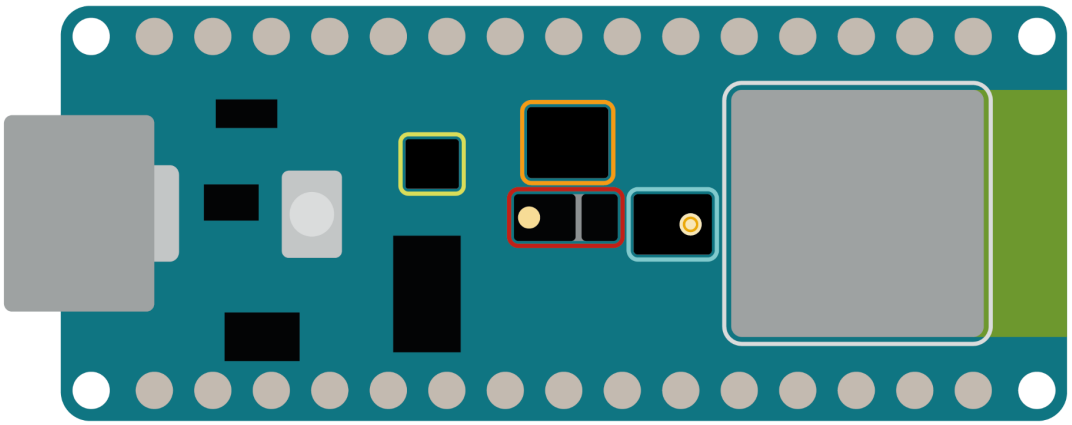
Machine Learning

---

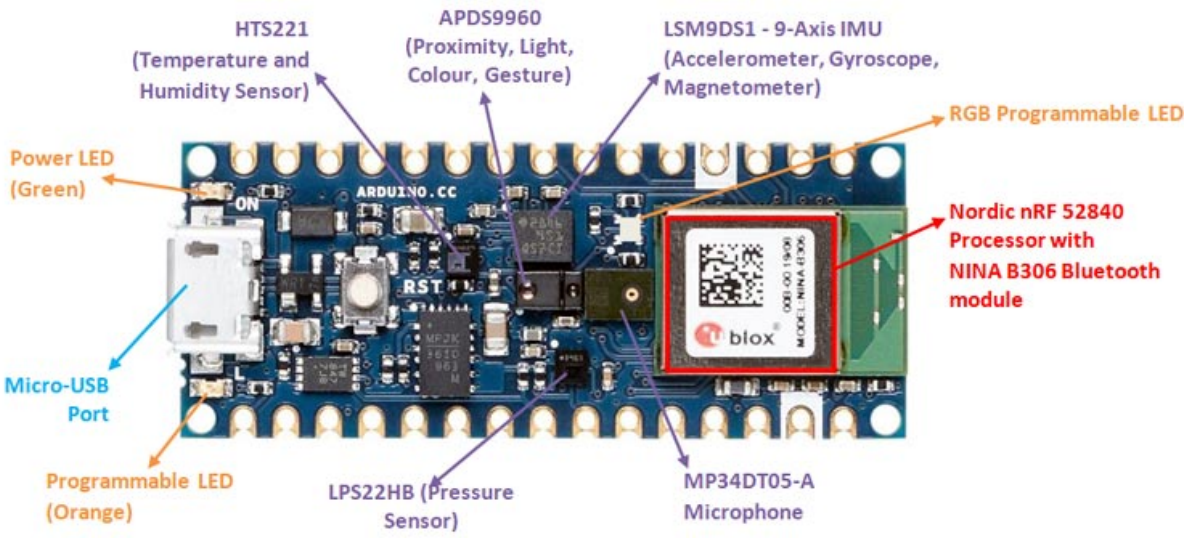
with the Arduino

Pinouts i sensory

NANO 33 BLE SENSE



- Color, brightness, proximity and gesture sensor
- Digital microphone
- Motion, vibration and orientation sensor
- Temperature, humidity and pressure sensor
- Arm Cortex-M4 microcontroller and BLE module

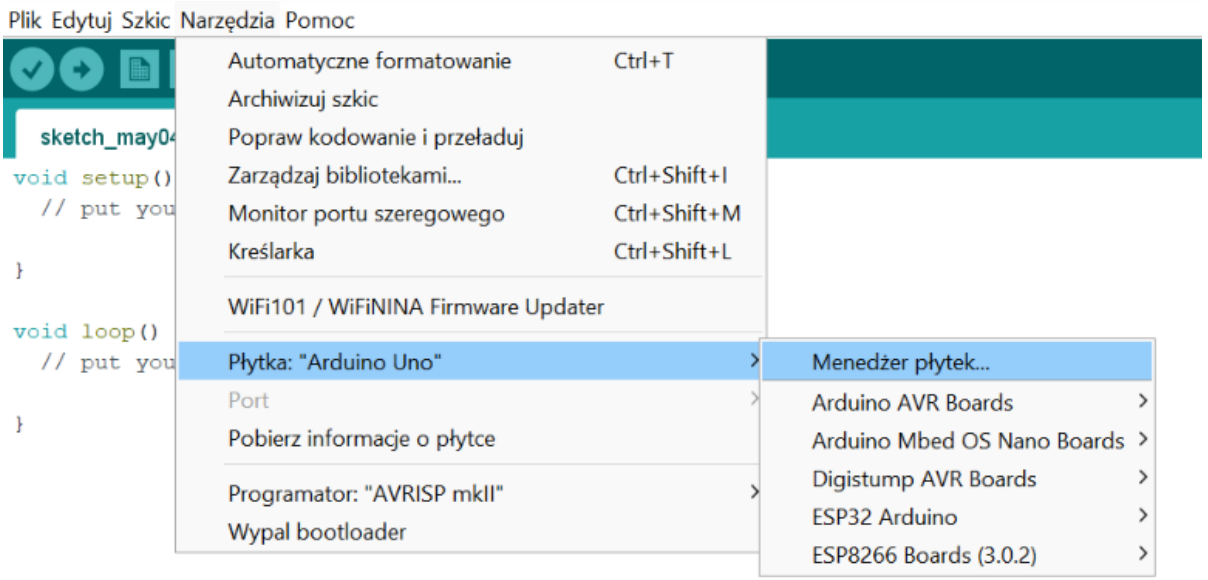


- ➔ Bluetooth w standardzie 5.0 LE
- ➔ Programowalny LED RGB
- ➔ HTS221 - czujnik temperatury i wilgotności
- ➔ APDS9960 - czujnik zbliżenia, światła, koloru i gestów
- ➔ LSM9DS1 - 3 osiowy żyroskop, akcelerometr i magnetometr
- ➔ MP34DT05-A - czujnik dźwięku
- ➔ LPS22HB - czujnik ciśnienia

Instalacja środowiska oraz bibliotek

W pierwszej kolejności instalujemy środowisko Arduino IDE 1.8.19 https://www.arduino.cc/en/software

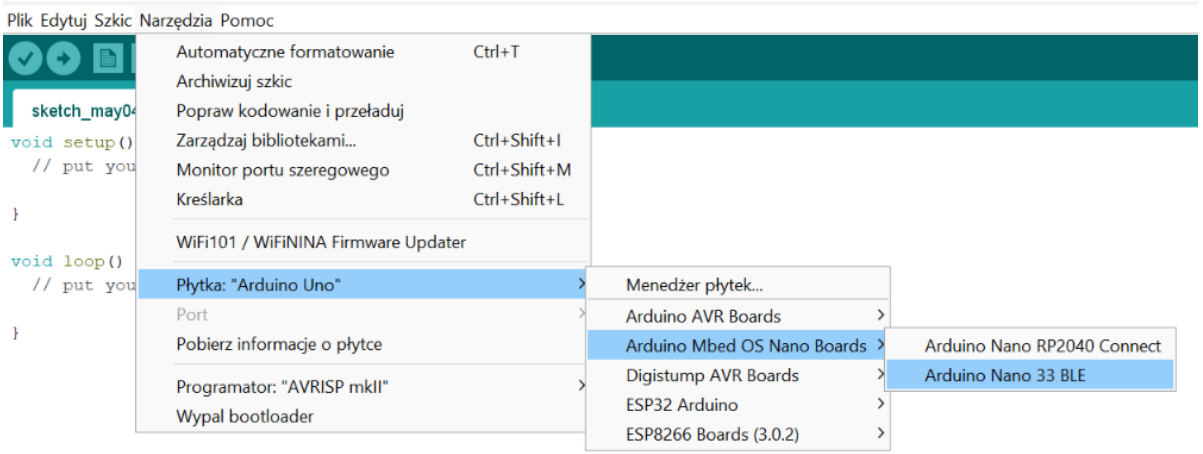
Instalacja biblioteki płytki Arduino Nano 33 BLE



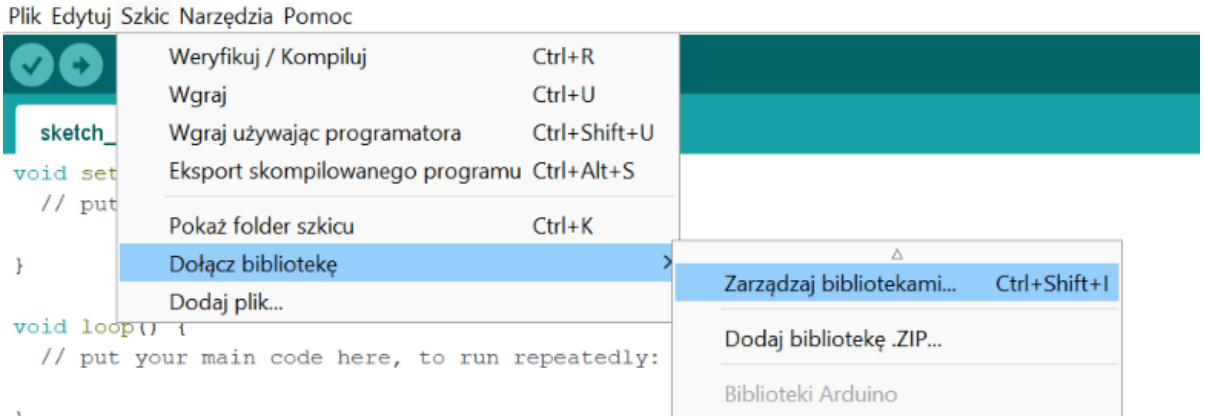
Wyszukujemy *Arduino Nano 33*

Arduino Mbed OS Nano Boards

by **Arduino** wersja **4.0.2** **INSTALLED**
Płytki dołączone w tej paczce:
Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense, Arduino Nano RP2040 Connect.
Online Help
More Info

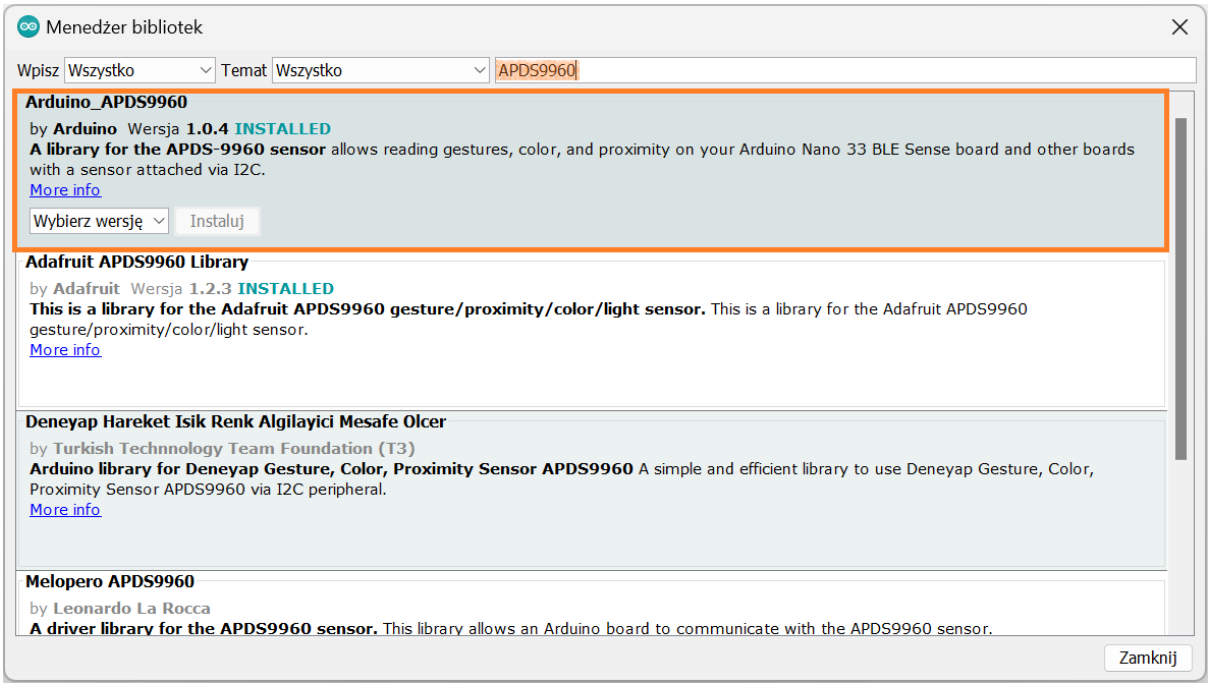


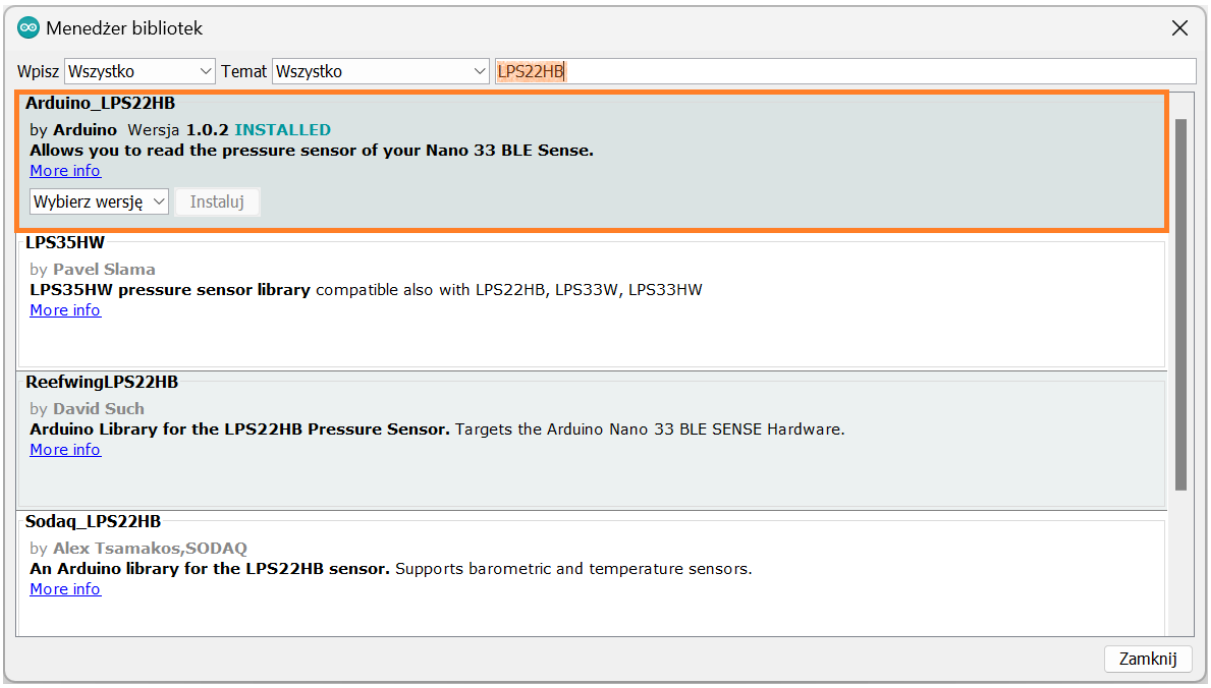
Instalacja bibliotek do sensorów

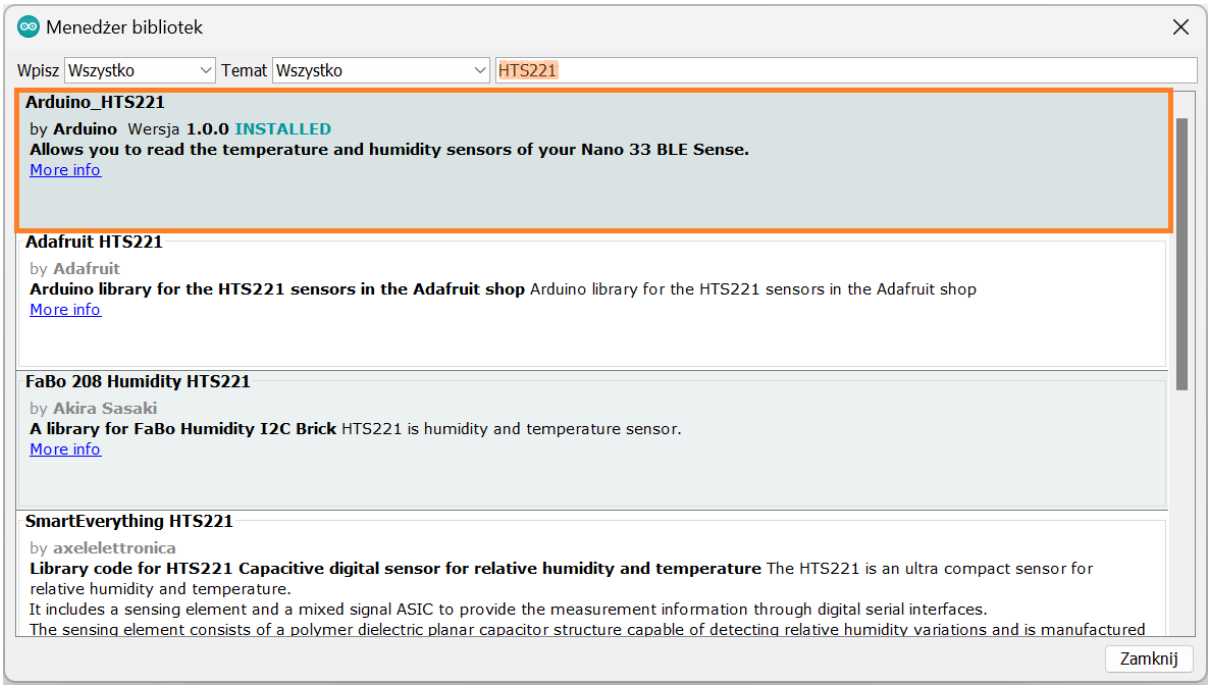


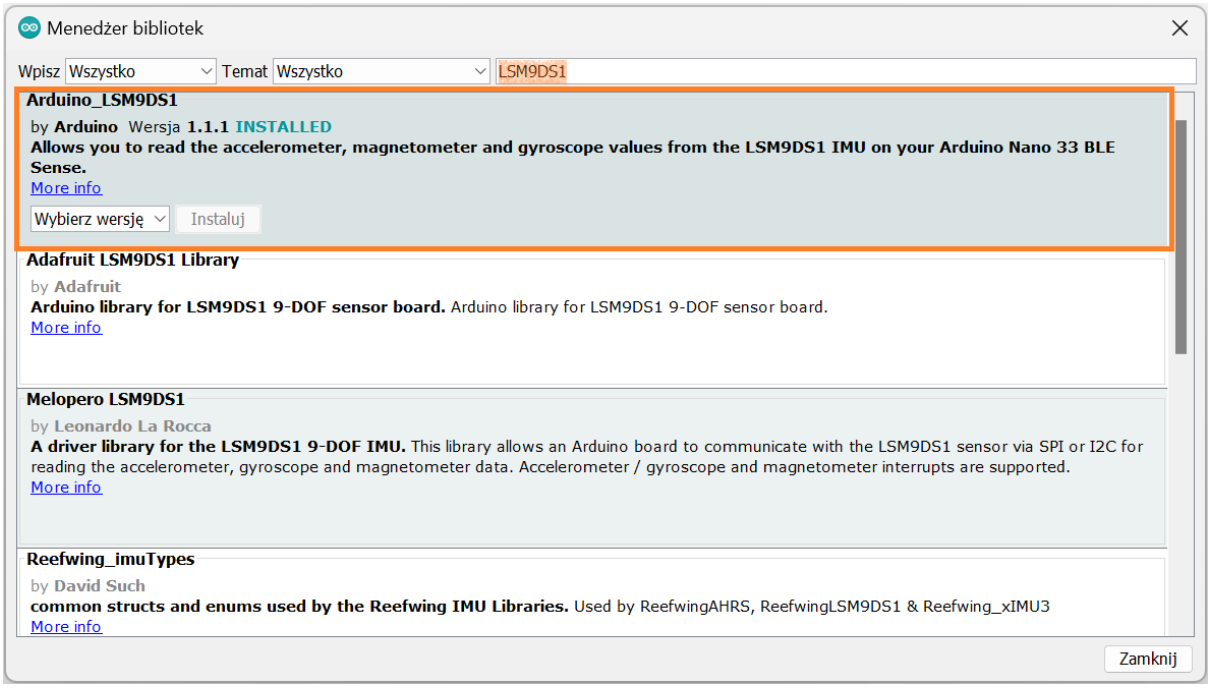
Instalujemy następujące biblioteki:

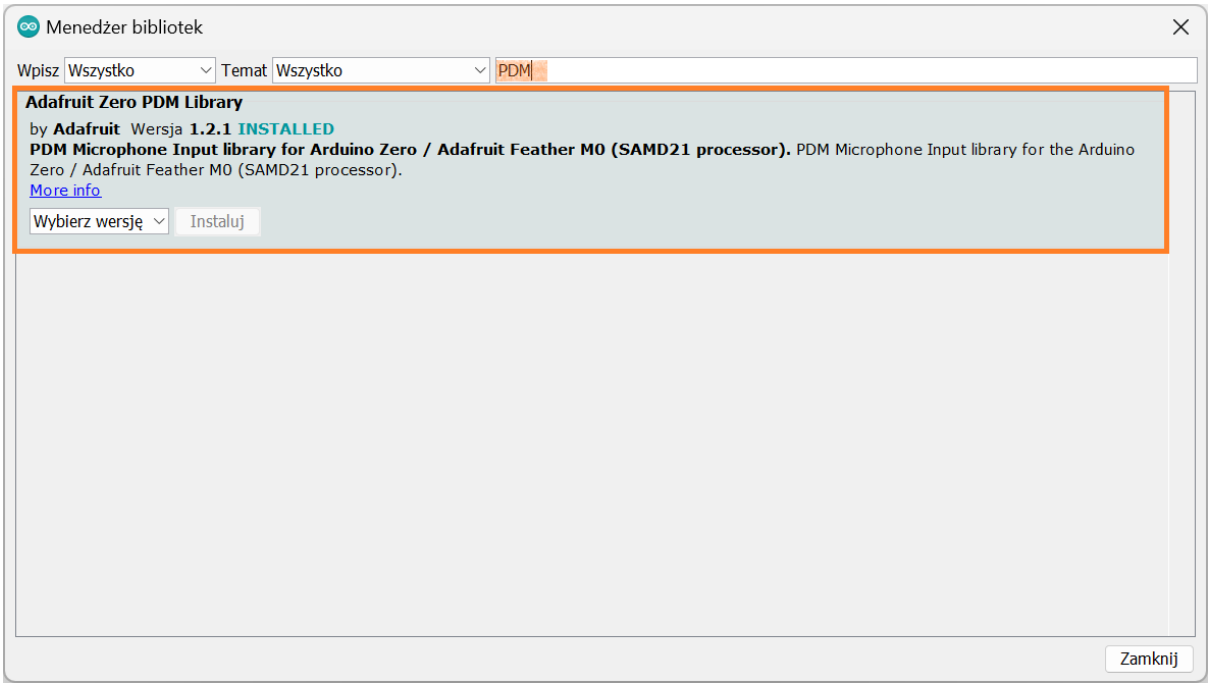
- Arduino_APDS9960
- LPS22HB
- Arduino_HTS221
- Arduino_LSM9DS1
- Adafruit Zero PDM Library
- Adafruit TensorFlow Lite
- Harvard_TinyMLx


Menedżer bibliotek
Wpisz Wszystko
Temat Wszystko
APDS9960
Arduino_APDS9960
by Arduino Wersja 1.0.4 INSTALLED
A library for the APDS-9960 sensor allows reading gestures, color, and proximity on your Arduino Nano 33 BLE Sense board and other boards with a sensor attached via I2C.
More info
Wybierz wersję
Instaluj
Adafruit APDS9960 Library
by Adafruit Wersja 1.2.3 INSTALLED
This is a library for the Adafruit APDS9960 gesture/proximity/color/light sensor. This is a library for the Adafruit APDS9960 gesture/proximity/color/light sensor.
More info
Deneyap Hareket Isik Renk Algilayici Mesafe Olcer
by Turkish Technnology Team Foundation (T3)
Arduino library for Deneyap Gesture, Color, Proximity Sensor APDS9960 A simple and efficient library to use Deneyap Gesture, Color, Proximity Sensor APDS9960 via I2C peripheral.
More info
Melopero APDS9960
by Leonardo La Rocca
A driver library for the APDS9960 sensor. This library allows an Arduino board to communicate with the APDS9960 sensor.
Zamknij

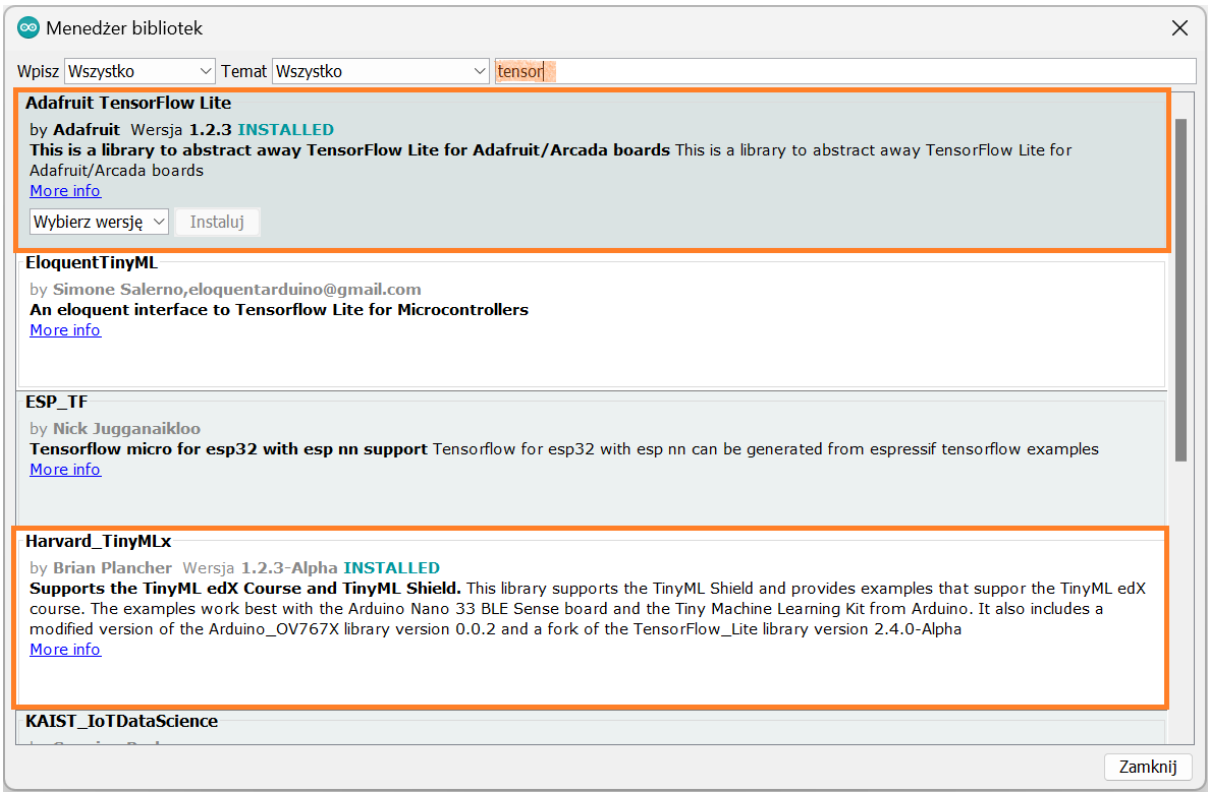

Menedżer bibliotek
Wpisz Wszystko
Temat Wszystko
LPS22HB
Arduino_LPS22HB
by Arduino Wersja 1.0.2 INSTALLED
Allows you to read the pressure sensor of your Nano 33 BLE Sense.
More info
Wybierz wersję
Instaluj
LPS35HW
by Pavel Slama
LPS35HW pressure sensor library compatible also with LPS22HB, LPS33W, LPS33HW
More info
ReefwingLPS22HB
by David Such
Arduino Library for the LPS22HB Pressure Sensor. Targets the Arduino Nano 33 BLE SENSE Hardware.
More info
Sodaq_LPS22HB
by Alex Tsamakos,SODAQ
An Arduino library for the LPS22HB sensor. Supports barometric and temperature sensors.
More info
Zamknij


Menedżer bibliotek
Wpisz Wszystko
Temat Wszystko
HTS221
Arduino_HTS221
by Arduino Wersja 1.0.0 INSTALLED
Allows you to read the temperature and humidity sensors of your Nano 33 BLE Sense.
More info
Adafruit HTS221
by Adafruit
Arduino library for the HTS221 sensors in the Adafruit shop Arduino library for the HTS221 sensors in the Adafruit shop
More info
FaBo 208 Humidity HTS221
by Akira Sasaki
A library for FaBo Humidity I2C Brick HTS221 is humidity and temperature sensor.
More info
SmartEverything HTS221
by axelelettronica
Library code for HTS221 Capacitive digital sensor for relative humidity and temperature The HTS221 is an ultra compact sensor for relative humidity and temperature.
It includes a sensing element and a mixed signal ASIC to provide the measurement information through digital serial interfaces.
The sensing element consists of a polymer dielectric planar capacitor structure capable of detecting relative humidity variations and is manufactured
Zamknij


Menedżer bibliotek
Wpisz Wszystko
Temat Wszystko
LSM9DS1
Arduino_LSM9DS1
by Arduino Wersja 1.1.1 INSTALLED
Allows you to read the accelerometer, magnetometer and gyroscope values from the LSM9DS1 IMU on your Arduino Nano 33 BLE Sense.
More info
Wybierz wersję
Instaluj
Adafruit LSM9DS1 Library
by Adafruit
Arduino library for LSM9DS1 9-DOF sensor board. Arduino library for LSM9DS1 9-DOF sensor board.
More info
Melopero LSM9DS1
by Leonardo La Rocca
A driver library for the LSM9DS1 9-DOF IMU. This library allows an Arduino board to communicate with the LSM9DS1 sensor via SPI or I2C for reading the accelerometer, gyroscope and magnetometer data. Accelerometer / gyroscope and magnetometer interrupts are supported.
More info
Reefwing_imuTypes
by David Such
common structs and enums used by the Reefwing IMU Libraries. Used by ReefwingAHRS, ReefwingLSM9DS1 & Reefwing_xIMU3
More info
Zamknij



Instalacja biliotek AI



Intalacjia biblioteki BLE

https://github.com/arduino-libraries/ArduinoBLE

Test biblioteki sensorów

Przykłady i obsługa sensorów

Dioda RGB

- ➔ Jest to standardowa dioda RGB 4 pinowa. W celu włączenia odpowiednich kolorów należy zaprogramować odpowiednie piny na stan wysoki. Ze względu na brak PWM na pinach możliwe jest uzyskanie tylko 7 kolorów.
  - ◆ Red - pin 22
  - ◆ Green - pin 23
  - ◆ Blue - pin 24

APDS9960

- ➔ **ColorSensor - RGB** - przykładowy kod służy do prostego rozpoznawania składowych koloru RGB
- ➔ **ProximitySensor** - wykrywanie zbliżenia, nie zwraca dokładniej wartości dystansu, a jedynie informacje czy coś jest blisko czy daleko (0 => close; 255 => far)
- ➔ **GestureSensor** - rozpoznaje kierunek ręki nad sensorem (Up, Down, Left, Right)
  - ◆ APDS.begin()
  - ◆ APDS.gestureAvailable()
  - ◆ int gesture = APDS.readGesture()

HTS221

- ➔ **ReadSensors** - pozwala na odczyt wartość temperatury i wilgotności powietrza
  - ◆ HTS.begin()
  - ◆ float temperature = HTS.readTemperature()
  - ◆ float humidity = HTS.readHumidity()

LSM9DS1

IMU.begin()

- ➔ **SimpleAccelerometer**
  - ◆ IMU.begin()
  - ◆ IMU.accelerationSampleRate()
  - ◆ IMU.accelerationAvailable()
  - ◆ IMU.readAcceleration(x, y, z)
- ➔ **SimpleGyroscope**
  - ◆ IMU.gyroscopeSampleRate()
  - ◆ IMU.gyroscopeAvailable()
  - ◆ IMU.readGyroscope(x, y, z)
- ➔ **SimpleMagnetometer**
  - ◆ IMU.magneticFieldSampleRate()

- IMU.magneticFieldAvailable()
- IMU.readMagneticField(x, y, z)

LPS22HB

➔ **ReadPressure**

- BARO.begin()
- BARO.readPressure()
- BARO.readTemperature()

Adafruit PDM - Microphone

➔

- PDM.onReceive(onPDMdata)
- PDM.setGain(30) - domyślnie jest 20
- PDM.begin(1, 16000)
- onPDMdata
- int bytesAvailable = PDM.available();
- PDM.read(sampleBuffer, bytesAvailable)

➔ https://www.szynalski.com/tone-generator/

Harvard_TinyMLx

➔ **micro_speech**

Ten kod to implementacja modelu uczenia maszynowego opartego na sieci neuronowej, który rozpoznaje krótkie fragmenty mowy i przypisuje im jedną z trzech etykiet: "yes", "no" lub "silence".

Kod zawiera różne pliki nagłówkowe, które zawierają dane i parametry potrzebne do działania modelu. Są to m.in. dane opisujące cechy dźwięków ("micro_features"), dane treningowe dla poszczególnych etykiet ("yes_micro_features_data", "no_micro_features_data", "silence_micro_features_data") oraz parametry modelu ("tiny_conv_micro_features_params", "yes_micro_features_params", "no_micro_features_params", "silence_micro_features_params").

W kodzie jest również zdefiniowany obiekt "error_reporter", który służy do raportowania błędów oraz obiekt "recognizer", który zawiera funkcje służące do uruchomienia modelu i przetworzenia sygnału dźwiękowego.

Cały kod składa się z pętli "loop", która w każdym cyklu odczytuje wartości z mikrofonu i przetwarza je za pomocą modelu. Jeśli model rozpozna jedną z etykiet, odpowiednia informacja zostanie wyświetlona na ekranie. Kod można dostosować do różnych zastosowań, np. sterowania urządzeniami na podstawie mowy lub monitorowania hałasu w pomieszczeniu.

➔ **magic_wand**

To implementacja modelu uczenia maszynowego opartego na sieci neuronowej, który wykorzystuje akcelerometr do rozpoznawania ruchów ręki i reprezentuje go za pomocą uproszczonego schematu.

Kod składa się z plików nagłówkowych i źródłowych oraz dwóch funkcji głównych: "setup" i "loop". Funkcja "setup" inicjalizuje akcelerometr i model, a także ustawia parametry działania programu. Funkcja "loop" działa w pętli, odczytuje dane z akcelerometru i przetwarza je za pomocą modelu.

➔ **person_detection**

Kod person_detection jest przykładem implementacji sieci neuronowej do detekcji ludzi na zdjęciach przy użyciu mikrokontrolera. W skrócie, kod ten pobiera obraz z kamery wideo, a następnie analizuje go, aby wykryć, czy w obrazie znajduje się osoba.

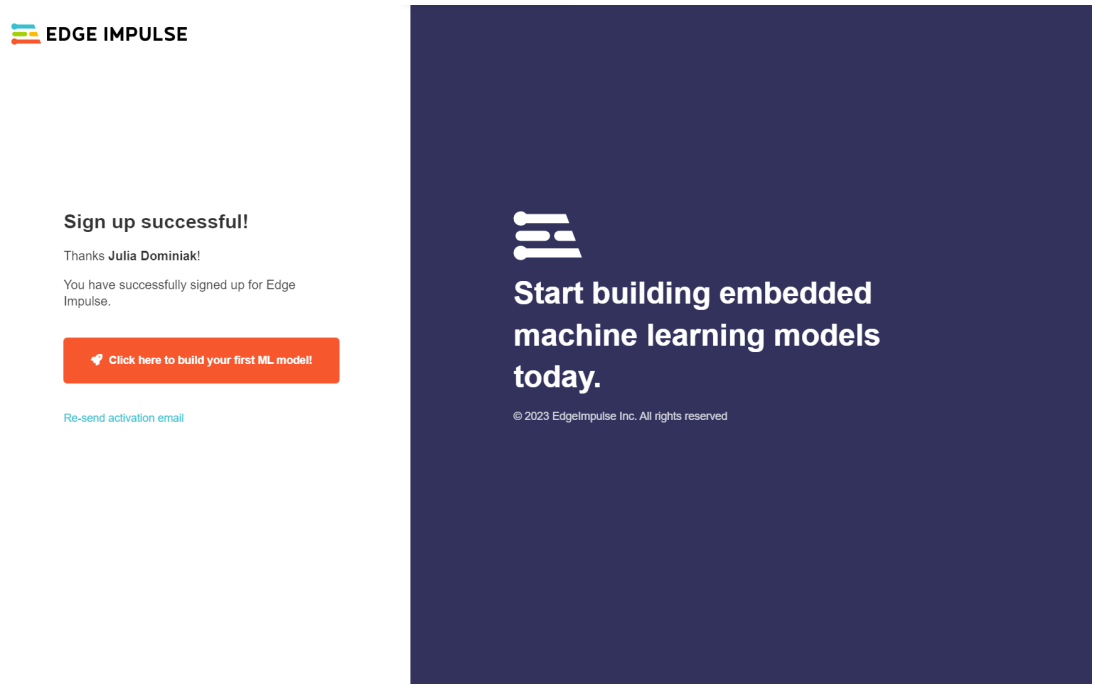
Cały kod składa się z kilku plików, w tym plików nagłówkowych, plików źródłowych oraz pliku konfiguracyjnego. W głównym pliku źródłowym, o nazwie "person_detection.cpp", znajduje się funkcja "run_person_detection()", która jest głównym interfejsem dla programu.

Funkcja ta inicjuje wszystkie niezbędne moduły, takie jak kamerę, wyświetlacz i sieć neuronową. Następnie, w pętli nieskończonej, funkcja pobiera obraz z kamery, przekształca go na wymagany przez sieć neuronową format i wywołuje funkcję "invoke()", która jest odpowiedzialna za przeprowadzenie detekcji osoby.

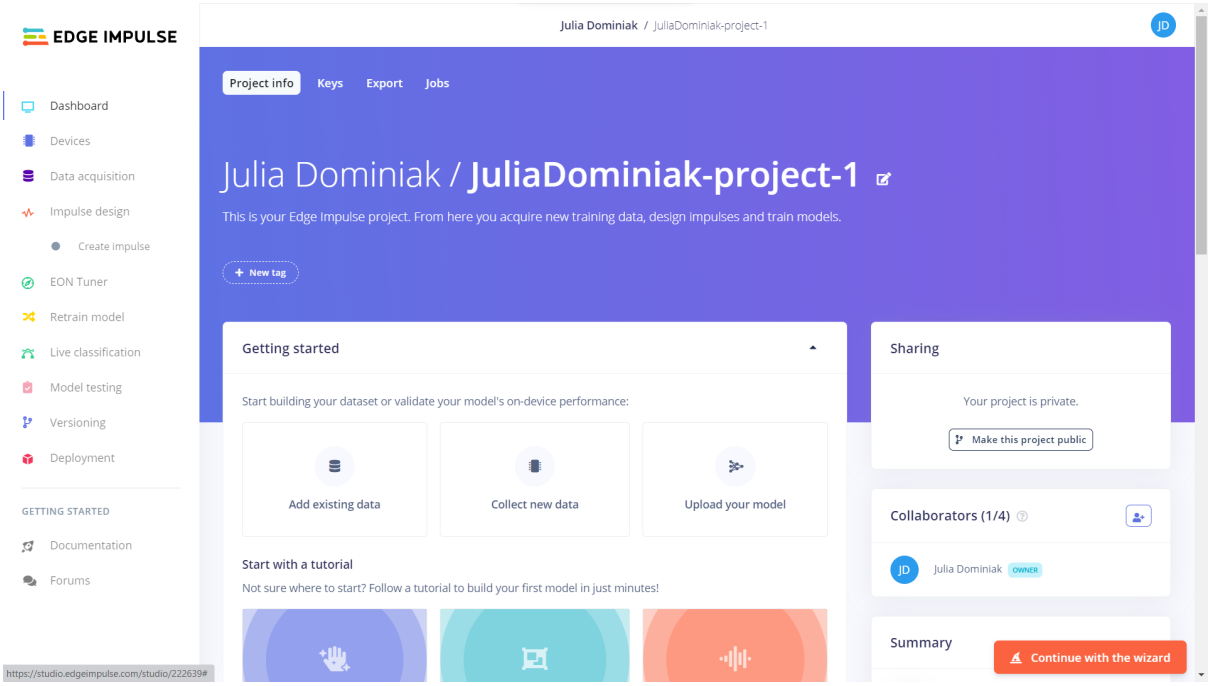
W funkcji "invoke()" sieć neuronowa jest zainicjowana, a następnie obraz jest podawany do sieci jako wejście. Sieć neuronowa zwraca wynik detekcji w postaci dwóch wartości: prawdopodobieństwa obecności osoby na zdjęciu i jej braku.

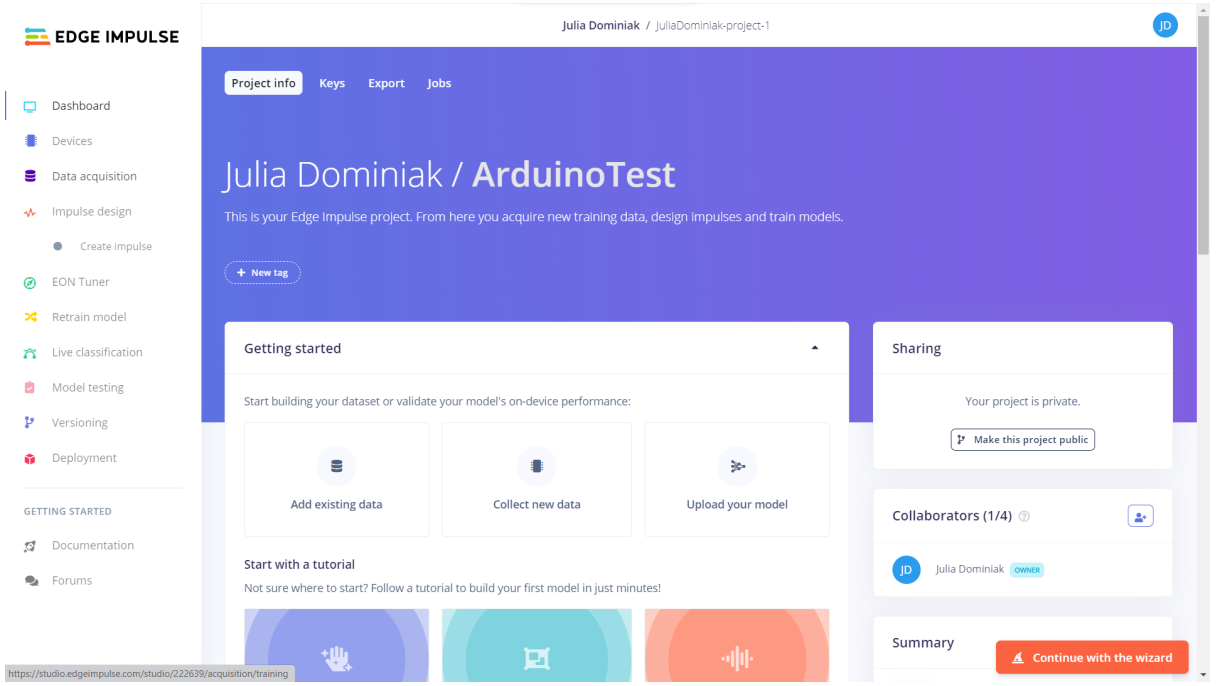
Przygotowanie środowiska

➔ wchodzimy na stronę https://www.edgeimpulse.com/



➔ zakładamy konto
➔ tworzymy nowy projekt

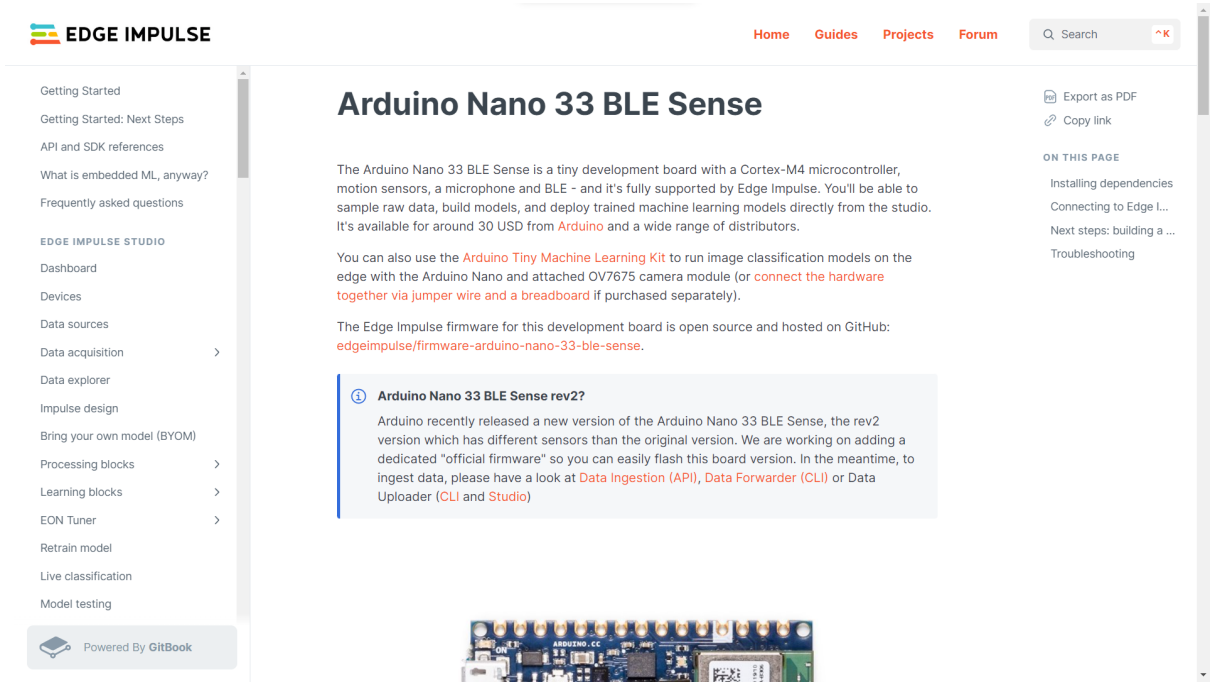


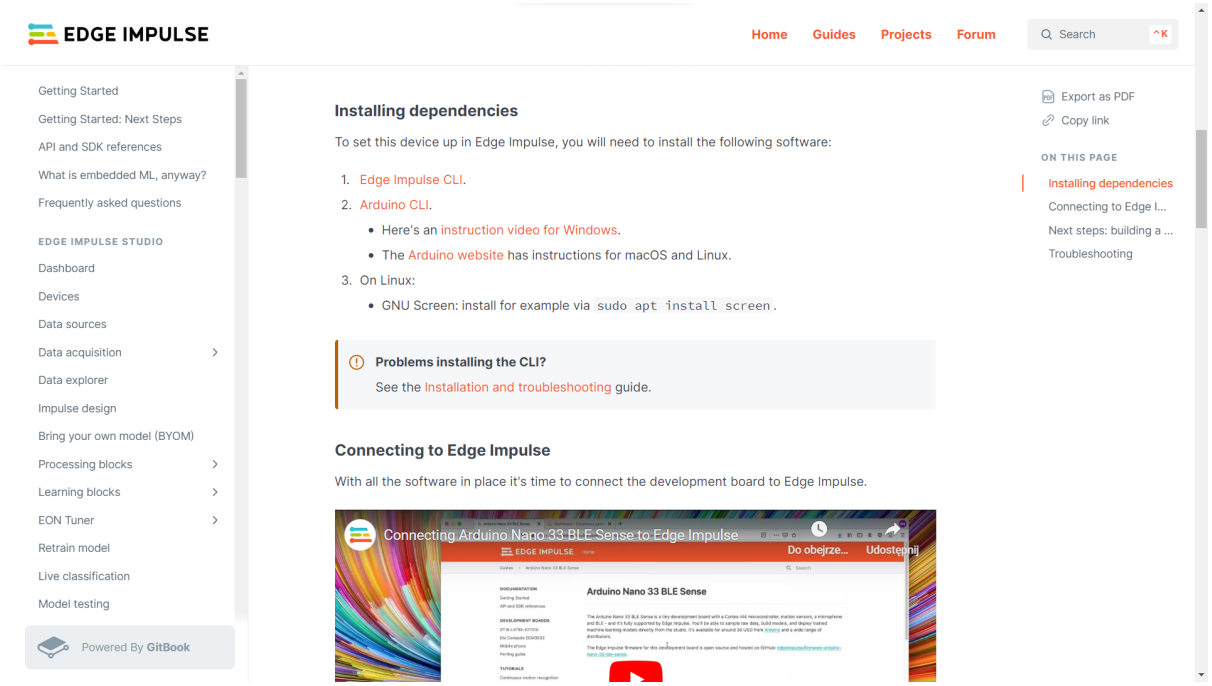


➔ w projekcie zaznaczamy że będziemy korzystać z kompatybilnej płytki (bo taką jest Arduino BLE33)

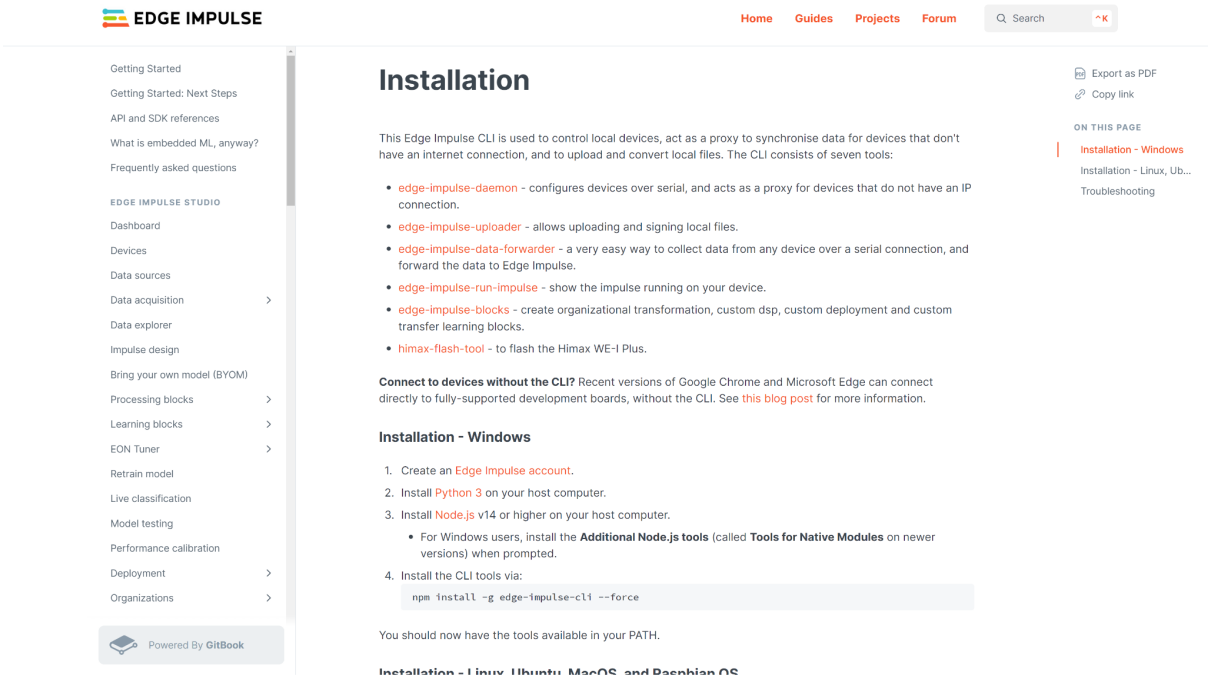
➔ następnie przechodzimy jak ją połączyć

➔ tutaj są informacje co musimy zainstalować https://docs.edgeimpulse.com/docs/development-platforms/officially-supported-mcu-targets/arduino-nano-33-ble-sense





➔ po kliknięciu na Edge Impulse CLI przekierowuje nas na stronę dotyczącą przygotowania środowiska https://docs.edgeimpulse.com/docs/edge-impulse-cli/cli-installation



➔ przedewszystkim należy pobrać pythona3 (3.11.4) i node.js (18.16.1 LST) aby uruchomić aplikację do kontroli i zbierania

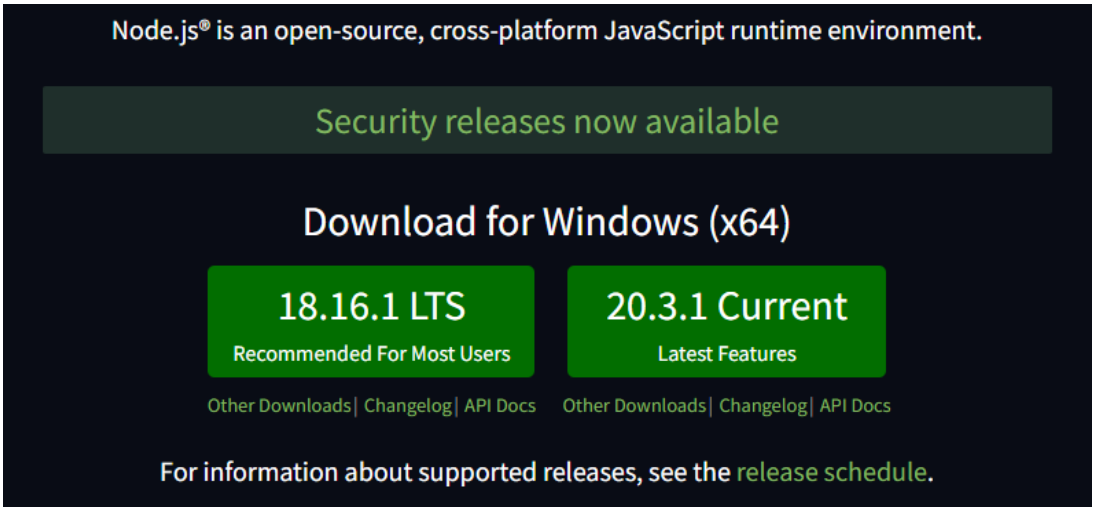
danych z lokalnego mikrokontrolera

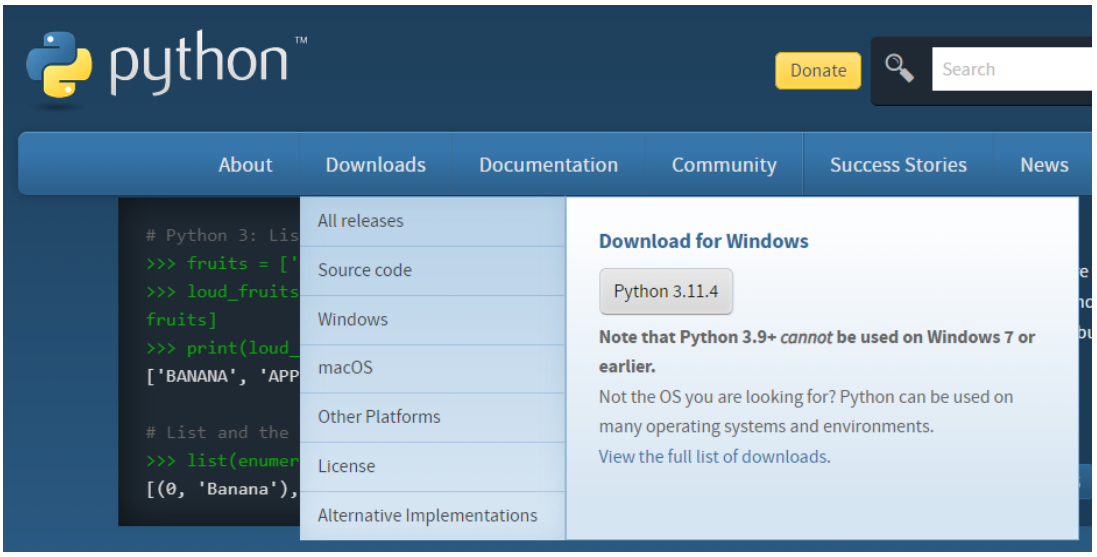
Installation - Windows

1. Create an Edge Impulse account.
2. Install Python 3 on your host computer.
3. Install Node.js v14 or higher on your host computer.
   - For Windows users, install the **Additional Node.js tools** (called **Tools for Native Modules** on newer versions) when prompted.
4. Install the CLI tools via:

```
npm install -g edge-impulse-cli --force
```

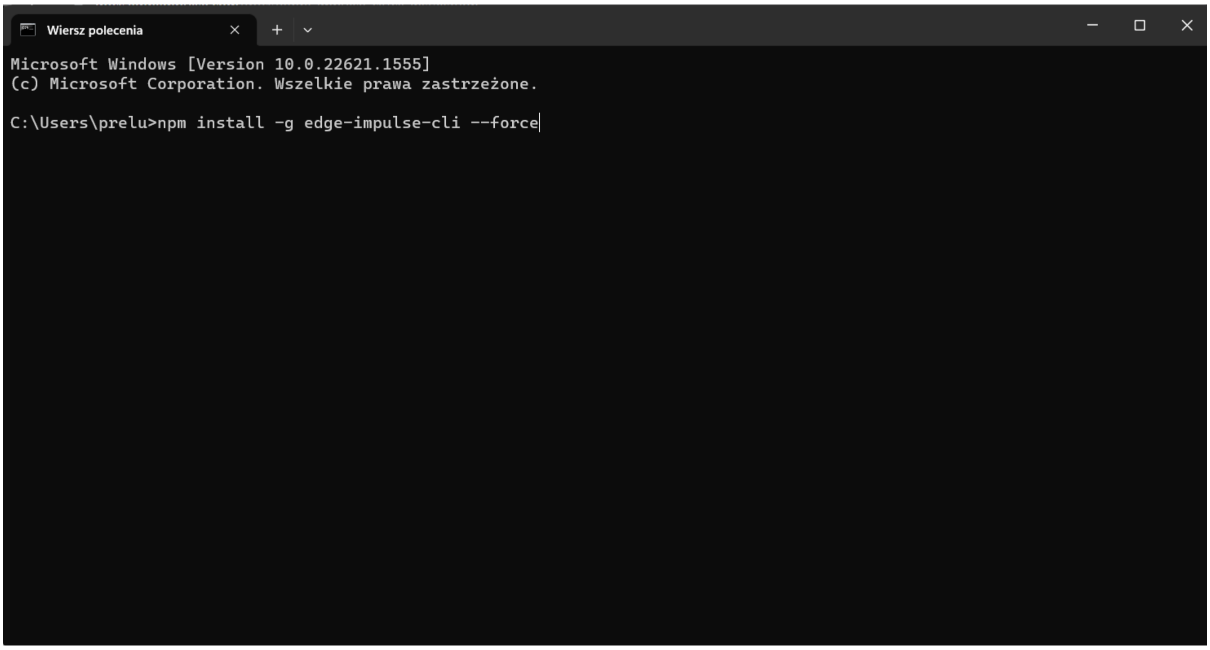
You should now have the tools available in your PATH.

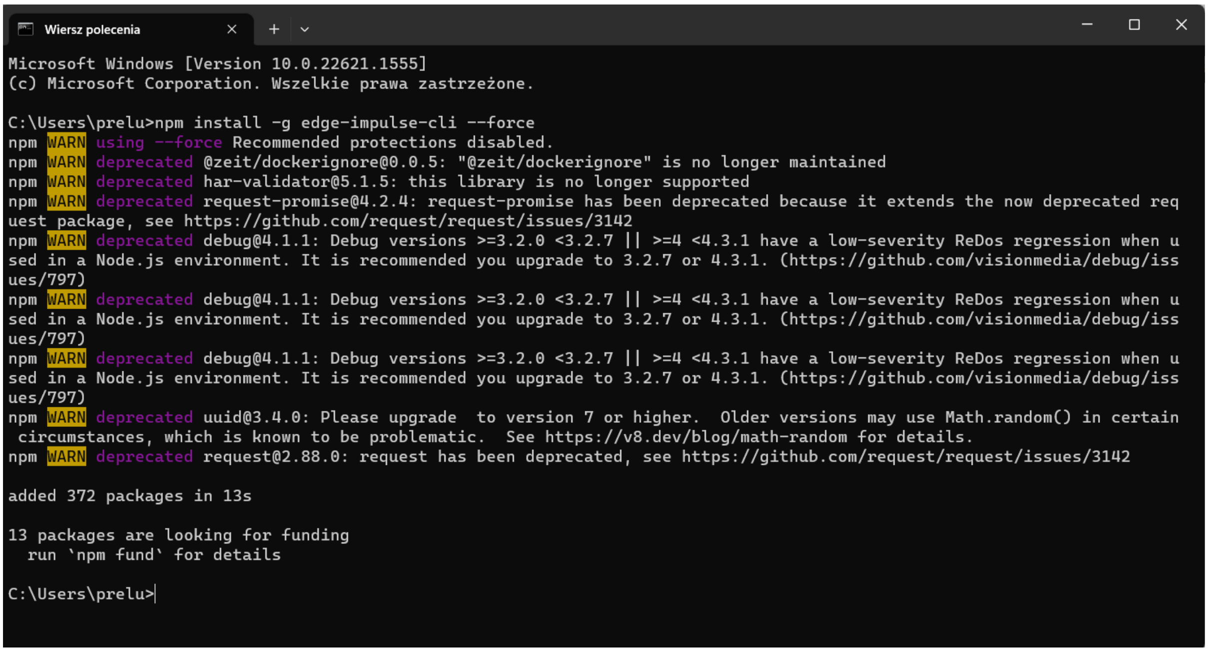




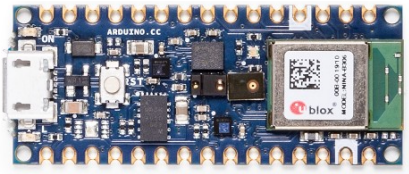
➔ ważne jest aby node.js zainstalować ze wszystkimi niezbędnymi dodatkami

➔ następnie w Wierszu Poleceń użyć komendy *npm install -g edge-impulse-cli --force*


Wiersz polecenia
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.
C:\Users\prelu>npm install -g edge-impulse-cli --force


Wiersz polecenia
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.
C:\Users\prelu>npm install -g edge-impulse-cli --force
npm WARN using --force Recommended protections disabled.
npm WARN deprecated @zeit/dockerignore@0.0.5: "@zeit/dockerignore" is no longer maintained
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated request-promise@4.2.4: request-promise has been deprecated because it extends the now deprecated request package, see https://github.com/request/request/issues/3142
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/issues/3142
added 372 packages in 13s
13 packages are looking for funding
run `npm fund` for details
C:\Users\prelu>

➔ po przejściu tych instrukcji i zainstalowaniu należy przejść do sekcji aktualizacji firmwaru



Arduino Nano 33 BLE Sense

Export as PDF
Copy link

ON THIS PAGE

- Installing dependencies
- Connecting to Edge Impulse
- Next steps: building a machine l...
- Troubleshooting

Installing dependencies

To set this device up in Edge Impulse, you will need to install the following software:

1. Edge Impulse CLI.
2. Arduino CLI.
   - Here's an instruction video for Windows.
   - The Arduino website has instructions for macOS and Linux.

➔ przechodzimy do Arduino CLI

Arduino CLI 0.32 | Search | arduino/arduino-cli 0.32.2 3.9k 343

Arduino CLI
- Home
- Installation
- Upgrading
- Getting started
- Command line completion
- How to contribute
- FAQ
- Command reference
- gRPC reference
- Configuration
- Integration options
- Sketch build process
- Sketch specification
- Sketch project file
- Library specification
- Platform specification
- Pluggable discovery specification
- Pluggable monitor specification
- Package index specification
- Guides

Home

Arduino CLI is an all-in-one solution that provides Boards/Library Managers, sketch builder, board detection, uploader, and many other tools needed to use any Arduino compatible board and platform from command line or machine interfaces.

In addition to being a standalone tool, Arduino CLI is the heart of all official Arduino development software (Arduino IDE, Arduino Web Editor). Parts of this documentation apply to those tools as well.

Installation

You have several options to install the latest version of the Arduino CLI on your system, see the installation page.

Getting started

Follow the Getting started guide to see how to use the most common CLI commands available.

Using the gRPC interface

The client_example folder contains a sample program that shows how to use the gRPC interface of the CLI. Available services and messages are detailed in the gRPC reference pages.

Table of contents
- Installation
- Getting started
- Using the gRPC interface

Installation

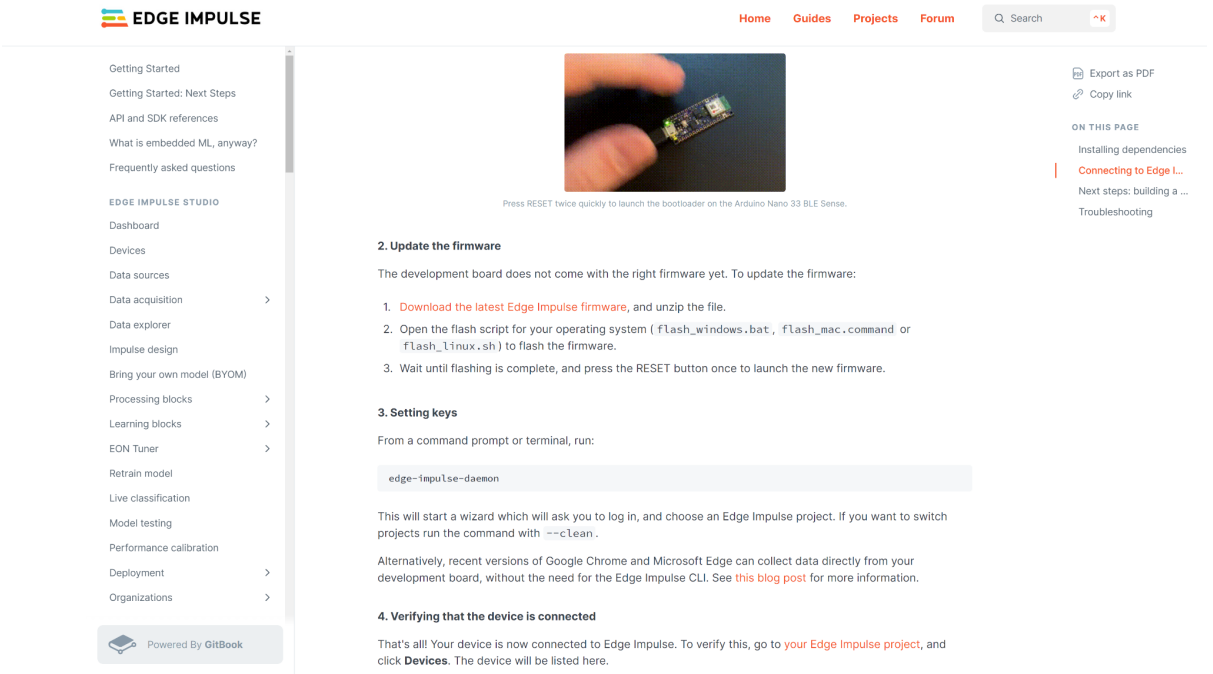
You have several options to install the latest version of the Arduino CLI on your system, see the installation page.

➔ i wybieramy interesującą nas opcję (w tym przypadku Windows exe)

Latest release

Platform		
Linux	32 bit	64 bit
Linux ARM	32 bit	64 bit
Linux ARMv6	32 bit	
Windows exe	32 bit	64 bit
Windows msi		64 bit
macOS		64 bit
macOS ARM		64 bit

➔ pobieramy również Edge Impulse firmware

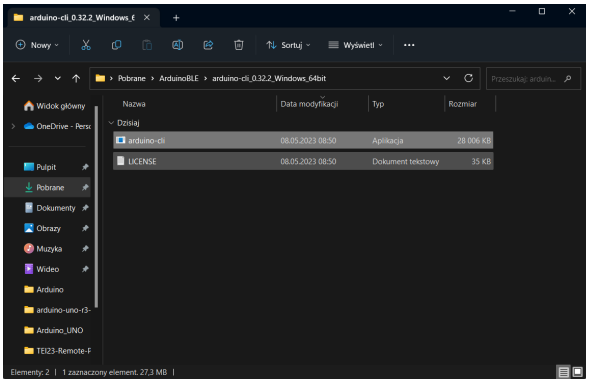


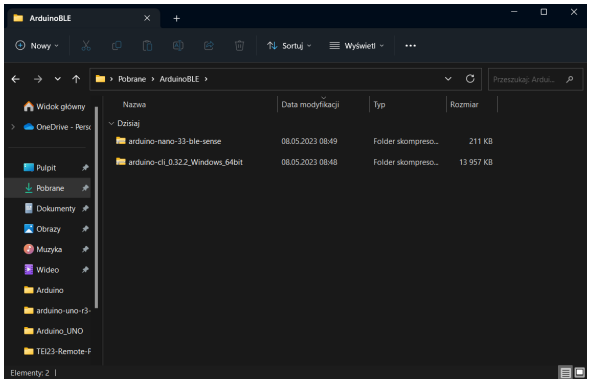
2. Update the firmware

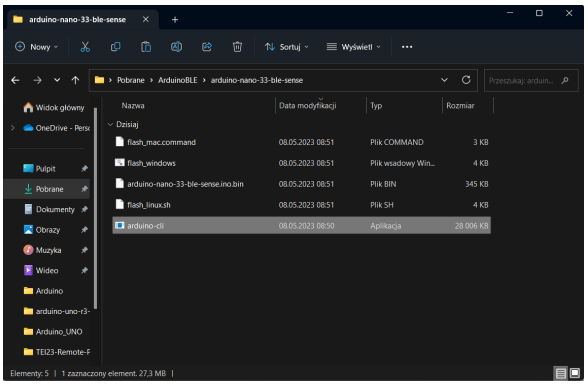
The development board does not come with the right firmware yet. To update the firmware:

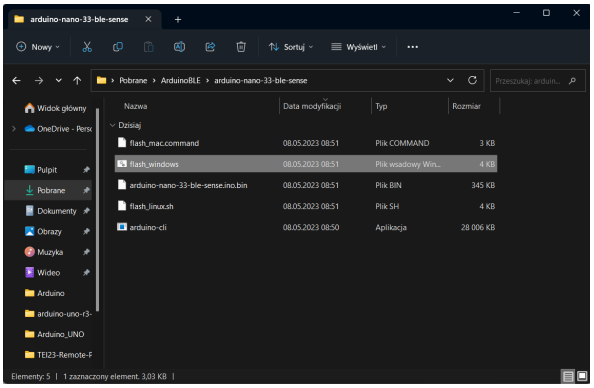
1. Download the latest Edge Impulse firmware, and unzip the file.
2. Open the flash script for your operating system (`flash_windows.bat` , `flash_mac.command` or `flash_linux.sh`) to flash the firmware.
3. Wait until flashing is complete, and press the RESET button once to launch the new firmware.

➔ po pobraniu firmware'u należy przenieść do folderu plik .exe z wypakowanego folderu Arduino CLI



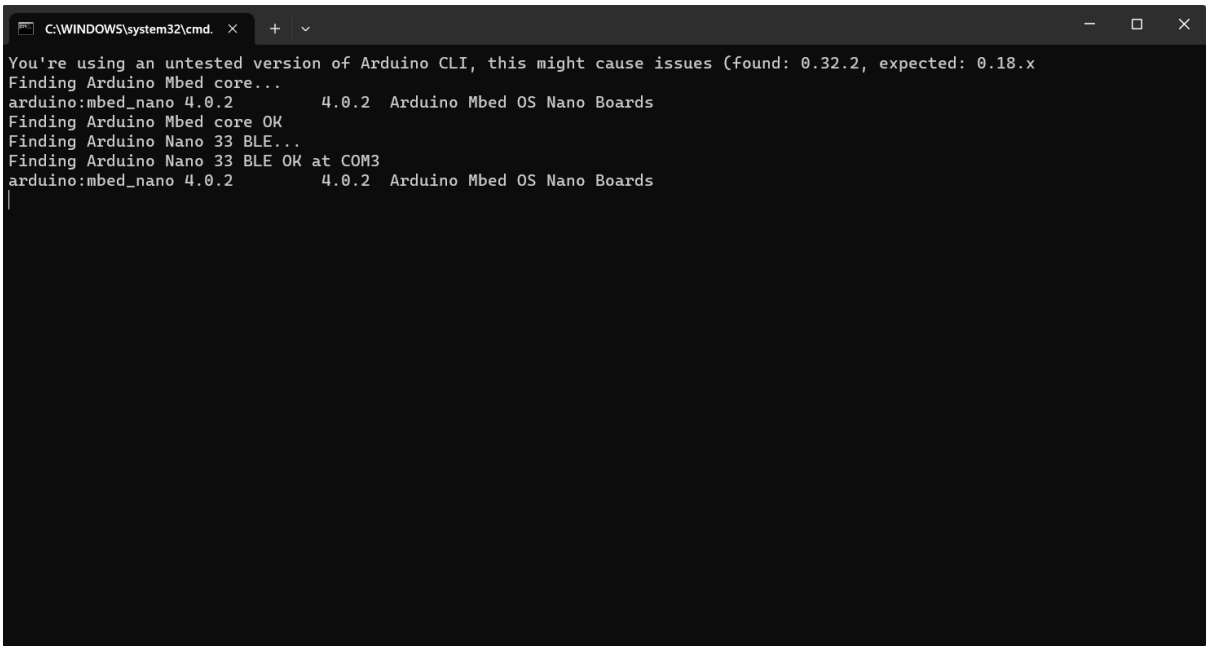






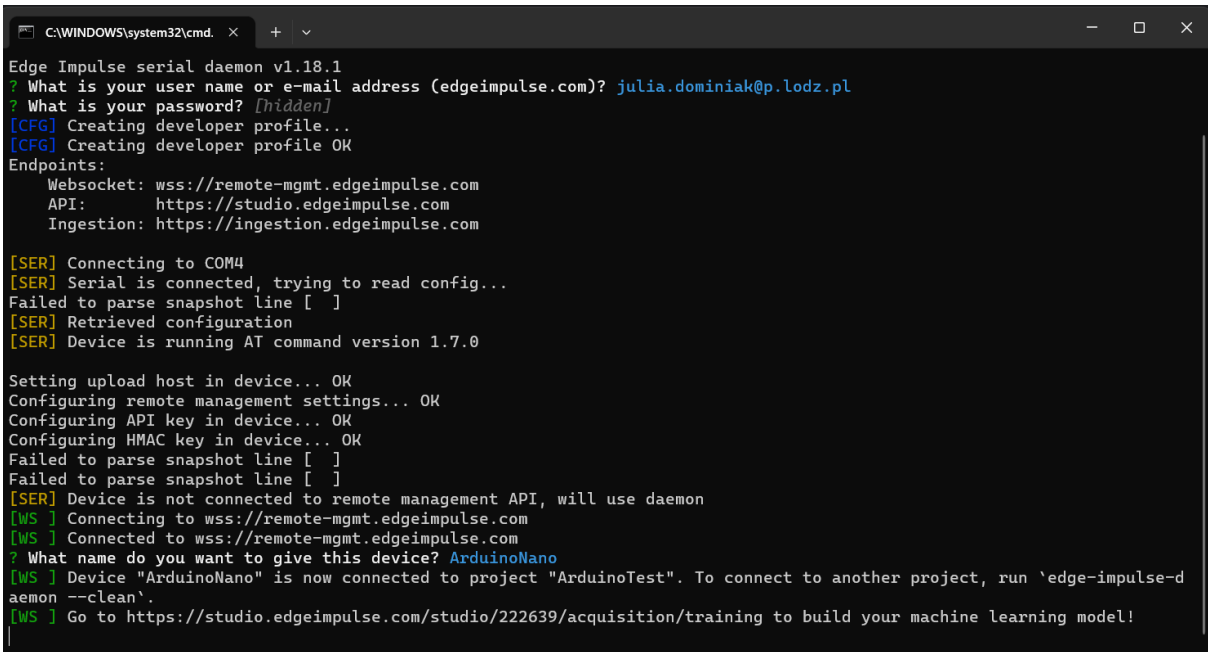
➔ następnie uruchomić flash_windows.bat

➔ komputer powinien wykryć Arduino na COM-ie i wgrać odpowiednie oprogramowanie (ważne - jest to program do przesyłu danych, po wgraniu innego kodu ponownie należy uruchomić plik)

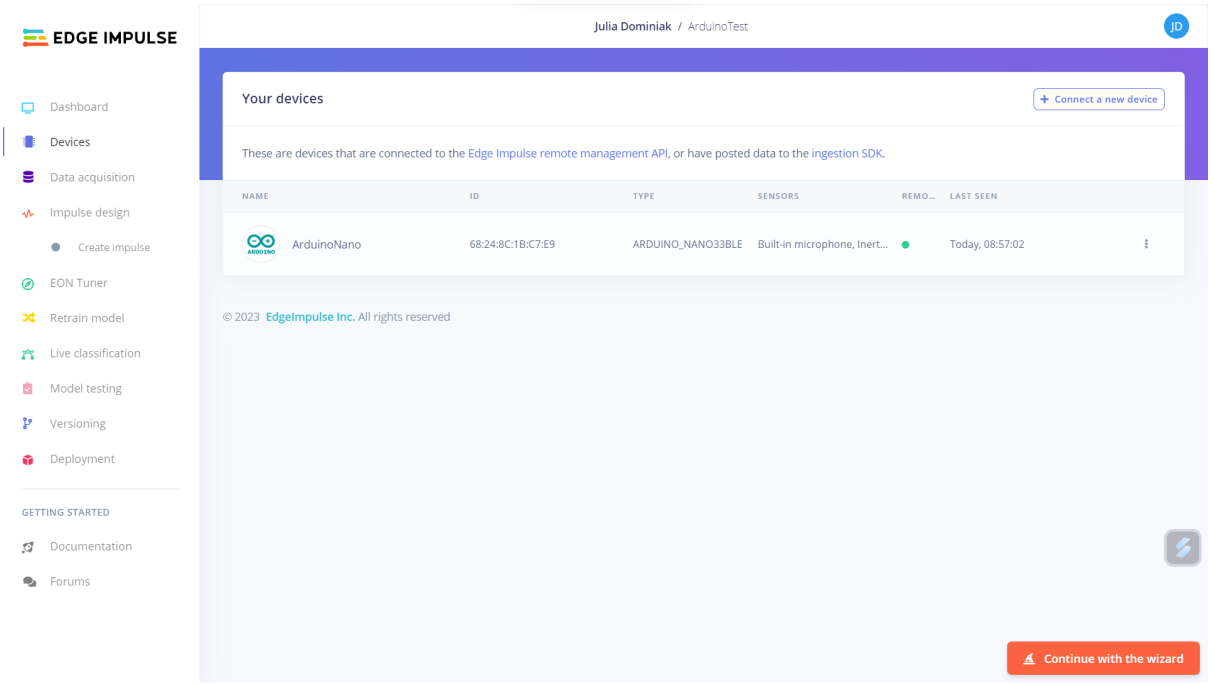


➔ kolejnym krokiem jest połączenie urządzenia z kontem na Edge Impulse za pomocą komendy w Wierszu Poleceń uruchamiamy połączenie z naszym kontem (standardowo logujemy się za pomocą maila i hasła)

```
edge-impulse-daemon
```

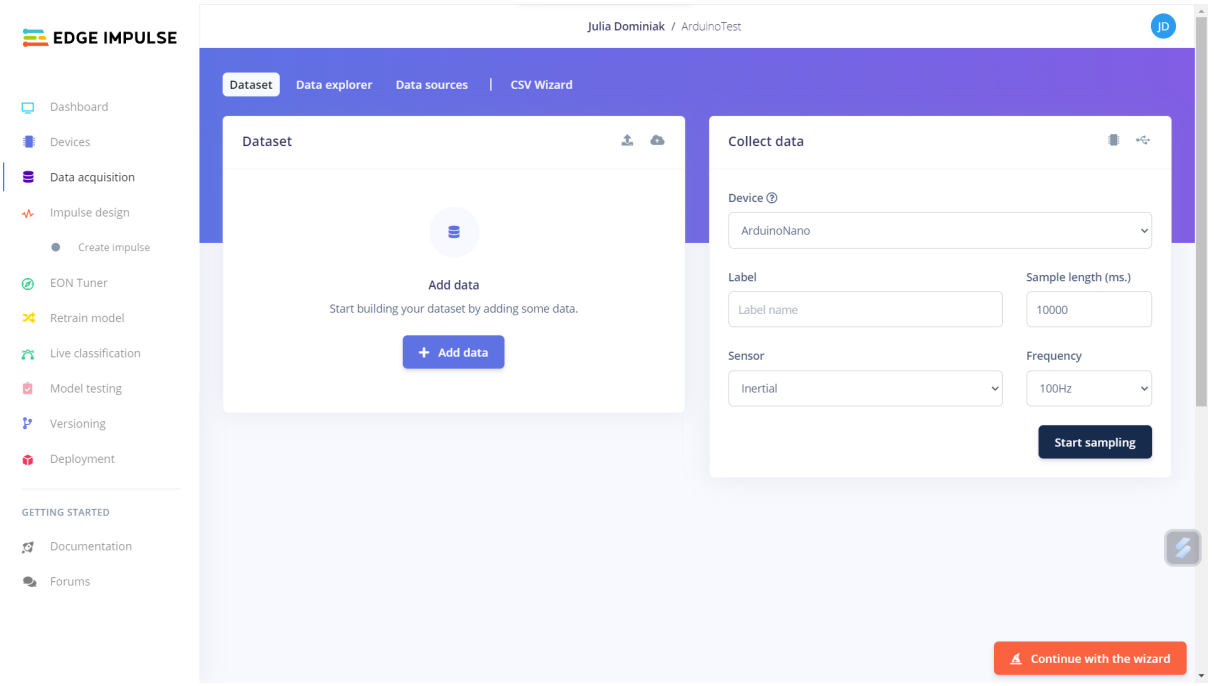


➔ po tym etapie na stronie projektu w Edge Impulse na liście "Devices" pojawi się nasze urządzenie

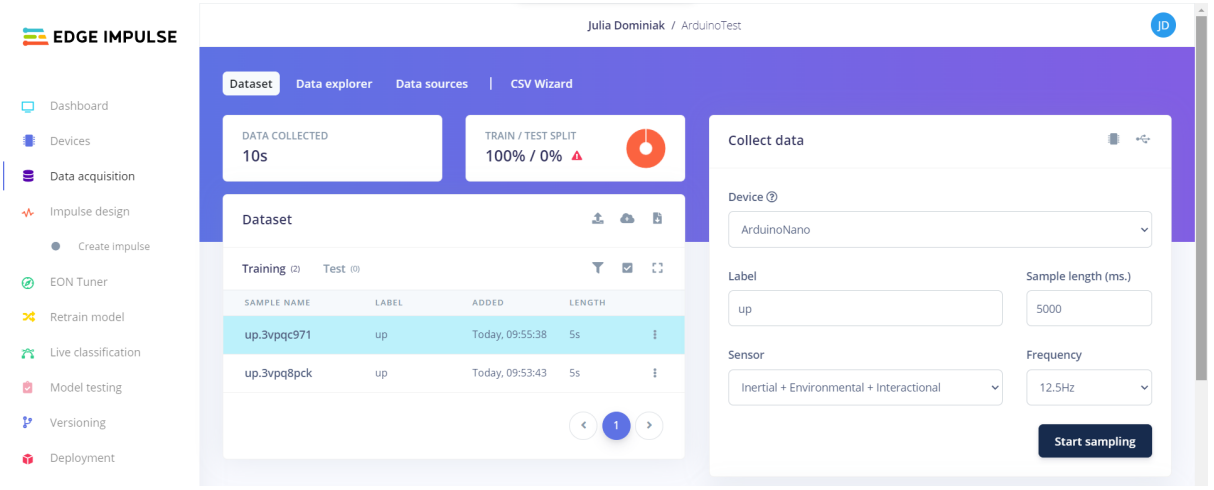


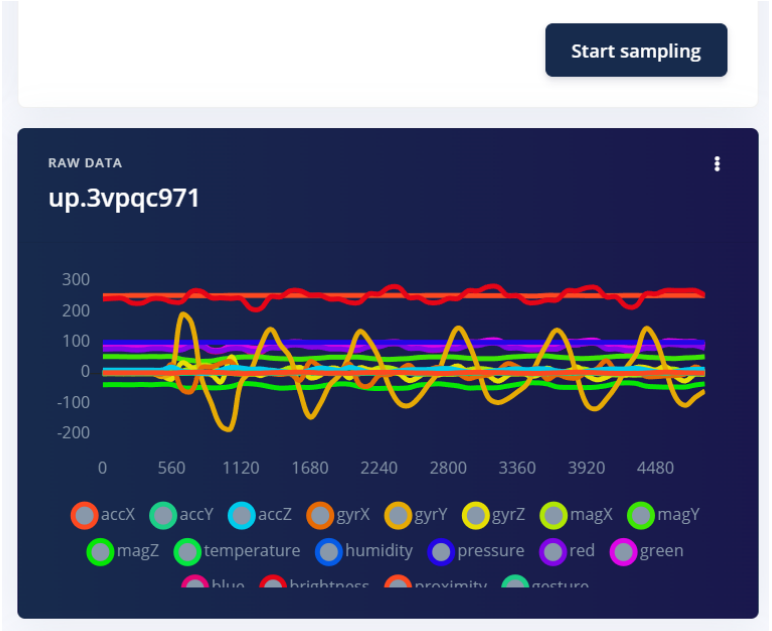
Zbieranie danych i uczenie modelu

- po podłączeniu mikrokontrolera możemy przejść do zbierania danych (Data acquisition)

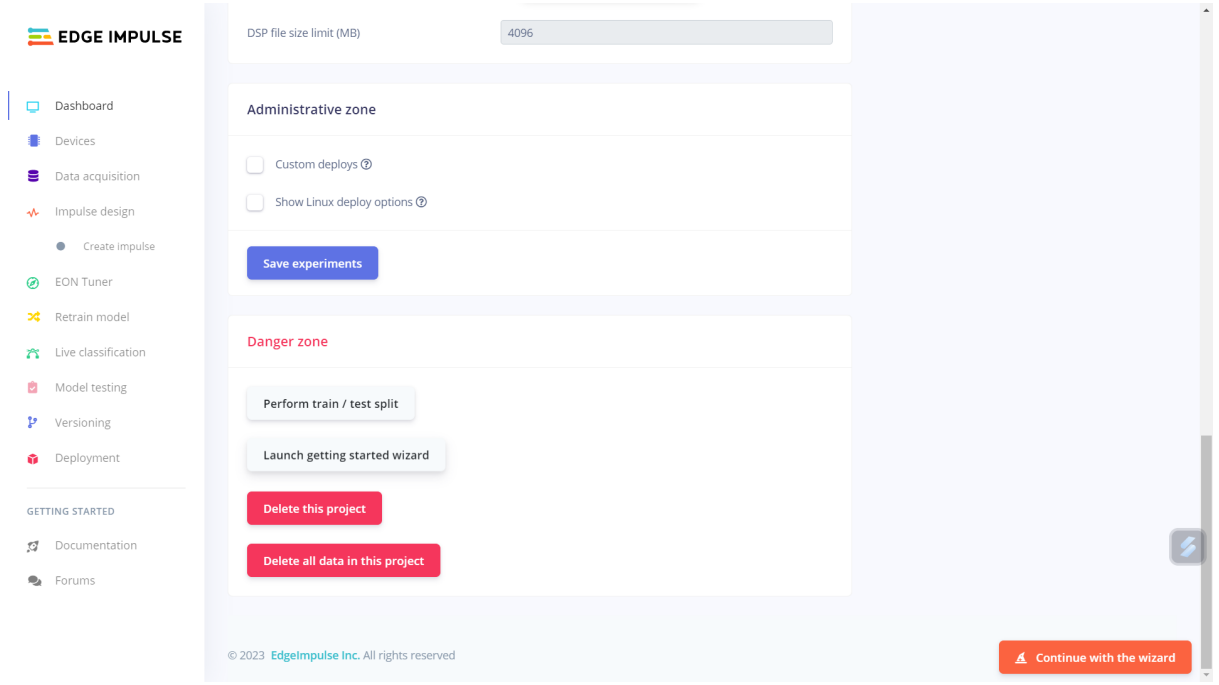


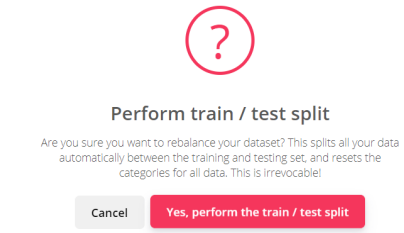
- w tym celu należy przejść do zakładki Data acquisition
  - sprawdzić nazwę urządzenia
  - opisać Label czyli nazwę która będzie przyporządkowana zebranym danom
  - długość trwania próbki
  - sensory
    - Interactional - RGB, jasność, zbliżenie, gesty
    - Environmental - temperatura, wilgotność, ciśnienie
    - Internal - akcelerometr, żyroskop, magnetometr
- w zależności od wybranego sensora tworzymy zbiór danych poprzez "Start sampling"
  - w przypadku zbierania danych z mikrofonu podczas jednej sesji zbierania danych możemy kilkukrotnie wypowiedzieć komendę a następnie próbkę podzielić

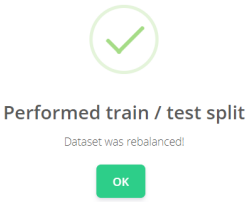




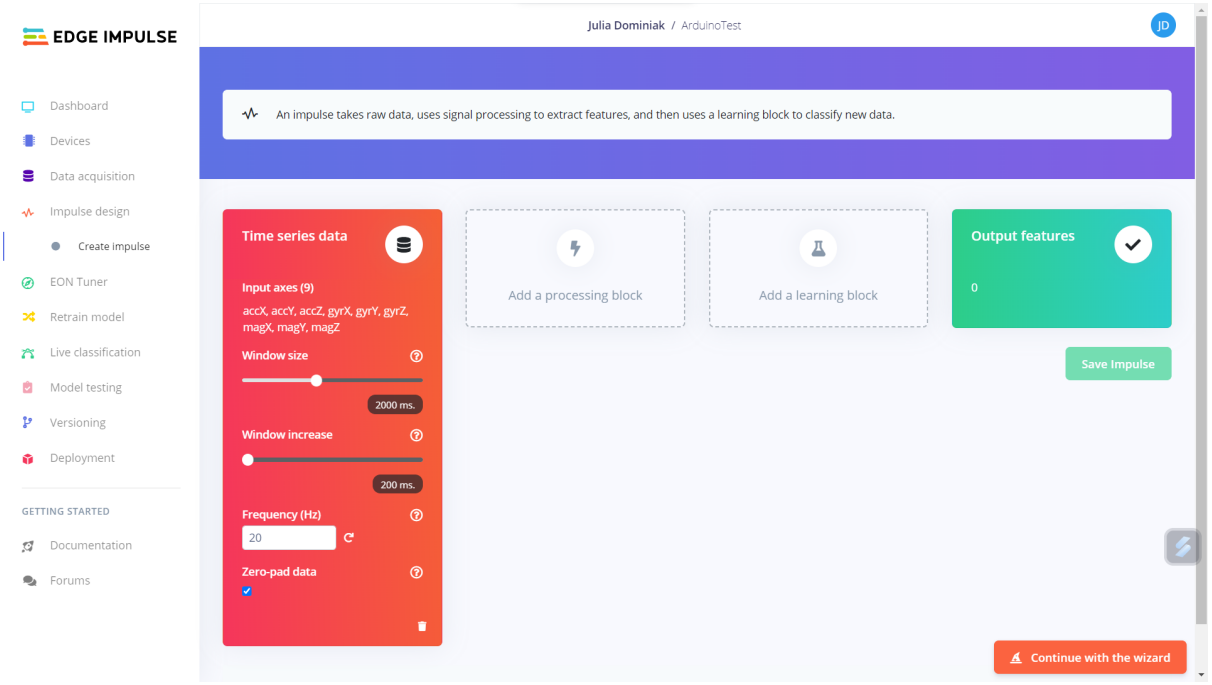
- ➔ w przypadku uczenia dwóch stanów, minimalną ilością będzie około 20-30 sesji nagrywania danych (ok. 1 minuty)
- ➔ należy też pamiętać o zebraniu takiej samej ilości dla stanu neutralnego/błędnego/nieokreślonego
- ➔ po ukończeniu nagrywania należy zoptymalizować i podzielić dane na mniejsze porcje w tym celu wchodzimy w Dashboard i na samym dole wybieramy Perform train/test split



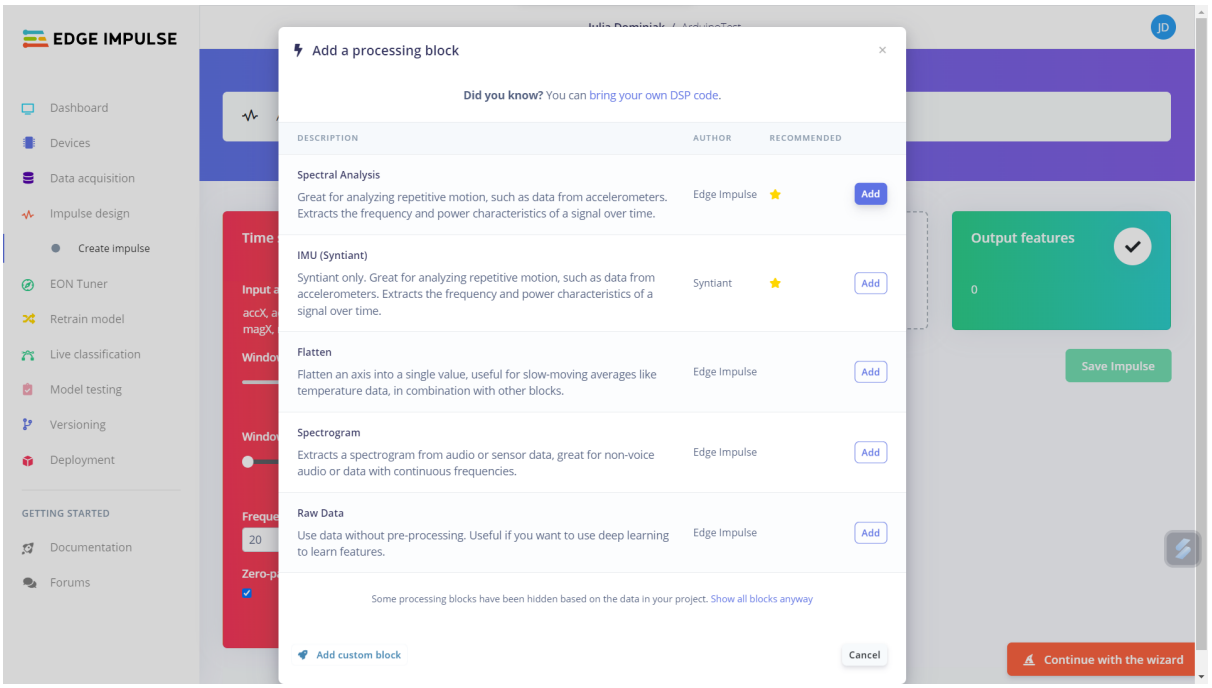


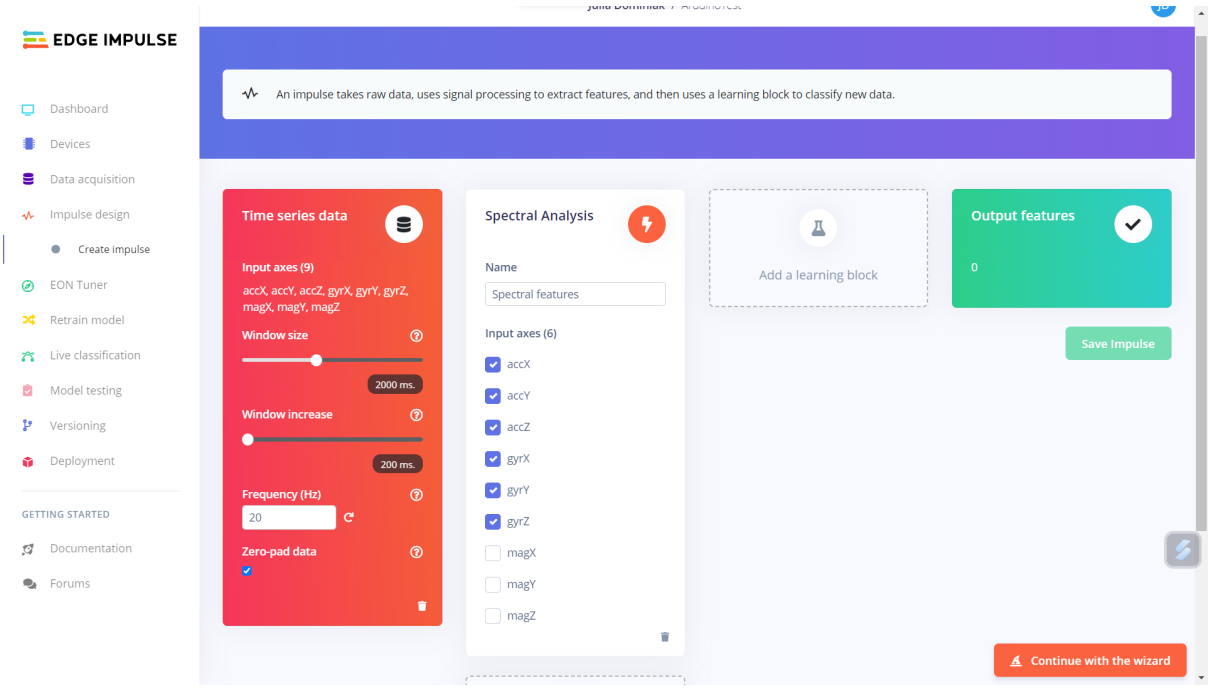


➔ kolejnym krokiem jest stworzenie "Impulsu" czyli pipelinu dla uczenia maszynowego
- ◆ blok wejściowy - wskazuje typ danych wejściowych
  - lista wszystkich osi wejściowych
  - rozmiar okna to rozmiar nieprzetworzonych funkcji używanych do trenowania
  - zwiększenie okna służy do sztucznego tworzenia większej ilości funkcji
  - częstotliwość jest obliczana automatycznie na podstawie próbek treningowych i nie jest wskazane jej zmienianie
  - w tym bloku istnieje również opcja wypełnienia brakujących funkcji wartościami zerowymi

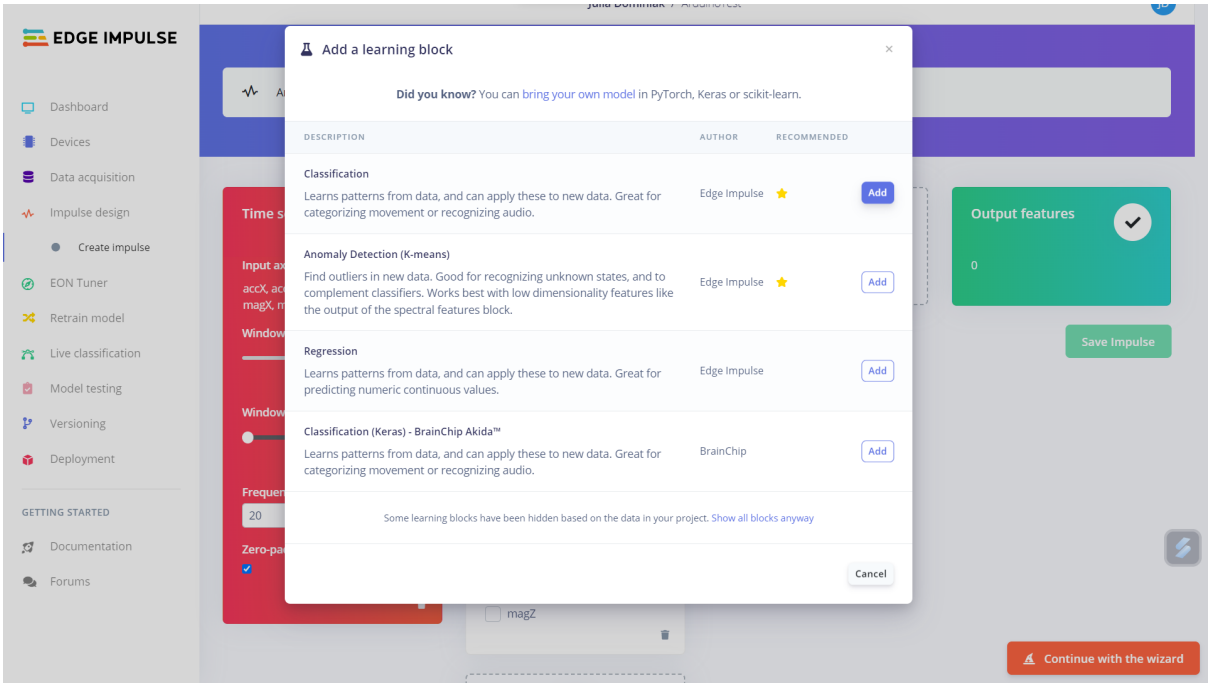


- processing block - blok składa się z operacji DSP (Digital Signal Processing), która służy do wyodrębnienia funkcji, na których uczy się nasz model
  - operacja różni się w zależności od używanych danych i to my wybieramy stosowany blok
  - najbardziej zalecane bloki są oznaczane gwiazdką
  - mamy tu również możliwość wybrania istotnych odczytów dla naszego uczenia

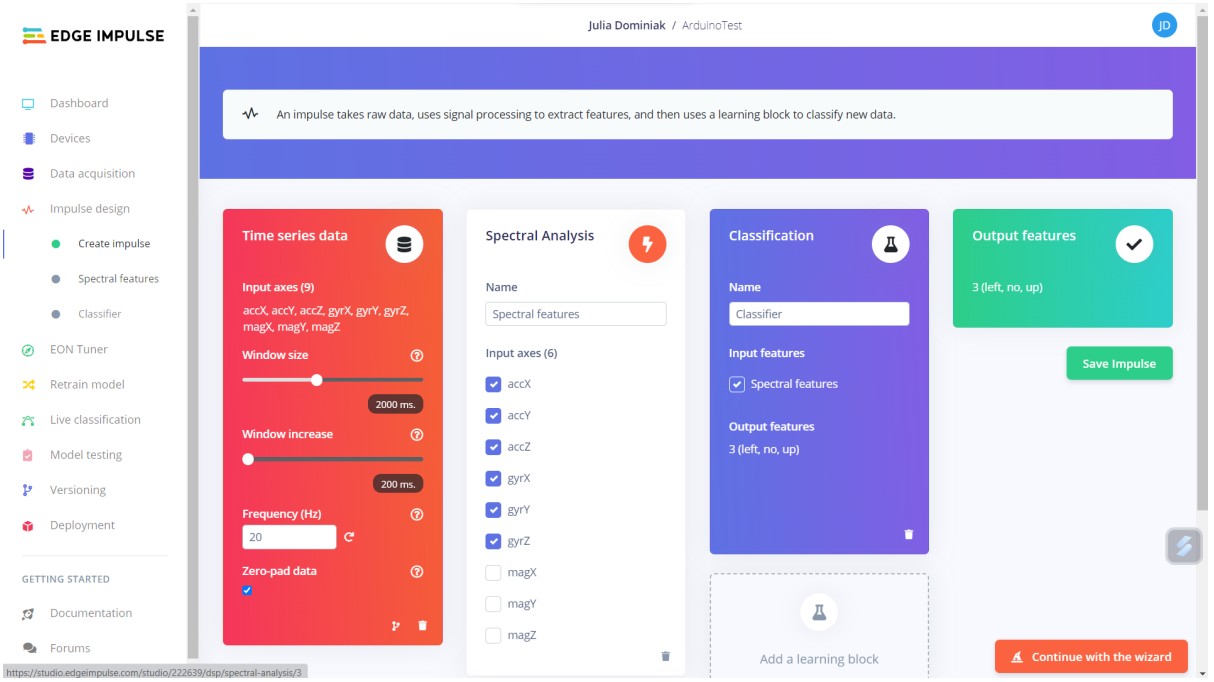




- learning block - tutaj nadchodzi wybór sieci neuronowej wyszkolonej do uczenia się na danych
  - tutaj też wybór zależy od nas, a polecane wybory są oznaczone gwiazdką

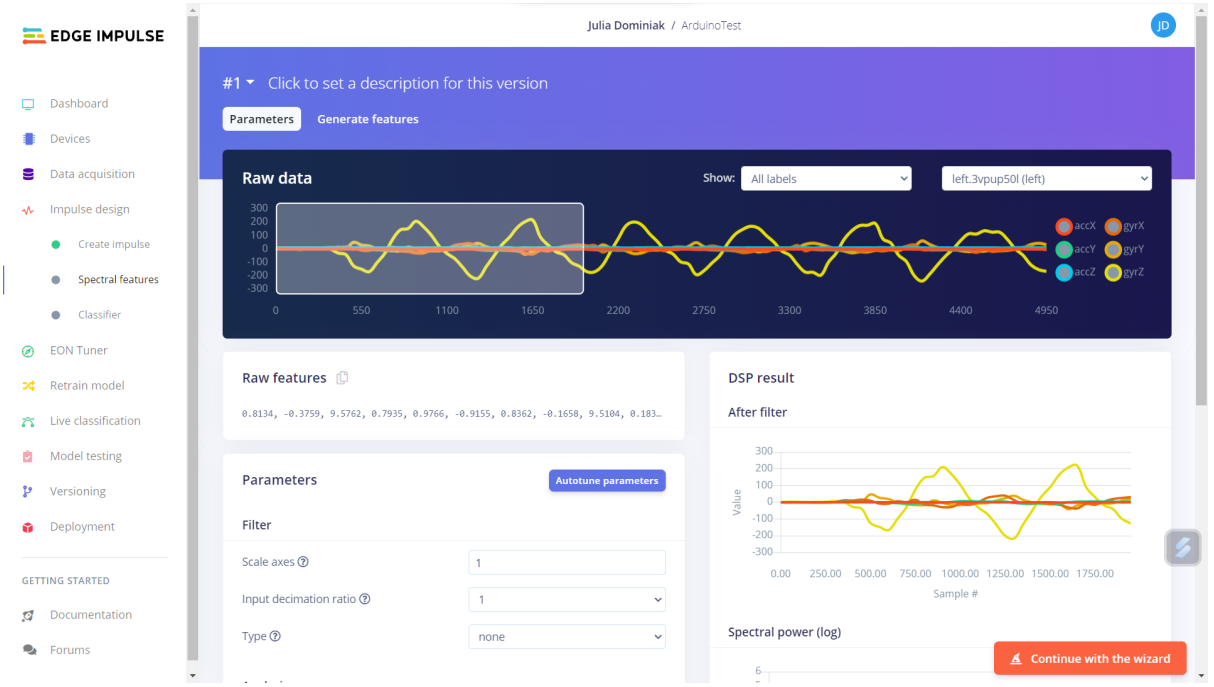


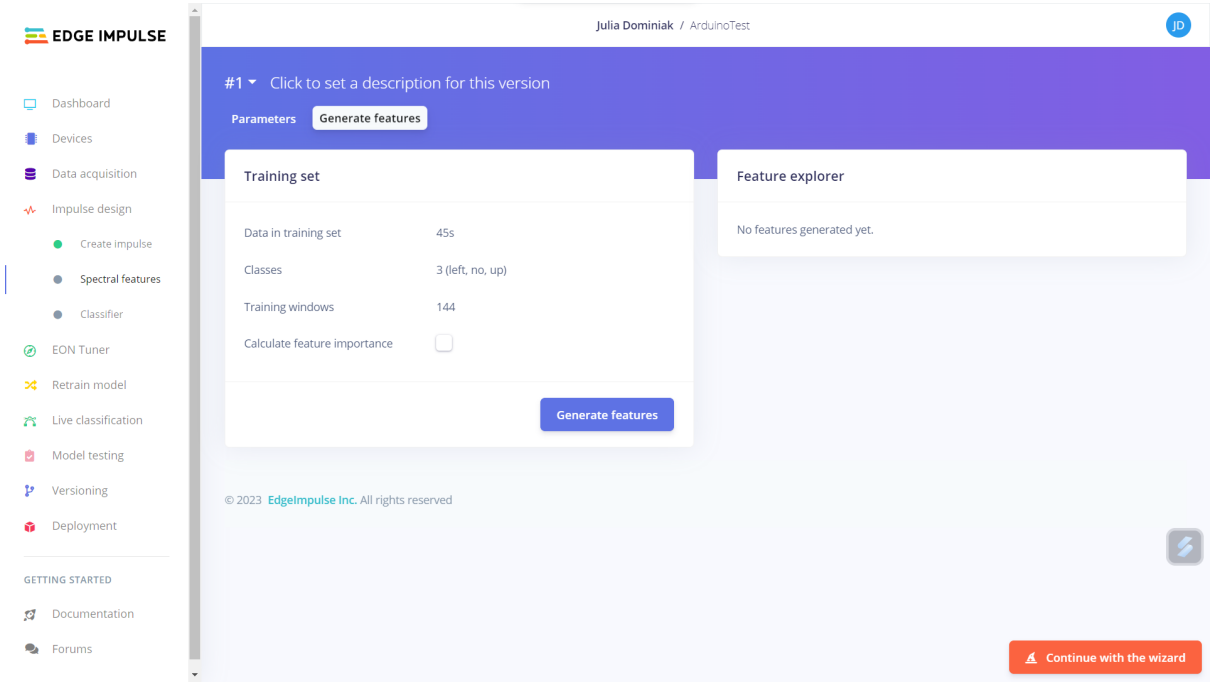
➔ po dostosowaniu wszystkich bloków zapisujemy impuls

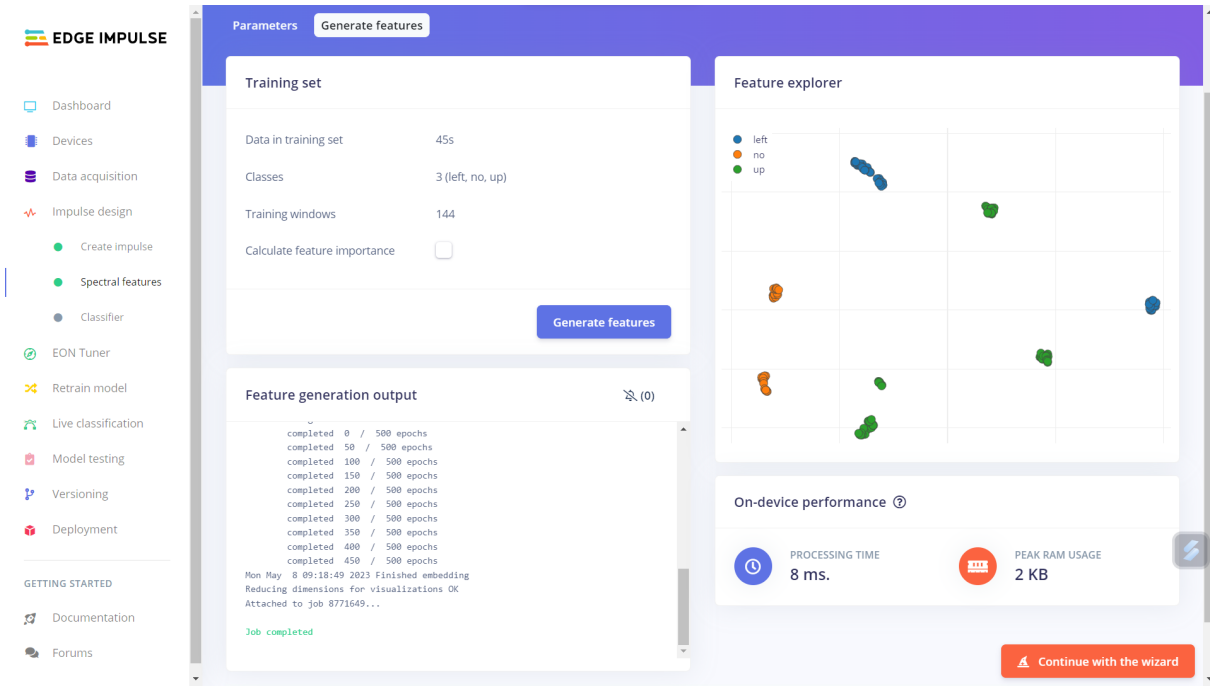


➔ kolejnym krokiem w tworzeniu impulsu jest wygenerowanie spectral features

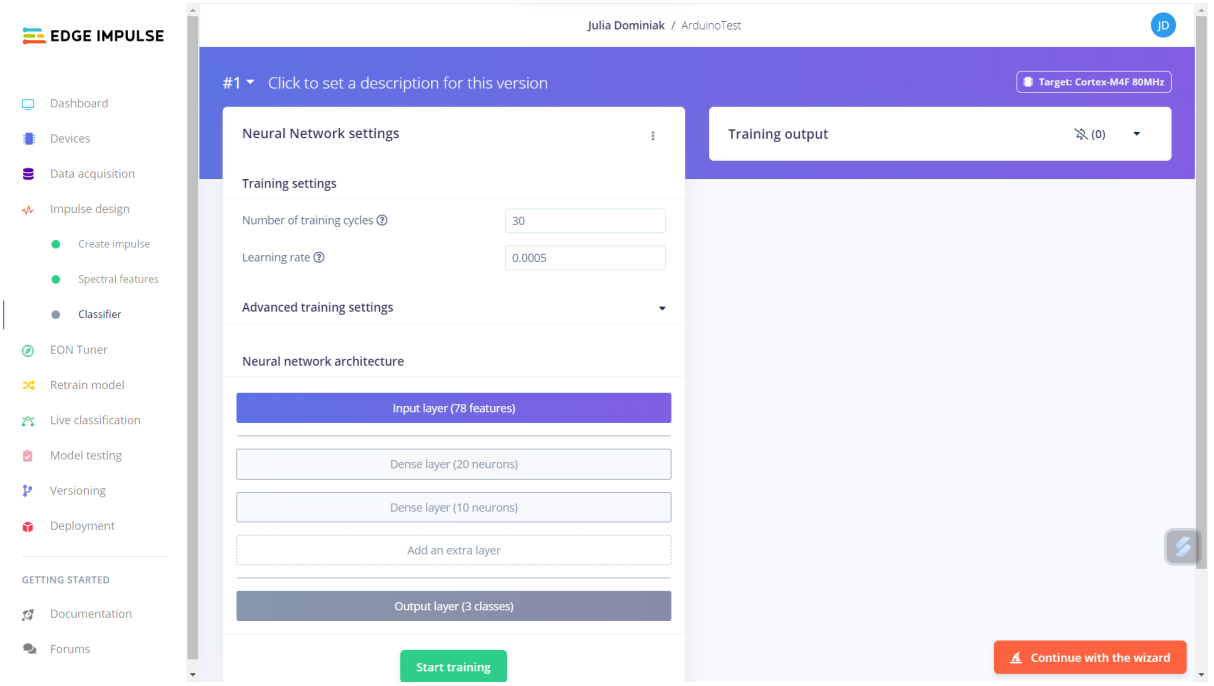
- jest to wyodrębnienie częstotliwości, mocy innych cech sygnału
- po tym etapie będzie już możliwe rozróżnienie cech dla poszczególnych zbiorów danych (labels)

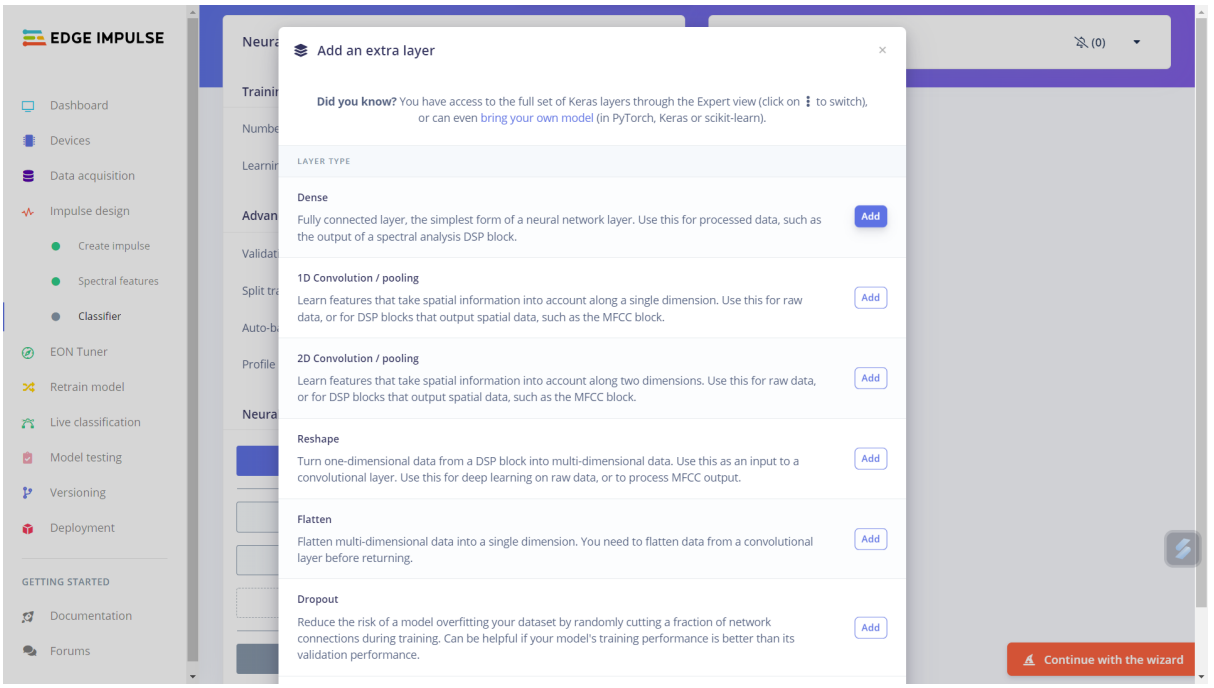


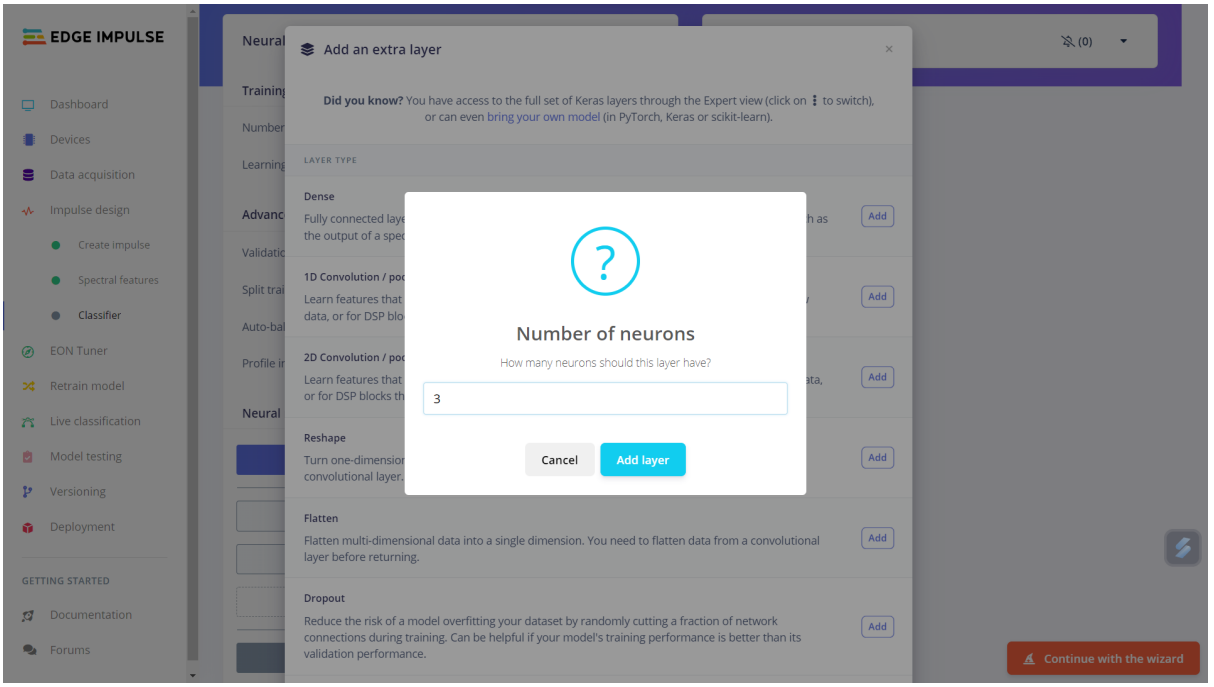


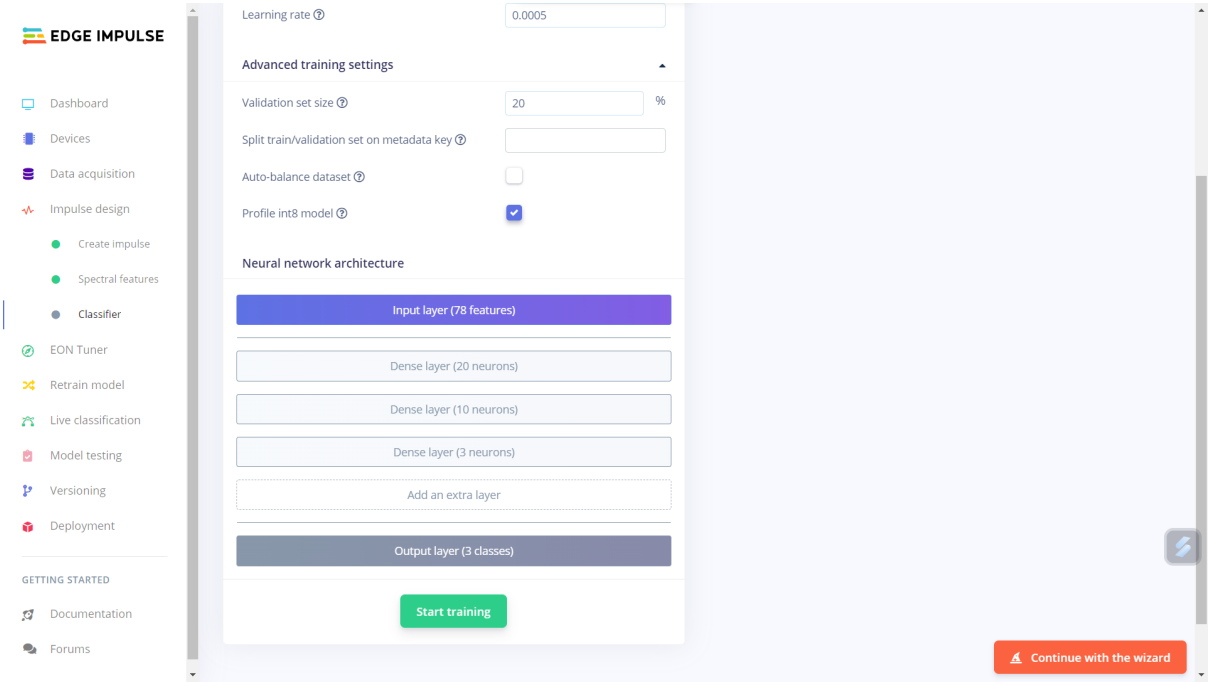


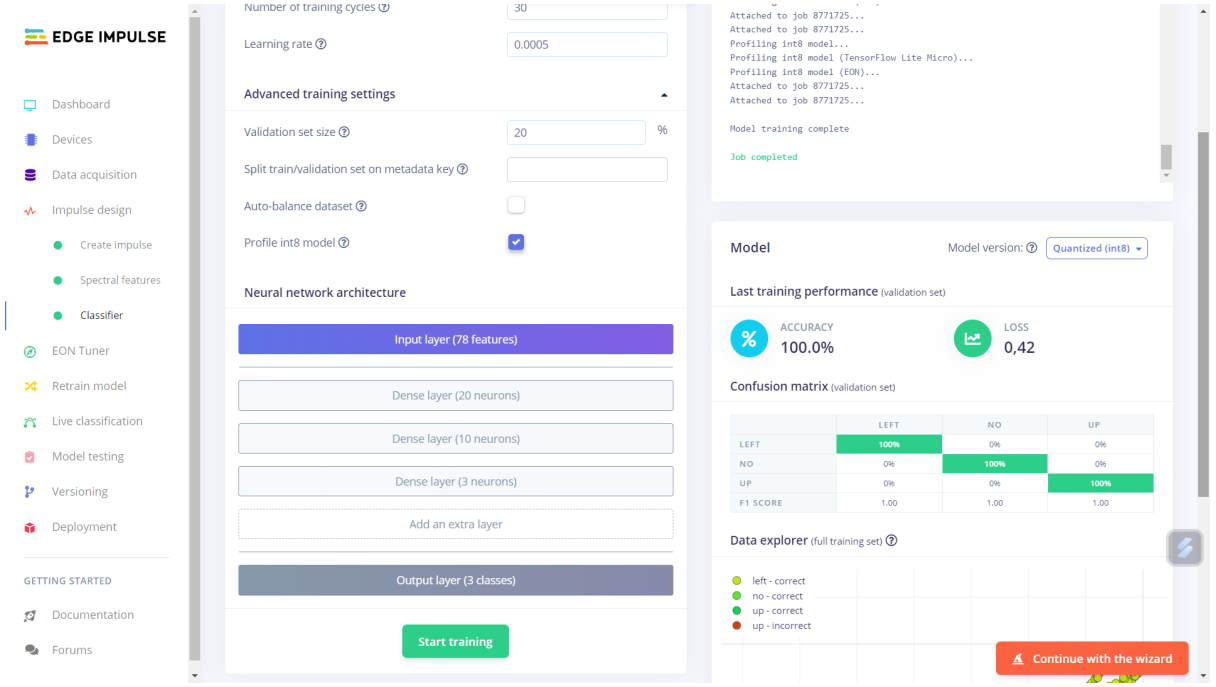
➔ ostatnim etapem jest zrealizowanie uczenia (w poniższym przykładzie jest to klasyfikator - Classifier)
- ◆ oprócz ustawień sieci, możemy jeszcze edytować architekturę
  - zalecane jest dodanie trzeciej warstwy (Dense) której liczba neuronów będzie równa liczbie zbiorów które rozpoznajemy
  - po dodaniu rozpoczynamy trenowanie sieci - Strat traing


EDGE IMPULSE
Julia Dominiak / ArduinoTest
JD
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Spectral features
Classifier
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
#1 Click to set a description for this version
Target: Cortex-M4F 80MHz
Neural Network settings
Training output
(0)
Training settings
Number of training cycles
30
Learning rate
0.0005
Advanced training settings
Neural network architecture
Input layer (78 features)
Dense layer (20 neurons)
Dense layer (10 neurons)
Add an extra layer
Output layer (3 classes)
Start training
Continue with the wizard


EDGE IMPULSE
Add an extra layer
Did you know? You have access to the full set of Keras layers through the Expert view (click on ⋮ to switch), or can even bring your own model (in PyTorch, Keras or scikit-learn).
LAYER TYPE
Dense
Fully connected layer, the simplest form of a neural network layer. Use this for processed data, such as the output of a spectral analysis DSP block.
Add
1D Convolution / pooling
Learn features that take spatial information into account along a single dimension. Use this for raw data, or for DSP blocks that output spatial data, such as the MFCC block.
2D Convolution / pooling
Learn features that take spatial information into account along two dimensions. Use this for raw data, or for DSP blocks that output spatial data, such as the MFCC block.
Reshape
Turn one-dimensional data from a DSP block into multi-dimensional data. Use this as an input to a convolutional layer. Use this for deep learning on raw data, or to process MFCC output.
Flatten
Flatten multi-dimensional data into a single dimension. You need to flatten data from a convolutional layer before returning.
Dropout
Reduce the risk of a model overfitting your dataset by randomly cutting a fraction of network connections during training. Can be helpful if your model's training performance is better than its validation performance.
Continue with the wizard

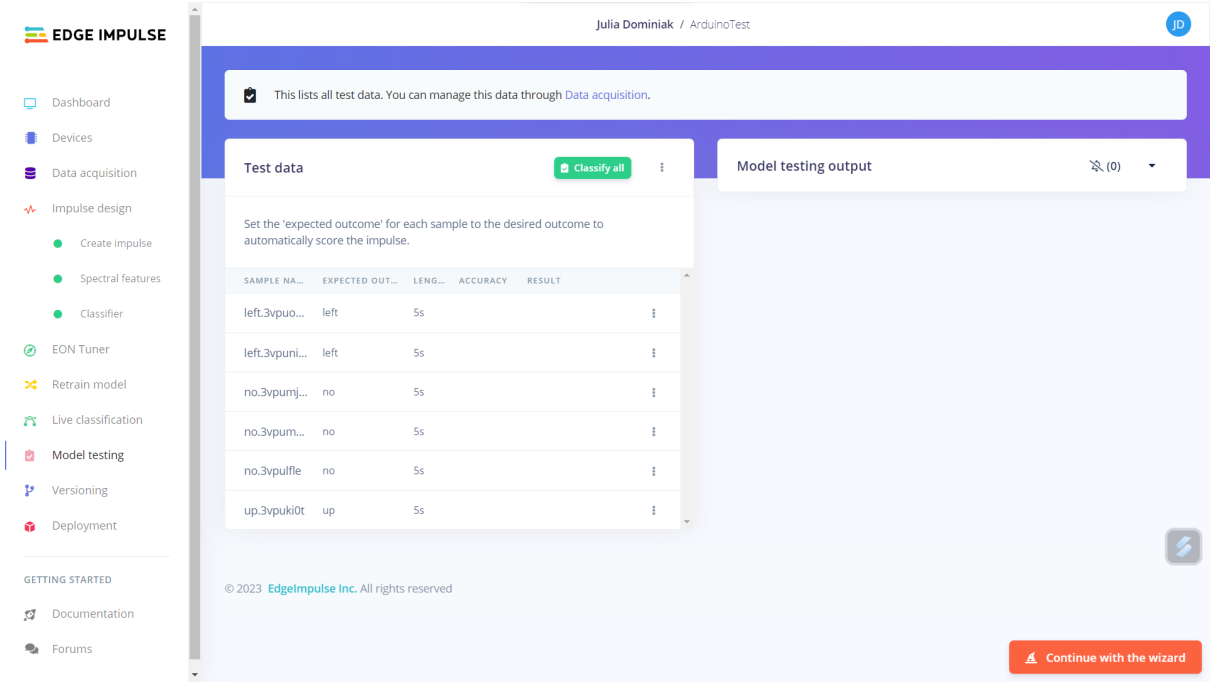

EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Spectral features
Classifier
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Add an extra layer
Did you know? You have access to the full set of Keras layers through the Expert view (click on ⋮ to switch), or can even bring your own model (in PyTorch, Keras or scikit-learn).
LAYER TYPE
Dense
Add
1D Convolution / poo
Add
2D Convolution / poo
Add
Reshape
Add
Flatten
Flatten multi-dimensional data into a single dimension. You need to flatten data from a convolutional layer before returning.
Add
Dropout
Reduce the risk of a model overfitting your dataset by randomly cutting a fraction of network connections during training. Can be helpful if your model's training performance is better than its validation performance.
Add
Number of neurons
How many neurons should this layer have?
3
Cancel
Add layer
Continue with the wizard


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Spectral features
Classifier
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Learning rate
0.0005
Advanced training settings
Validation set size
20
%
Split train/validation set on metadata key
Auto-balance dataset
Profile int8 model
Neural network architecture
Input layer (78 features)
Dense layer (20 neurons)
Dense layer (10 neurons)
Dense layer (3 neurons)
Add an extra layer
Output layer (3 classes)
Start training
Continue with the wizard



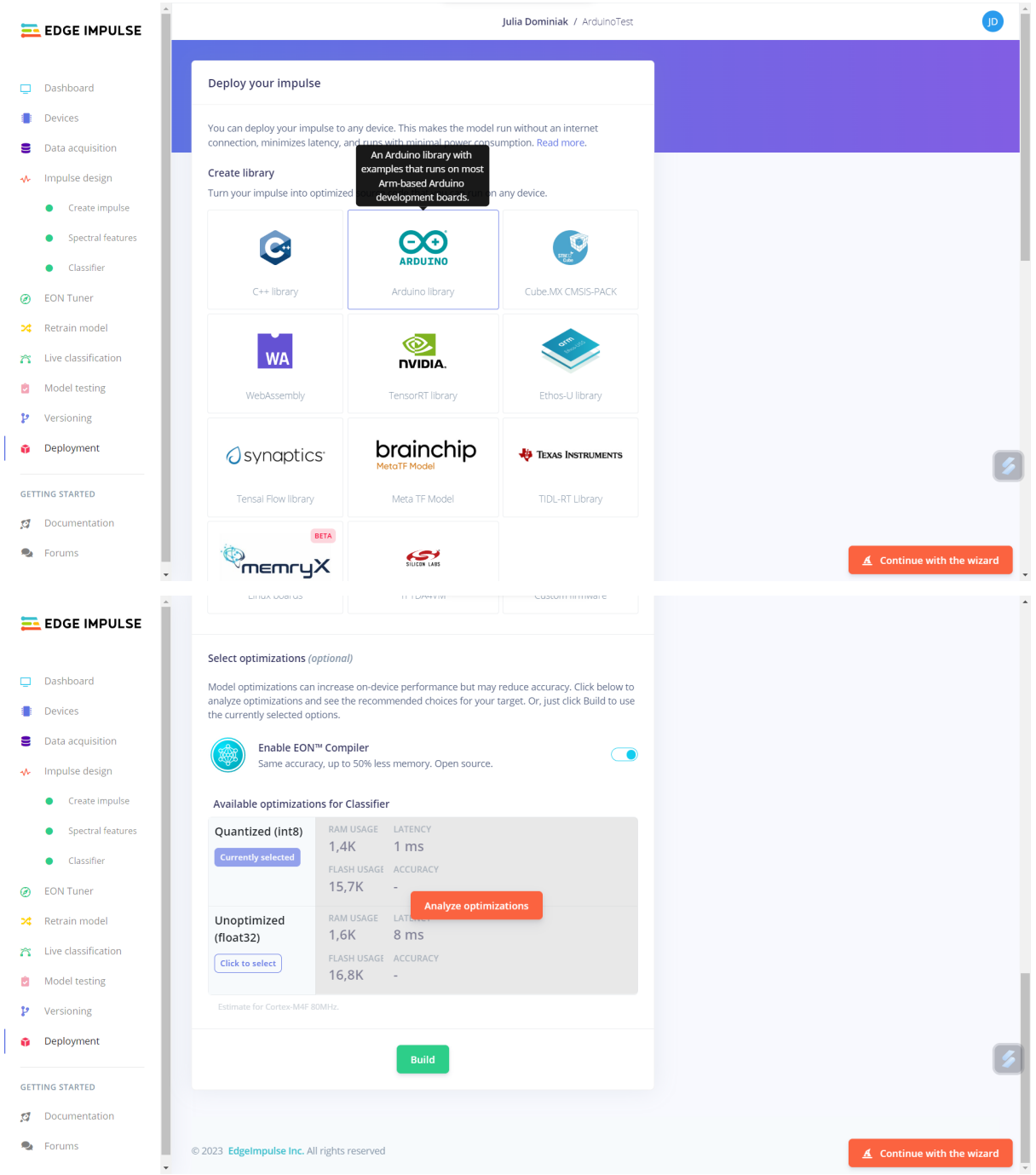
➔ po wyuczeniu sieci zalecane jest jej przetestowanie na próbkach które były oddzielone od zbioru na którym sieć była uczona

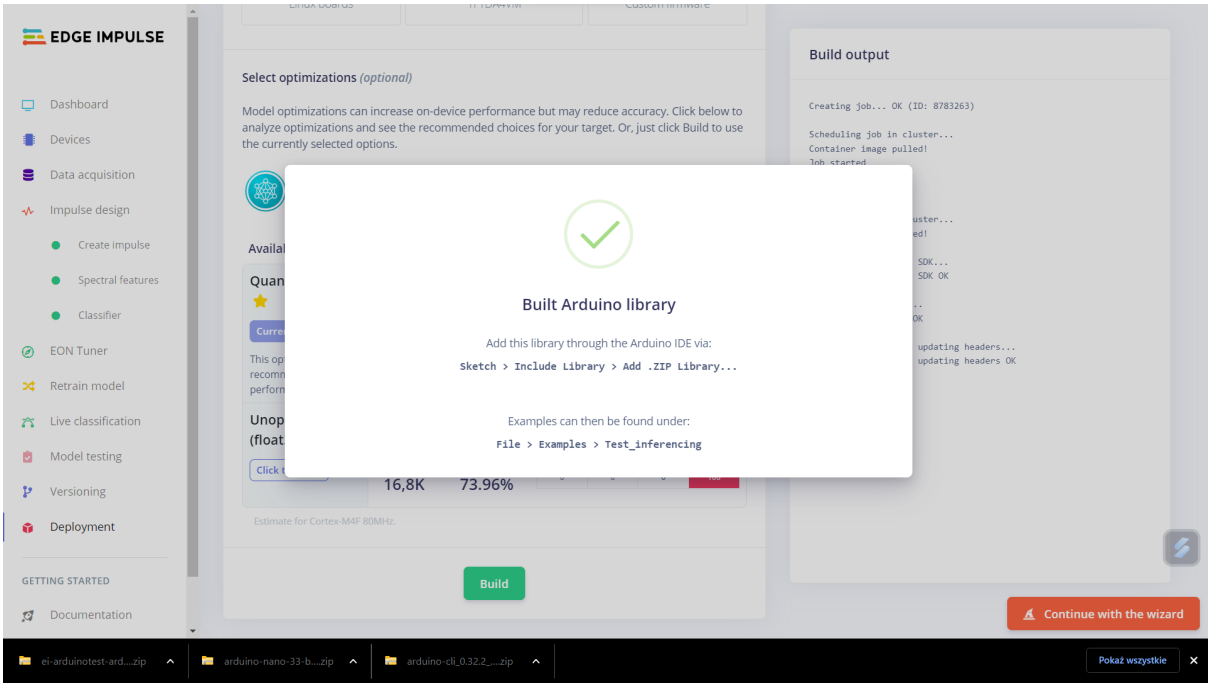
- wybieramy z listy Model testing i uruchamiamy proces



➔ po sprawdzeniu sieci ostatnim etapem jest wyeksportowanie pliku który będzie możliwy do wgrania w Arduino IDE

- z listy wybieramy Deployment
- Arduino
- dokonujemy Analyze optimizations
- Build


EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Spectral features
Classifier
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
Julia Dominiak / ArduinoTest
JD
Deploy your impulse
You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.
Create library
Turn your impulse into optimized
n any device.
An Arduino library with examples that runs on most Arm-based Arduino development boards.
C++ library
Arduino library
Cube.MX CMSIS-PACK
WebAssembly
TensorRT library
Ethos-U library
Tensal Flow library
Meta TF Model
TIDL-RT Library
BETA
Continue with the wizard
EDGE IMPULSE
Dashboard
Devices
Data acquisition
Impulse design
Create impulse
Spectral features
Classifier
EON Tuner
Retrain model
Live classification
Model testing
Versioning
Deployment
GETTING STARTED
Documentation
Forums
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 Classifier
Quantized (int8)
Currently selected
RAM USAGE
1,4K
LATENCY
1 ms
FLASH USAGE
15,7K
ACCURACY
-
Analyze optimizations
Unoptimized (float32)
Click to select
RAM USAGE
1,6K
8 ms
FLASH USAGE
16,8K
ACCURACY
-
Estimate for Cortex-M4F 80MHz.
Build
© 2023 EdgeImpulse Inc. All rights reserved
Continue with the wizard



➔ uzyskany plik wgrywamy do Arduino IDE w postaci biblioteki .zip

➔ testujemy nasz model