

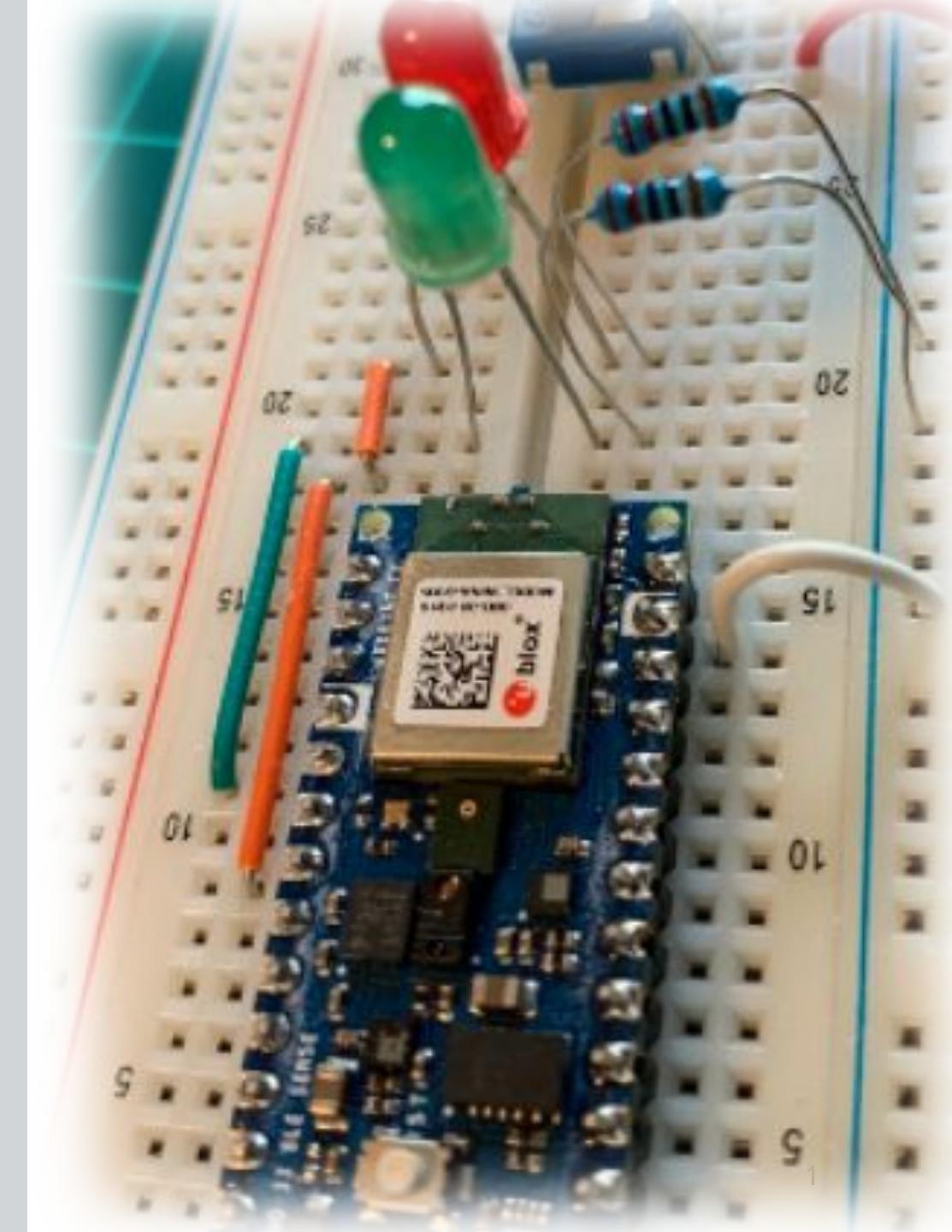
IESTI01 – TinyML

Embedded Machine Learning

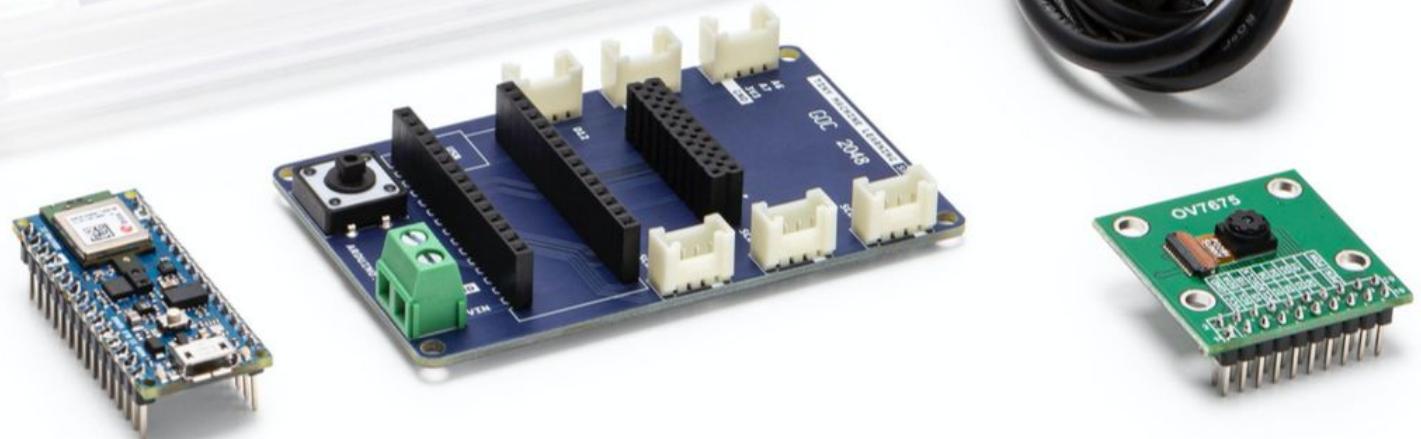
17. TinyML Kit Overview - HW and SW installation & Test



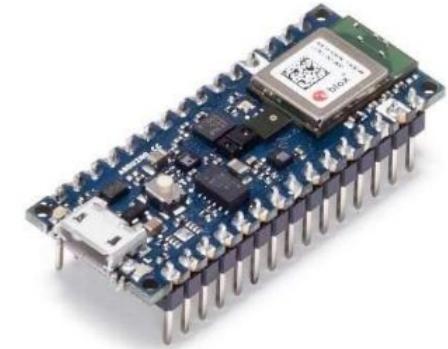
Prof. Marcelo Rovai
UNIFEI



TinyML Kit Overview



Nano 33 BLE Sense (+ USB cable)

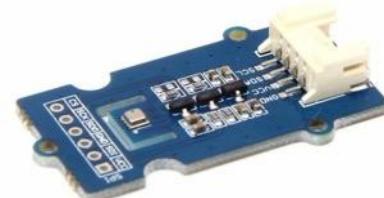


Purpose

AI-enabled developmental **microcontroller board** with USB-A to microB cable

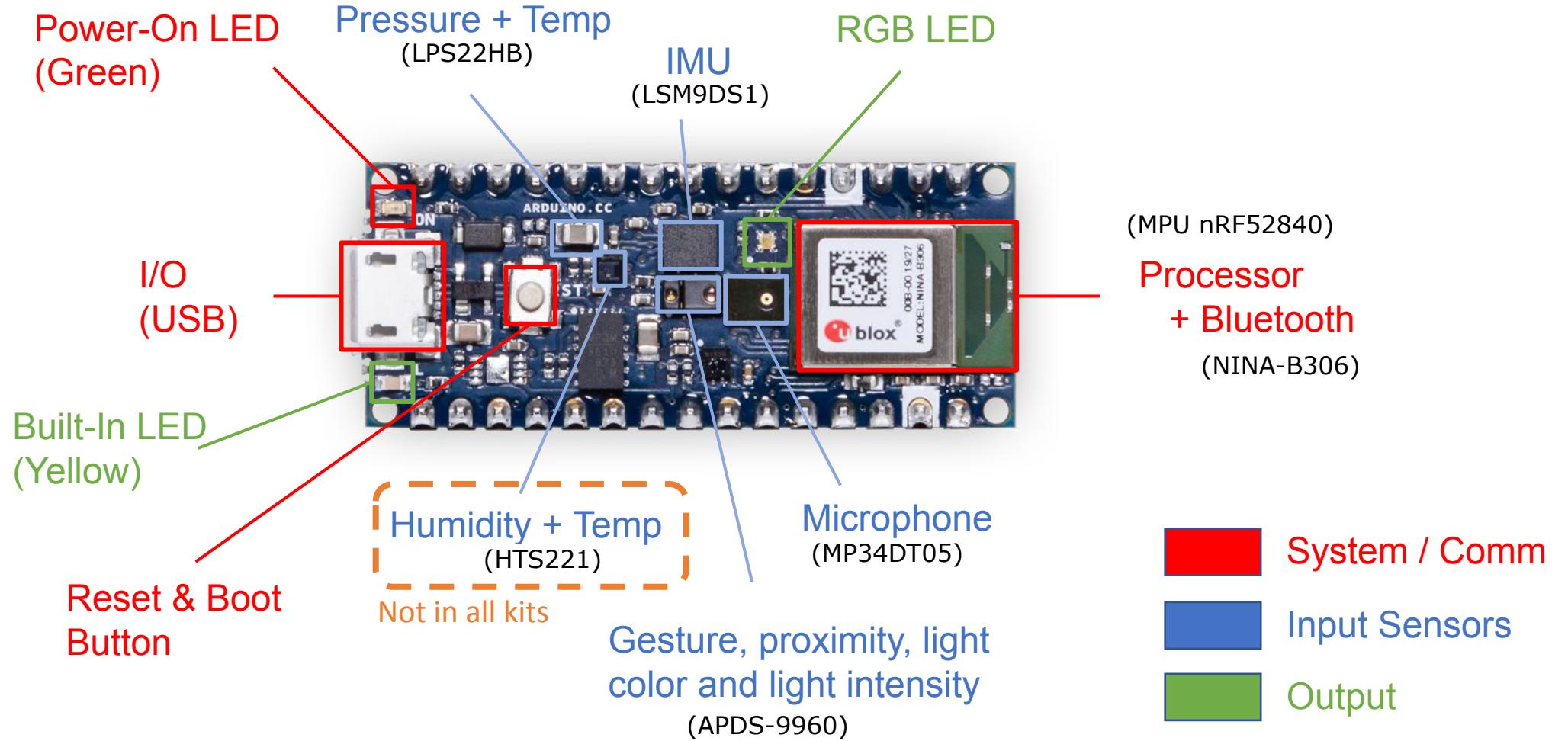
Specifications

- **MPU:** Nordic nRF52840 (ARM Cortex-M4 w/FPU): **3.3V**, 64MHz, 1MB flash, **256 kB RAM**
- **Sensors on board:** microphone, IMU (9 axis), color, light, proximity, barometric, temperature, **humidity***, gesture, and light intensity.
- BLE module covered by ArduinoBLE library
- RGB LEDs



* Not included in some packages. For projects you can use the external Grove - Temp&Humi&Barometer Sensor (BME280)

Nano 33 BLE Sense (Development board)



OV 7675 Camera Module



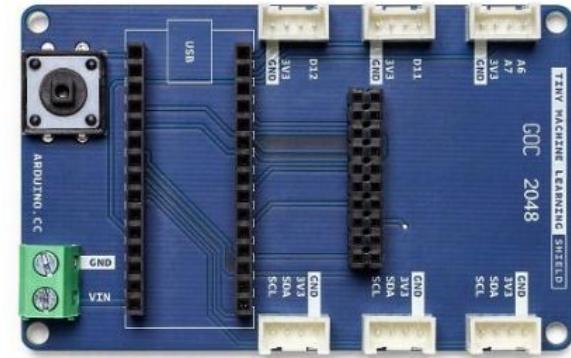
Purpose

Breakout PCB for *tiny* camera.

Specifications

- Low-cost, Low-voltage, **0.3 MP** CMOS VGA (can step down to **QVGA**, QQVGA) image sensor
- Serial Camera Control Bus (SCCB) + Camera Parallel Interface (CPI) / Digital Video Port (DVP) interface
- Breaks ribbon cable out to 2x10 pin array
- **1 or 5 fps** (Frames per Second)

Tiny Machine Learning Shield

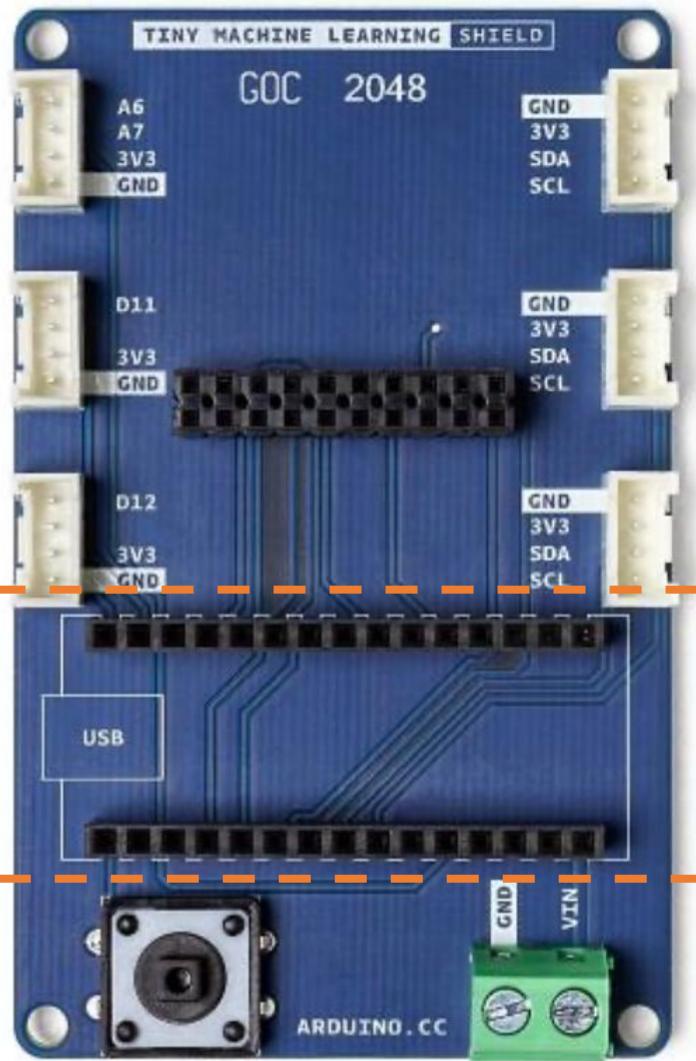


Purpose

A daughter PCB designed to **breakout the I/O** from the Nano 33 BLE sense to permit easy, reliable **communication with** other local, **off-board elements**

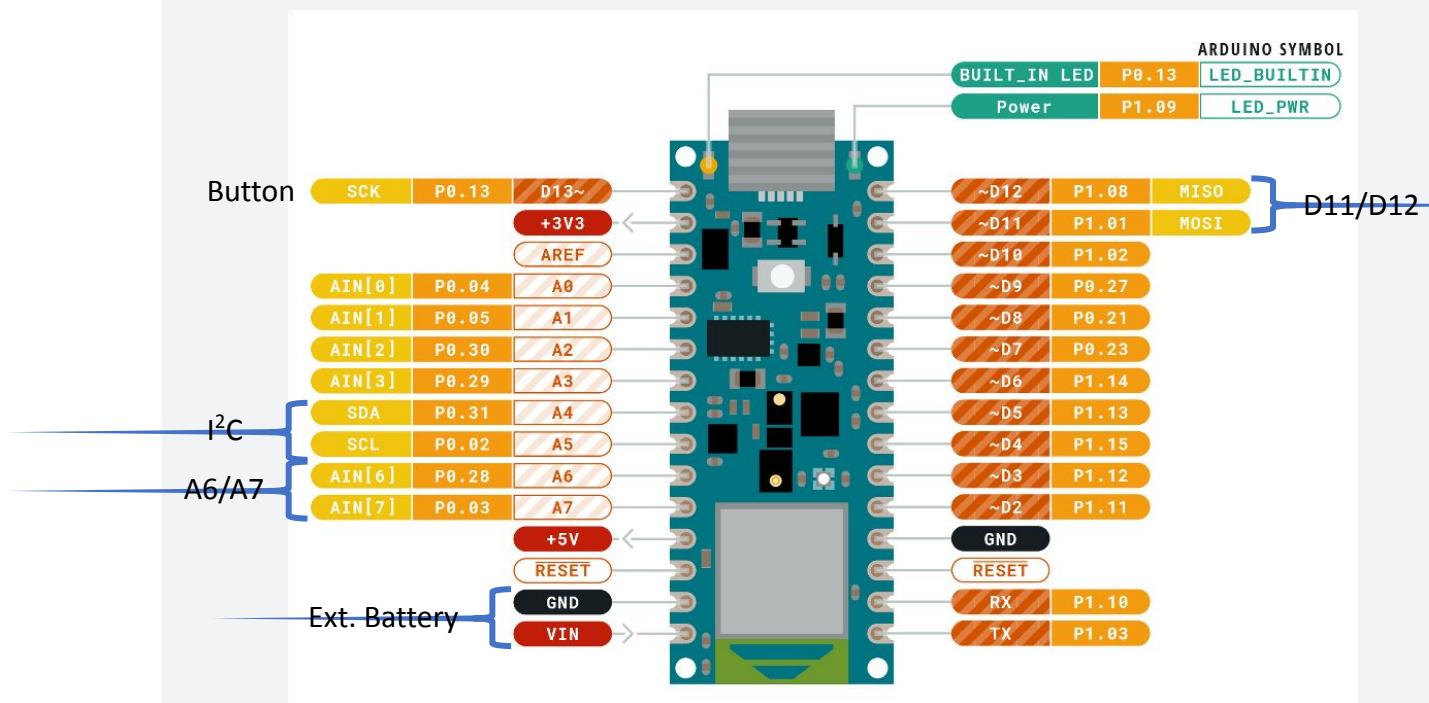
Specifications

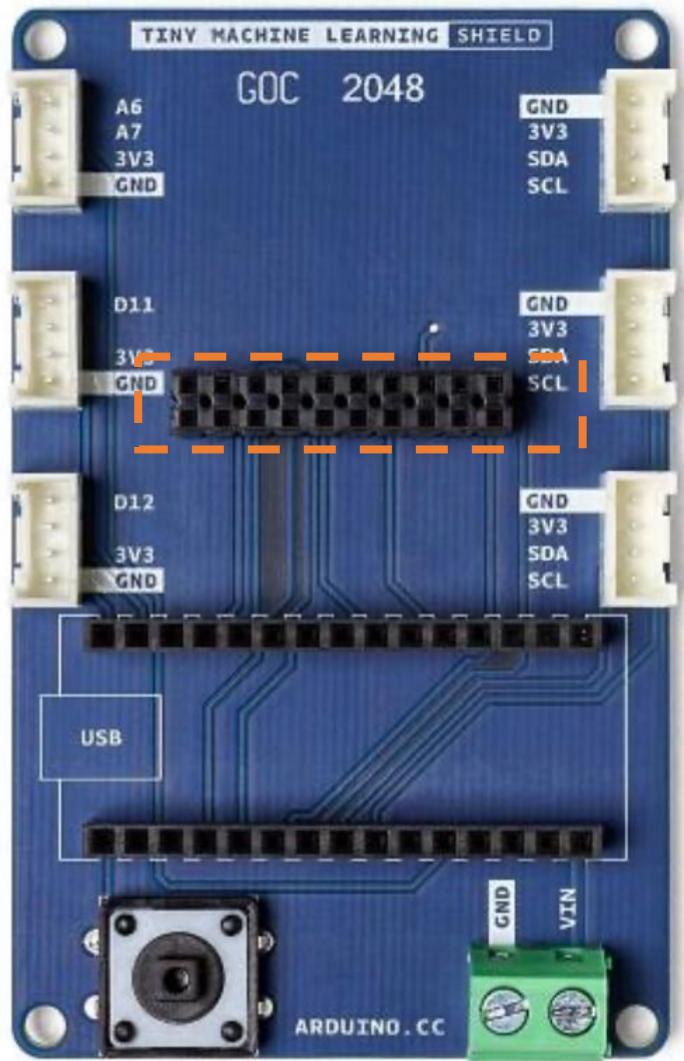
- Grove connectors (3.3V I2C and simple digital / analog - see pinouts)
- 2x10 pin array for OV7675 camera module
- Voltage input terminal block, accepts 4.5 to 21V (down regulated to 3.3V on Nano 33)



TinyML Shield

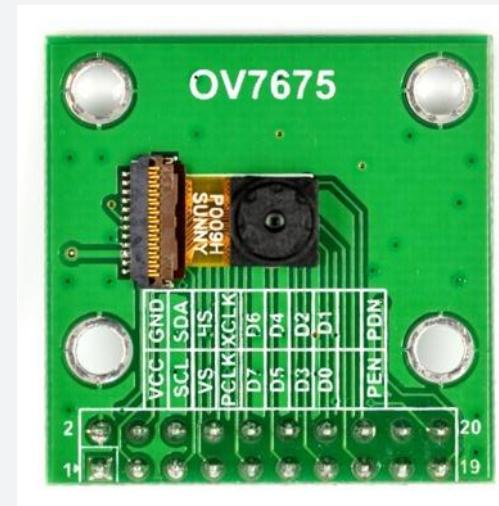
Two rows of 1x15 headers
that you can slot the Nano
33 BLE sense into



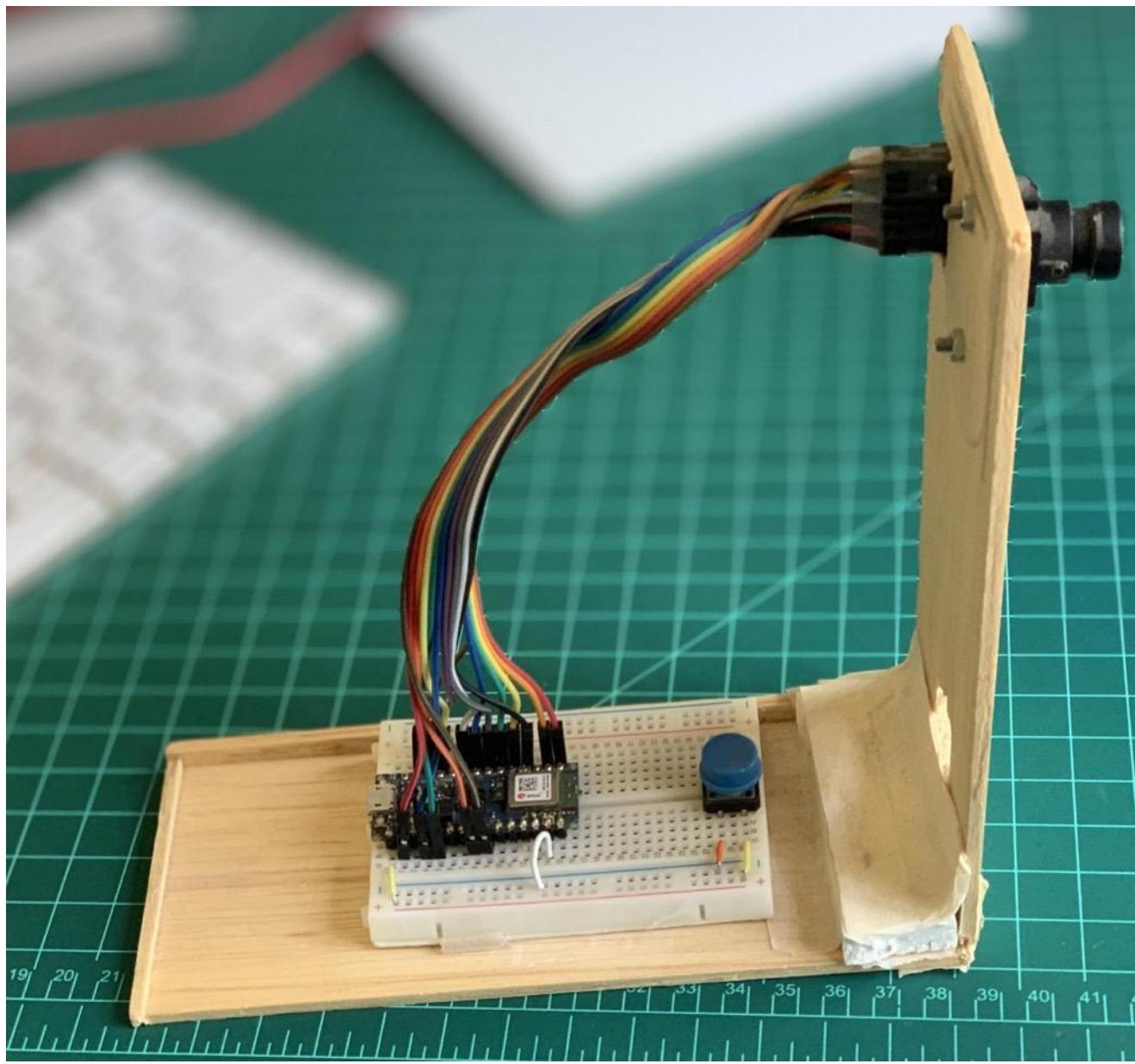


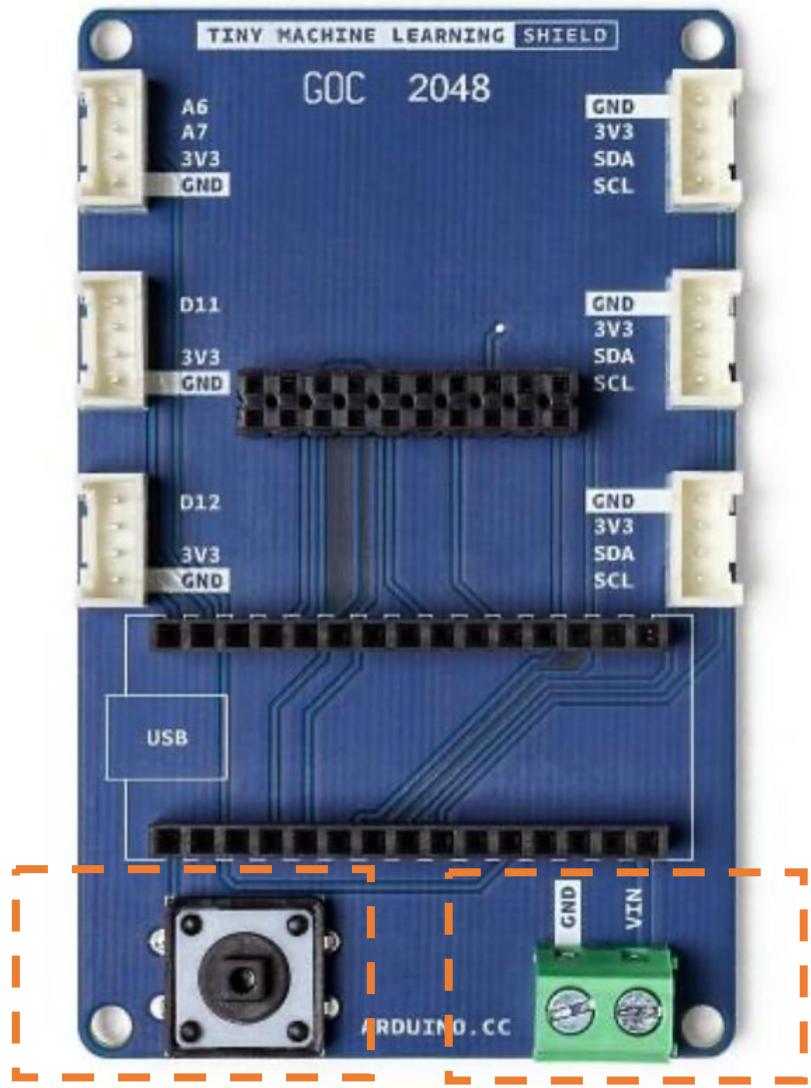
TinyML Shield

2x10 header that is intended to receive the corresponding pins of the OV7675 camera module



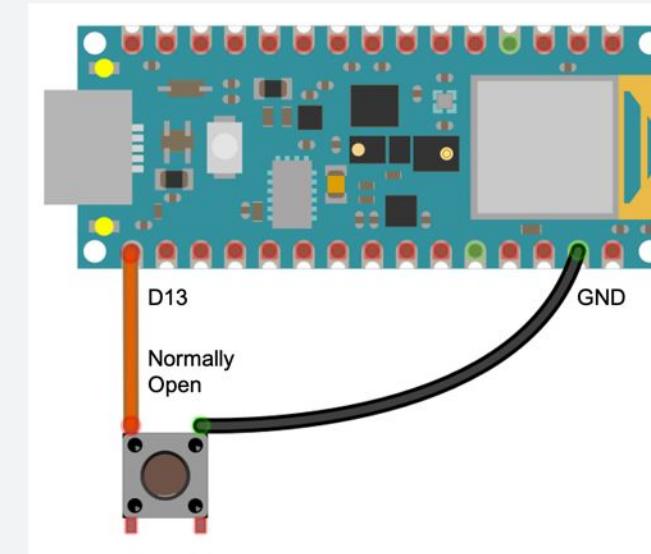
OV7670_VSYNC	8
OV7670_HREF	A1
OV7670_PLK	A0
OV7670_XCLK	9
OV7670_D0	10
OV7670_D1	1
OV7670_D2	0
OV7670_D3	2
OV7670_D4	3
OV7670_D5	5
OV7670_D6	6
OV7670_D7	4



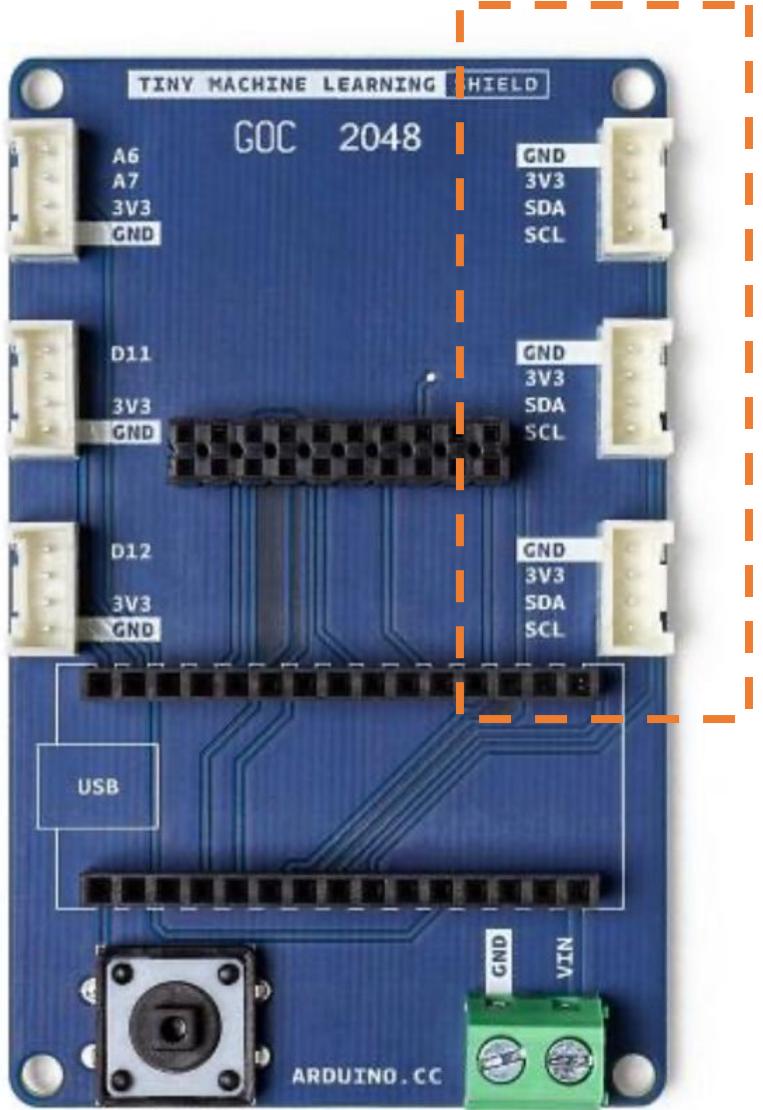


TinyML Shield

A easily programmable
button on the left

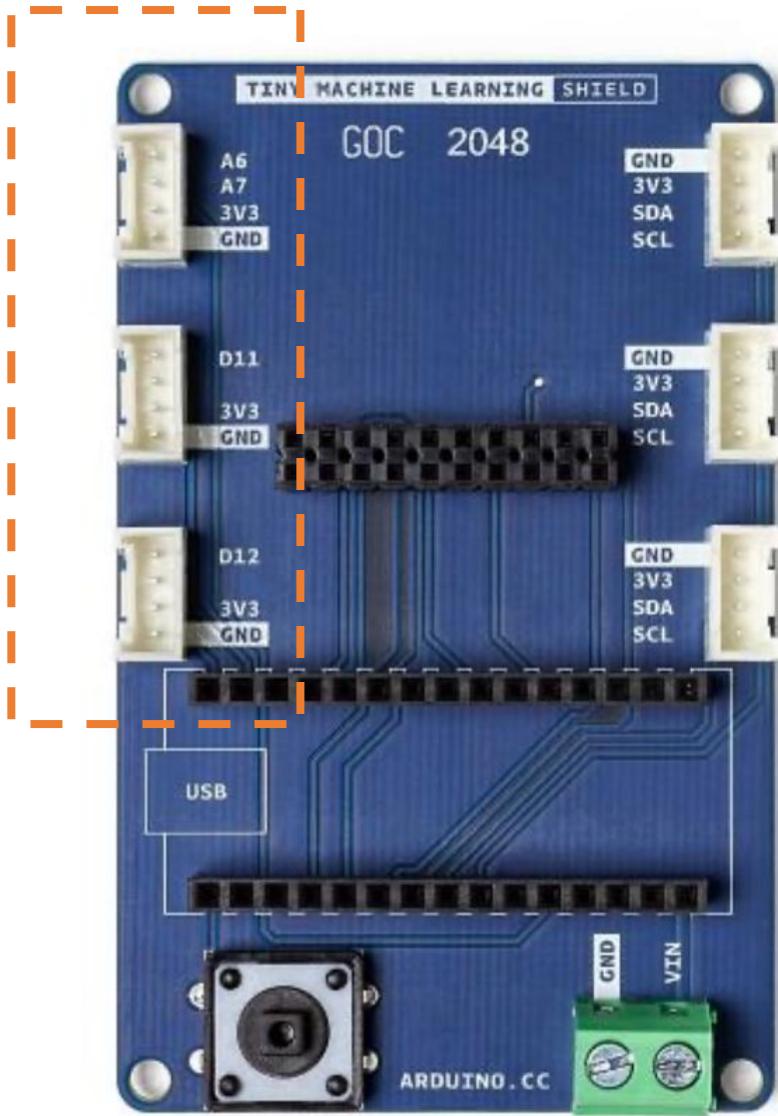


Screw-in terminal block for
external (battery) power (4.5V to 21V)



TinyML Shield

Standard Grove
connectors, to permit
serial communication (I2C
= power + data + clock)
with modules (both
sensors and actuators)



TinyML Shield

Grove connectors that
break out analog and
digital GPIO

Grove Connectors



Purpose

Facilitate **plug-and-play connections** to off-board modules to extend the possible scope of functionality to new **TinyML** applications

Specifications

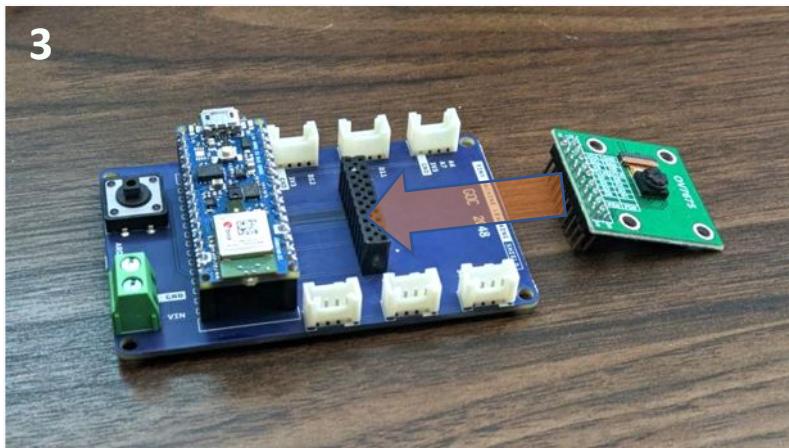
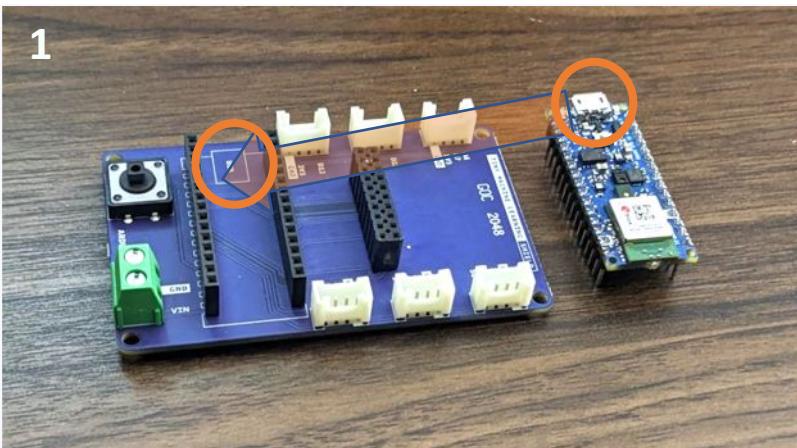
- Proprietary connection system from SeeedStudio, similar to JST PH-type connectors
- Large catalog of sensors, actuators available at seeedstudio.com
- Be sure to check the voltage requirements and pinout of any new Grove module for compatibility with this shield before purchasing or connecting said module

TinyML Kit Installation

- Hardware Set-up
- Software Set-up



Installing the Hardware



Installing the Arduino IDE 2.0

The screenshot shows the Arduino website (arduino.cc) with the "SOFTWARE" tab selected in the navigation bar. The main content area displays the Arduino Web Editor interface, which allows users to code and upload sketches online. Below this, there are two buttons: "CODE ONLINE" and "GETTING STARTED".

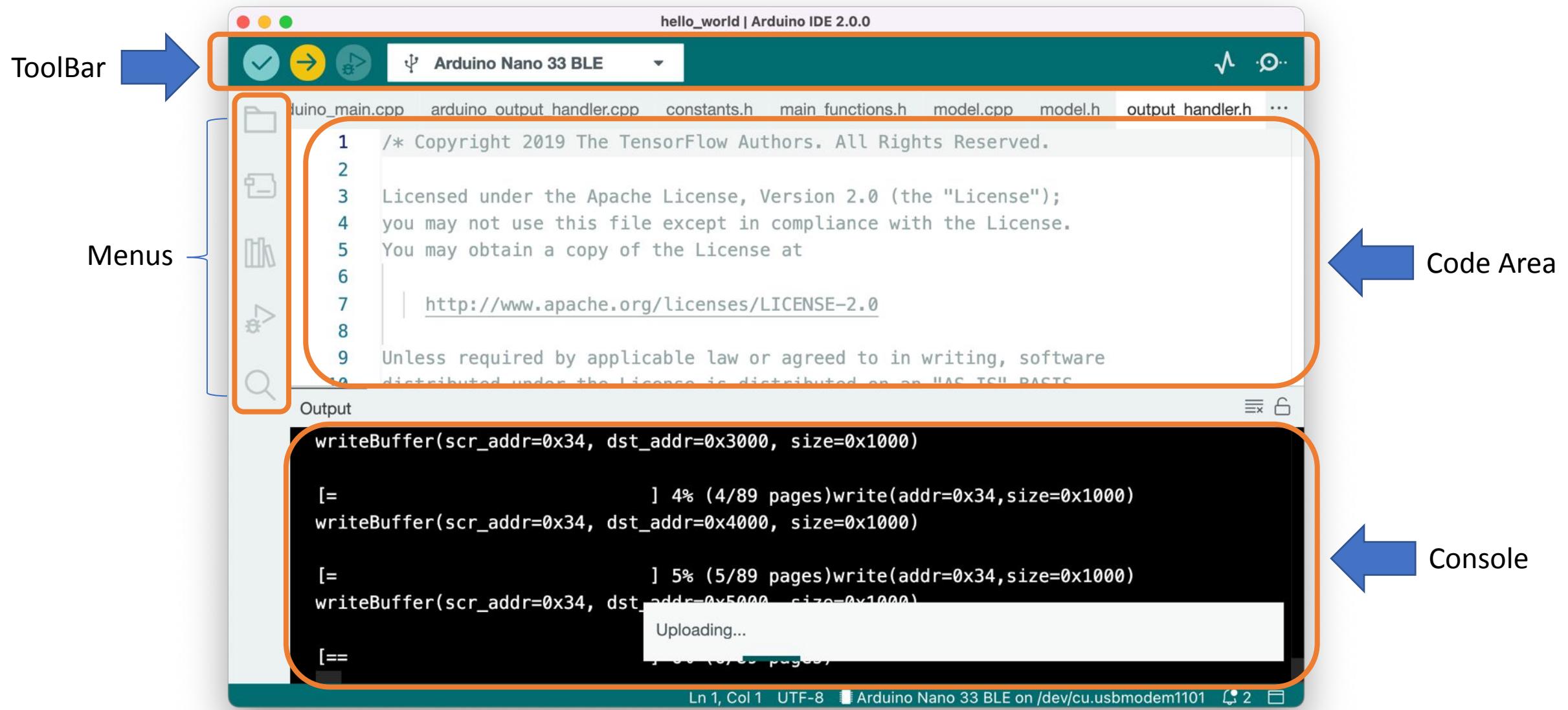
Downloads

The "DOWNLOAD OPTIONS" section is highlighted with an orange rectangle. It lists download links for different operating systems:

- Windows**: Win 10 and newer, 64 bits
- Windows**: MSI installer
- Windows**: ZIP file
- Linux**: AppImage 64 bits (X86-64)
- Linux**: ZIP file 64 bits (X86-64)
- macOS**: 10.14: "Mojave" or newer, 64 bits

At the bottom right of the download section is a "Help" button.

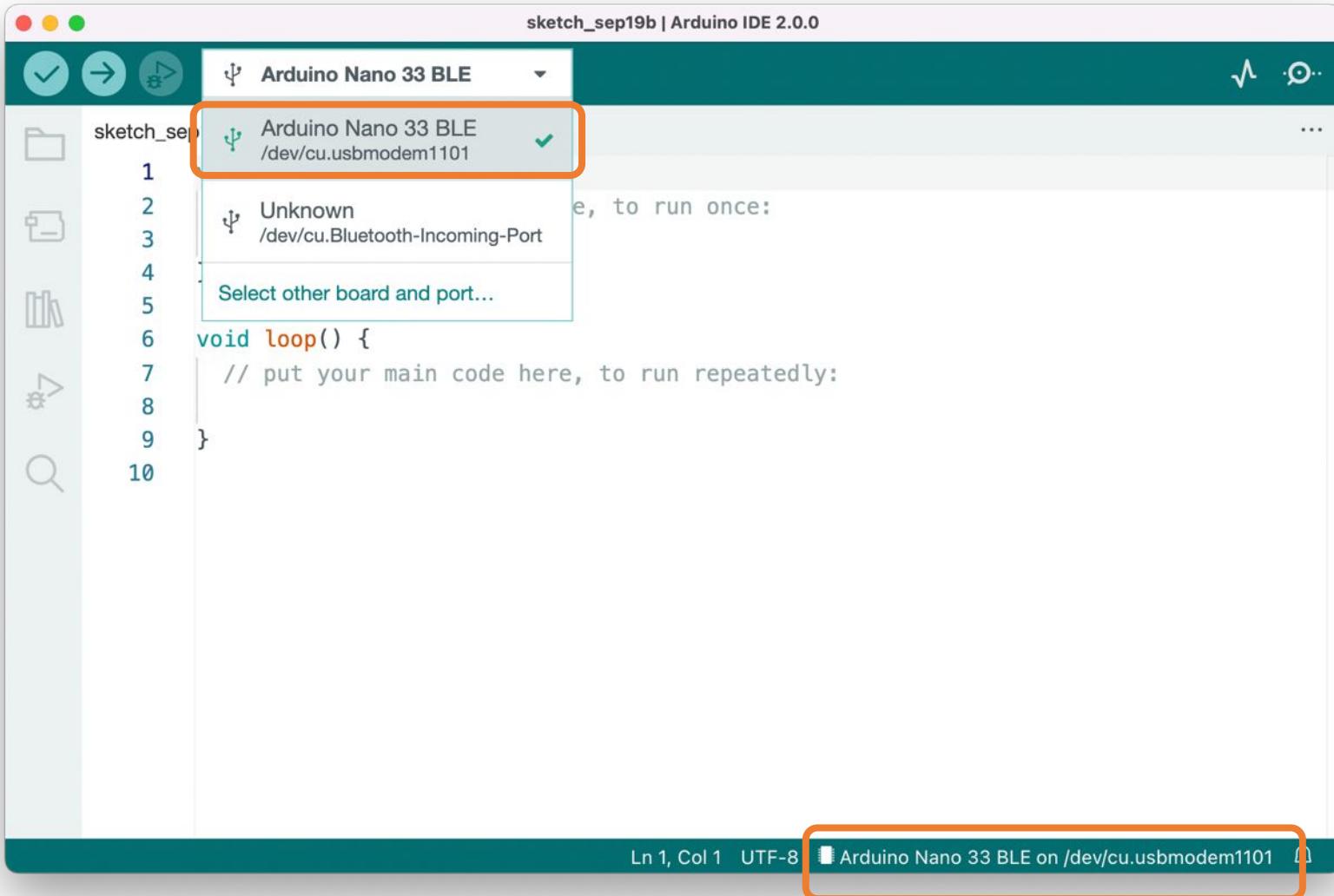
The Arduino IDE 2.0



Installing the Board Files

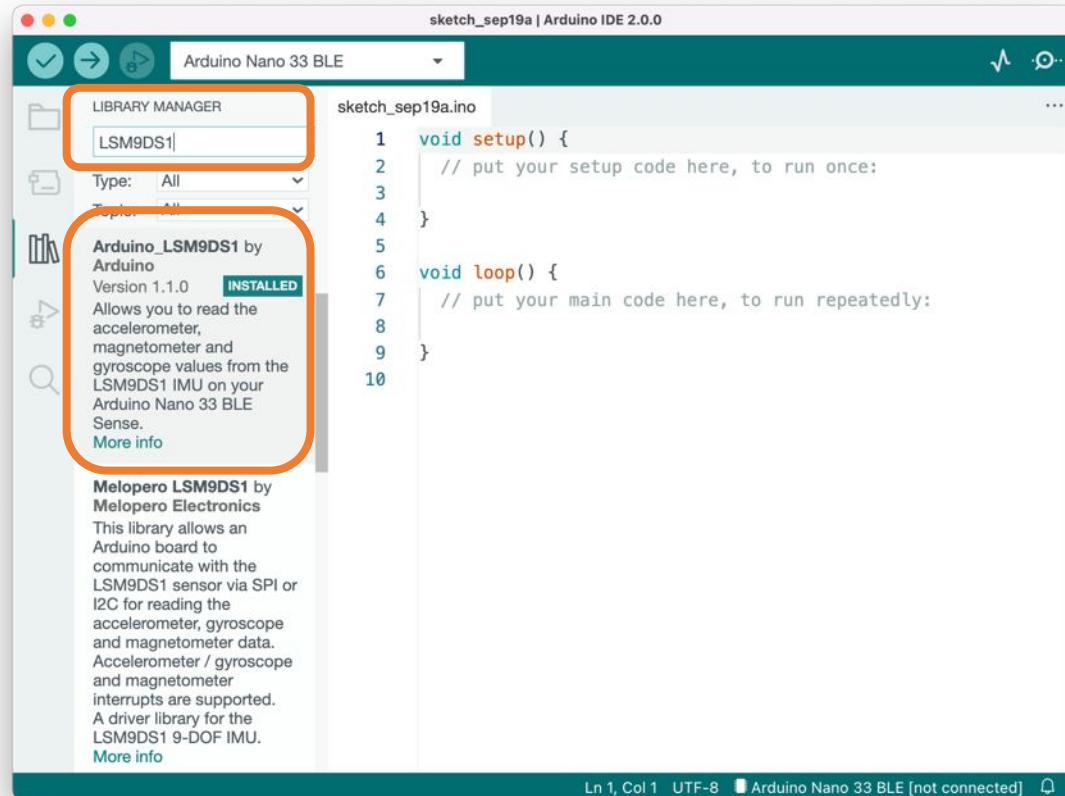


Select Board and Port

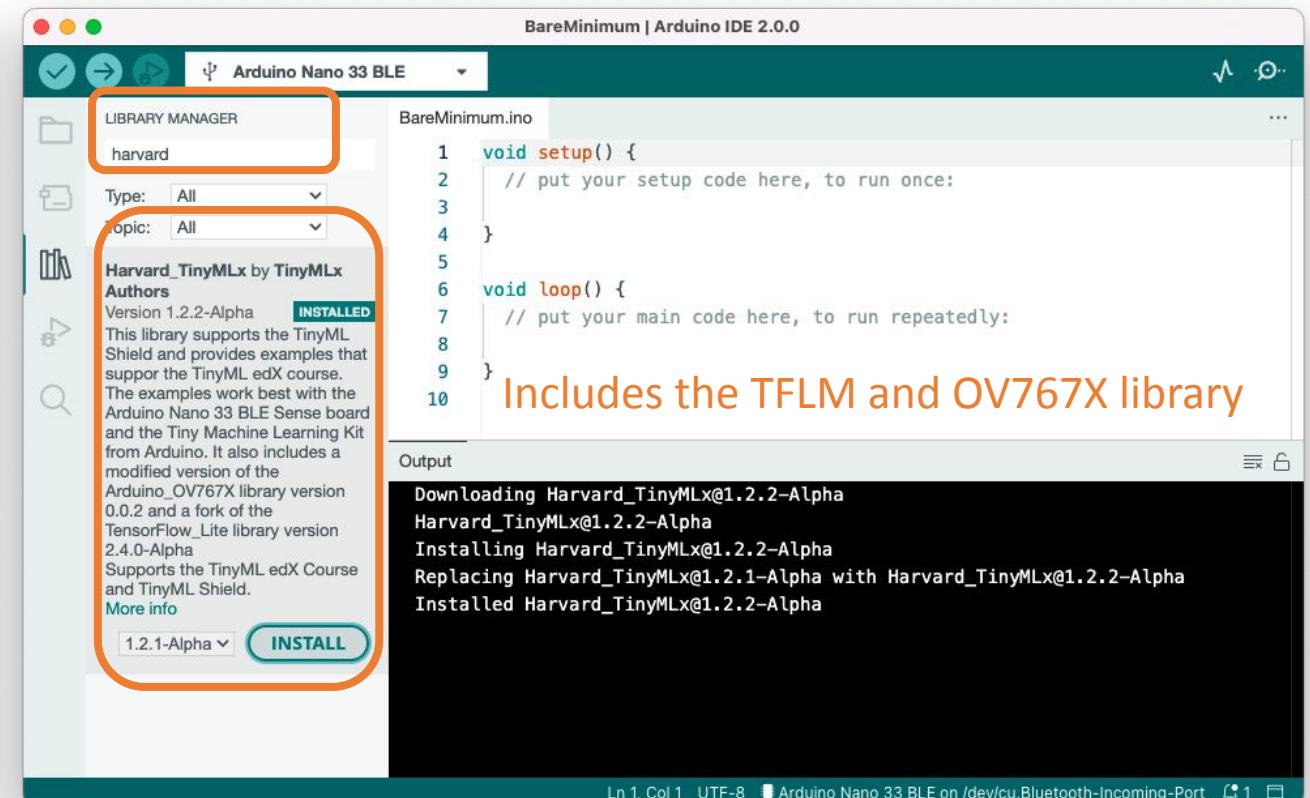


Installing Sensor and Auxilary Libraries

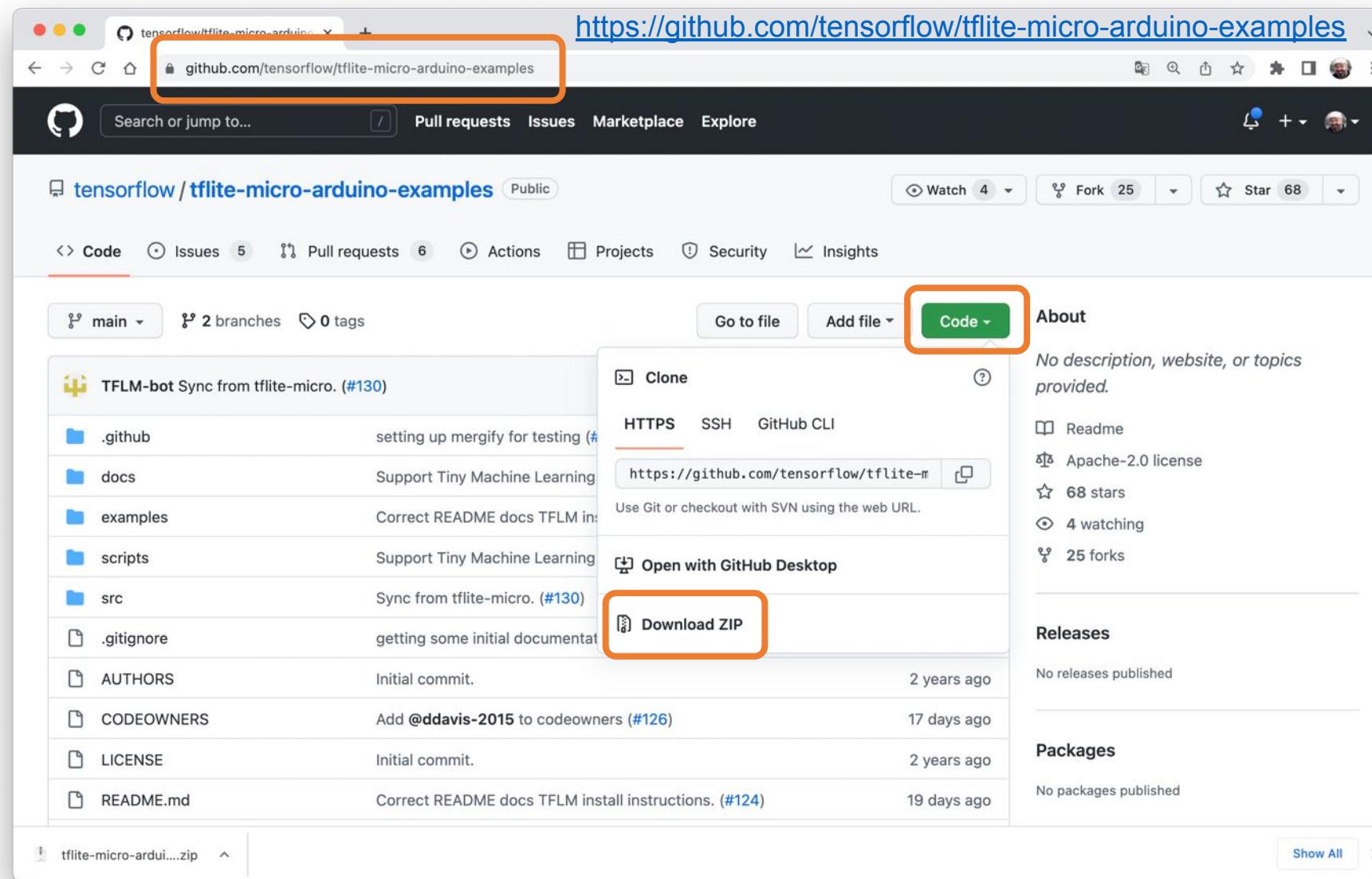
IMU library



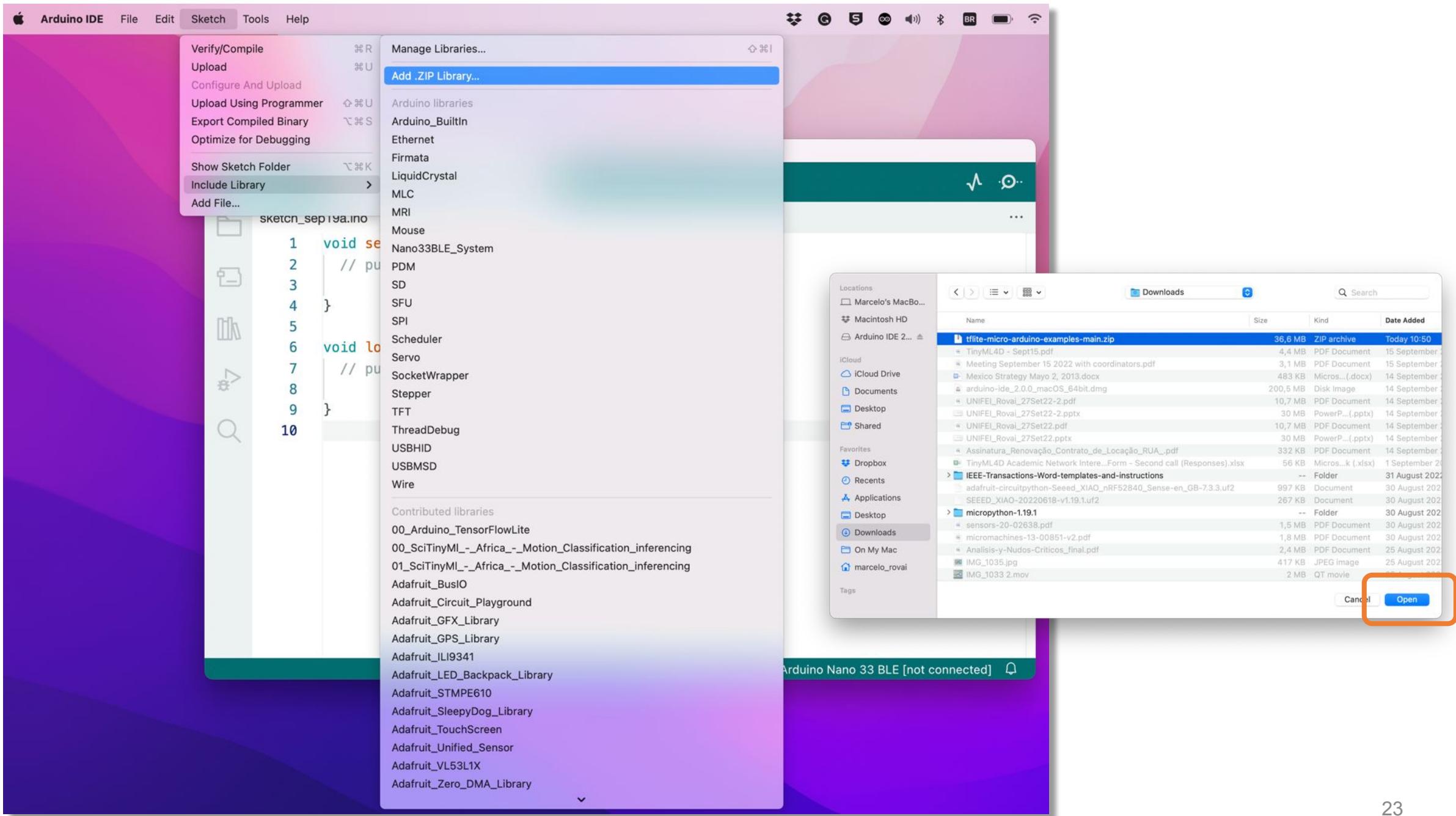
Shield, TFLM and Camera libraries



Installing Tensorflow Lite Micro (TFLM) Library (optional)



If you need install the TFLM library to be used stand alone
The TFLM library is not available directly on the Arduino IDE



The screenshot shows the Arduino IDE 2.0.0 interface with the title bar "sketch_sep19a | Arduino IDE 2.0.0" and the board selected as "Arduino Nano 33 BLE". The code editor contains the following sketch:

```
sketch_sep19a.ino
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here, to run repeatedly:
8
9 }
10
```

A progress bar at the bottom indicates "Processing tflite-micro-arduino-examples-main.zip". The status bar at the bottom shows "Ln 10, Col 1 UTF-8" and "Arduino Nano 33 BLE [not connected]".

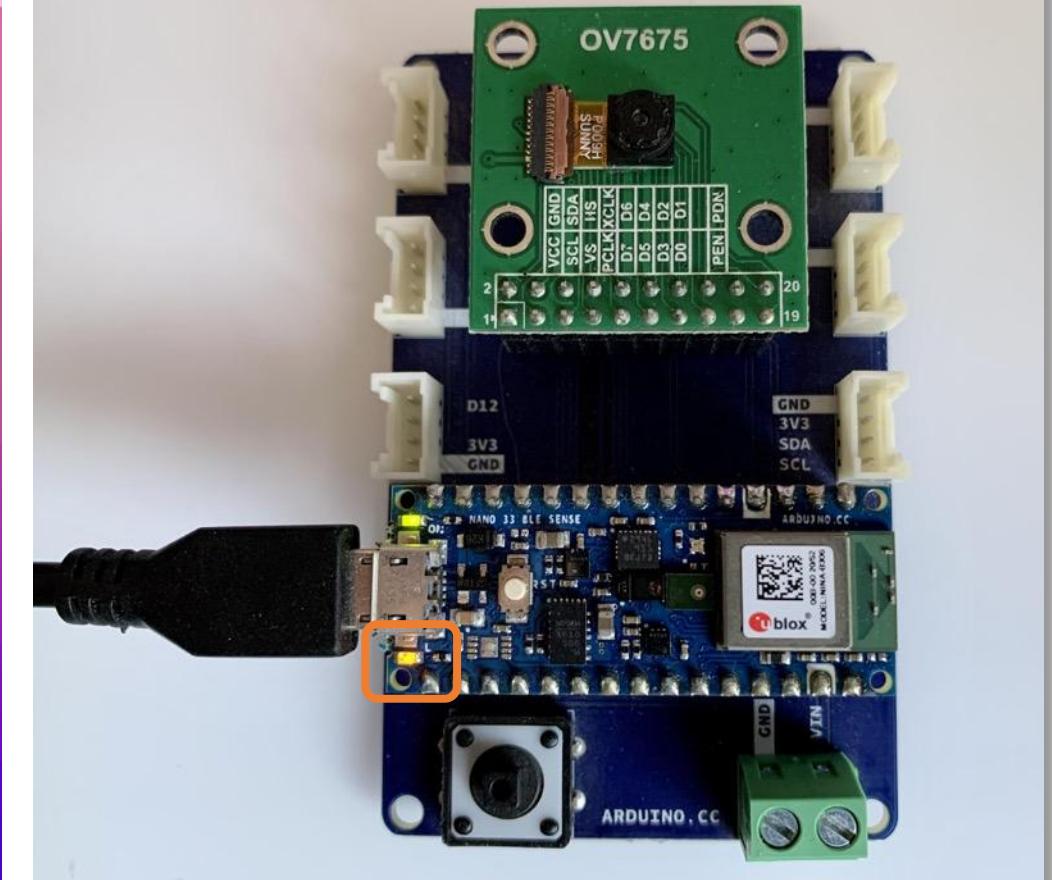
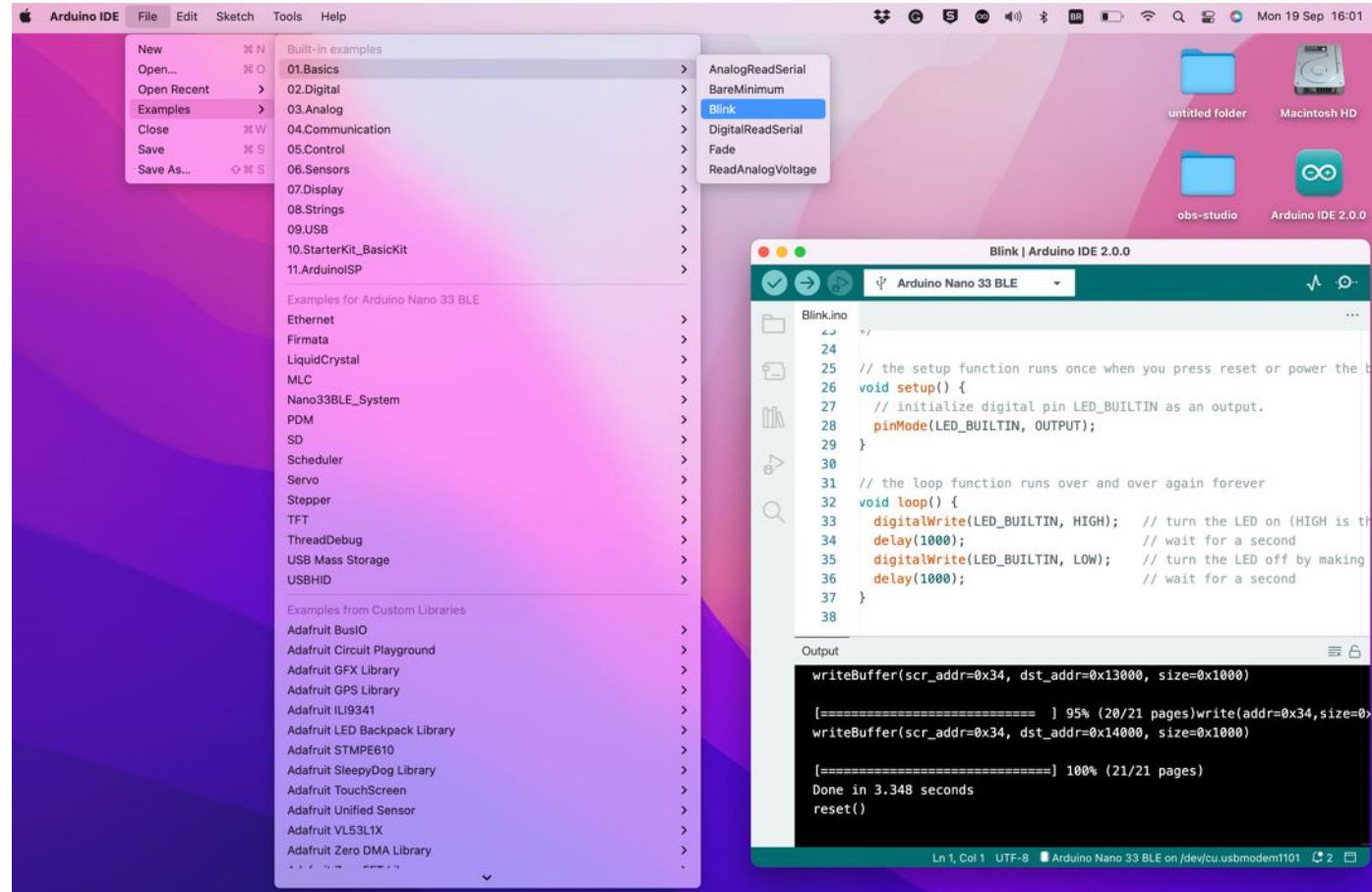
The screenshot shows the Arduino IDE 2.0.0 interface with the title bar "sketch_sep19a | Arduino IDE 2.0.0" and the board selected as "Arduino Nano 33 BLE". The code editor contains the same sketch as the first screenshot. The output window below the code editor displays the message "Library installed". The status bar at the bottom shows "Ln 10, Col 1 UTF-8" and "Arduino Nano 33 BLE [not connected]".

TinyML Kit Test

- MCU test (Blink)
- Sensors Test (IMU, MIC, CAMERA)



MCU installation test (Blink)



Testing IMU

Arduino IDE Menu -> Files/Examples/Harvard_TinyMLx/test_IMU

```
test_IMU.ino
76 if (imuIndex == 0) { // testing accelerometer
77     if (IMU.accelerationAvailable()) {
78         IMU.readAcceleration(x, y, z);
79
80         Serial.print("Ax:");
81         Serial.print(x);
82         Serial.print(',');
83         Serial.print("Ay:");
84         Serial.print(y);
85         Serial.print(',');
86         Serial.print("Az:");
87         Serial.println(z);
88     }
89 }
```

Output Serial Monitor x

Message (⌘ + Enter to send message to 'Arduino Nano 33 BLE' on '/dev/cu.usbmodem1101')

Welcome to the IMU test for the built-in IMU on the Nano 33 BLE Sense

Available commands:

- a - display accelerometer readings in g's in x, y, and z directions
- g - display gyroscope readings in deg/s in x, y, and z directions
- m - display magnetometer readings in uT in x, y, and z directions

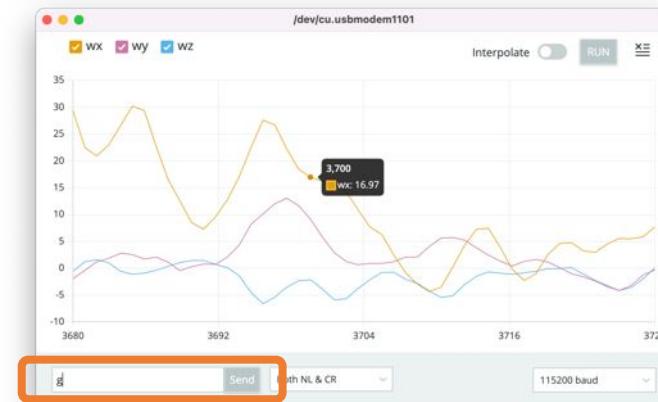
Ax:0.05,Ay:-0.03,Az:0.99
Ax:0.05,Ay:-0.03,Az:0.99

Ln 116, Col 8 UTF-8 Arduino Nano 33 BLE on /dev/cu.usbmodem1101 4 2

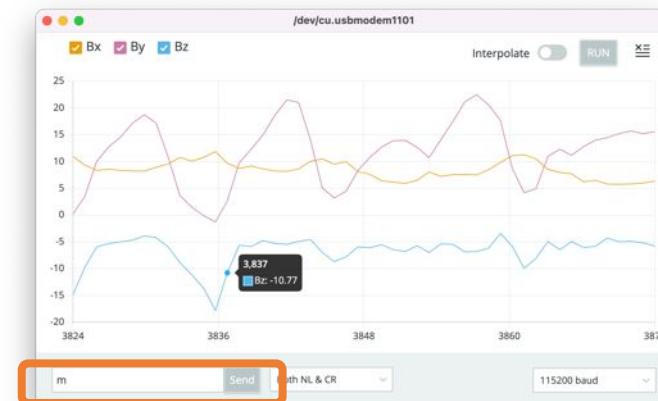
a ->



g ->



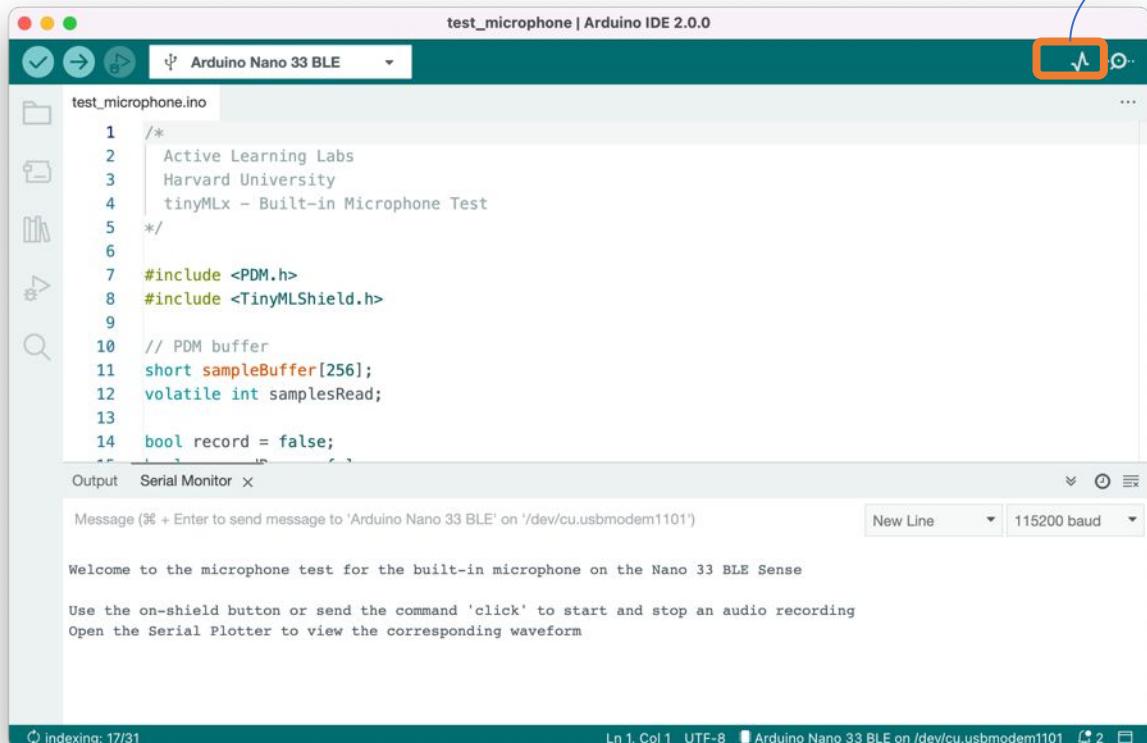
m ->



Note: To be shown simultaneously on Plotter, variables must be separated by coma (Change the sketch)

Testing Microphone

Arduino IDE Menu -> Files/Examples/Harvard_TinyMLx/test_microphone



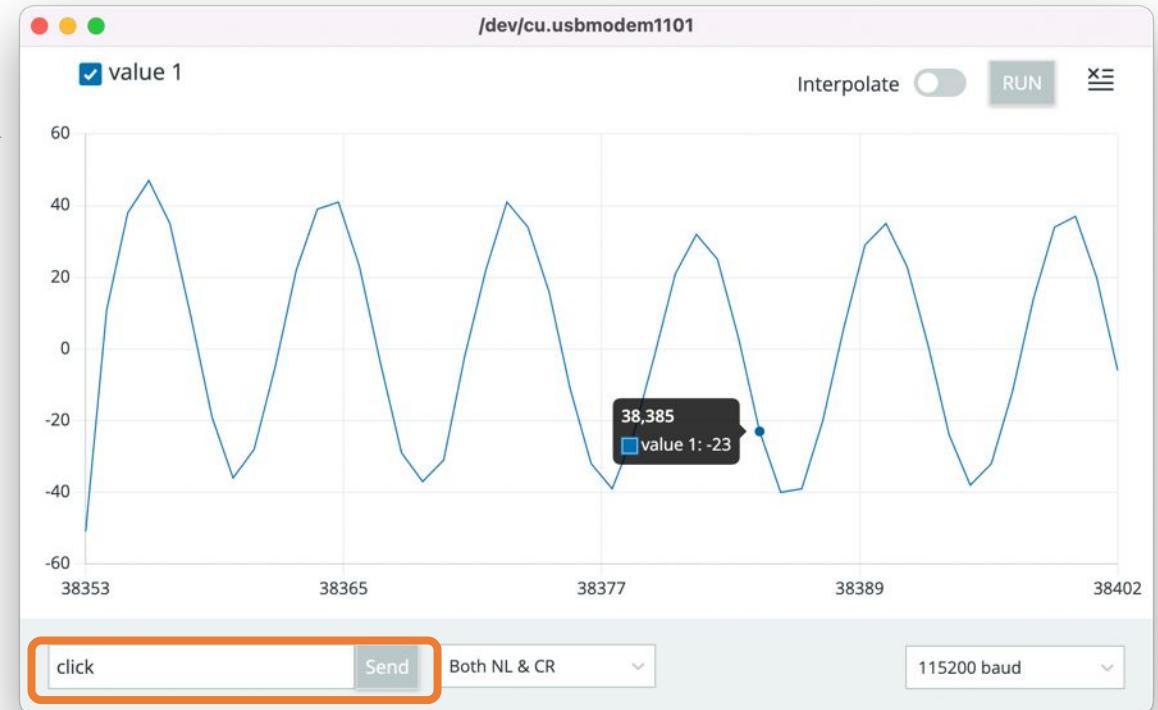
The screenshot shows the Arduino IDE 2.0.0 interface with the file `test_microphone.ino` open. The code is as follows:

```
1  /*  
2   Active Learning Labs  
3   Harvard University  
4   tinyMLx - Built-in Microphone Test  
5 */  
6  
7 #include <PDM.h>  
8 #include <TinyMLShield.h>  
9  
10 // PDM buffer  
11 short sampleBuffer[256];  
12 volatile int samplesRead;  
13  
14 bool record = false;
```

The Serial Monitor window displays the following text:

```
Welcome to the microphone test for the built-in microphone on the Nano 33 BLE Sense  
Use the on-shield button or send the command 'click' to start and stop an audio recording  
Open the Serial Plotter to view the corresponding waveform
```

At the bottom of the Serial Monitor window, there is a message input field with the placeholder "Message (⌘ + Enter to send message to 'Arduino Nano 33 BLE' on '/dev/cu.usbmodem1101')".

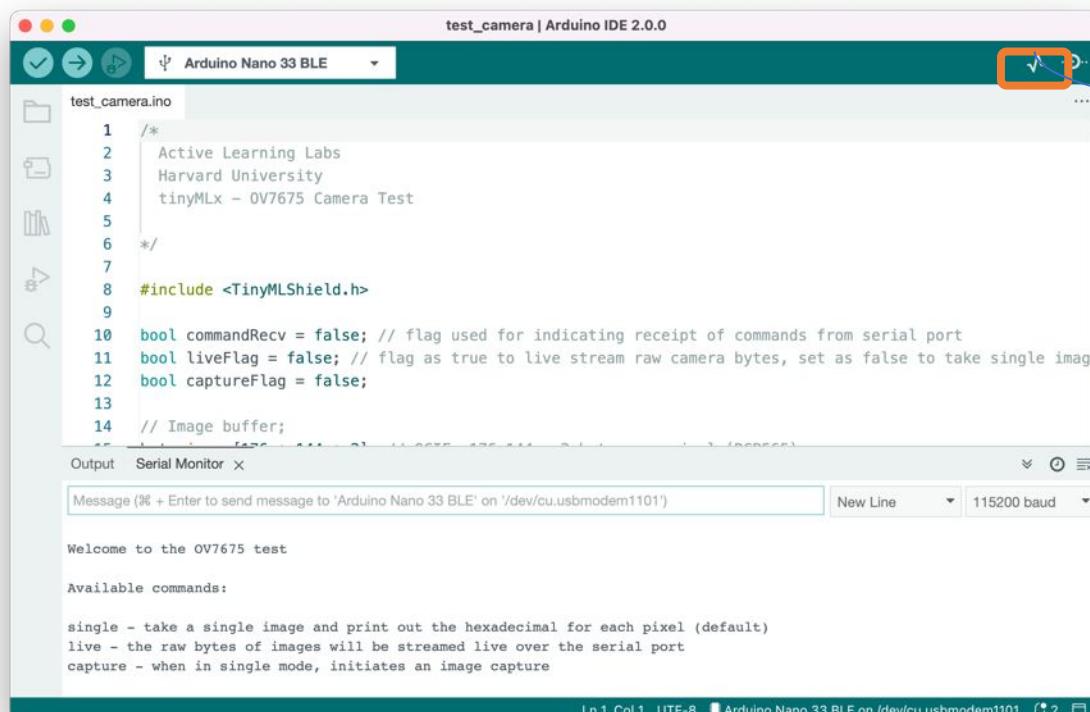


Note: Press the button on shield or send “click” as a command to start/stop sound record



Testing Camera

Arduino IDE Menu -> Files/Examples/Harvard_TinyMLx/test_camera



```
test_camera | Arduino IDE 2.0.0
test_camera.ino
1  /*
2   Active Learning Labs
3   Harvard University
4   tinyMLx - OV7675 Camera Test
5
6 */
7
8 #include <TinyMLShield.h>
9
10 bool commandRecv = false; // flag used for indicating receipt of commands from serial port
11 bool liveFlag = false; // flag as true to live stream raw camera bytes, set as false to take single image
12 bool captureFlag = false;
13
14 // Image buffer;
```

Output Serial Monitor x

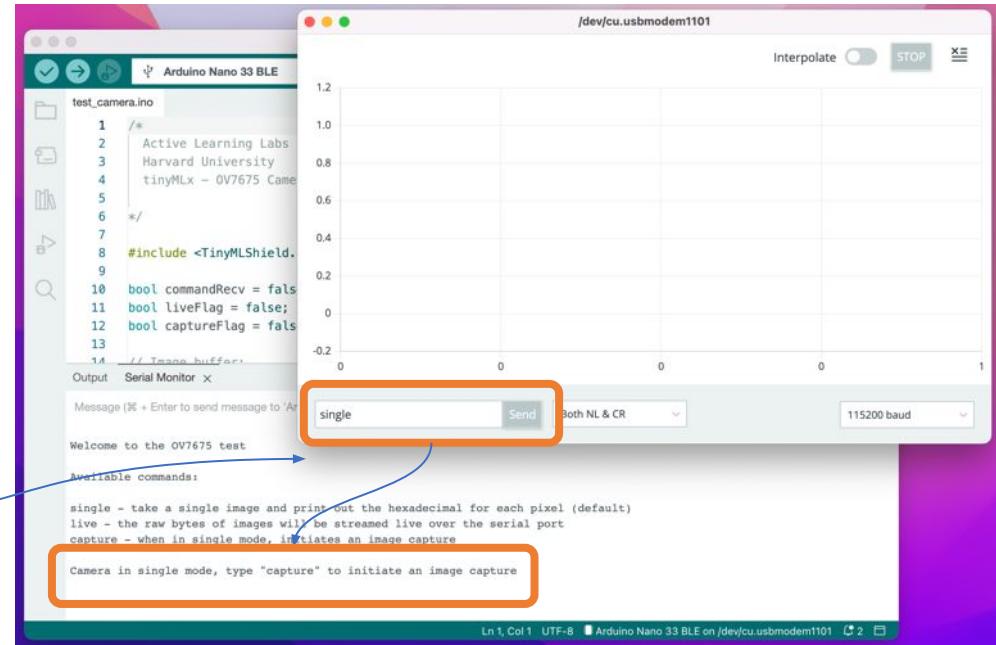
Message (% + Enter to send message to 'Arduino Nano 33 BLE' on '/dev/cu.usbmodem1101') New Line 115200 baud

Welcome to the OV7675 test

Available commands:

single - take a single image and print out the hexadecimal for each pixel (default)
live - the raw bytes of images will be streamed live over the serial port
capture - when in single mode, initiates an image capture

Ln 1, Col 1 UTF-8 ■ Arduino Nano 33 BLE on /dev/cu.usbmodem1101 2



```
Arduino Nano 33 BLE
test_camera.ino
1  /*
2   Active Learning Labs
3   Harvard University
4   tinyMLx - OV7675 Camera
5
6 */
7
8 #include <TinyMLShield.h>
9
10 bool commandRecv = false;
11 bool liveFlag = false;
12 bool captureFlag = false;
13
14 // Image buffer;
```

Output Serial Monitor x

Message (% + Enter to send message to 'Arduino Nano 33 BLE' on '/dev/cu.usbmodem1101') Send Both NL & CR 115200 baud

Welcome to the OV7675 test

Available commands:

single - take a single image and print out the hexadecimal for each pixel (default)
live - the raw bytes of images will be streamed live over the serial port
capture - when in single mode, initiates an image capture

Camera in single mode, type "capture" to initiate an image capture

Image data will be printed out in 3 seconds...
0x2E6B, 0x2D6B, 0xD6B, 0x2D6B, 0x2D6B, 0xD6B, 0x2D6B, 0xD6B, 0xEC62, 0xCC5A, 0xAB5A, 0x8B52, 0xAA5A, 0xAB5A, 0xAB5A, 0x8B5

Ln 1, Col 1 UTF-8 ■ Arduino Nano 33 BLE on /dev/cu.usbmodem1101 2

Note: Press the button on shield or send “capture” as a command to capture a image



Testing Camera



```
Output Serial Monitor X

Message (% + Enter to send message to 'Arduino Nano 33 BLE' on '/dev/cu.usbmodem1101')
New Line ▾ 115200

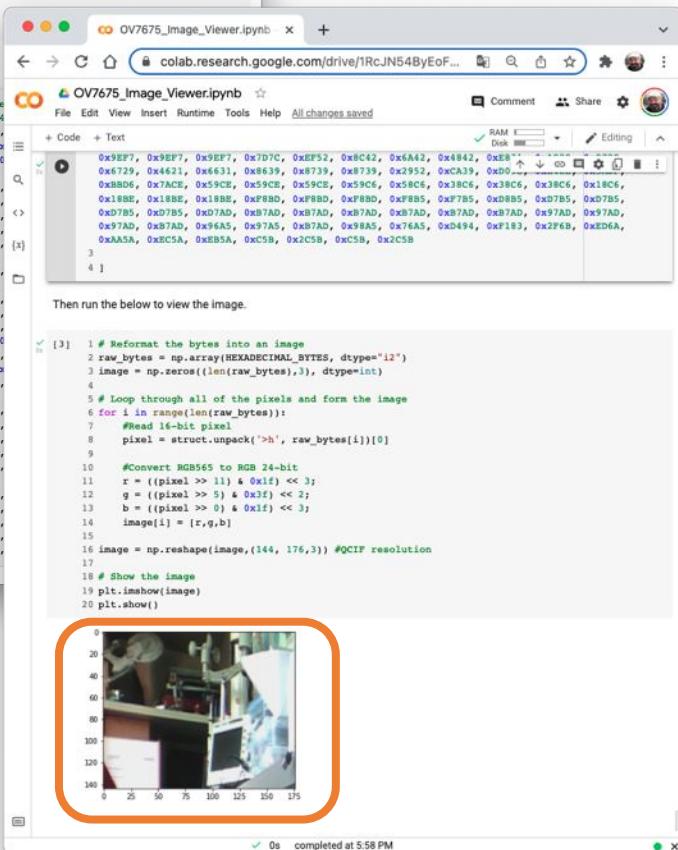
Welcome to the OV7675 test

Available commands:

single - take a single image and print out the hexadecimal for each pixel (default)
live - the raw bytes of images will be streamed live over the serial port
capture - when in single mode, initiates an image capture

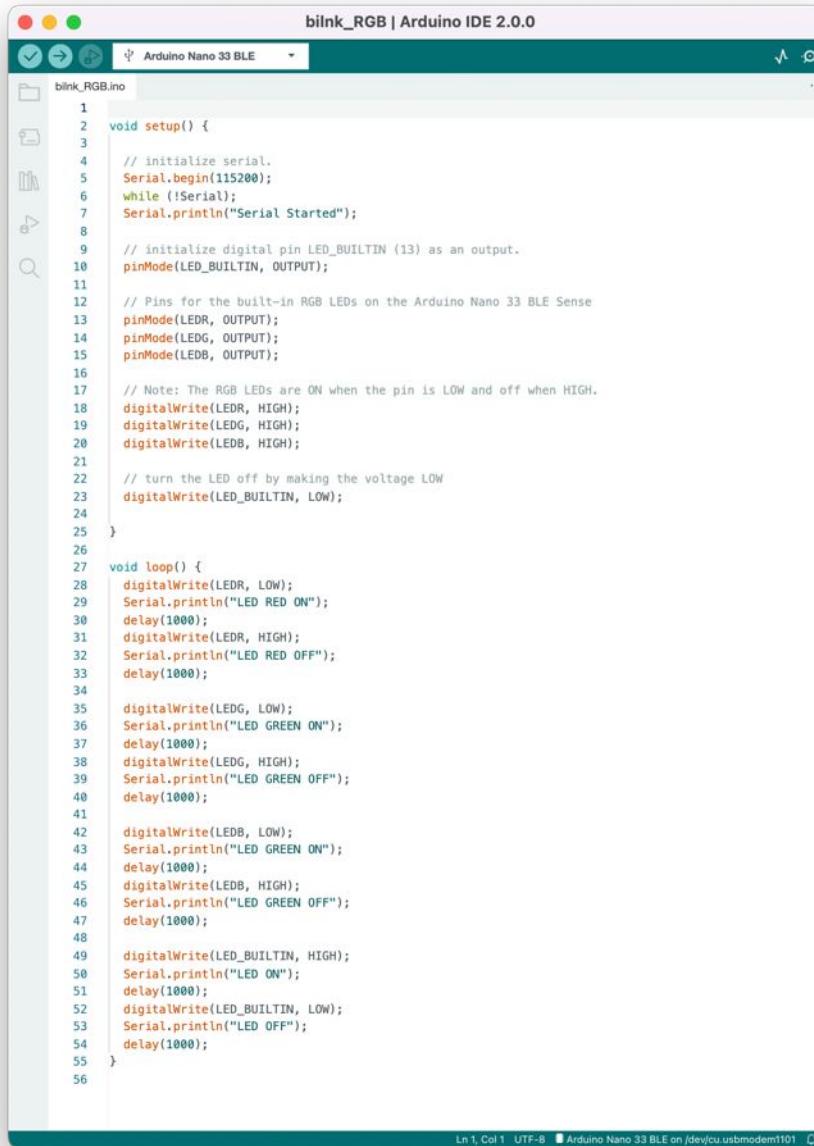
Camera in single mode, type "capture" to initiate an image capture

Image data will be printed out in 3 seconds...
0x2E6B, 0x2D6B, 0xD6B, 0x2D6B, 0x2D6B, 0xD6B, 0x2D6B, 0xD6B, 0xEC62, 0xCC5A, 0xAB5A, 0x8B52, 0xAA5A, 0xAB5A, 0xAB5A, 0x
```



OV7675 Image Viewer.ipynb

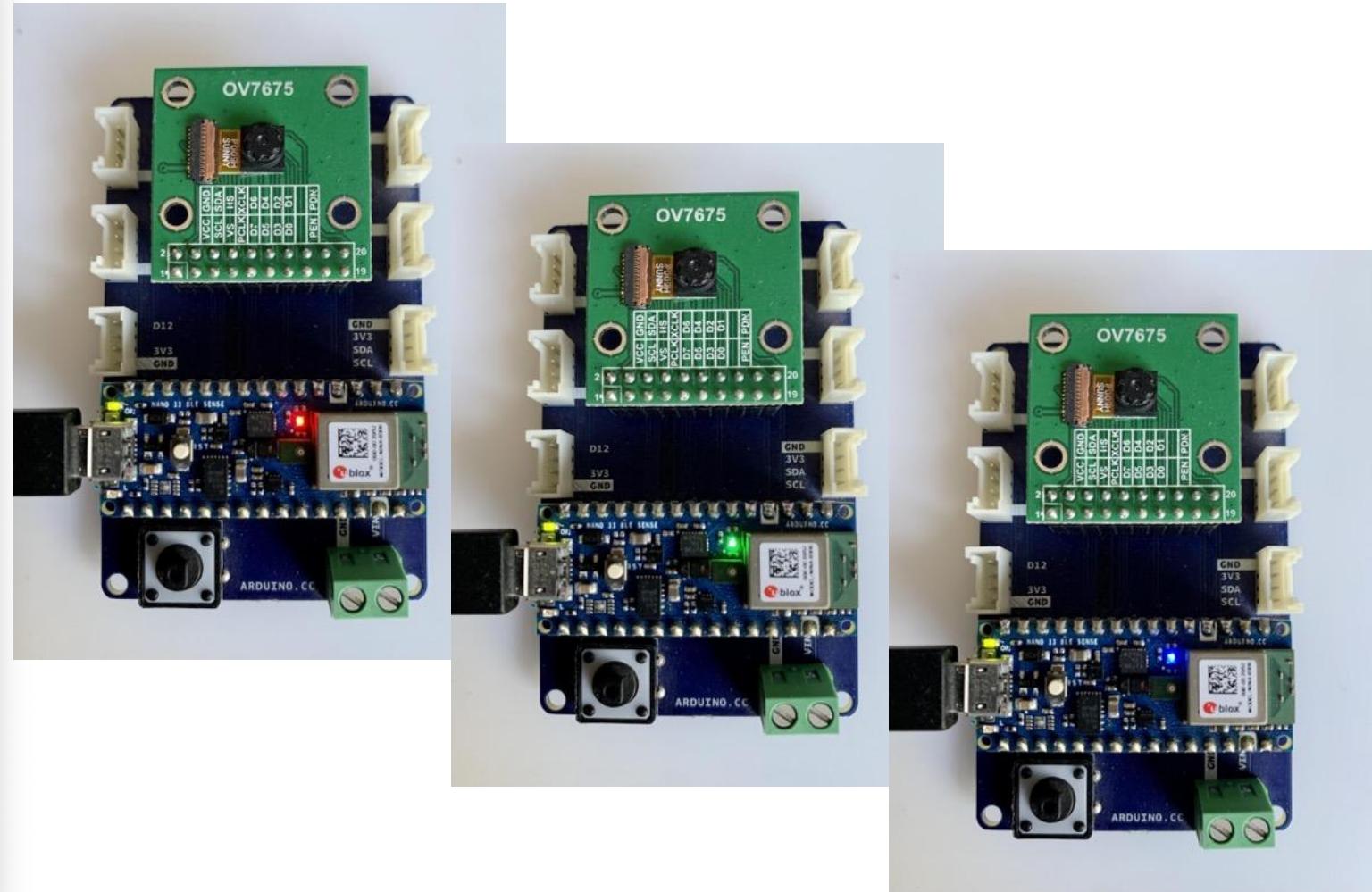
Optional Tests (RGB LEDs)



The screenshot shows the Arduino IDE 2.0.0 interface with a project titled "blink_RGB". The code is as follows:

```
1 void setup() {
2     // initialize serial.
3     Serial.begin(115200);
4     while (!Serial);
5     Serial.println("Serial Started");
6
7     // initialize digital pin LED_BUILTIN (13) as an output.
8     pinMode(LED_BUILTIN, OUTPUT);
9
10    // Pins for the built-in RGB LEDs on the Arduino Nano 33 BLE Sense
11    pinMode(LED_R, OUTPUT);
12    pinMode(LED_G, OUTPUT);
13    pinMode(LED_B, OUTPUT);
14
15    // Note: The RGB LEDs are ON when the pin is LOW and off when HIGH.
16    digitalWrite(LED_R, HIGH);
17    digitalWrite(LED_G, HIGH);
18    digitalWrite(LED_B, HIGH);
19
20    // turn the LED off by making the voltage LOW
21    digitalWrite(LED_BUILTIN, LOW);
22
23 }
24
25 void loop() {
26     digitalWrite(LED_R, LOW);
27     Serial.println("LED RED ON");
28     delay(1000);
29     digitalWrite(LED_R, HIGH);
30     Serial.println("LED RED OFF");
31     delay(1000);
32
33     digitalWrite(LED_G, LOW);
34     Serial.println("LED GREEN ON");
35     delay(1000);
36     digitalWrite(LED_G, HIGH);
37     Serial.println("LED GREEN OFF");
38     delay(1000);
39
40     digitalWrite(LED_B, LOW);
41     Serial.println("LED GREEN ON");
42     delay(1000);
43     digitalWrite(LED_B, HIGH);
44     Serial.println("LED GREEN OFF");
45     delay(1000);
46
47     digitalWrite(LED_BUILTIN, HIGH);
48     Serial.println("LED ON");
49     delay(1000);
50     digitalWrite(LED_BUILTIN, LOW);
51     Serial.println("LED OFF");
52     delay(1000);
53 }
```

At the bottom of the IDE window, it says "Ln 1, Col 1 UTF-8" and "Arduino Nano 33 BLE on /dev/cu.usbmodem1101".



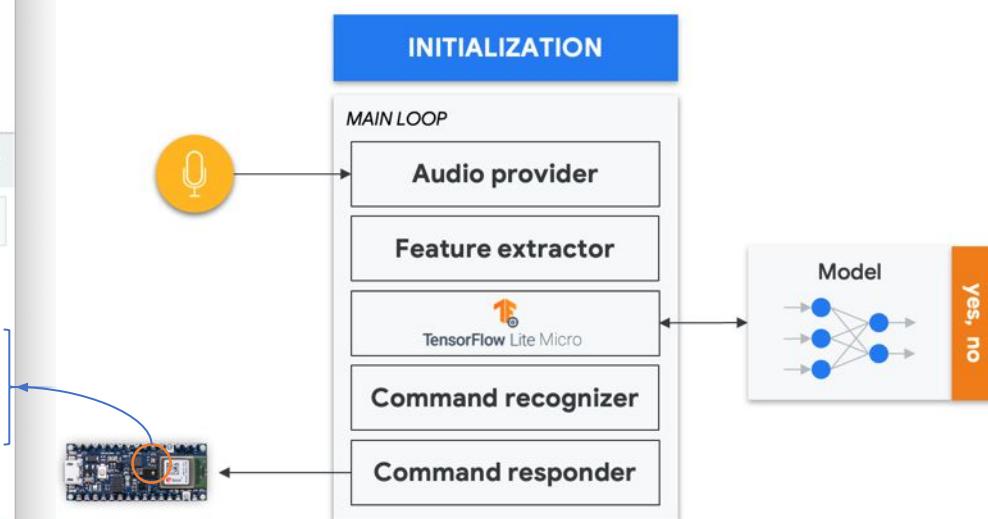
KeyWord Spotting (Optional)

The screenshot shows the Arduino IDE 2.0.0 interface with the project titled "micro_speech". The code editor displays the Apache License 2.0 header. The serial monitor window shows the following output:

```
Heard yes (210) @39952ms
Heard no (211) @45440ms
Heard yes (203) @48832ms
```

Below the serial monitor, there is a status bar with the text: "Ln 1, Col 1 UTF-8" and "Arduino Nano 33 BLE on /dev/cu.usbmodem1101".

You will find the code under:
File > Examples > Harvard_TinyMLx > micro_speech



Reading Material

Main references

- [Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning](#)
- [Professional Certificate in Tiny Machine Learning \(TinyML\) – edX/Harvard](#)
- [Introduction to Embedded Machine Learning - Coursera/Edge Impulse](#)
- [Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse](#)
- Fundamentals textbook: “[Deep Learning with Python](#)” by François Chollet
- Applications & Deploy textbook: “[TinyML](#)” by Pete Warden, Daniel Situnayake
- Deploy textbook “[TinyML Cookbook](#)” by Gian Marco Iodice

I want to thank **Shawn Hymel** and **Edge Impulse**, **Pete Warden** and **Laurence Moroney** from Google, Professor **Vijay Janapa Reddi** and **Brian Plancher** from Harvard, and the rest of the **TinyMLEdu** team for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the [TinyML4D](#), an initiative to make TinyML education available to everyone globally.

Thanks



UNIFEI