BOISE STATE UNIVERSITY

GETTING STARTED

HARDWARE & SOFTWARE NEEDED

- An **Arduino Nano 33 BLE Sense** board
- A Micro USB cable to connect the Arduino board to your desktop machine or laptop
- To program your board, you can use the **Arduino Cloud Editor** or install the **Arduino IDE**.
- **TensorFlow Lite Micro Library** (download only available via GitHub).

ARDUINO NANO 33 BLE SENSE FEATURES AND SENSORS

The Arduino Nano 33 BLE Sense has a variety of onboard sensors meaning potential for some cool TinyML applications:

- Voice – digital microphone
- Motion – 9-axis IMU (accelerometer, gyroscope, magnetometer)
- Environmental – temperature, humidity and pressure
- Light – brightness, color and object proximity

Unlike classic Arduino Uno, the board combines a microcontroller with onboard sensors which means you can address many use cases without additional hardware or wiring. The board is also small enough to be used in end applications like wearables. As the name suggests it has Bluetooth® Low Energy connectivity so you can send data (or inference results) to a laptop, mobile app or other Bluetooth® Low Energy boards and peripherals.

Tip: Sensors on a USB stick – Connecting the BLE Sense board over USB is an easy way to capture data and add multiple sensors to single board computers without the need for additional wiring or hardware – a nice addition to a Raspberry Pi, for example.

MICROCONTROLLERS AND TINYML

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

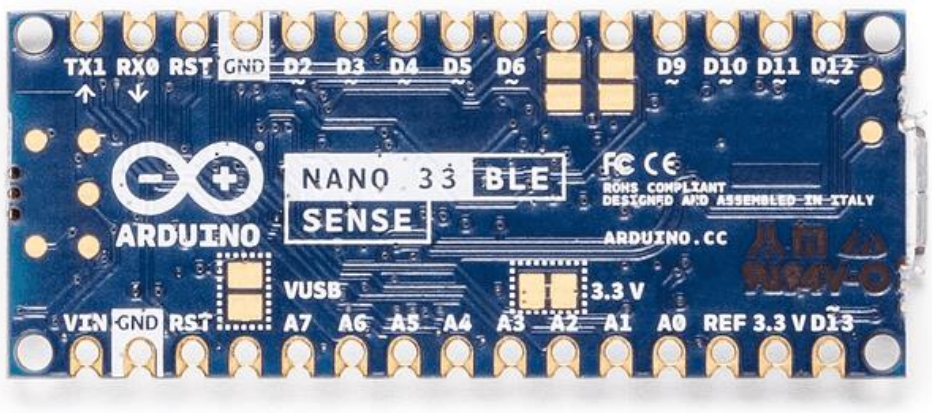


Figure: Arduino Nano 33 BLE Sense Board

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.

There's a final goal which we're building towards that is very important:

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

On the machine learning side, there are techniques you can use to fit neural network models into memory constrained devices like microcontrollers. One of the key steps is the **quantization** of the weights from floating point to 8-bit integers. This also has the effect of making inference quicker to calculate and more applicable to lower clock-rate devices.

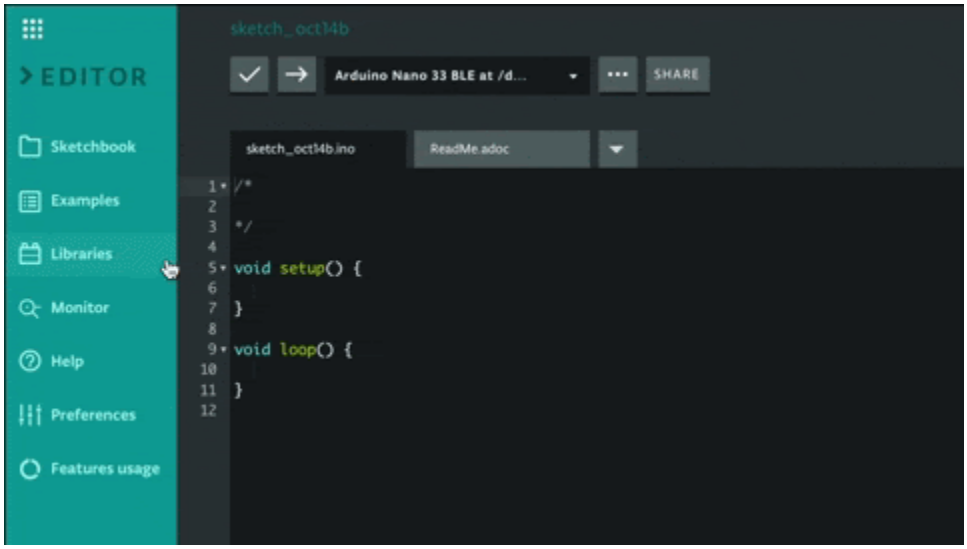
What this Class will Offer

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. In this course we'll show you how to run them. The examples are:

- Deploying key word spotting model using pretrained data and customized data
- Person Detection using the camera sensor included in the kit
- Gesture detection utilizing the additional sensors and onboard IMU

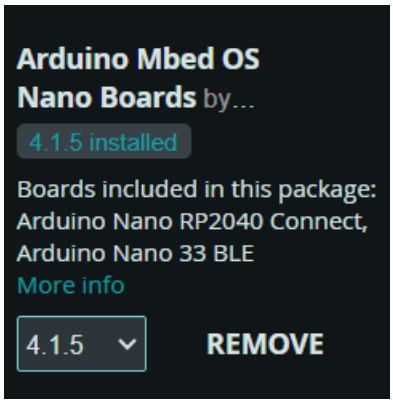
Arduino Create Cloud Editor

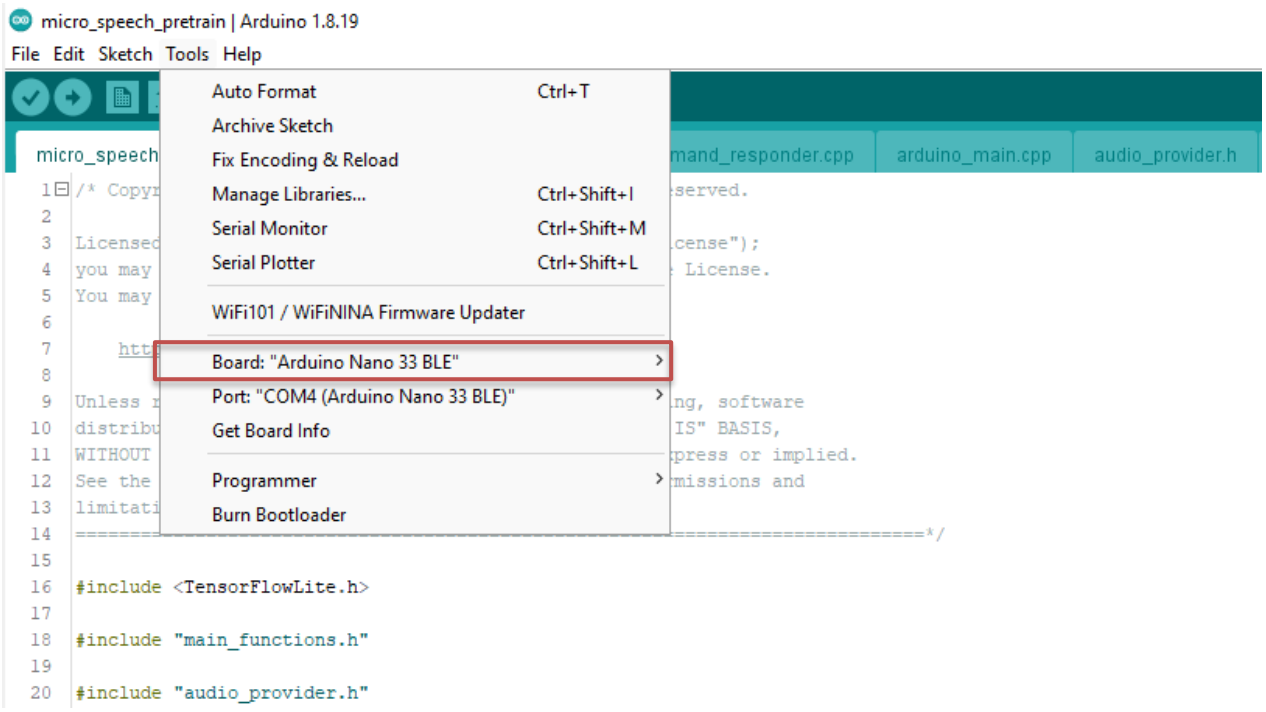
Once you connect your Arduino Nano 33 BLE Sense to your desktop machine with a USB cable you will be able to compile and run the following TensorFlow examples on the board by using the **Arduino Create** Cloud Editor:



ARDUINO IDE SETUP

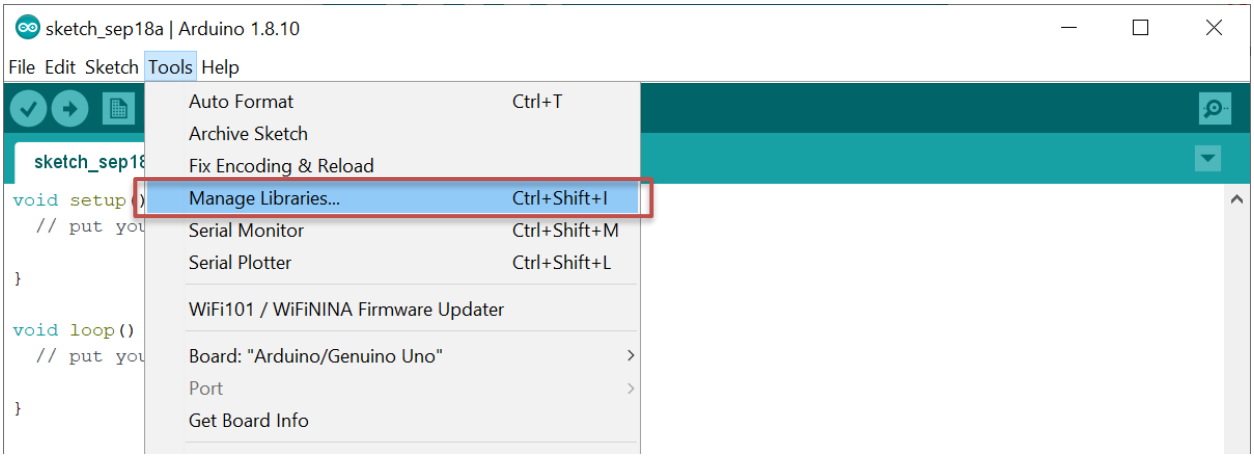
1. First, let's make sure we have the drivers for the Nano 33 BLE boards installed.
2. In the Arduino IDE, go to **Tools > Board > Boards Manager.**
3. In the Boards Manager, search for "**Arduino mbed-enabled Boards**".
4. Install the package "**Arduino Mbed Nano OS Boards**" by Arduino. This package includes support for the **Arduino Nano 33 BLE.**

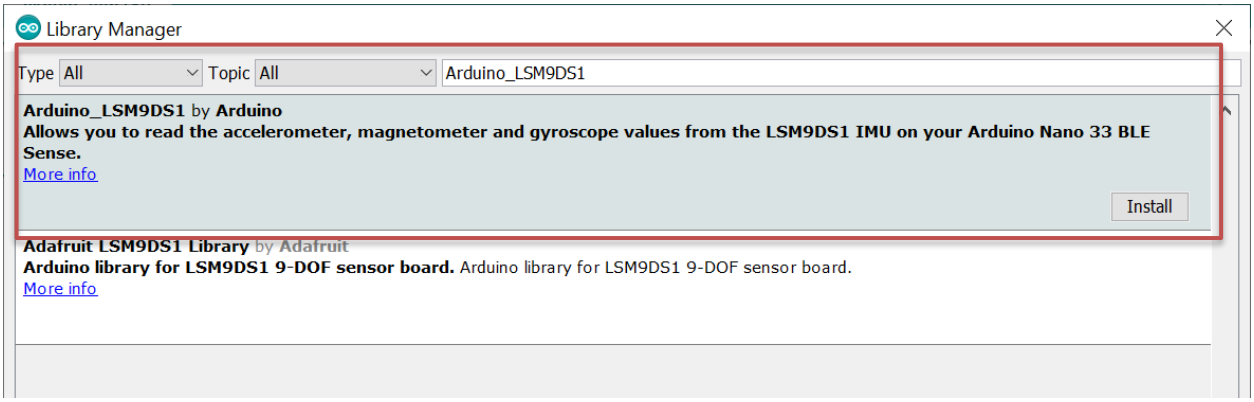


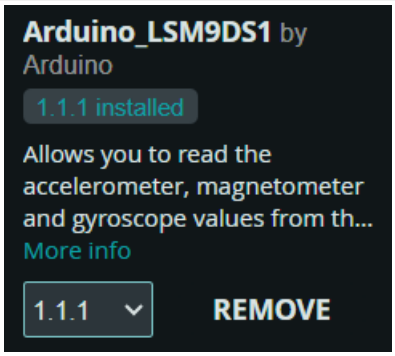


Install Nano BLE 33 Board

5. The **Arduino_LSM9DS1** can be installed in the library manager in the IDE (install the latest version):







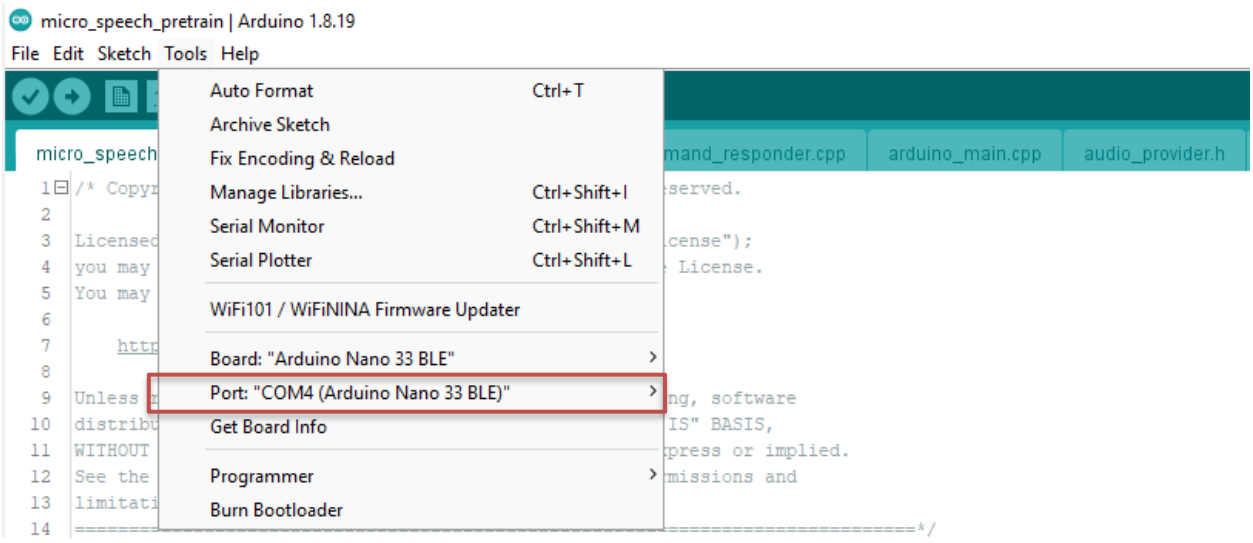
Install Arduino_LSM9DS1

6. Finally, we need to download the [TensorFlow Lite Micro Library] from the repository. It is not available in the library manager, so it needs to be installed manually.
7. Once the zip has been downloaded, in the Arduino IDE, choose **Sketch > Include Library > Add .ZIP Library** and select the file.
8. ***You may need to restart the IDE for it to work.***

TEST YOUR BOARD IS WORKING

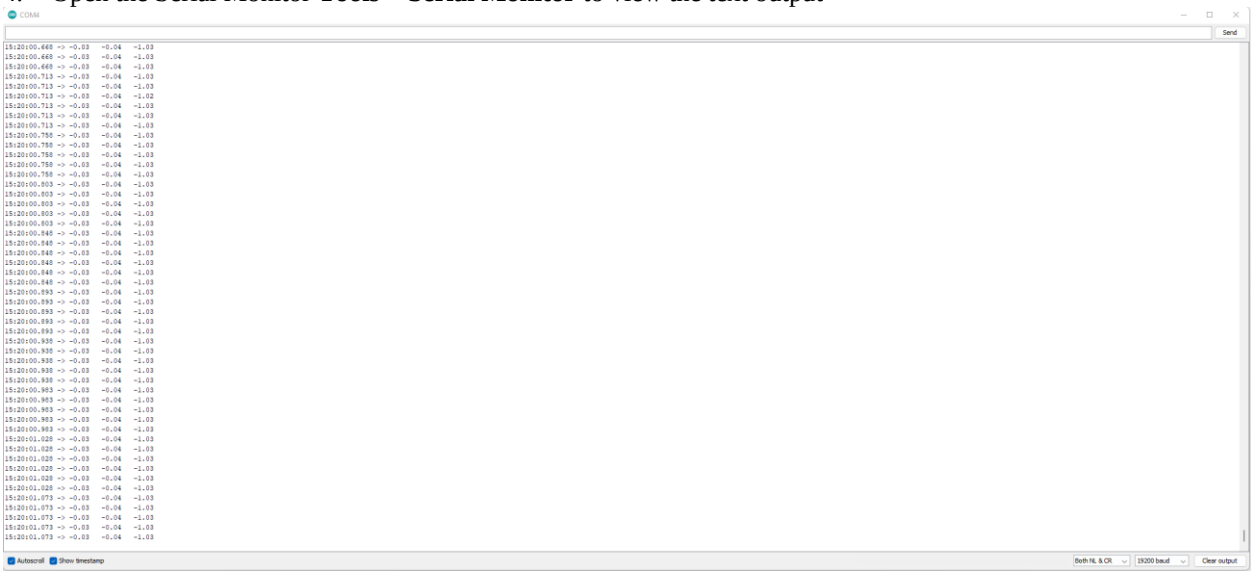
1. Open the Simple Accelerometer sketch using **File > Examples > Arduino_LSM9DS1 > SimpleAccelerometer**
2. Upload the sketch to the board using the **Sketch > Upload** menu or the  button from the tool bar.

3. Make sure your Arduino board is connected to the associated port and should show something like this:



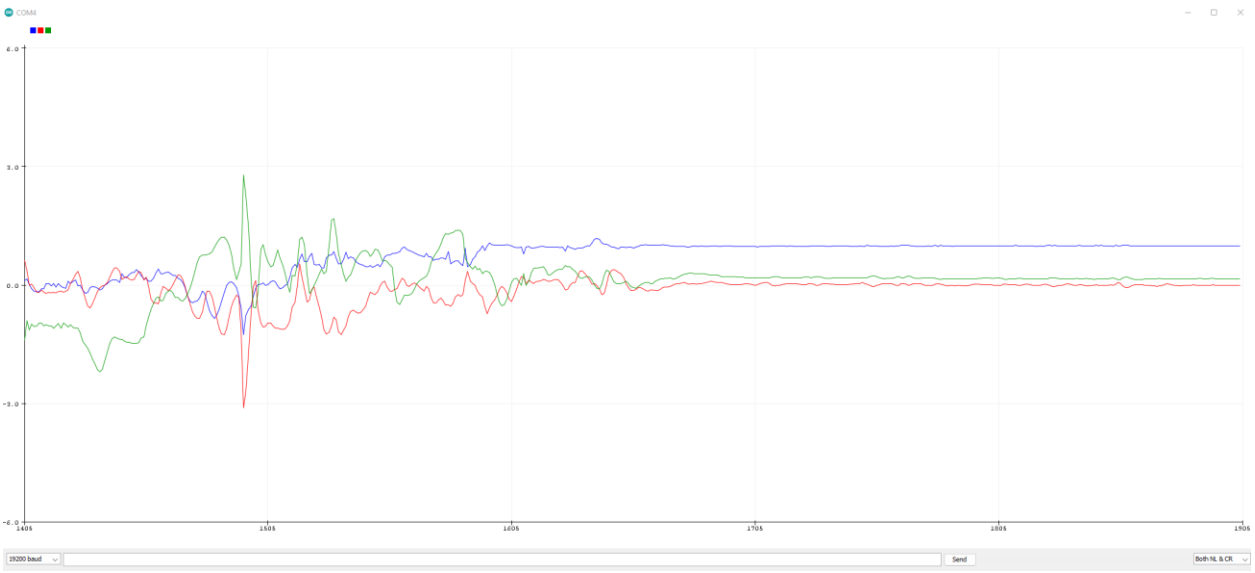
Your laptop port might show a different COM number (i.e. COM3, COM4… …)

4. Open the Serial Monitor **Tools > Serial Monitor** to view the text output



Serial monitor graphical window

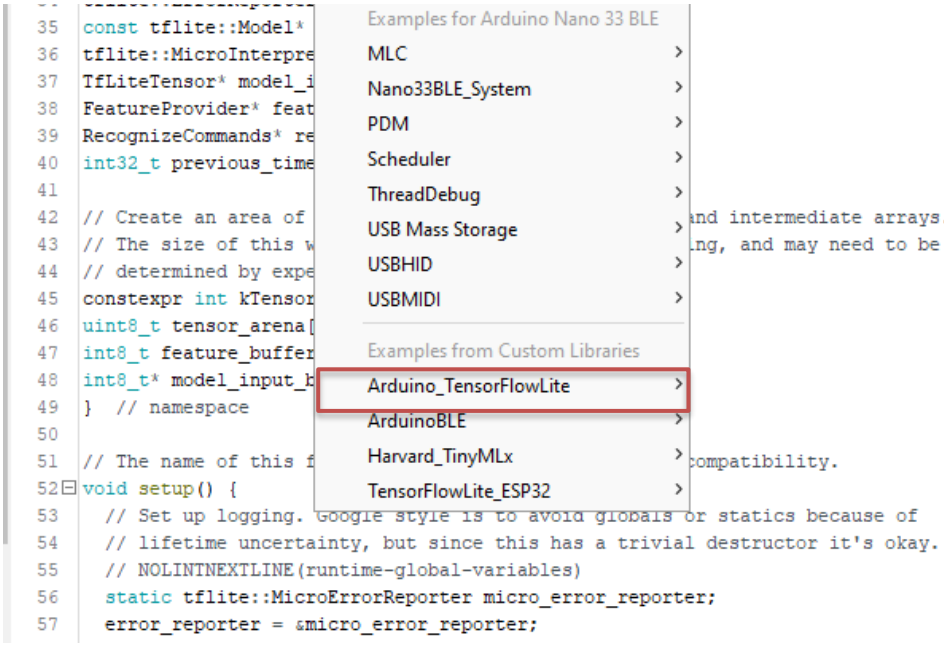
5. Open the Serial Plotter **Tools > Serial Plotter** to view the output on a graph



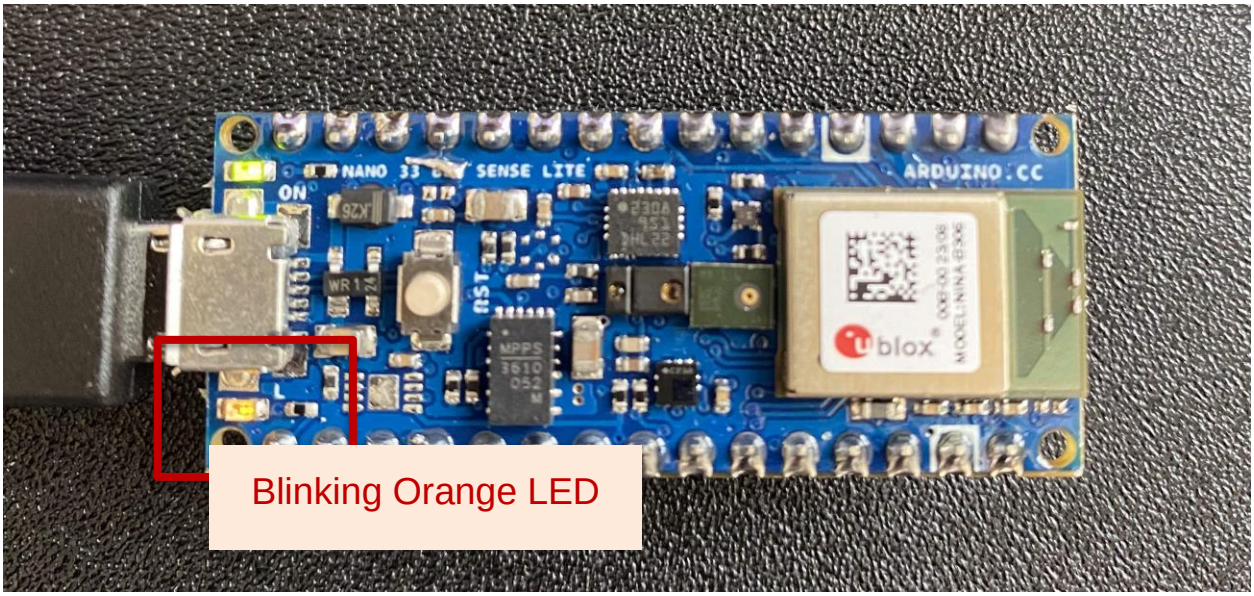
Serial plotter graphical window

TEST YOUR ARDUINO_TENSROFLOWLITE IS WORKING

1. Once the library has been added, go to **File -> Examples**. You should see an entry within the list named **Arduino_TensroflowLite** Select it and click **hello_world** to load the example.
2. Use the Arduino IDE to build and upload the example. Once it is running, you should see the built-in LED on your device flashing.

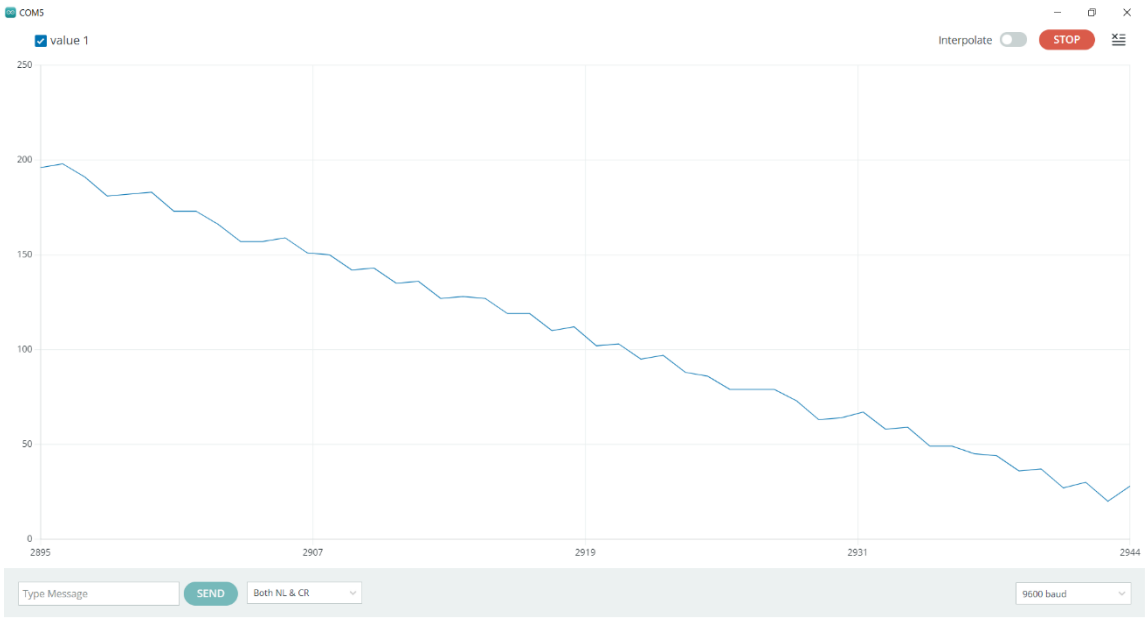


Example from Arduino_TensorFlowLite > hello_world



3. The Arduino Desktop IDE includes a plotter that we can use to display the sine wave graphically. To view it, go to **Tools -> Serial Monitor.** You will see one datapoint being logged for each inference cycle, expressed as a number between 0 and 255 (~ +- 15 between the range). (It will run extremely fast but you can change it if you select different baud rate/counts)

What is baud? It refers to the rate at which data is transmitted over a serial communication interface, such as UART (Universal Asynchronous Receiver-Transmitter). It is defined as the number of signal or symbol changes (which represent bits) that occur per second. For instance, if the baud rate is set to 9600, it means that 9600 bits per second are transmitted or received.



Serial plotter graphical view of the sinewave in 9600 baud rate