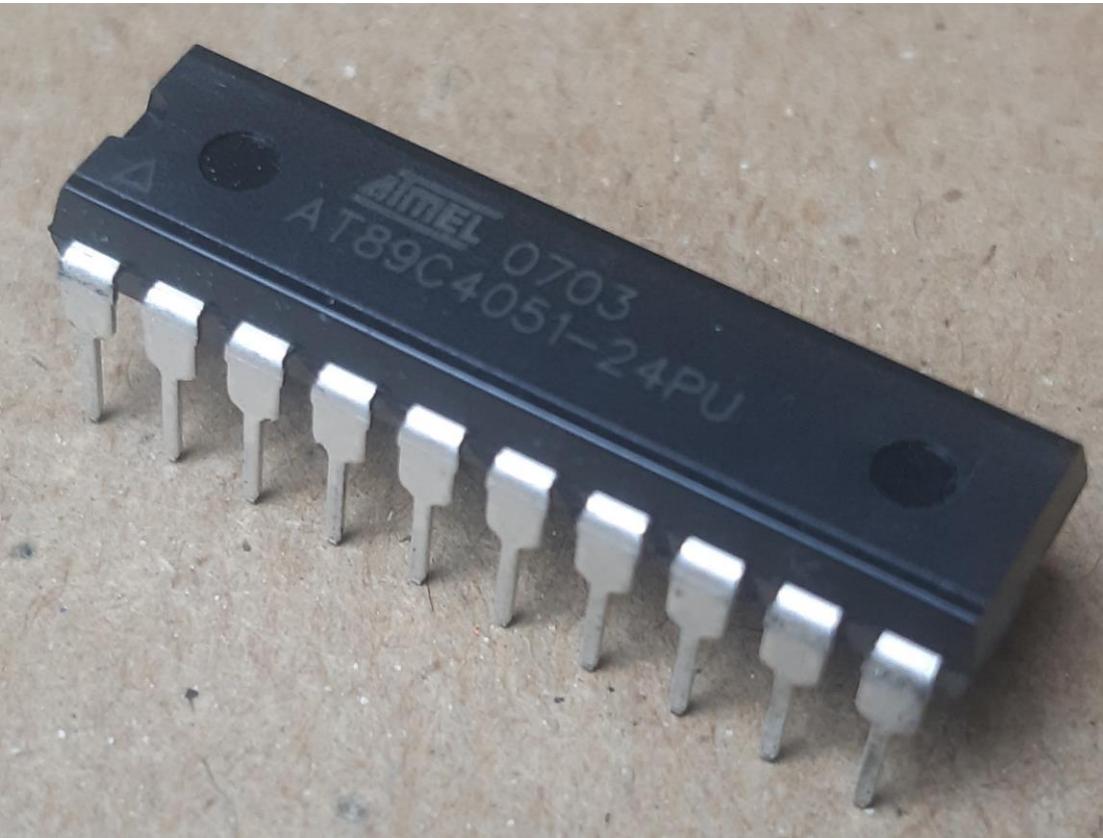


Arduino Nano 33 BLE Sense Rev 2

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

Jest przeznaczona dla każdego, kto tworzy interaktywne projekty.



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

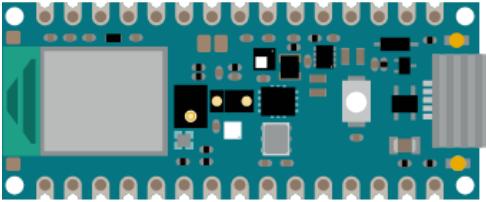
Arduino bada otoczenie, otrzymując dane z wielu czujników i wpływa na nie, sterując światłami, silnikami i innymi urządzeniami elektronicznymi.

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>

Arduino tutorials

opisy jak skorzystać z czujników, Bluetooth, mikrofonu itp.



Arduino Nano 33 BLE Sense Rev2

HARDWARE SOFTWARE ▾ CLOUD PROGRAMMING ▾ TUTORIALS LEARN

Overview Essentials Tutorials Get Inspired Resources

 [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

 [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

 [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

 [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

 [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 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

The image shows two screenshots of the Arduino IDE. The left screenshot displays the main menu with the 'Examples' option selected. A dropdown menu lists various built-in examples, including 'Blink'. The 'Blink' option is highlighted with a red circle and has an orange arrow pointing to the right screenshot. The right screenshot shows the code for the 'Blink' sketch. The code is as follows:

```
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:
<https://www.arduino.cc/en/Main/Products>

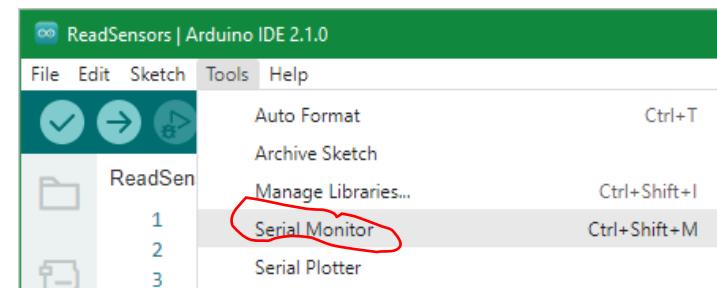
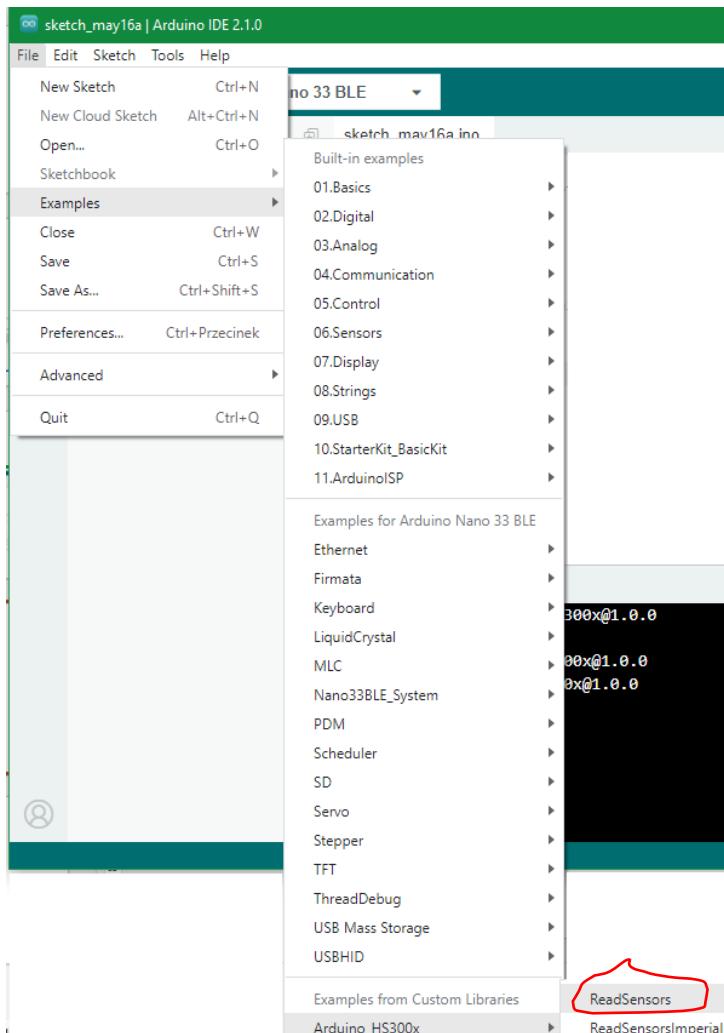
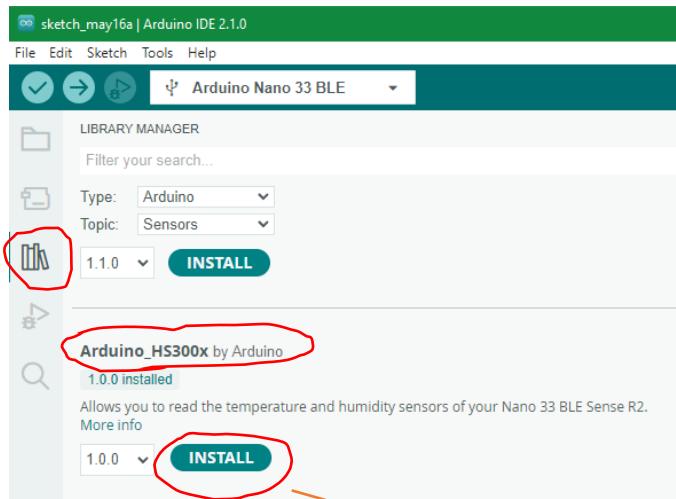
This example code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink>

```
/*
 * the setup function runs once when you press reset or power the board
 */
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

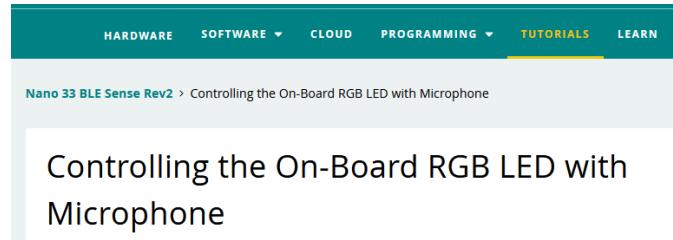
Arduino temperature, humidity



The screenshot shows the "Serial Monitor" window. It displays a series of messages with temperature and humidity values. The text is as follows:

```
Temperature = 27.23 °C  
Humidity = 42.52 %  
  
Temperature = 24.21 °C  
Humidity = 42.50 %  
  
Temperature = 24.21 °C  
Humidity = 42.46 %  
  
Temperature = 24.21 °C  
Humidity = 42.48 %  
  
Temperature = 24.21 °C  
Humidity = 42.46 %  
  
Temperature = 24.23 °C  
Humidity = 42.48 %
```

Arduino examples



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 zaledwie kilkoma kilobajtami pamięci.

TensorFlow Lite for Microcontrollers

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

ML na małych urządzeniach (mobilnych, wbudowanych, brzegowych)



TensorFlow

Tensor Flow Lite

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

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

<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, Jueum Choi, Yeonhee Kim, Jonghyun Baek, Yongjae Kim

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



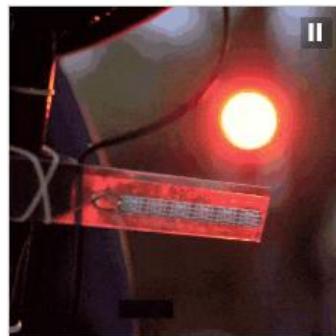
SNORING GUARDIAN
by Naveen Kumar

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



SQUATS COUNTER
by Manas Rane

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



VOICE TURN
by Álvaro González-Vila

A safer way for cyclists to signal using their voice.

przykłady użycia biblioteki TensorFlowLite

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

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>

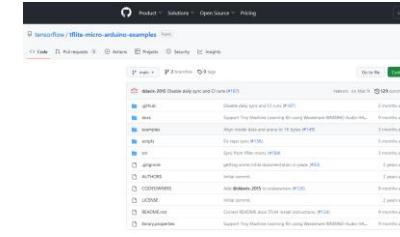
Arduino ML

Opis instalacji biblioteki Arduino_TensorFlow:

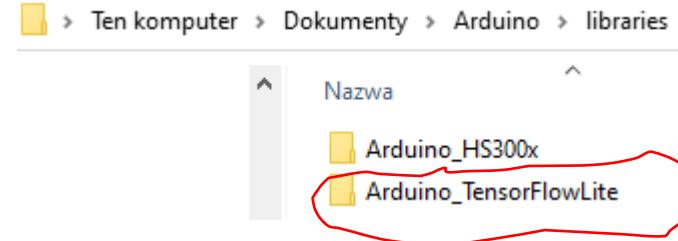
1. zainstalować Gita
2. przejść do folderu My Documents\Arduino\Libraries
3. uruchomić konsolę Git Bash Here w tym folderze
4. wpisać komendę:

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

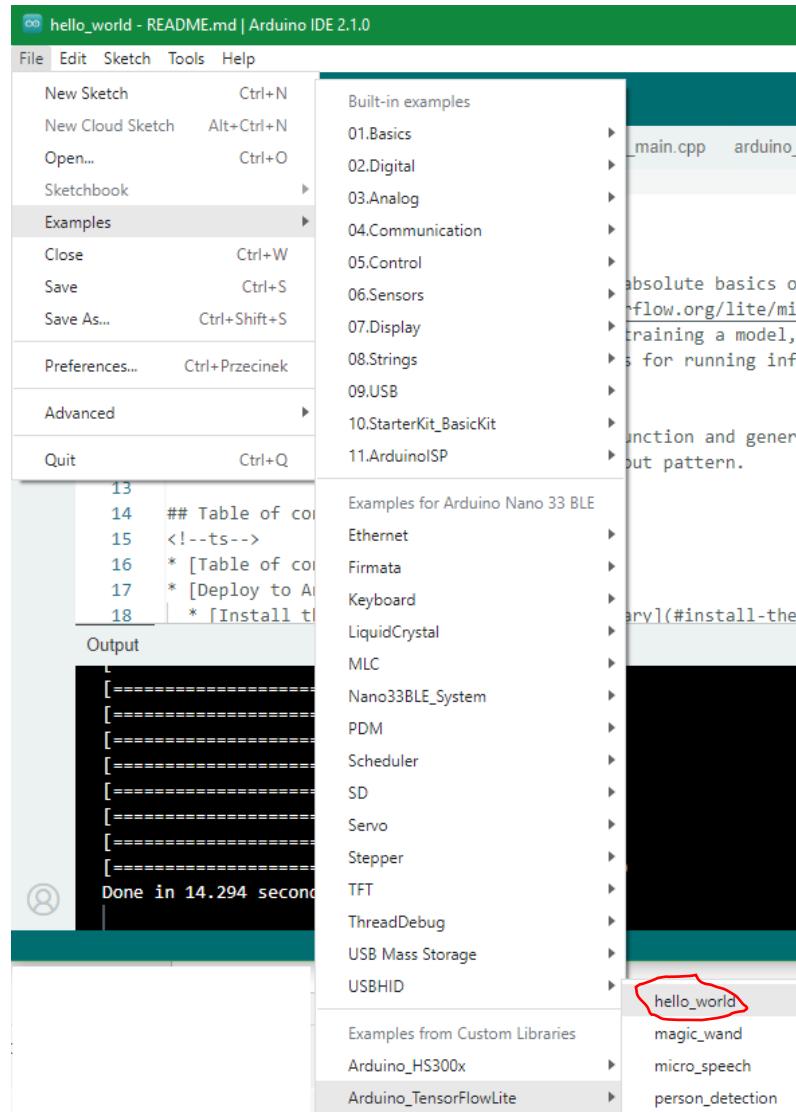
5. pojawi się nowy folder z biblioteką Arduino_TensorFlowLite



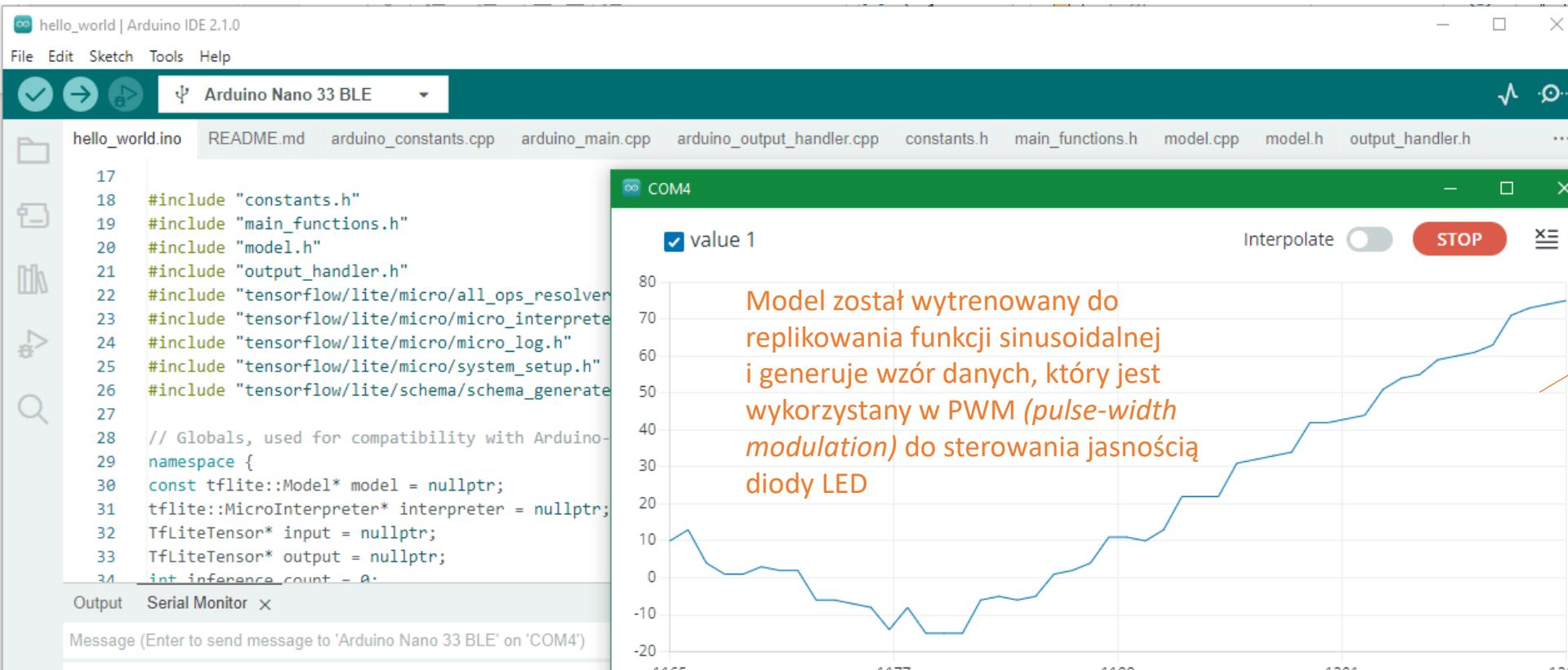
<https://github.com/tensorflow/tflite-micro-arduino-examples>



Arduino ML Hello World



Arduino ML Hello World



The screenshot shows the Arduino IDE interface. The left pane displays the code for 'hello_world.ino'. The right pane shows the 'Serial Monitor' window titled 'COM4' with the message 'value 1' and a plot of data over time. The plot shows a sinusoidal wave, which corresponds to the PWM signal controlling the LED brightness. The Arduino board is also shown at the bottom left, with an orange arrow pointing from its LED to the plot.

```
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 namespace {
29 const tflite::Model* model = nullptr;
30 tflite::MicroInterpreter* interpreter = nullptr;
31 TfLiteTensor* input = nullptr;
32 TfLiteTensor* output = nullptr;
33 int inference_count = 0;
34 }
```

Message (Enter to send message to 'Arduino Nano 33 BLE' on 'COM4')

71

Serial Plotter

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

Interpolate STOP

Type Message SEND New Line 9600 baud

1165 1177 1189 1201 1214

80
70
60
50
40
30
20
10
0
-10
-20

NANO 33 BLE REV2 ARDUINO.cc u-blox VUSB

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

- braku konieczności konfigurowania,
- Bezpośredni 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

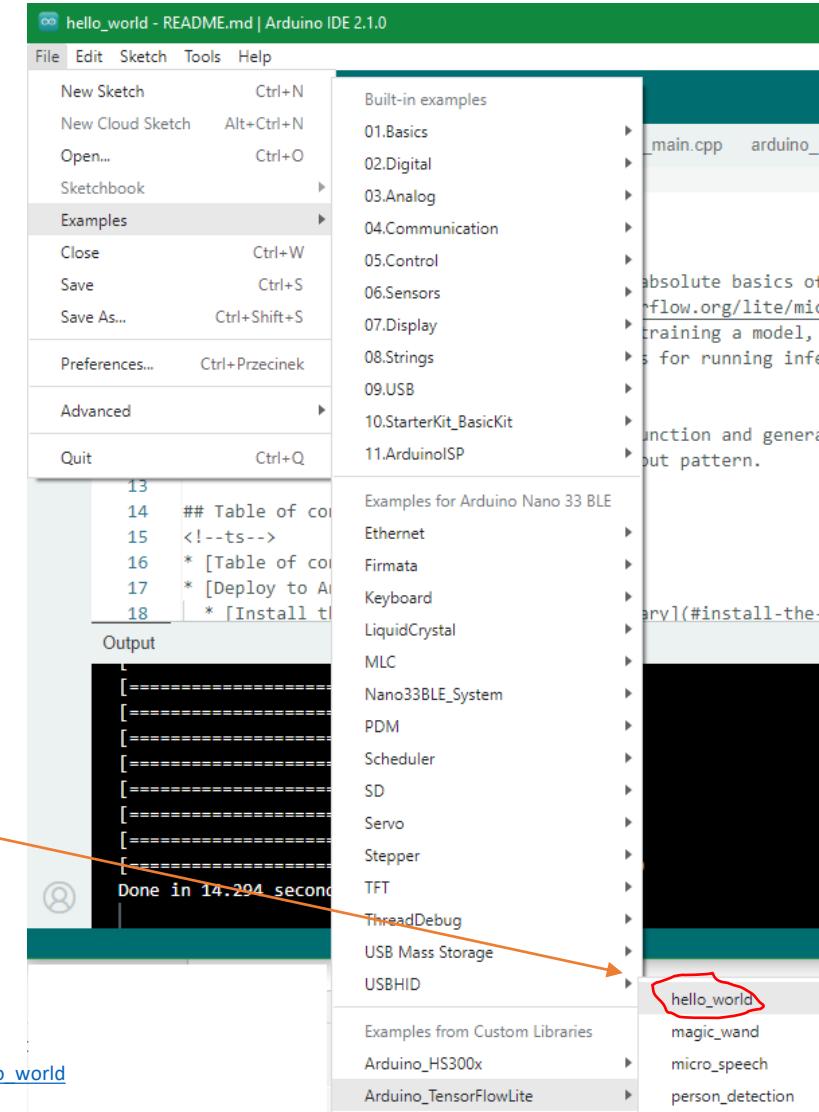
instalacja innej wersji Hello World

<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

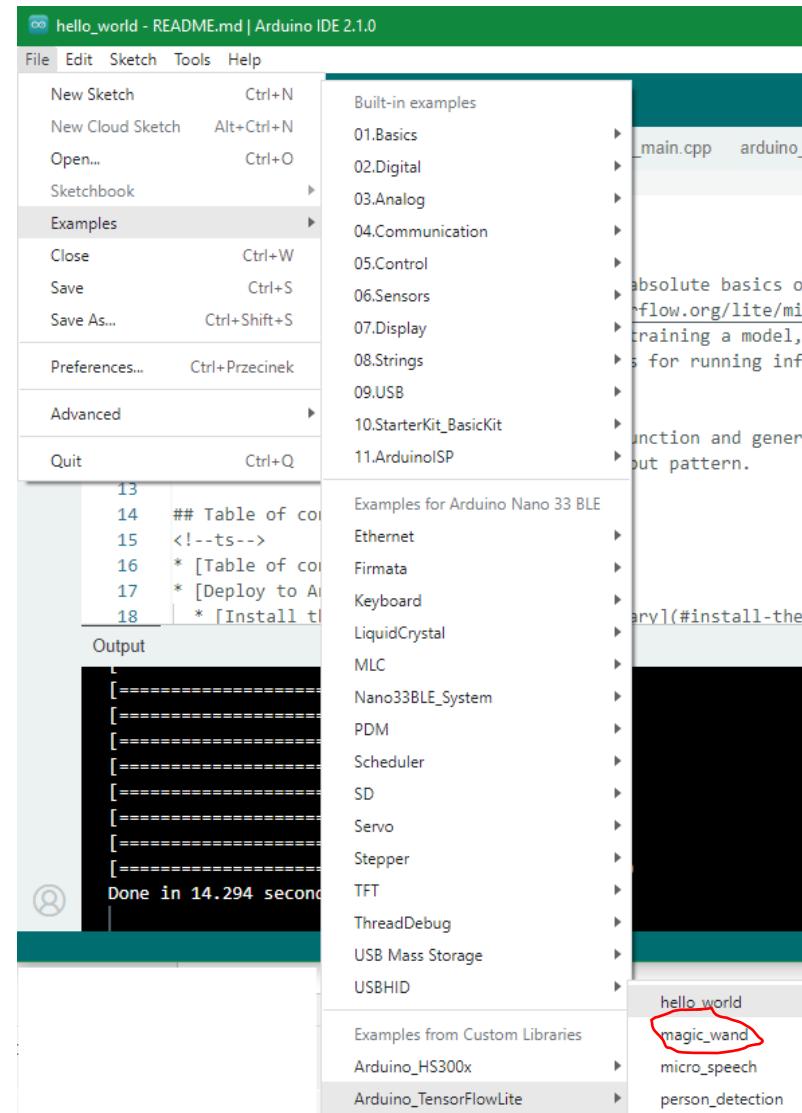
```
/* Copyright 2019 The TensorFlow Authors. All Rights Reserved.
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable 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.h"
// We need to keep the data array aligned on some architectures.
#ifndef __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,
    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, 0x00, 0xac, 0x03,
    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, 0x38, 0x00, 0x00, 0x00,
    0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67,
    0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x01, 0x00, 0x00, 0x00,
    0x04, 0x00, 0x00, 0x00, 0x94, 0xff, 0xff, 0xb, 0x00, 0x00, 0x00,
```

sine_model_quantized.cc

```
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, 0x00, 0xac, 0x03,
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, 0x38, 0x00, 0x00, 0x00,
```

```
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, 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, 0x00,
8     0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00,
9     0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67, 0x5f,
10    0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x01, 0x00, 0x00, 0x00, 0x00,
11    0x04, 0x00, 0x00, 0x00, 0x94, 0xff, 0xff, 0xb, 0x00, 0x00, 0x00, 0x00, 0x00}
```

Arduino ML Magic Wand



Arduino ML Magic Wand

The screenshot shows the Arduino IDE Library Manager interface. At the top, there's a green header bar with the title "magic_wand | Arduino IDE 2.1.0". Below it is a toolbar with icons for saving, loading, and preferences. The main area has tabs for "LIBRARY MANAGER", "magic_wand.ino", "README.md", "magic_wand_model_data.cpp", and "magic_wand". On the left, there's a sidebar with a search icon, a gear icon, and a magnifying glass icon. It lists two libraries: "ArduinoBLE by Arduino" (version 1.3.4) and "Arduino_ScienceJournal" (version 1.2.4). Both have "INSTALL" buttons. The "ArduinoBLE" entry is circled in red. The "ArduinoBLE" card on the right contains its description, version 1.3.4, and an "INSTALL" button. The "magic_wand.ino" tab is active, showing the 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,
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" section below shows the compilation command and the error message:

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

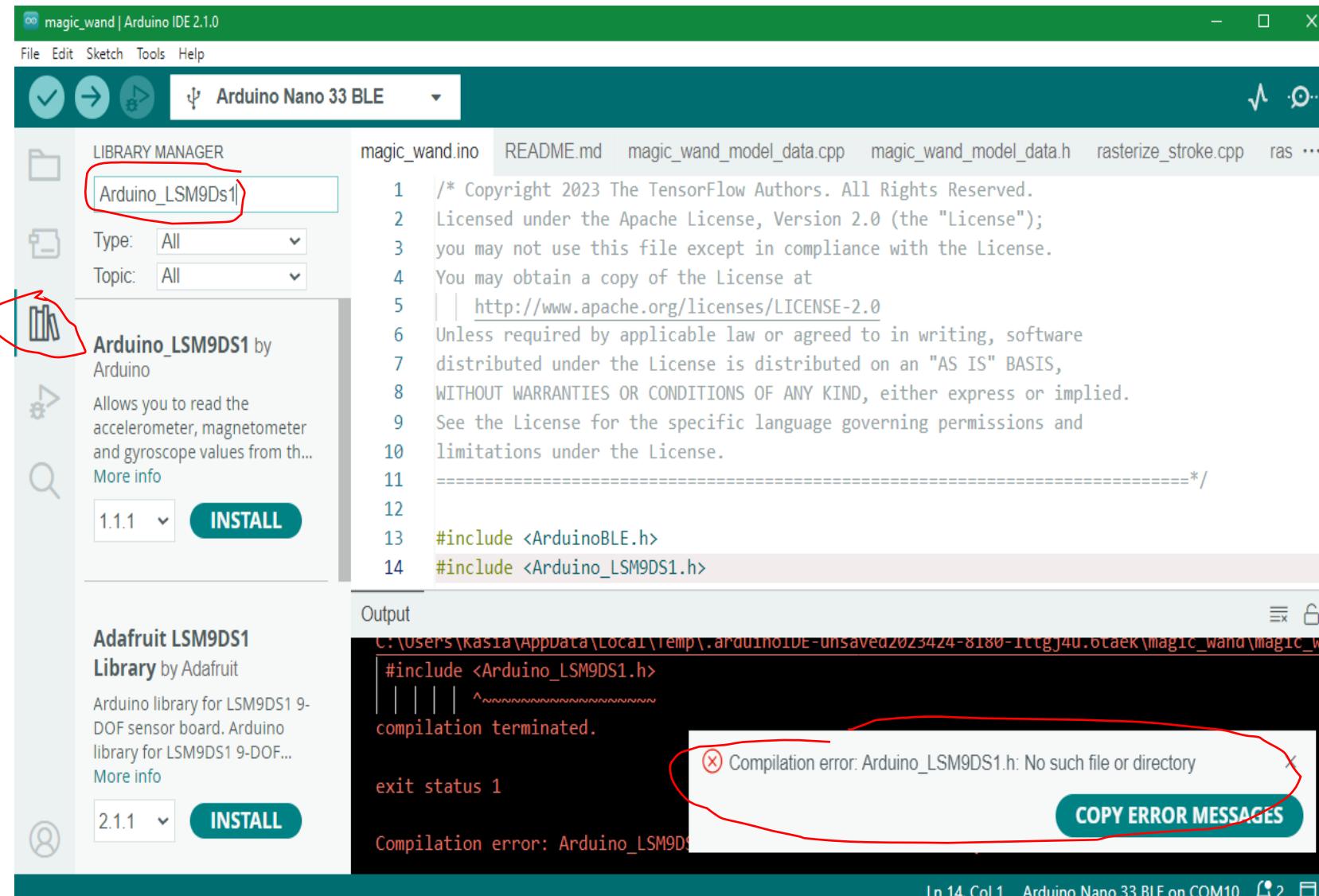
exit status 1

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

Output

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

Arduino ML Magic Wand

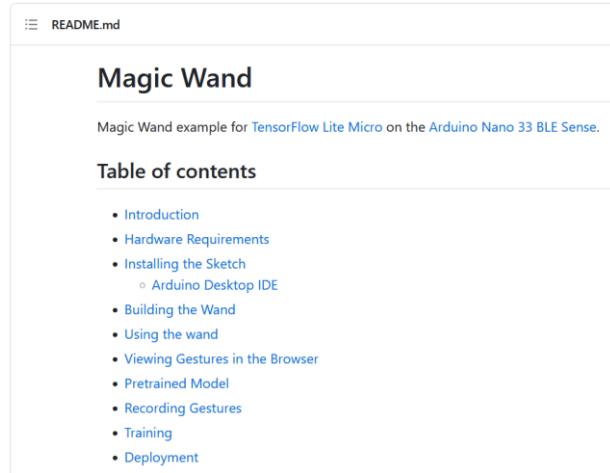


Output

```
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



The screenshot shows the contents of a README.md file. At the top, there's a navigation bar with a file icon and the text "README.md". Below it is a section titled "Magic Wand" with a bold header. A descriptive text follows: "Magic Wand example for TensorFlow Lite Micro on the Arduino Nano 33 BLE Sense.". Underneath, there's a "Table of contents" section with a bold header. The table of contents lists the following items:

- 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-G94GGI”

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

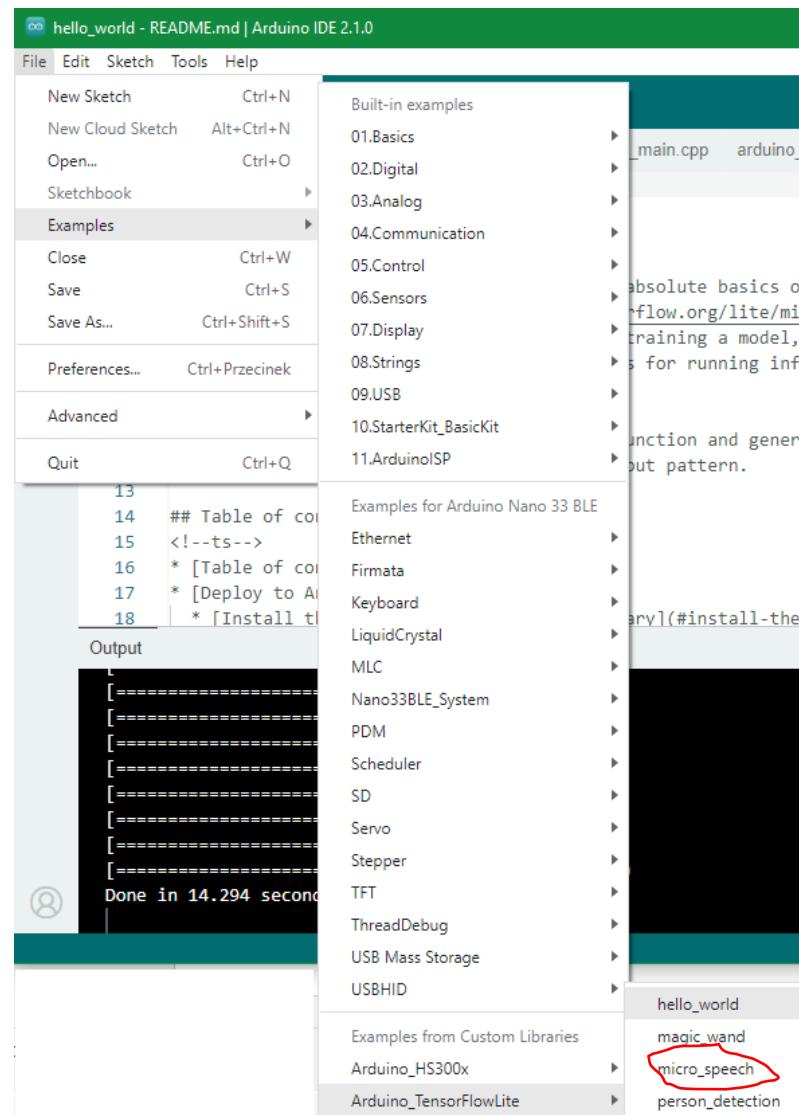
Click button to connect to the board

na
leży
włączyć
Bluetoooh w
komputerze

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

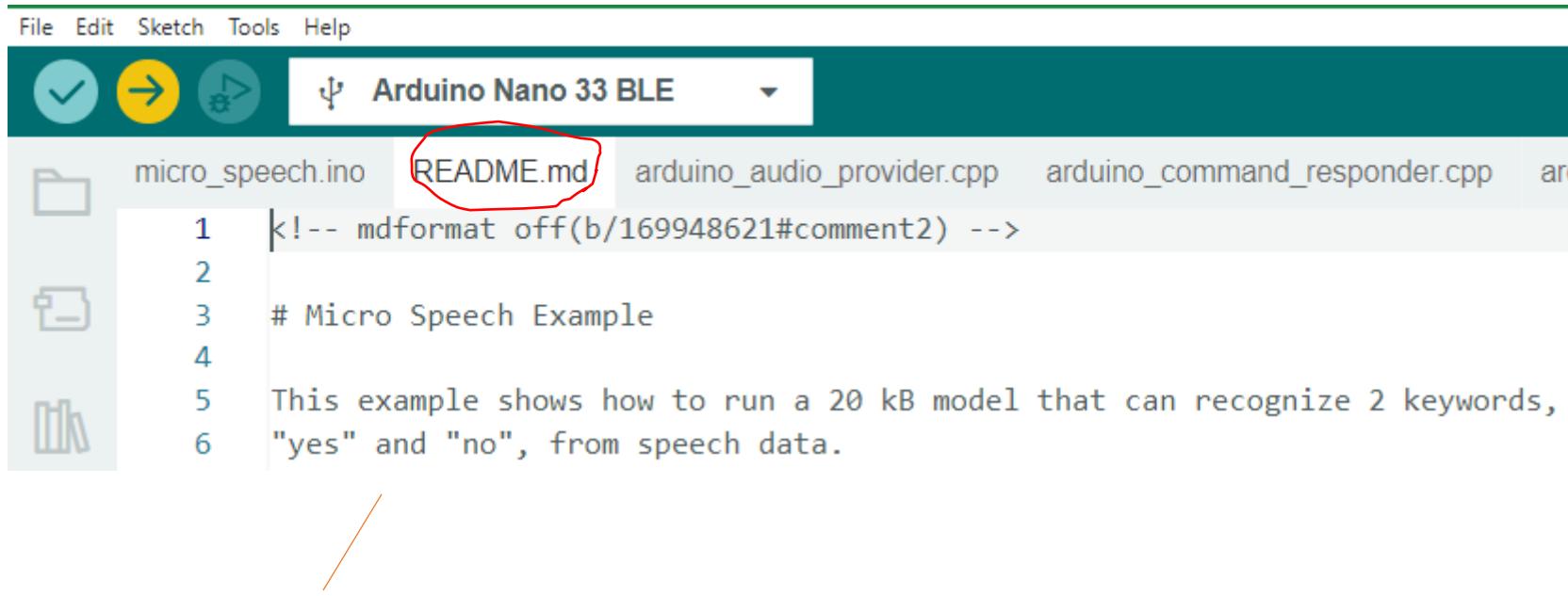
<https://github.com/tensorflow/tflite-micro-arduino-examples>

Arduino ML Micro Speech



Arduino

ML Micro Speech



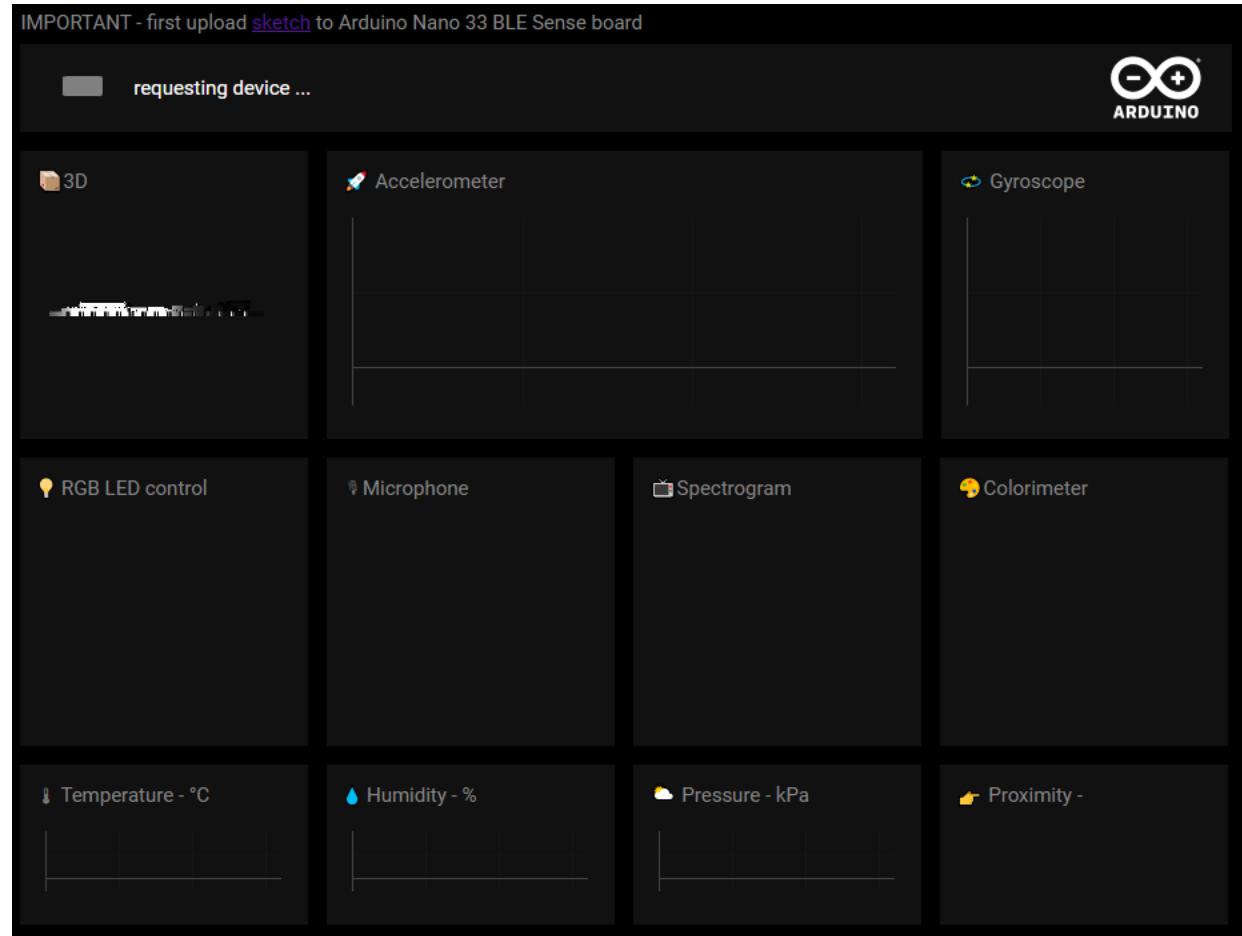
The screenshot shows the Arduino IDE interface. The title bar reads "Arduino Nano 33 BLE". Below the title bar, there is a toolbar with three icons: a green checkmark, a yellow arrow, and a blue gear. The main workspace shows a folder structure on the left with files listed on the right. A file named "README.md" is highlighted with a red circle. The code editor window displays the following text:

```
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
przykłdu

Arduino ML

strona do podglądu czujników



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

Arduino ML

```
File Edit Sketch Tools Help
New Sketch Ctrl+N
New Cloud Sketch Alt+Ctrl+N
Open... Ctrl+O
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Preferences... Ctrl+Przecinek
Advanced
Quit Ctrl+Q

ino Nano 33 BLE

11 You can
12 nRF Conn
13 created
14
15 This ex
16 */
17
18 #include <
19
20 // Bluetoo
21 BLEService
22
23 // Bluetoo
24 BLEUnsigned
25 | BLERead
26
27 int oldBa
28 long prev:
29
Output Serial Monitor
Message (Enter to send)
Started
Failed to initial
Started
Failed to initial
Started
Failed to initial
Started
Failed to initial

Advertising
BatteryMonitor
ButtonLED
CallbackLED
EncryptedBatteryMonitor
LED

Examples from Custom Libraries
Arduino_APDS9960
Arduino_HS300x
Arduino_HTS221
Arduino_LPS22HB
Arduino_LSM9DS1
Arduino_TensorFlowLite
ArduinoBLE
Central
Peripheral

E' on 'COM10')

Examples for Arduino Nano 33 BLE
Ethernet
Firmata
Keyboard
LiquidCrystal
MLC
Nano33BLE_System
PDM
Scheduler
SD
Servo
Stepper
TFT
ThreadDebug
USB Mass Storage
USBHID
```

Google Play Gry Aplikacje Filmy Książki Dla dzieci

nRF Connect for Mobile

Nordic Semiconductor ASA

4.8★ 2,39 tys. opinii 1 min+ Potrafi PEŁNA

Zainstaluj Dodaj do listy życzeń



Kontakt z deweloperem

Podobne aplikacje



test Bluetooth BLE

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

Bluetooth BLE 2 Arduino

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