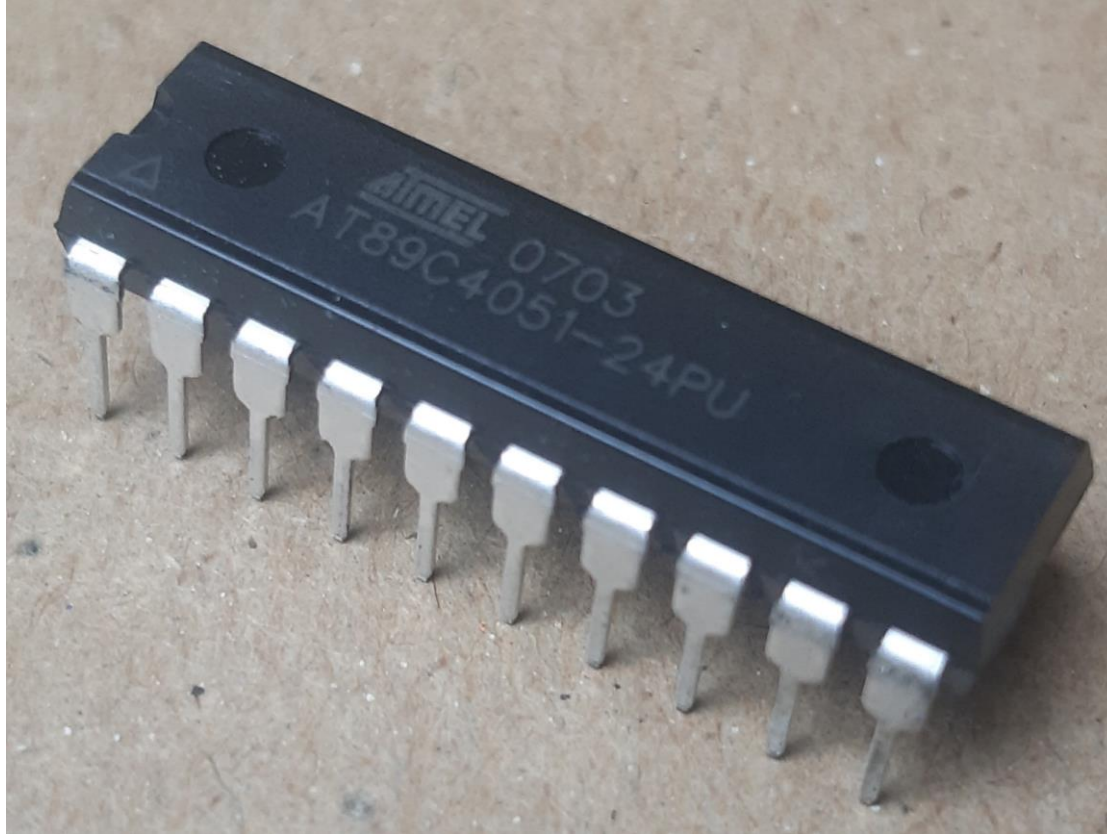


Arduino Nano 33 BLE Sense Rev 2

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.

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.

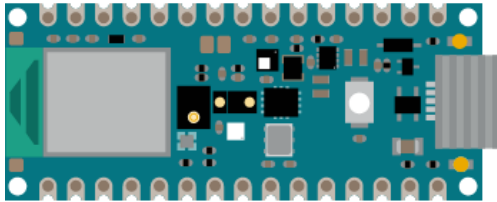
Arduino bada otoczenie, otrzymując dane z wielu czujników i wpływa na nie, sterując światłami, silnikami i innymi urządzeniami elektronicznymi.

Arduino setup

HARDWARE SOFTWARE ▾ CLOUD PROGRAMMING ▾ TUTORIALS LEARN

Overview **Essentials** Tutorials Get Inspired Resources

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 ↓

Setup Guide

×

How do you want to program?

Arduino has developed software tools that work for beginners as well as advanced users. Select your preferred choice and continue with the quickstart.

Arduino IDE 2.0

With features like debugging and code highlighting the new Arduino IDE 2 makes coding faster and easier. A great choice for both beginners and more advanced makers.

Arduino IDE 1.8

The classic offline editor that has been downloaded over 50 million times. You may choose this if you're following a book or other projects based on this older version.

Web Editor

Our web-based editor with cores and libraries already installed. Ideal for beginners and who needs to keep their sketches saved securely in the cloud and synced across devices.

NEXT

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

opisy jak skorzystać z czujników, Bluetooth, mikrofonu itp.

Access Barometric Pressure Sensor Data on Nano 33 BLE Sense

Learn how to read data from the LPS22HB barometric pressure sensor on the Nano 33 BLE Sense board.

Barometric pressure Sensor

Connecting Nano 33 BLE Devices over Bluetooth®

Learn about the history of Bluetooth®, how Bluetooth® Low Energy works and how to connect two Nano BLE devices over Bluetooth®.

Bluetooth® Bluetooth® Low Energy

Controlling RGB LED Through Bluetooth®

Learn how to control the built in RGB LED on the Nano 33 BLE Sense board over Bluetooth®, using an app on your phone.

Bluetooth® Bluetooth® Low Energy

Accessing Accelerometer Data on Nano 33 BLE Sense Rev2

Learn how to measure the relative position of the Nano 33 BLE Sense Rev2 through the BMI270 and BMM150 IMU system.

IMU Accelerometer

Accessing Gyroscope Data on Nano 33 BLE Sense Rev2

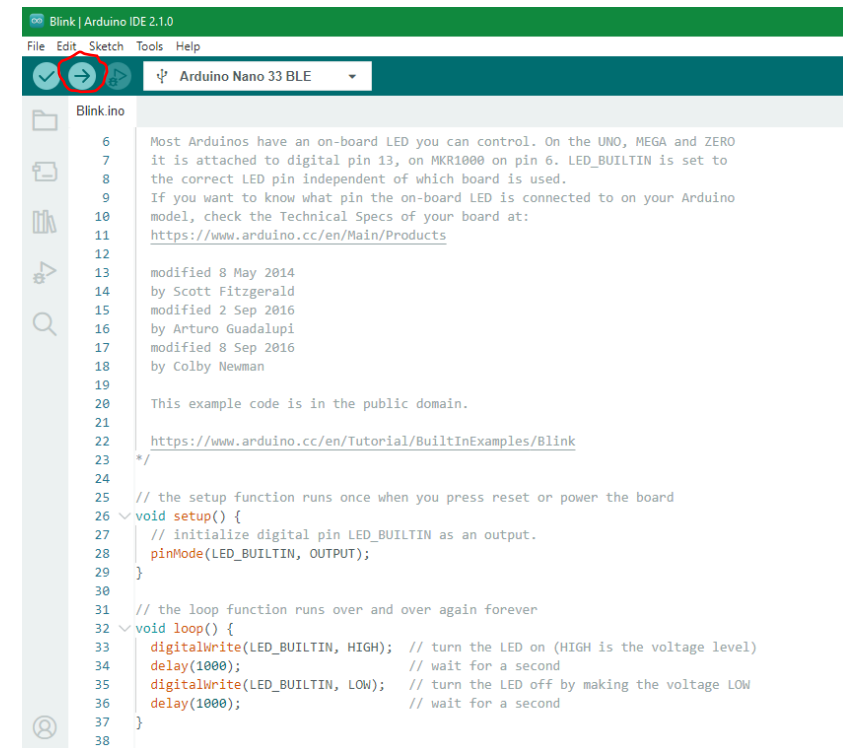
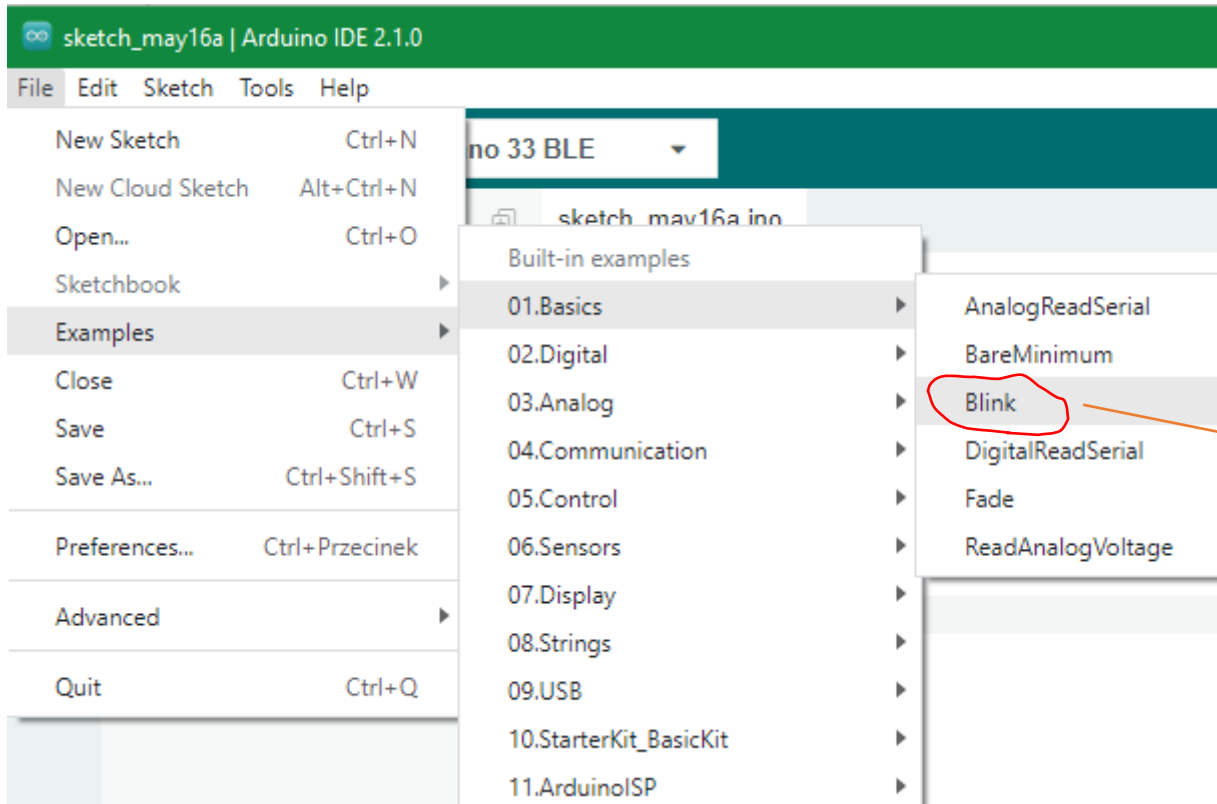
Learn how to measure the direction of force to emulate an object's crash using the Nano 33 BLE Sense Rev2.

IMU Gyroscope

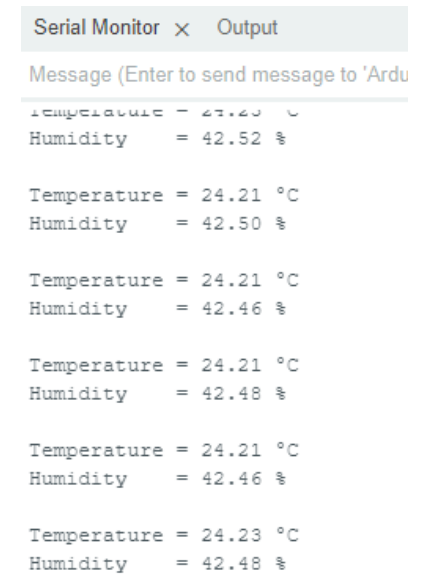
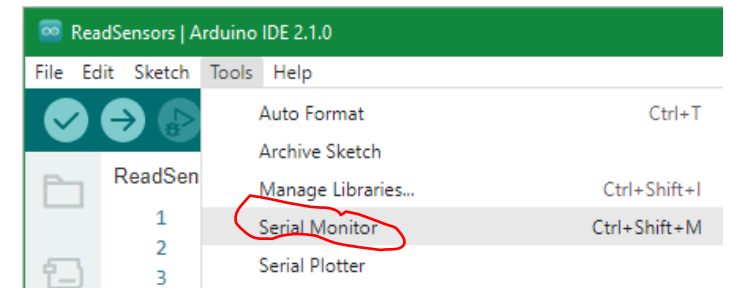
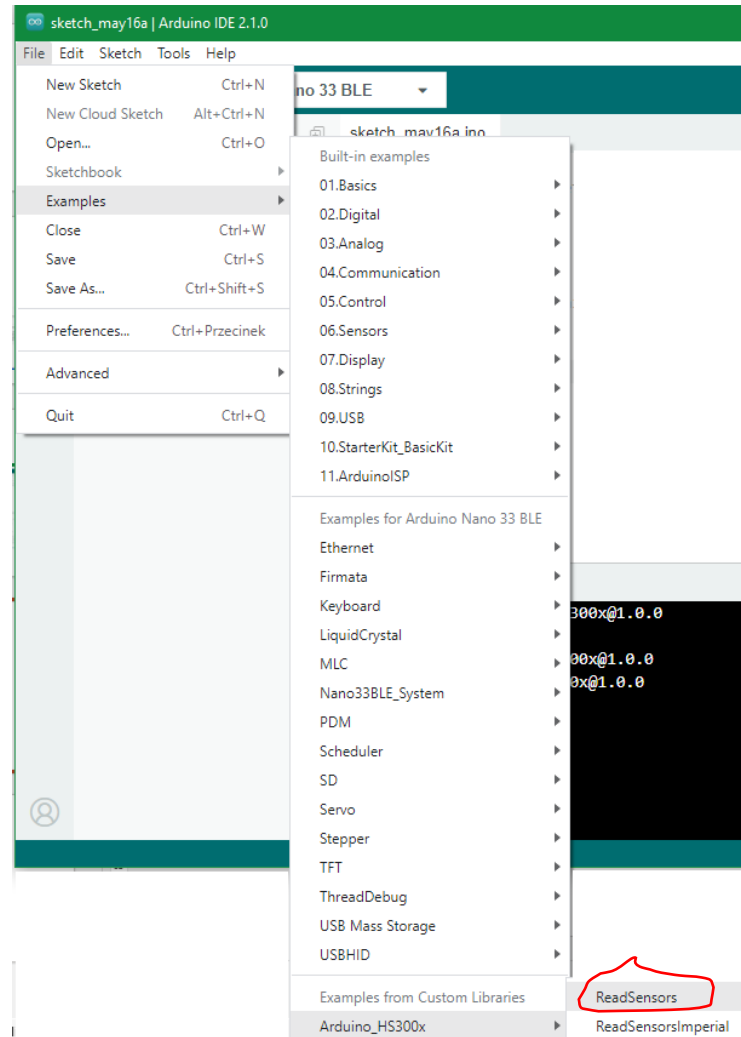
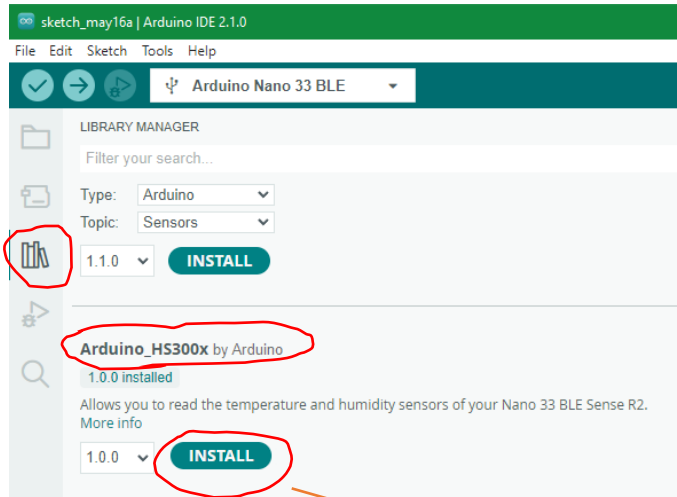
Accessing Magnetometer Data on Nano 33 BLE Sense Rev2

Learn how to detect disturbances in the magnetic field around an electronic device using the Nano 33 BLE Sense Rev2 board.

Arduino Blink example



Arduino temperature, humidity



Arduino examples

[HARDWARE](#) [SOFTWARE](#) [CLOUD](#) [PROGRAMMING](#) [TUTORIALS](#) [LEARN](#)

[Nano 33 BLE Sense Rev2](#) > Controlling the On-Board RGB LED with Microphone

Controlling the On-Board RGB LED with Microphone

Built-in Examples

Learn the basics of Arduino through this collection tutorials. All code examples are available directly in all IDEs.

TensorFlow

Kompleksowa platforma open source do uczenia maszynowego

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

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

ML na małych urządzeniach (mobilnych, wbudowanych, brzegowych)



Tensor Flow Lite

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

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

TensorFlow Lite for Microcontrollers

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 Microcontroller 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

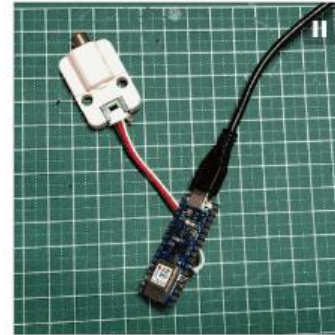
Take control of lighting and video projections with your dance moves.



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

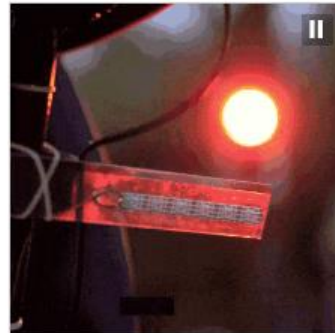
A snore-no-more device embedded in your pillow.



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


Nano 33 BLE Sense > Get Started With Machine Learning on Arduino


Get Started With Machine Learning on Arduino

Learn how to train and use machine learning models with the Arduino Nano 33 BLE Sense

 AUTHOR: **Sandeep Mistry and Dominic Pajak**

 LAST REVISION: **16.05.2023, 11:50**

 This post was originally published by Sandeep Mistry and Dominic Pajak on the [TensorFlow blog](#).

 Important notice! The [TensorFlow Lite Micro Library](#) is no longer available in the Arduino Library Manager. This library will need to be manually downloaded, and included in your IDE.

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

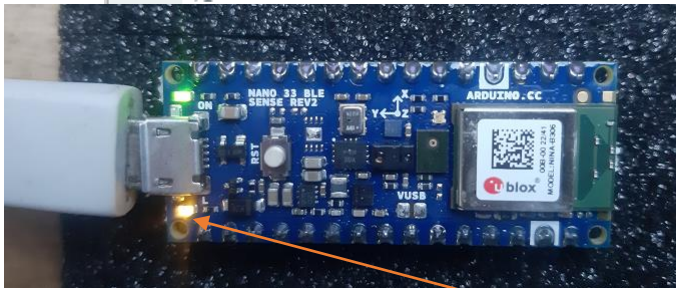
Arduino ML Hello World

```
17
18 #include "constants.h"
19 #include "main_functions.h"
20 #include "model.h"
21 #include "output_handler.h"
22 #include "tensorflow/lite/micro/all_ops_resolver.h"
23 #include "tensorflow/lite/micro/micro_interpreter.h"
24 #include "tensorflow/lite/micro/micro_log.h"
25 #include "tensorflow/lite/micro/system_setup.h"
26 #include "tensorflow/lite/schema/schema_generate.h"
27
28 // Globals, used for compatibility with Arduino-
29 namespace {
30 const tflite::Model* model = nullptr;
31 tflite::MicroInterpreter* interpreter = nullptr;
32 TfliteTensor* input = nullptr;
33 TfliteTensor* output = nullptr;
34 int inference_count = 0;
```

Serial Plotter

Model został wytrenowany do replikowania funkcji sinusoidalnej i generuje wzór danych, który jest wykorzystany w PWM (pulse-width modulation) do sterowania jasnością diody LED

Ln 5, Col 23 Arduino Nano 33 BLE on COM4



jasność diody LED zmienia się sinusoidalnie

Colaboratory



Witamy w Colaboratory

Plik Edytuj Widok Wstaw Środowisko wykonawcze Narzędzia Pomoc

Spis treści

- Wprowadzenie
- Badanie danych
- Systemy uczące się
- Więcej zasobów
 - Przykłady
- Sekcja

+ Kod + Tekst Skopiuj na Dysk

Witamy w Colab

Jeśli już znasz Colab, obejrzyj ten film, aby dowiedzieć się więcej o tabelach interaktywnych, widoku historii wykonanego kodu oraz palecie poleceń.



Czym jest Colab?

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

- brakowi konieczności konfigurowania,
- Bezpłatny dostęp do GPU
- łatwemu udostępnianiu.

Colab może Ci ułatwić pracę niezależnie od tego, czy jesteś **studentem**, **badaczem danych** czy **badaczem sztucznej inteligencji**. Obejrzyj [Wprowadzenie do Colab](#), by dowiedzieć się więcej, lub po prostu zjedź niżej i zacznij korzystać z tej usługi.

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

Arduino ML Hello World

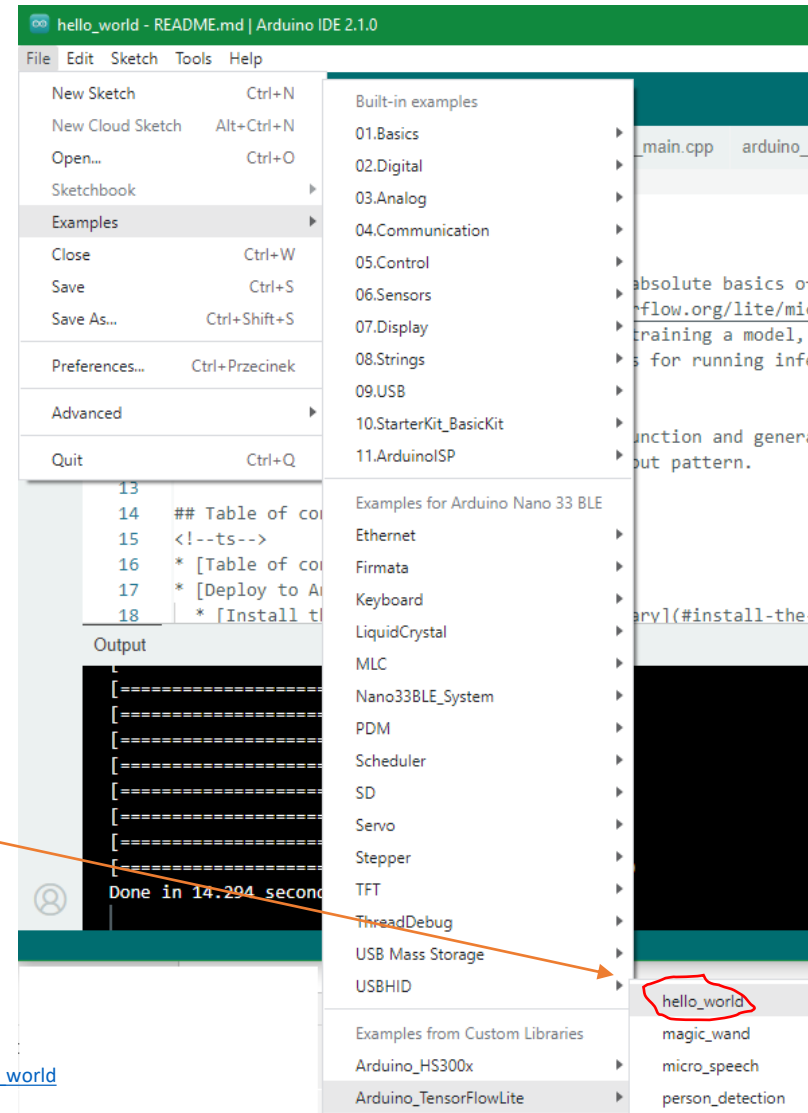
instalacja innej wersji Hello World

<https://oreil.ly/blgB8>

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 X sine_model_quantized.cc X
1 /* Copyright 2019 The TensorFlow Authors. All Rights Reserved.
2
3 Licensed under the Apache License, Version 2.0 (the "License");
4 you may not use this file except in compliance with the License.
5 You may obtain a copy of the License at
6
7 http://www.apache.org/licenses/LICENSE-2.0
8
9 Unless required by applicable law or agreed to in writing, software
10 distributed under the License is distributed on an "AS IS" BASIS,
11 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 See the License for the specific language governing permissions and
13 limitations under the License.
14 -----*/
15
16 // Automatically created from a TensorFlow Lite flatbuffer using the command:
17 // xxd -i sine_model.tflite > sine_model_data.cc
18 // See the README for a full description of the creation process.
19
20 #include "sine_model_data.h"
21
22 // We need to keep the data array aligned on some architectures.
23 #ifndef __has_attribute
24 #define HAVE_ATTRIBUTE(x) __has_attribute(x)
25 #else
26 #define HAVE_ATTRIBUTE(x) 0
27 #endif
28 #if HAVE_ATTRIBUTE(aligned) || (defined(__GNUC__) && !defined(__clang__))
29 #define DATA_ALIGN_ATTRIBUTE __attribute__((aligned(4)))
30 #else
31 #define DATA_ALIGN_ATTRIBUTE
32 #endif
33
34 const unsigned char g_sine_model_data[] DATA_ALIGN_ATTRIBUTE = {
35     0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
36     0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00, 0x00,
37     0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00,
38     0x88, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00, 0xac, 0x03, 0x00, 0x00,
39     0xbc, 0x03, 0x00, 0x00, 0x58, 0x0b, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
40     0x01, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x02, 0xfc, 0xff, 0xff,
41     0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00,
42     0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67, 0x5f,
43     0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x00, 0x01, 0x00, 0x00, 0x00,
44     0x04, 0x00, 0x00, 0x00, 0x94, 0xff, 0xff, 0xff, 0x0b, 0x00, 0x00, 0x00,
```

```
sine_model_quantized.cc X
1 unsigned char sine_model_quantized_tflite[] = {
2     0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20,
3     0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00,
4     0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00,
5     0x88, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00, 0xac, 0x03, 0x00,
6     0xbc, 0x03, 0x00, 0x00, 0x58, 0x0b, 0x00, 0x00, 0x03, 0x00, 0x00,
7     0x01, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x02, 0xfc, 0xff,
8     0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00,
```

```
sine_model_data.cpp X sine_model_quantized.cc X
1 unsigned char sine_model_quantized_tflite[] = {
2     0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
3     0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00, 0x00,
4     0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
5     0x88, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00, 0xac, 0x03, 0x00, 0x00,
6     0xbc, 0x03, 0x00, 0x00, 0x58, 0x0b, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
7     0x01, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x02, 0xfc, 0xff, 0xff,
8     0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00,
9     0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67, 0x5f,
10    0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x00, 0x01, 0x00, 0x00, 0x00,
11    0x04, 0x00, 0x00, 0x00, 0x94, 0xff, 0xff, 0xff, 0x0b, 0x00, 0x00, 0x00,
```


Arduino ML Magic Wand

magic_wand | Arduino IDE 2.1.0

File Edit Sketch Tools Help

Arduino Nano 33 BLE

LIBRARY MANAGER

ArduinoBLE.h

Type: All

Topic: All

ArduinoBLE by Arduino

This library supports creating a Bluetooth® Low Energy peripheral & central mode....

More info

1.3.4

INSTALL

Arduino_ScienceJourn. by Arduino

This library depends on the ArduinoBLE and MKRIMU libraries. The Arduino Science...

More info

1.2.4

INSTALL

magic_wand.ino README.md magic_wand_model_data.cpp magic_wand

```
1 /* Copyright 2023 The TensorFlow Authors. All Rights Reserved.
2 Licensed under the Apache License, Version 2.0 (the "License");
3 you may not use this file except in compliance with the License.
4 You may obtain a copy of the License at
5 | | http://www.apache.org/licenses/LICENSE-2.0
6 Unless required by applicable law or agreed to in writing, software
7 distributed under the License is distributed on an "AS IS" BASIS,
8 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
9 See the License for the specific language governing permissions and
10 limitations under the License.
11 =====
12
13 #include <ArduinoBLE.h>
14 #include <Arduino_LSM9DS1.h>
```

Output

```
C:\Users\Kasia\AppData\Local\Temp\.arduinoIDE-unsaved2023424-...
#include <ArduinoBLE.h>
| | | | ^
compilation terminated.

exit status 1

Compilation error: ArduinoBLE.h: No such file or directory
```

Output

```
Downloading ArduinoBLE@1.3.4
Installing ArduinoBLE@1.3.4
Installed ArduinoBLE@1.3.4
```

Arduino ML Magic Wand

The screenshot shows the Arduino IDE 2.1.0 interface. The top bar indicates the board is set to "Arduino Nano 33 BLE". The Library Manager on the left shows the search results for "Arduino_LSM9DS1". The code editor in the center contains a license header for TensorFlow and includes the following code:

```
1 /* Copyright 2023 The TensorFlow Authors. All Rights Reserved.
2 Licensed under the Apache License, Version 2.0 (the "License");
3 you may not use this file except in compliance with the License.
4 You may obtain a copy of the License at
5 | | http://www.apache.org/licenses/LICENSE-2.0
6 Unless required by applicable law or agreed to in writing, software
7 distributed under the License is distributed on an "AS IS" BASIS,
8 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
9 See the License for the specific language governing permissions and
10 limitations under the License.
11 =====*/
12
13 #include <ArduinoBLE.h>
14 #include <Arduino_LSM9DS1.h>
```

The Output window at the bottom shows the following error message:

```
Output
C:\Users\Kasia\AppData\Local\Temp\arduinoide-unsaved2023424-8180-11tgj4u.6taek\magic_wand\magic_w
#include <Arduino_LSM9DS1.h>
| | | | ^~~~~~
compilation terminated.

exit status 1

Compilation error: Arduino_LSM9DS1.h: No such file or directory

COPY ERROR MESSAGES
```

Output

```
Downloading Arduino_LSM9DS1@1.1.1
Arduino_LSM9DS1@1.1.1
Installing Arduino_LSM9DS1@1.1.1
Installed Arduino_LSM9DS1@1.1.1
```

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

- Introduction
- Hardware Requirements
- Installing the Sketch
 - Arduino Desktop IDE
- Building the Wand
- Using the wand
- Viewing Gestures in the Browser
- Pretrained Model
- Recording Gestures
- Training
- Deployment

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

Arduino ML Magic Wand

strona do podglądu i nagrywania gestów

Bluetooth i inne urządzenia

+ Dodaj urządzenie Bluetooth lub inne

Bluetooth

Włączone

Urządzenie jest teraz możliwe do wykrycia jako „DESKTOP-G94GGU”

To get started recording magic wand gestures:

- Upload the [Magic Wand Capture sketch](#) to an Arduino Nano BLE Sense board
- Connect to the board using the Bluetooth button below.
- Wave the wand to make gestures. They'll be recorded and displayed on the right.
- Review the gestures, add labels by clicking on the '?', and remove mistakes.
- Download the gestures as a JSON data file, ready for model training.

Download Data

Bluetooth

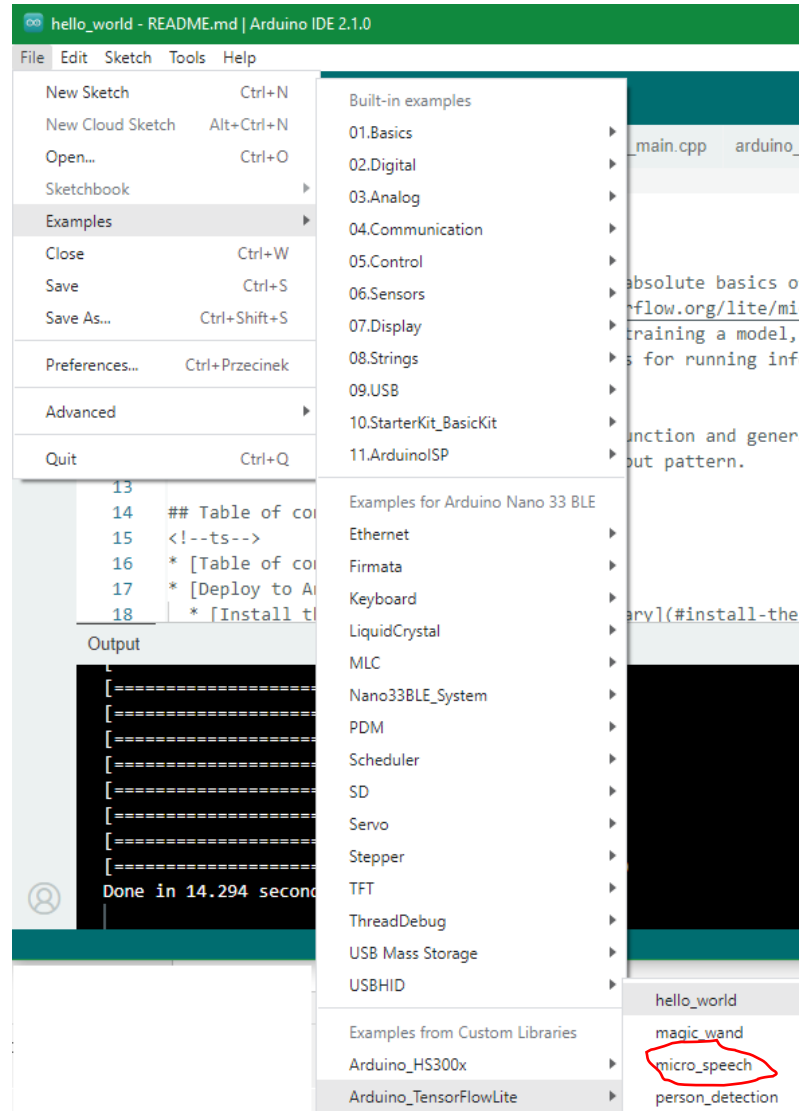
Click button to connect to the board

należy
włączyć
Bluetooth w
komputerze

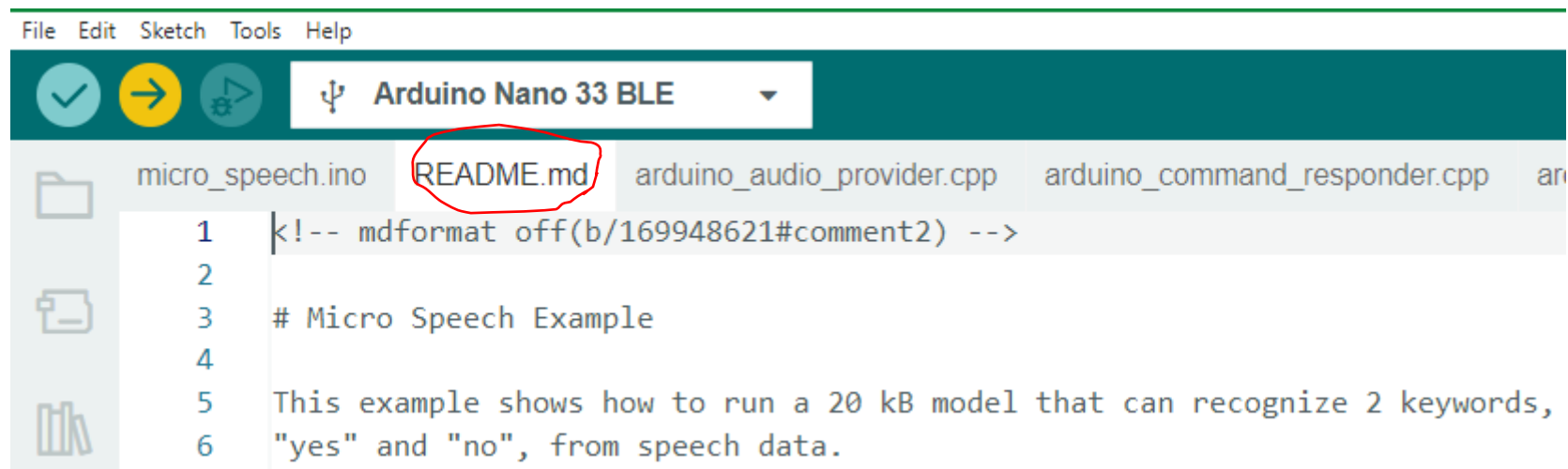
https://github.com/tensorflow/tflite-micro-arduino-examples/tree/main/examples/magic_wand/website

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

Arduino ML Micro Speech



Arduino ML Micro Speech

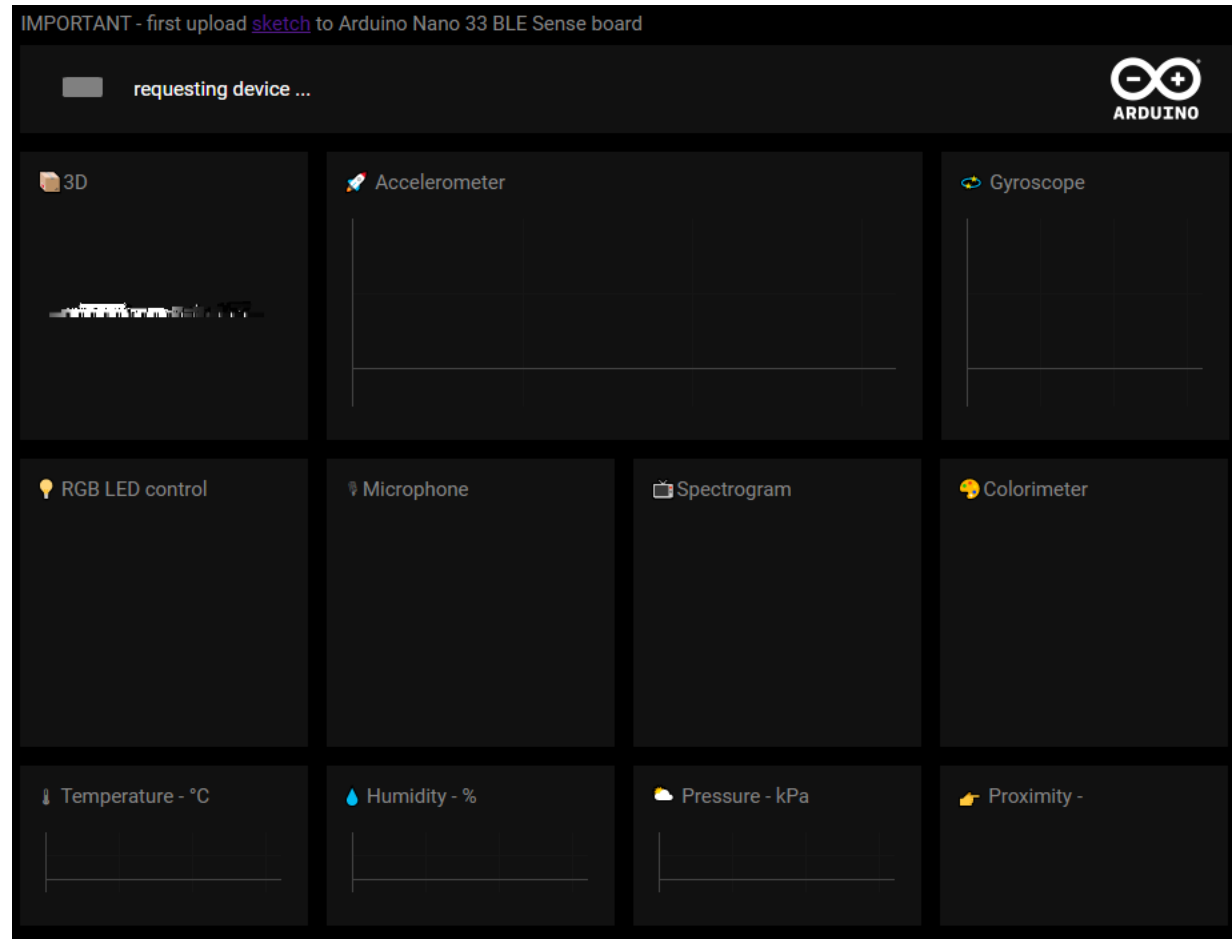


```
File Edit Sketch Tools Help
Arduino Nano 33 BLE
micro_speech.ino README.md arduino_audio_provider.cpp arduino_command_responder.cpp ar
1 |<!-- mdformat off(b/169948621#comment2) -->
2
3 # Micro Speech Example
4
5 This example shows how to run a 20 kB model that can recognize 2 keywords,
6 "yes" and "no", from speech data.
```

opis
przykładu

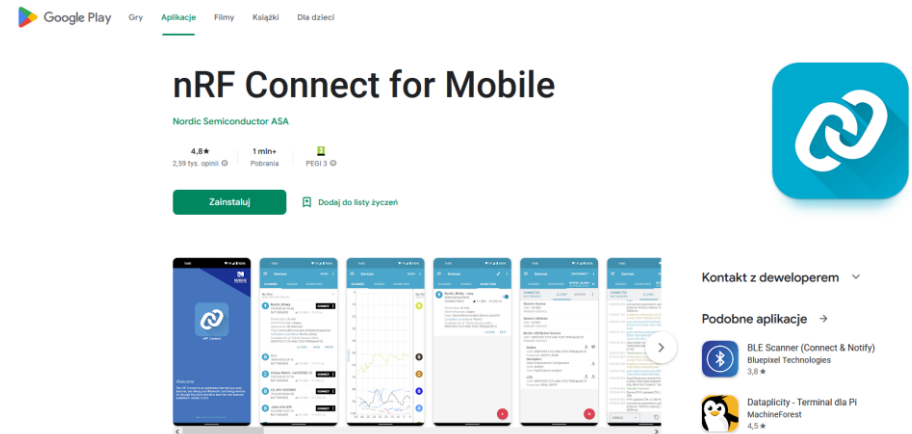
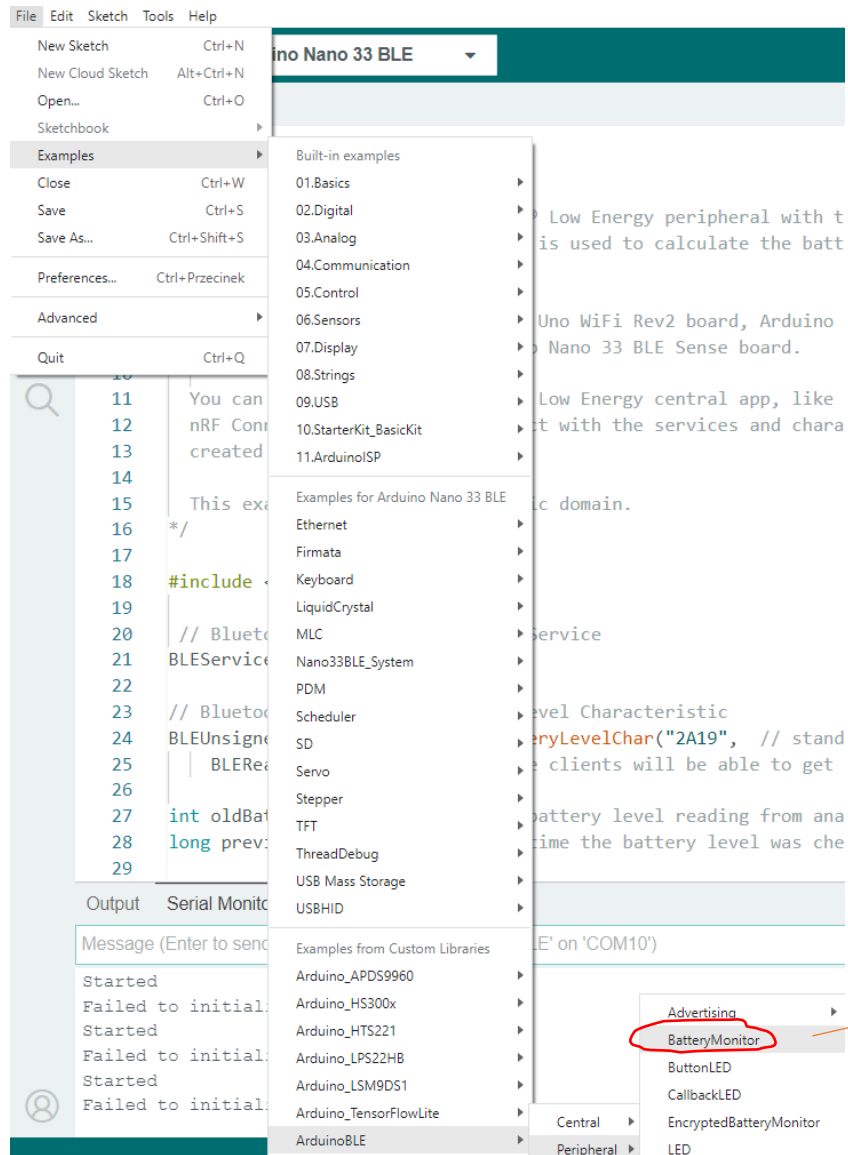
Arduino ML

strona do podglądu czujników



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

Arduino ML



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

test Bluetooth BLE

Bluetooth BLE 2 Arduino

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