

# SciTinyML

Scientific Use of Machine Learning on Low Power Devices

## Regional Workshops

TinyML Kit Overview - HW and SW Installation & Test

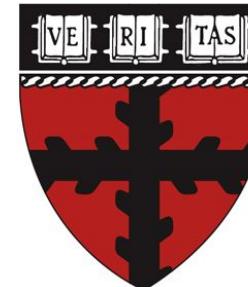
---

Brian Plancher

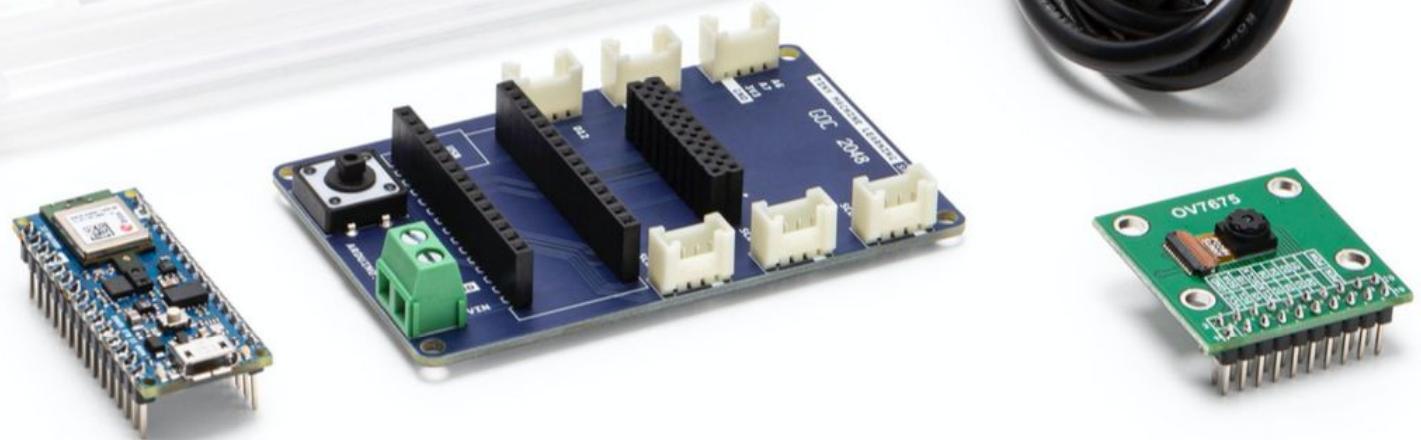
Harvard John A. Paulson School of Engineering and Applied Sciences

Marcelo J. Rovai

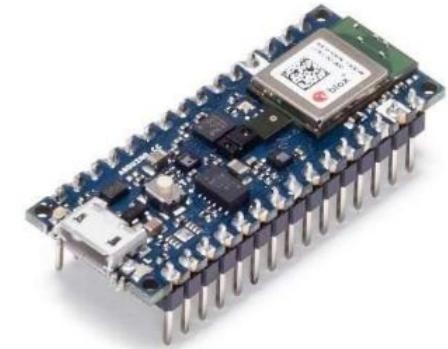
UNIFEI - Federal University of Itajubá, Brazil



# TinyML Kit Overview



# Nano 33 BLE Sense (+ USB cable)

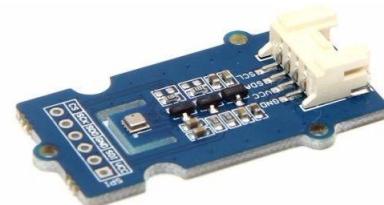


## Purpose

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

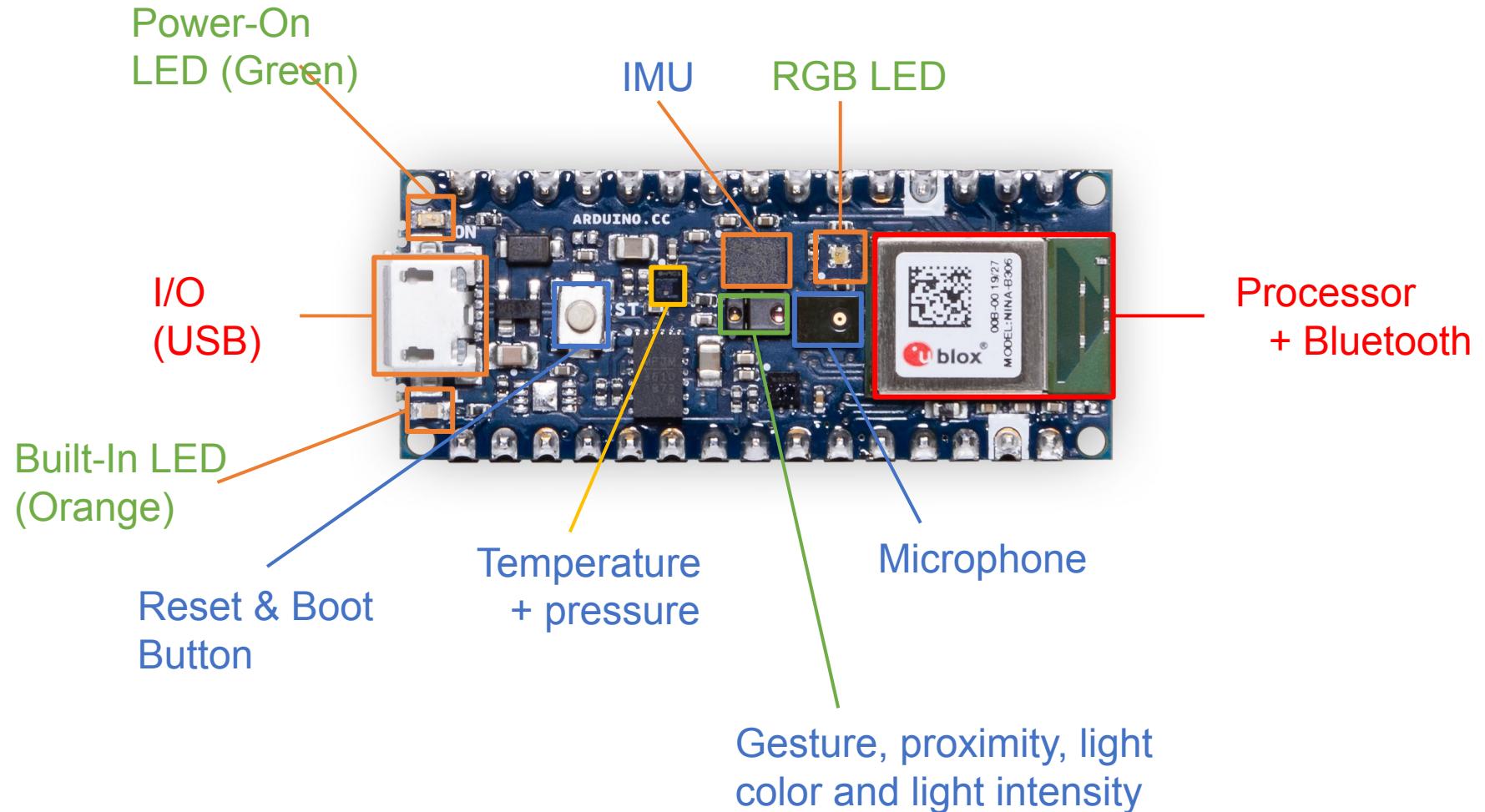
## Specifications

- **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 the package. For projects we will use the external Grove - Temp&Humi&Barometer Sensor (BME280)

# Nano 33 BLE Sense (Development board)



# OV 7675 Camera Module



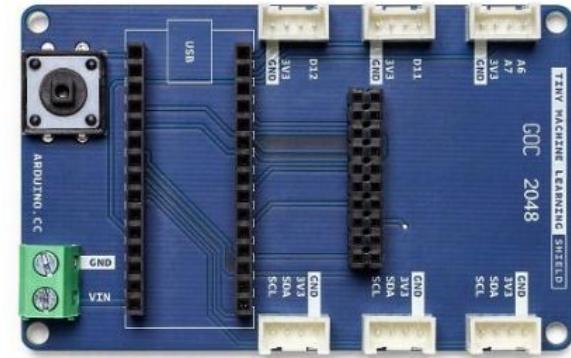
## Purpose

Breakout PCB for *tiny* camera.

## Specifications

- Low-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

# Tiny Machine Learning Shield

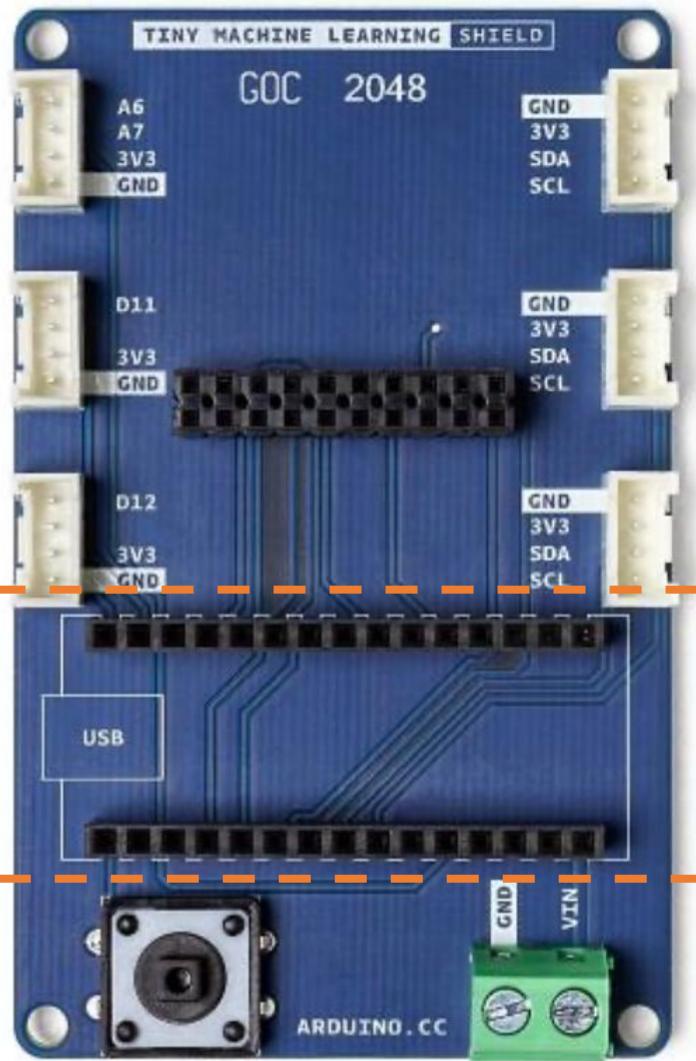


## Purpose

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**

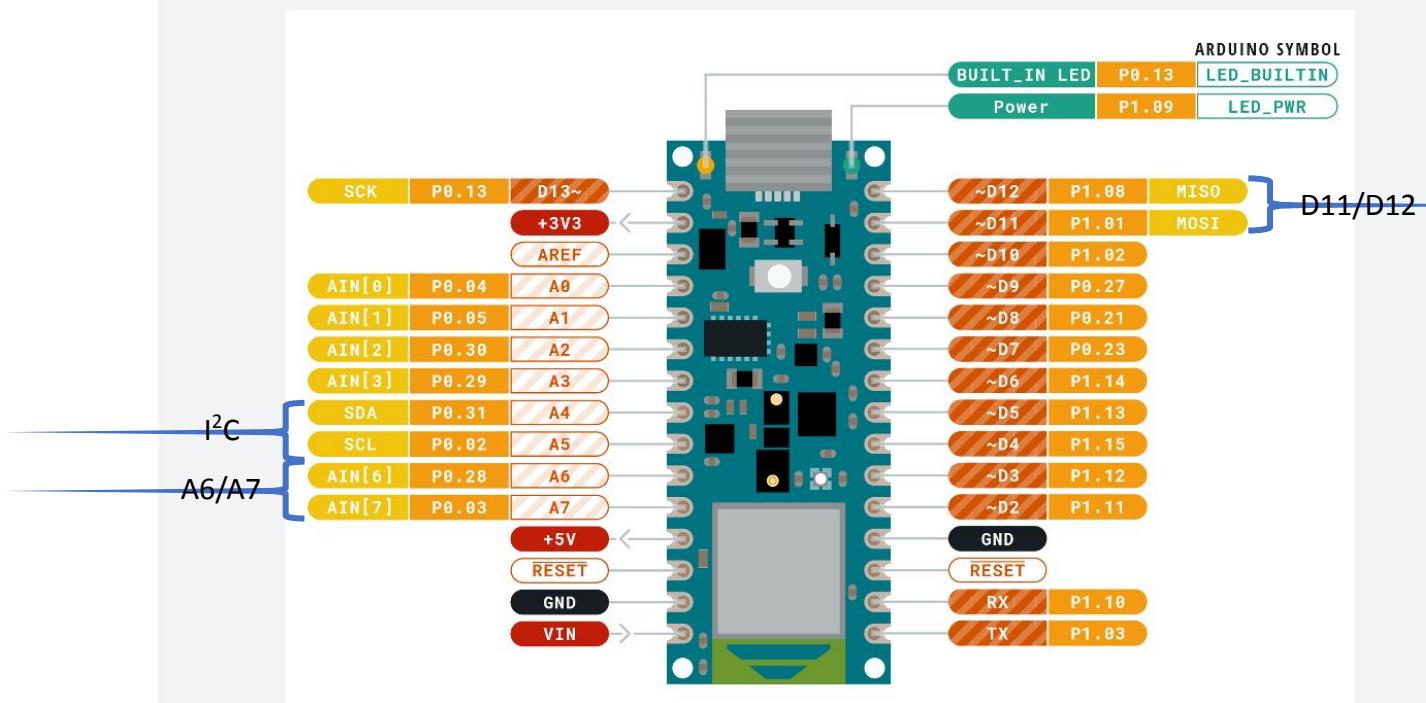
## Specifications

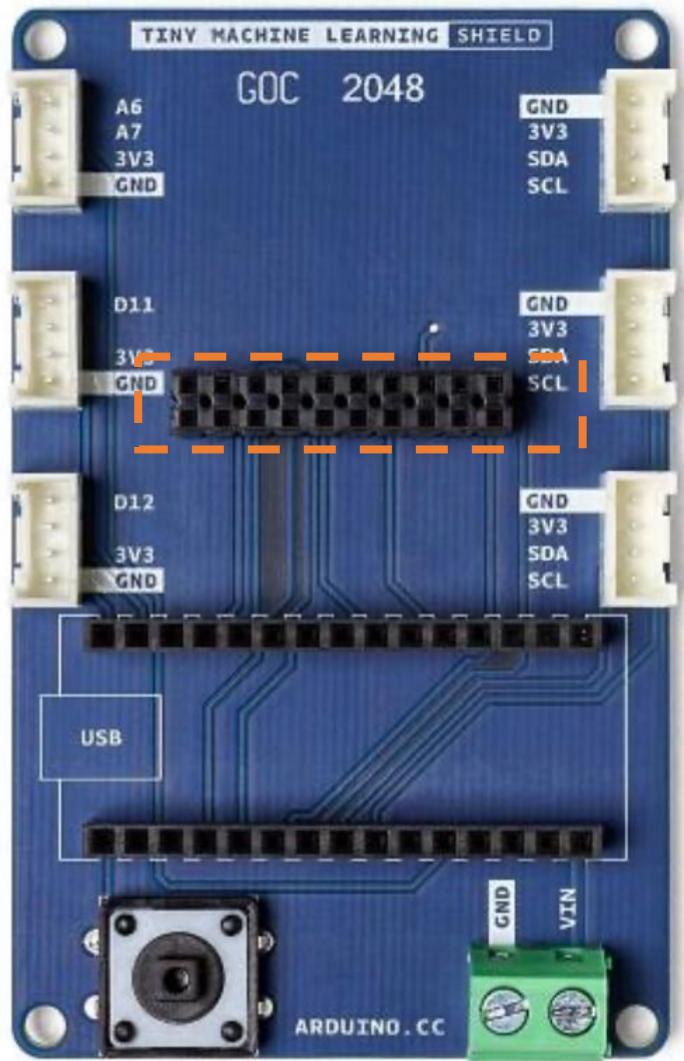
- Grove connectors (3.3V I2C and simple digital / analog - see pinouts)
- 2x10 pin array for OV7675 camera module
- Voltage input terminal block, accepts 4.5 to 21V (down regulated to 3.3V on Nano 33)



# TinyML Shield

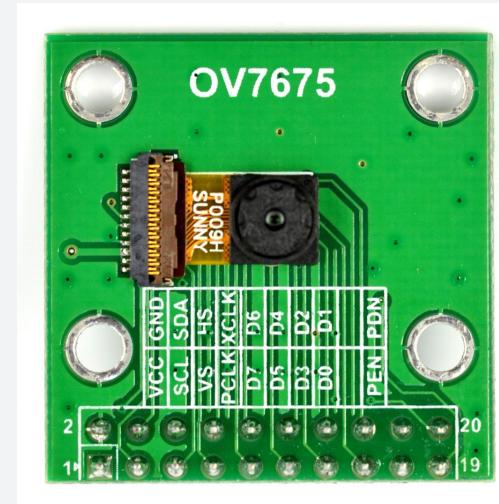
Two rows of 1x15 headers  
that you can slot the Nano  
33 BLE sense into



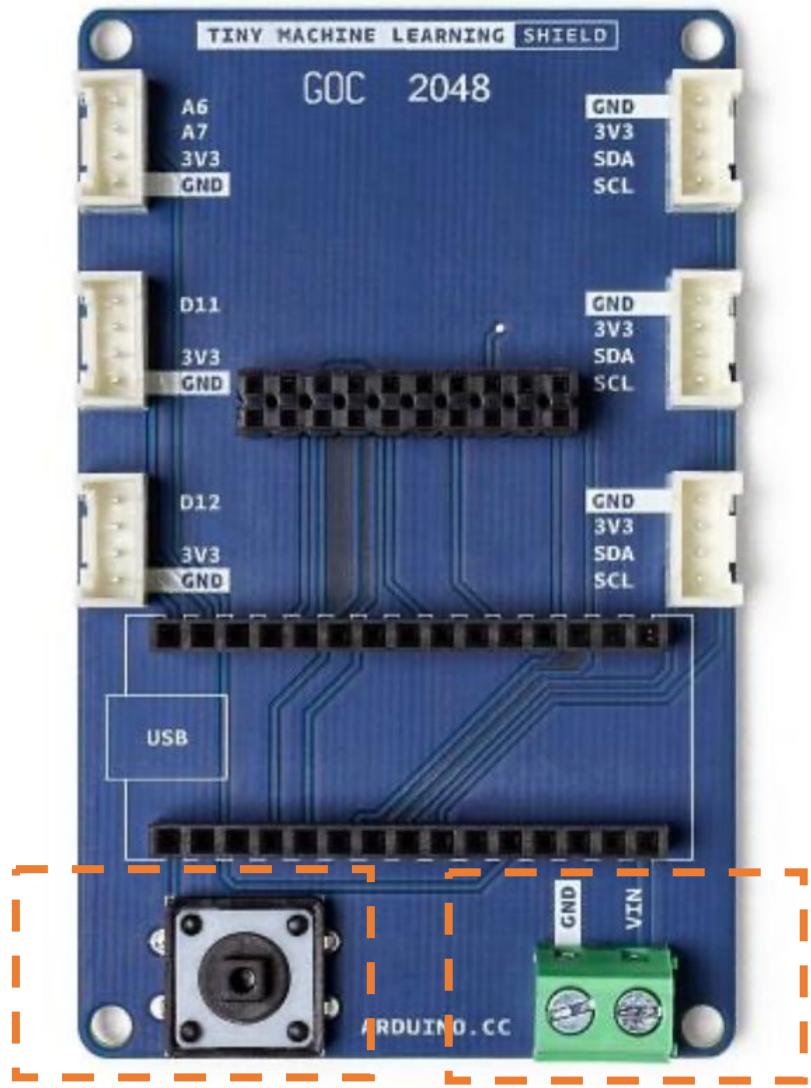


# TinyML Shield

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

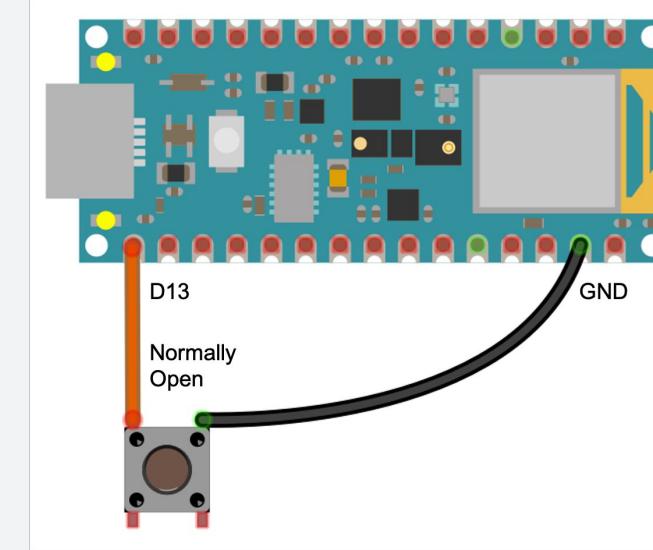


OV7670_VSYNC	8
OV7670_HREF	A1
OV7670_PLK	A0
OV7670_XCLK	9
OV7670_D0	10
OV7670_D1	1
OV7670_D2	0
OV7670_D3	2
OV7670_D4	3
OV7670_D5	5
OV7670_D6	6
OV7670_D7	4

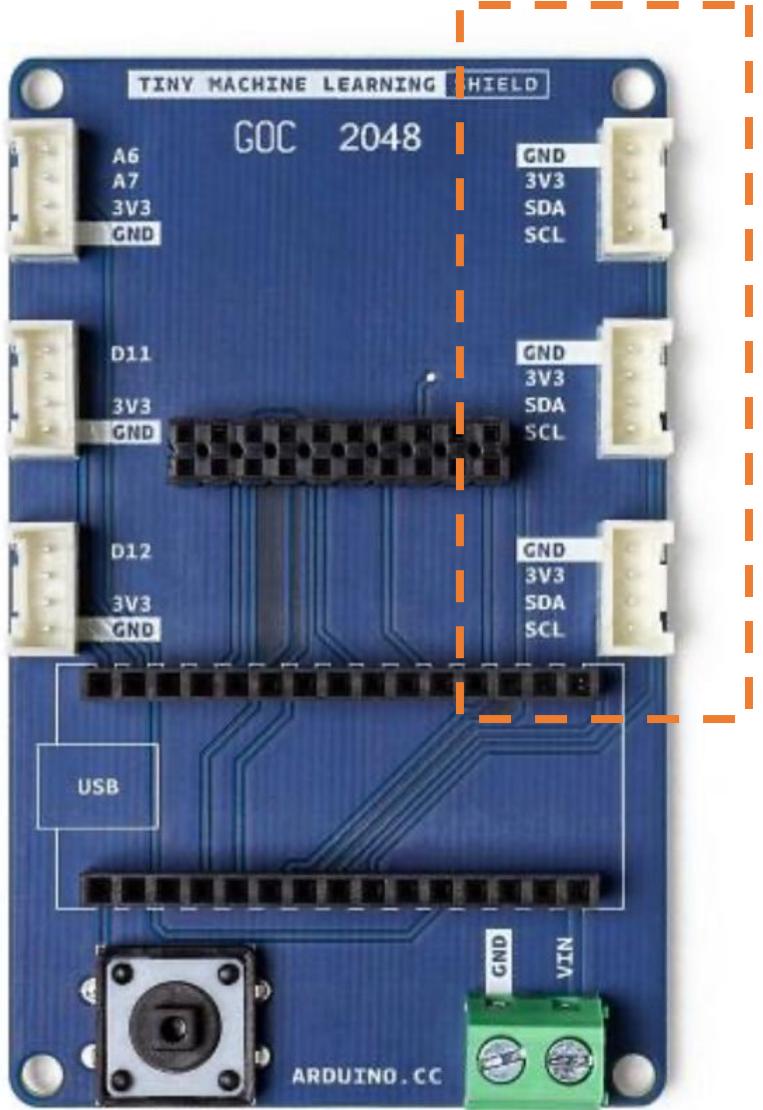


# TinyML Shield

A easily programmable  
button on the left

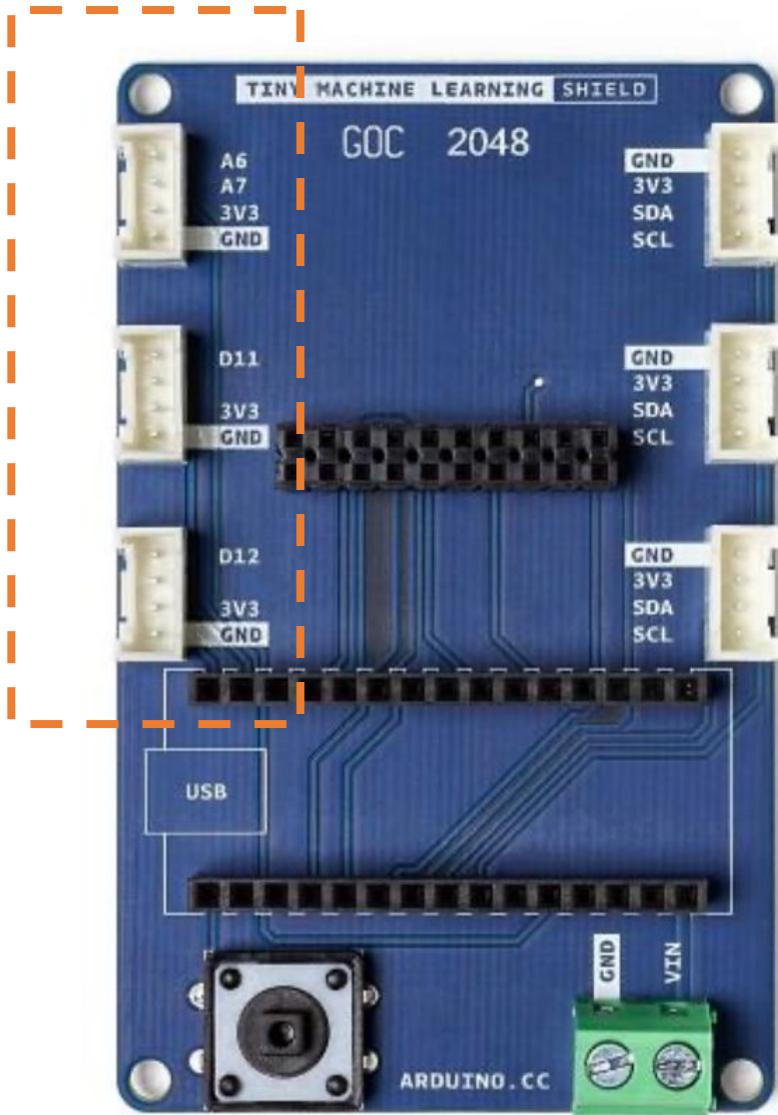


Screw-in terminal block for  
external (battery) power



# 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

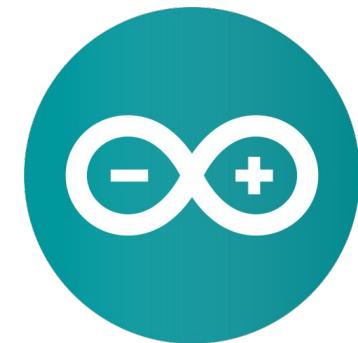
Facilitate **plug-and-play connections** to off-board modules to extend the possible scope of functionality to new **TinyML** applications

## Specifications

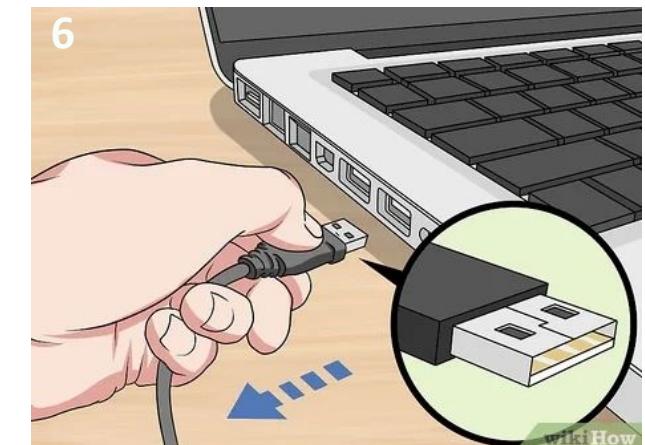
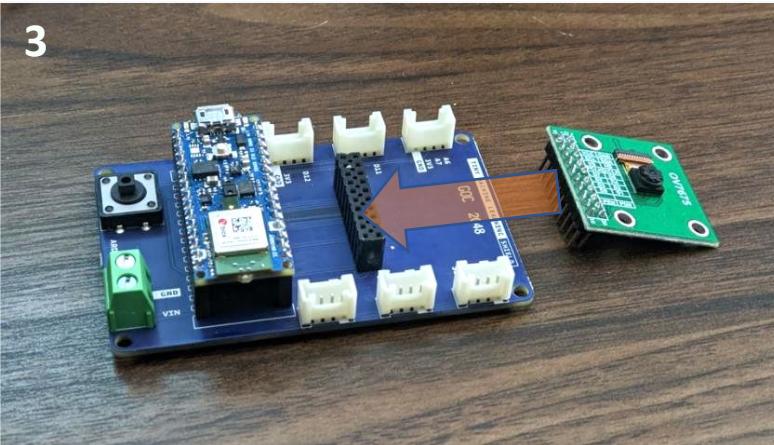
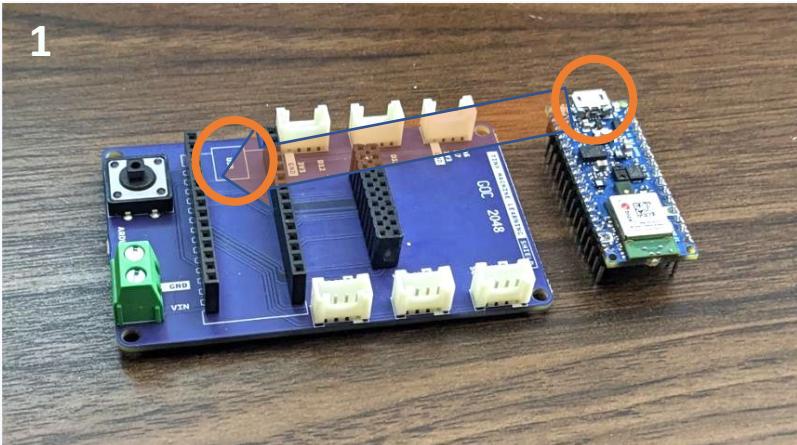
- Proprietary connection system from SeeedStudio, similar to JST PH-type connectors
- Large catalog of sensors, actuators available at [seeedstudio.com](http://seeedstudio.com)
- Be sure to check the voltage requirements and pinout of any new Grove module for compatibility with this shield before purchasing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

# TinyML Kit Installation

- Hardware Set-up
- Software Set-up



# Installing the Hardware



# Installing the Arduino IDE

This page is available in another language. Switch to: English

**Arduino Web Editor**  
Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

[CODE ONLINE](#)   [GETTING STARTED](#)

**Arduino Cloud**  
Set up automated lighting in minutes.  
[Get started!](#)

## Downloads

**Arduino IDE 1.8.19**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

**SOURCE CODE**  
Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this gpg key](#).

**DOWNLOAD OPTIONS**

**Windows** Win 7 and newer  
**Windows** ZIP file

**Windows app** Win 8.1 or 10 [Get](#)

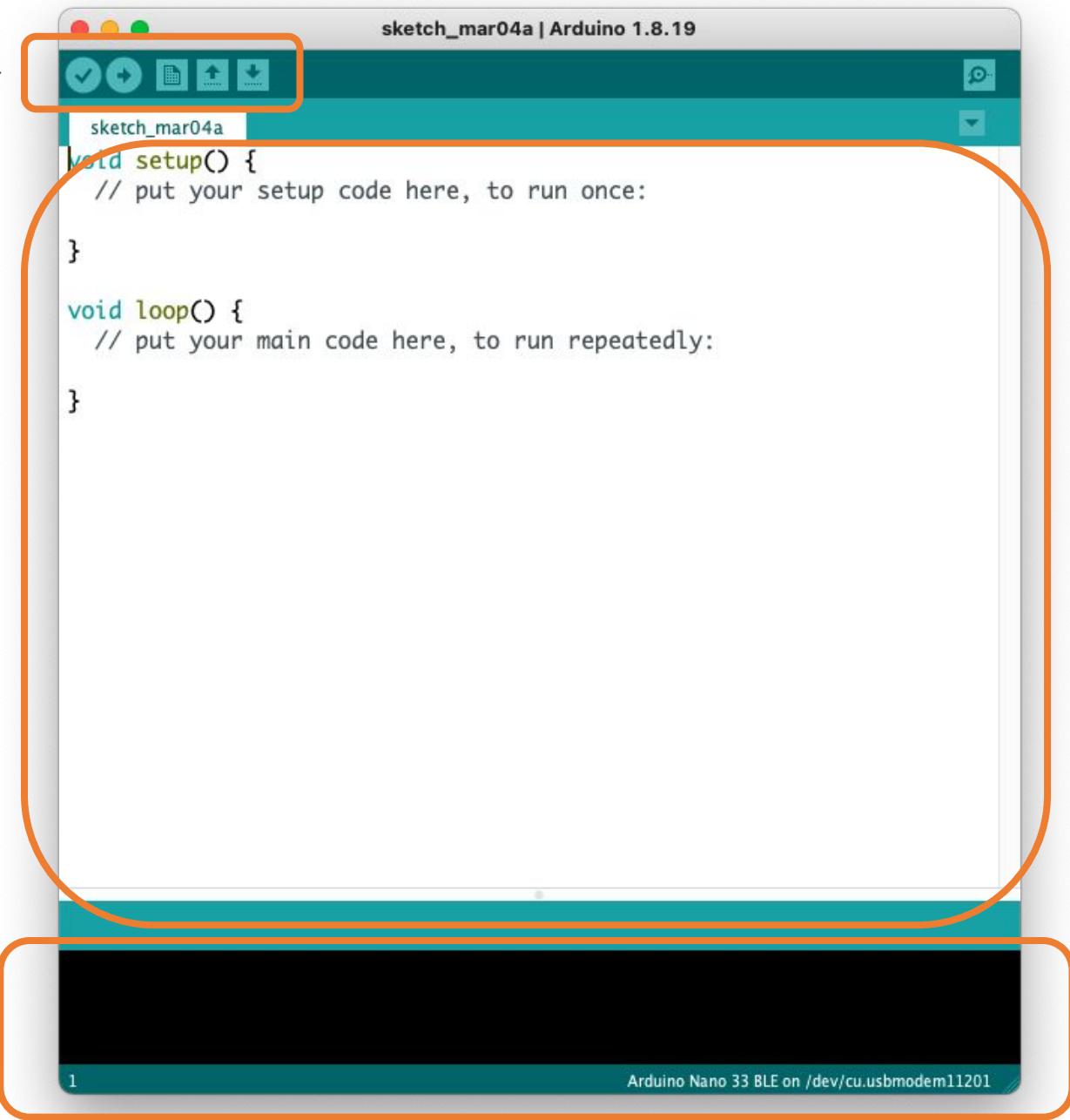
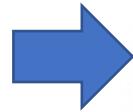
**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#)  
[Checksums \(sha512\)](#)

[Help](#)

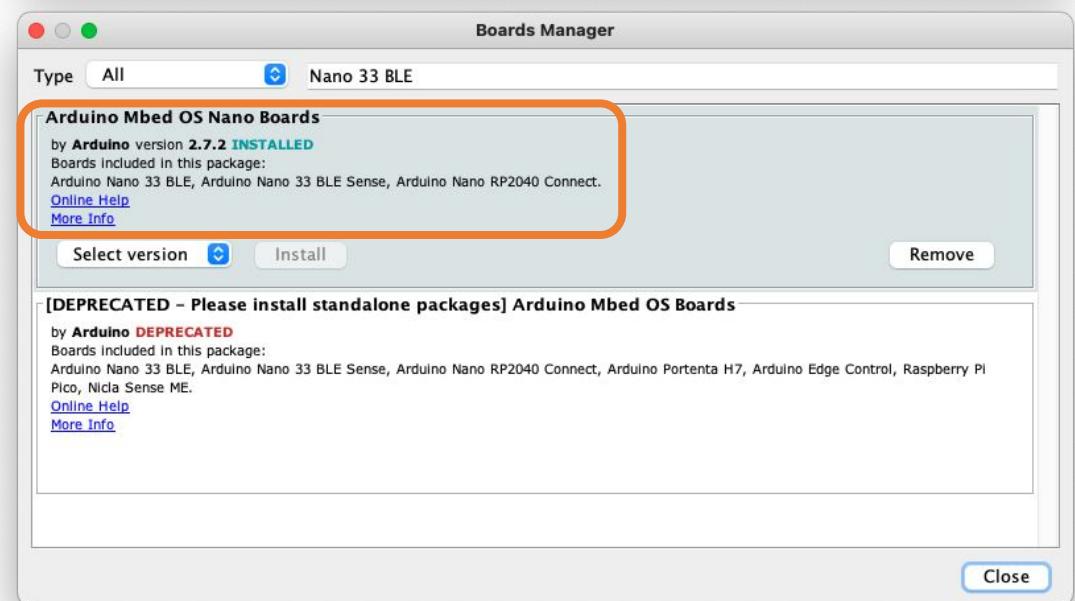
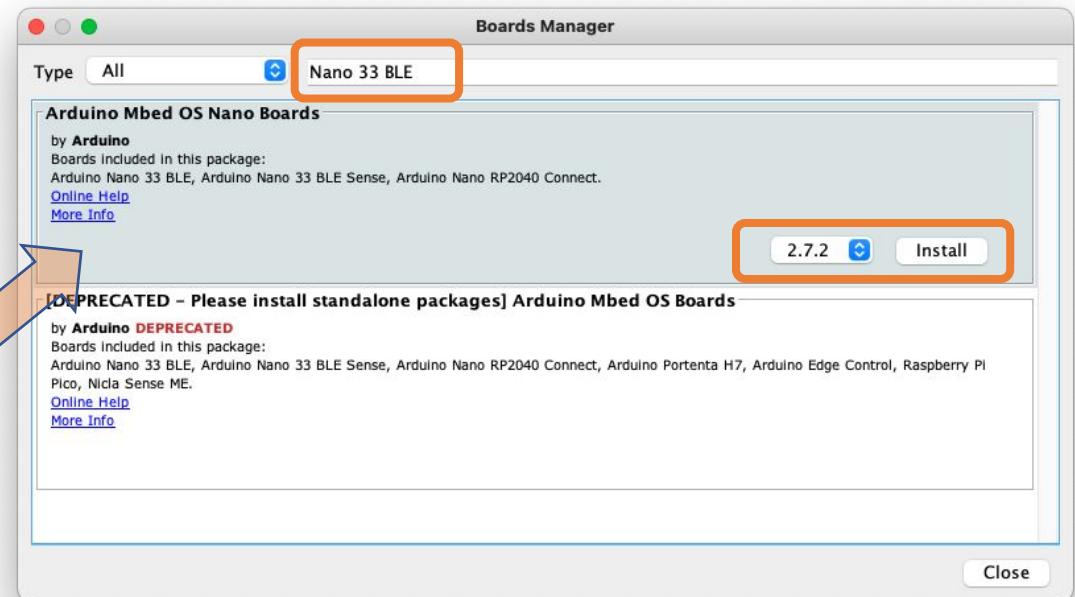
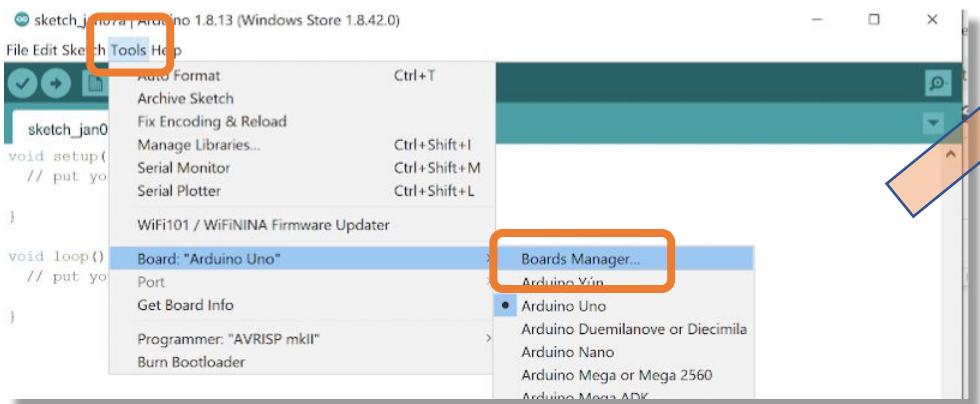
Menus  
and  
ToolBar



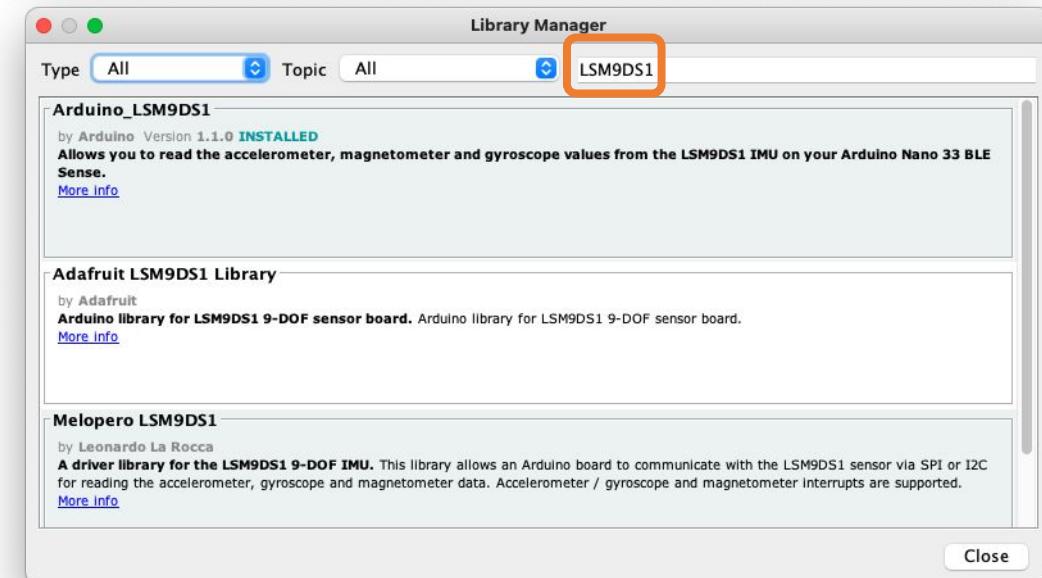
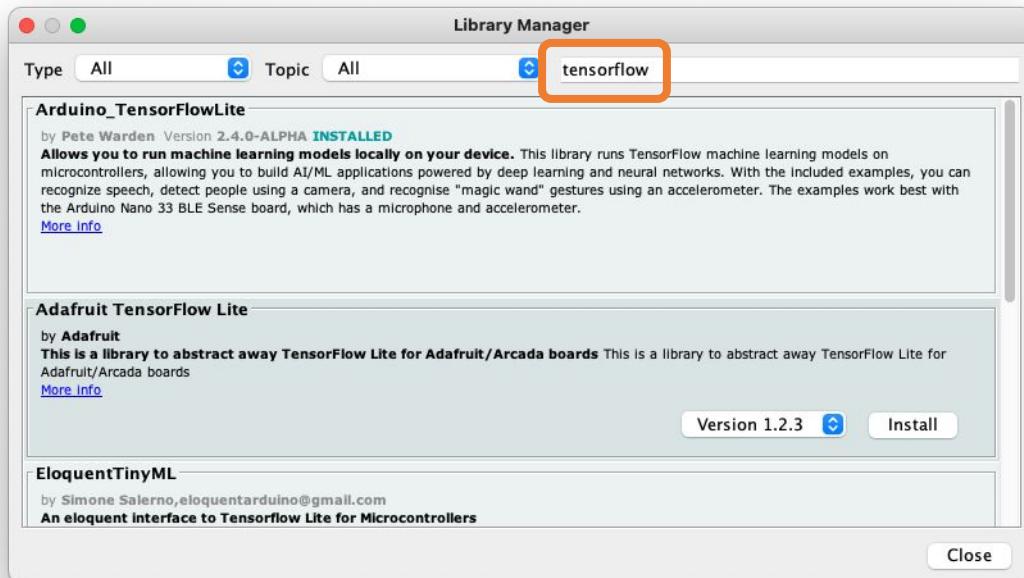
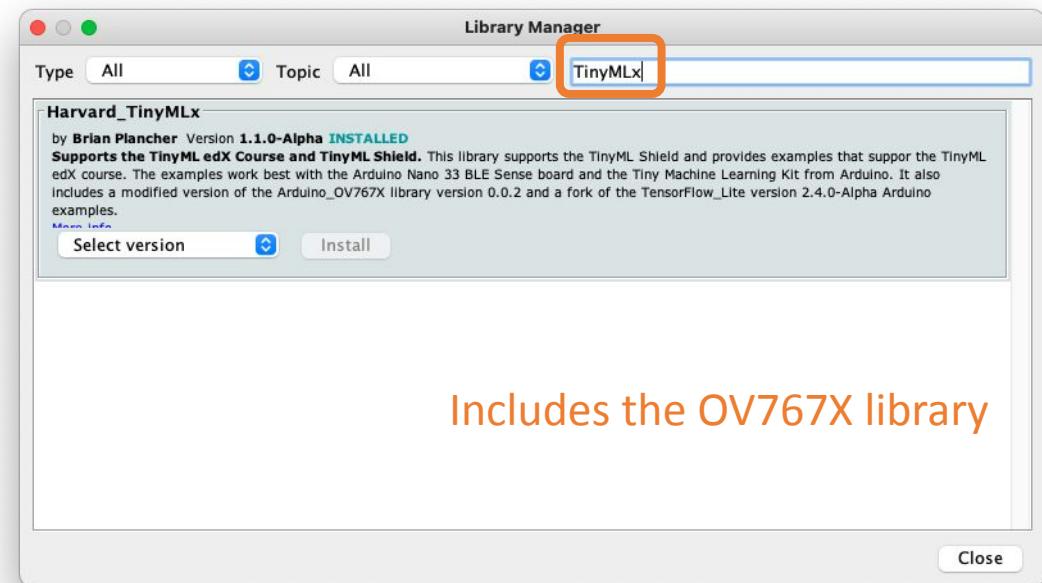
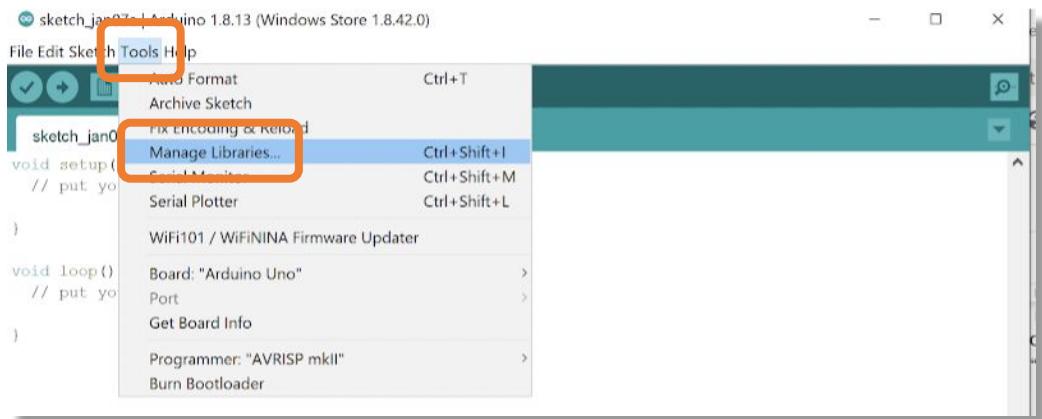
Code Area

Console

# Installing the Board Files

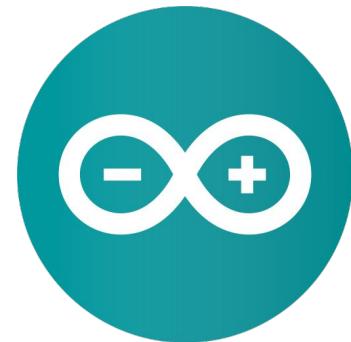


# Installing the Main Libraries

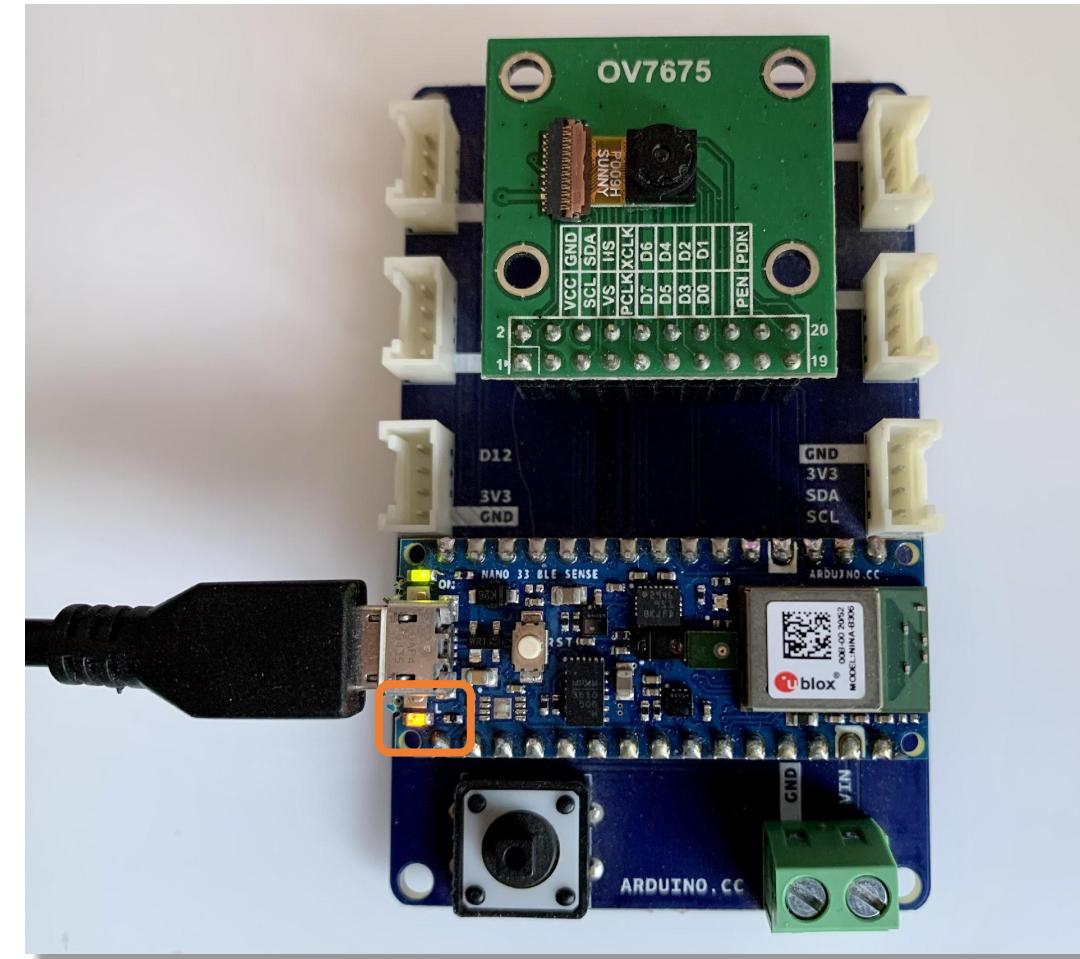
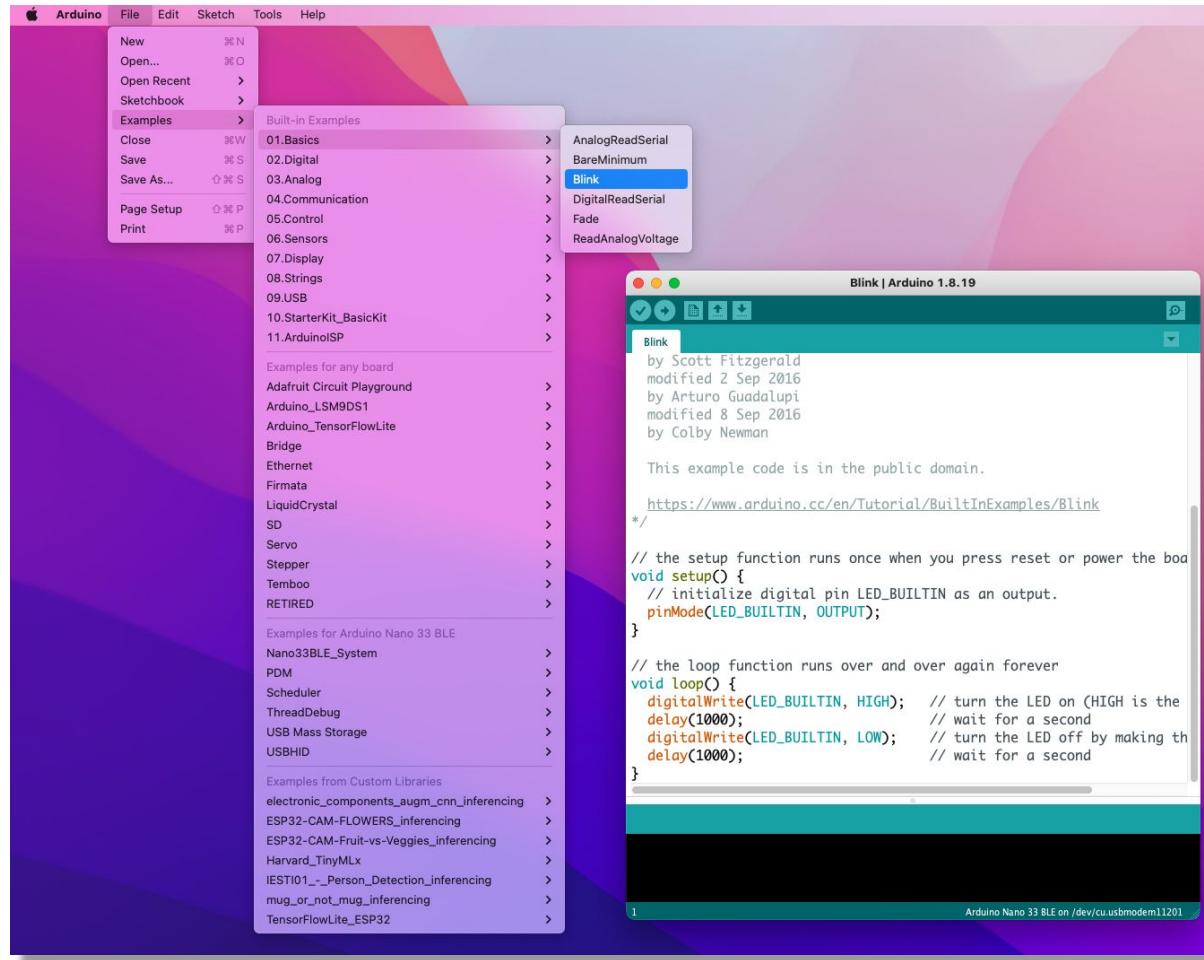


# TinyML Kit Test

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



# MCU installation test (Blink)



# Testing IMU

The screenshot shows the Arduino IDE interface. The menu bar includes File, Edit, Sketch, Tools, and Help. The main window displays the code for the `test_IMU` sketch. The code is as follows:

```
/*
 * Active Learning Labs
 * Harvard University
 * tinyMLx - Sensor Test
 *
 * Requires the Arduino_LSM9DS1 library library
 */

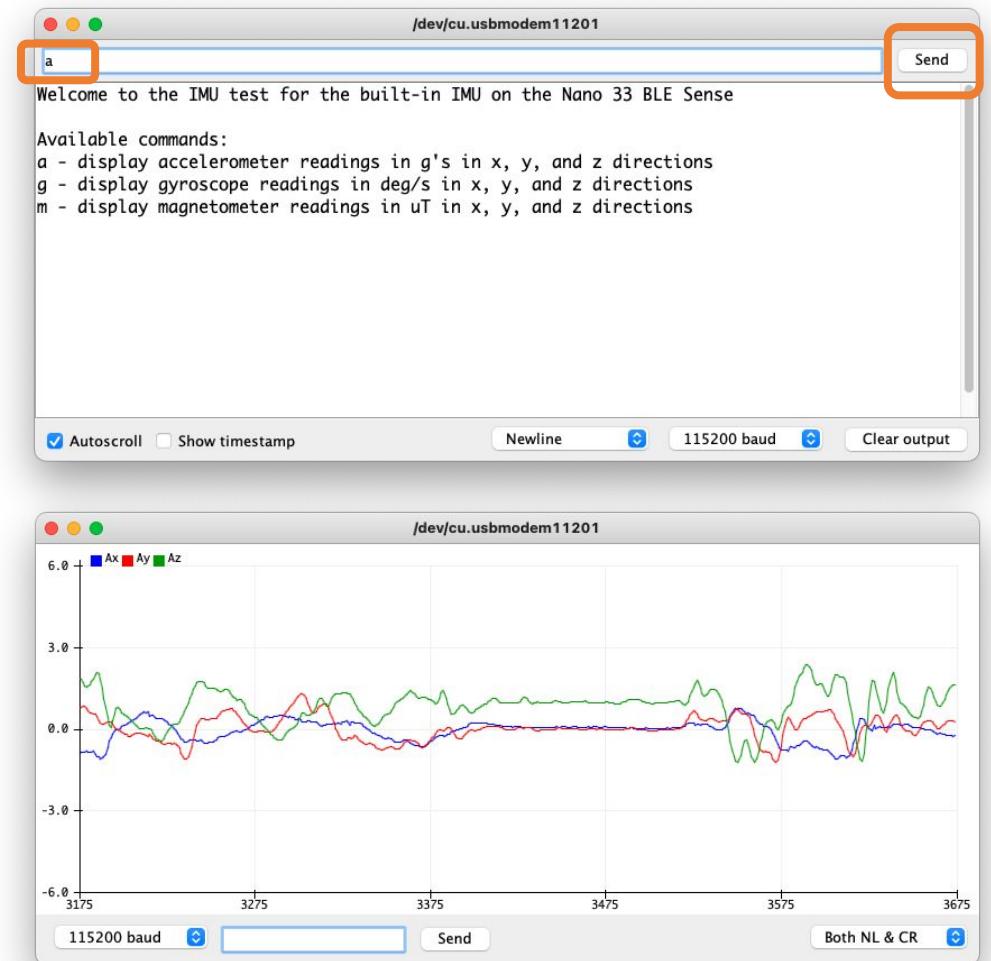
#include <Arduino_LSM9DS1.h>

int imuIndex = 0; // 0 - accelerometer, 1 - gyroscope, 2 - magnetometer
bool commandRecv = false; // flag used for indicating receipt of command
bool startStream = false;

void setup() {
    Serial.begin(9600);
    while (!Serial);

    // Initialize IMU
    if (!IMU.begin()) {
        Serial.println("Failed to initialize IMU");
        while (1);
    }
}
```

The status bar at the bottom indicates "Arduino Nano 33 BLE on /dev/cu.usbmodem11201". A dropdown menu in the bottom right corner shows the following options: magic\_wand, micro\_speech, multi\_tenant, person\_detection, test\_camera, **test\_IMU**, test\_microphone, extras.

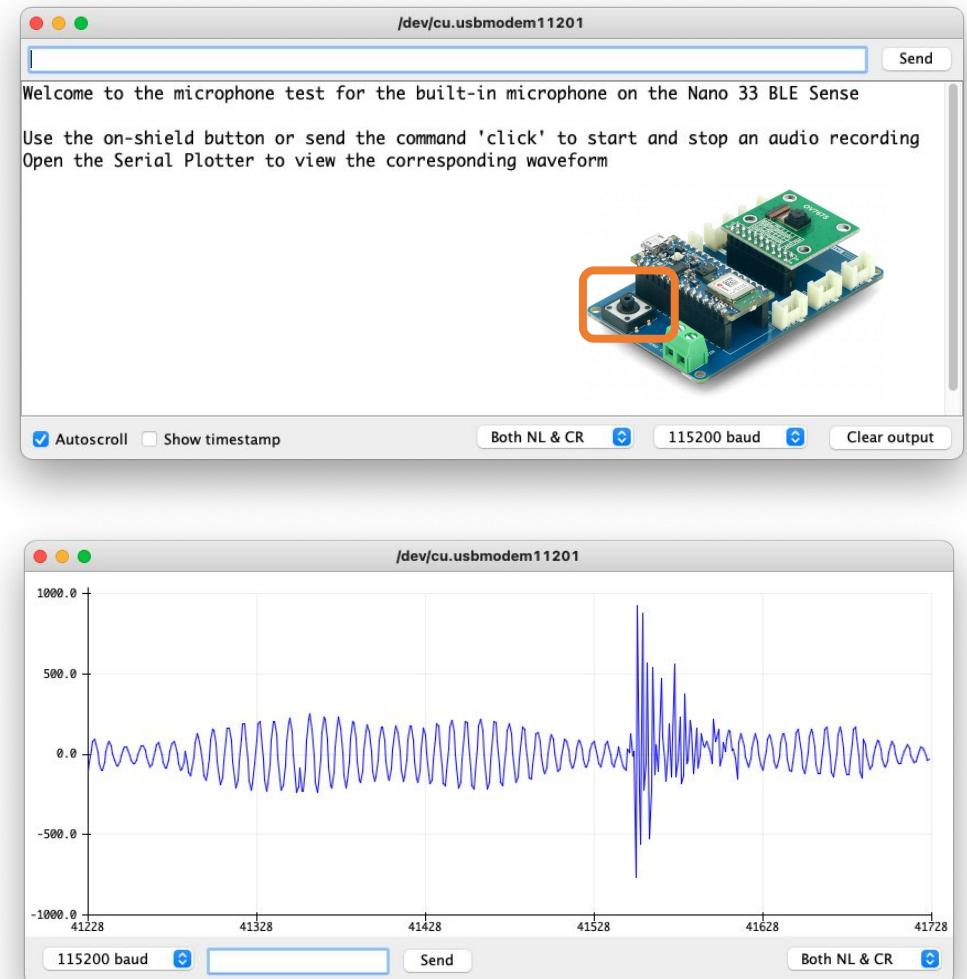


Notes: Close the Serial Monitor before open the Plotter  
Repeat test for 'g' and 'm'

# Testing Microphone

The screenshot shows the Arduino IDE interface. The menu bar includes File, Edit, Sketch, Tools, and Help. A context menu is open over the 'test\_microphone' sketch, highlighting the 'extras' option. The 'extras' submenu lists several custom libraries: magic\_wand, micro\_speech, multi\_tenant, person\_detection, test\_camera, test\_IMU, test\_microphone, and extras. The main code editor displays the 'test\_microphone' sketch, which includes comments for Active Learning Labs at Harvard University and the tinyMLx - Built-in Microphone Test. The code uses the PDM library to handle audio input and the TinyMLShield library to initialize the shield. It includes setup and loop functions for initializing the serial port, starting the TinyML shield, and handling PDM data receive.

```
Active Learning Labs  
Harvard University  
tinyMLx - Built-in Microphone Test  
  
include <PDM.h>  
include <TinyMLShield.h>  
  
// PDM buffer  
short sampleBuffer[256];  
volatile int samplesRead;  
  
bool record = false;  
bool commandRecv = false;  
  
void setup() {  
    Serial.begin(9600);  
    while (!Serial);  
  
    // Initialize the TinyML Shield  
    initializeShield();  
  
    PDM.onReceive(onPDMdata);  
}  
  
void loop() {  
}
```



Note: Close the Serial Monitor before open the Plotter

# Testing Camera

The screenshot shows the Arduino IDE interface. The left sidebar displays the 'Examples' menu with various categories and sub-examples. In the center, the code editor window is open with the 'test\_camera' sketch. The code is as follows:

```
/*
  Active Learning Labs
  Harvard University
  tinyMLx - OV7675 Camera Test

  */

#include <TinyMLShield.h>

bool commandRecv = false; // flag used for indicating receipt of comm
bool liveFlag = false; // flag as true to live stream raw camera byte
bool captureFlag = false;

// Image buffer;
byte image[176 * 144 * 2]; // QCIF: 176x144 x 2 bytes per pixel (RGB5
int bytesPerFrame;

void setup() {
  Serial.begin(9600);
  while (!Serial);

  initializeShield();
}

// Initialize the OV7675 camera
```

The status bar at the bottom indicates "Arduino Nano 33 BLE on /dev/cu.usbmodem11201".

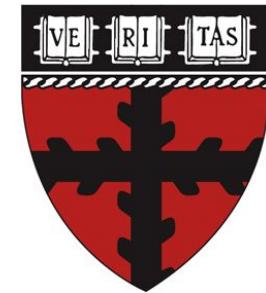
The image shows two terminal windows. The top window has the title bar "/dev/cu.usbmodem11201" and the text "single" in the input field. The bottom window has the title bar "/dev/cu.usbmodem11201" and the text "capture" in the input field. Both windows show the message "Welcome to the OV7675 test" and "Available commands:" followed by the options "single", "live", and "capture". The bottom window also displays the message "Camera in single mode, type "capture" to initiate an image capture" and "Image data will be printed out in 3 seconds...". The terminal settings are set to "Both NL & CR" and "115200 baud". Orange boxes highlight the "single" and "capture" inputs in both windows.

Note: You can Press Button instead of send 'capture'

# Testing Camera



**Thanks**  
And stay safe!



**UNIFEI**

