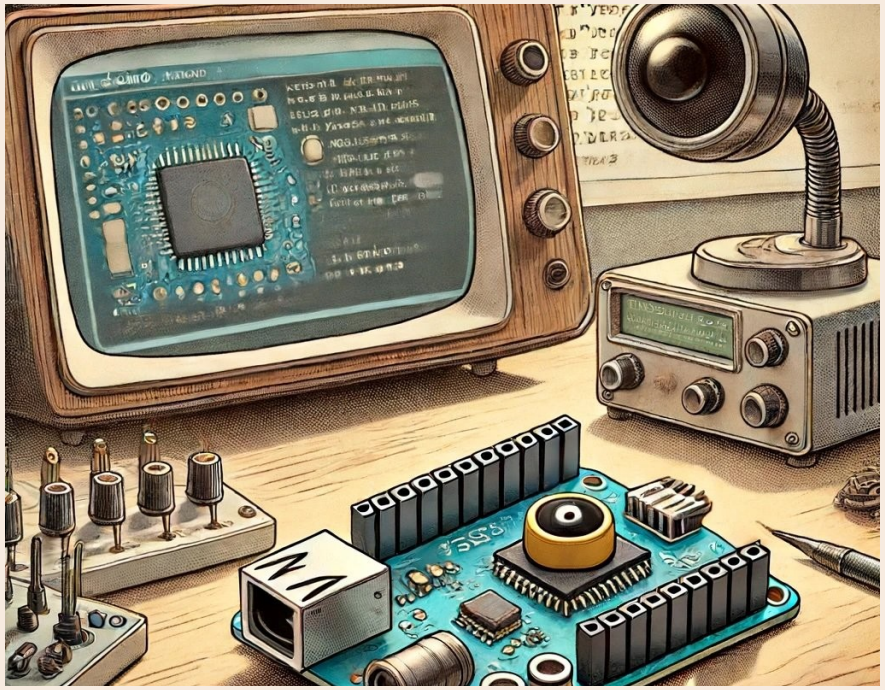
CHAPTER 1 · TINYML ON SEEED

XIAOML Kit Setup

Unbox it, connect it, test every sensor



XIAO ESP32S3 SENSE

SENSECRAFT AI

ARDUINO IDE

46 slides · 1 lab · XIAO ESP32S3 Sense

AIEng4D

01

WHERE WE ARE • PART I

Test the hardware before you trust the model

When a model fails, you want to know it is the model, not a loose camera.

Hardware | No code | Arduino IDE | Sensors

LEARNING OBJECTIVES

What you will be able to do

01

Know the kit

The XIAO ESP32S3 Sense, the expansion board, and how they fit together.

02

Run a model with no code

SenseCraft AI, in the browser, from plugging in to the first inference.

03

Set up the Arduino IDE

The right board package, the right board, the right port, and a blink.

04

Test every sensor

Microphone, IMU, OLED display, Wi-Fi, and camera, each with its own sketch.

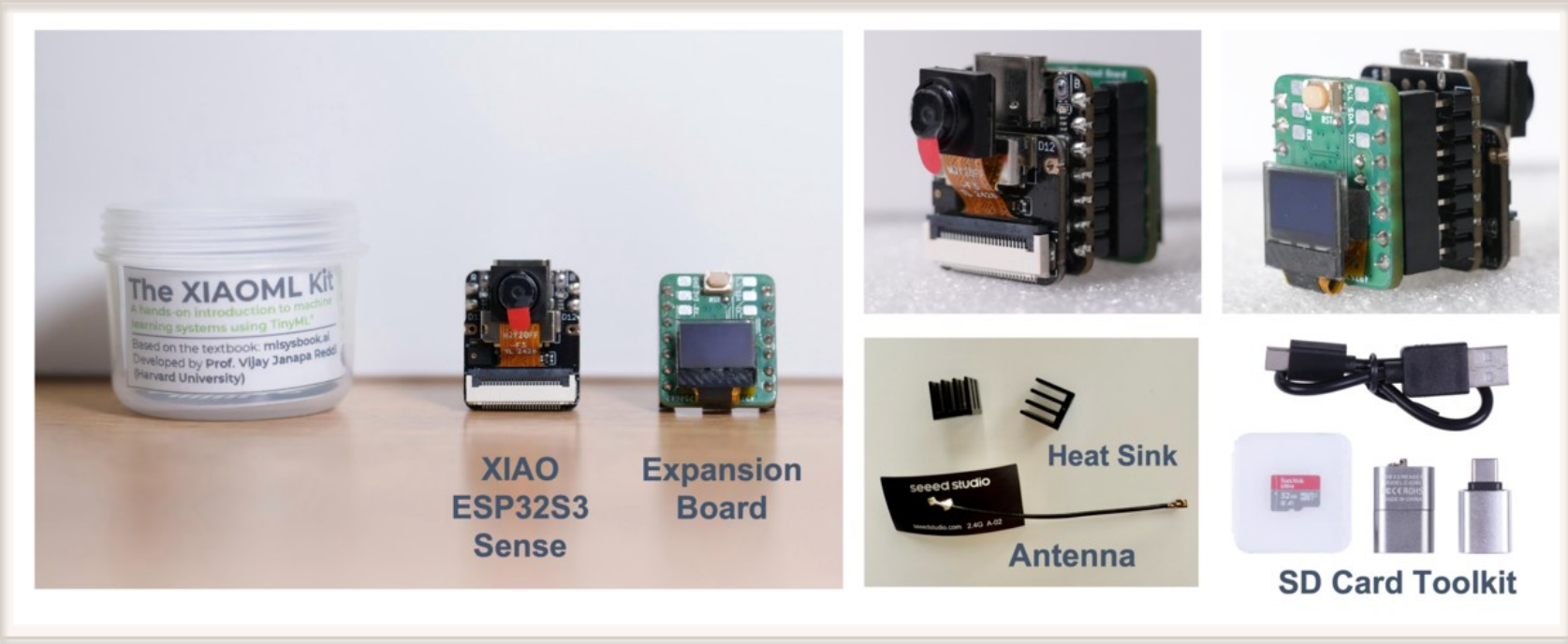
WHY THIS BOARD

A 240 MHz dual-core processor with 8 MB of PSRAM beside it. That memory is what makes vision possible here.

For comparison, the Arduino Nano 33 BLE Sense has 256 KB of RAM for everything, images included.

IN THE BOX

The XIAOML Kit



1. XIAO ESP32S3 Sense, the main board, with camera, microphone, and SD card slot
2. Expansion board with a 6-axis IMU and a 0.42-inch OLED display
3. USB-C cable, Wi-Fi antenna, and two heat sinks
4. SD card and USB adapter — check your box, some kits ship without them

CORE BOARD

The core board, its camera, and its antenna



1. The XIAO ESP32S3, with the Sense shield clipped on top
2. The detachable camera module, on a flat ribbon cable
3. The Wi-Fi and Bluetooth antenna, on a tiny U.FL connector

DATASHEET IN ONE SLIDE

XIAO ESP32S3 Sense, key features

FEATURE	VALUE	WHY IT MATTERS
Processor	ESP32S3, dual-core Xtensa LX7, 240 MHz	Arduino and MicroPython support
Memory	8 MB PSRAM, 8 MB flash	room for frames and models
Camera	OV2640 or OV3660, detachable	up to 1600 × 1200; we use 96 × 96
Microphone	digital PDM microphone	keyword spotting labs
Storage	microSD slot, up to 32 GB FAT32	log images and audio
Wireless	2.4 GHz Wi-Fi and BLE	U.FL antenna connector
Power	Li-ion charger, deep sleep at 14 µA	battery projects
Size	21 × 17.5 mm	the classic XIAO form factor

Older boards ship with an OV2640 camera; newer ones with an OV3660. Both work in every lab.

MEMORY

Three kinds of memory, three jobs

01

SRAM · 512 KB

Inside the chip. The fastest memory, and the smallest.

02

PSRAM · 8 MB

Outside the chip, on the module. A little slower, sixteen times larger. Frames and tensors live here.

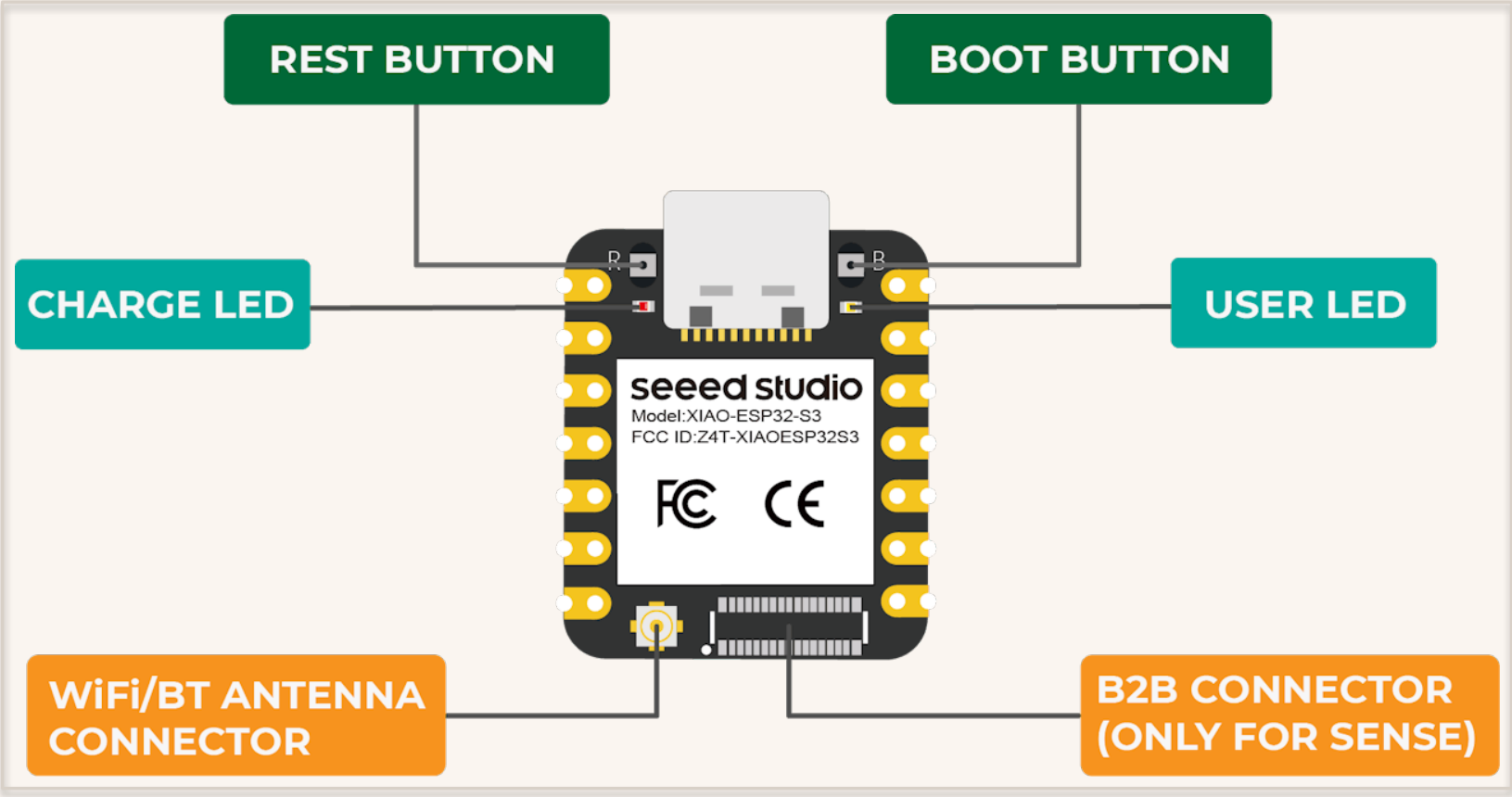
03

Flash · 8 MB

Holds your program and the model weights. It keeps them when the power goes off.

BUTTONS AND LEDS

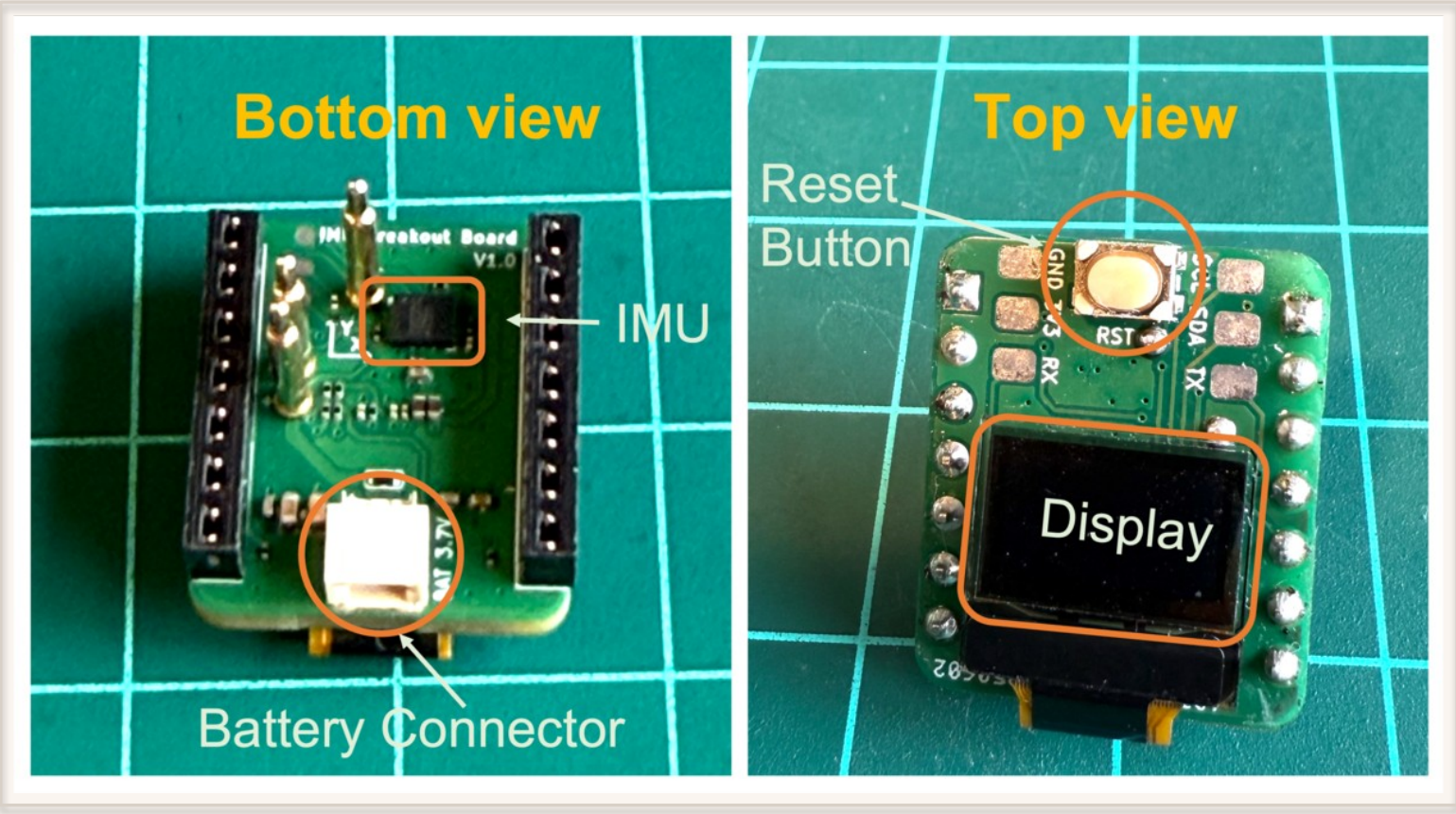
Small buttons, one user LED



1. Reset button: press once to restart the program
2. Boot button: hold it while you plug in the cable if an upload fails
3. User LED on GPIO 21, with inverted logic — LOW turns it on
4. Charge LED, and the B2B connector used only by the Sense shield

EXPANSION BOARD

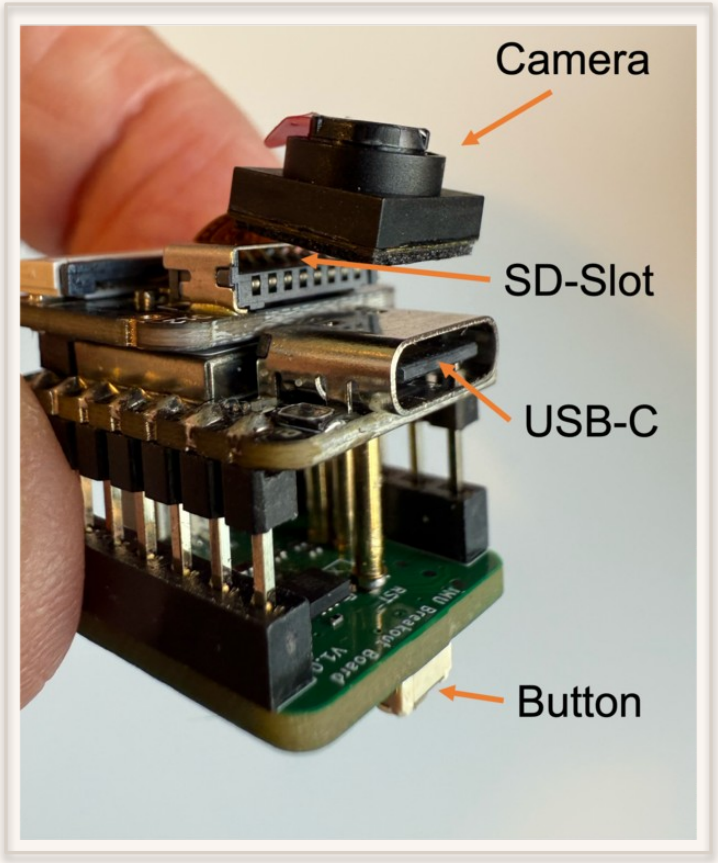
Motion and a display, on a board that clips underneath



1. IMU LSM6DS3TR-C: 3-axis accelerometer and 3-axis gyroscope, I2C at 0x6A
2. 0.42-inch OLED, 72 × 40 pixels, SSD1306 controller, I2C at 0x3C
3. A reset button you can actually reach, wired to the EN pin
4. A white connector for a 3.7 V battery, wired to BAT+ and BAT-

ASSEMBLY

Mount it the right way round



1. Camera, SD slot, and USB-C on one side; the expansion board's button on the other

2. Press the boards together until the pins are fully seated

3. You only need the expansion board for motion, the OLED, or a battery

THREE DETAILS

Three small things that can cost you an hour

01

Heat sinks block the expansion board

Leave them off if you use the expansion board. Without it, they help with long camera and Wi-Fi sessions.

02

The antenna goes on one side first

Remove the Sense shield, hook one side of the U.FL connector, then press the other side gently.

03

Close the serial port

Only one program can own the USB port. Close the Arduino IDE before SenseCraft, and the other way round.

02

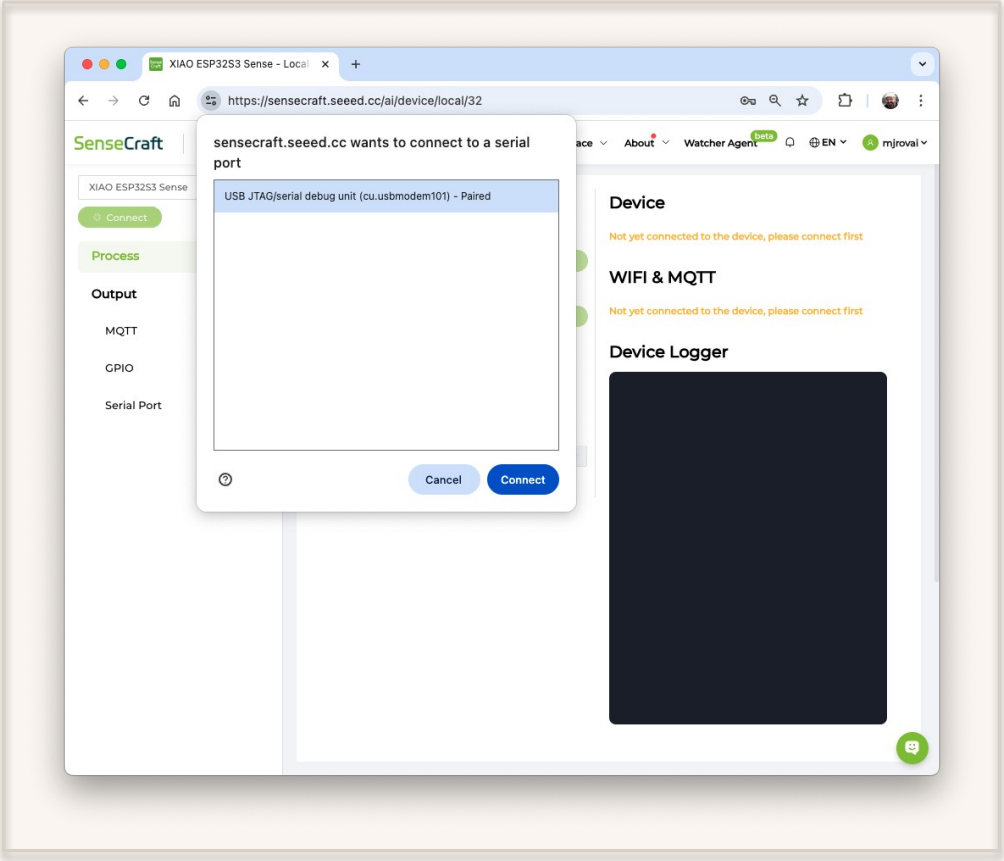
NO CODE

A model running in five minutes

SenseCraft AI flashes a pre-trained model from the browser.

SENSECRAFT AI · CONNECT

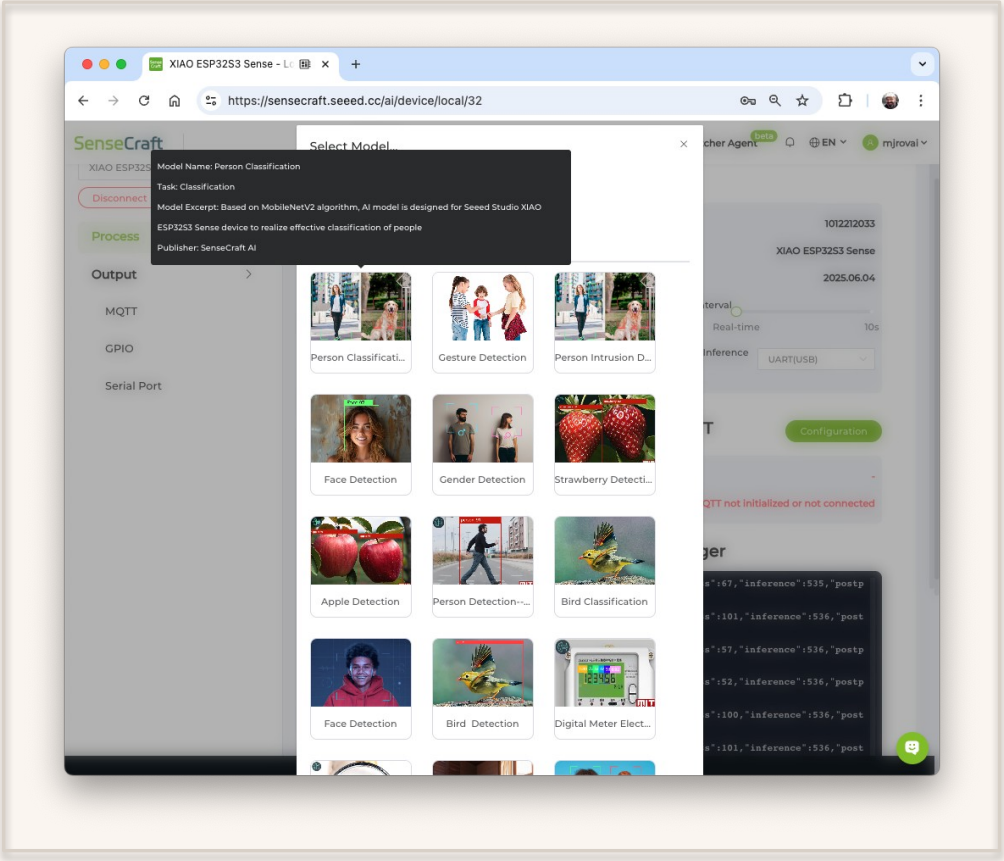
Open the workspace in Chrome and connect the board



1. Plug the kit into your computer with the USB-C cable
2. Open the SenseCraft AI workspace in Chrome, and sign in
3. Pick the serial port in the pop-up and press Connect
4. Safari can't talk to USB from a web page. Use Chrome.

MODEL LIBRARY

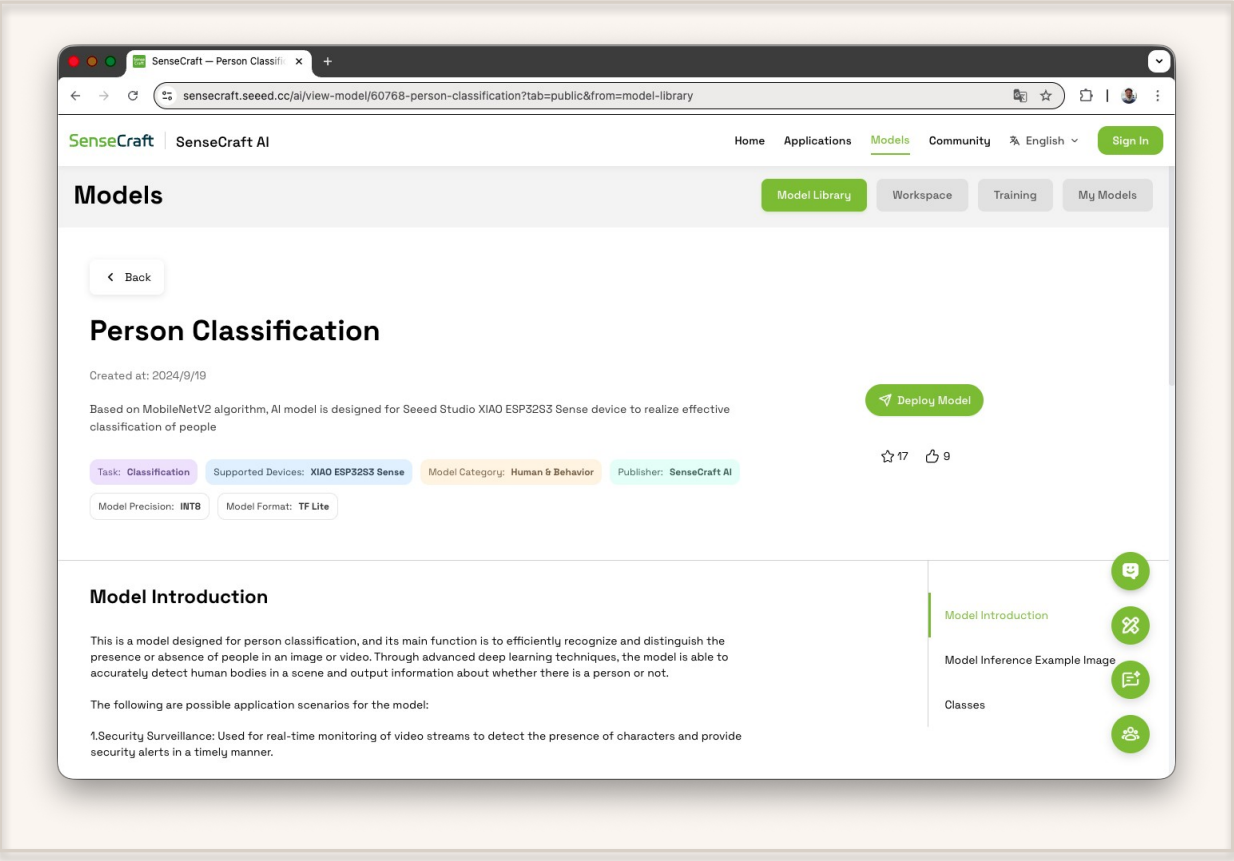
Filter by task and by device, then read the card



1. Select Model opens the library of ready-to-use models
2. Filter by task and by supported device — the same library serves Jetson boards
3. Hover a model to see its task, algorithm, and sometimes its metrics

PERSON CLASSIFICATION

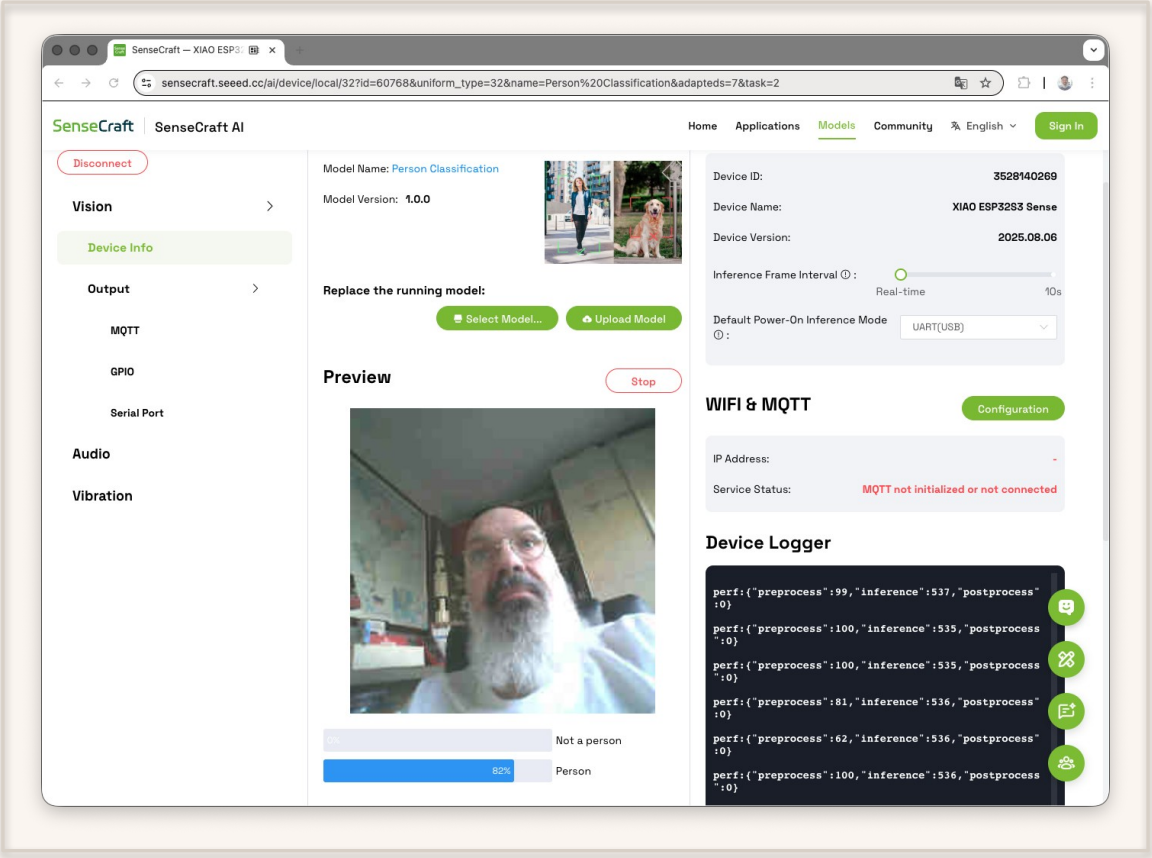
Two classes, person and not a person



1. An int8 MobileNet V2, built with TensorFlow Lite
2. Two classes, person and not a person
3. Deploy Model flashes new firmware and the model to the board

INFERENCE

The camera works, and the model sees you



1. Preview shows the live feed and the top class
2. Inference Frame Interval sets how often the model runs
3. The Device Logger prints the timing of every frame

DEVICE LOGGER

About 536 ms per inference

```
perf:{"preprocess":99,"inference":537,"postprocess":0}
perf:{"preprocess":100,"inference":535,"postprocess":0}
perf:{"preprocess":100,"inference":535,"postprocess":0}
perf:{"preprocess":81,"inference":536,"postprocess":0}
perf:{"preprocess":62,"inference":536,"postprocess":0}
perf:{"preprocess":100,"inference":536,"postprocess":0}
```

PRE-PROCESS

62–100 ms

INFERENCE

~536 ms

FRAME RATE

~1.6 FPS

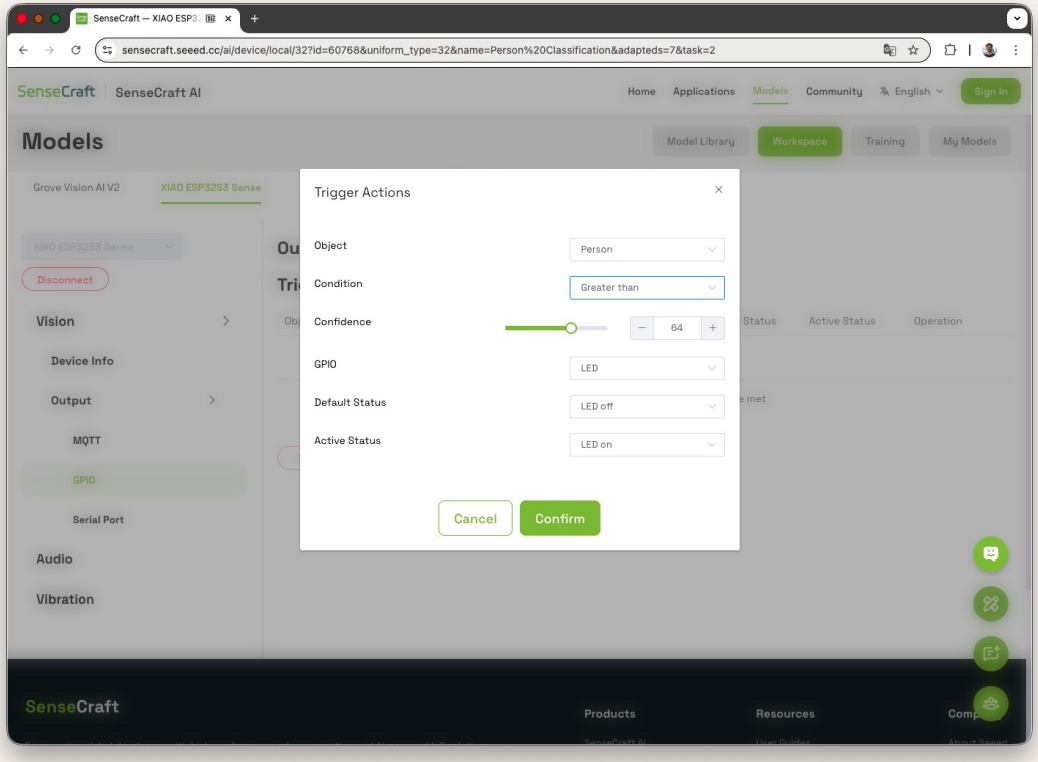
THE BUDGET

Smaller input, faster model. At 96 × 96 the same board reaches about 3.4 frames per second.

Measured in Chapter 2 with a MobileNet V2 0.35 trained at 96 × 96 — 81 ms pre-processing, 205 ms inference.

POST-PROCESSING · GPIO

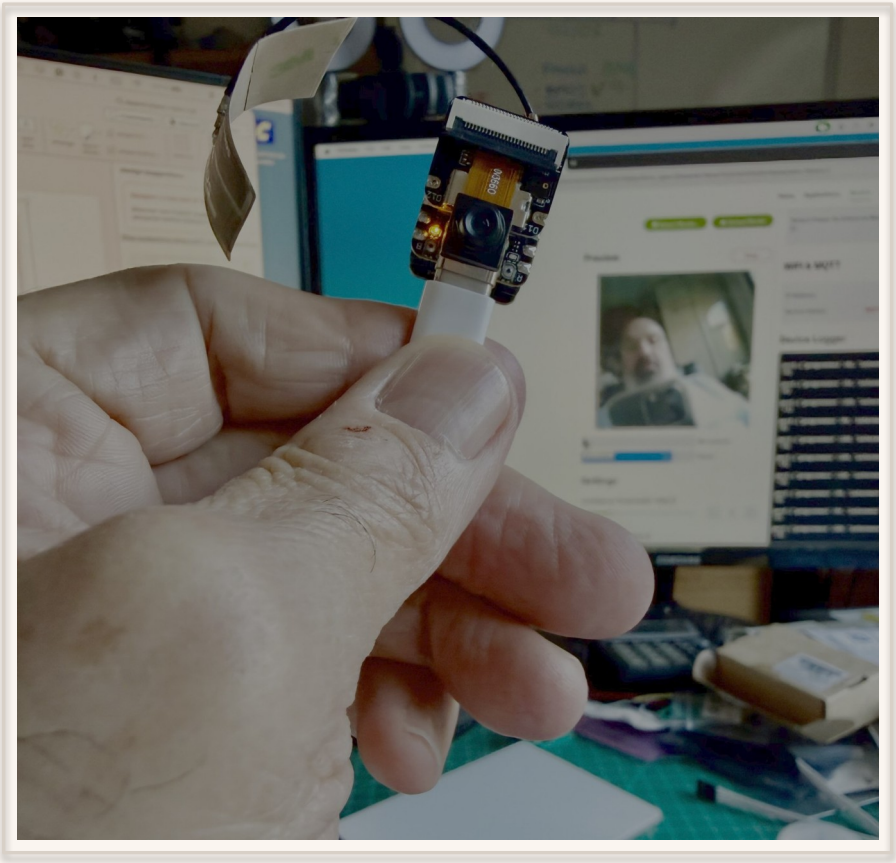
If person above 60 %, turn the LED on



1. Output, then GPIO, then add a trigger action
2. Condition: the class person, with a confidence above 60 %
3. Action: the built-in LED, or a pin such as D0, D1, or D2
4. Confirm, then Send to store the rule on the board

THE RULE IN ACTION

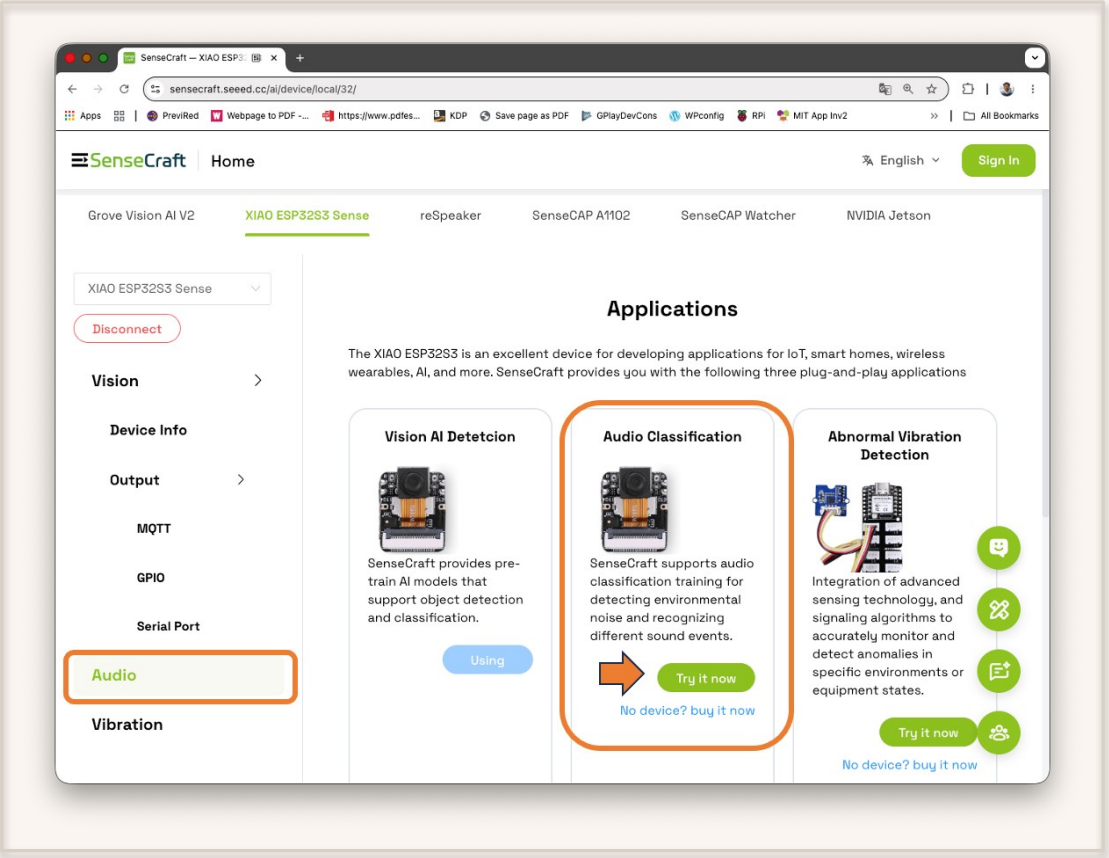
A person in frame, and the LED lights



1. The camera sees a person with more than 60 % confidence
2. The user LED turns on — no code was written
3. Swap the LED for a relay, and the same rule switches a lamp

SENSECRAFT AI · AUDIO

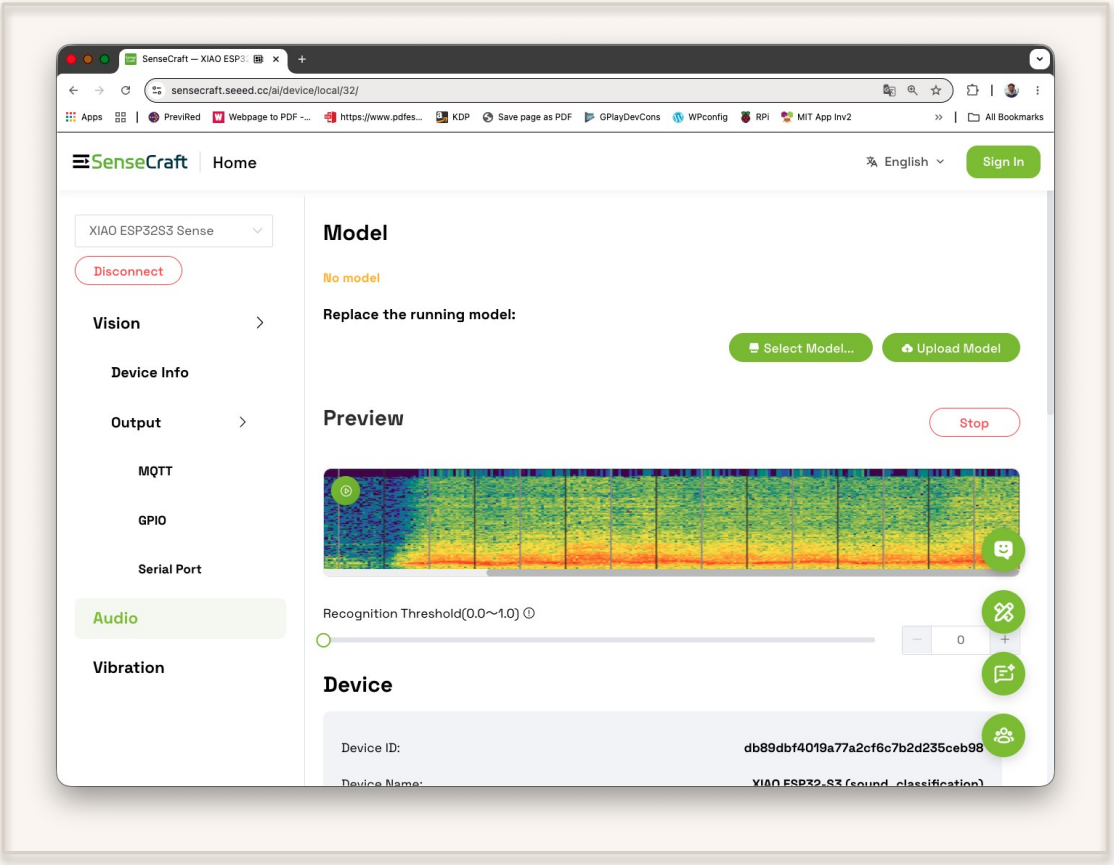
Switch to Audio to test the microphone



1. SenseCraft expects a vision model by default — select Audio on the left
2. Choose Audio Classification, then Try it now
3. New firmware is flashed to read the microphone

SPECTROGRAM

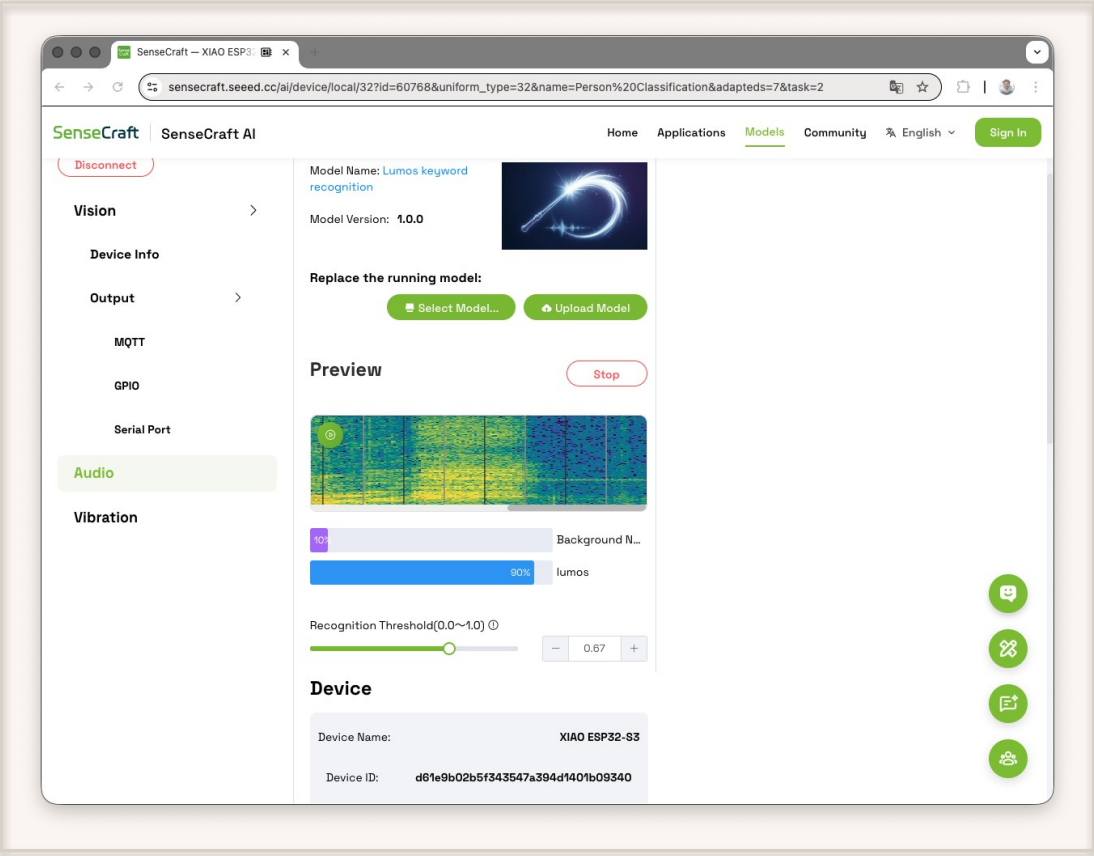
You can see your voice



1. Time runs left to right, frequency bottom to top
2. Speak, and bright bands appear; stay silent, and they fade
3. A spectrogram is an image, so sound can be classified like an image

KEYWORD SPOTTING

A ready-made keyword model, Lumos



1. Select Model, then the Lumos keyword recognition model
2. Two classes, background noise and the word lumos
3. Recognition Threshold sets how sure the model must be

COLLECT AND TRAIN

SenseCraft can also collect data and train

01

Capture and export

Record images or audio from the board, label them, and export a ZIP.

02

Train in the browser

Free account. The image trainer uses a fixed MobileNet V2 at 224 × 224.

03

Need more control?

For a smaller input or a smaller alpha, train in Edge Impulse, starting in Chapter 2.

03

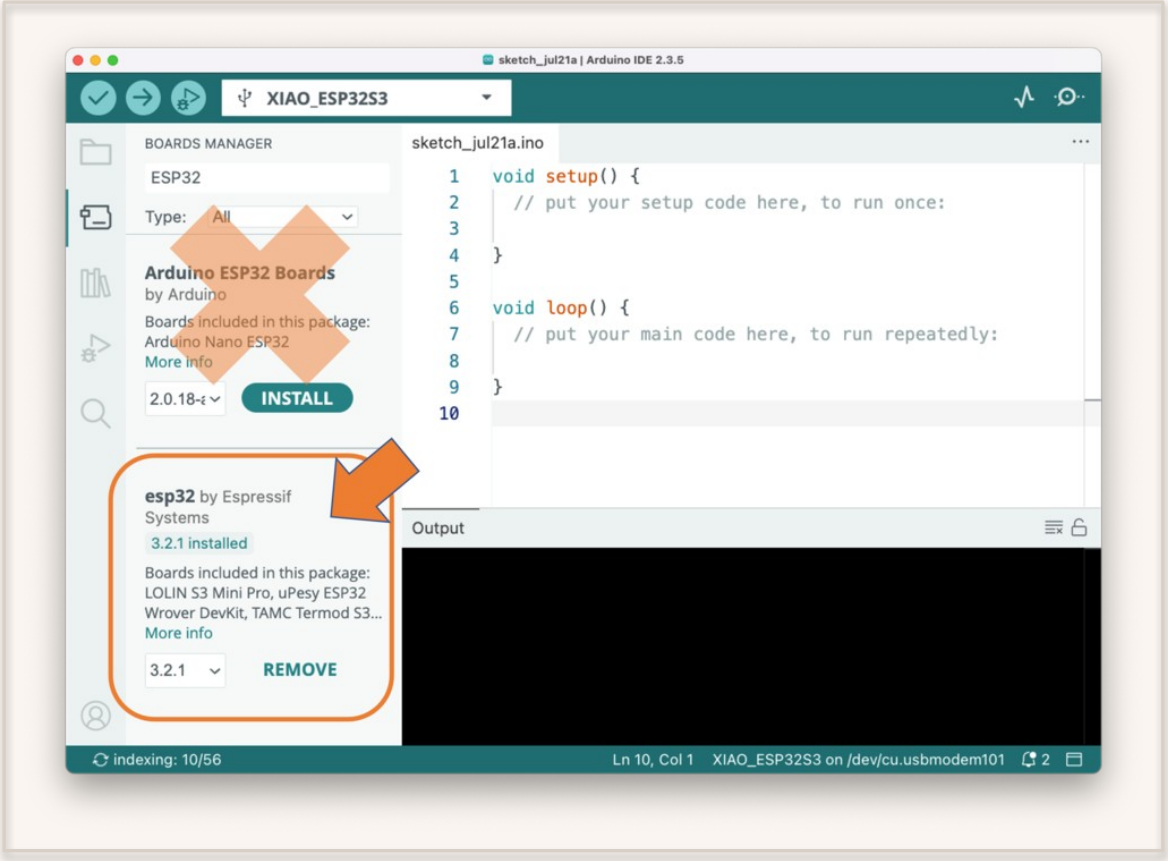
ARDUINO IDE

Install the board, then blink

The first program on any new hardware is a blinking LED.

BOARDS MANAGER

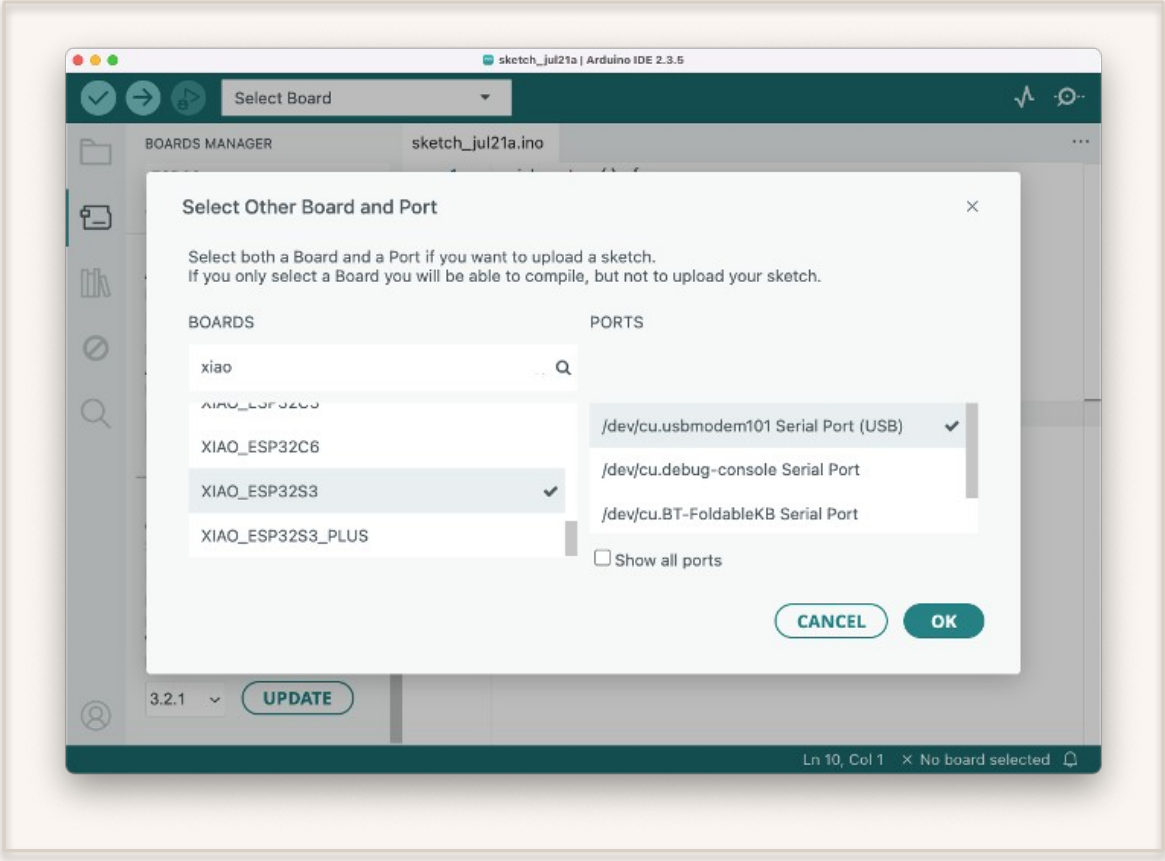
Install esp32 by Espressif Systems



1. Open the Boards Manager and search for esp32
2. Install esp32 by Espressif Systems
3. Not Arduino ESP32 Boards — that package is for the Arduino Nano ESP32
4. Edge Impulse code may fail on core 3.x; then use 2.0.17

BOARD AND PORT

Select XIAO_ESP32S3 and its port



1. Select Board, then type xiao or esp32s3
2. Choose XIAO_ESP32S3 and the port where it is connected
3. The status bar confirms the board and the port

BLINK

The LED is on GPIO 21, and LOW turns it on

Blink.ino

```
#define LED_BUILT_IN 21 // This line is optional

void setup() {
  pinMode(LED_BUILT_IN, OUTPUT); // Set the pin as output
}

// Remember that the pins work with inverted logic
// LOW to turn on and HIGH to turn off
void loop() {
  digitalWrite(LED_BUILT_IN, LOW); //Turn on
  delay (1000); //Wait 1 sec
  digitalWrite(LED_BUILT_IN, HIGH); //Turn off
  delay (1000); //Wait 1 sec
}
```

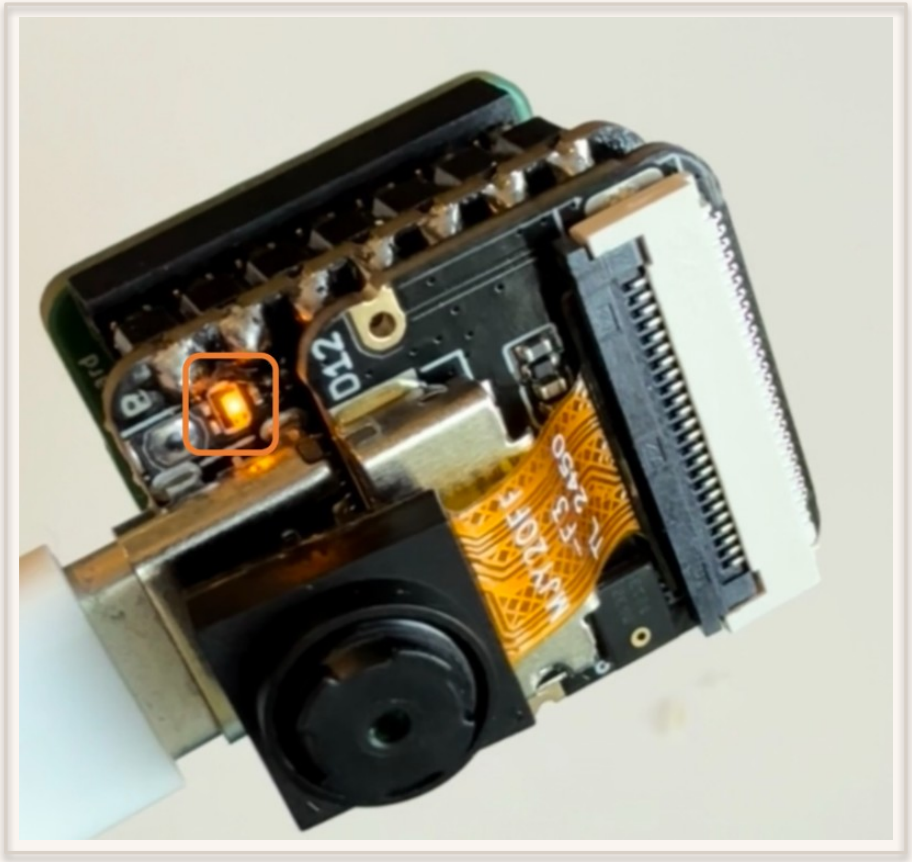
— File, Examples, Basics, Blink works as is
— LED_BUILTIN already maps to pin 21

— Verify compiles; Upload flashes the board

— Inverted logic — LOW is on, HIGH is off

IT WORKS

One second on, one second off



04

SENSORS

One sketch per sensor

Microphone, SD card, IMU, display, Wi-Fi, and camera.

MICROPHONE

Read the PDM microphone at 16 kHz

XiaoEsp32s3_Mic_Test.ino (ESP32 core 3.x, condensed)

```
#include <ESP_I2S.h>
I2SClass I2S;

void setup() {
  Serial.begin(115200);
  while (!Serial) {}
  // setup 42 PDM clock and 41 PDM data pins
  I2S.setPinsPdmRx(42, 41);
  // start I2S at 16 kHz with 16-bits per sample
  if (!I2S.begin(I2S_MODE_PDM_RX, 16000,
                 I2S_DATA_BIT_WIDTH_16BIT, I2S_SLOT_MODE_MONO)) {
    Serial.println("Failed to initialize I2S!");
    while (1); // do nothing
  }
}

void loop() {
  int sample = I2S.read();  // read a sample
  if (sample && sample != -1 && sample != 1) {
    Serial.println(sample);
  }
}
```

— The microphone speaks PDM over the I2S peripheral

— Clock on pin 42, data on pin 41

— 16 kHz, 16 bits, mono — the format of the audio labs

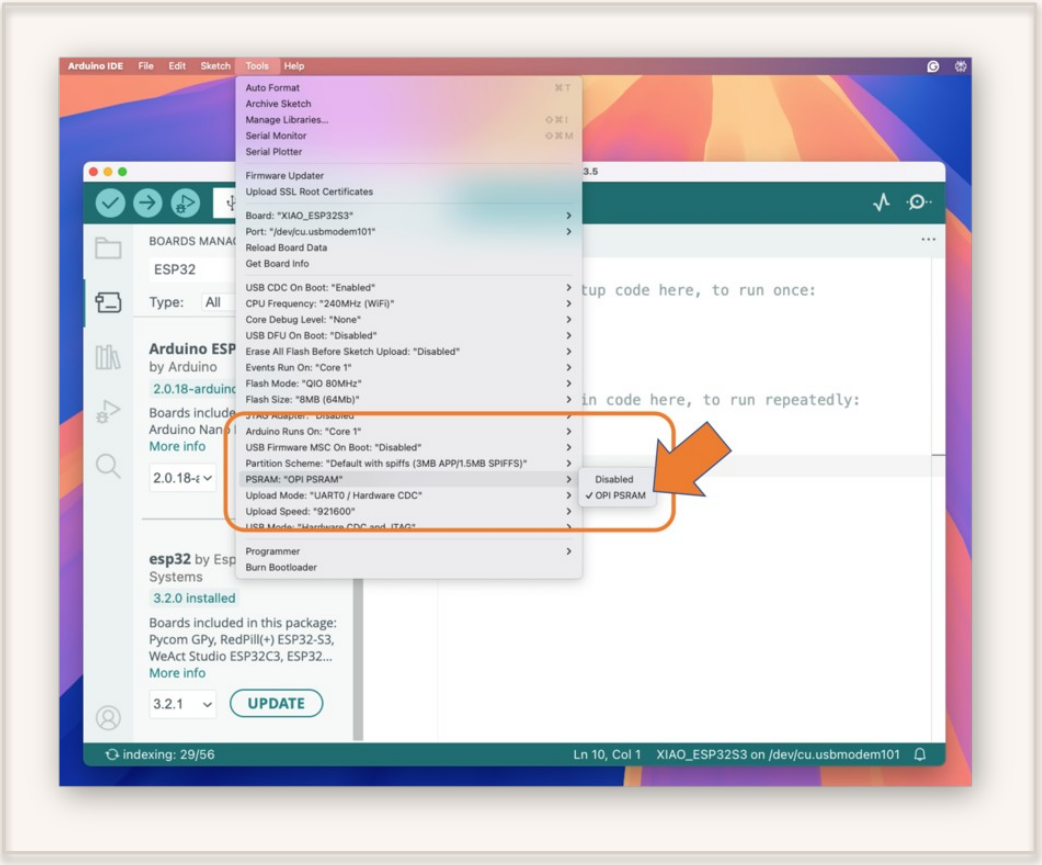
SERIAL PLOTTER

Speak, and the curve jumps



SD CARD

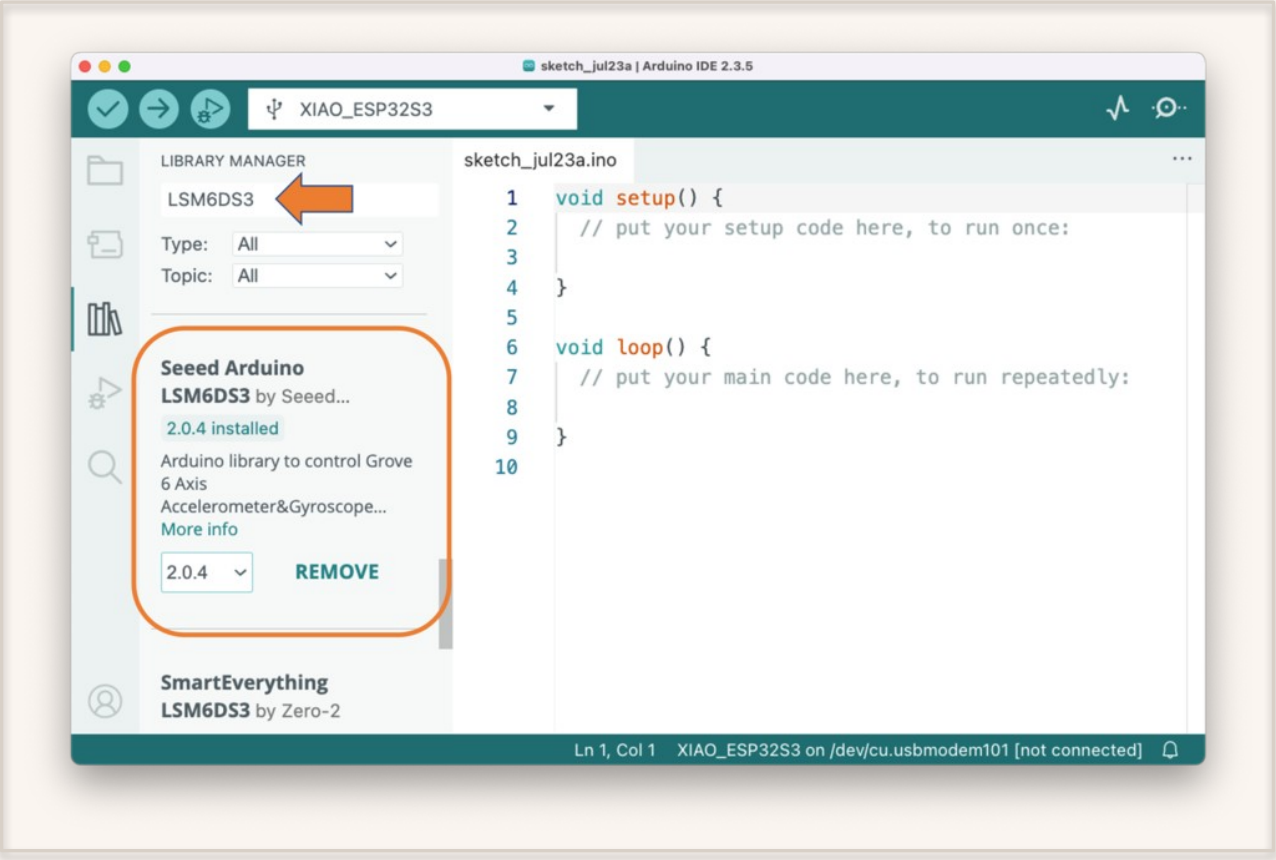
Record a WAV file, but turn on PSRAM first



1. Tools, PSRAM, OPI PSRAM — the audio buffer needs it
2. A microSD card up to 32 GB, formatted FAT32
3. The book's Wav_Record sketch records 20 seconds with recordWAV()
4. The SD card chip select is GPIO 21, shared with the LED

IMU · LIBRARY

Install Seeed Arduino LSM6DS3, not the Arduino one



1. Library Manager, search for LSM6DS3
2. Install Seeed Arduino LSM6DS3 by Seeed
3. Arduino_LSM6DS3 by Arduino is for other boards

IMU · TEST

Six values, ten times per second

IMU_test.ino (excerpt)

```
#include <LSM6DS3.h>
#include <Wire.h>

// LSM6DS3TR-C sensor is located at I2C address 0x6A
LSM6DS3 myIMU(I2C_MODE, 0x6A);

void loop() {
  // Read accelerometer data (in g-force units)
  accelX = myIMU.readFloatAccelX();
  accelY = myIMU.readFloatAccelY();
  accelZ = myIMU.readFloatAccelZ();

  // Read gyroscope data (in degrees per second)
  gyroX = myIMU.readFloatGyroX();
  gyroY = myIMU.readFloatGyroY();
  gyroZ = myIMU.readFloatGyroZ();

  // ... prints readable values, then one CSV line

  // Update rate: 10 Hz (100ms delay)
  delay(100);
}
```

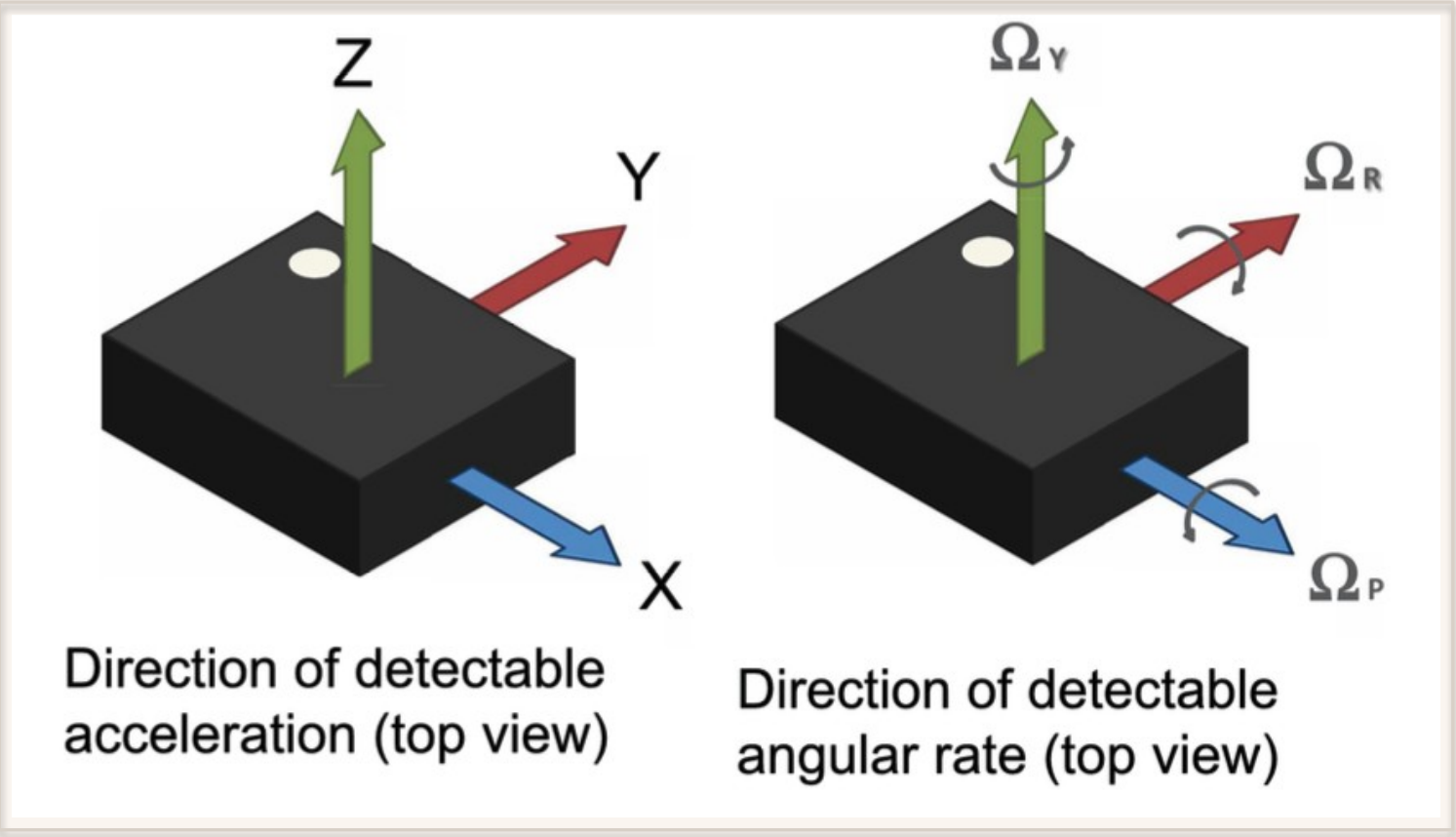
— Accelerometer at ±2 g, gyroscope at ±250 dps by default

— The CSV line feeds the Serial Plotter

— The full sketch is in the book

IMU · AXES

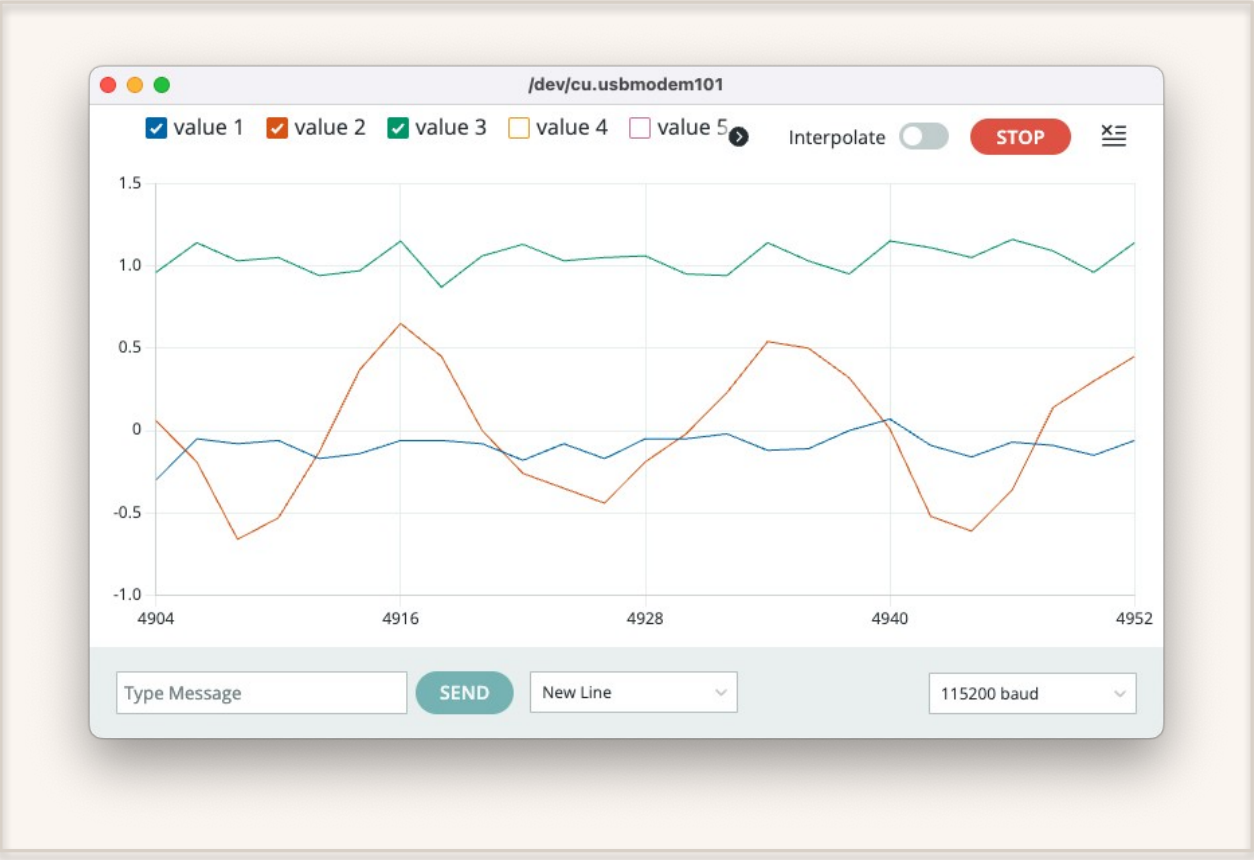
X to the right, Y forward, Z up



1. Right-hand coordinate system, seen with the point mark visible
2. The accelerometer measures acceleration along each axis, gravity included
3. The gyroscope measures rotation rate around each axis

SERIAL PLOTTER

At rest, Z reads 1 g



1. Value 3, the Z axis, sits near 1.0 g — that is gravity
2. Move the kit along Y and value 2 follows
3. Select values 4 to 6 to see the gyroscope

OLED DISPLAY

Hello World on 72 × 40 pixels

oled_test.ino (excerpt) · library U8g2 by oliver

```
#include <U8g2lib.h>
#include <Wire.h>

// SSD1306 controller, 72x40 resolution, I2C interface
U8G2_SSD1306_72X40_ER_1_HW_I2C u8g2(U8G2_R2, U8X8_PIN_NONE);

void loop() {
  // Start drawing sequence
  u8g2.firstPage();
  do {
    u8g2.setFont(u8g2_font_ncenB08_tr);
    u8g2.setCursor(8, 15);
    u8g2.print("Hello");
    u8g2.setCursor(12, 30);
    u8g2.print("World!");
    u8g2.drawFrame(2, 2, 68, 36);
  } while (u8g2.nextPage());
}
```

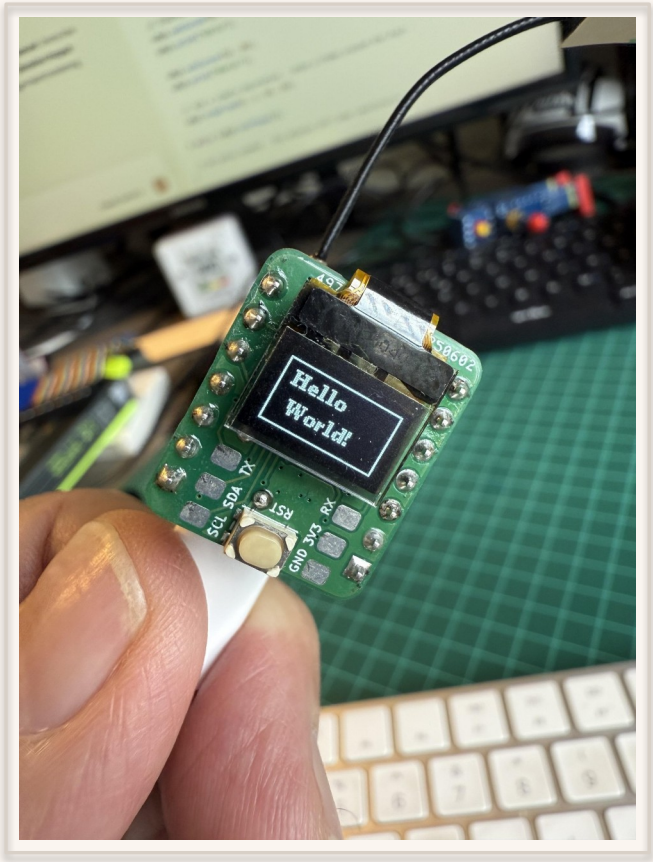
— Install U8g2 by oliver first

— U8G2_R2 rotates the image 180°

— setCursor(x, y) — y is the text baseline

OLED DISPLAY

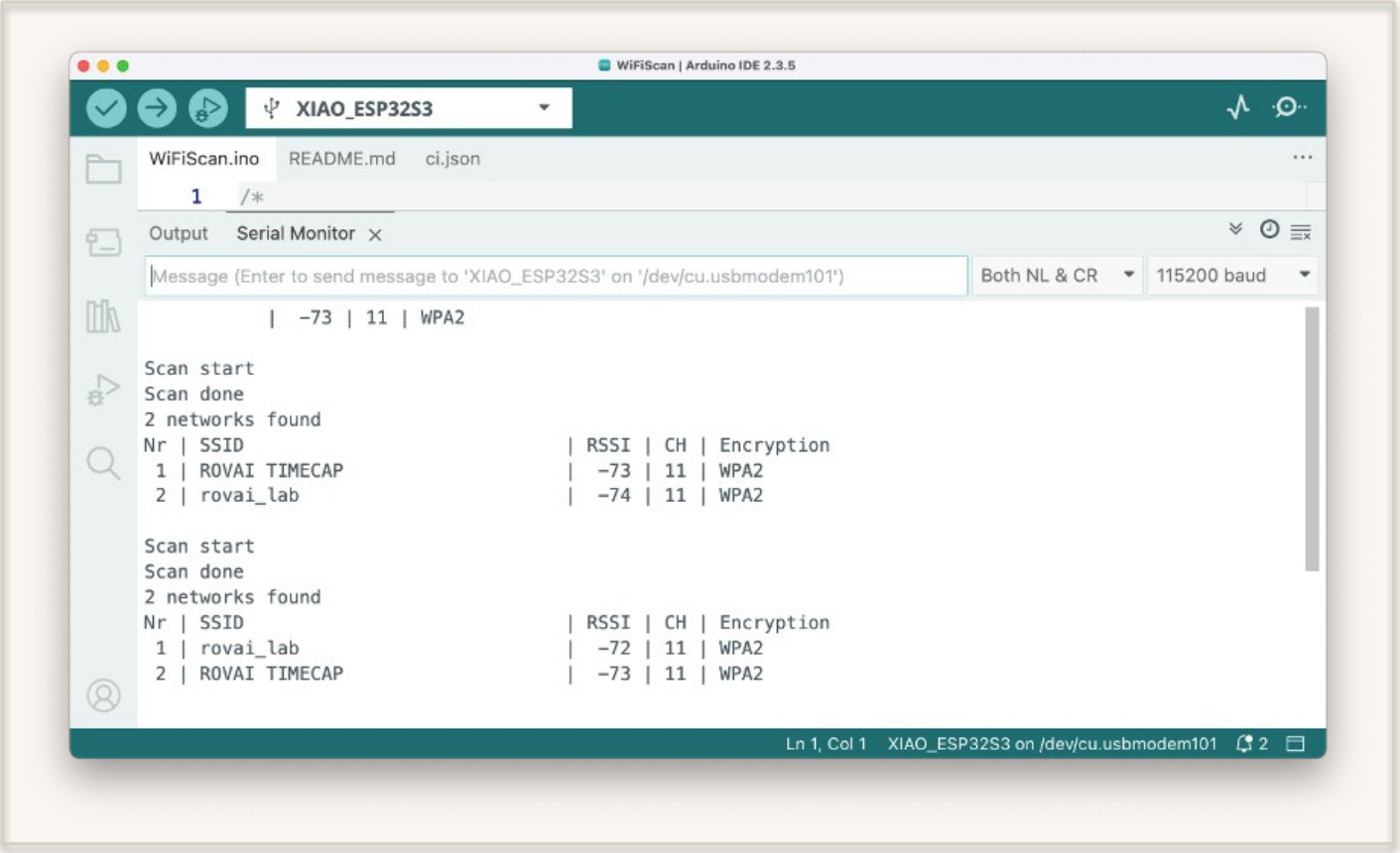
Your first post-processing screen



1. Origin at the top-left; X runs 0 to 71, Y runs 0 to 39
2. Fonts from 4 × 6 pixels to large bold
3. Frames, boxes, circles, lines, and pixels
4. If the screen stays dark, press reset once

WI-FI

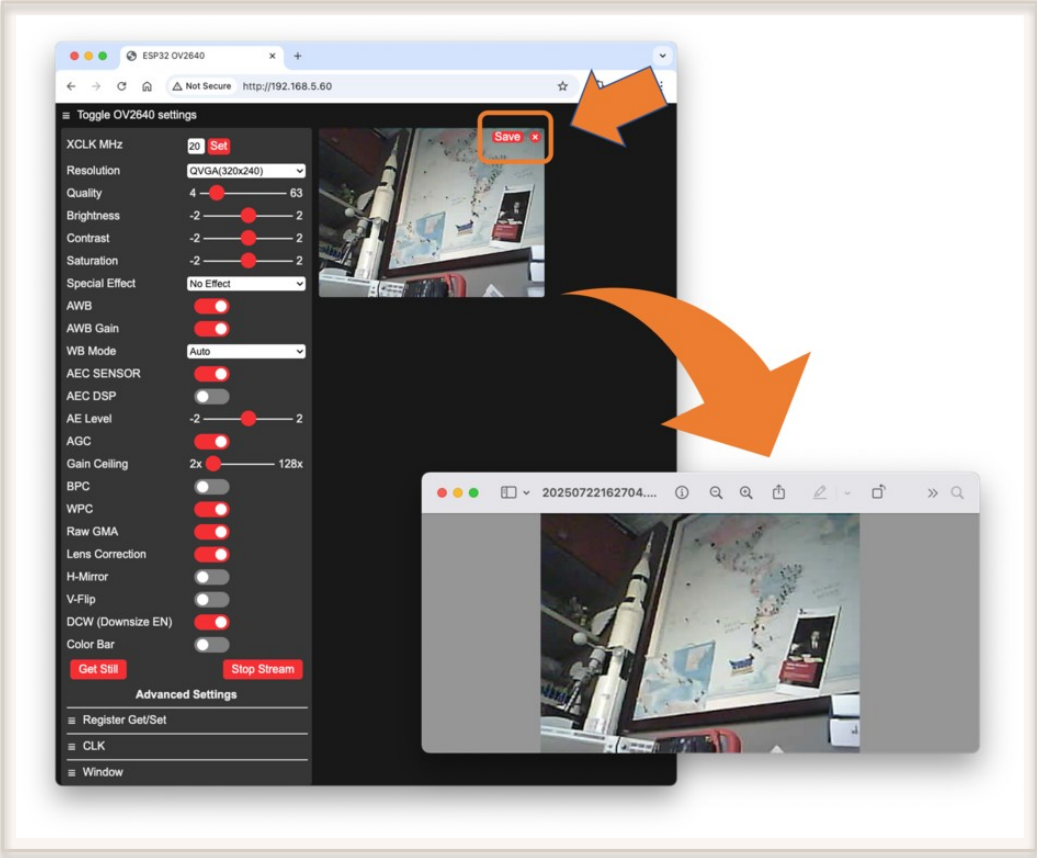
Scan the networks, then serve a page



1. Attach the antenna first for a stronger signal
2. Examples, WiFi, WiFiScan lists SSIDs and signal strength
3. SimpleWiFiServer toggles the LED from a browser — change pin 5 to pin 21

CAMERA

CameraWebServer, the camera in your browser



1. File, Examples, ESP32, Camera, CameraWebServer
2. In board_config.h, keep only CAMERA_MODEL_XIAO_ESP32S3
3. PSRAM enabled, Wi-Fi credentials entered, then upload
4. Open the address, pick QVGA, Start Stream, and Save frames

CHECKLIST

Every part tested, and how

PART	WHERE	TEST	LIBRARY
Camera	Sense shield	SenseCraft, CameraWebServer	esp32 core
Microphone	Sense shield, PDM 42/41	SenseCraft, Mic_Test, WAV to SD	ESP_I2S
User LED	GPIO 21, inverted	Blink	—
IMU LSM6DS3TR-C	expansion board, I2C 0x6A	IMU_test + Serial Plotter	Seeed Arduino LSM6DS3
OLED SSD1306	expansion board, I2C 0x3C	oled_test	U8g2 by oliver
Wi-Fi	U.FL antenna	WiFiScan, SimpleWiFiServer	WiFi

If every row works on your kit, you are ready for the labs.

SUMMARY

Three things to remember

01 Memory is why this board does vision

8 MB of PSRAM next to a 240 MHz dual core. Enable it in Tools when your sketch needs it.

02 One port, one owner

SenseCraft in Chrome, or the Arduino IDE — never both at once.

03 Input size is the first budget knob

The pre-trained model needs about 536 ms per inference. At 96 × 96, you get about 3.4 FPS.

CHECK YOURSELF

SenseCraft can't connect to your board. What do you check first?

A Reflash the board with the Arduino IDE

B Whether the Arduino IDE still holds the port, and whether you are in Chrome

C Whether the model is too large for the PSRAM

Only one program can own the serial port, and the page needs Chrome to reach USB. Close the IDE, use Chrome, and try again.

The interactive deck gives per-answer feedback.

Questions?

Next — Image Classification, where you train your own model to tell a box from a wheel.

