

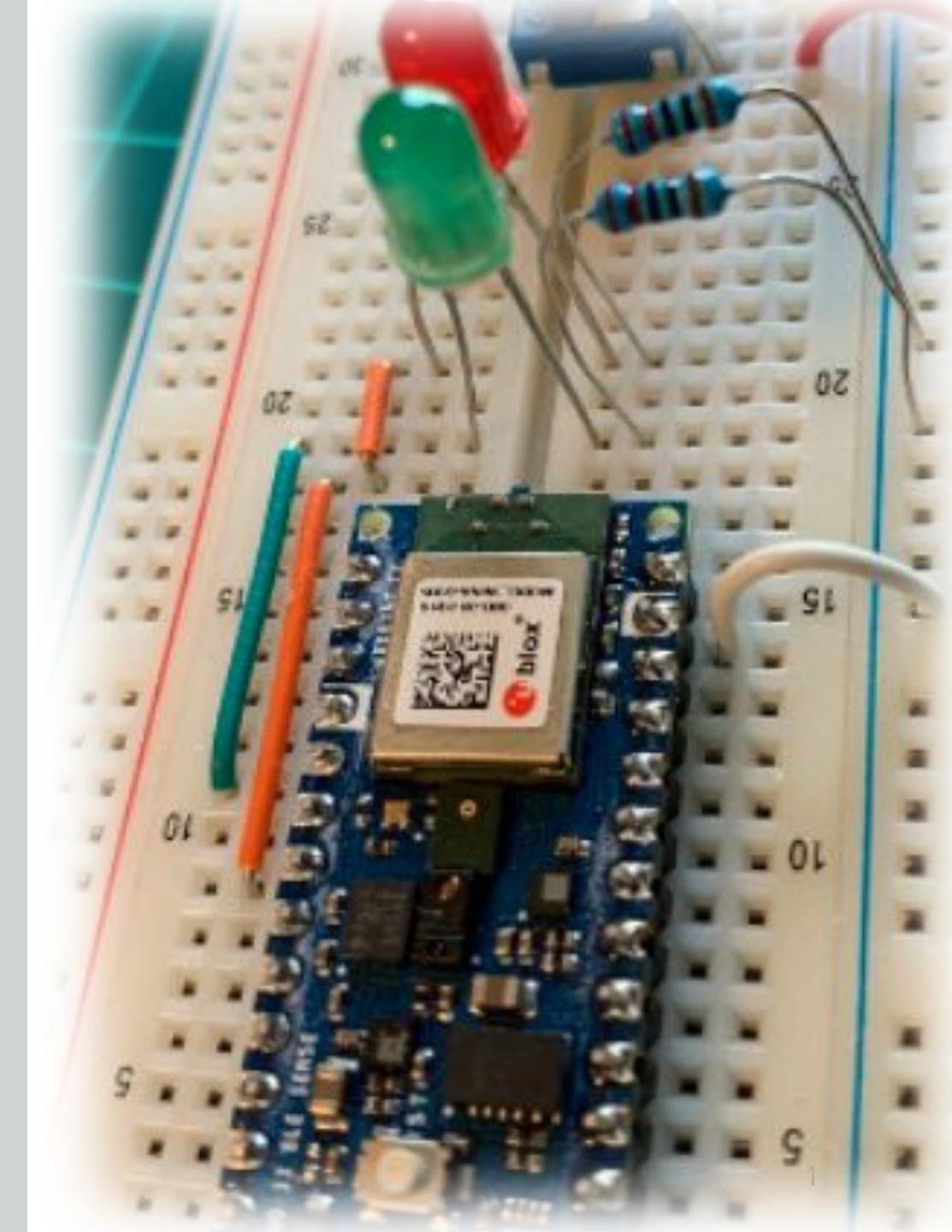
IESTI01 – TinyML

Embedded Machine Learning

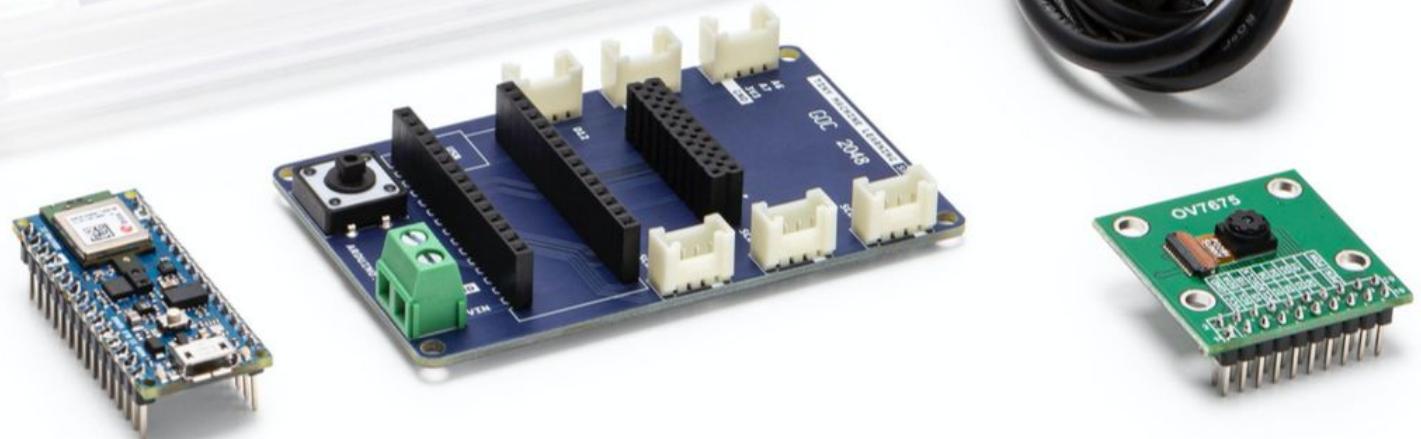
17. TinyML Kit Overview - HW and SW installation & Test



Prof. Marcelo Rovai
UNIFEI



TinyML Kit Overview



Nano 33 BLE Sense (+ USB cable)

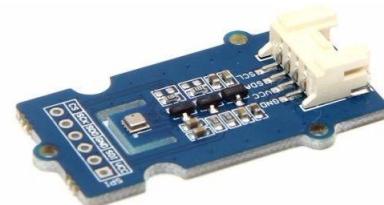


Purpose

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

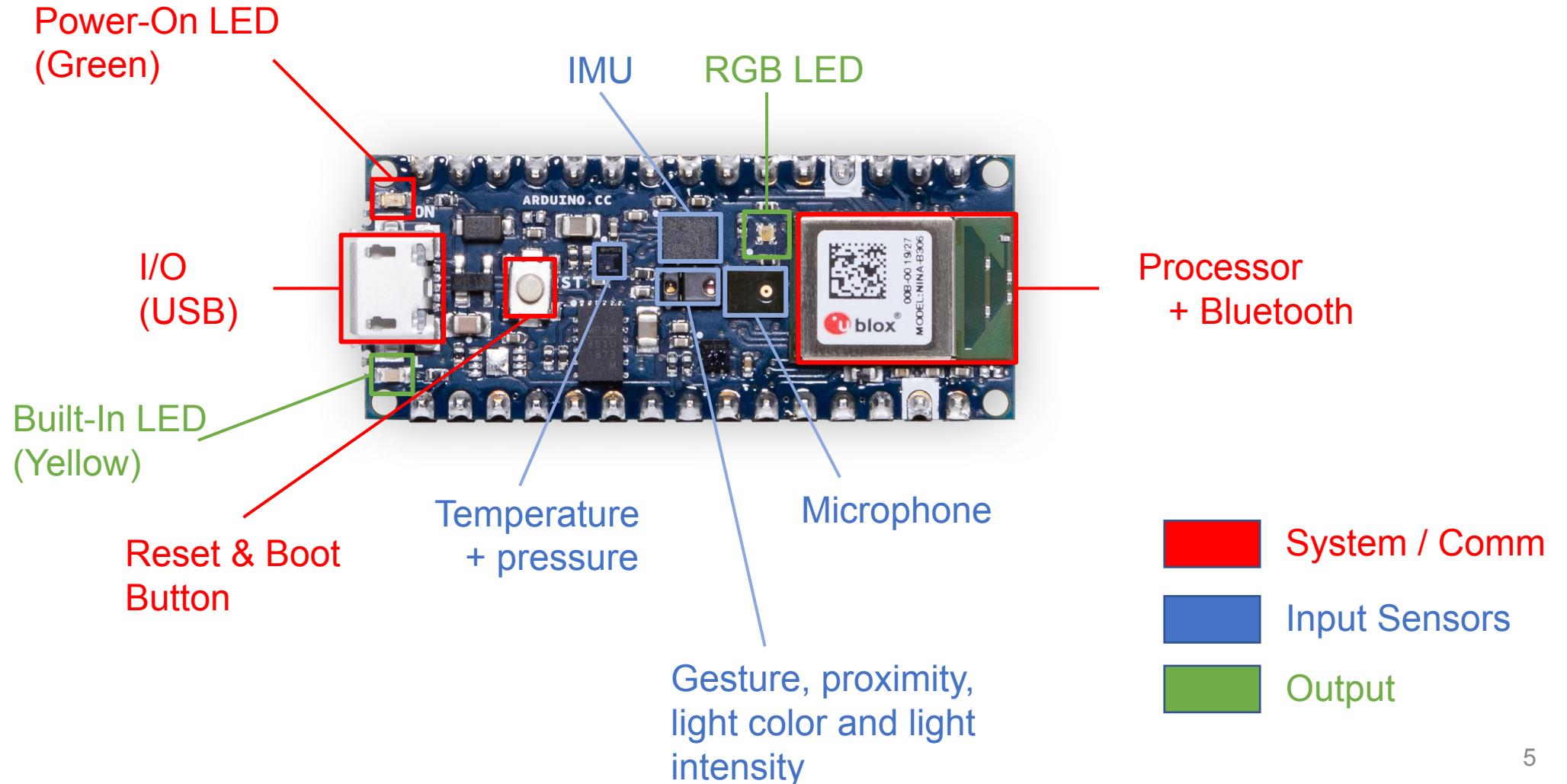
Specifications

- **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



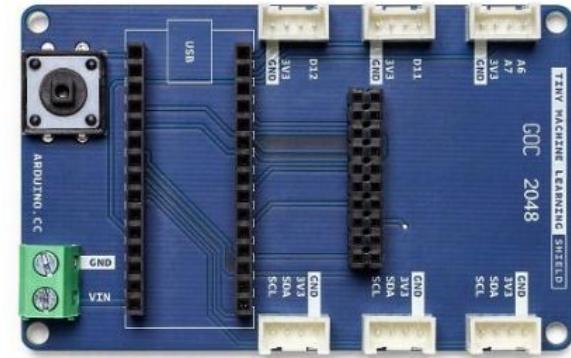
Purpose

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

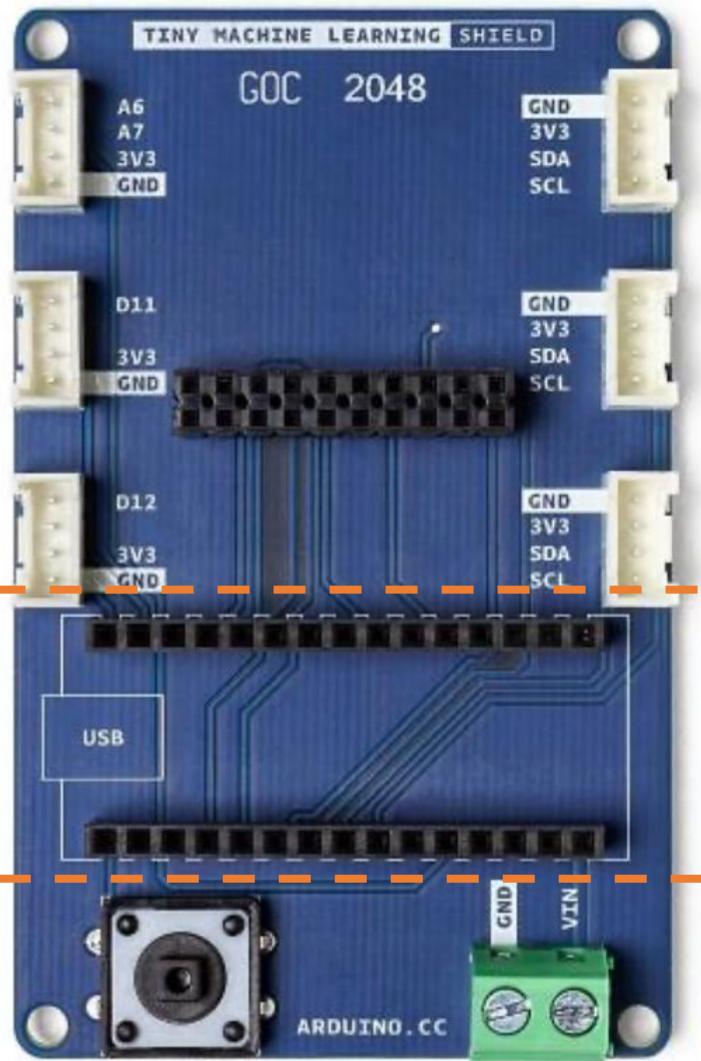


Purpose

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**

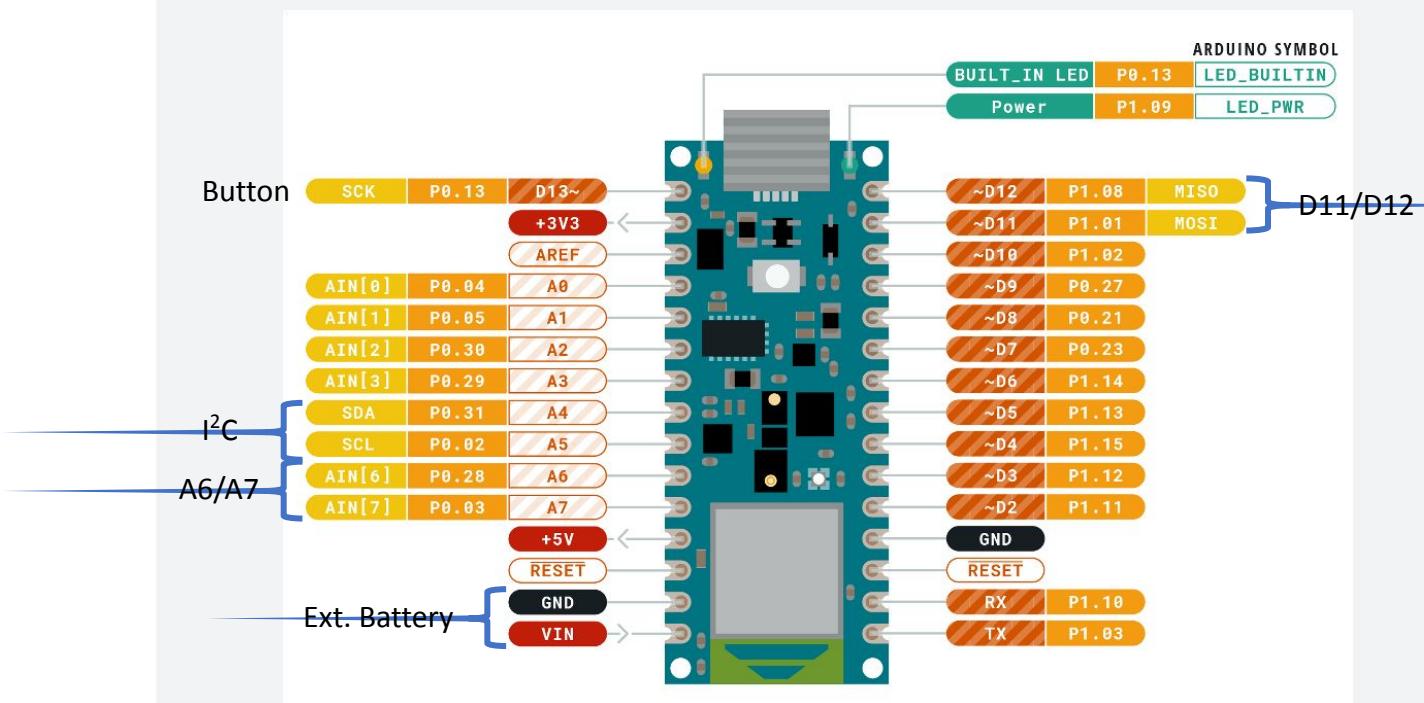
Specifications

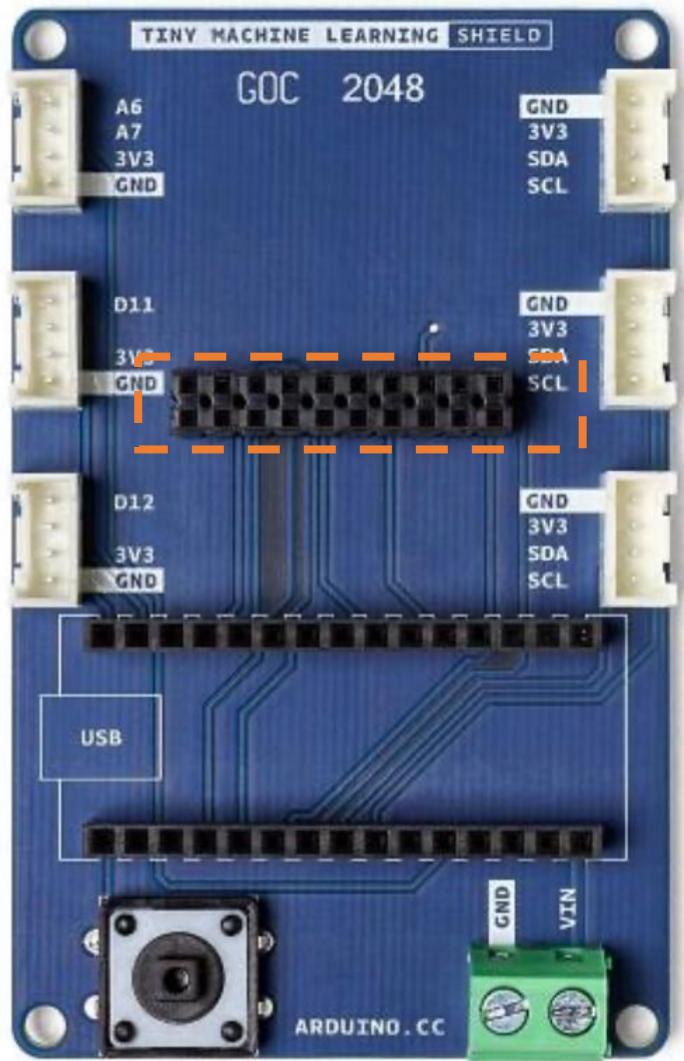
- Grove connectors (3.3V I2C and simple digital / analog - see pinouts)
- 2x10 pin array for OV7675 camera module
- Voltage input terminal block, accepts 4.5 to 21V (down regulated to 3.3V on Nano 33)



TinyML Shield

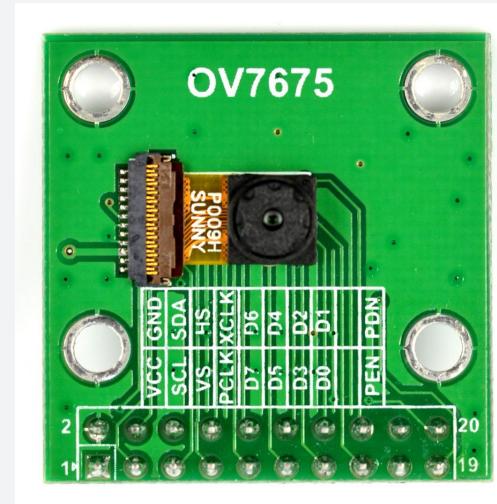
Two rows of 1x15 headers
that you can slot the Nano
33 BLE sense into



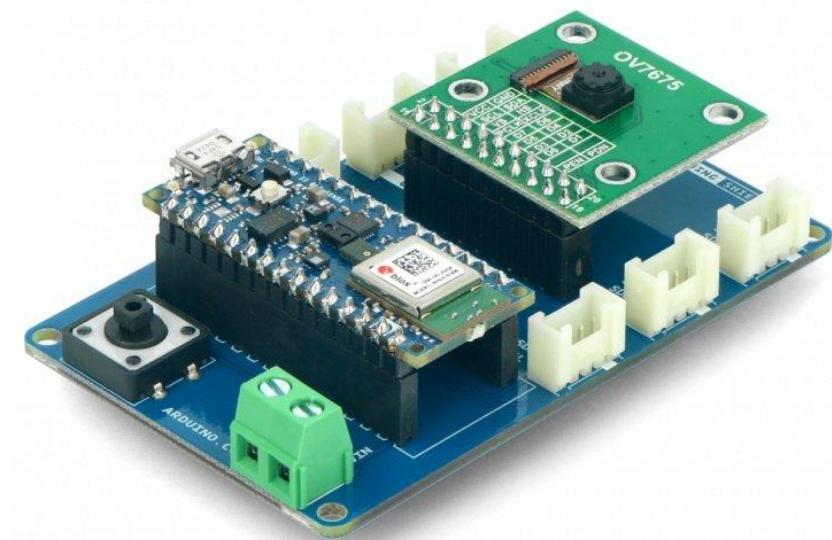
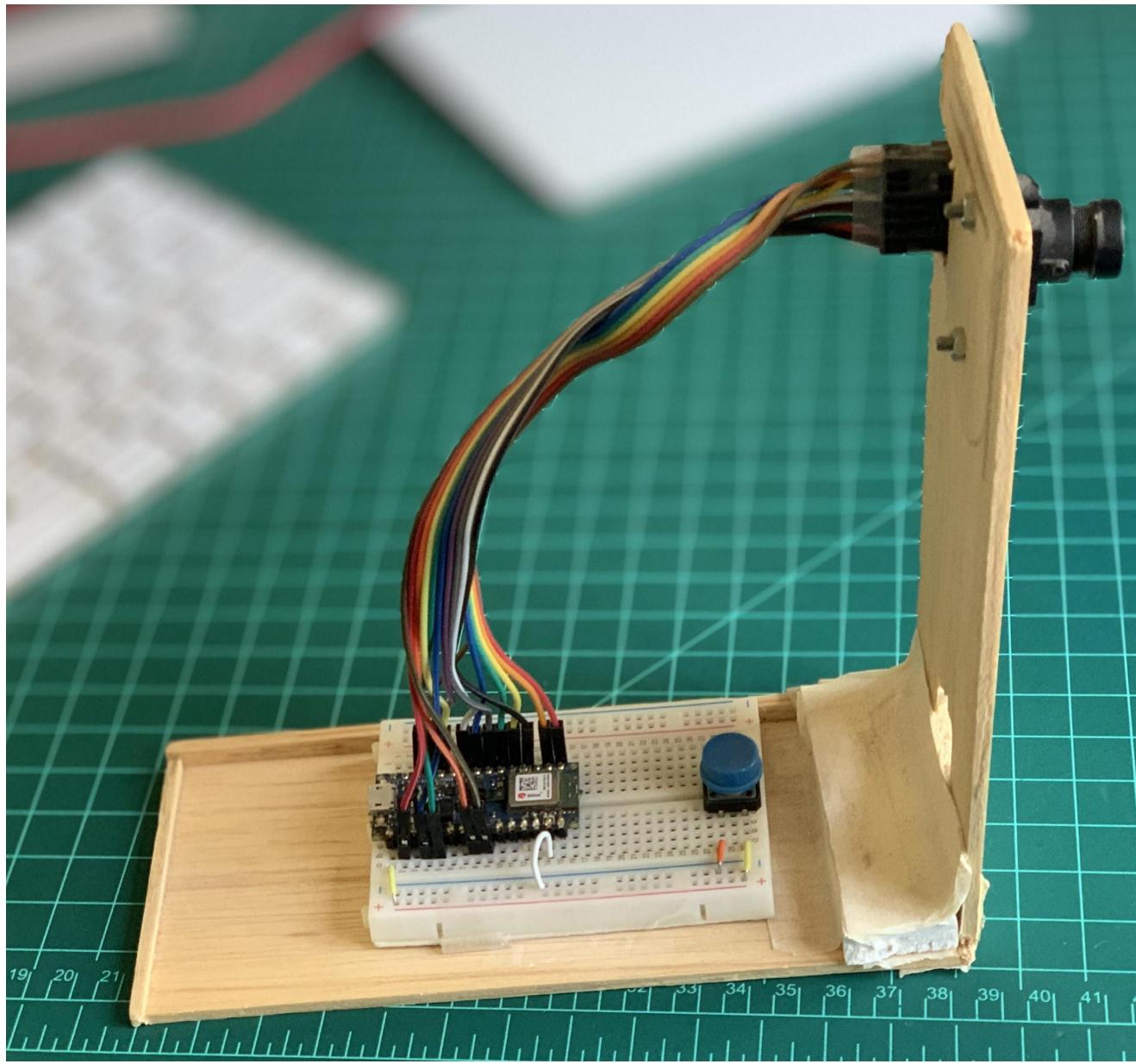


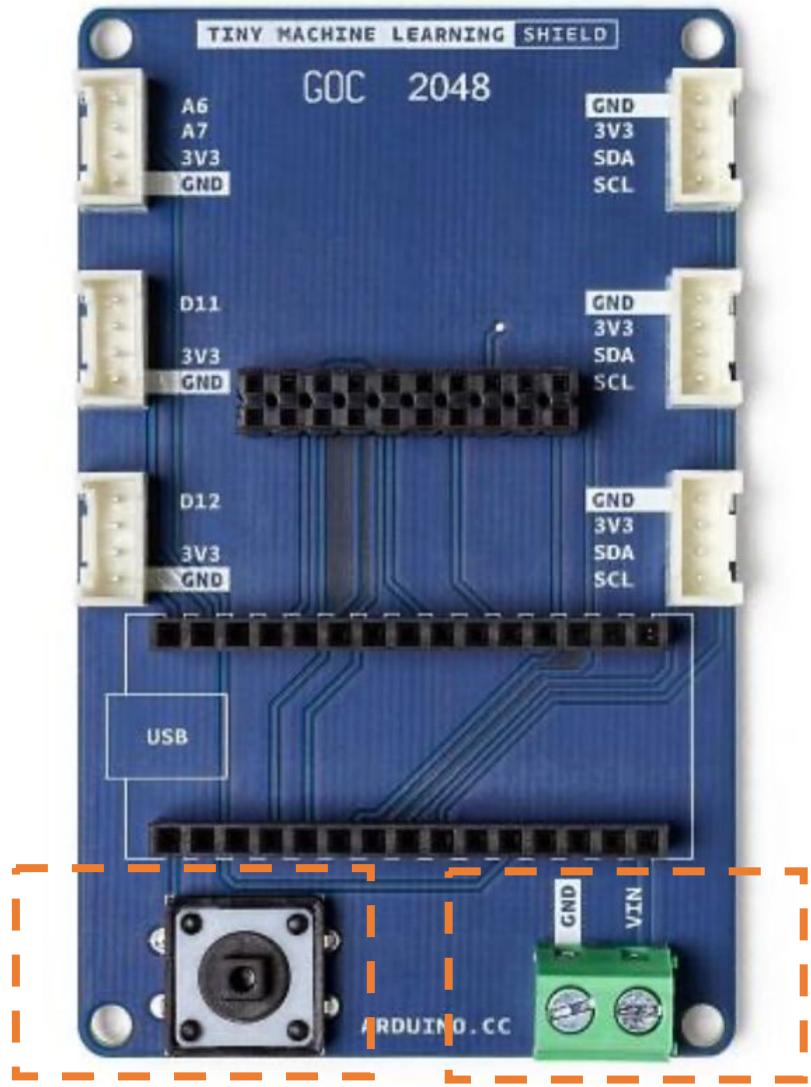
TinyML Shield

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



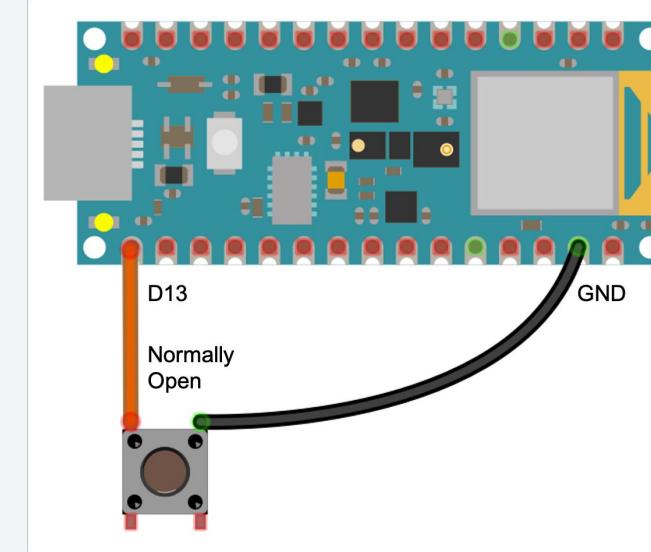
OV7670_VSYNC	8
OV7670_HREF	A1
OV7670_PLK	A0
OV7670_XCLK	9
OV7670_D0	10
OV7670_D1	1
OV7670_D2	0
OV7670_D3	2
OV7670_D4	3
OV7670_D5	5
OV7670_D6	6
OV7670_D7	4



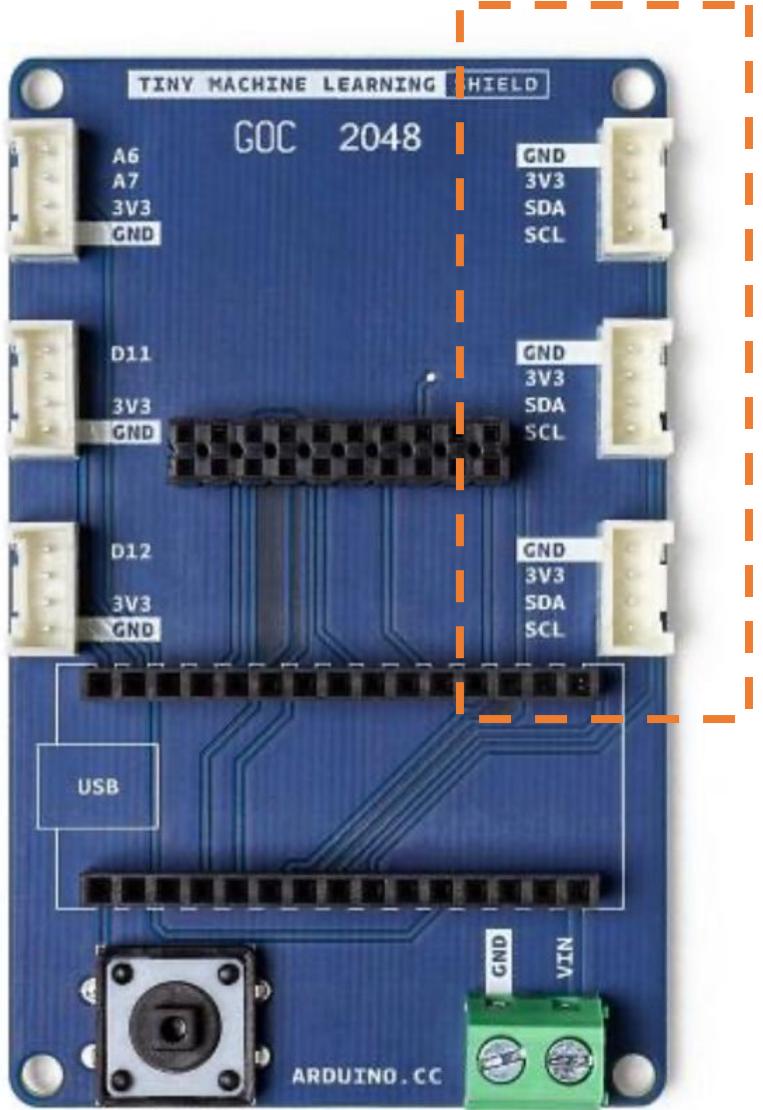


TinyML Shield

A easily programmable
button on the left

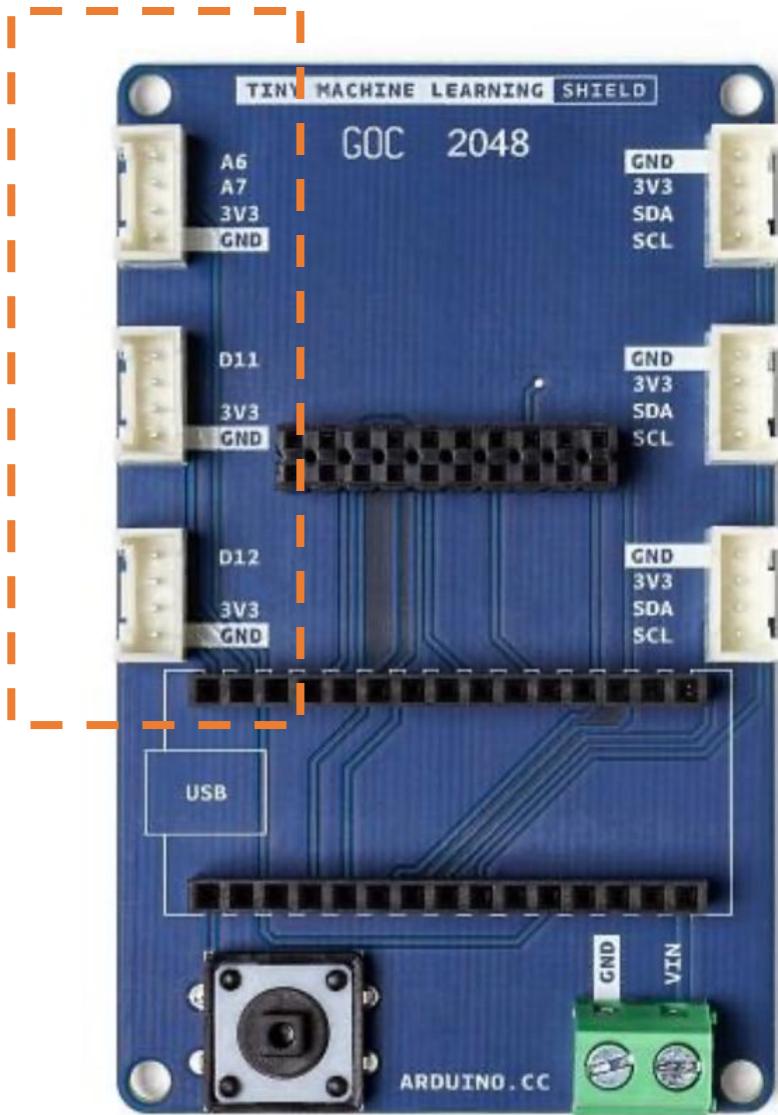


Screw-in terminal block for
external (battery) power (4.5V to 21V)



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

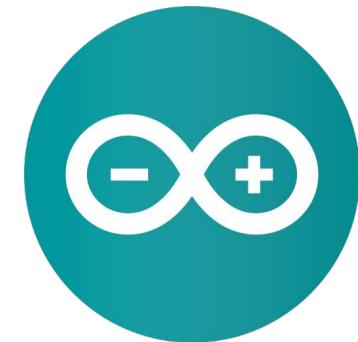
Facilitate **plug-and-play connections** to off-board modules to extend the possible scope of functionality to new **TinyML** applications

Specifications

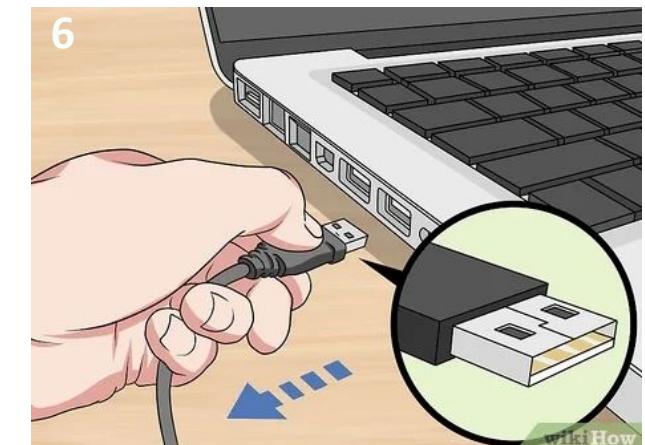
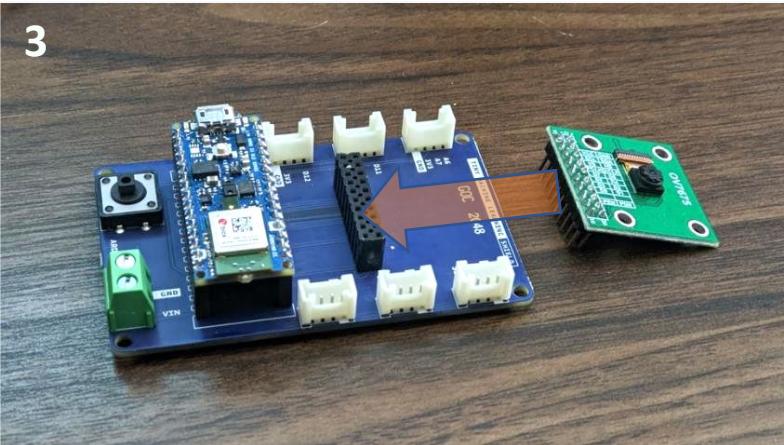
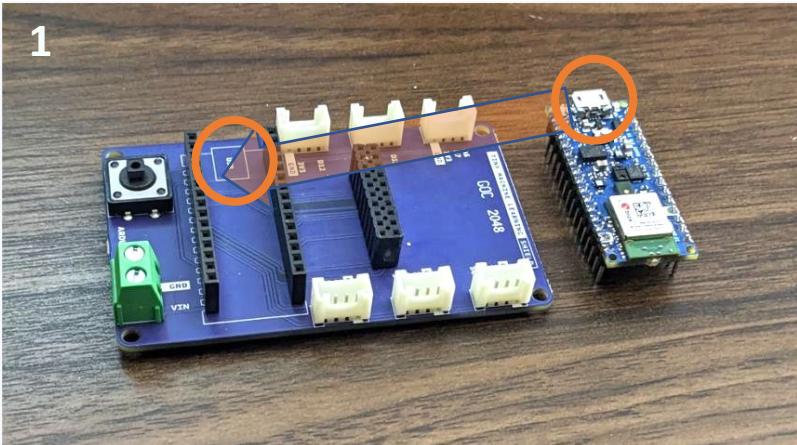
- Proprietary connection system from SeeedStudio, similar to JST PH-type connectors
- Large catalog of sensors, actuators available at seeedstudio.com
- Be sure to check the voltage requirements and pinout of any new Grove module for compatibility with this shield before purchasing or connecting said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

TinyML Kit Installation

- Hardware Set-up
- Software Set-up



Installing the Hardware



Installing the Arduino IDE

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](#)

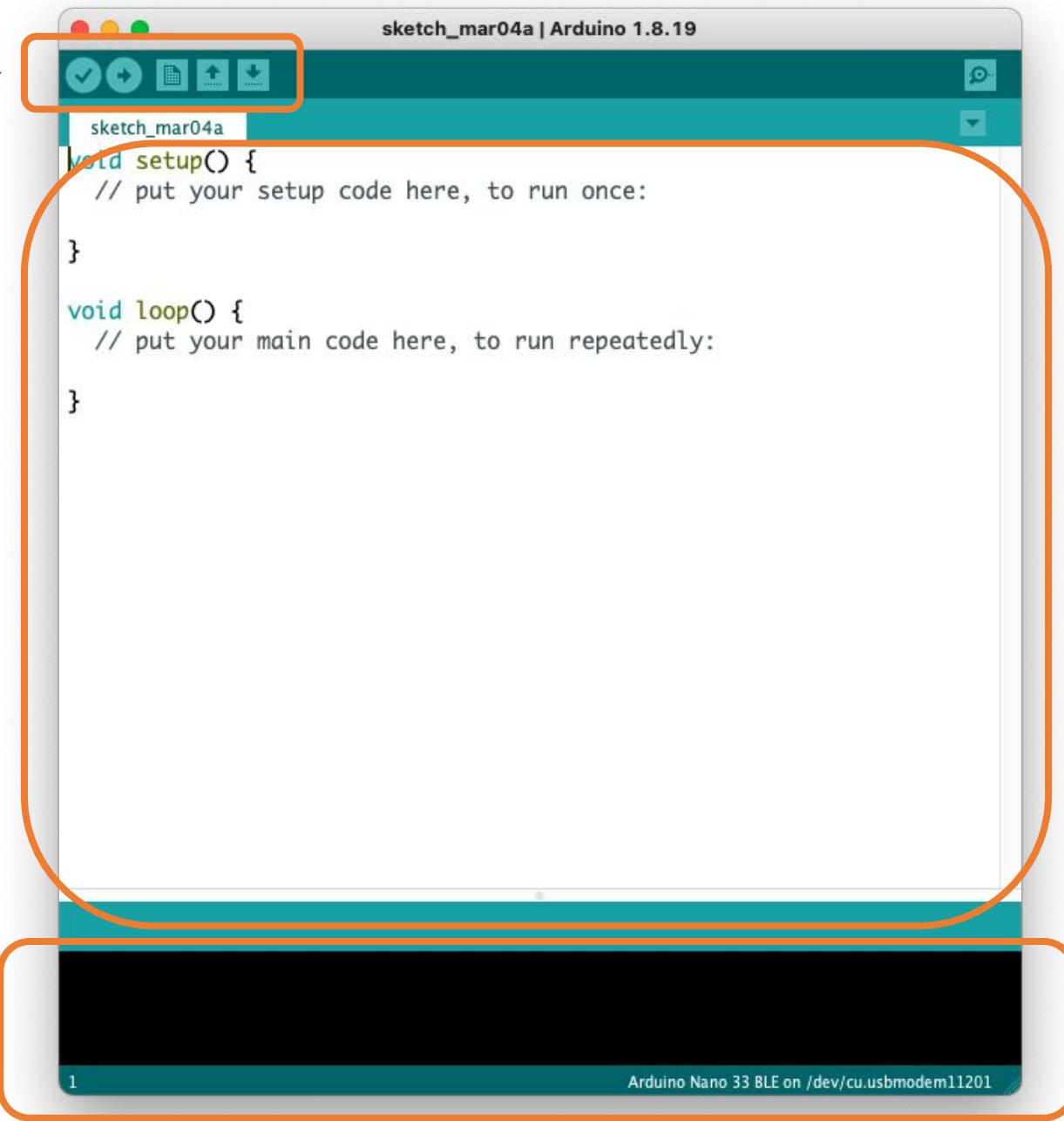
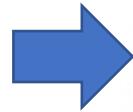
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

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

[Help](#)

Menus
and
ToolBar



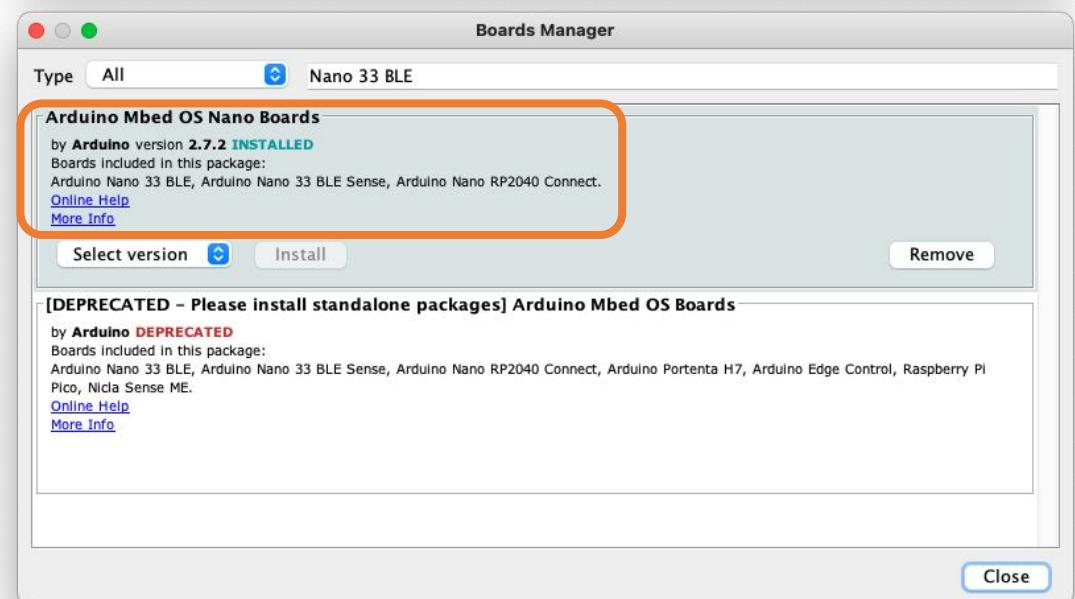
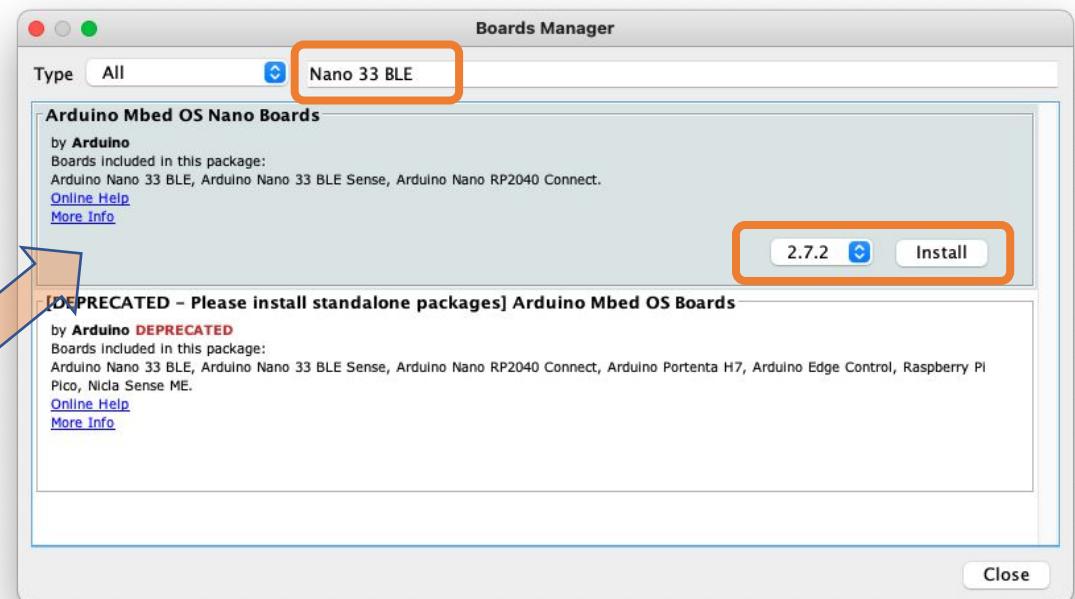
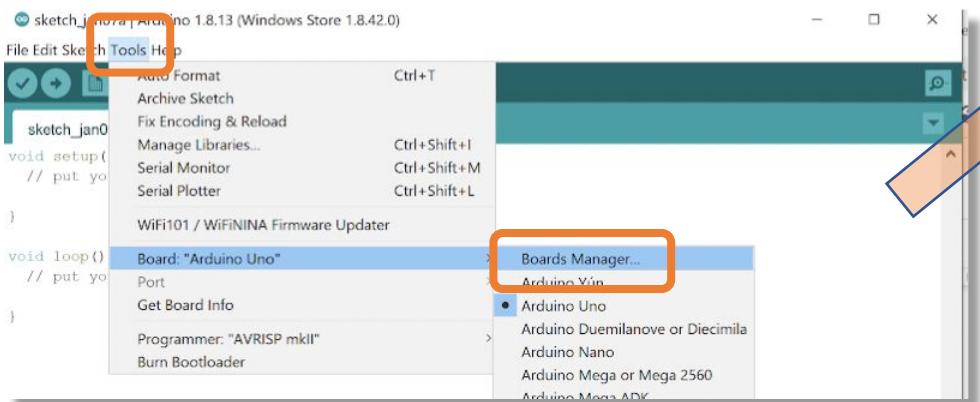
Code Area



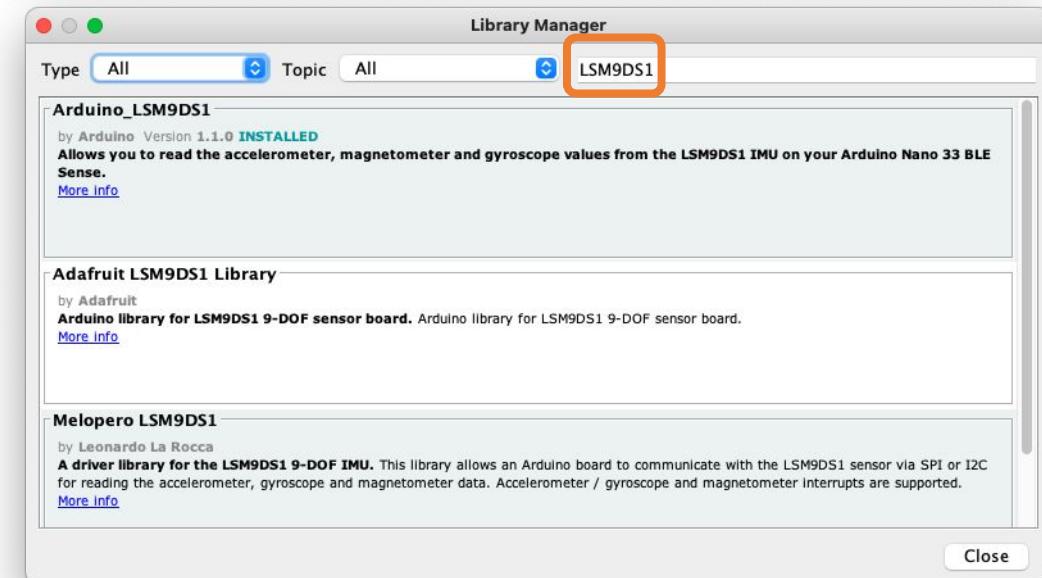
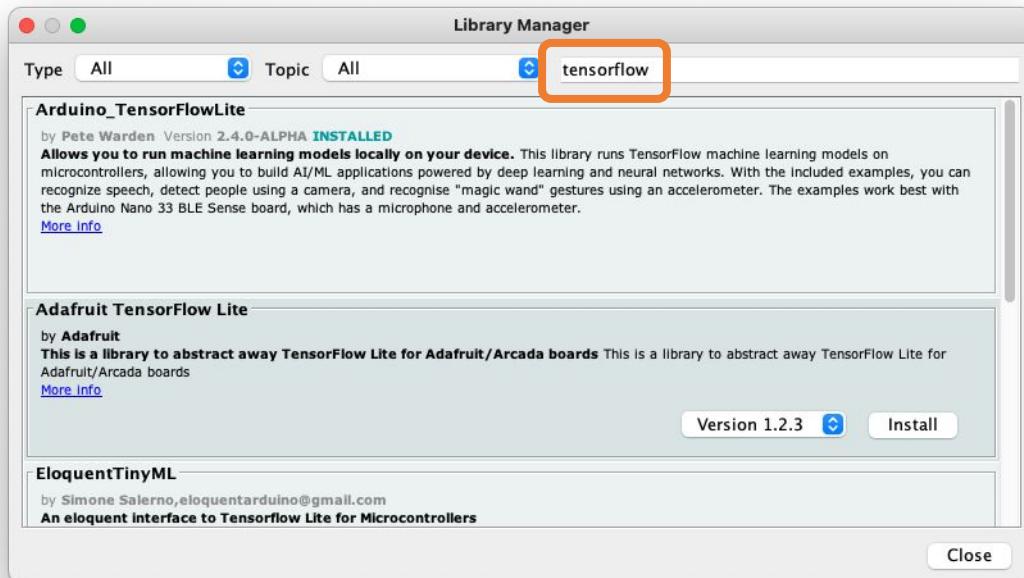
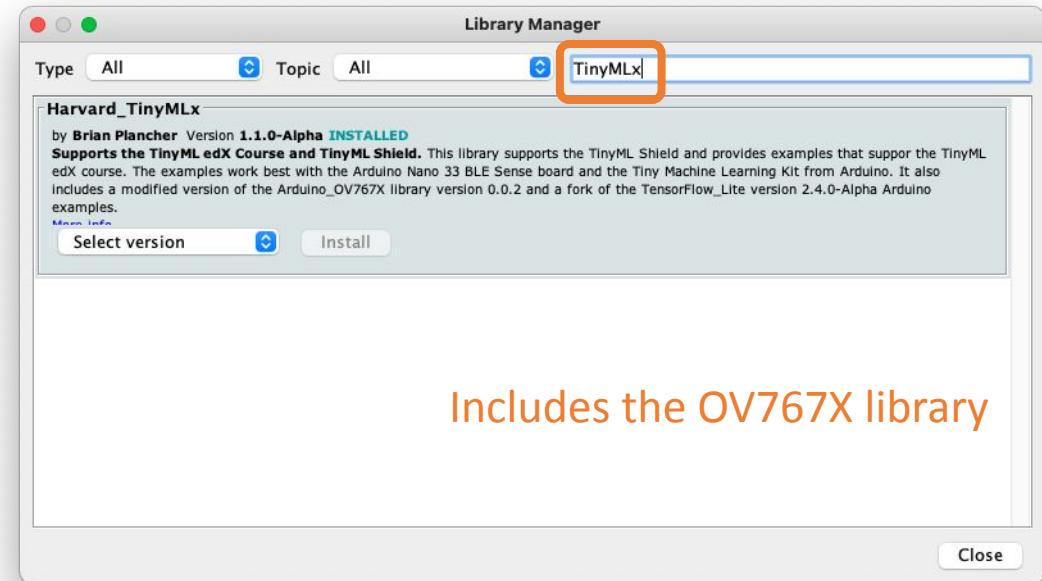
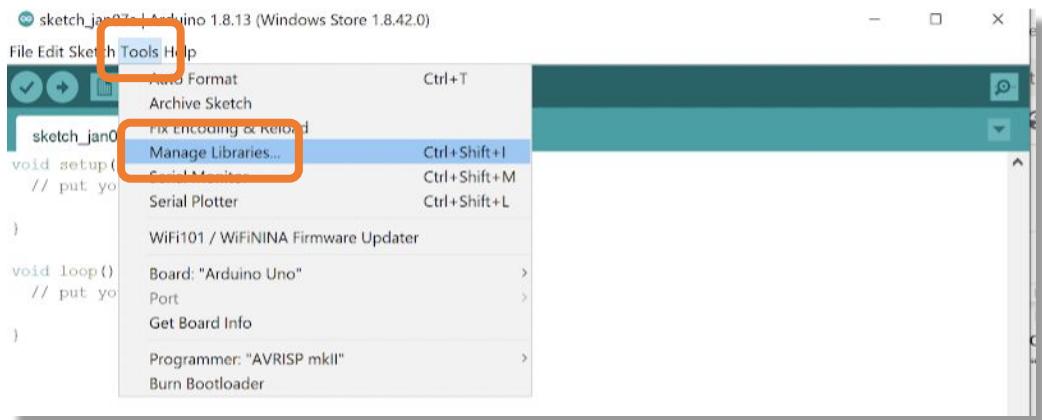
Console



Installing the Board Files

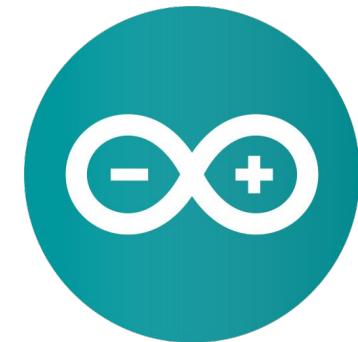


Installing the Main Libraries

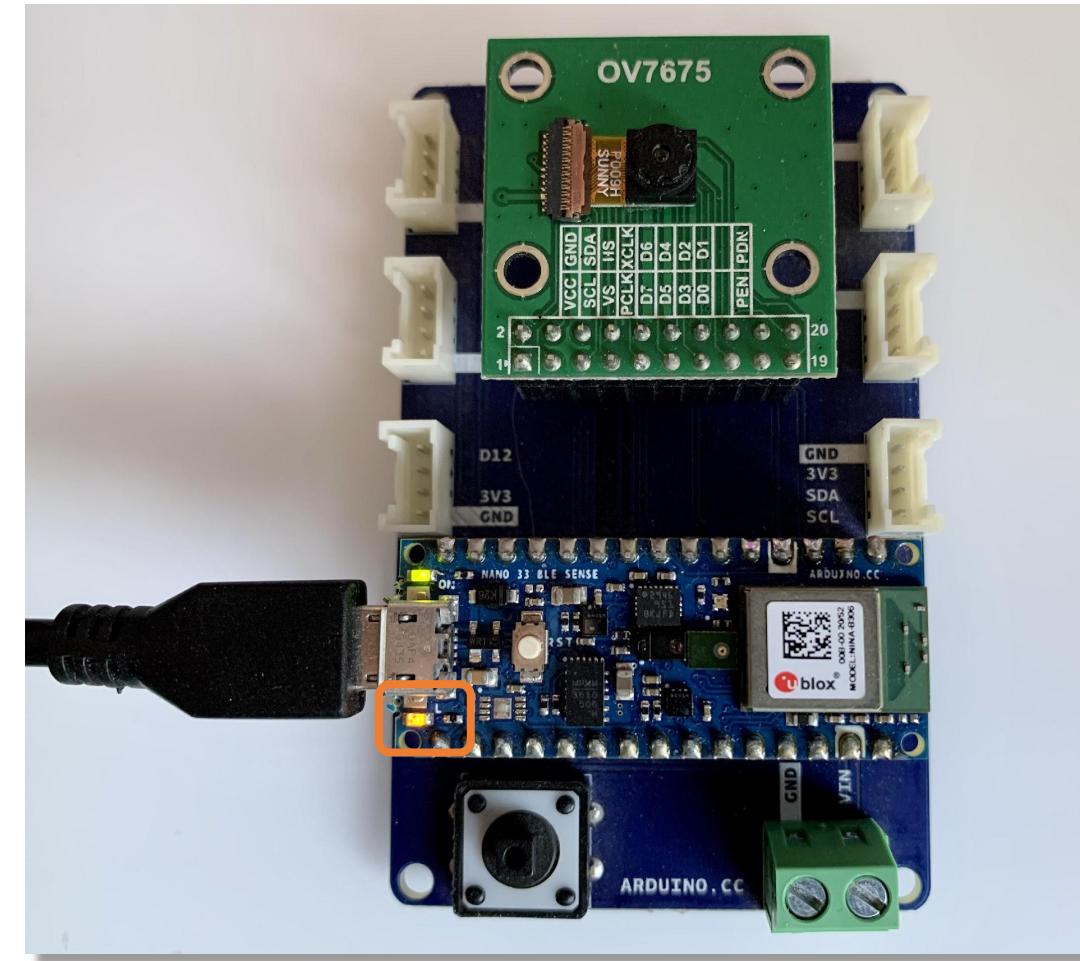
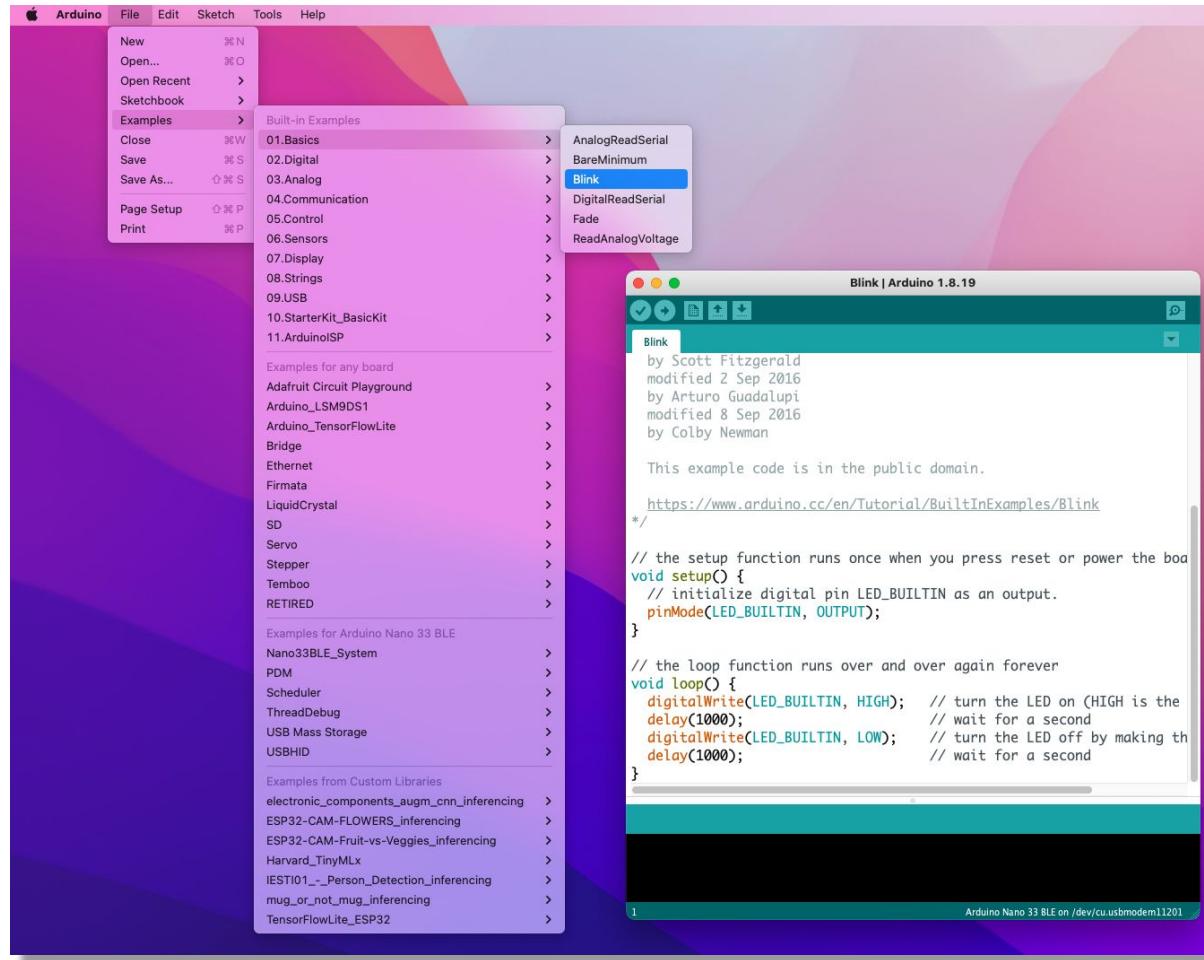


TinyML Kit Test

- 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);
    }
}

void loop() {
    if (startStream) {
        switch (imuIndex) {
            case 0:
                IMU.getAccel();
                break;
            case 1:
                IMU.getGyro();
                break;
            case 2:
                IMU.getMagnet();
                break;
        }
        startStream = false;
    }
}
```

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.

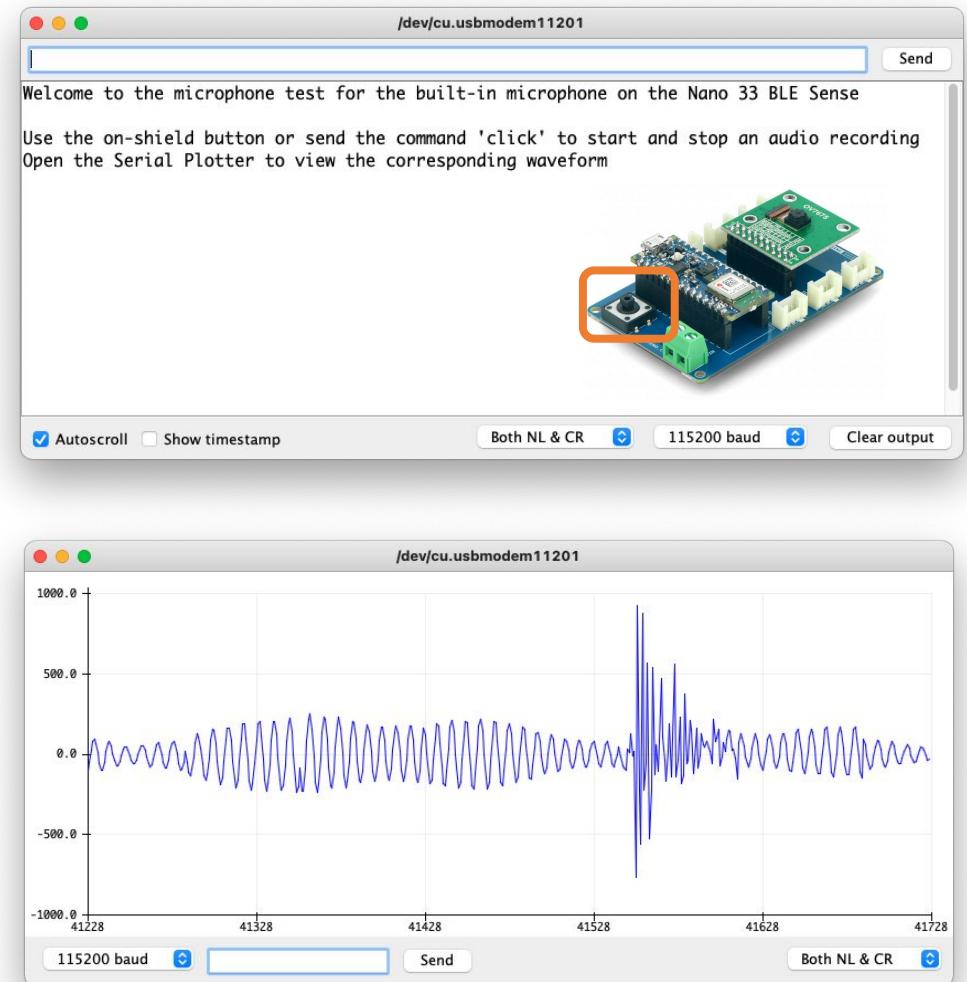


Notes: Close the Serial Monitor before open the Plotter
Repeat test for 'g' and 'm'

Testing Microphone

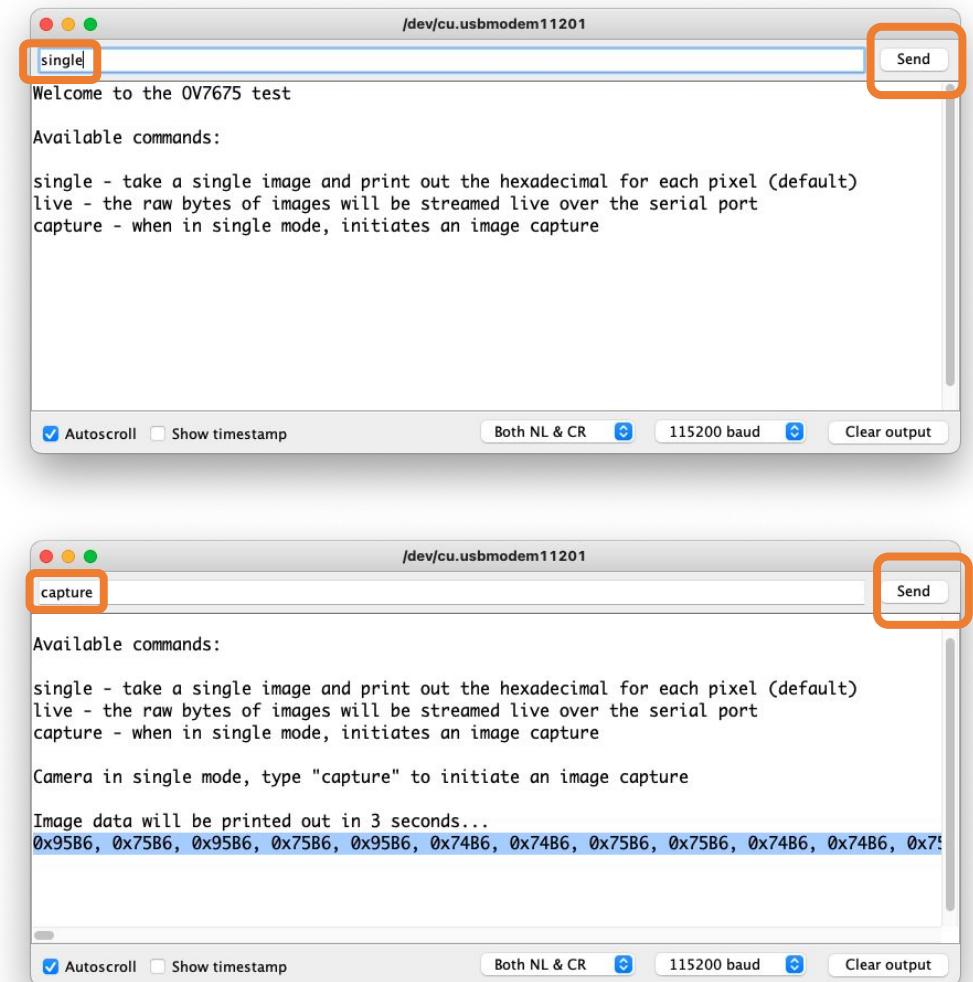
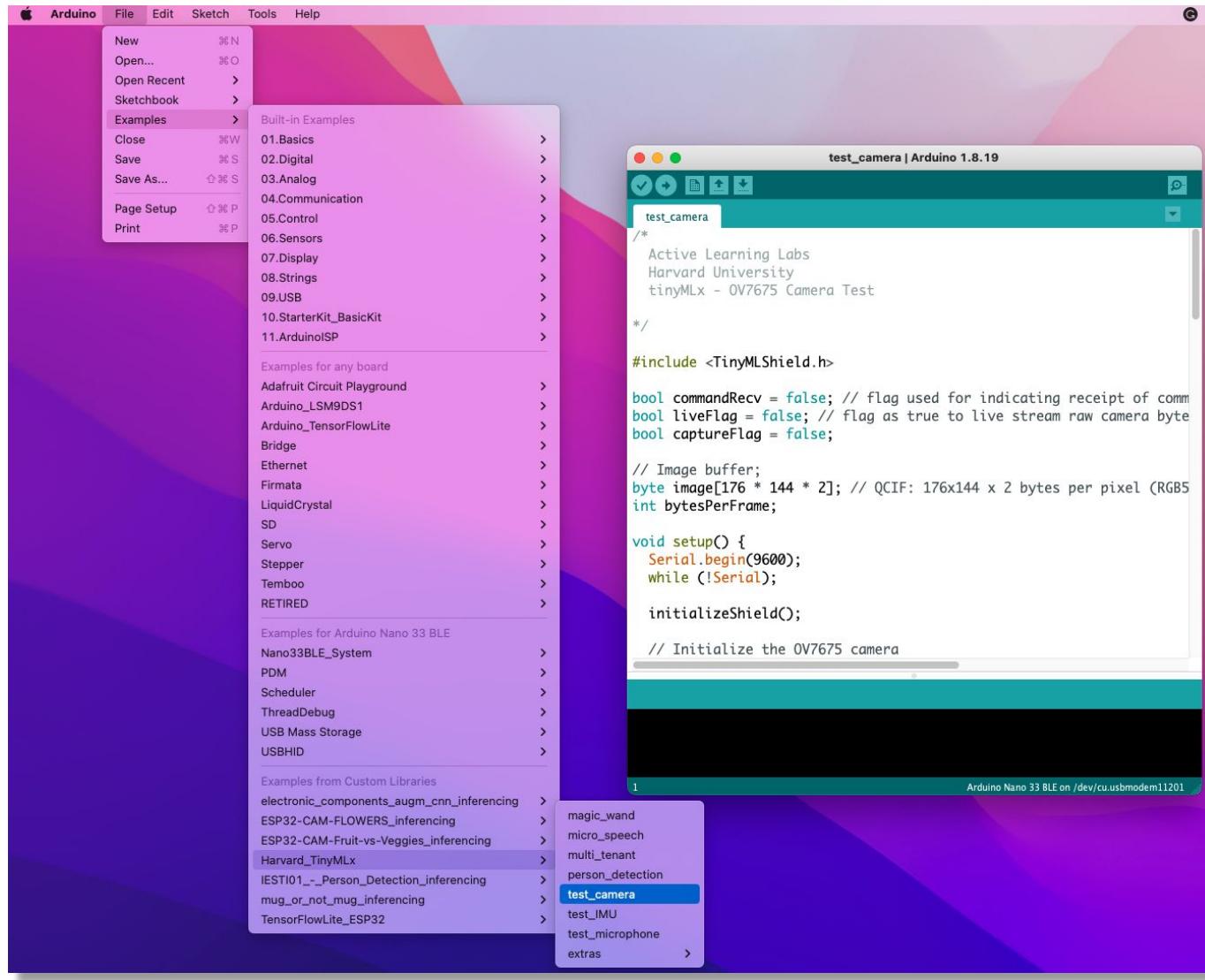
The screenshot shows the Arduino IDE interface. The menu bar includes File, Edit, Sketch, Tools, and Help. A context menu is open over the 'test_microphone' sketch, highlighting the 'extras' option. The 'extras' submenu lists several custom libraries: magic_wand, micro_speech, multi_tenant, person_detection, 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



Note: You can Press Button instead send 'capture'

Testing Camera

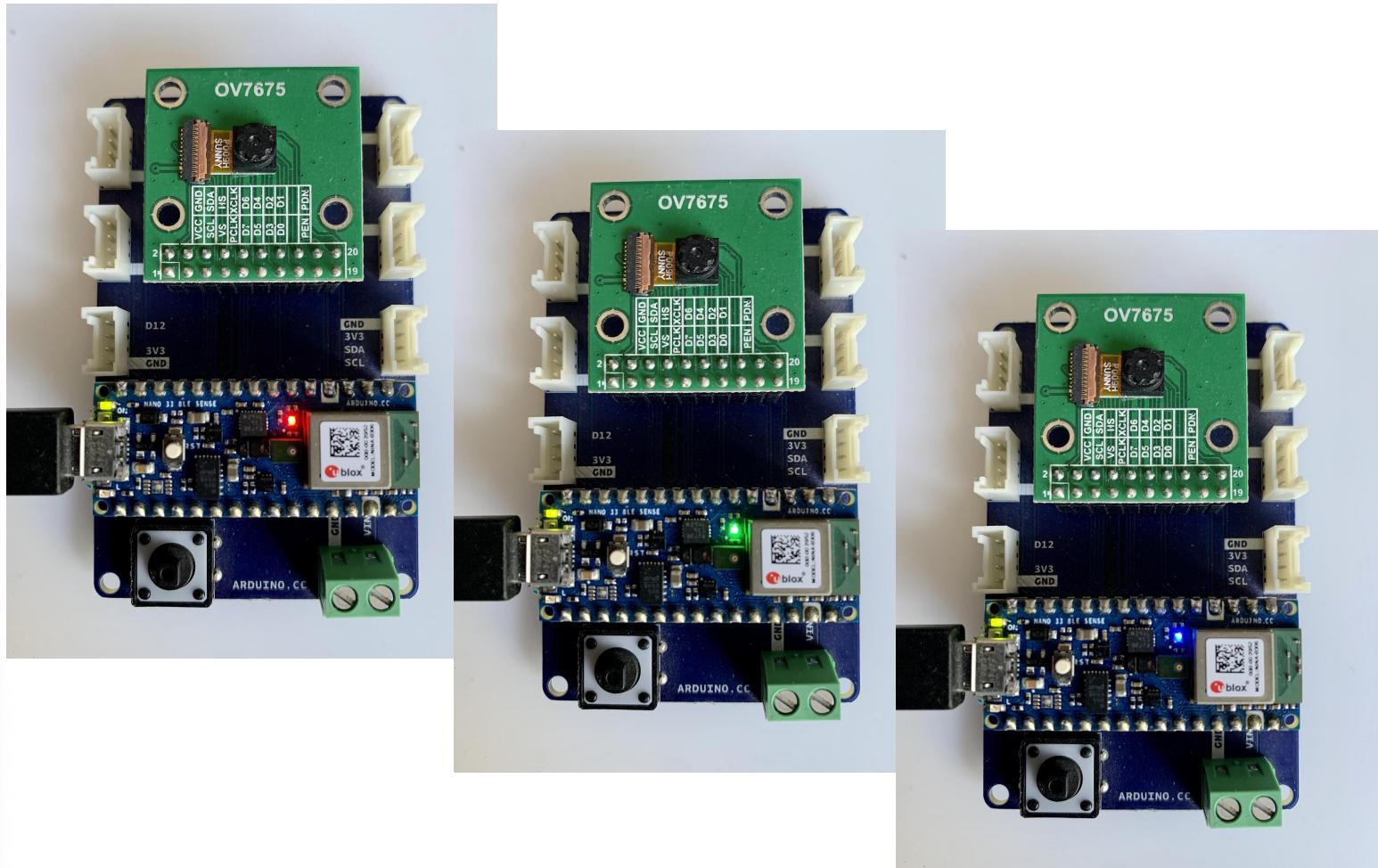


Optional Tests (RGB LEDs)

blink_RGB | Arduino 1.8.19

```
void setup() {  
    // Pins for the built-in RGB LEDs on the Arduino Nano 33 BLE Sense  
    pinMode(LED_R, OUTPUT);  
    pinMode(LED_G, OUTPUT);  
    pinMode(LED_B, OUTPUT);  
  
    // Note: The RGB LEDs are ON when the pin is LOW and off when HIGH.  
    digitalWrite(LED_R, HIGH);  
    digitalWrite(LED_G, HIGH);  
    digitalWrite(LED_B, HIGH);  
}  
  
void loop() {  
    digitalWrite(LED_R, LOW);  
    delay(1000);  
    digitalWrite(LED_R, HIGH);  
    delay(1000);  
  
    digitalWrite(LED_G, LOW);  
    delay(1000);  
    digitalWrite(LED_G, HIGH);  
    delay(1000);  
  
    digitalWrite(LED_B, LOW);  
    delay(1000);  
    digitalWrite(LED_B, HIGH);  
    delay(1000);  
}  
  
Done uploading.  
Done in 0.001 seconds  
Write 83944 bytes to flash (21 pages)  
[=====] 100% (21/21 pages)  
Done in 3.378 seconds
```

Arduino Nano 33 BLE on /dev/cu.usbmodem11201



Optional Tests (KeyWord Spotting)

micro_speech | Arduino 1.8.19

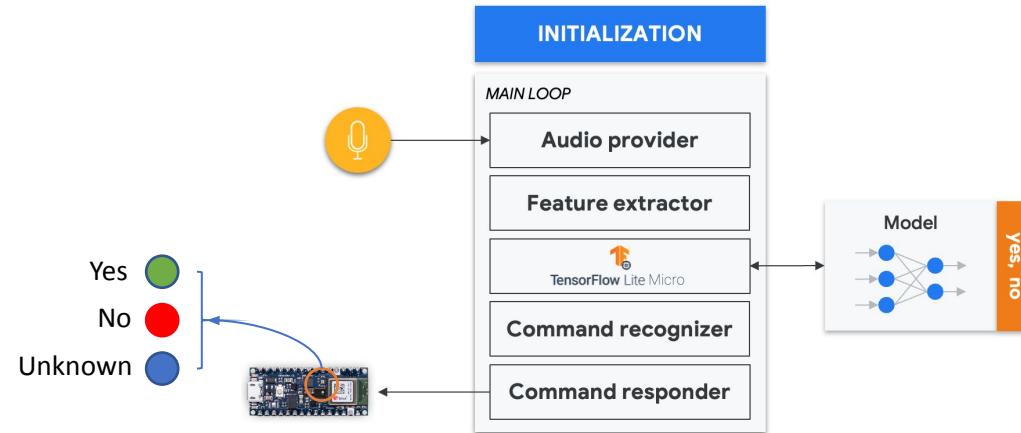
```
#include <TensorFlowLite.h>

#include "main_functions.h"

#include "audio_provider.h"
#include "command_responder.h"
#include "feature_provider.h"
#include "micro_features_micro_model_settings.h"
#include "micro_features_model.h"
#include "recognize_commands.h"
#include "tensorflow/lite/micro/micro_error_reporter.h"
#include "tensorflow/lite/micro/micro_interpreter.h"
#include "tensorflow/lite/micro/micro_mutable_op_resolver.h"
#include "tensorflow/lite/schema/schema_generated.h"
#include "tensorflow/lite/version.h"

// Globals, used for compatibility with Arduino-style sketches.
namespace {
tflite::ErrorReporter* error_reporter = nullptr;
```

Done in 0.001 seconds
Write 171992 bytes to flash (42 pages)
[=====] 100% (42/42 pages)
Done in 6.733 seconds



```
Heard silence (204) @1408ms
Heard yes (204) @6416ms
Heard yes (201) @8784ms
Heard unknown (207) @11280ms
Heard yes (209) @16656ms
Heard no (201) @25312ms
Heard no (201) @28608ms
Heard unknown (202) @35552ms
```

Optional Tests (Person Detection)

person_detection | Arduino 1.8.19

person_detection arduino_detection_responder.cpp arduino_image_provider.cpp arduir _main

```
#include <TensorFlowLite.h>
#include "main_functions.h"
#include "detection_responder.h"
#include "image_provider.h"
#include "model_settings.h"
#include "person_detect_model_data.h"
#include "tensorflow/lite/micro/micro_error_reporter.h"
#include "tensorflow/lite/micro/micro_interpreter.h"
#include "tensorflow/lite/micro/micro_mutable_op_resolver.h"
#include "tensorflow/lite/schema/schema_generated.h"
#include "tensorflow/lite/version.h"

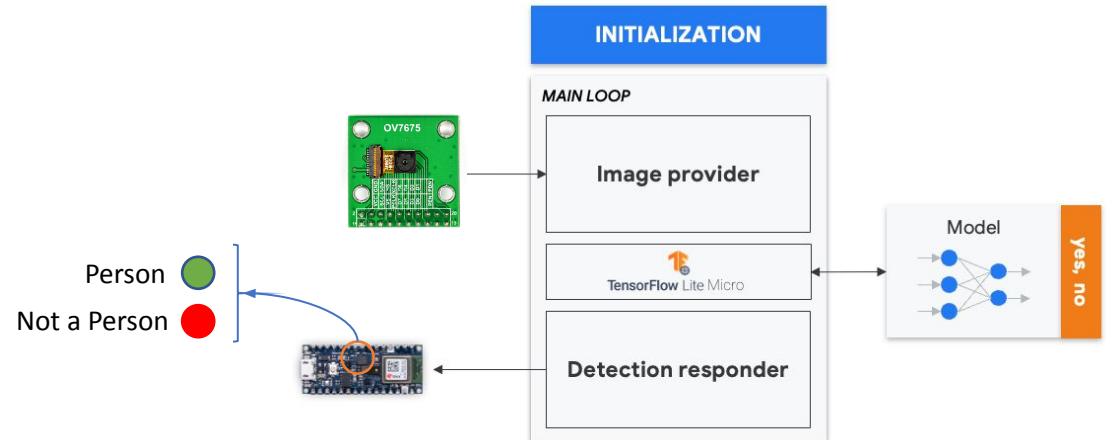
// Globals, used for compatibility with Arduino-style sketches.
namespace {
tflite::ErrorReporter* error_reporter = nullptr;
const tflite::Model* model = nullptr;
tflite::MicroInterpreter* interpreter = nullptr;
}

Done uploading.

Done in 0.001 seconds
Write 451984 bytes to flash (111 pages)
[=====] 100% (111/111 pages)
Done in 17.863 seconds
```

1

Arduino Nano 33 BLE on /dev/cu.usbmodem11201



/dev/cu.usbmodem11201

Send

```
Person score: -37 No person score: 37
Person score: -39 No person score: 39
Person score: 10 No person score: -10
Person score: 2 No person score: -2
Person score: 0 No person score: 0
Person score: 22 No person score: -22
Person score: 22 No person score: -22
Person score: 21 No person score: -21
Person score: 9 No person score: -9
Person score: -1 No person score: 1
Person score: 14 No person score: -14
Person score: 12 No person score: -12
Person score: -40 No person score: 40
Person score: -34 No person score: 34
```

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

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