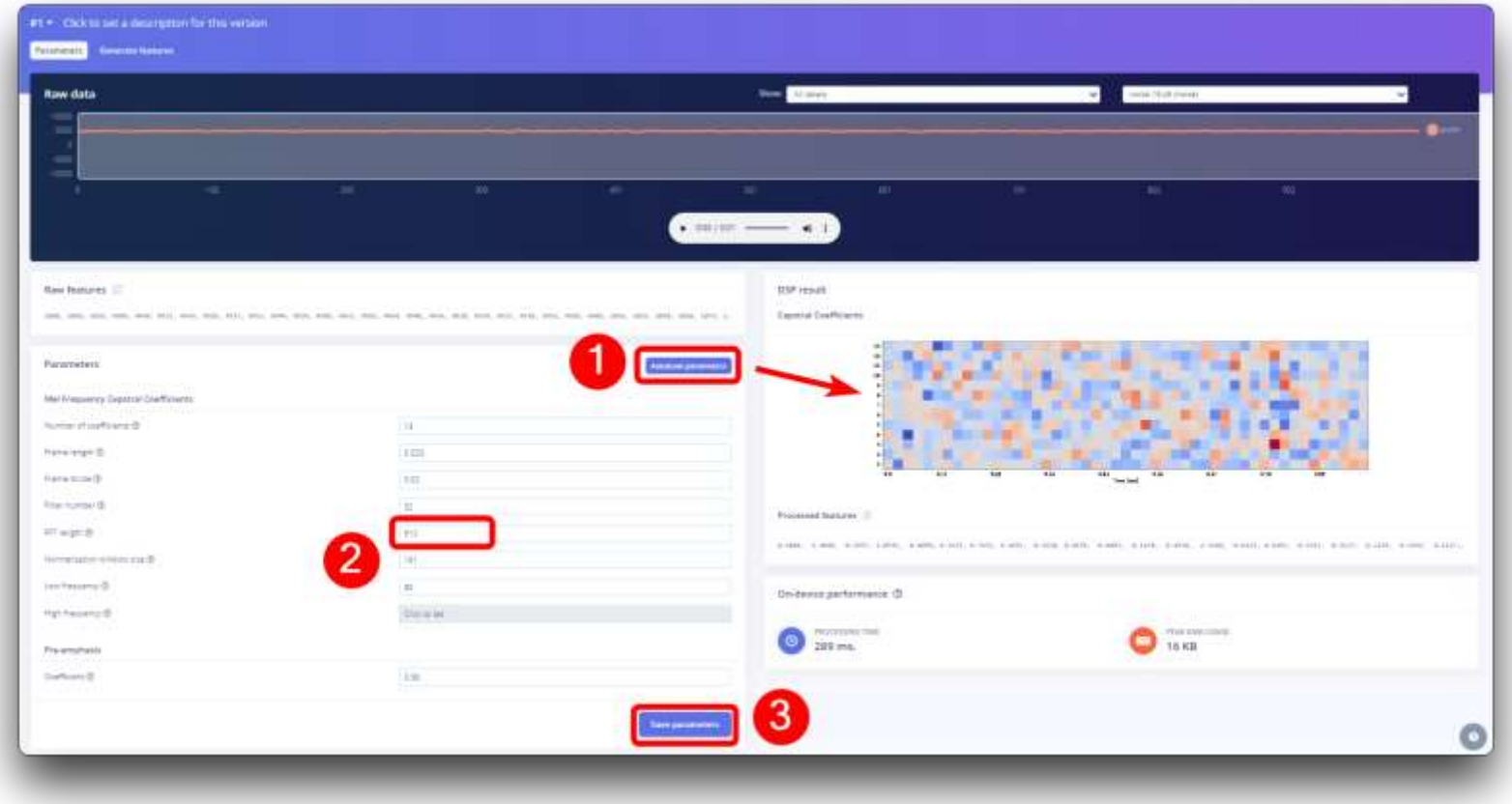
# Training Exported Models with Edge Impulse




An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

First, we will take the data points with a 1-second window, augmenting the data, sliding that window each 500ms. Note that the option **zero-pad data** is set. This is important to fill with zeros samples smaller than 1 second (in some cases, I reduced the 1000 ms window on the **split tool** to avoid noises and spikes).

# Training Exported Models with Edge Impulse

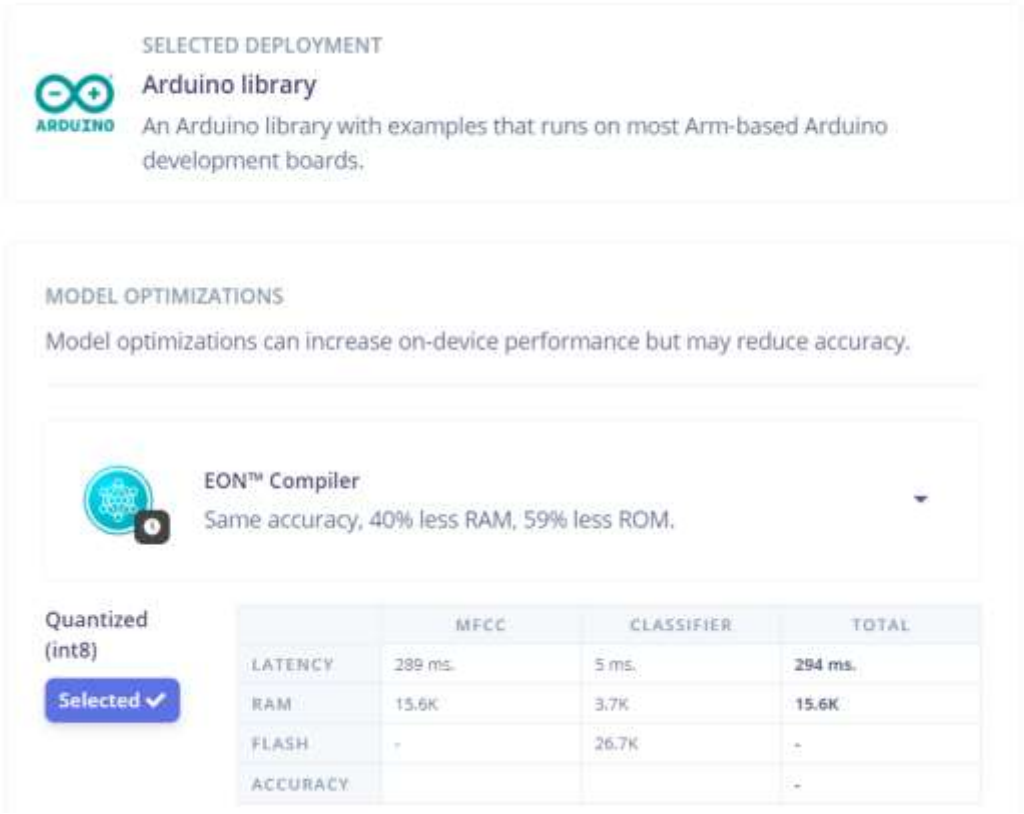



The next step is to create the images to be trained in the next phase. We can keep the default parameter values or take advantage of the DSP **Autotuneparameters option**, which we will do.

# Training Exported Models with Edge Impulse


We will use a Convolution Neural Network (CNN) model. The basic architecture is defined with two blocks of Conv1D + MaxPooling (with 8 and 16 neurons, respectively) and a 0.25 Dropout. And on the last layer, after Flattening four neurons, one for each class.

# Training Exported Models with Edge Impulse

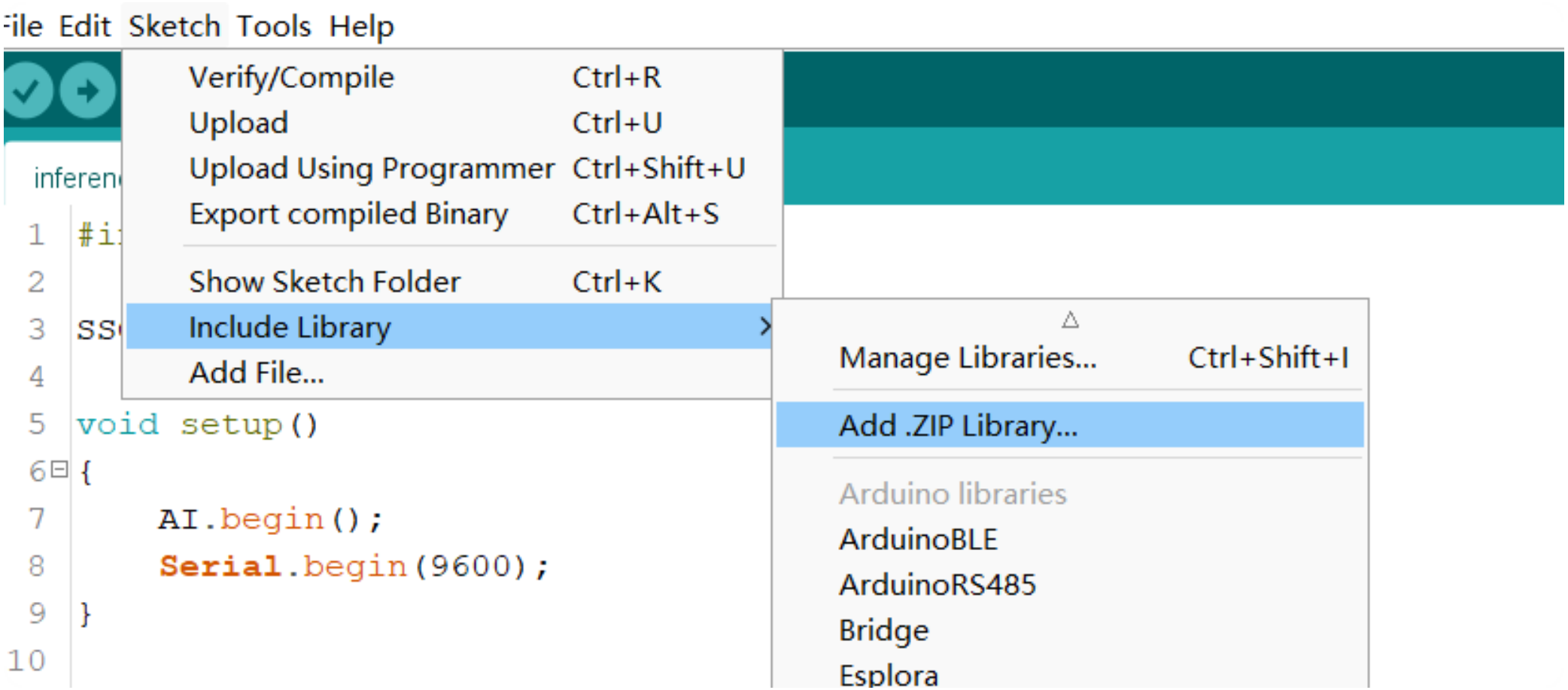



Edge Impulse will package all the needed libraries, preprocessing functions, and trained models, downloading them to your computer. You should select the option Arduino Library and at the bottom, select Quantized (Int8) and press the button Build.

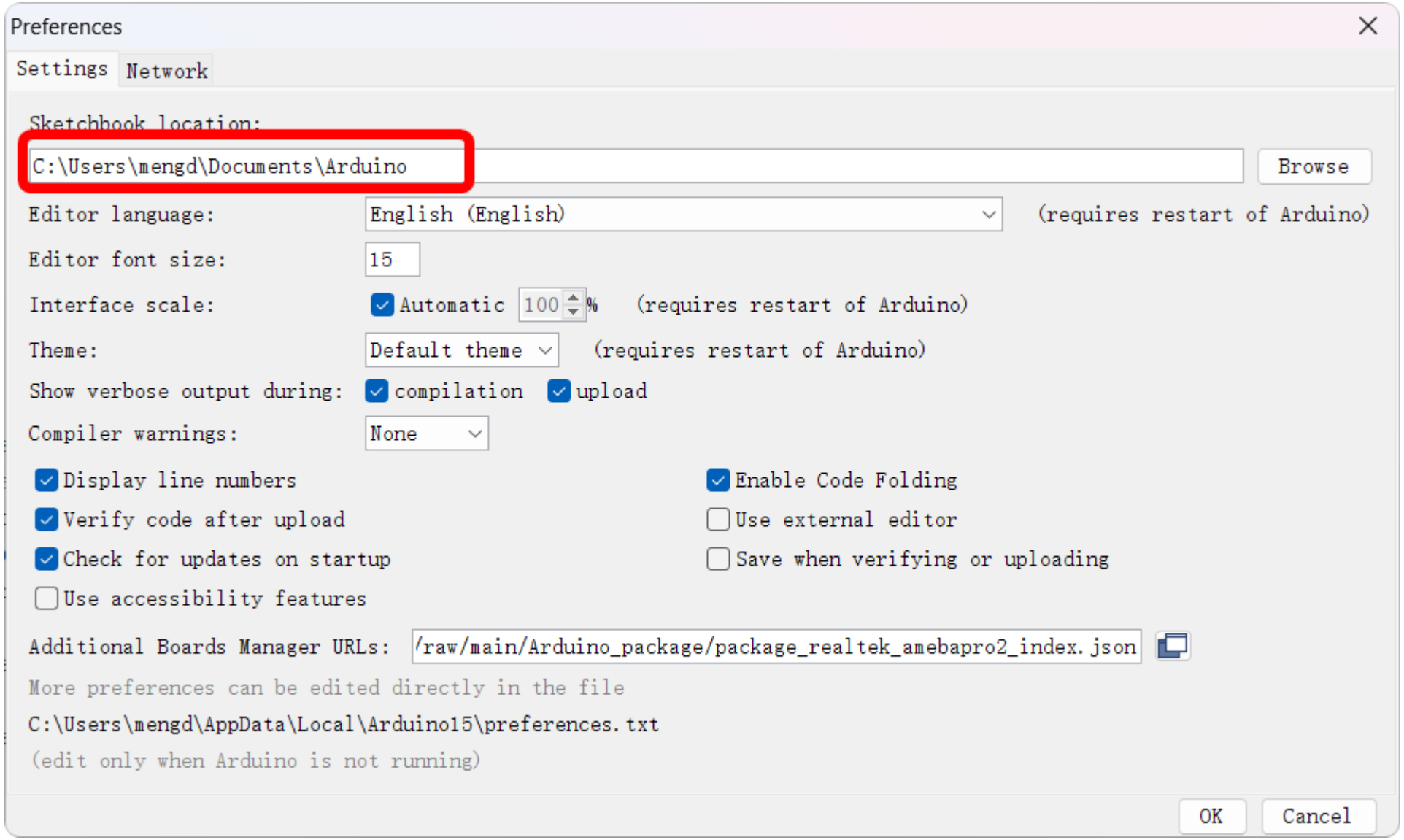
When the Build button is selected, a Zip file will be created and downloaded to your computer.

# Training Exported Models with Edge Impulse

Upload the zip file to you Arduino IDE

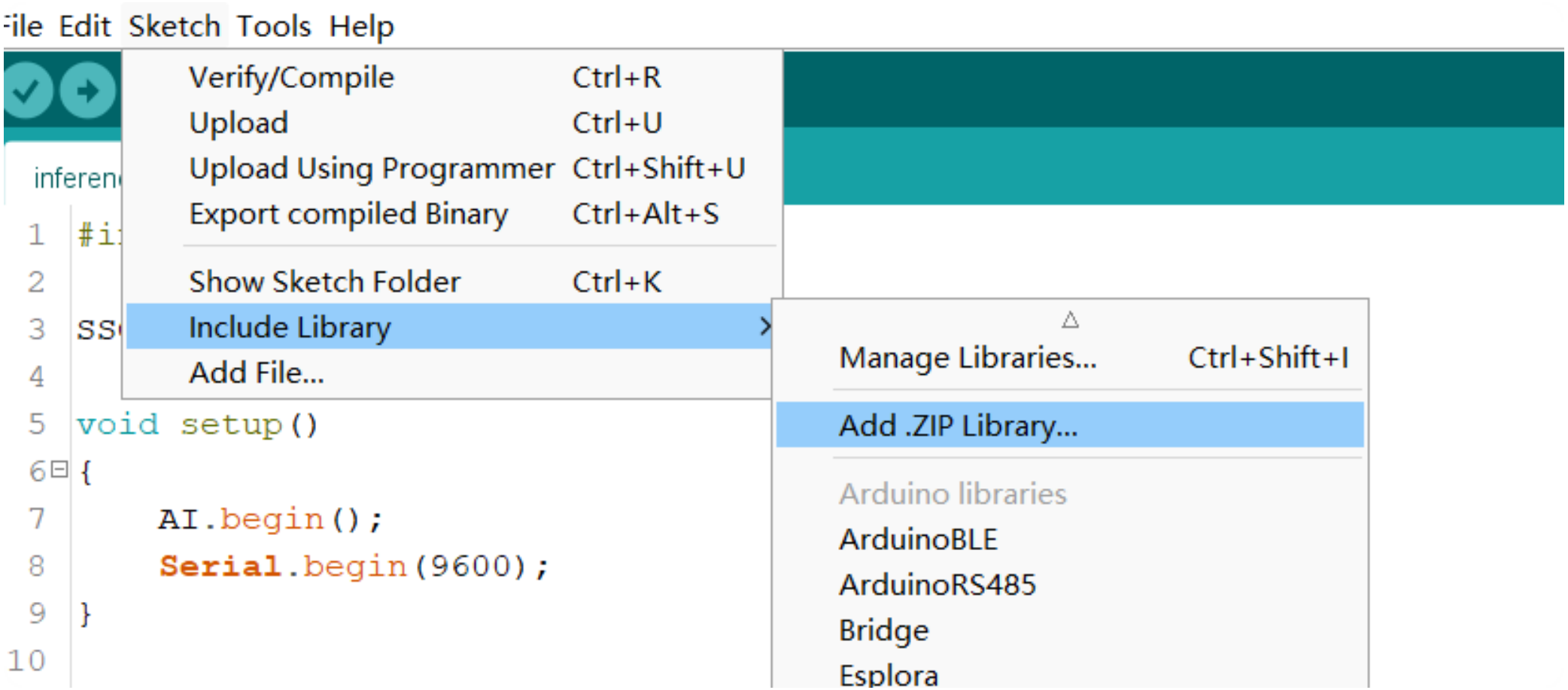


# Training Exported Models with Edge Impulse

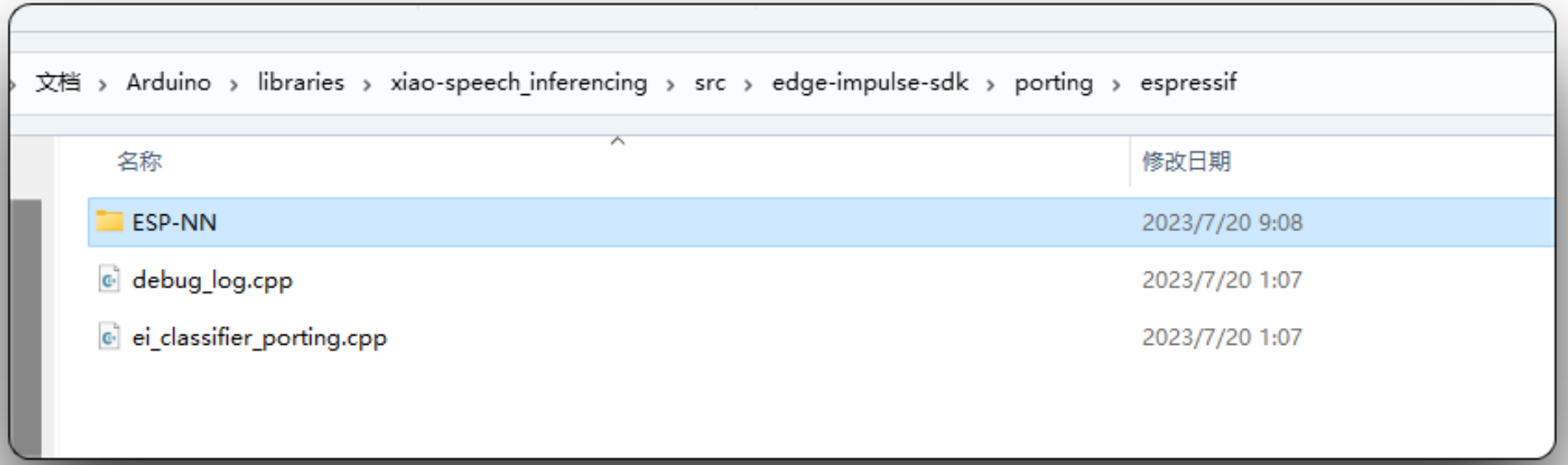


# Training Exported Models with Edge Impulse

Upload the zip file to you Arduino IDE



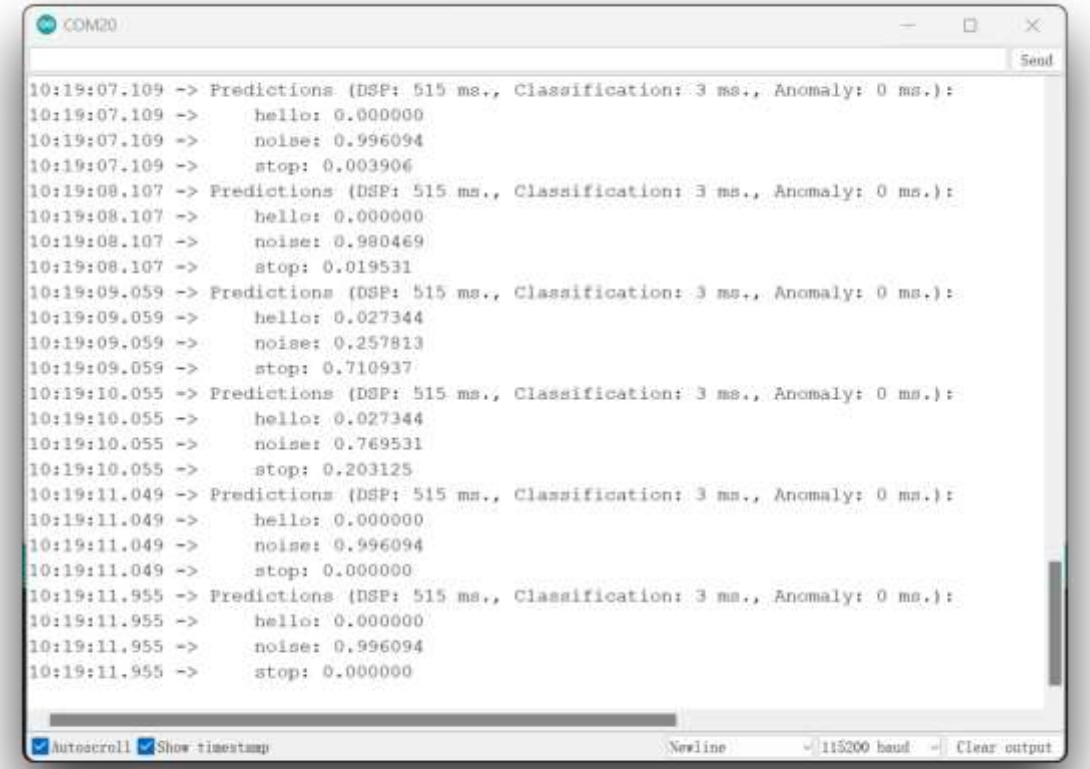
# Deploying models to XIAO ESP32S3 Sense




Before we use the downloaded library, we need to enable the ESP NN Accelerator. For that, you can download a preliminary version from the project GitHub, unzip it, and replace the ESP NN folder with it under: src/edge-impulse-sdk/porting/espressif/ESP-NN, in your Arduino library folder.

**Link Address**: https://github.com/Mjrovai/XIAO-ESP32S3-Sense/blob/main/ESP-NN.zip

# Deploying models to XIAO ESP32S3 Sense




You can find the complete code on the project's GitHub. Upload the sketch to your board and test some real inferences.

https://github.com/Mjrovai/XIAO-ESP32S3-Sense/tree/main/xiao_esp32s3_microphone_led

# Deploying models to XIAO ESP32S3 Sense

```
144         // Display inference result
145         if ((pred_index == 3) && (pred_value > 0.8)){
146           digitalWrite(LED_BUILT_IN, LOW); //Turn on
147         }
148         else{
149           digitalWrite(LED_BUILT_IN, HIGH); //Turn off
150         }
```

**Pred_index**: Index of identified tags

**Pred_value**: Confidence level

**LED_BUILT_IN**: Pin numbering of on-board LED

# Thanks you!