Tiny Machine Learning using Arduino Nano 33 BLE Sense

Summary

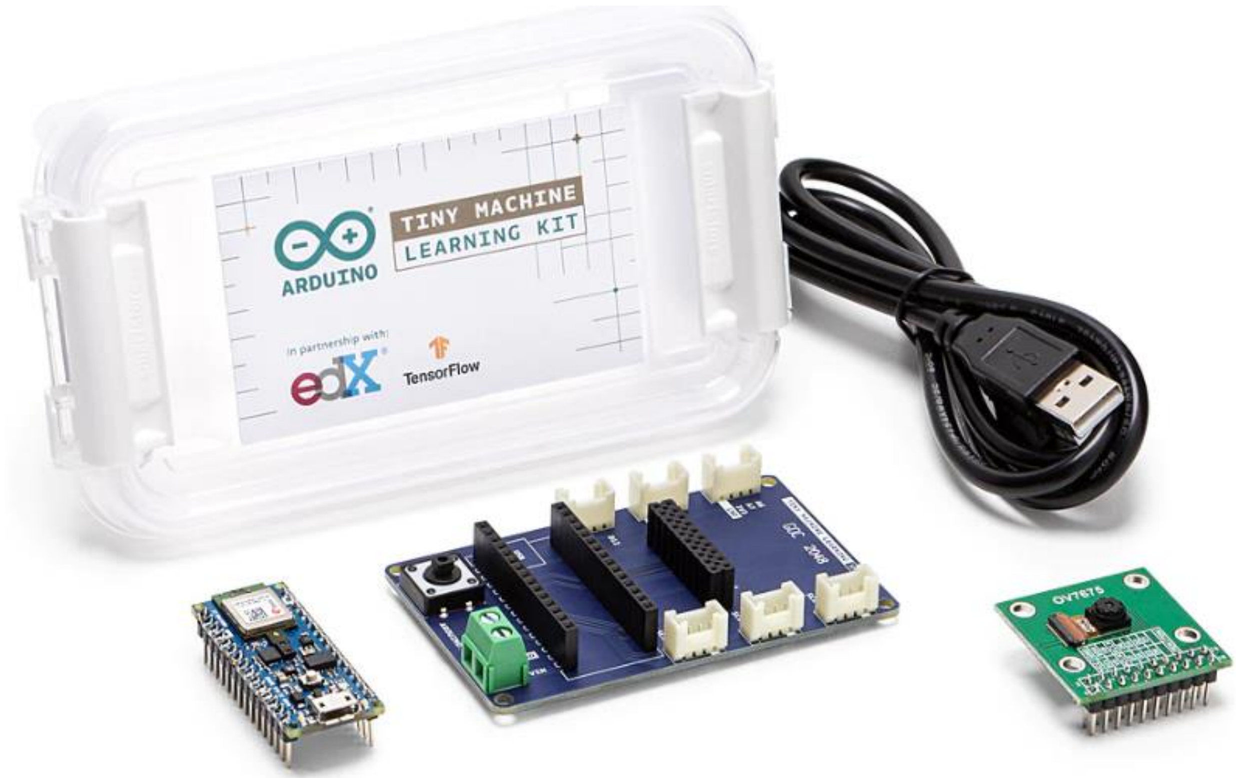
1-Presentation of Arduino Nano 33 BLE Sense
2-Components of Arduino Nano 33 BLE Sense
3-Microcontrollers and TinyML
4-TensorFlow Lite for Microcontrollers Microcontrollers examples
5- Streaming sensor data from the Arduino board
6- Capturing gesture training data
7- Training in TensorFlow
8- Classifying IMU Data

References:
https://blog.arduino.cc/2019/10/15/get-started-with-machine-learning-on-arduino/

Tiny Machine Learning using Arduino Nano 33 BLE Sense

1-Presentation of Arduino Nano 33 BLE Sense

Arduino Tiny Machine Learning Kit:
- 1 Arduino Nano 33 BLE Sense board
- 1 OV7675 Camera
- 1 Arduino Tiny Machine Learning Shield
- 1 USB A to Micro USB Cable



Tiny Machine Learning using Arduino Nano 33 BLE Sense

1-Presentation of Arduino Nano 33 BLE Sense

“The Future of Machine Learning is Tiny and Bright. We’re excited to see what you’ll do!”
Prof. Vijay Janapa Reddi, Harvard University and Pete Warden, Google

Tiny Machine Learning using Arduino Nano 33 BLE Sense

1-Presentation of Arduino Nano 33 BLE Sense

The kit consists of a powerful board equipped with a microcontroller and a wide variety of sensors (Arduino Nano 33 BLE Sense*).

The board can sense movement, acceleration, rotation, temperature, humidity, barometric pressure, sounds, gestures, proximity, color, and light intensity.

The kit also includes a camera module (OV7675)

and custom Arduino shield to make it easy to attach your components and create your very own unique TinyML project.

You will be able to explore practical ML use cases using classical algorithms as well as deep neural networks powered by TensorFlow Lite Micro. The possibilities are limited only by your imagination!

Tiny Machine Learning using Arduino Nano 33 BLE Sense

1-Presentation of Arduino Nano 33 BLE Sense

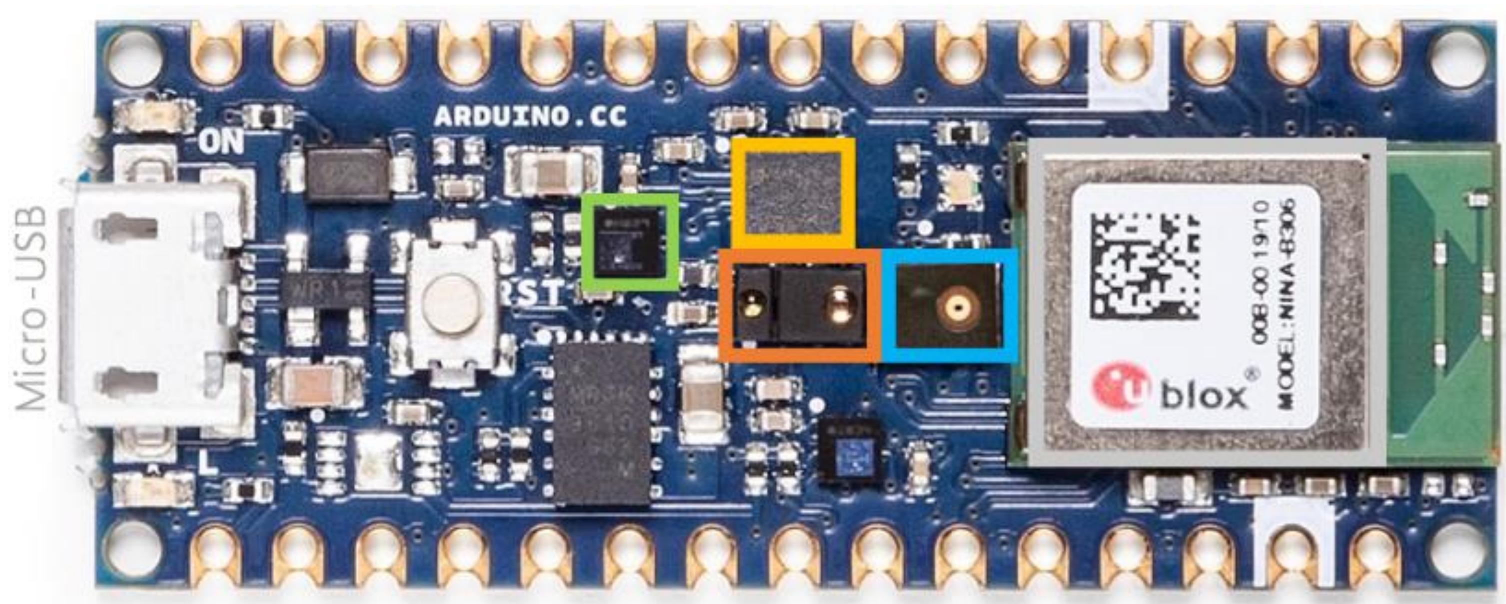
Ever wondered how to build a small intelligent device that reacts to sounds like a keyword being spoken, recognizes gestures like waving a magic wand, or even recognize faces?

With this kit combined with the power of Tiny Machine Learning (TinyML) you can do all of that and much more!

We want to show you how these possibilities can be part of your own tiny smart device!

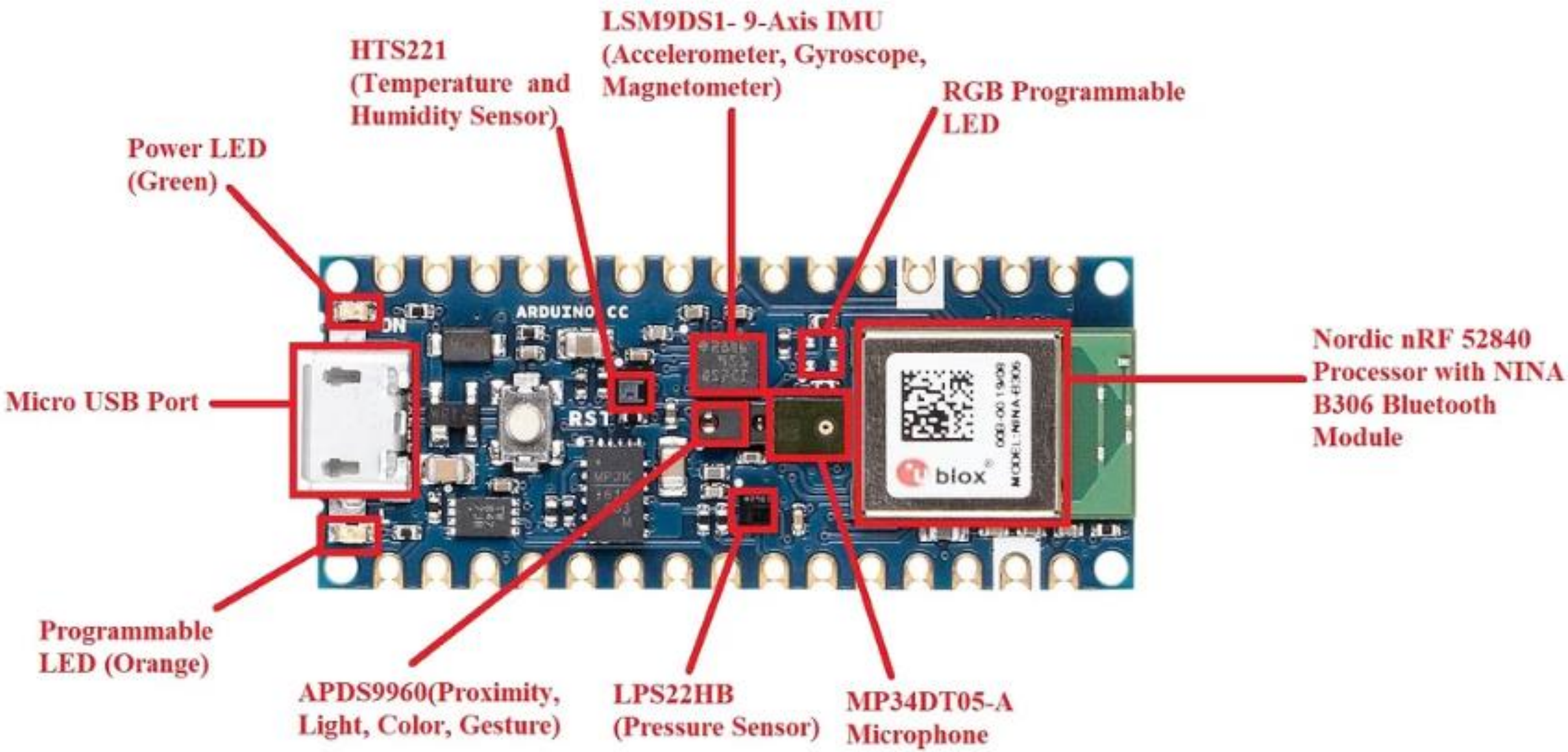
Tiny Machine Learning using Arduino Nano 33 BLE Sense

2-Components of Arduino Nano 33 BLE Sense



Tiny Machine Learning using Arduino Nano 33 BLE Sense

2-Components of Arduino Nano 33 BLE Sense



Tiny Machine Learning using Arduino Nano 33 BLE Sense

2-Components of Arduino Nano 33 BLE Sense

Motion, vibration and orientation sensor: what makes this board ideal for wearable devices to get highly accurate measurements of the environmental conditions

Temperature, humidity and pressure sensor: you could make a simple weather station

Digital microphone: to capture and analyse sound in real time

Color, brightness, proximity and gesture sensor : estimate the room's luminosity, but also whether someone is moving close to the board

Tiny Machine Learning using Arduino Nano 33 BLE Sense

3-Microcontrollers and TinyML

Arduino is on a mission to make machine learning simple enough for anyone to use.

We've been working with the TensorFlow Lite team over the past few months and are excited to show you what we've been up to together: bringing TensorFlow Lite Micro to the Arduino Nano 33 BLE Sense.

In this workshop, we'll show you how to install and run several new TensorFlow Lite Micro examples that are available in the Arduino Library Manager

Tiny Machine Learning using Arduino Nano 33 BLE Sense

3-Microcontrollers and TinyML

Microcontrollers, such as those used on Arduino boards, are low-cost, single chip, self-contained computer systems. They're the invisible computers embedded inside *billions* of everyday gadgets like wearables, drones, 3D printers, toys, rice cookers, smart plugs, e-scooters, washing machines. The trend to connect these devices is part of what is referred to as the Internet of Things.

The board we're using here has an Arm Cortex-M4 microcontroller running at 64 MHz with 1MB Flash memory and 256 KB of RAM. This is tiny in comparison to cloud, PC, or mobile but reasonable by microcontroller standards.

Tiny Machine Learning using Arduino Nano 33 BLE Sense

3-Microcontrollers and TinyML

There are practical reasons you might want to squeeze ML on microcontrollers, including:

- Function – wanting a smart device to act quickly and locally (independent of the Internet).
- Cost – accomplishing this with simple, lower cost hardware.
- Privacy – not wanting to share all sensor data externally.
- Efficiency – smaller device form-factor, energy-harvesting or longer battery life.

- Machine learning can make microcontrollers accessible to developers who don’t have a background in embedded development

4-TensorFlow Lite for Microcontrollers Microcontrollers examples

The inference examples for TensorFlow Lite for Microcontrollers are now packaged and available through the Arduino Library Manager making it possible to include and run them on Arduino in a few clicks.

The examples are:
micro_speech – speech recognition using the onboard microphone
magic_wand – gesture recognition using the onboard IMU
person_detection – person detection using an external ArduCam camera

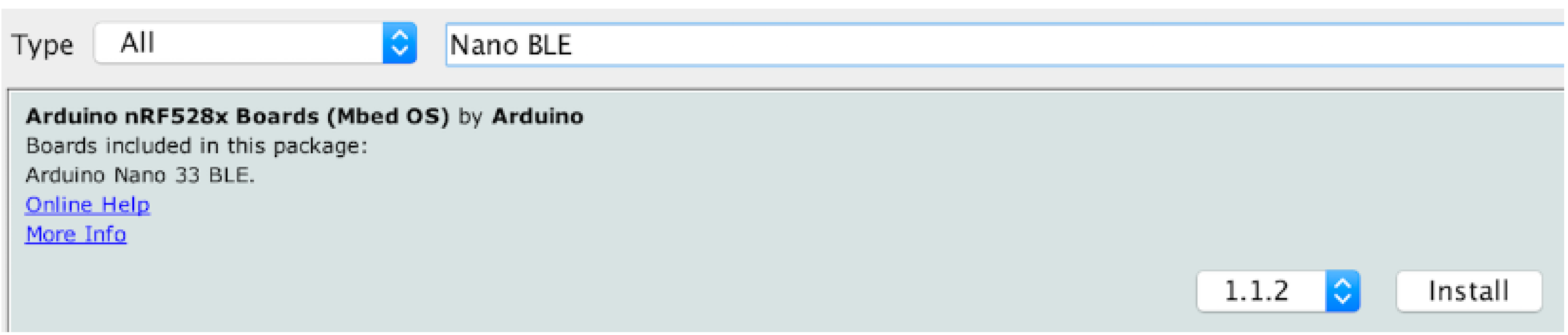
We show in the next how to deploy and run them on an Arduino.

Tiny Machine Learning using Arduino Nano 33 BLE Sense

4-TensorFlow Lite for Microcontrollers Microcontrollers examples

In the Arduino IDE, via the File >
Examples > Arduino_TensorFlowLite menu.
Select an example and the sketch will open, compile, upload and run the examples on the board.

In the Arduino IDE menu select Tools > Board > Boards Manager…Search for "Nano BLE" and press install on the board



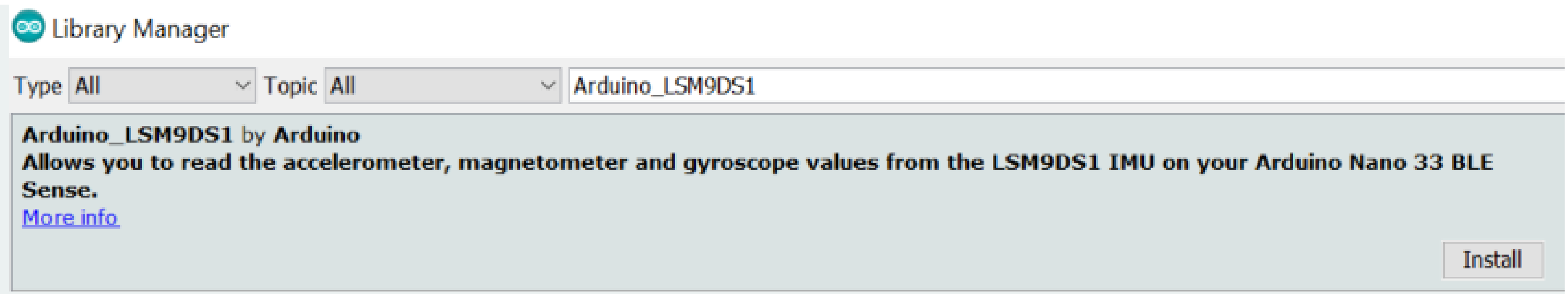
Tiny Machine Learning using Arduino Nano 33 BLE Sense

4-TensorFlow Lite for Microcontrollers Microcontrollers examples

Go to the Library Manager Tools > Manage Libraries...

Search for and install the Arduino_TensorFlowLite library

Next search for and install the Arduino_LSM9DS1 library:



Finally, plug the micro USB cable into the board and your computer

Choose the board Tools > Board > Arduino Nano 33 BLE

Choose the port Tools > Port > COM5 (Arduino Nano 33 BLE)

Note that the actual port name may be different on your computer

Tiny Machine Learning using Arduino Nano 33 BLE Sense

5- Streaming sensor data from the Arduino board

We'll be using a pre-made sketch IMU_Capture.ino which does the following:
- Monitor the board's accelerometer and gyroscope
- Trigger a sample window on detecting significant linear acceleration of the board
- Sample for one second at 119Hz, outputting CSV format data over USB
- Loop back and monitor for the next gesture

To program the board with this sketch in the Arduino IDE:
Download IMU_Capture.ino and open it in the Arduino IDE
Compile and upload it to the board with Sketch > Upload

Tiny Machine Learning using Arduino Nano 33 BLE Sense

5- Streaming sensor data from the Arduino board

In the Arduino IDE, open the Serial Plotter Tools > Serial Plotter

If you get an error that the board is not available, reselect the port: Tools > Port > portname (Arduino Nano 33 BLE)

Pick up the board and practice your punch and flex gestures

You'll see it only sample for a one second window, then wait for the next gesture

You should see a live graph of the sensor data capture.

Tiny Machine Learning using Arduino Nano 33 BLE Sense

6- Capturing gesture training data

To capture data as a CSV log to upload to TensorFlow, you can use Arduino IDE > Tools > Serial Monitor to view the data and export it to your desktop machine:

Reset the board by pressing the small white button on the top

Pick up the board in one hand (picking it up later will trigger sampling)

In the Arduino IDE, open the Serial Monitor Tools > Serial Monitor

  If you get an error that the board is not available, reselect the port:

  Tools > Port > portname (Arduino Nano 33 BLE)

Tiny Machine Learning using Arduino Nano 33 BLE Sense

6- Capturing gesture training data

Make a punch gesture with the board in your hand (Be careful whilst doing this!)

- Make the outward punch quickly enough to trigger the capture
- Return to a neutral position slowly so as not to trigger the capture again

Repeat the gesture capture step 10 or more times to gather more data

Copy and paste the data from the Serial Console to new text file called punch.csv

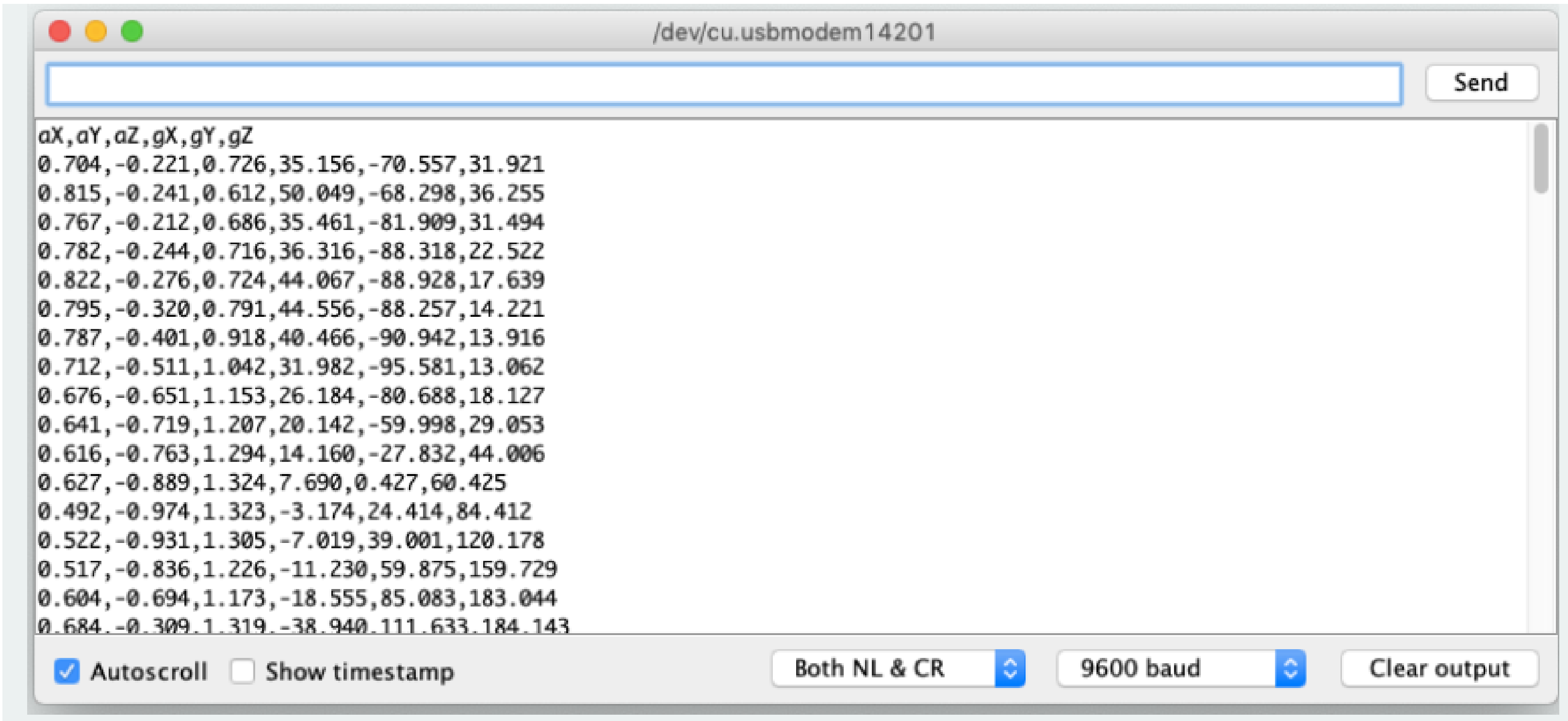
Clear the console window output and repeat all the steps above, this time with a flex gesture in a file called flex.csv

- Make the inward flex fast enough to trigger capture returning slowly each time

Note the first line of your two csv files should contain the fields aX,aY,aZ,gX,gY,gZ.

Tiny Machine Learning using Arduino Nano 33 BLE Sense

6- Capturing gesture training data



Tiny Machine Learning using Arduino Nano 33 BLE Sense

7- Training in TensorFlow

We're going to use [Google Colab] to train our machine learning model using the data we collected from the Arduino board in the previous section. Colab provides a Jupyter notebook that allows us to run our TensorFlow training in a web browser.

The colab will step you through the following:
Set up Python environment
Upload the punch.csv and flex.csv data
Parse and prepare the data
Build and train the model
Convert the trained model to TensorFlow Lite
Encode the model in an Arduino header file
The final step of the colab is generates the model.h file to download and include in our Arduino IDE gesture classifier project in the next section:

Let's open the notebook in Colab and run through the steps in the cells – arduino_tinyml_workshop.ipynb

Tiny Machine Learning using Arduino Nano 33 BLE Sense

8- Classifying IMU Data

Next we will use model.h file we just trained and downloaded from Colab in the previous section in our Arduino IDE project:

Open IMU_Classifier.ino in the Arduino IDE.
Create a new tab in the IDE. When asked name it model.h
Open the model.h tab and paste in the version you downloaded from Colab
Upload the sketch: Sketch > Upload
Open the Serial Monitor: Tools > Serial Monitor
Perform some gestures
The confidence of each gesture will be printed to the Serial Monitor (0 = low confidence, 1 = high confidence)