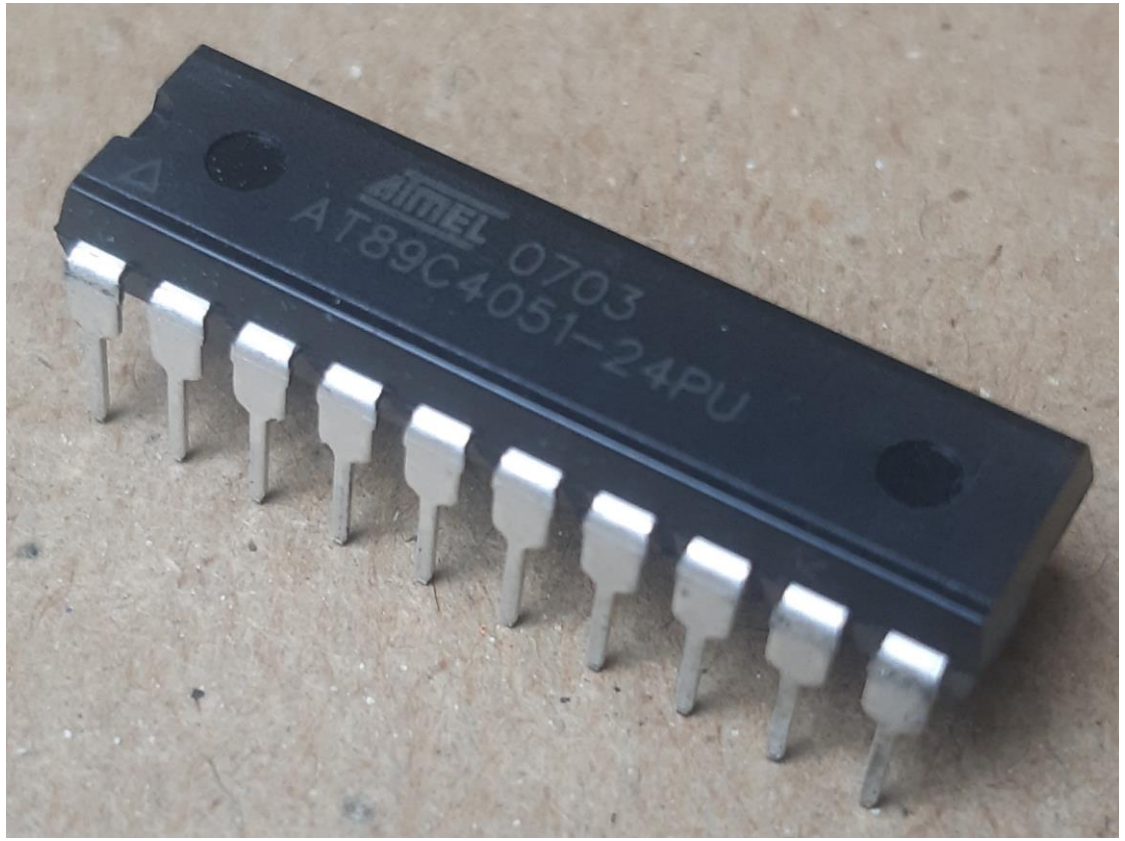
Arduino Nano 33 BLE Sense Rev 2

Mikrokontroler

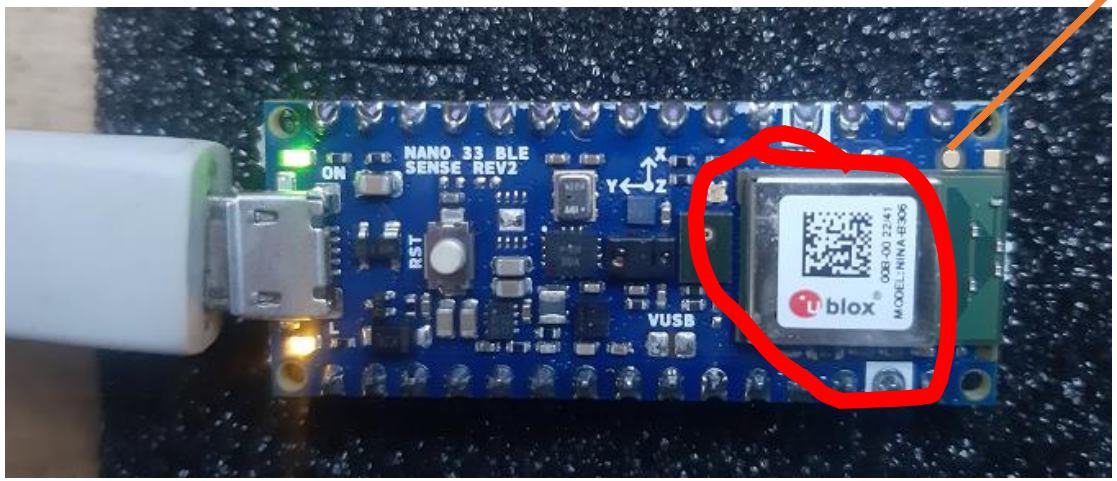


cały komputer w małym opakowaniu ☺

https://forbot.pl/blog/mikrokontroler-wszystko-co-powinniscie-wiedziec-o-jego-dzialaniu-id1314

https://pl.wikipedia.org/wiki/Mikrokontroler

Arduino Nano 33 BLE Sense Rev 2



moduł Bluetooth® 5 z multiprotocol radio NINA B306 module, oparty na Nordic nRF52480 i zawierający ARM Cortex M4F

ARM to brytyjska firma, która zajmuje się projektowaniem i licencjonowaniem architektury ARM. Inni producenci produkują na jej podstawie mikrokontrolery.

https://en.wikipedia.org/wiki/ARM_Cortex-M#Cortex-M4

Arduino

Edukacyjna platforma elektroniczna open source oparta na łatwym w użyciu sprzęcie i oprogramowaniu

Arduino to hardware (board) oraz software (program)

tania platforma edukacyjna do nauki programowania oraz elektroniki

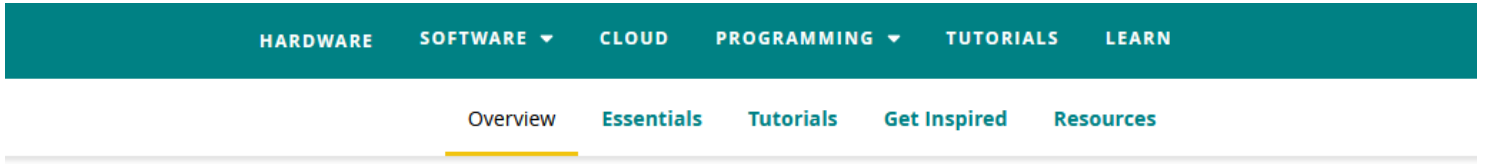
do programowania używany jest język oparty na Wiring oraz Arduino IDE oparte na technologii Processing.

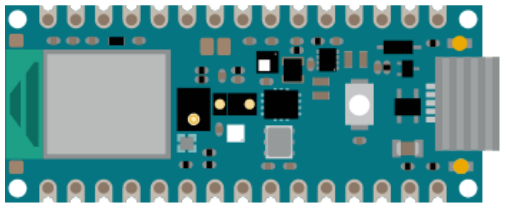
Jest przeznaczona dla każdego, kto tworzy interaktywne projekty.

Arduino bada otoczenie, otrzymując dane z wielu czujników i wpływa na nie, sterując światłami, silnikami i innymi urządzeniami elektronicznymi.

https://www.arduino.cc/

Arduino setup

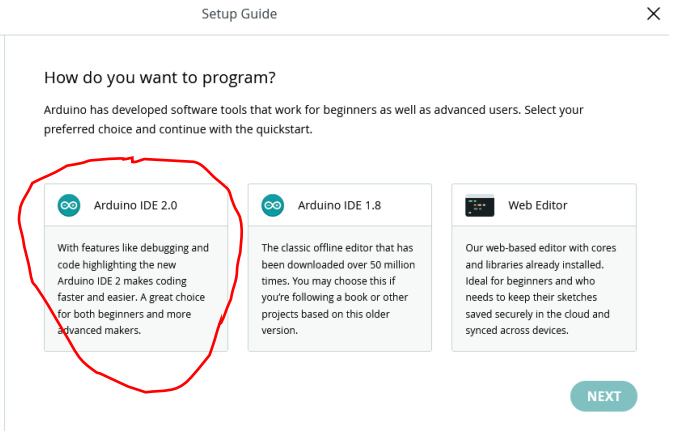




Nano 33 BLE Sense Rev2

The Arduino Nano 33 BLE Sense Rev2 combines a tiny form factor, different environment sensors and the possibility to run AI using TinyML and TensorFlow™ Lite. Whether you are looking at creating your first embedded ML application or you want to use Bluetooth® Low Energy to connect your project to your phone, the Nano 33 BLE Sense Rev2 will make that journey easy.

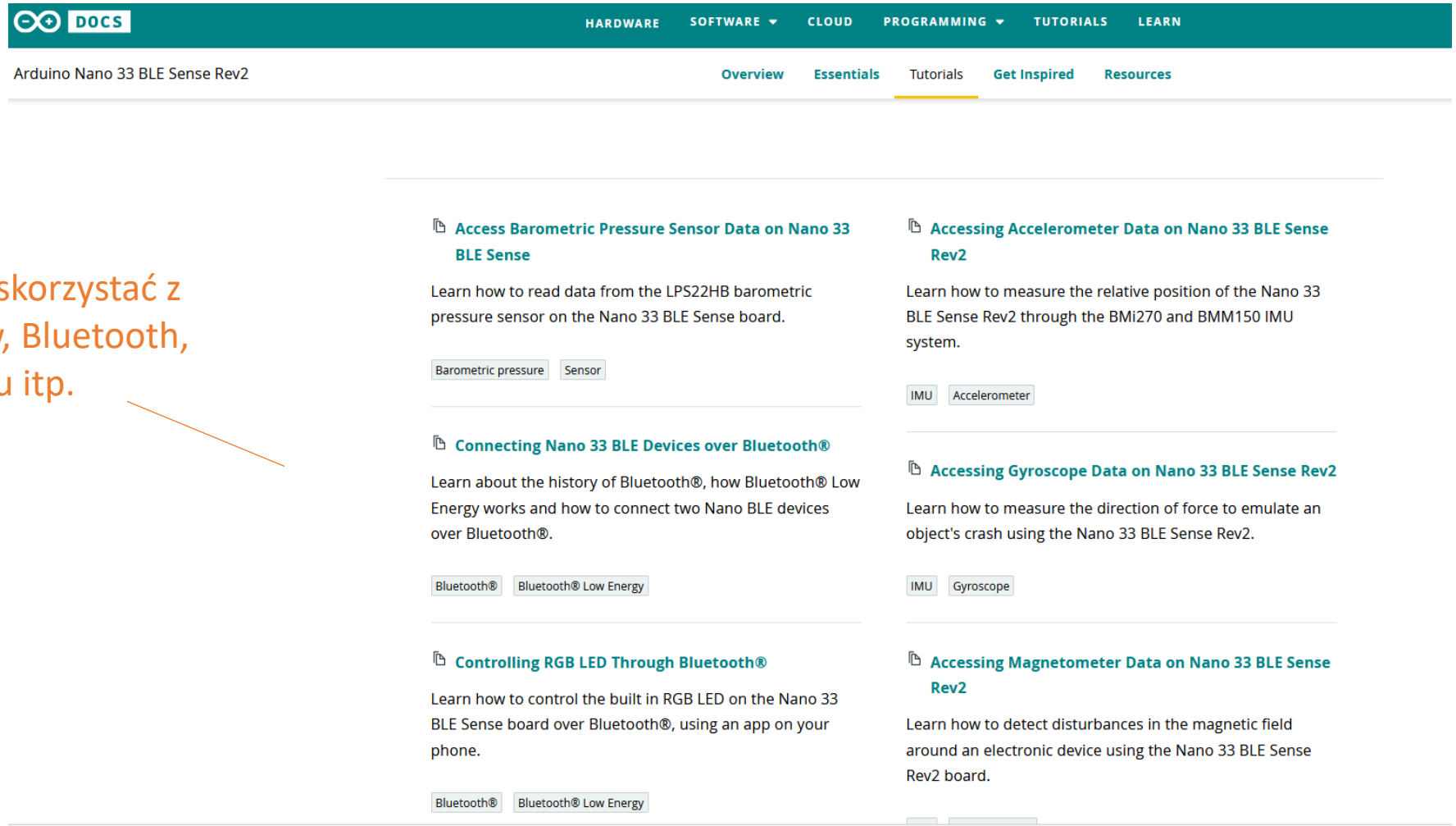
SETUP GUIDE → PINOUT DATASHEET



https://docs.arduino.cc/hardware/nano-33-ble-sense-rev2

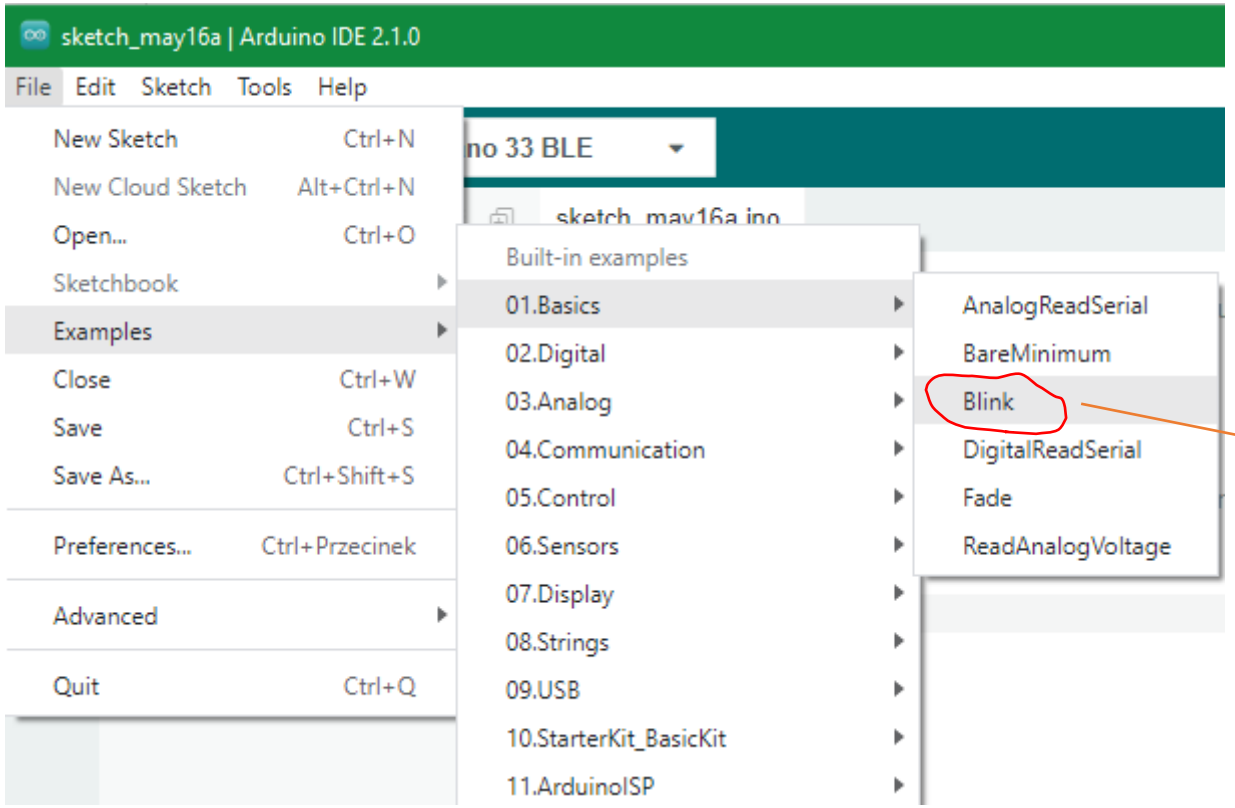
Arduino tutorials

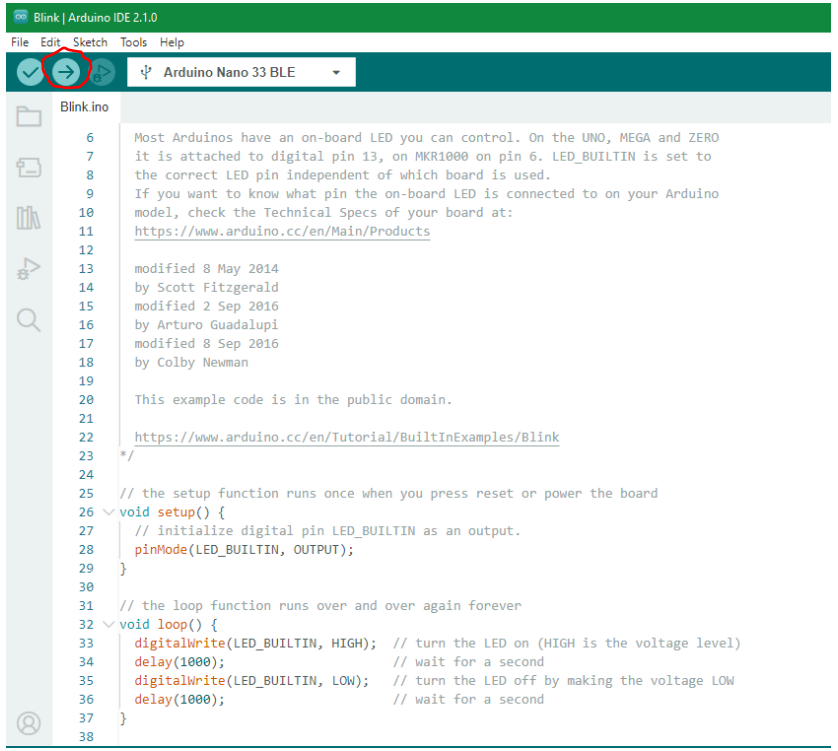
opisy jak skorzystać z czujników, Bluetooth, mikrofonu itp.



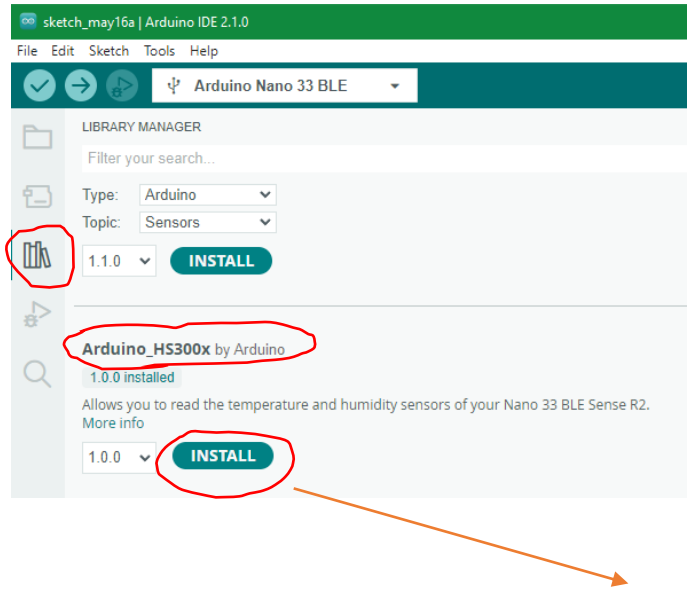
https://docs.arduino.cc/hardware/nano-33-ble-sense-rev2

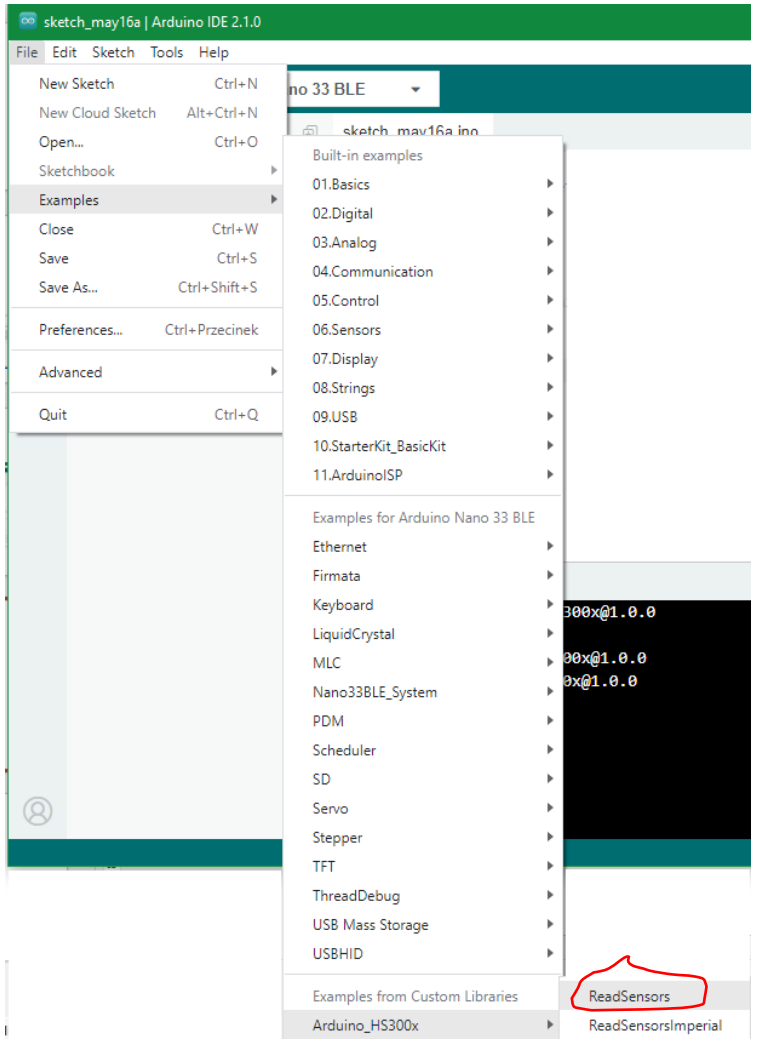
Arduino Blink example

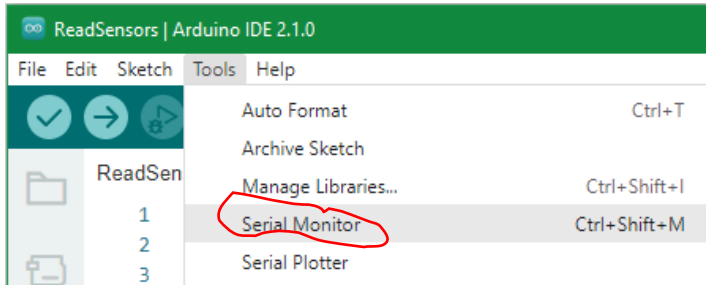


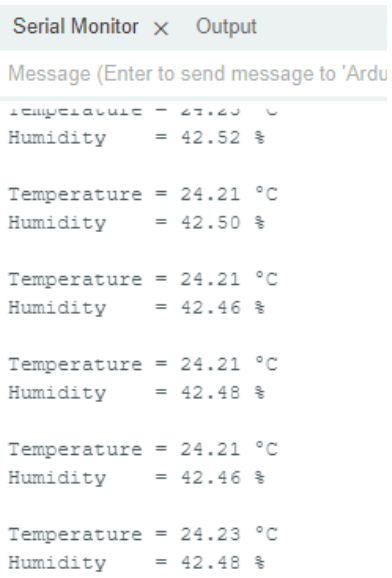


Arduino temperture, humidity

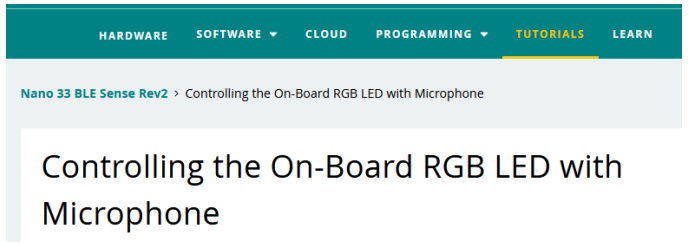








Arduino examples



TensorFlow

Kompleksowa platforma open source do uczenia maszynowego

obsługiwane platformy: m.in. Android, iOS, Raspberry Pi

ML na małych urządzeniach (mobilnych, wbudowanych, brzegowych)

przeznaczony do uruchamiania modeli uczenia maszynowego na mikrokontrolerach i innych urządzeniach z zaledwie kilkoma kilobajtami pamięci.

Tensor Flow Lite

TensorFlow 2.0 został zaprojektowany, aby ułatwić budowanie sieci neuronowych do uczenia maszynowego

TensorFlow Lite for Microcontrollers

Zoptymalizowany pod katem:
opóźnienia (brak łączności z serwerem - chmurą)
prywatności (dane zapisywane są lokalnie),
łączności (internet nie jest wymagany),
rozmiar (model jest zredukowany) i
pobór mocy

napisany w C++ 17 i wymaga platformy 32-bitowej

https://www.tensorflow.org/?hl=pl
https://www.tensorflow.org/lite/microcontrollers?hl=pl
https://www.tensorflow.org/lite/guide?hl=pl

TensorFlow

CHALLENGE WINNERS

When we launched the TensorFlow Microcontoller Challenge, we invited developers to push the boundaries of TensorFlow Lite for Microcontrollers. We want to send a big thank you to everyone who participated for their incredibly inspiring submissions. And a huge congratulations to our five winners!

MAPPING DANCE

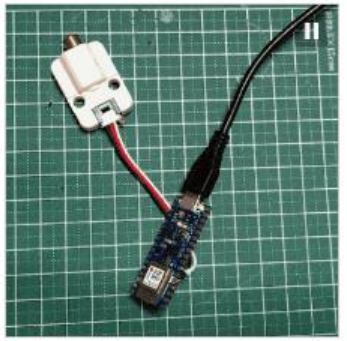
by Eduardo Padrón

Take control of lighting and video projections with your dance moves.

MOVE!

by Eunji Lee, Jueun Choi, Yeonhee Kim, Jonghyun Baek, Yongjae Kim

Stay active, using movement to control a variety of games.



SNORING GUARDIAN

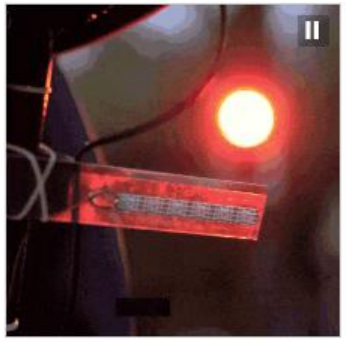
by Naveen Kumar

A snore-no-more device embedded in your pillow.

SQUATS COUNTER

by Manas Pange

Focus on your form, while this tracker counts your squats.



VOICE TURN

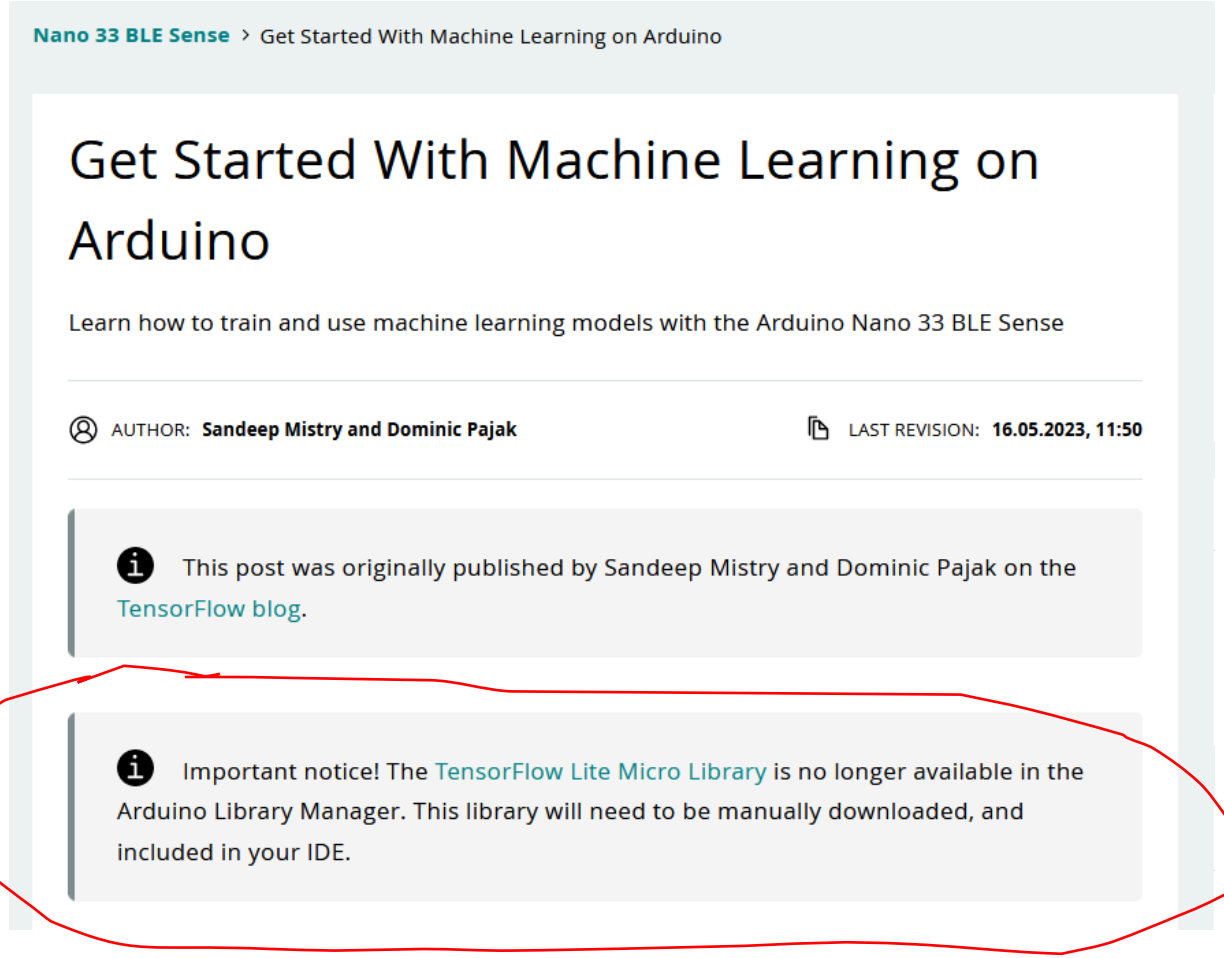
by Alvaro Gonzalez-Vila

A safer way for cyclists to signal using their voice.

przykłady użycia biblioteki TensorFlowLite

https://experiments.withgoogle.com/collection/tfliteformicrocontrollers

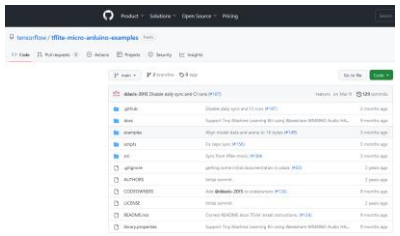
Arduino ML



https://docs.arduino.cc/tutorials/nano-33-ble-sense/get-started-with-machine-learning

Arduino ML

Opis instalacji biblioteki Arduino_TensorFlow:

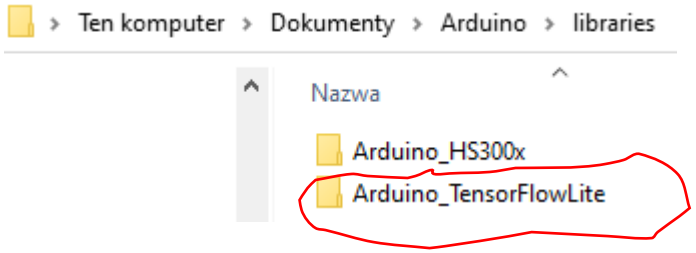


https://github.com/tensorflow/tflite-micro-arduino-examples

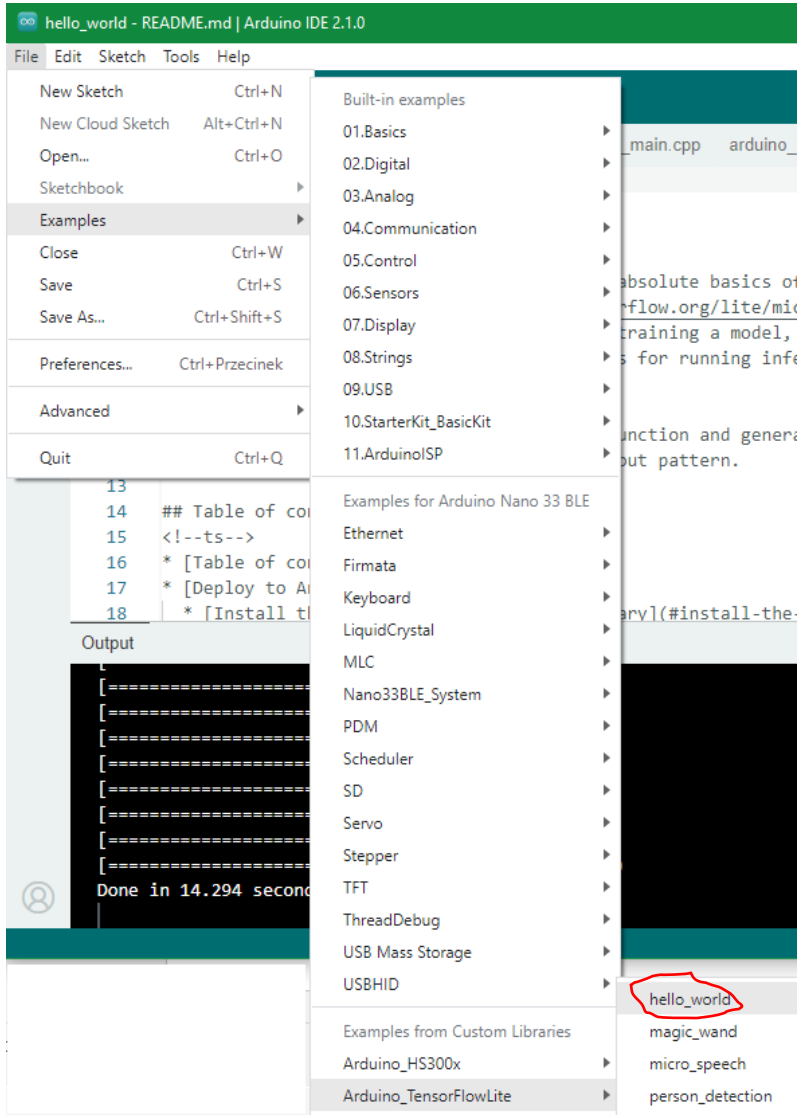
1. zainstalować Gita
2. przejść do folderu My Documents\Arduino\Libraries
3. uruchomić konsolę Git Bush Here w tym folderze
4. wpisać komendę:

git clone https://github.com/tensorflow/tflite-micro-arduino-examples Arduino_TensorFlowLite

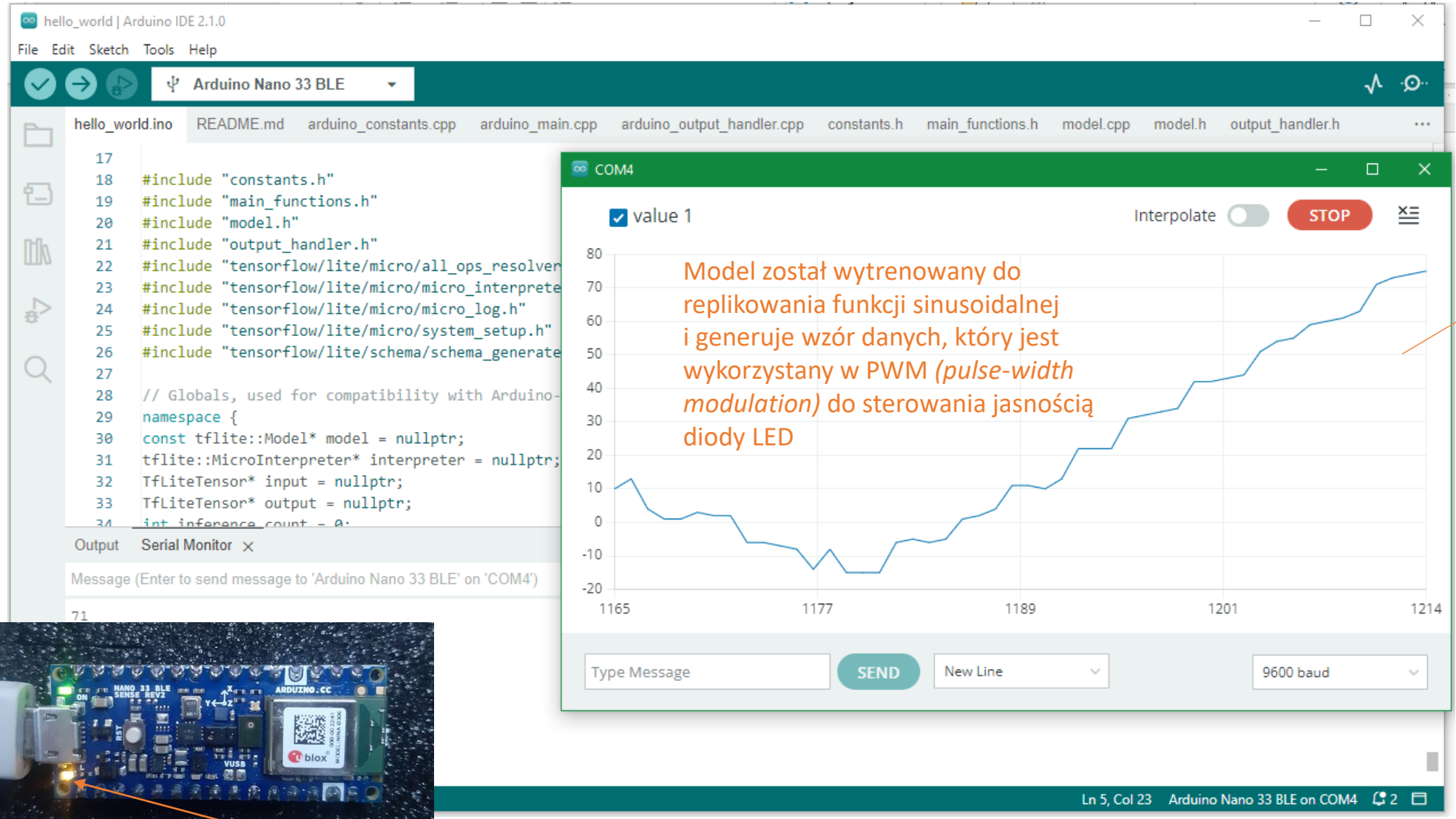
5. pojawi się nowy folder z biblioteką Arduino_TensorFLowLite



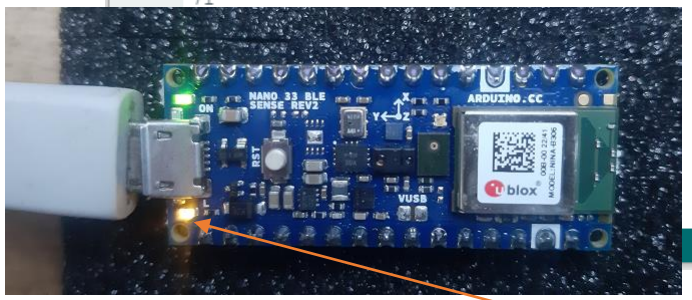
Arduino ML Hello World



Arduino ML Hello World

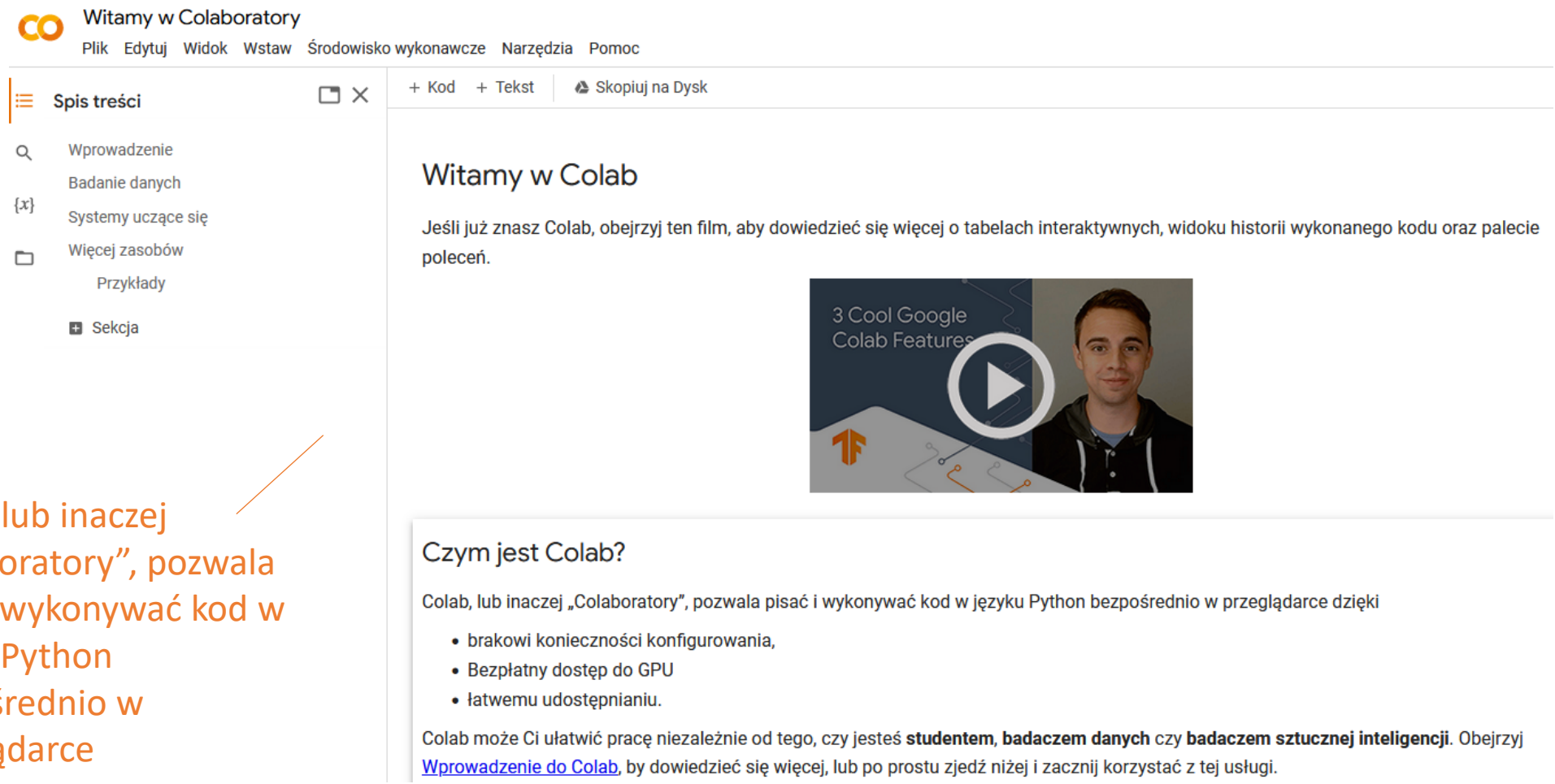


Serial Plotter



jasność diody LED zmienia się sinusoidalnie

Colaboratory



Colab, lub inaczej „Colaboratory", pozwala pisać i wykonywać kod w języku Python bezpośrednio w przeglądarce

https://colab.research.google.com/

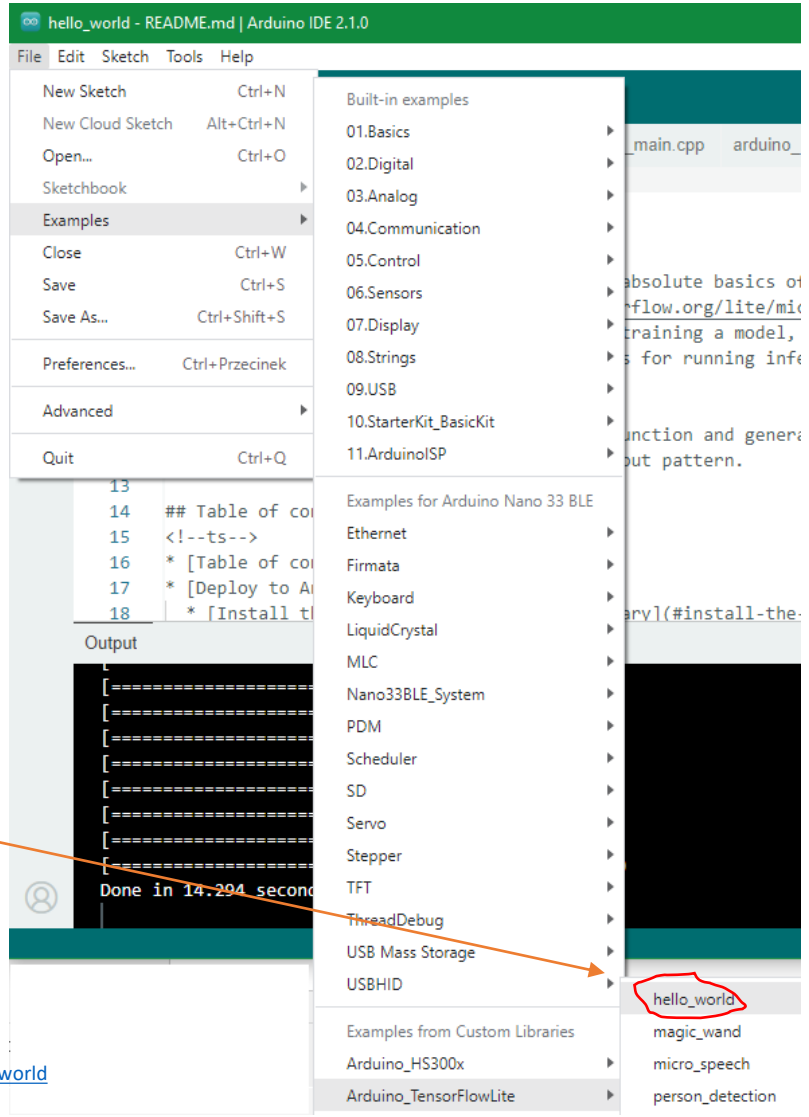
Arduino ML Hello World

instalacja innej wersji Hello World

https://oreil.ly/bIgB8

tensorflow_lite

skopiować do folderu My Documents\Arduino\Libraries



https://github.com/tensorflow/tensorflow/tree/be4f6874533d78f662d9777b66abe3cdde98f901/tensorflow/lite/experimental/micro/examples/hello_world

Arduino ML Hello World

podmieniamy model w pliku sine_model_data.cpp na nasz model z collab

sine_model_data.cpp | sine_model_quantized.cc

```
/* Copyright 2019 The TensorFlow Authors. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
==============================================================================*/

// Automatically created from a TensorFlow Lite flatbuffer using the command:
// xxd -i sine_model.tflite > sine_model_data.cc
// See the README for a full description of the creation process.

#include "sine_model_data.h"

// We need to keep the data array aligned on some architectures.
#ifdef __has_attribute
#define HAVE_ATTRIBUTE(x) __has_attribute(x)
#else
#define HAVE_ATTRIBUTE(x) 0
#endif
#if HAVE_ATTRIBUTE(aligned) || (defined(__GNUC__) && !defined(__clang__))
#define DATA_ALIGN_ATTRIBUTE __attribute__((aligned(4)))
#else
#define DATA_ALIGN_ATTRIBUTE
#endif

const unsigned char g_sine_model_data[] DATA_ALIGN_ATTRIBUTE = {
  0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
  0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
  0x88, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00, 0xac, 0x03, 0x00, 0x00,
  0xbc, 0x03, 0x00, 0x00, 0x58, 0x0b, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x02, 0xfc, 0xff, 0xff,
  0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00,
  0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67, 0x5f,
  0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x00, 0x01, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x94, 0xff, 0xff, 0xff, 0x0b, 0x00, 0x00, 0x00,
```

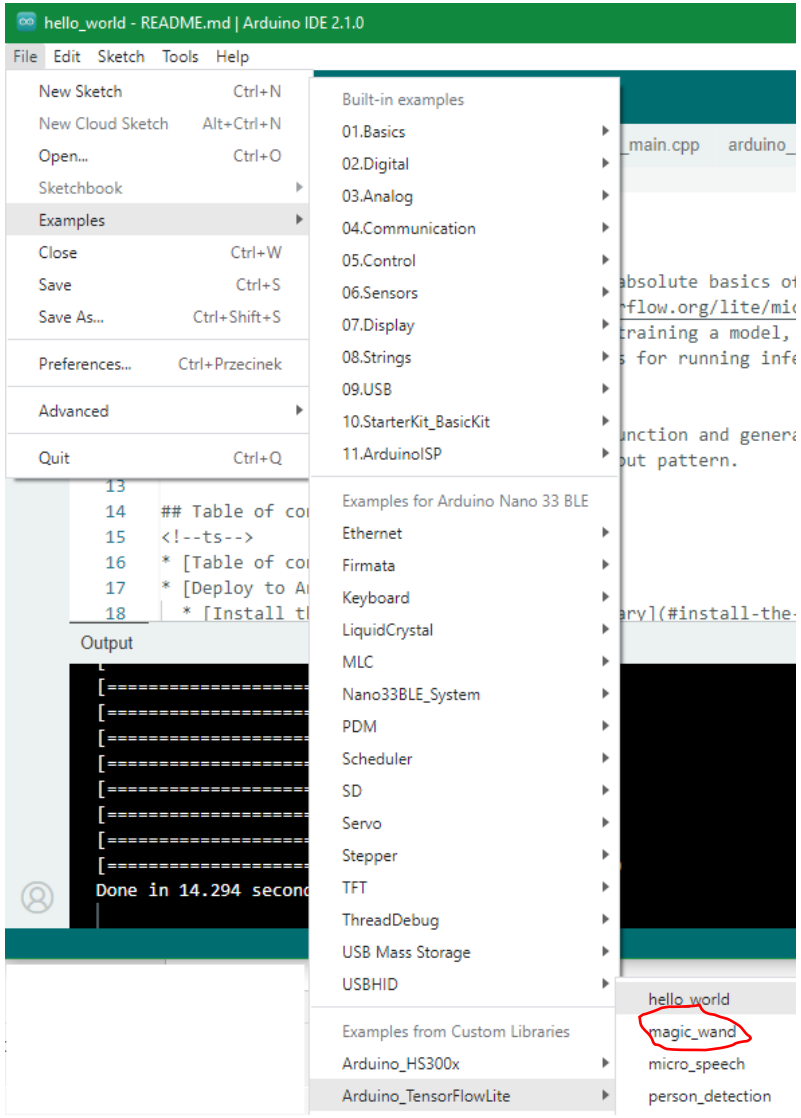
sine_model_quantized.cc

```
unsigned char sine_model_quantized_tflite[] = {
  0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20,
  0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00,
  0x88, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00, 0xac, 0x03, 0x00,
  0xbc, 0x03, 0x00, 0x00, 0x58, 0x0b, 0x00, 0x00, 0x03, 0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x02, 0xfc, 0xff,
  0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00,
```

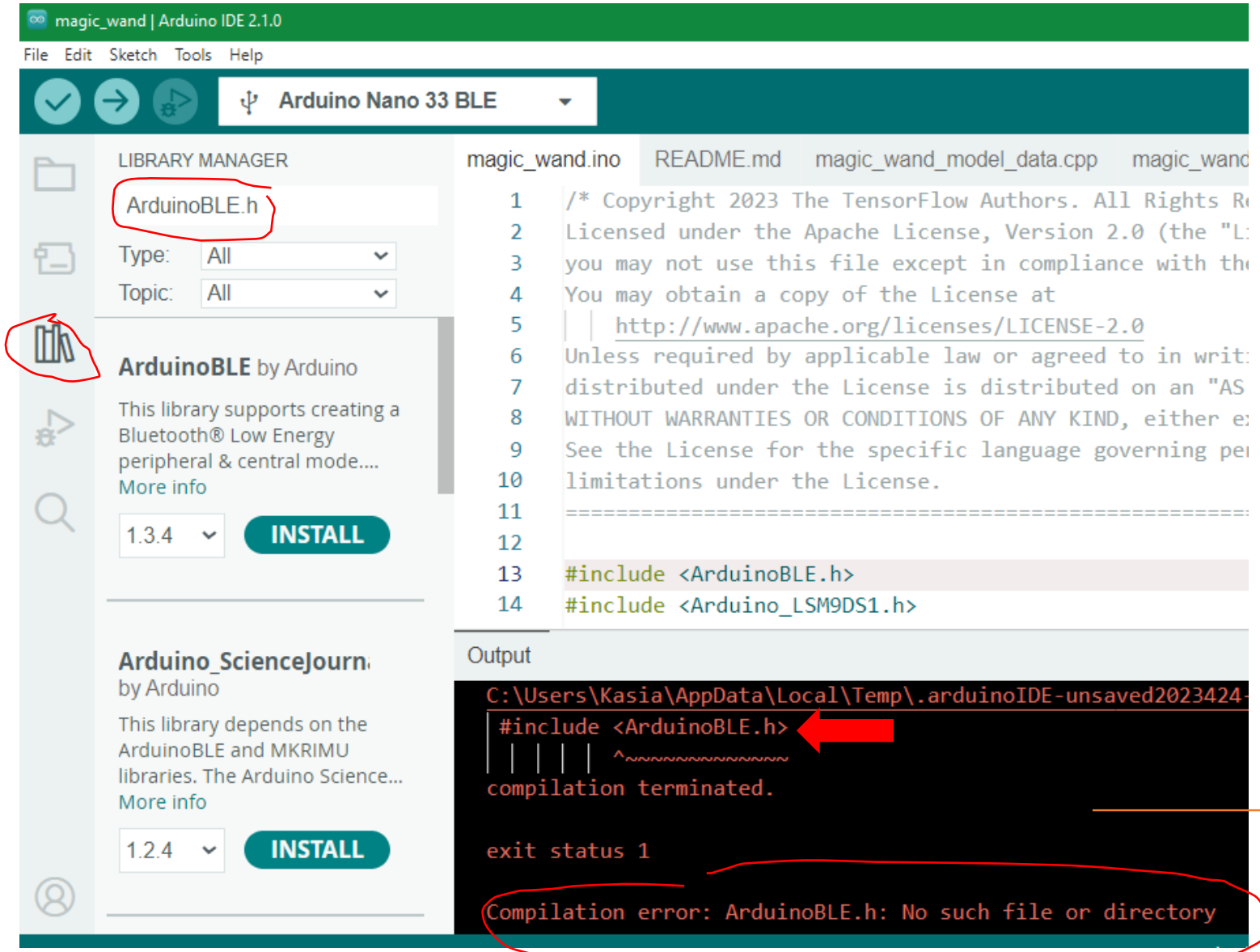
sine_model_data.cpp | sine_model_quantized.cc

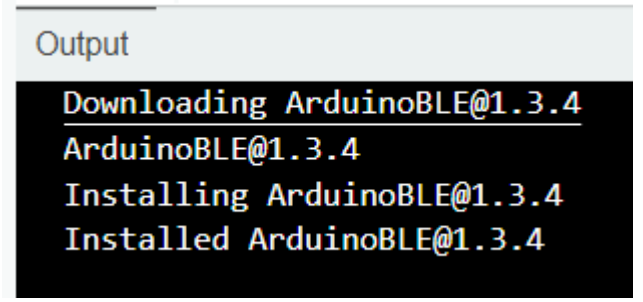
```
unsigned char sine_model_quantized_tflite[] = {
  0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
  0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
  0x88, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00, 0xac, 0x03, 0x00, 0x00,
  0xbc, 0x03, 0x00, 0x00, 0x58, 0x0b, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x02, 0xfc, 0xff, 0xff,
  0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00,
  0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67, 0x5f,
  0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x00, 0x01, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x94, 0xff, 0xff, 0xff, 0x0b, 0x00, 0x00, 0x00,
```

Arduino ML Magic Wand

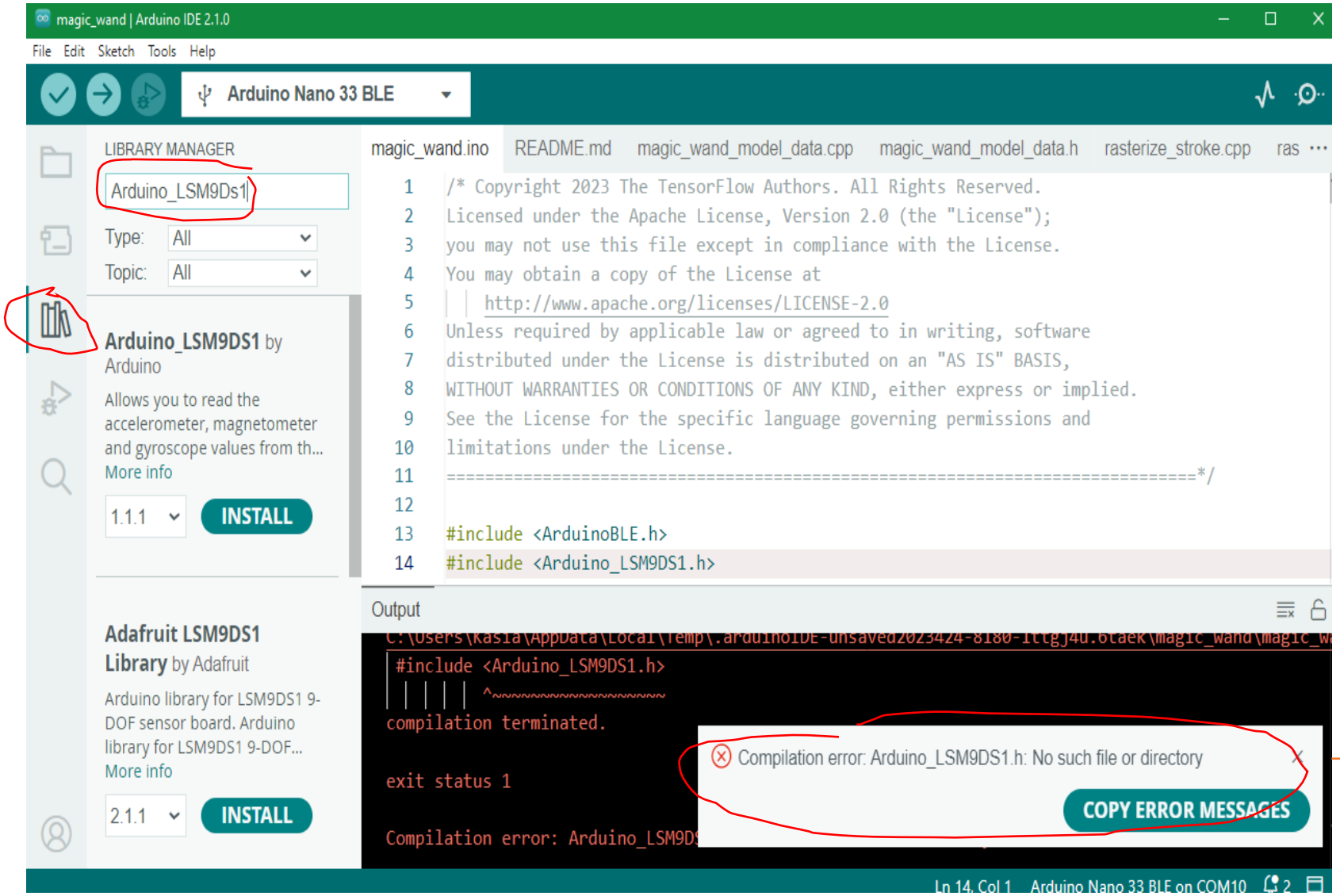


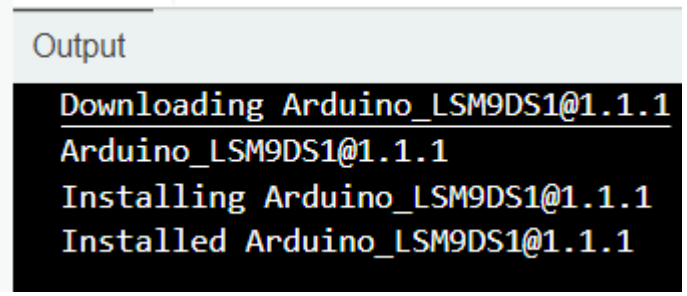
Arduino ML Magic Wand





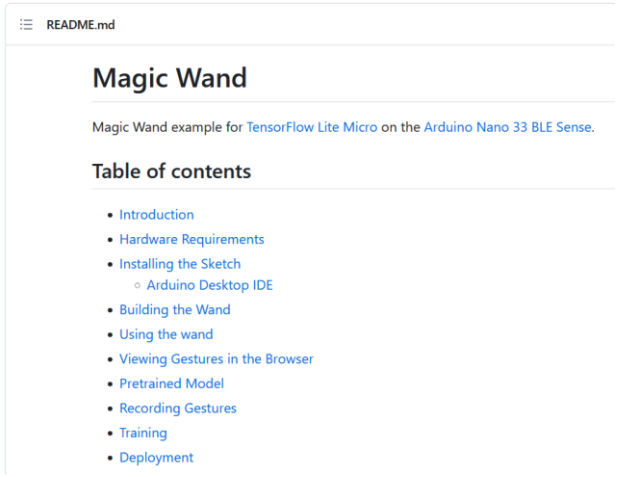
Arduino ML Magic Wand





Arduino ML Magic Wand

instrukcja



README.md

Magic Wand

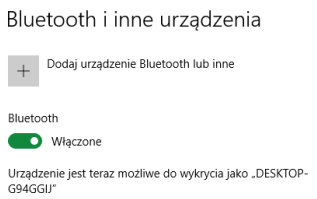
Magic Wand example for TensorFlow Lite Micro on the Arduino Nano 33 BLE Sense.

Table of contents

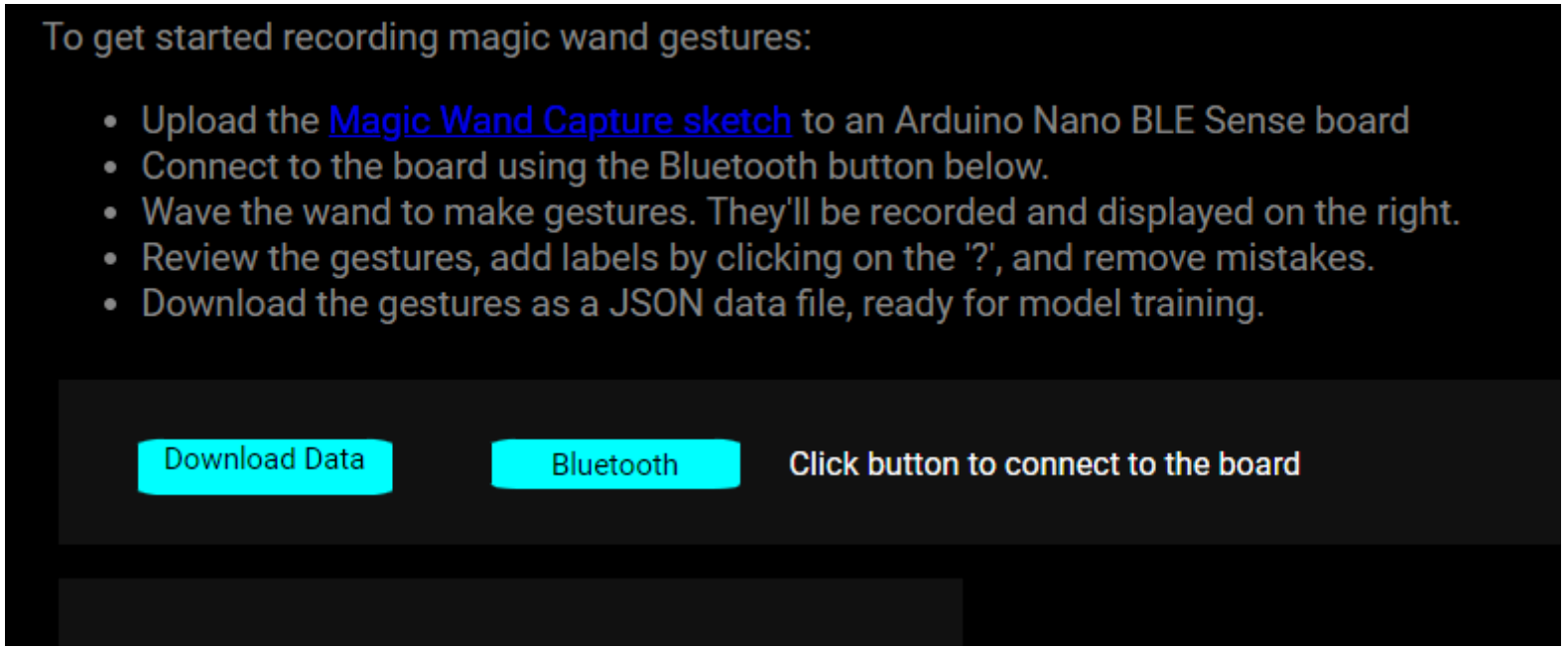
https://github.com/tensorflow/tflite-micro-arduino-examples/tree/main/examples/magic_wand

Arduino ML Magic Wand

strona do podglądu i nagrywania gestów



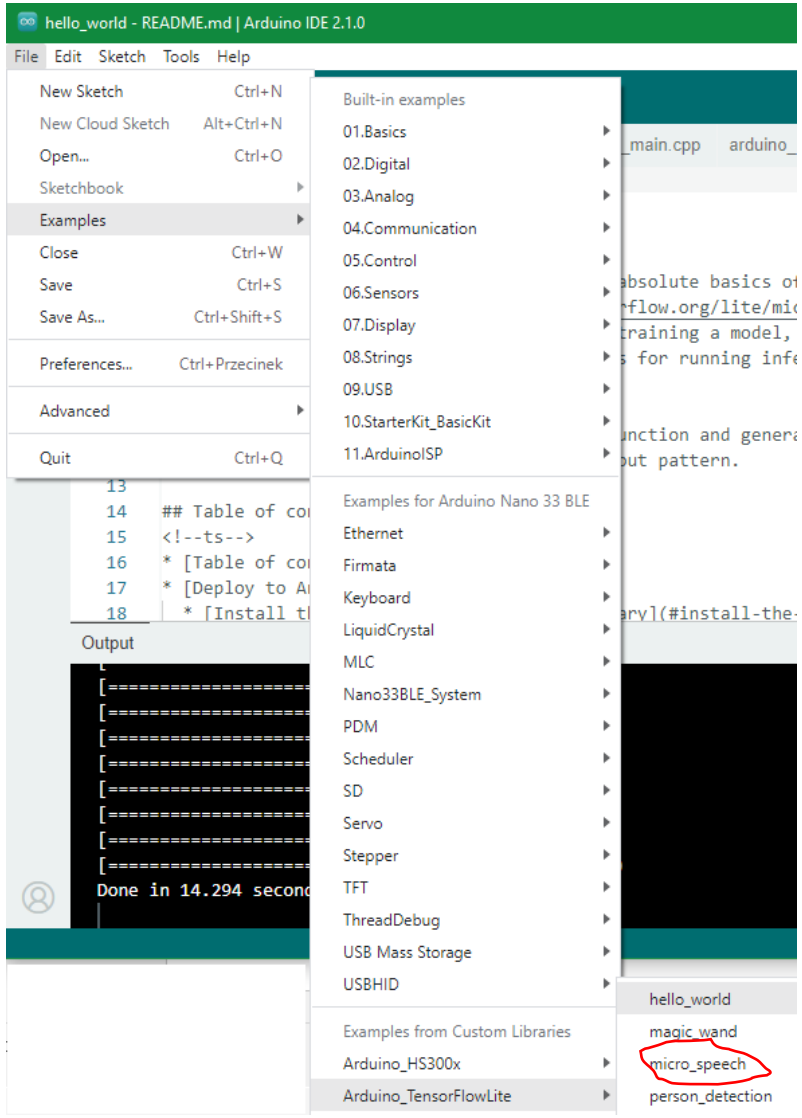
należy włączyć Bluetooh w komputerze



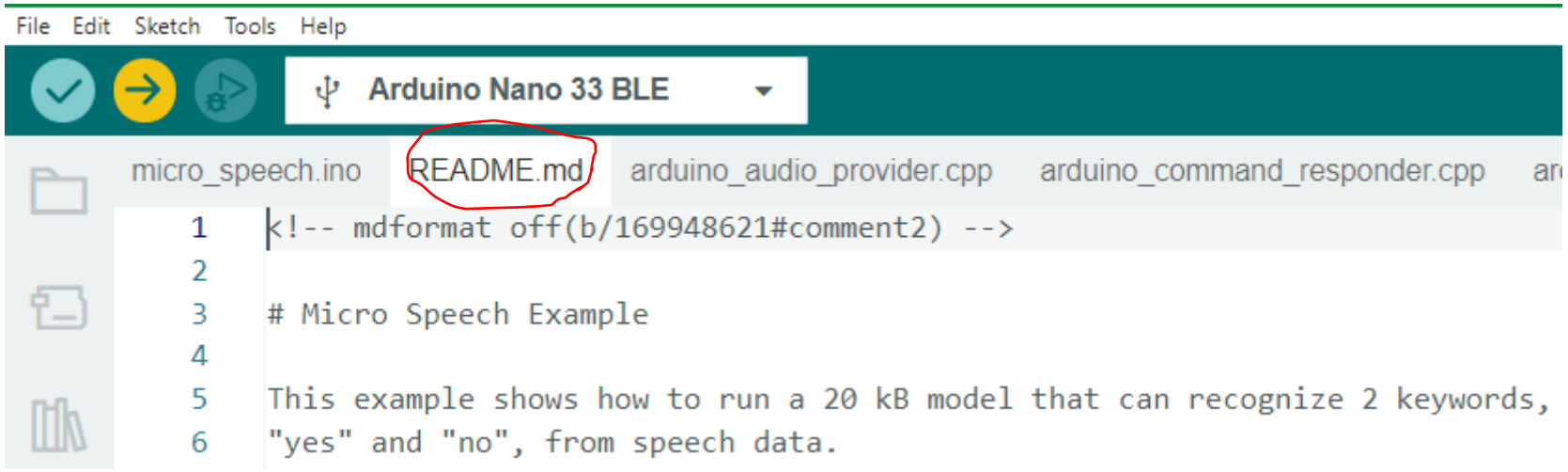
https://github.com/tensorflow/tflite-micro-arduino-examples/tree/main/examples/magic_wand/website

https://github.com/tensorflow/tflite-micro-arduino-examples

Arduino ML Micro Speech



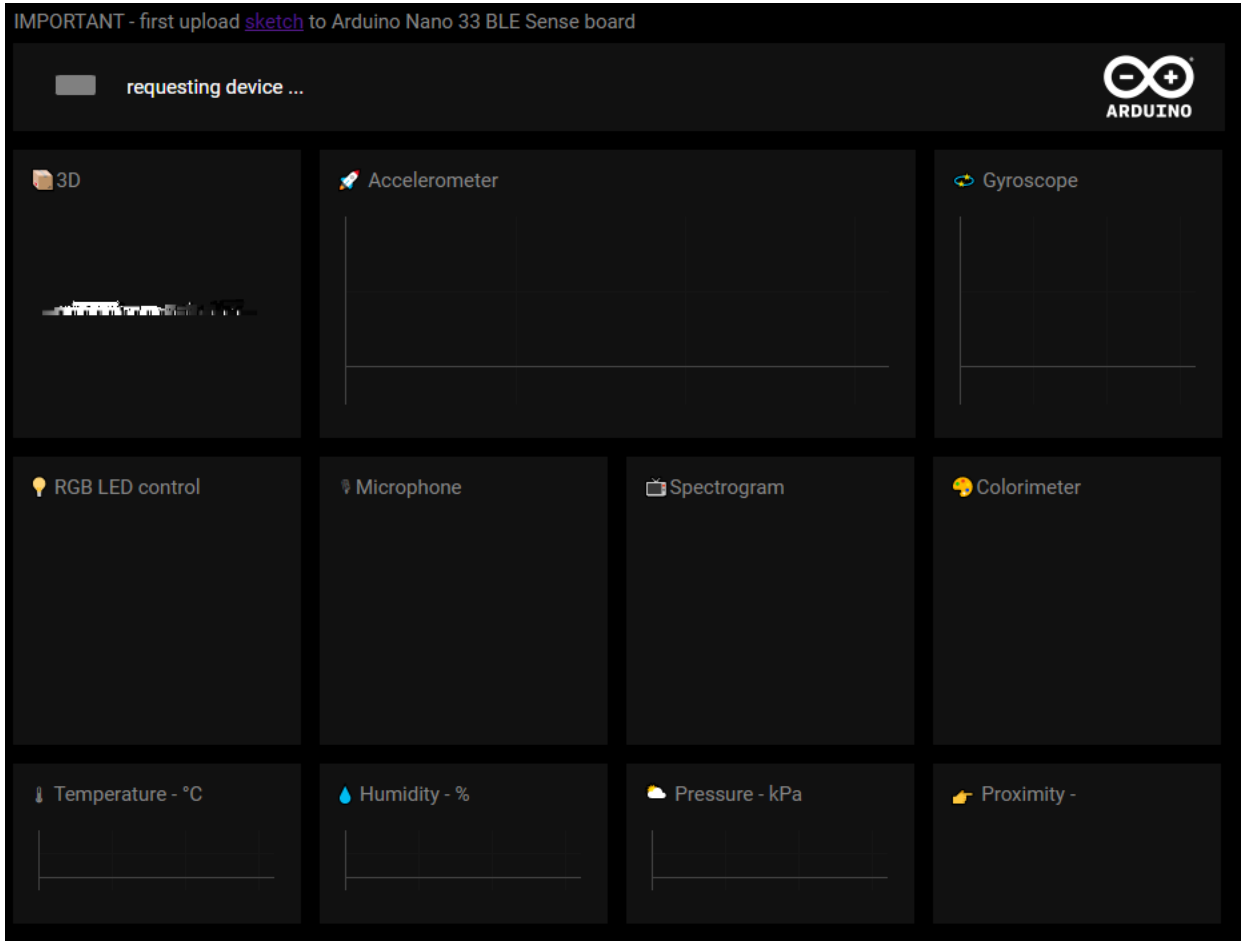
Arduino ML Micro Speech



opis
przykładu

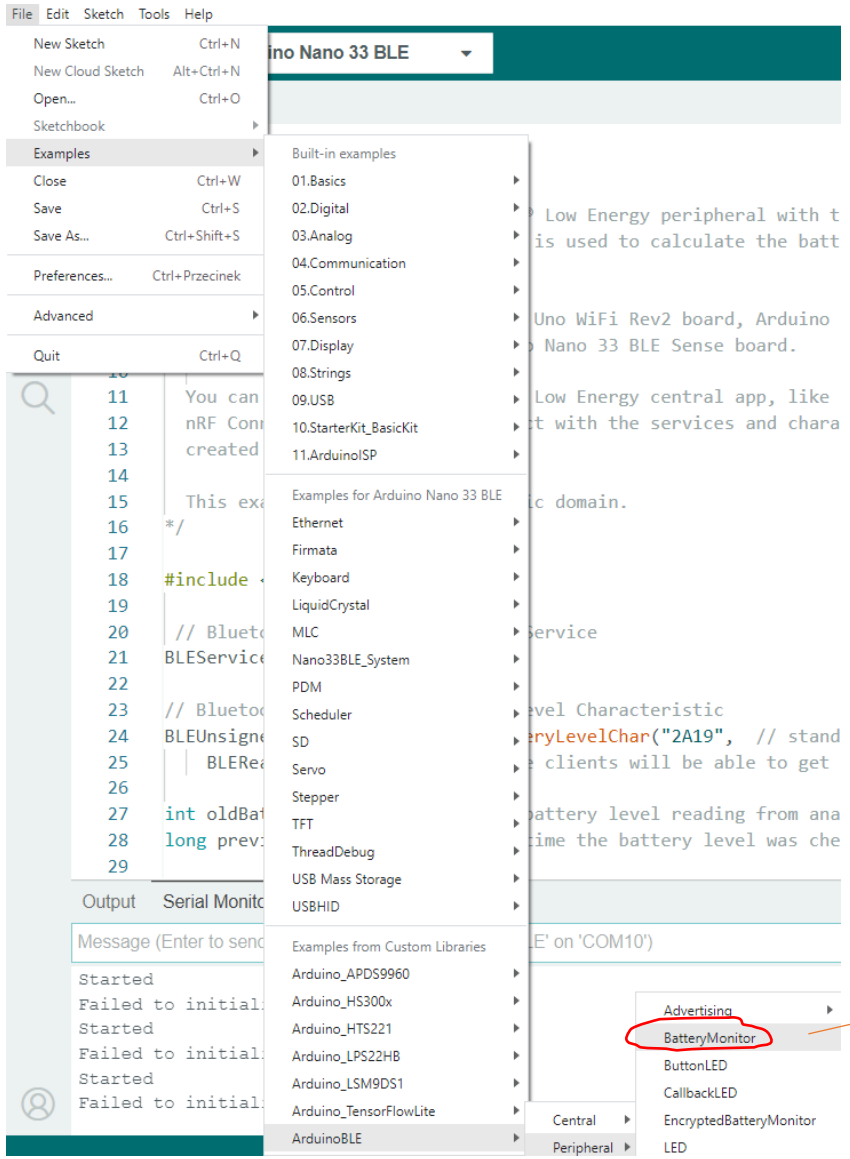
Arduino ML

strona do podglądu czujników

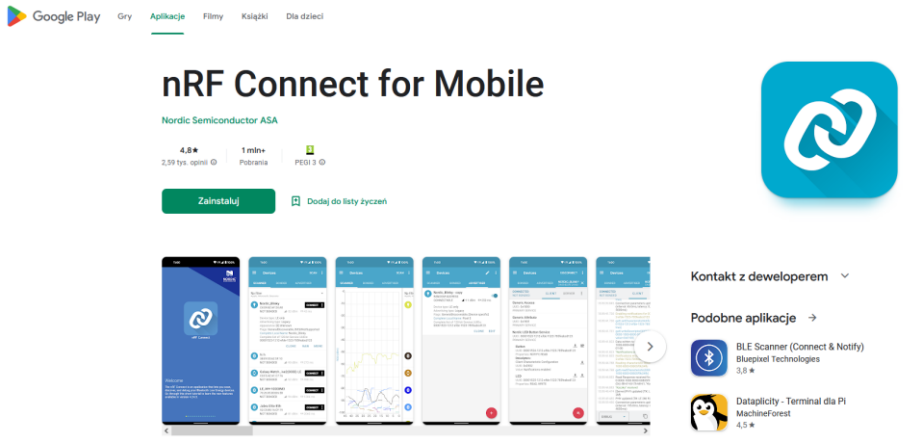


https://arduino.github.io/ArduinoAI/BLESense-test-dashboard/

Arduino ML



test Bluetooth BLE



aplikacja do łączenia telefonu z Androidem z zewnętrznymi urządzeniami poprzez Bluetooth BLE

Blutooth BLE 2 Arduino

https://docs.arduino.cc/tutorials/nano-33-ble-sense/ble-device-to-device