

# ESP32 - Motion Classification

Prof. Marcelo José Rovai  
UNIFEI - Federal University of Itajubá, Brazil  
TinyML4D Academic Network Co-Chair

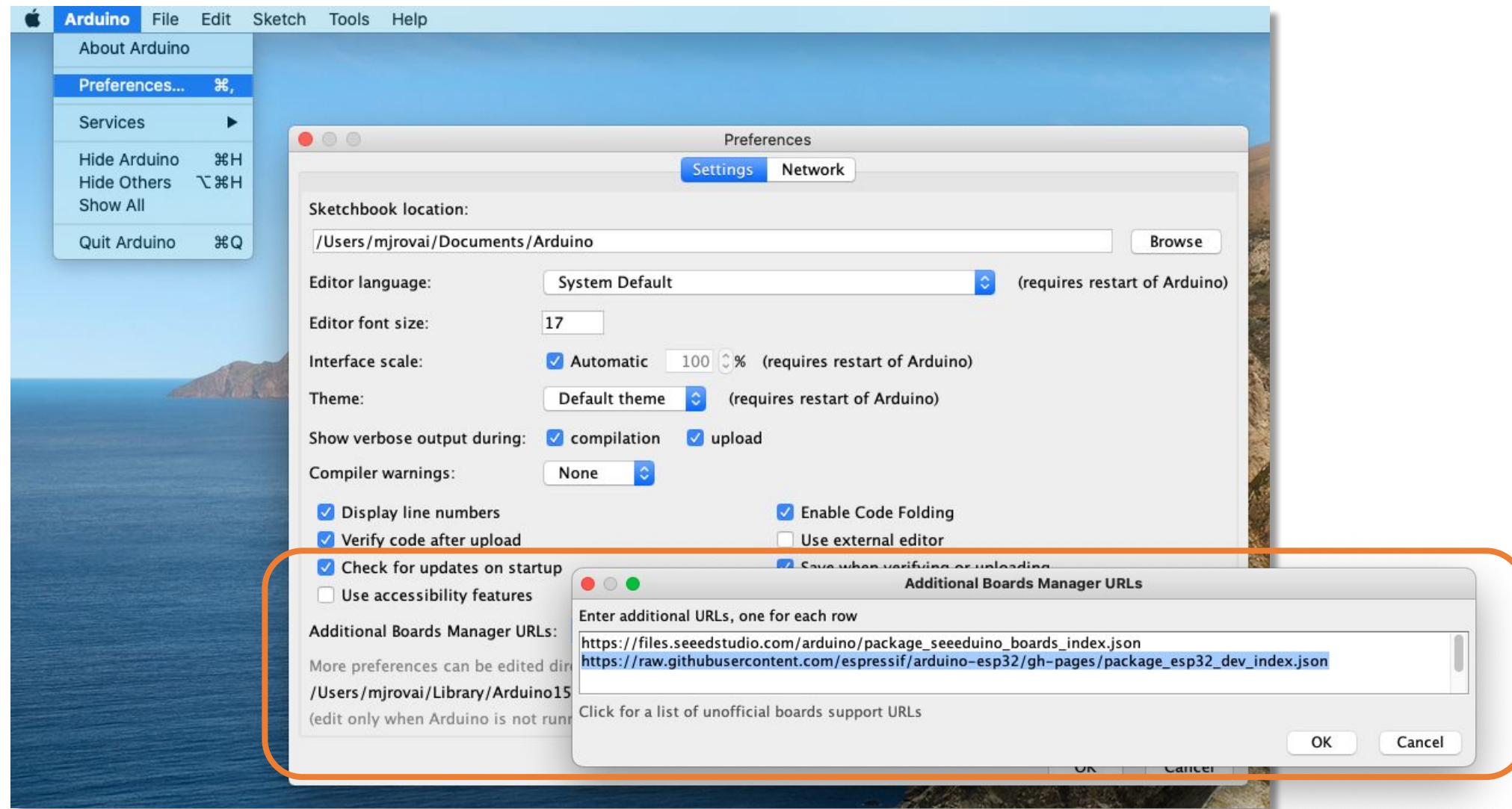


**UNIFEI**

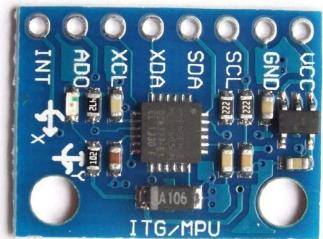
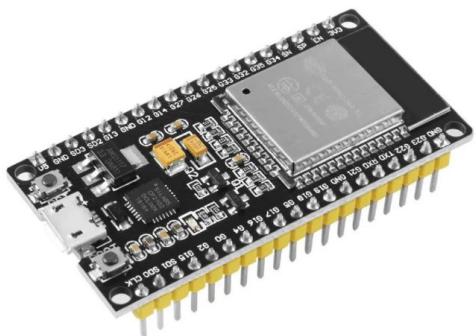
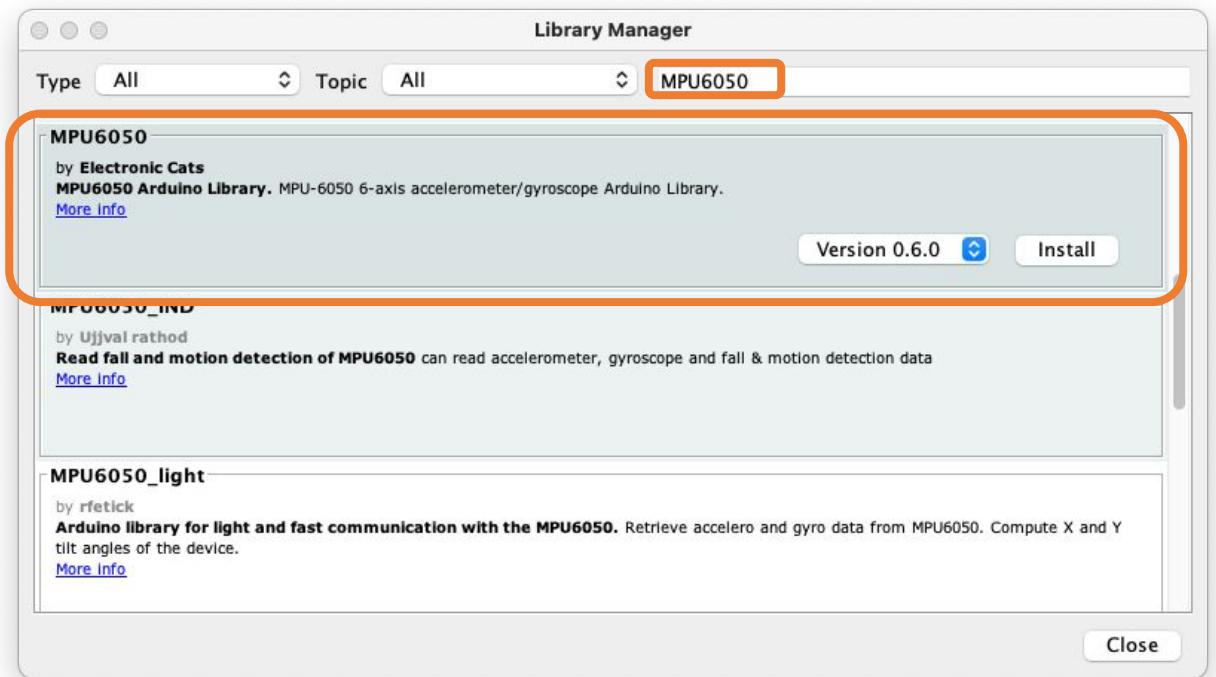
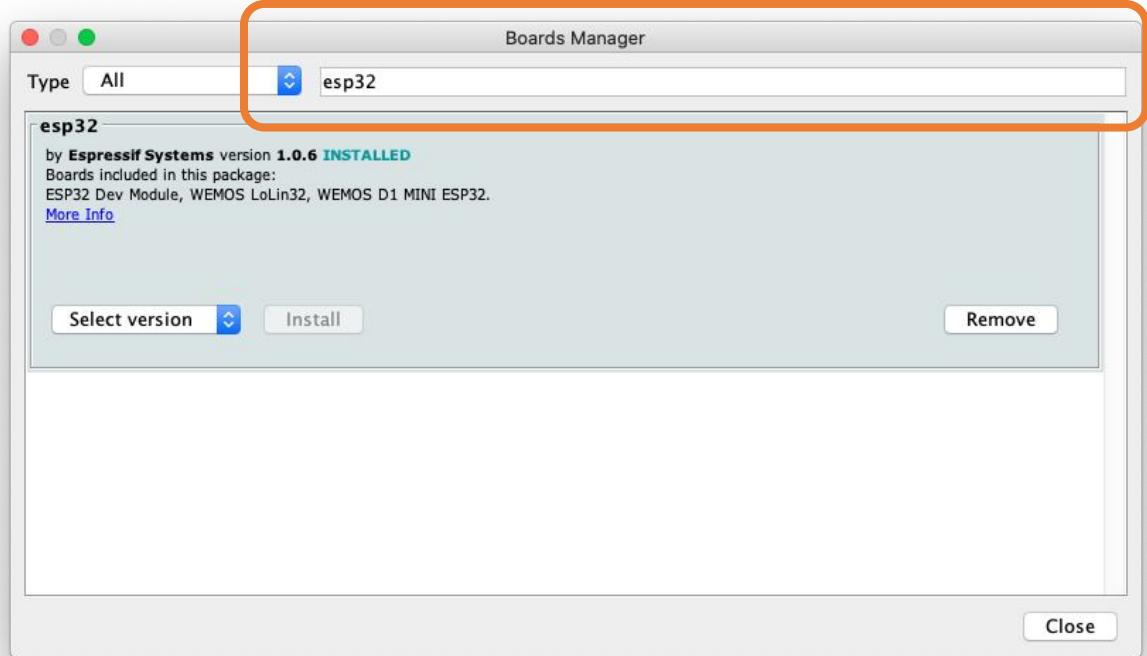
# Code available at:

[https://github.com/Mjrovai/ESP32-TinyML/tree/main/ESP32-Motion\\_Classification](https://github.com/Mjrovai/ESP32-TinyML/tree/main/ESP32-Motion_Classification)

On Arduino IDE, navigate to **File > Preferences**, and fill in "Additional Boards Manager URLs" with the URL below:  
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json)

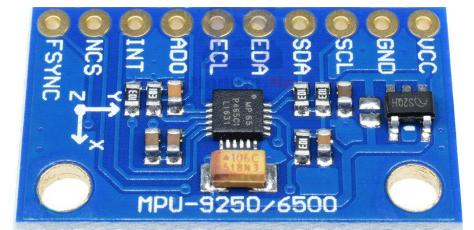


# Install HW Libraries

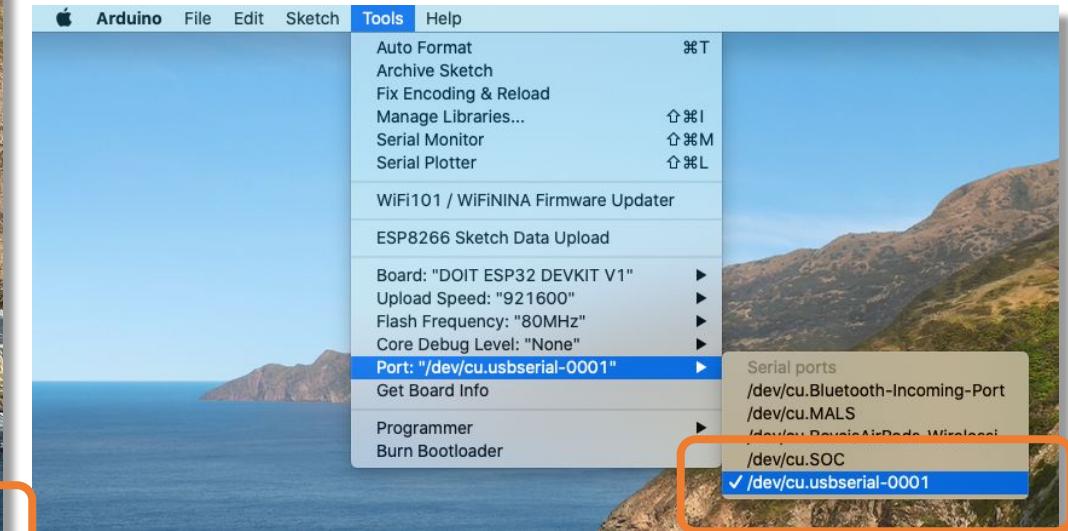
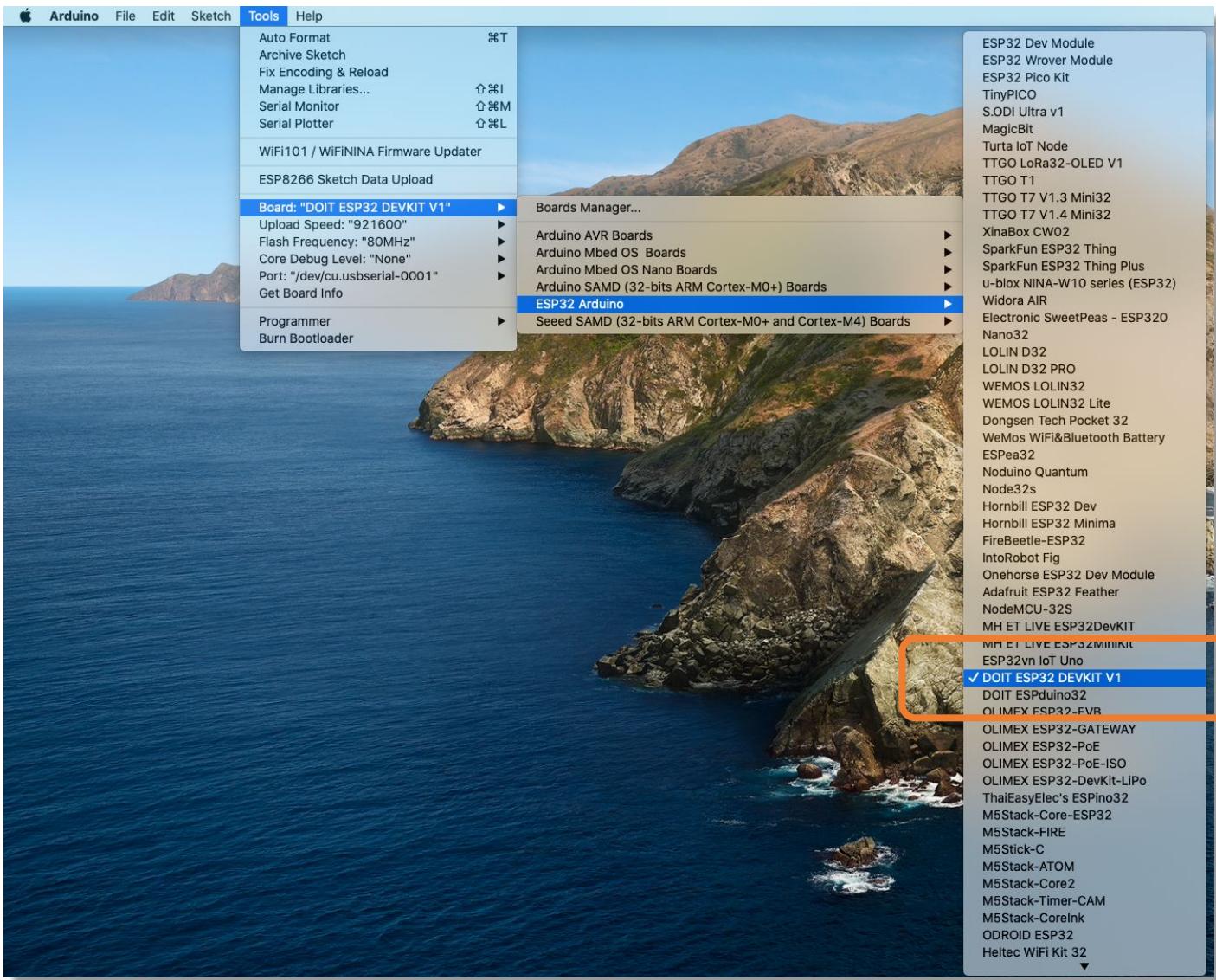


MPU6050

I2C - Accelerometer & Gyroscope - I2C/SPI



MPU6500



Select Board and Port

GetChipID | Arduino 1.8.16

```
/* The true ESP32 chip ID is essentially its MAC address.
This sketch provides an alternate chip ID that matches
the output of the ESP.getChipId() function on ESP8266
(i.e. a 32-bit integer matching the last 3 bytes of
the MAC address. This is less unique than the
MAC address chip ID, but is helpful when you need
an identifier that can be no more than a 32-bit integer
(like for switch...case).
created 2020-06-07 by cweinhofer
with help from Cicicok */
uint32_t chipId = 0;
void setup() {
  Serial.begin(115200);
}
void loop() {
  for(int i=0; i<17; i=i+8) {
    chipId |= ((ESP.getEfuseMac() >> (40 - i)) & 0xff) << i;
  }
  Serial.printf("ESP32 Chip model = %s Rev %d\n", ESP.getChipModel(), ESP.getChipRev());
  Serial.printf("This chip has %d cores\n", ESP.getChipCores());
  Serial.print("Chip ID: "); Serial.println(chipId);
  delay(3000);
}
```

Done uploading.  
Compressed 3072 bytes to 128...  
Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 2623.9 kbi  
Hash of data verified.

Leaving...  
Hard resetting via RTS pin...

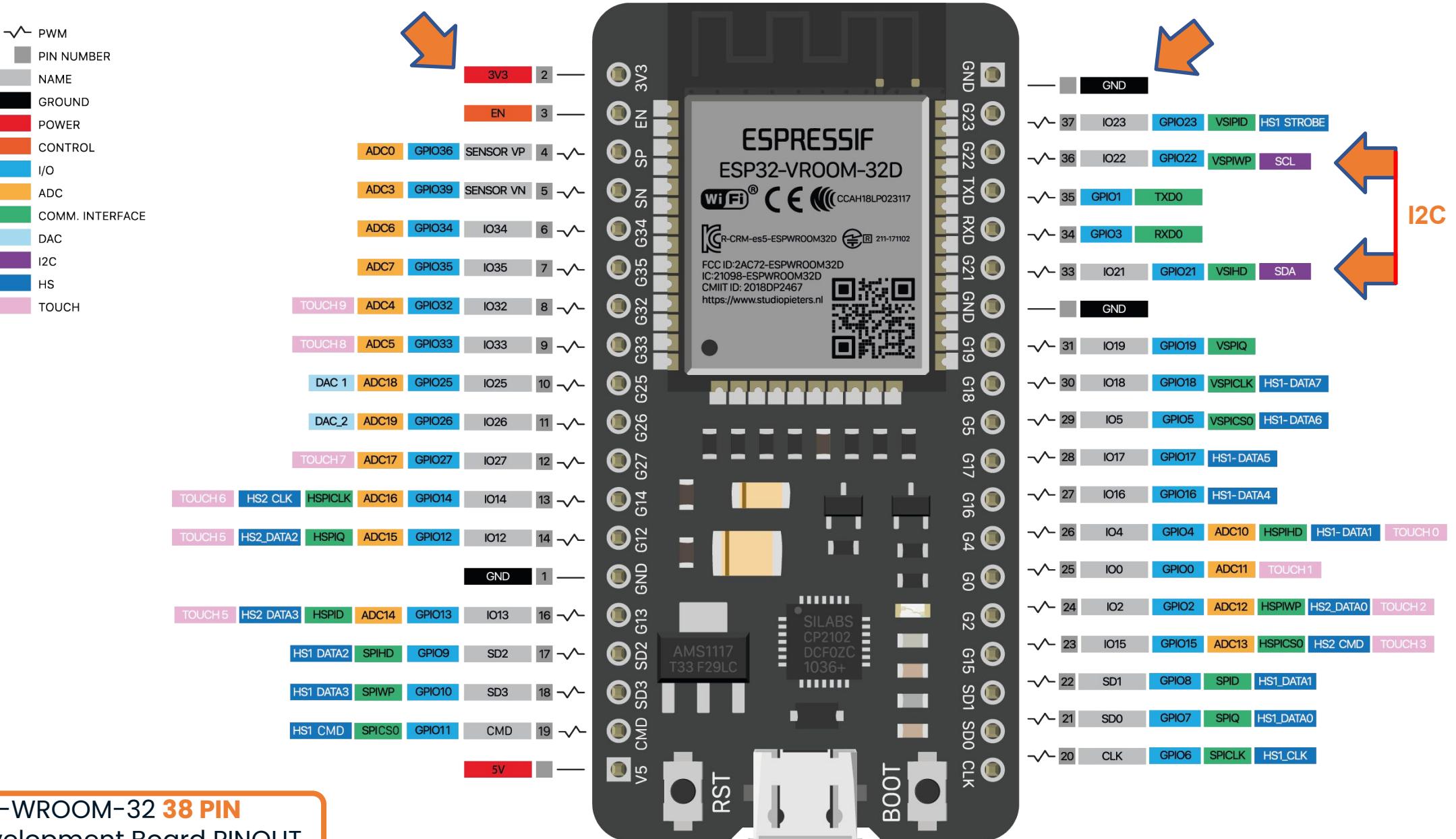
25 DOIT ESP32 DEVKIT V1, 80MHz, 921600, None on /dev/cu.usbserial-0001

# Test ESP32 Installation

/dev/cu.usbserial-0001

```
ESP32 Chip model = ESP32-D0WDQ6 Rev 1
This chip has 2 cores
Chip ID: 4487988
ESP32 Chip model = ESP32-D0WDQ6 Rev 1
This chip has 2 cores
Chip ID: 4487988
ESP32 Chip model = ESP32-D0WDQ6 Rev 1
This chip has 2 cores
Chip ID: 4487988
ESP32 Chip model = ESP32-D0WDQ6 Rev 1
This chip has 2 cores
Chip ID: 4487988
ESP32 Chip model = ESP32-D0WDQ6 Rev 1
This chip has 2 cores
Chip ID: 4487988
ESP32 Chip model = ESP32-D0WDQ6 Rev 1
This chip has 2 cores
Chip ID: 4487988
ESP32 Chip model = ESP32-D0WDQ6 Rev 1
This chip has 2 cores
Chip ID: 4487988
```

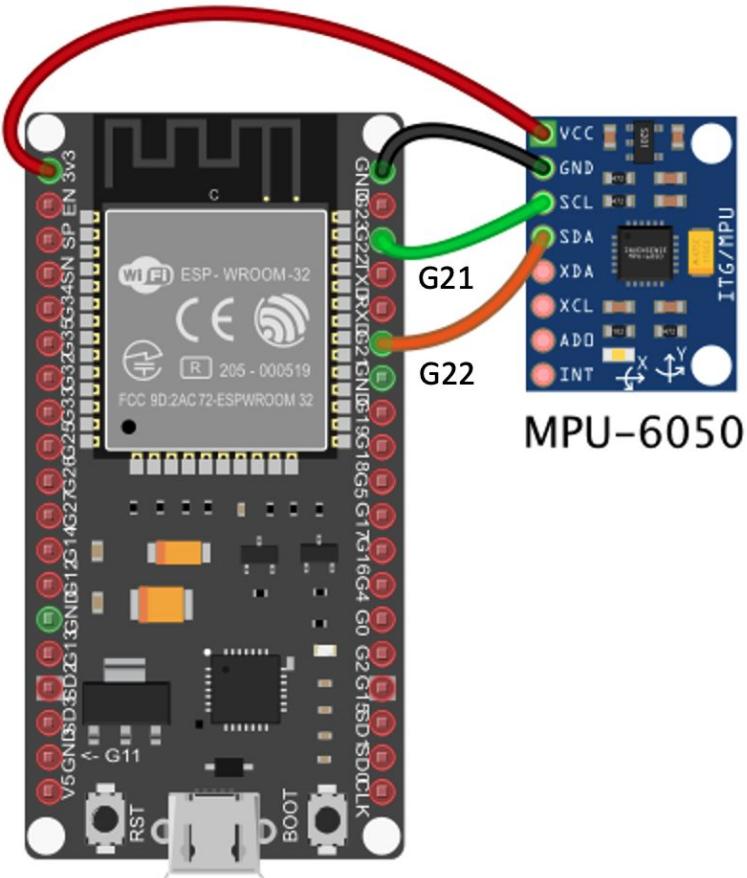
Autoscroll  Show timestamp Both NL & CR 115200 baud Clear output



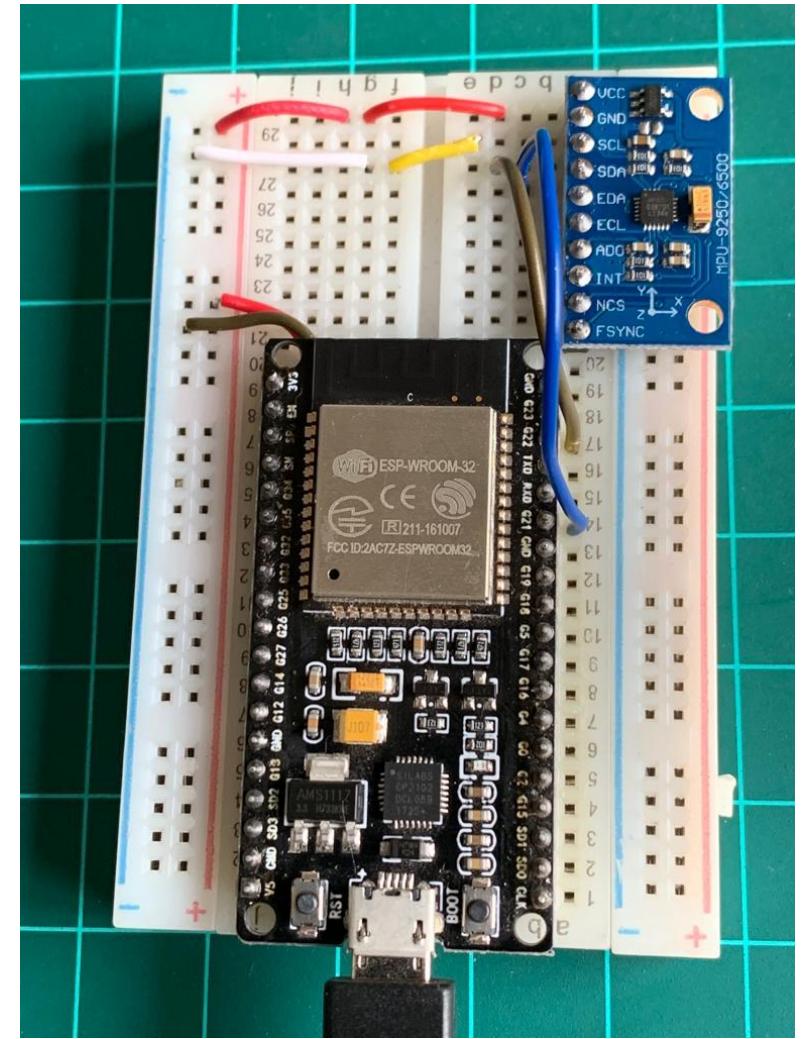
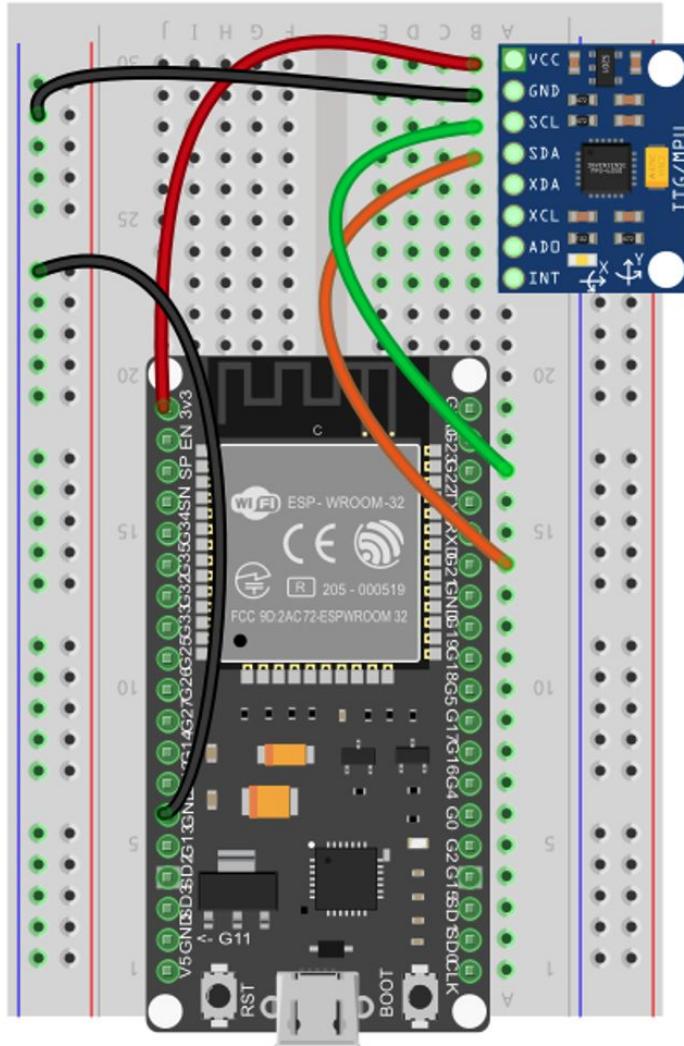
# ESP-WROOM-32 **38 PIN** Development Board PINOUT

<https://www.studiopieters.nl/esp32-pinout/>

# Connect HW



ESP32-38PinWide



MPU6050\_Acc\_Data\_Acquisition | Arduino 1.8.19

```

MPU6050_Acc_Data_Acquisition
1 /*
2 * Based on I2C device class (I2Cdev) Arduino sketch for MPU6050 class
3 * and Edge Impulse Data Forwarder Example (Arduino) - https://docs.edgeimpulse.com
4 *
5 * Developed by M.Rovai @11May23
6 */
7
8 #include "I2Cdev.h"
9 #include "MPU6050.h"
10 #include "Wire.h"
11
12 #define FREQUENCY_HZ      50
13 #define INTERVAL_MS      (1000 / (FREQUENCY_HZ + 1))
14 #define ACC_RANGE        1 // 0: -/+2G; 1: -/+4G
15
16 // convert factor g to m/s2 ==> [-32768, +32767] ==> [-2g, +2g]
17 #define CONVERT_G_TO_MS2 (9.81/(16384.0/(1.+ACC_RANGE)))
18
19 static unsigned long last_interval_ms = 0;
20
21 MPU6050 imu;
22 int16_t ax, ay, az;
23
24 void setup() {
25
26   Serial.begin(115200);
27
28   // initialize device

```

Test Sensor

Arduino Nano 33 BLE on /dev/cu.usbmodem1101

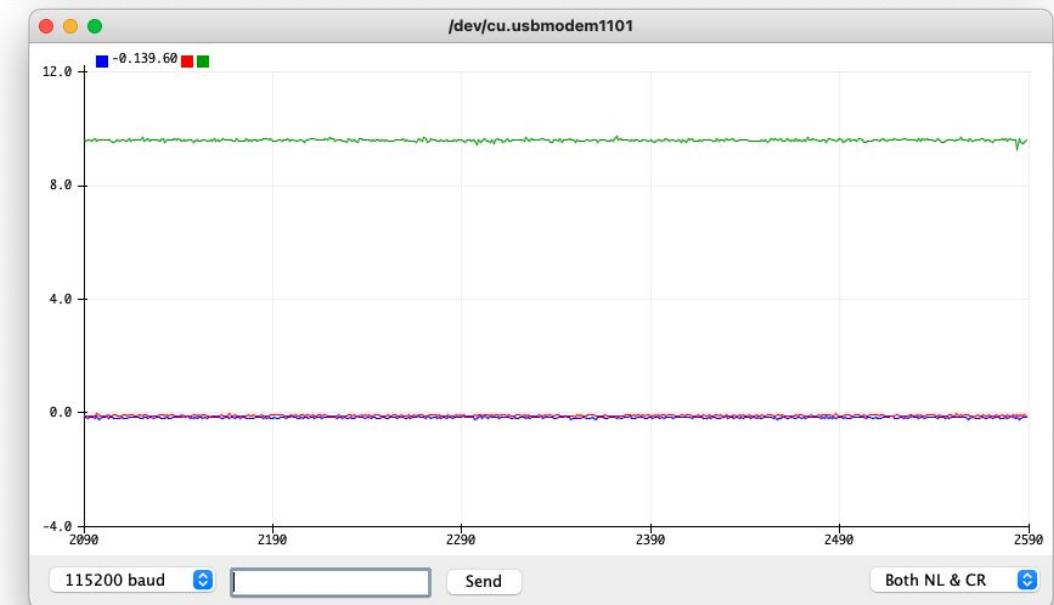
/dev/cu.usbmodem1101

Initializing I2C devices...

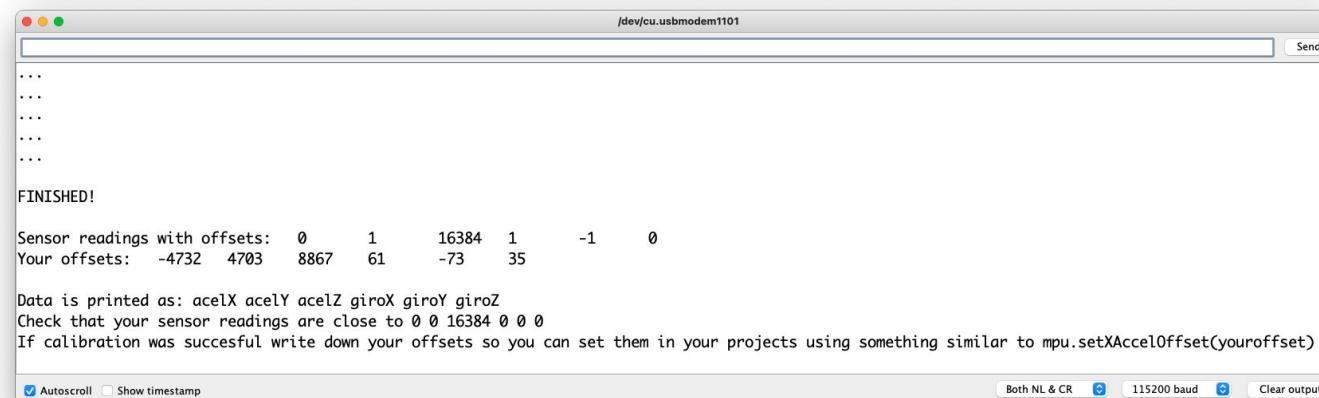
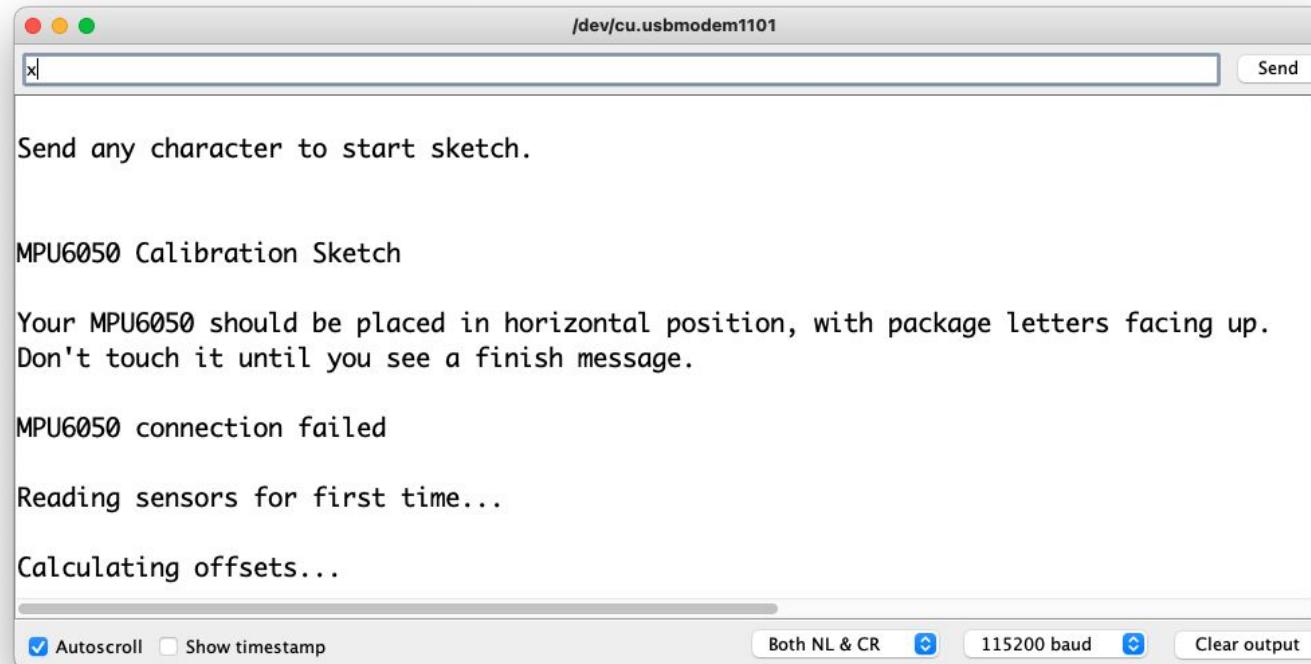
IMU Error

-0.08	-0.11	19.55	• Left the sensor over a table without motion.
-0.03	-0.10	9.72	
-0.01	-0.07	9.73	
-0.03	-0.06	9.71	• The reading should be around 0, 0, 9.78.
-0.04	-0.07	9.71	
0.01	-0.10	9.66	
-0.01	-0.01	9.73	• If not, run the calibration sketch
-0.03	-0.06	9.67	
-0.01	-0.08	9.71	
-0.02	-0.07	9.76	
-0.03	-0.05	9.70	
0.00	-0.05	9.73	
-0.02	-0.08	9.71	

Autoscroll  Show timestamp  Both NL & CR



# Optional Calibration



Dashboard - ESP32-Motion-CI

studio.edgeimpulse.com/studio/54781

**EDGE IMPULSE**

Project info Keys Export

MJRoBot (Marcelo Rovai)

**MJRoBot (Marcelo Rovai) / ESP32-Motion-Classification**

This is your Edge Impulse project. From here you acquire new training data, design impulses and train models.

**Creating your first impulse (100% complete)**

**Acquire data**  
Every Machine Learning project starts with data. You can capture data from a development board or your phone, or import data you already collected.  
[LET'S COLLECT SOME DATA](#)

**Design an impulse**  
Teach the model to interpret previously unseen data, based on historical data. Use this to categorize new data, or to find anomalies in sensor readings.  
[GETTING STARTED: CONTINUOUS MOTION RECOGNITION](#)  
[GETTING STARTED: RESPONDING TO YOUR VOICE](#)  
[GETTING STARTED: ADDING SIGHT TO YOUR SENSORS](#)

**Deploy**  
Package the complete impulse up, from signal processing code to trained model, and deploy it on your device. This ensures that the impulse runs with low latency and without requiring a network connection.  
[DEPLOY YOUR MODEL](#)

**Download block output**

TITLE	TYPE	SIZE
Spectral features training data	NPY file	2414 windows

**Sharing**  
Your project is private.  
[Make this project public](#)

**Summary**

DEVICES CONNECTED  
1

DATA COLLECTED  
5m 59s

**Collaborators**

MJRoBot (Marcelo Rovai) OWNER

**Project info**

Project ID  
54781

Labeling method  
One label per data item

Latency calculations  
Cortex-M4F 80MHz

# Connect ESP32 to EDGE IMPULSE CLI (Via USB)

The screenshot shows a terminal window titled "mjrovai — node /usr/local/bin/edge-impulse-data-forwarder — 139x34". The command entered is "edge-impulse-data-forwarder". The output shows the Edge Impulse data forwarder version 1.13.16 and its endpoints: Websocket, API, and Ingestion. It then connects to the serial port /dev/tty.usbserial-0001 and establishes a websocket connection to the remote management server. The user is prompted to connect the device to a project. The project "MJRoBot (Marcelo Rovai) / ESP32-Motion-Classification" is selected. The user is asked to name the device "ESP32" and map three sensor axes: accX, accY, and accZ. Finally, the message "Device 'ESP32' is now connected to project 'ESP32-Motion-Classification'" is displayed, along with a link to build a machine learning model.

```
mjrovai — node /usr/local/bin/edge-impulse-data-forwarder — 139x34
[(base) MacBook-Pro-de-Marcelo:~ mjrovai$ edge-impulse-data-forwarder
Edge Impulse data forwarder v1.13.16
Endpoints:
  WebSocket: wss://remote-mgmt.edgeimpulse.com
  API:       https://studio.edgeimpulse.com/v1
  Ingestion: https://ingestion.edgeimpulse.com

[SER] Connecting to /dev/tty.usbserial-0001
[SER] Serial is connected (00:01)
[WS ] Connecting to wss://remote-mgmt.edgeimpulse.com
[WS ] Connected to wss://remote-mgmt.edgeimpulse.com

? To which project do you want to connect this device? MJRoBot (Marcelo Rovai) / ESP32-Motion-Classification
[SER] Detecting data frequency...
[SER] Detected data frequency: 51Hz
? 3 sensor axes detected (example values: [-1084, 436, 17144]). What do you want to call them? Separate the names with ',': accX, accY, accZ
? What name do you want to give this device? ESP32
[WS ] Device "ESP32" is now connected to project "ESP32-Motion-Classification"
[WS ] Go to https://studio.edgeimpulse.com/studio/54781/acquisition/training to build your machine learning model!
```

```
mjrovai — node /usr/local/bin/edge-impulse-data-forwarder — 139x34
[(base) MacBook-Pro-de-Marcelo:~ mjrovai$ edge-impulse-data-forwarder
Edge Impulse data forwarder v1.13.16
Endpoints:
  Websocket: wss://remote-mgmt.edgeimpulse.com
  API:      https://studio.edgeimpulse.com/v1
  Ingestion: https://ingestion.edgeimpulse.com

[SER] Connecting to /dev/tty.usbserial-0001
[SER] Serial is connected (00:01)
[WS ] Connecting to wss://remote-mgmt.edgeimpulse.com
[WS ] Connected to wss://remote-mgmt.edgeimpulse.com

? To which project do you want to connect this device? MJRobot (Marcelo Rovai) / ESP32-Motion-Classification
[SER] Detecting data frequency...
[SER] Detected data frequency: 51Hz
? 3 sensor axes detected (example values: [-1084,436,17144]). What do you want to call them? Separate the names with ',': accX, accY, accZ
? What name do you want to give this device? ESP32
[WS ] Device "ESP32" is now connected to project "ESP32-Motion-Classification"
[WS ] Go to https://studio.edgeimpulse.com/studio/54781/acquisition/training to build your machine learning model!
[WS ] Incoming sampling request {
  path: '/api/testing/data',
  label: 'idle',
  length: 10000,
  interval: 19.607843137254903,
  hmacKey: 'a81704600713d60c2f2ab34721143d08',
  sensor: 'Sensor with 3 axes (accX, accY, accZ)'
}
[SER] Waiting 2 seconds...
[SER] Reading data from device...
[SER] Reading data from device OK (504 samples at 51Hz)
[SER] Uploading sample to https://ingestion.edgeimpulse.com/api/testing/data...
[SER] Sampling finished
```

Devices - ESP32-Motion-Class X +

studio.edgeimpulse.com/studio/54781/devices

EDGE IMPULSE MJRoBot (Marcelo Rovai)

Deleted device ("ESP32")

Your devices

+ Connect a new device

These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.

NAME	ID	TYPE	SENSORS	REMO...	LAST SEEN
00:01	00:01	DATA_FORWARDER	Sensor with 3 axes (accX, a...	●	Today, 16:30:34

© 2021 Edgimpulse Inc. All rights reserved

Dashboard Devices Data acquisition Impulse design Create impulse Spectral features NN Classifier EON Tuner Retrain model Live classification Model testing Versioning Deployment

GETTING STARTED Documentation Forums

The screenshot shows the Edge Impulse studio interface. On the left is a sidebar with various project management and documentation links. The main area is titled 'DEVICES (ESP32-MOTION-CLASSIFICATION)' and displays a green banner indicating a 'Deleted device ("ESP32")'. Below this is a table titled 'Your devices' listing a single device entry: '00:01' (ID), 'DATA\_FORWARDER' (TYPE), 'Sensor with 3 axes (accX, a...' (SENSORS), and 'Today, 16:30:34' (LAST SEEN). The 'DATA\_FORWARDER' row is highlighted with an orange rectangle. At the bottom of the page, there is a copyright notice for Edgimpulse Inc. and links for 'Documentation' and 'Forums'.

CREATE IMPULSE (ESP32-MOTION-CLASSIFICATION)

An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

**Time series data**

Axes: accX, accY, accZ

Window size: 2000 ms.

Window increase: 80 ms.

Frequency (Hz): 50

Zero-pad data:

**Spectral Analysis**

Name: Spectral features

**Classification (Keras)**

Name: NN Classifier

Input axes: accX, accY, accZ

Input features:  Spectral features

Output features: 4 (idle, lift, maritime, terrestrial )

**Output features**

4 (idle, lift, maritime, terrestrial )

**Save Impulse**

Add a processing block

Add a learning block

© 2021 Edgimpulse Inc. All rights reserved

Spectral features - ESP32-Motion-classification

studio.edgeimpulse.com/studio/54781/dsp/spectral-analysis/3/generate-features

**EDGE IMPULSE**

**SPECTRAL FEATURES (ESP32-MOTION-CLASSIFICATION)**

#1 ▾ Click to set a description for this version

**Parameters** **Generate features**

**Training set**

- Data in training set 3m 59s
- Classes 4 (idle, lift, maritime, terrestrial)
- Window length 2000 ms.
- Window increase 80 ms.
- Training windows 2,414

**Feature explorer (2,414 samples)**

X Axis accX RMS Y Axis accY RMS Z Axis accZ RMS

Legend:
 

- idle (blue)
- lift (orange)
- maritime (green)
- terrestrial (red)

**Feature generation output**

```

Scheduling job in cluster...
Job started
Creating windows from 24 files...
[ 0/24] Creating windows from files...
[ 1/24] Creating windows from files...
[24/24] Creating windows from files...
Created 2414 windows: idle: 596, lift: 606, maritime: 606, terrestrial : 606

Creating features
[ 1/2414] Creating features...
[1046/2414] Creating features...
[2076/2414] Creating features...
[2414/2414] Creating features...
Created features

Job completed
  
```

© 2021 EdgeImpulse Inc. All rights reserved

NN CLASSIFIER (ESP32-MOTION-CLASSIFICATION)  
#1 ▾ Click to set a description for this version

### Neural Network settings

Training settings

Number of training cycles ② 30

Learning rate ② 0.0005

### Neural network architecture

```

graph TD
    Input[Input layer (33 features)] --- Dense1[Dense layer (20 neurons)]
    Dense1 --- Dense2[Dense layer (10 neurons)]
    Dense2 --- Output[Output layer (4 features)]
    
```

Add an extra layer

**Start training**

**Model**

Last training performance (validation set)


**ACCURACY**  
**96.9%**

**LOSS**  
**0.95**

Confusion matrix (validation set)

	IDLE	LIFT	MARITIME	TERRESTRIAL
IDLE	100%	0%	0%	0%
LIFT	3.3%	96.7%	0%	0%
MARITIME	1.7%	2.6%	95.7%	0%
TERRESTRIAL	0.8%	4.1%	0%	95.0%
F1 SCORE	0.97	0.95	0.98	0.97

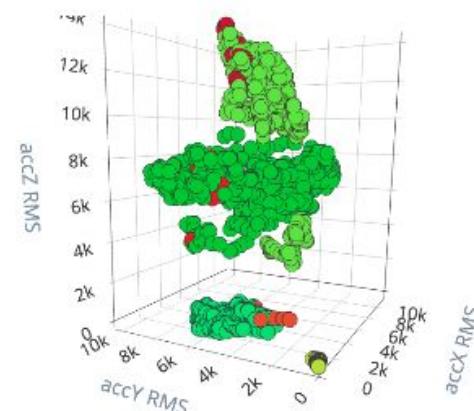
Feature explorer (full training set) ②

accX RMS

accY RMS

accZ RMS

- idle - correct
- lift - correct
- maritime - correct
- terrestrial - correct
- lift - incorrect
- maritime - incorrect
- terrestrial - incorrect



On-device performance ②


**INFERENCE TIME**  
**1 ms.**

**PEAK RAM USAGE**  
**1.7K**

**FLASH USAGE**  
**19.0K**

Model testing - ESP32-Motion X + studio.edgeimpulse.com/studio/54781/validation

## Test data

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCOME	LENGTH	ACCURACY	RESULT
idle.json.2jb4ipqd	idle	10s	89%	90 idle, 6 lift, 4 uncertain, 1 maritime
testing.json.2jb2djgq	maritime	10s	98%	99 maritime, 2 idle
testing.json.2jb2cdi2	terrestrial	10s		101 terrestrial
testing.json.2jb29lqg	lift	10s	70%	71 lift, 30 maritime
idle.json.2jb1mdjv	idle	10s	18%	76 maritime, 19 idle, 4 lift, 2 uncertain
idle.json.2jb1j3th	idle	10s	100%	101 idle
lift.json.2jb1hmvt	lift	10s	85%	86 lift, 15 maritime
terrestrial.json.2jb1gu...	terrestrial	10s		101 terrestrial
maritime.json.2jb1e80t	maritime	10s	100%	101 maritime
maritime.json.2jb1c79k	maritime	10s	100%	101 maritime
terrestrial.json.2jb18...	terrestrial	10s		101 terrestrial
lift.json.2jb130ml	lift	10s	94%	95 lift, 6 maritime

## Model testing output

Classifying data for NN Classifier...  
 Copying features from processing blocks...  
 Copying features from DSP block...  
 Copying features from DSP block OK  
 Copying features from processing blocks OK

Classifying data for float32 model...  
 Scheduling job in cluster...  
 Job started  
 Classifying data for NN Classifier OK

Job completed

## Model testing results

ACCURACY 83.94%

	IDLE	LIFT	MARITIME	TERRESTRIAL	UNCERTAIN
IDLE	69.3%	3.3%	25.4%	0%	2.0%
LIFT	0%	83.2%	16.8%	0%	0%
MARITIME	0.7%	0%	99.3%	0%	0%
TERRESTRIAL	-	-	-	-	-
F1 SCORE	0.82	0.89	0.82	0.00	

## Feature explorer

accX RMS accY RMS accZ RMS

Legend:

- idle - correct
- lift - correct
- maritime - correct
- idle - incorrect
- lift - incorrect
- maritime - incorrect
- terrestrial

Model testing - ESP32-Motion X + studio.edgeimpulse.com/studio/54781/validation

**Test data**

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCOME	LENGTH	ACCURACY	RESULT
idle.json.2jb4ipqd	idle	10s	89%	90 idle, 6 lift, 4 uncertain, 1 maritime
testing.json.2jb2djgq	maritime	10s	98%	99 maritime, 2 idle
testing.json.2jb2cdi2	terrestrial	10s		101 terrestrial
testing.json.2jb29lqg	lift	10s	70%	71 lift, 30 maritime
idle.json.2jb1mdjv	idle	10s		
idle.json.2jb1j3th	idle	10s		
lift.json.2jb1hmvt	lift	10s		
terrestrial.json.2jb1gu...	terrestrial	10s		
maritime.json.2jb1e80t	maritime	10s		
maritime.json.2jb1c79k	maritime	10s		
terrestrial.json.2jb18...	terrestrial	10s		
lift.json.2jb130ml	lift	10s	94%	95 lift, 6 maritime

**Model testing output**

Classifying data for NN Classifier...  
 Copying features from processing blocks...  
 Copying features from DSP block...  
 Copying features from DSP block OK  
 Copying features from processing blocks OK

Classifying data for float32 model...  
 Scheduling job in cluster...  
 Job started  
 Classifying data for NN Classifier OK

**Job completed**

**Set confidence thresholds**

Every learning block has a threshold. This can be the minimum confidence that a neural network needs to have, or the maximum anomaly score before a sample is tagged as an anomaly. You can configure these thresholds to tweak the sensitivity of these learning blocks. This affects both live classification and model testing.

**NN Classifier**

Minimum confidence rating

**Set confidence thresholds**

IDLE	LIFT	MARITIME	TERRESTRIAL	UNCERTAIN
69.3%	3.3%	25.4%	0%	2.0%
0%	83.2%	16.8%	0%	0%
0.7%	0%	99.3%	0%	0%
-	-	-	-	-
0.82	0.89	0.82	0.00	

**Feature explorer**

accX RMS accY RMS accZ RMS

Legend:

- idle - correct
- lift - correct
- maritime - correct
- idle - incorrect
- lift - incorrect
- maritime - incorrect
- terrestrial

**DEPLOYMENT (ESP32-MOTION-CLASSIFICATION)**

### Deploy your impulse

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. [Read more.](#)

#### Create library

Turn your impulse into optimized source code that you can run on any device.

 C++ library	 Arduino library	 Cube.MX CMSIS-PACK
 WA	 NVIDIA.	

**Select optimizations (optional)**

Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.

**Enable EON™ Compiler**

Same accuracy, up to 50% less memory. Open source.

**Available optimizations for NN Classifier**

Optimization Type	RAM USAGE	LATENCY	CONFUSION MATRIX																				
<b>Quantized (int8) ★</b>	<b>1.7K</b>	<b>1 ms</b>	<table border="1"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>100</td></tr> <tr><td>0</td><td>80.9</td><td>15.2</td><td>0</td><td>4.0</td></tr> <tr><td>0.7</td><td>0</td><td>98.7</td><td>0</td><td>0.7</td></tr> <tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> </table>	0	0	0	0	100	0	80.9	15.2	0	4.0	0.7	0	98.7	0	0.7	-	-	-	-	-
0	0	0	0	100																			
0	80.9	15.2	0	4.0																			
0.7	0	98.7	0	0.7																			
-	-	-	-	-																			
This optimization is recommended for best performance.	<b>FLASH USAGE</b> <b>19.0K</b>	<b>ACCURACY</b> <b>59.85%</b>	<small>(?)</small>																				
<b>Unoptimized (float32)</b>	<b>1.8K</b>	<b>1 ms</b>	<table border="1"> <tr><td>69.3</td><td>3.3</td><td>25.4</td><td>0</td><td>2.0</td></tr> <tr><td>0</td><td>83.2</td><td>16.8</td><td>0</td><td>0</td></tr> <tr><td>0.7</td><td>0</td><td>99.3</td><td>0</td><td>0</td></tr> <tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr> </table>	69.3	3.3	25.4	0	2.0	0	83.2	16.8	0	0	0.7	0	99.3	0	0	-	-	-	-	-
69.3	3.3	25.4	0	2.0																			
0	83.2	16.8	0	0																			
0.7	0	99.3	0	0																			
-	-	-	-	-																			
<small>Currently selected</small>	<b>FLASH USAGE</b> <b>21.3K</b>	<b>ACCURACY</b> <b>83.94%</b>	<small>(?)</small>																				

Estimate for Cortex-M4F 80MHz

**Build**

esp32\_sense\_accelerometer | Arduino 1.8.19

```
17 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES
18 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE
19 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
20 * SOFTWARE.
21 */
22
23/* Includes -----
24#include <ESP32-Motion-Classification_inferencing.h>
25#include "I2Cdev.h"
26#include "MPU6050.h"
27#include "Wire.h"
28
29#include "I2Cdev.h"
30#include "MPU6050.h"
31#include "Wire.h"
32
33/* Constant defines -----
34MPU6050 imu;
35int16_t ax, ay, az;
36
37#define ACC_RANGE      1 // 0: -/+2G; 1: +/-4G
38#define CONVERT_G_TO_MS2 (9.81/(16384/(1.+ACC_RANGE)))
39#define MAX_ACCEPTED_RANGE (2*9.81)+(2*9.81)*ACC_RANGE
40
41/*
42 ** NOTE: If you run into TFLite arena allocation issue.
43 **
44 ** This may be due to may dynamic memory fragmentation.
```

1

Arduino Nano 33 BLE on /dev/cu.usbmodem1101

esp32\_sense\_accelerometer | Arduino 1.8.19

```
62void setup()
63{
64    // put your setup code here, to run once:
65    Serial.begin(115200);
66    // comment out the below line to cancel the wait for USB com
67    while (!Serial);
68    Serial.println("Edge Impulse Inferencing Demo");
69
70    // initialize device
71    Serial.println("Initializing I2C devices...");
72    Wire.begin();
73    imu.initialize();
74    delay(10);
75
76    //Set MCU 6050 OffSet Calibration
77    imu.setXAccelOffset(-4732);
78    imu.setYAccelOffset(4703);
79    imu.setZAccelOffset(8867);
80    imu.setXGyroOffset(61);
81    imu.setYGyroOffset(-73);
82    imu.setZGyroOffset(35);
83
84    imu.setFullScaleAccelRange(ACC_RANGE);
85
86    if (EI_CLASSIFIER_RAW_SAMPLES_PER_FRAME != 3) {
87        ei_printf("ERR: EI_CLASSIFIER_RAW_SAMPLES_PER_FRAME shou
88        return;
89    }
90}
```

1

Arduino Nano 33 BLE on /dev/cu.usbmodem1101

```
esp32_sense_accelerometer | Arduino 1.8.19

void loop()
{
    ei_printf("\nStarting inferencing in 2 seconds...\n");
    delay(2000);
    ei_printf("Sampling...\n");
    // Allocate a buffer here for the values we'll read from the IMU
    float buffer[EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE] = { 0 };

    for (size_t ix = 0; ix < EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE; ix += 3) {
        // Determine the next tick (and then sleep later)
        uint64_t next_tick = micros() + (EI_CLASSIFIER_INTERVAL_MS * 1000);

        imu.getAcceleration(&ax, &ay, &az);
        buffer[ix + 0] = ax;
        buffer[ix + 1] = ay;
        buffer[ix + 2] = az;

        //ei_printf("raw values: %.2f, %.2f, %.2f\n", ax*CONVERT_G_TO_MS2, ay*CONVERT_G_TO_MS2, az*CONVERT_G_TO_MS2);

        buffer[ix + 0] *= CONVERT_G_TO_MS2;
        buffer[ix + 1] *= CONVERT_G_TO_MS2;
        buffer[ix + 2] *= CONVERT_G_TO_MS2;

        for (int i = 0; i < 3; i++) {
            if (fabs(buffer[ix + i]) > MAX_ACCEPTED_RANGE) {
                buffer[ix + i] = ei_get_sign(buffer[ix + i]) * MAX_ACCEPTED_RANGE;
            }
        }
    }
}
```

```
/dev/cu.usbserial-0001
Send

lift: 0.02348
maritime: 0.14678
terrestrial : 0.00562

Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 19 ms., Classification: 0 ms., Anomaly: 0 ms.):
idle: 0.88469
lift: 0.05896
maritime: 0.03198
terrestrial : 0.02436

Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 19 ms., Classification: 0 ms., Anomaly: 0 ms.):
idle: 0.86179
lift: 0.03734
maritime: 0.08310
terrestrial : 0.01776

Starting inferencing in 2 seconds...

 Autoscroll  Show timestamp Both NL & CR 115200 baud Clear output
```

```
maritime: 0.18839
terrestrial : 0.00147

Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 19 ms., Classification: 0 ms., Anomaly: 0 ms.):
    idle: 0.00000
    lift: 1.00000
    maritime: 0.00000
    terrestrial : 0.00000

Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 19 ms., Classification: 0 ms., Anomaly: 0 ms.):
    idle: 0.00000
    lift: 0.00000
    maritime: 1.00000
    terrestrial : 0.00000

Starting inferencing in 2 seconds...
Sampling...
```

Autoscroll  Show timestamp      Both NL & CR      115200 baud      Clear output

**Thanks**



**UNIFEI**