



Leveraging any Microcontrollers & Data Collection at **Edge Impulse Studio**

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



“**Edge AI** is a truly complete technology. As a topic, it makes use of knowledge from everything from the physical properties of semiconductor electronics all the way up to the engineering of high-level architectures that span devices and the cloud. It demands expertise in the most cutting-edge approaches to artificial intelligence and machine learning along with the most venerable skills of bare-metal embedded software engineering. It makes use of the entire history of computer science and electrical engineering, laid out end to end.”



Situnayake, Daniel; Plunkett, Jenny
AI at the Edge (pp. 215-216)
O'Reilly Media

Marcelo Rovai was born in São Paulo and holds a Master's degree in Data Science from the Universidad del Desarrollo (UDD) in Chile and an MBA from IBMEC (INSPER) in Brazil. He graduated in 1982 as an Engineer from UNIFEI, Federal University of Itajubá, with a specialization from Escola Politécnica de Engenharia of São Paulo University (USP), both institutions located in Brazil.

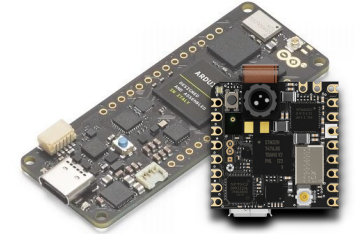
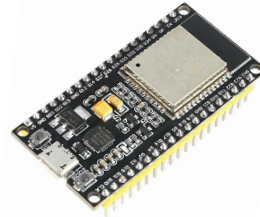
Rovai has experience as a teacher, engineer, and executive in several technology companies such as CDT/ETEP, AVIBRAS Aeroespacial, SID Informática, ATT-GIS, NCR, DELL, COMPAQ (HP), and more recently at IGT as a VP and a Senior Advisor for Latin America.

Marcelo Rovai publishes articles about electronics on websites such as MJRoBot.org, Hackster.io, Instructables.com, and Medium.com. Furthermore, he is a volunteer Professor at the UNIFEI in Brazil and a lecturer at several Congresses and Universities on IoT and TinyML. He is an active member and a Co-Chair of the TinyML4D group, an initiative to bring TinyML education to developing countries.



Hardware

Hardware



	Raspberry Pico (W)	Arduino Nano Sense	ESP 32	Seed XIAO Sense / ESP32S3	Arduino Pro
32Bits CPU	Dual-core Arm Cortex-M0+	Arm Cortex-M4F	Xtensa LX6 Dual Core	Arm Cortex-M4F (BLE) Xtensa LX7 Dual Core	Dual Core Arm Cortex M7/M4
CLOCK	133MHz	64MHz	240MHz	64 / 240MHz	480/240MHz
RAM	264KB	256KB	520KB (part available)	256KB / 8MB	1MB
ROM	2MB	1MB	2MB	2MB / 8MB	2MB
Radio	(Yes for W)	BLE	BLE/WiFi	BLE / WiFi (ESP32S3)	BLE/WiFi
Sensors	No	Yes	No	Yes (Sense)	Yes (Nicla)
Bat. Power Manag.	No	No	No	Yes	Yes
Price	\$	\$\$\$	\$	\$\$	\$\$\$\$\$

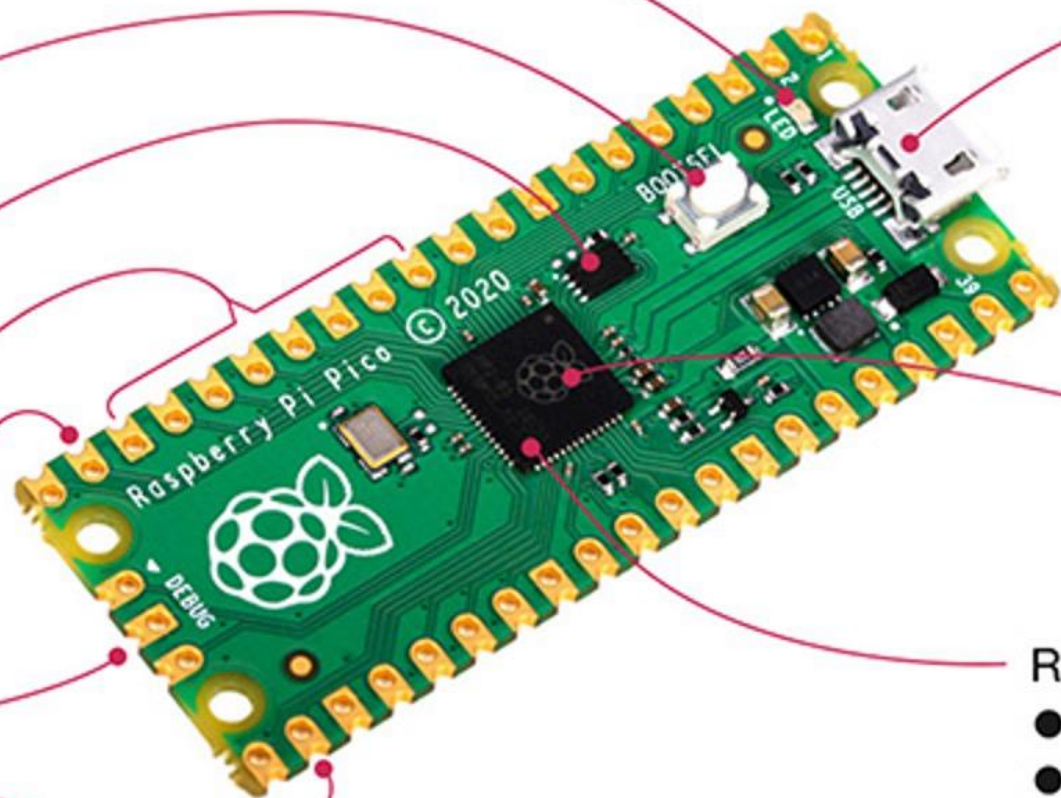
Programmable LED

Boot Selection Button

2 MByte Quad-SPI Flash

0.1" Spacing Pads
Breadboard friendly

SMT Friendly
Castellated Edges



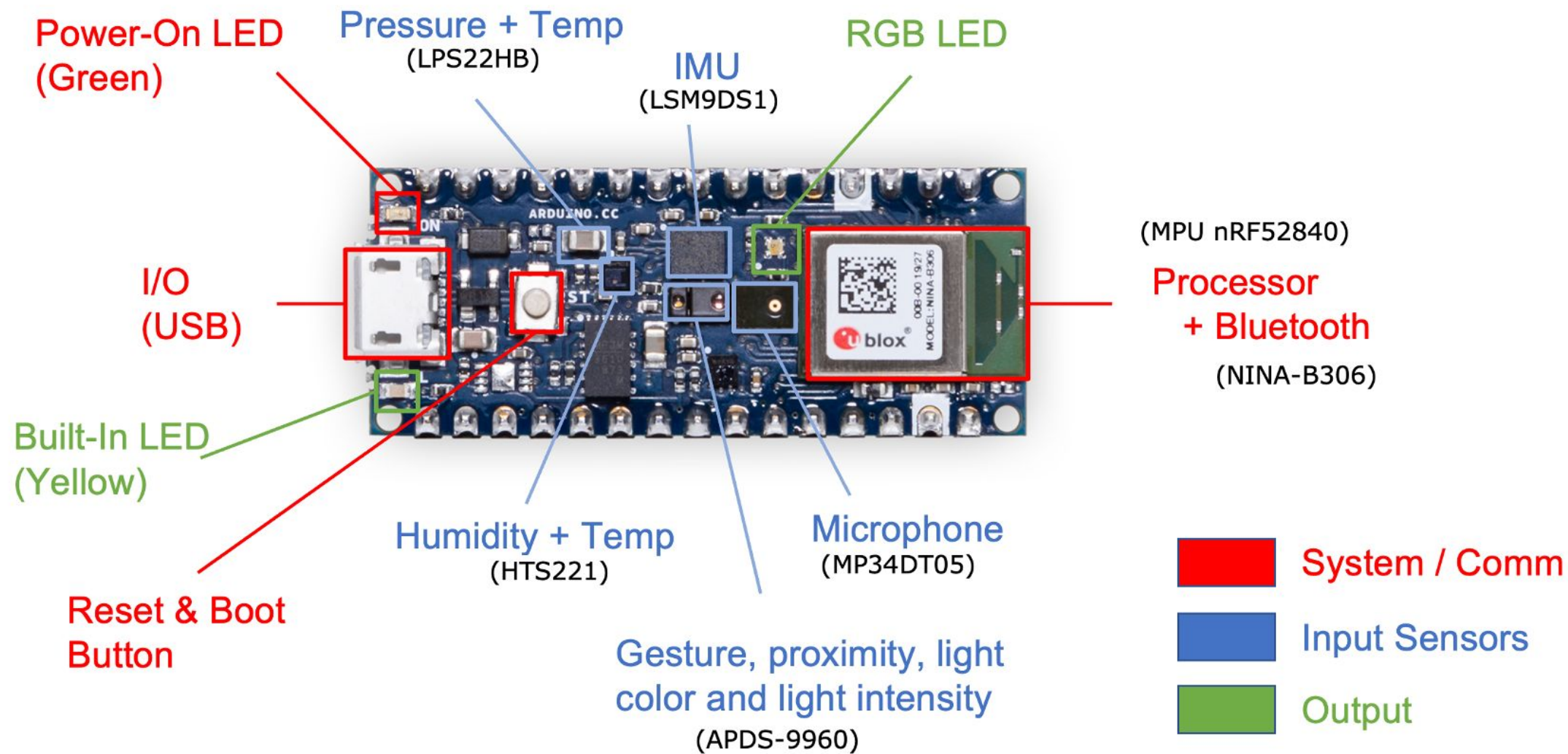
USB Micro B
Power & Data

Built-in 12-bit
Temperature Sensor

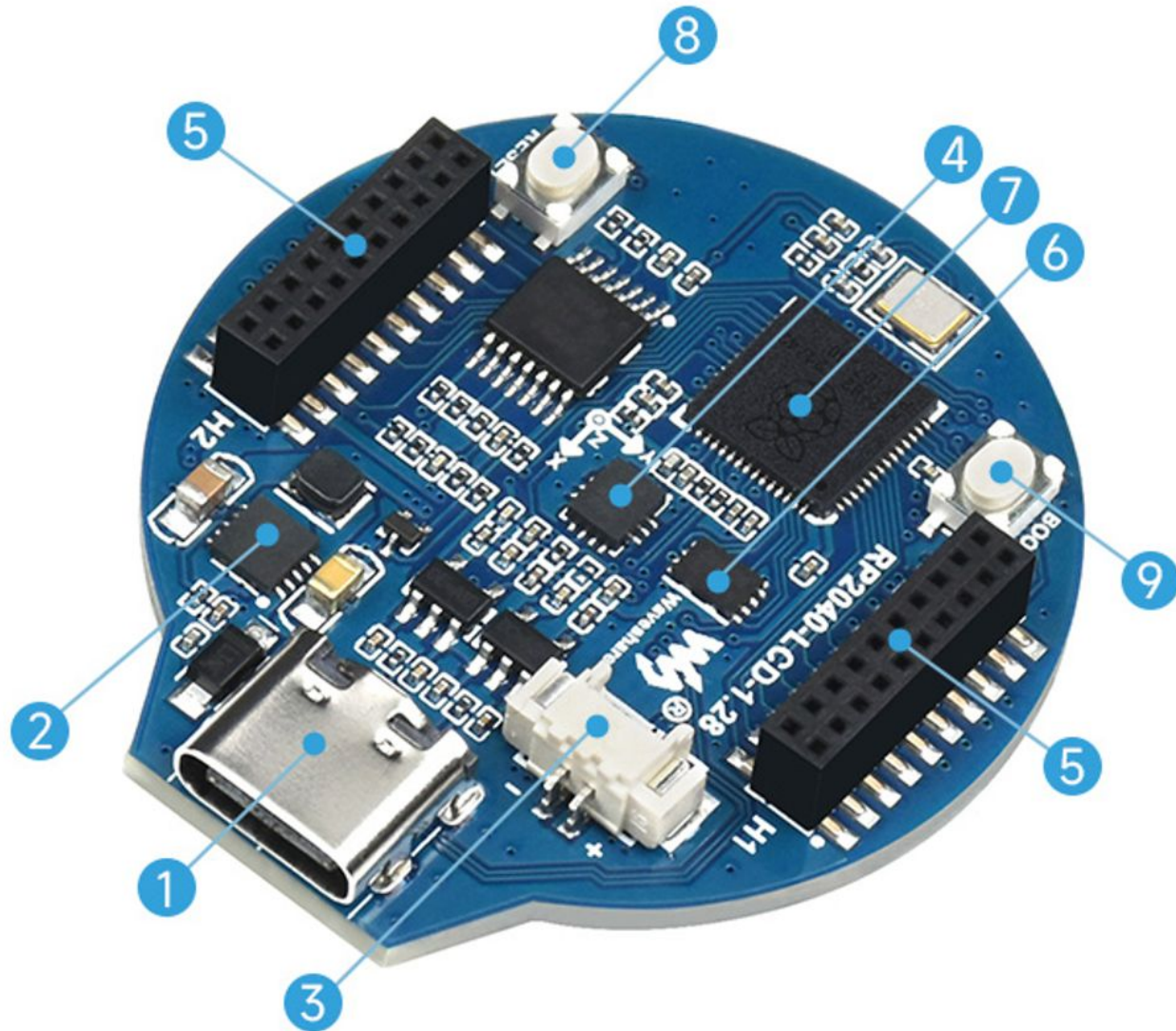
RP2040 MCU Silicon

- Dual-Core
- 32-bit ARM Cortex M0+
- 264 KByte SRAM
- Clock @48MHz, Max at 133MHz
- USB 1.1 Host and Device

Nano 33 BLE Sense (Development board)



RP2040 MCU Board, with LCD, accelerometer, and gyroscope Sensor



1. USB Type-C connector

USB 1.1 with device and host support

2. ETA6096

high efficiency Lithium battery recharge manager

3. Battery Header

MX1.25 header, for 3.7V Lithium battery, allows recharging the battery and powering the board at the same time

4. QMI8658C

IMU, includes a 3-axis gyroscope and a 3-axis accelerometer

5. 1.27mm pitch headers

Adapting all GPIO and Debug pins

6. W25Q16JVUXIQ

2MB NOR-Flash

7. RP2040

Dual-core processor, up to 133MHz operating frequency

8. RESET Button

9. BOOT Button

press it when resetting to enter download mode

<https://www.waveshare.com/rp2040-lcd-1.28.htm>

Application Complexity vs. HW



Application Complexity ↑

CPU Power / Memory →

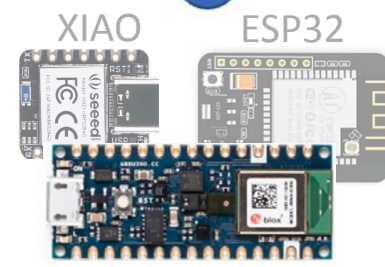


Anomaly Detection
 Sensor Classification
 20 KB



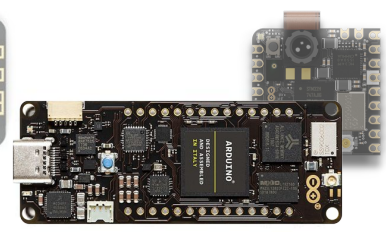
Rpi-Pico
 (Cortex-M0+)

KeyWord Spotting
 Audio Classification
 50 KB

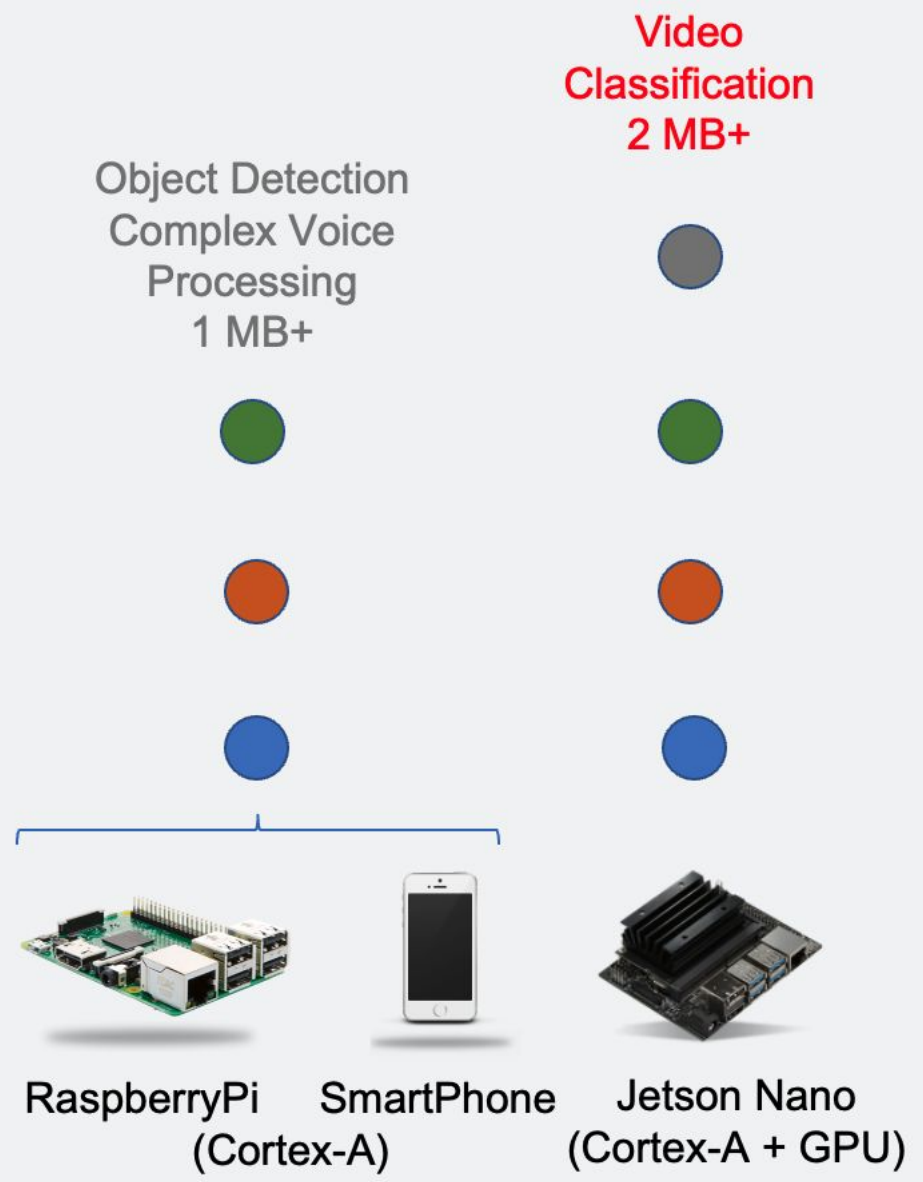


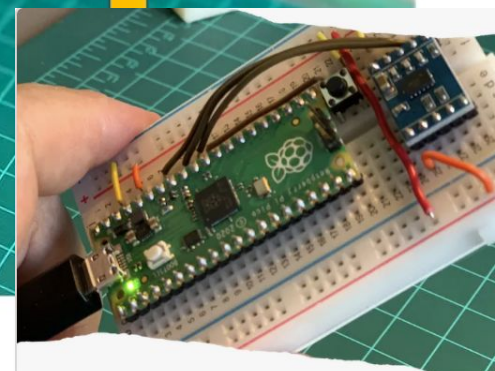
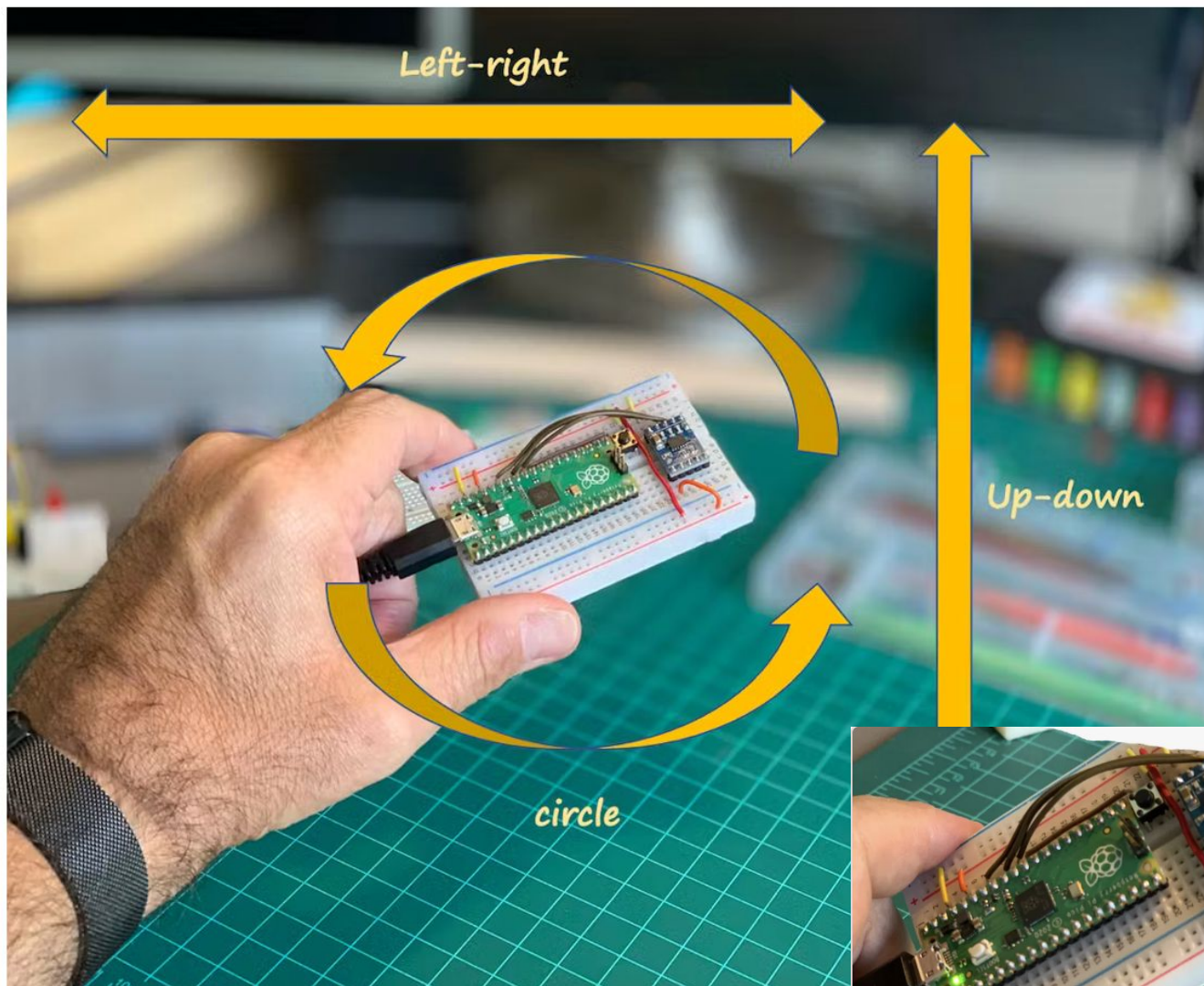
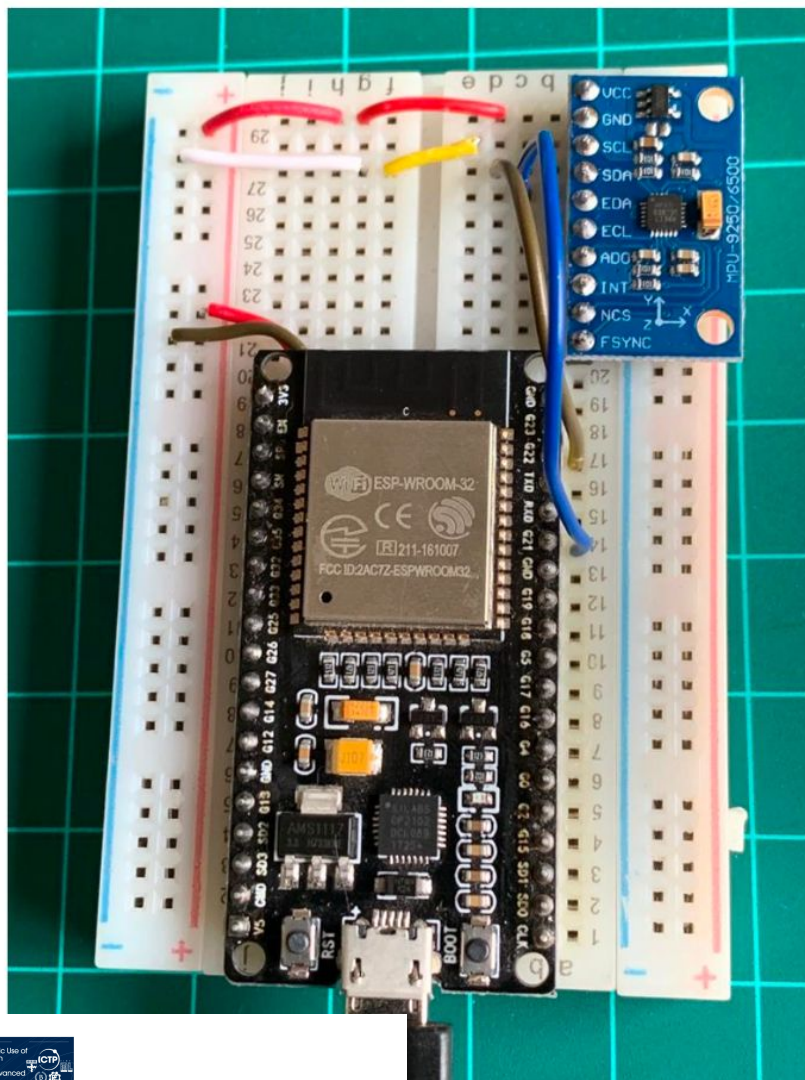
Arduino Nano
 (Cortex-M4)

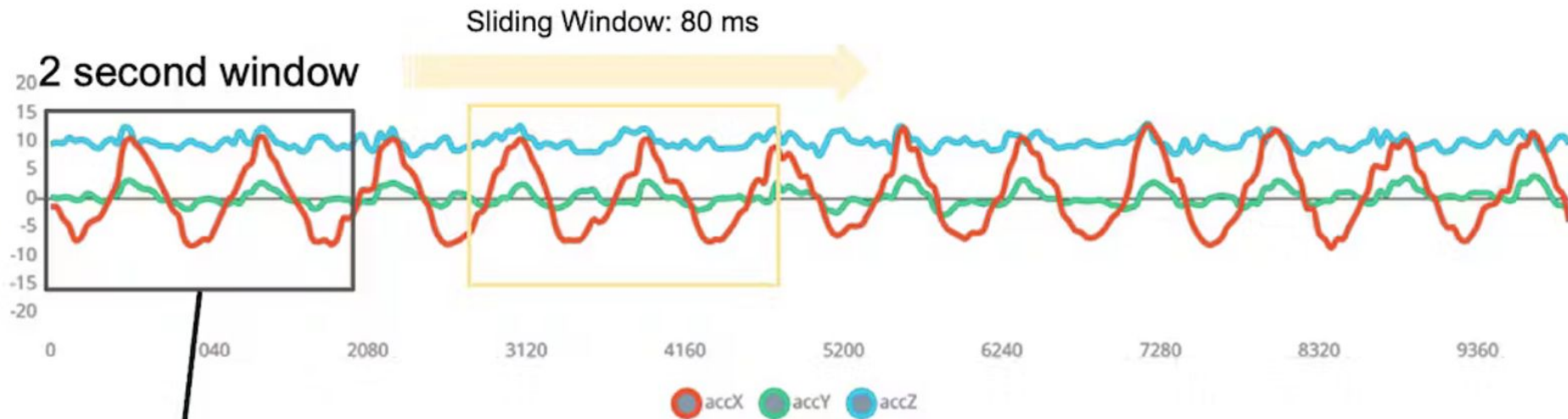
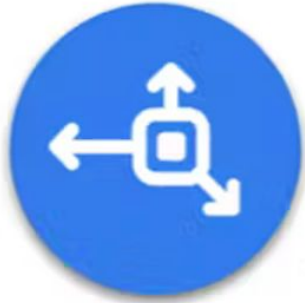
Image
 Classification
 250 KB+



Arduino Pro
 (Cortex-M7)







375 Raw Features

- Raw Data from sensor



Manual Feature
Extraction



33 Features

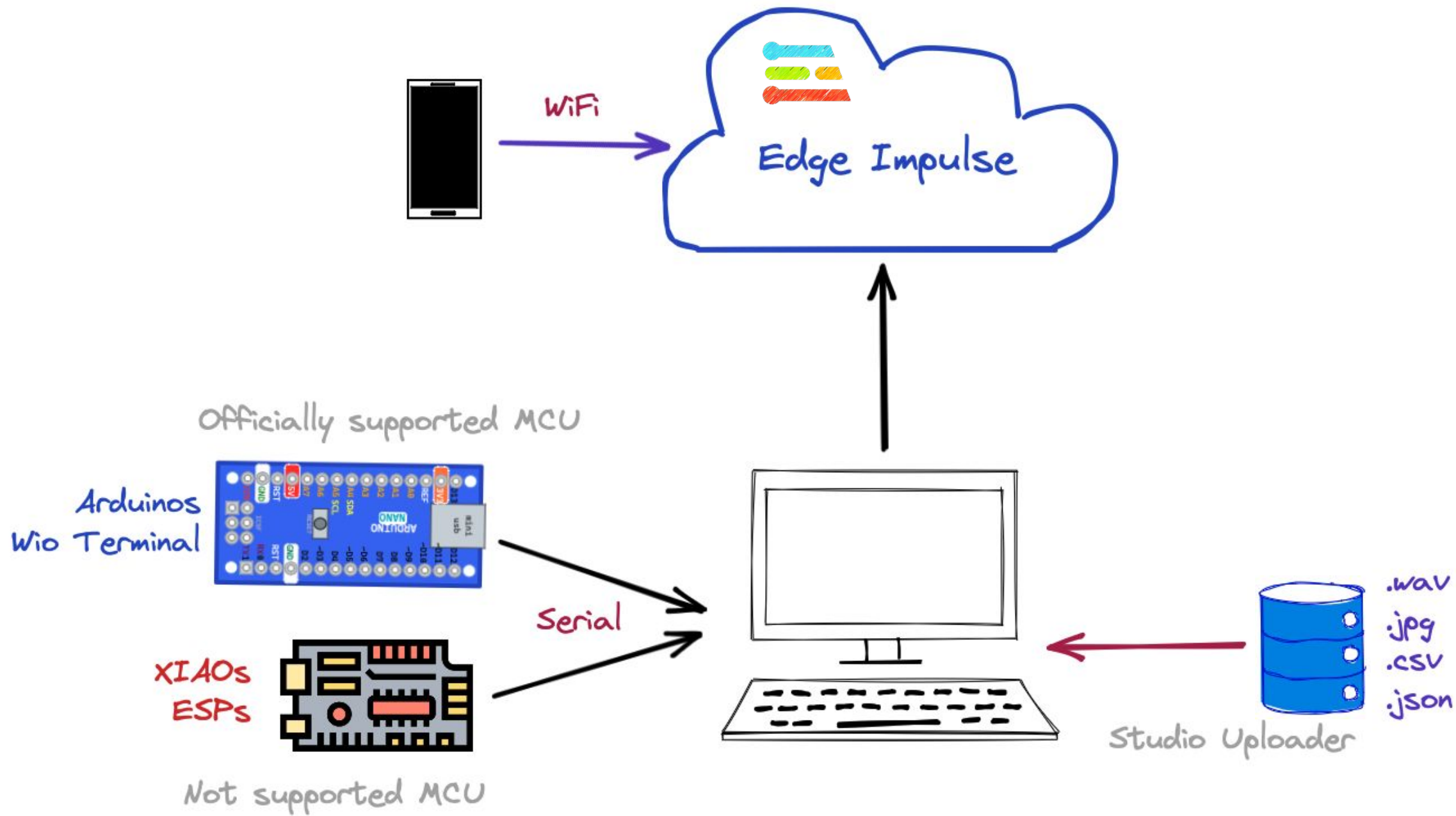
- RMS
- FFT
- PSD



TinyML under the hood: Spectral Analysis
MJRoBot (Marcelo Rovai)

El Studio Data Ingestion

Alternative methods



1. SmartPhone

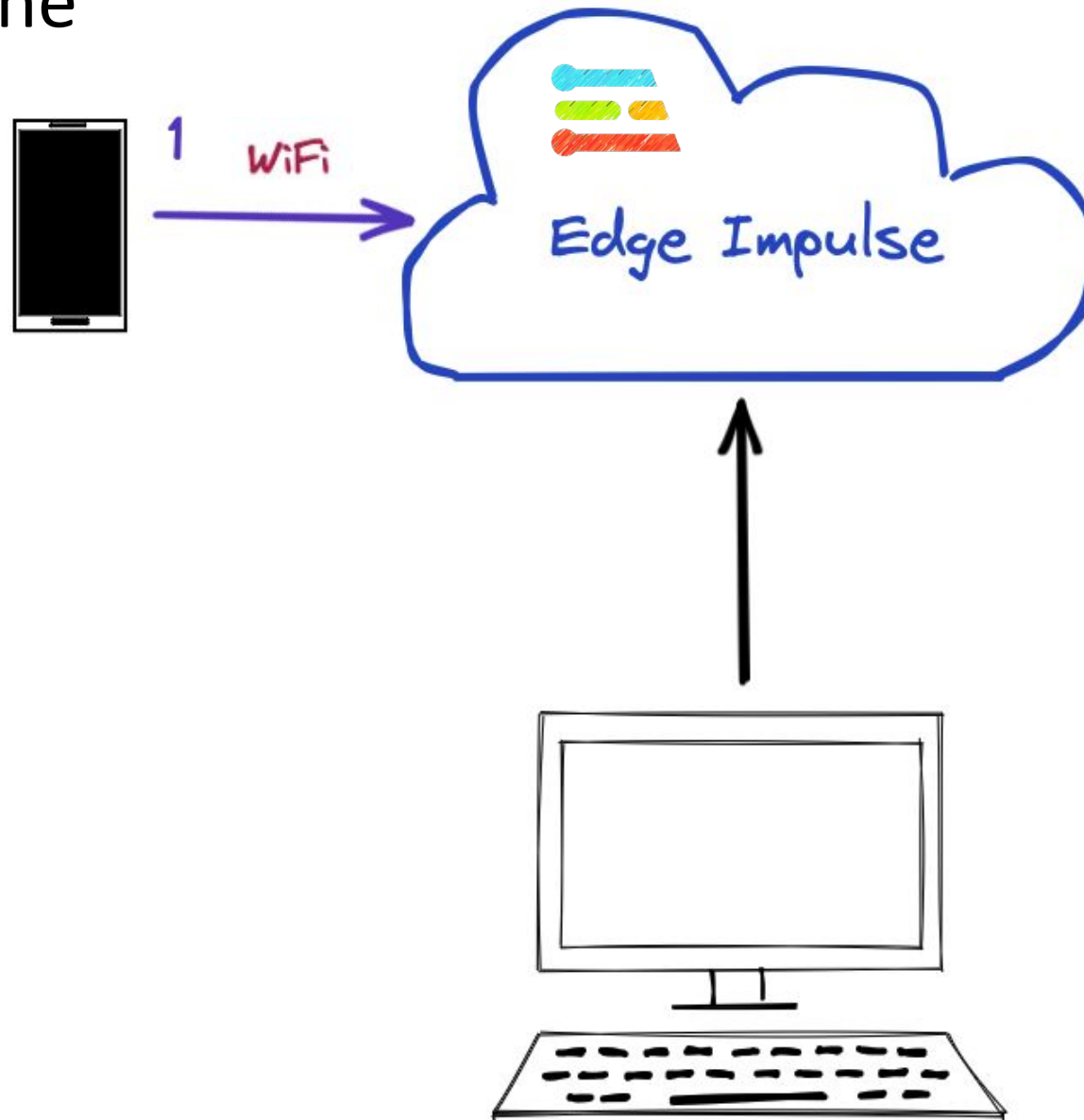
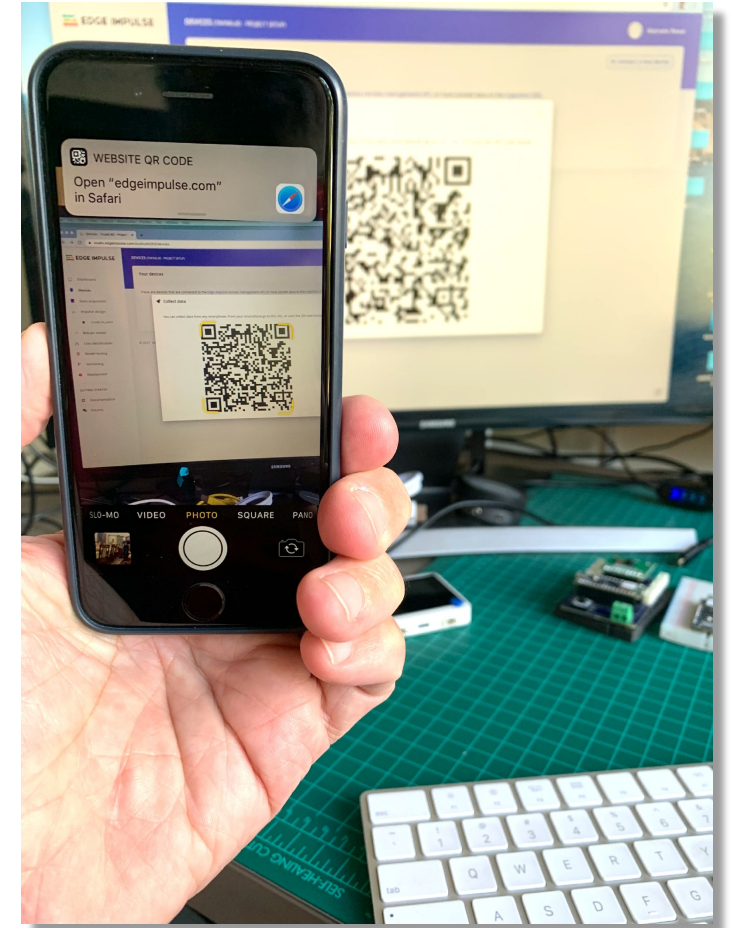
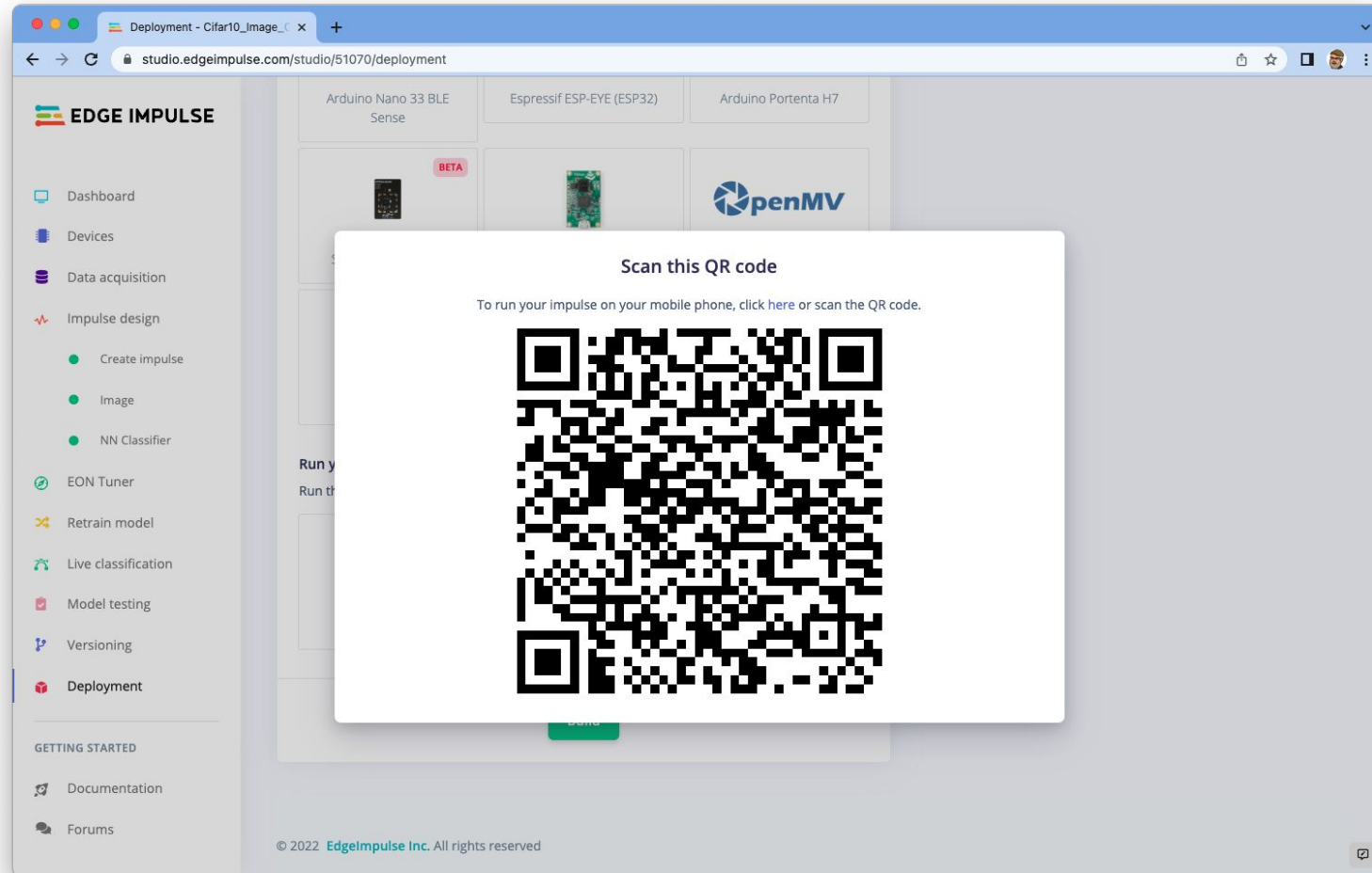


Image Classification using a **smartphone** and
Edge Impulse Studio

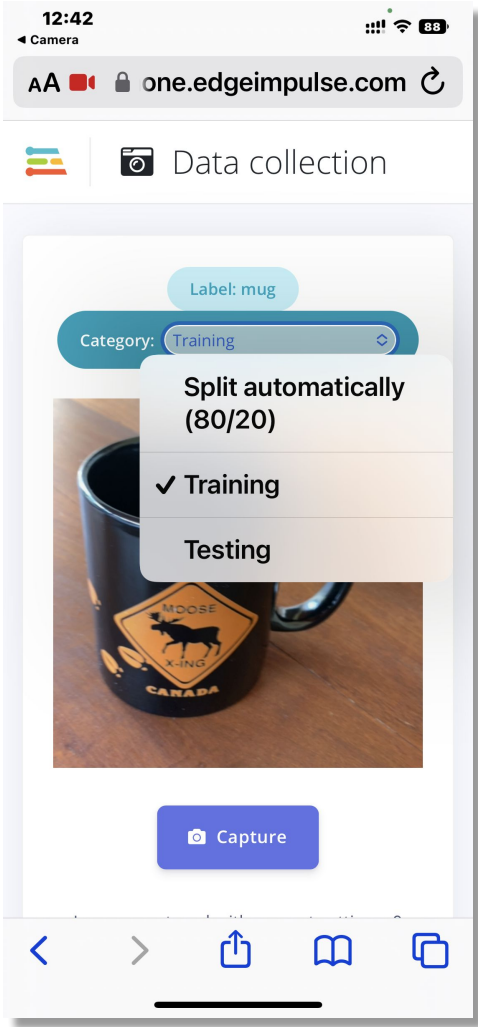
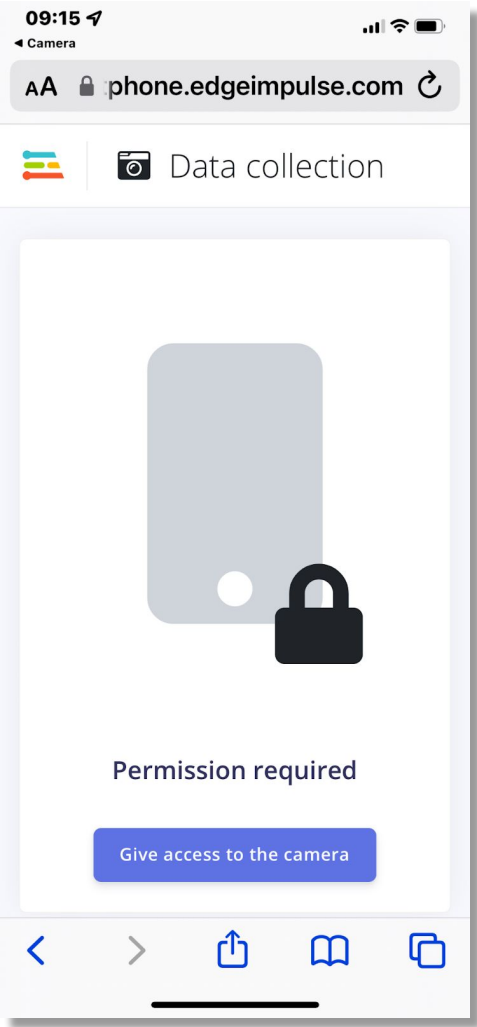
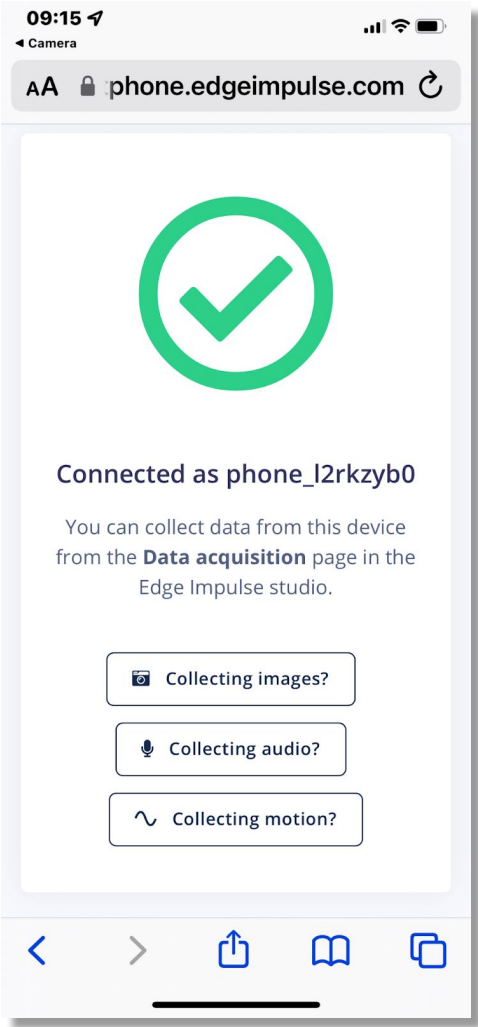
Prof. Marcelo José Raval
UNIFEI - Federal University of Itajubá, Brazil
TinyML40 Academic Network Co-Chair



1. Data Ingestion using Smart Phone



1. Data Capture and model Training




1. Off-Line Inference

14:48 63%

AA one.edgeimpulse.com

Classifier



Inferencing...

background


BACKGROUND	MUG
1...	0.00

< > Share Bookmarks Copy

14:49 62%

AA one.edgeimpulse.com

Classifier



Inferencing...

mug


BACKGROUND	MUG
1...	1.00

< > Share Bookmarks Copy

14:49 61%

AA one.edgeimpulse.com

Classifier



Inferencing...

mug


BACKGROUND	MUG
2...	0.98

< > Share Bookmarks Copy

14:49 60%

AA one.edgeimpulse.com

Classifier



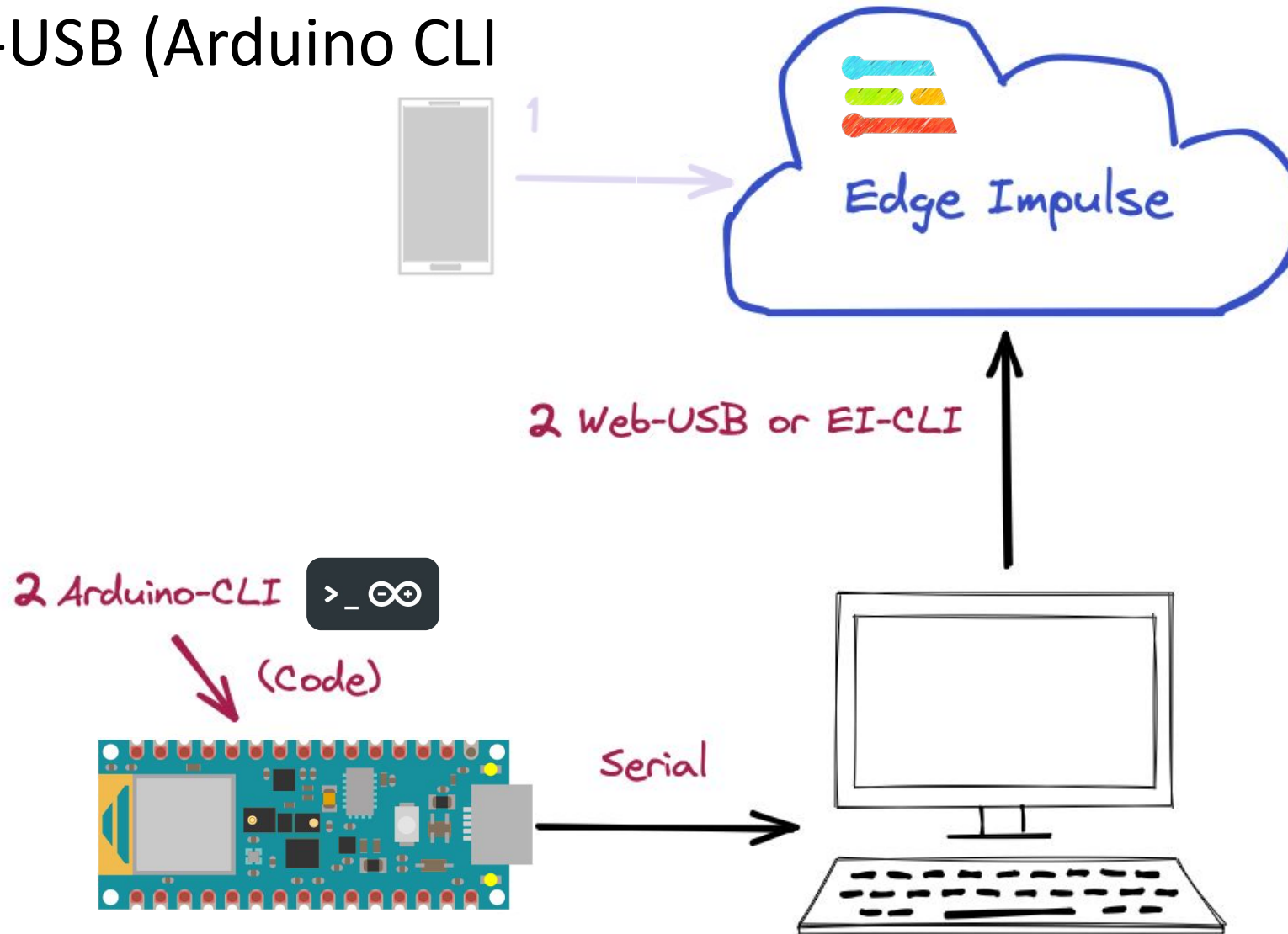
Inferencing...

uncertain

BACKGROUND	MUG
2...	0.70

< > Share Bookmarks Copy

2. Web-USB (Arduino CLI)



Issue: Limited MCU and sensors



TinyML Arduino Kit
Connection to Edge Impulse

Prof. Marcelo José Raval
UNIFEI - Federal University of Itajubá, Brazil
TinyML40 Academic Network Co-Chair



2. Data Ingestion using Arduino-Cli + Web-USB (or EI-CLI)

The image illustrates the process of data ingestion using an Arduino Nano 33 BLE development board and the Edge Impulse Studio web interface.

Terminal Output:

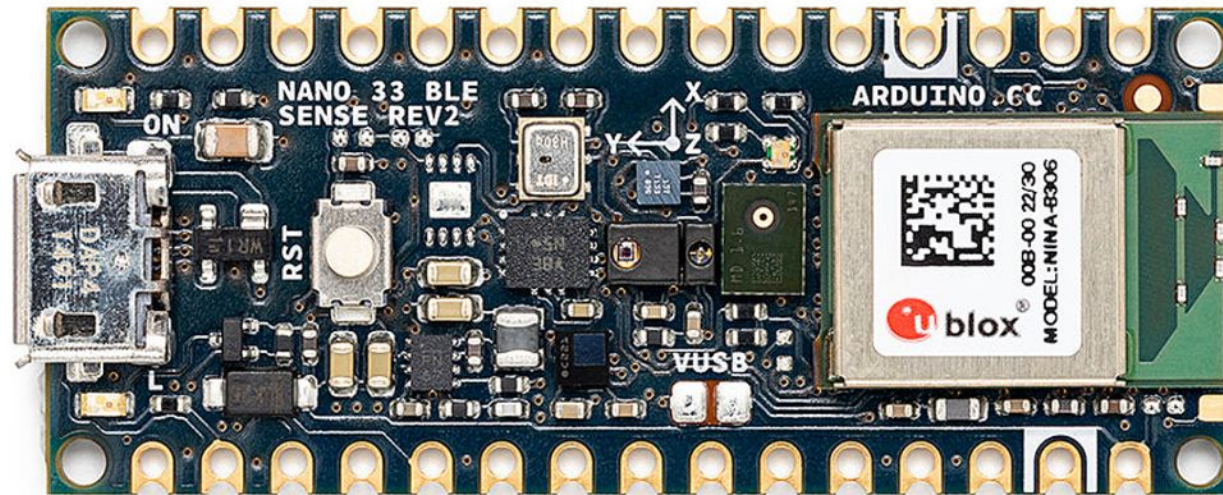
```
C:\WINDOWS\system32\cmd.exe
Finding Arduino Mbed core...
arduino:mbed_nano 2.0.0 2.0.0 Arduino Mbed OS Nano Boards
Finding Arduino Mbed core OK
Finding Arduino Nano 33 BLE...
Finding Arduino Nano 33 BLE OK at COM11
arduino:mbed_nano 2.0.0 2.0.0 Arduino Mbed OS Nano Boards
Device      : nRF52840-QIAA
Version     : Arduino Bootloader (SAM-BA extended) 2.0 [Arduino:IKXYZ]
Address     : 0x0
Pages      : 256
Page Size  : 4096 bytes
Total Size : 1024KB
Planes     : 1
Lock Regions : 0
Locked     : none
Security   : false
Erase flash

Done in 0.002 seconds
Write 525440 bytes to flash (129 pages)
[=====] 100% (129/129 pages)
Done in 22.296 seconds
Flashed your Arduino Nano 33 BLE development board
To set up your development with Edge Impulse, run 'edge-impulse'
To run your impulse on your development board, run 'edge-impulse run'
Pressione qualquer tecla para continuar. . .
```

Web Interface:

The screenshot shows the Edge Impulse Studio web interface. A dialog box prompts for connection to a serial port, listing available devices. The "Nano 33 BLE (cu.usbmodem144301) - Paired" device is selected. A "Connect" button is highlighted with a red dashed box. In the background, the "Record new data" section has a "Connect using WebUSB" button also highlighted with a red dashed box.

Arduino Nano 33 BLE Sense Rev2



- **IMU** - LSM9DS1 - 9 axis → BMI270 – 6 axis + BMM150 - 3 axis
- **Temperature and humidity sensor** - HTS221 → HS3003
- **Microphone** - MP34DT05 → MP34DT06JTR.

Record new data

Device ?

36:17:55:F9:70:F7

Label

lift

Sample length (ms.)

10000

Sensor

Inertial

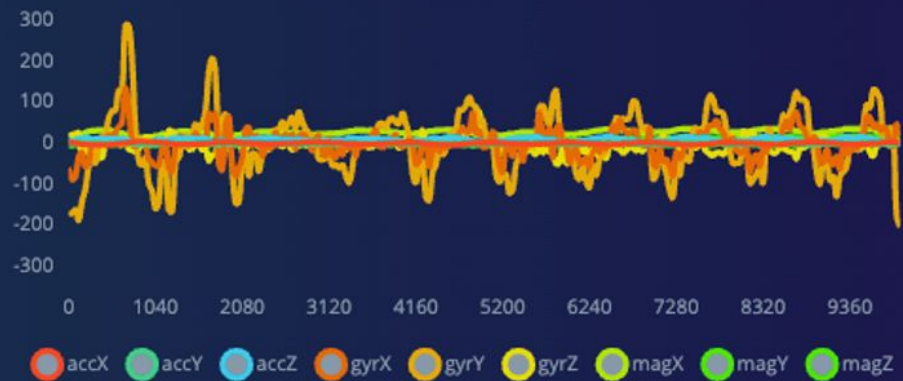
Frequency

62.5Hz

Start sampling

RAW DATA

lift.3soee1ar



Data acquisition

Impulse design

- Create impulse
- Spectral Analysis
- Classifier
- Anomaly detection

EON Tuner

Retrain model

Live classification

Model testing

Versioning

Deployment

GETTING STARTED

Documentation

Forums

Time series data

Input axes (9)

accX, accY, accZ, gyrX, gyrY, gyrZ, magX, magY, magZ

Window size



Window increase



Frequency (Hz)

62,5

Zero-pad data



Spectral Analysis

Name

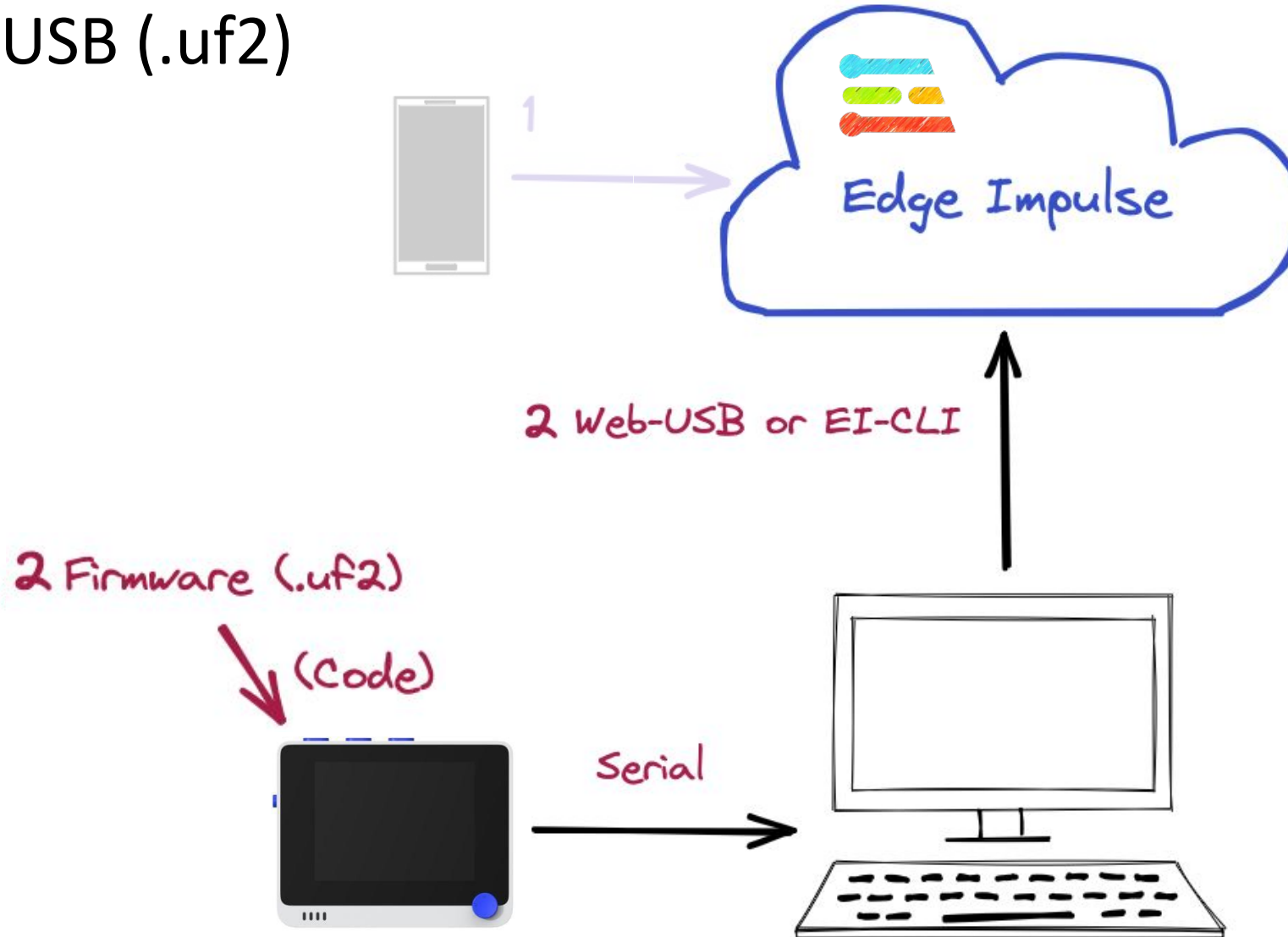
Spectral Analysis

Input axes (3)

- accX
- accY
- accZ
- gyrX
- gyrY
- gyrZ
- magX
- magY
- magZ

Add a processing

2. Web-USB (.uf2)



Issue: Limited MCU and sensors



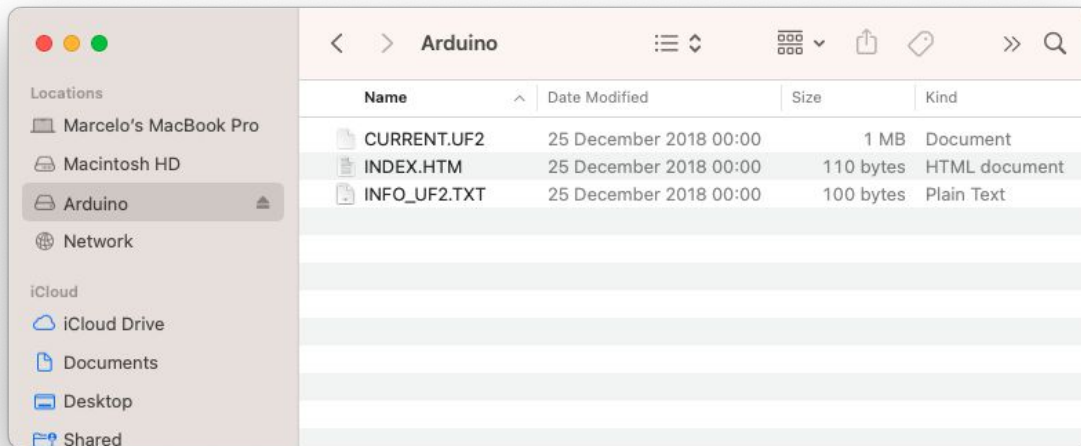
Wio Terminal
Installation & Tests

Prof. Marcelo José Rovai
UNIFEI - Federal University of Itajubá, Brazil
ThyMx-4D Academic Network Co-Chair



2. (.uf2) Firmware installation

1. Connect Wio Terminal to your computer.
2. Entering the bootloader mode by sliding the power switch twice quickly.
3. An external drive named Arduino should appear in your PC.
4. Drag the the downloaded [Edge Impulse uf2 firmware files](#) to the Arduino drive. Now, Edge Impulse is loaded on Seeeduno Wio Terminal!



Releases Tags

Latest release

1.4.0
f2ef296
Verified

Compare

AIWintermuteAI released this on Apr 15

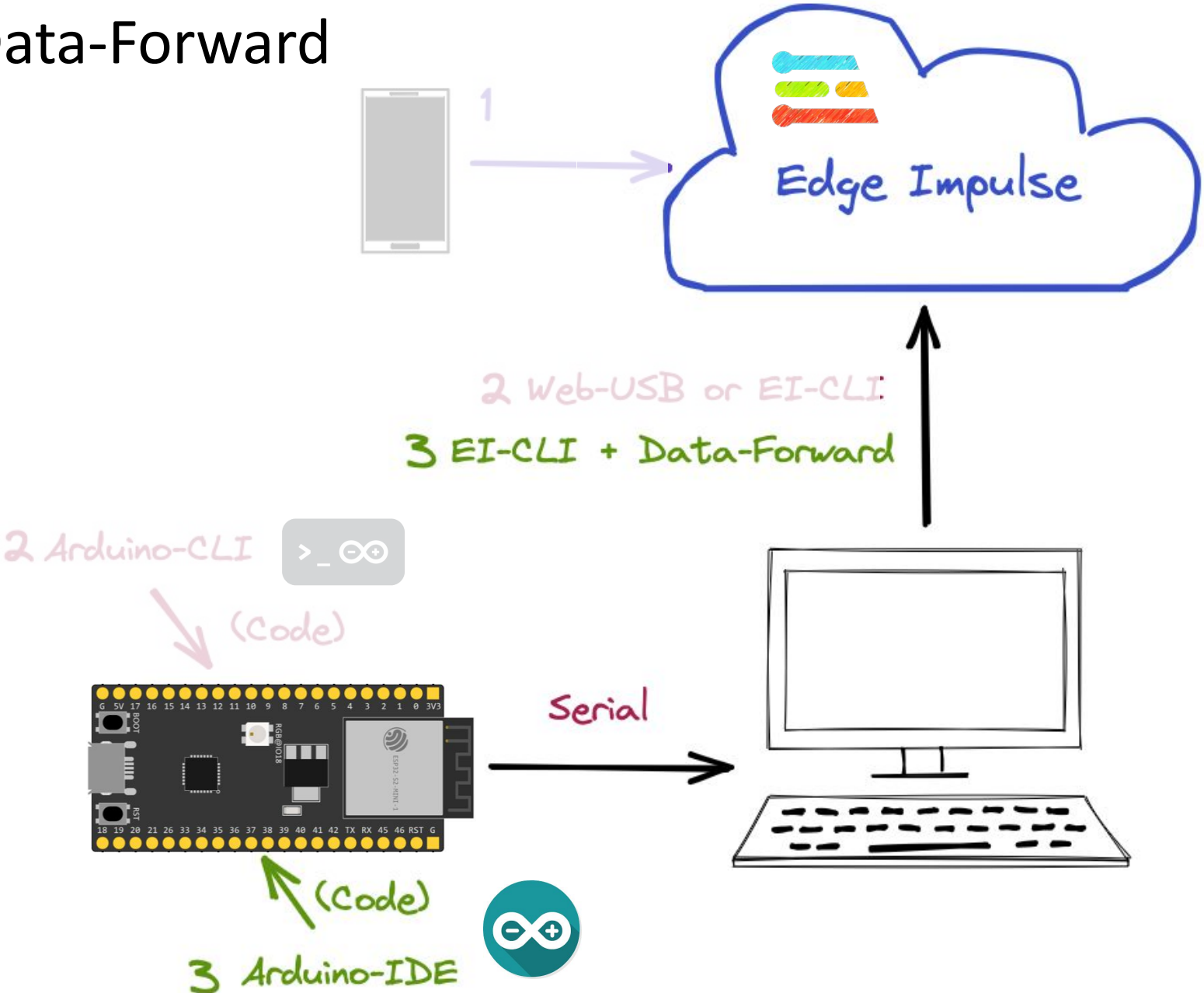
- Added built-in microphone support, recording in 16 kHz/16bit
- Added internal light sensor support

This version is not final, microphone sampling uses DMA ADC while simultaneously writing samples to flash. Quality is slightly worse than when placing samples to RAM, perhaps because of slow FLASH writing speed.

Assets 3

- [wio-terminal-ei-1.4.0.uf2](#) 239 KB
- [Source code \(zip\)](#)
- [Source code \(tar.gz\)](#)

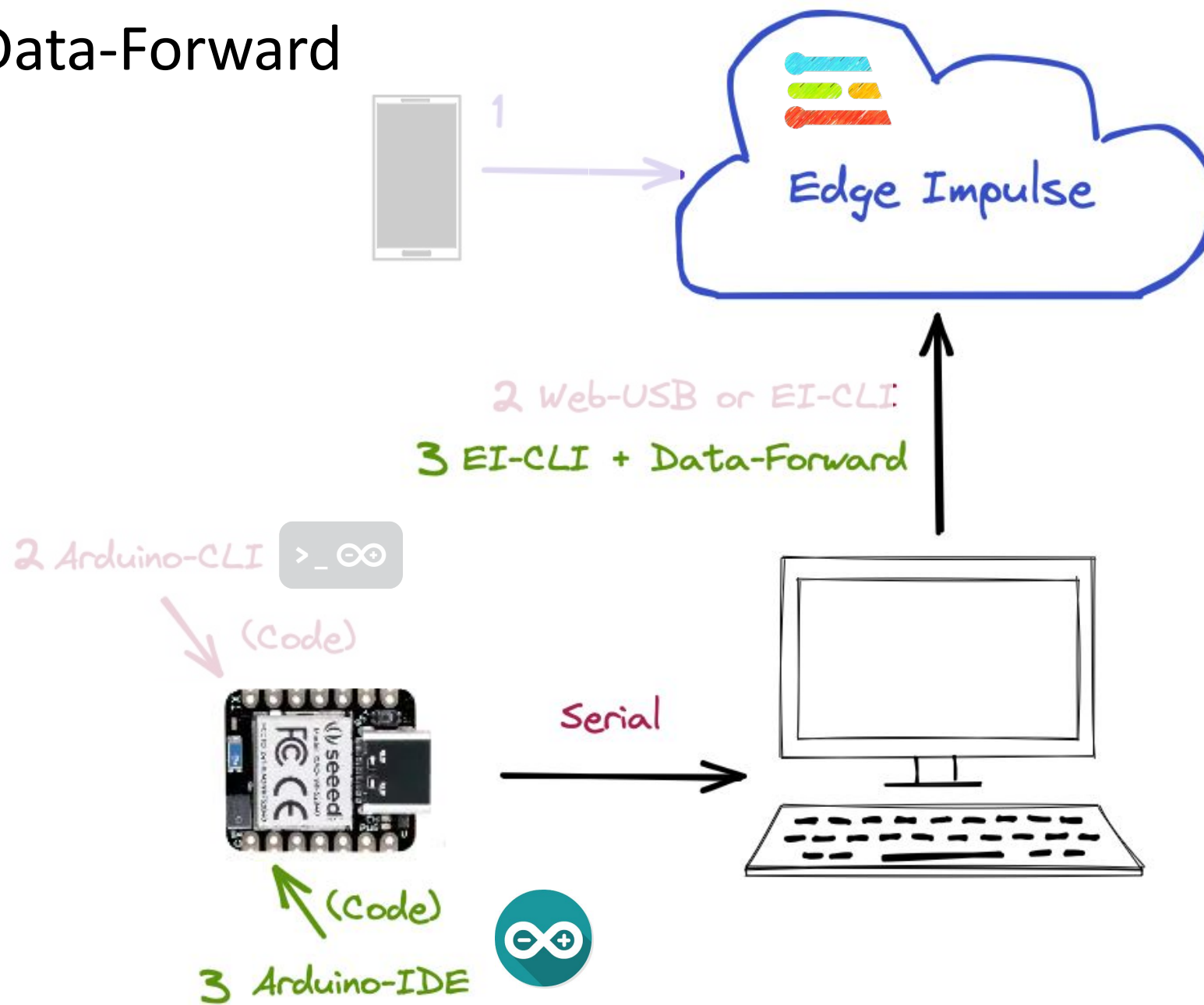
3. Data-Forward



ESP32 - Motion Classification

Prof. Marcelo José Rovati
UNIFEI - Federal University of Itajubá, Brazil
TinyML4AD Academic Network Co-Chair

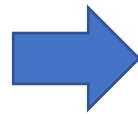
3. Data-Forward



2. Data Ingestion using El-Cli + Data Forward

```
XIAO_BLE_Sense_Accelerometer_Data_Forwarder | Arduino 1.8.19
XIAO_BLE_Sense_Accelerometer_Data_Forwarder $
9  Marcelo Rovai @July2022
10 */
11 #include "LSM6DS3.h"
12 #include "Wire.h"
13
14 //Create an instance of class LSM6DS3
15 LSM6DS3 xIMU(I2C_MODE, 0x6A); //I2C device address 0x6A
16
17 #define CONVERT_G_TO_MS2 9.80665f
18 #define FREQUENCY_HZ 50
19 #define INTERVAL_MS (1000 / (FREQUENCY_HZ + 1))
20 static unsigned long last_interval_ms = 0;
21
22 void setup() {
23   Serial.begin(115200);
24   while (!Serial);
25
26   if (xIMU.begin() != 0) {
27     Serial.println("Device error");
28   } else {
29     Serial.println("Device OK!");
30   }
31   Serial.println("Data Forwarder - Built-in IMU on the XIAO BLE Sense\n");
32 }
33
34 void loop() {
35   float x, y, z;
36   if (millis() > last_interval_ms + INTERVAL_MS) {
37     last_interval_ms = millis();
38     x = xIMU.readFloatAccelX();
39     y = xIMU.readFloatAccelY();
40     z = xIMU.readFloatAccelZ();
41
42     Serial.print(x * CONVERT_G_TO_MS2);
43     Serial.print('\t');
44     Serial.print(y * CONVERT_G_TO_MS2);
45     Serial.print('\t');
46     Serial.println(z * CONVERT_G_TO_MS2);
47   }
48 }
```

\$ edge-impulse-data-forwarder --clean

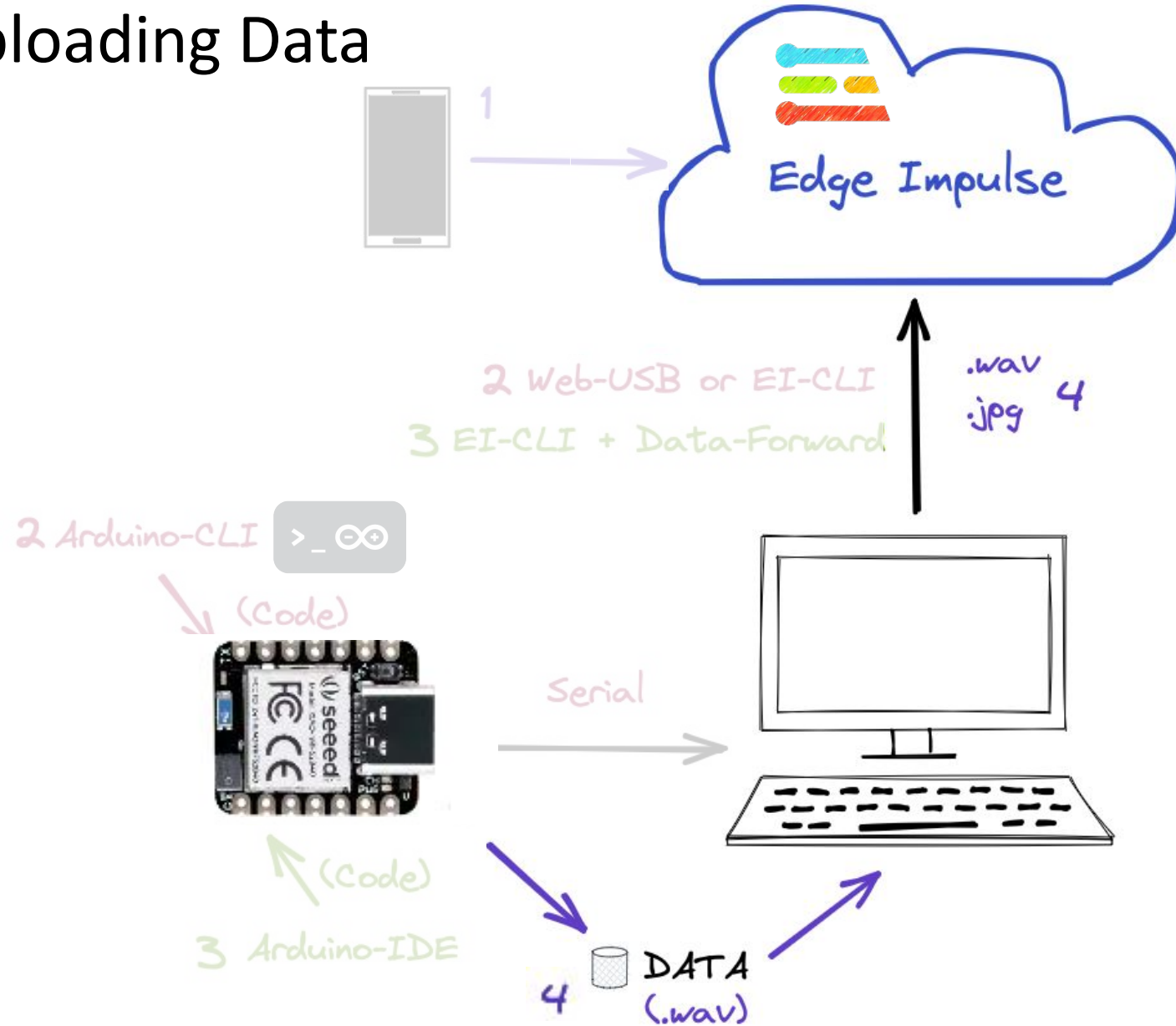


```
mjrovai — -bash — 80x41
[(base) MacBook-Pro-de-Marcelo:~ mjrovai$ edge-impulse-data-forwarder --clean
Edge Impulse data forwarder v1.12.2
[? What is your user name or e-mail address (edgeimpulse.com)? rovai@mjrobot.org ]
[? What is your password? [hidden] ]
Endpoints:
  Websocket: wss://remote-mgmt.edgeimpulse.com
  API:       https://studio.edgeimpulse.com/v1
  Ingestion: https://ingestion.edgeimpulse.com

[SER] Connecting to /dev/tty.usbmodem144301
[SER] Serial is connected (4A:5A:36:17:55:F9:70:F7)
[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) /
  IESTI01_Input_Data_Test
[SER] Detecting data frequency...
[SER] Detected data frequency: 51Hz
[? 3 sensor axes detected (example values: [-0.88,-0.34,9.82]). What do you want
  to call them? Separate the names with ',': accX, accY,
  accZ
? What name do you want to give this device? nano
[WS ] Device "nano" is now connected to project "IESTI01_Input_Data_Test"
[WS ] Go to https://studio.edgeimpulse.com/studio/39877/acquisition/training to
  build your machine learning model!
[WS ] Incoming sampling request {
  path: '/api/training/data',
  label: 'left-right',
  length: 10000,
  interval: 19.607843137254903,
  hmacKey: '6ee929b90e563aa74517f505a3ecb9c8',
  sensor: 'Sensor with 3 axes (accX, accY, accZ)'
}
```

4. Uploading Data



4. Data Ingestion using Upload existing Data

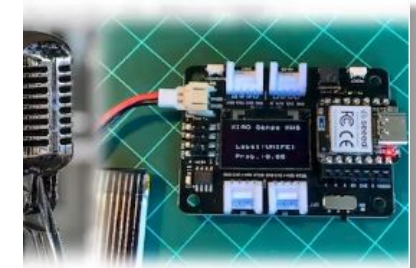
(CBOR, JSON, CSV), or as WAV, JPG or PNG files.

The screenshot displays a data ingestion interface. At the top left, a 'LABELS' section shows the number '5' and a colorful pie chart. Below it, a button labeled 'Upload existing data' is highlighted. The main interface shows a table with columns 'ADDED' and 'LENGTH'. A file explorer window is open, showing a directory structure with 'data', 'cool', and 'hot' folders. A list of WAV files is shown, with '20210710-125854.wav' selected. A blue arrow points from this file to the 'Upload existing data' dialog. The dialog has a title 'UPLOAD DATA (ICTP_PSYCHOACOUSTICS_TEMPERATURE_DEPENDENCE)' and a user profile 'MJRoBot (Marcelo Roval)'. It contains the following sections:

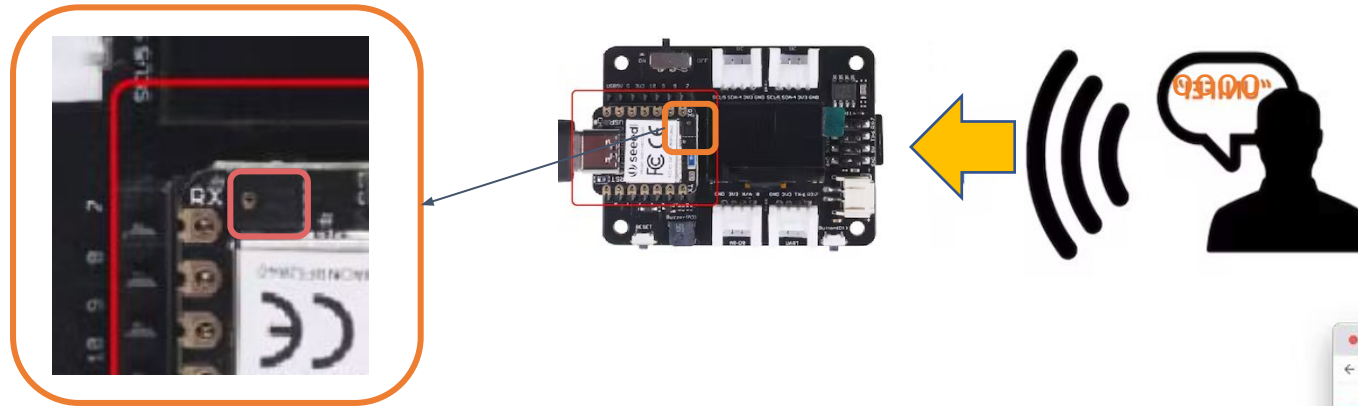
- Upload existing data:** A text box explaining that data can be uploaded in Data Acquisition Format (CBOR, JSON, CSV) or as WAV, JPG, or PNG files.
- Select files:** A 'Choose Files' button with the text 'No file chosen'.
- Upload into category:** Radio buttons for 'Automatically split between training and testing', 'Training' (selected), and 'Testing'.
- Label:** Radio buttons for 'Infer from filename' and 'Enter label:'. The 'Enter label:' option is selected, and a text input field contains the word 'hot'.
- Begin upload:** A green button at the bottom.

On the right side, an 'Upload output' panel shows a log of 14 files being uploaded successfully, each with a status of 'OK'. The log ends with the message: 'Done. Files uploaded successful: 14. Files that failed to upload: 0.' and a green status 'Job completed'.

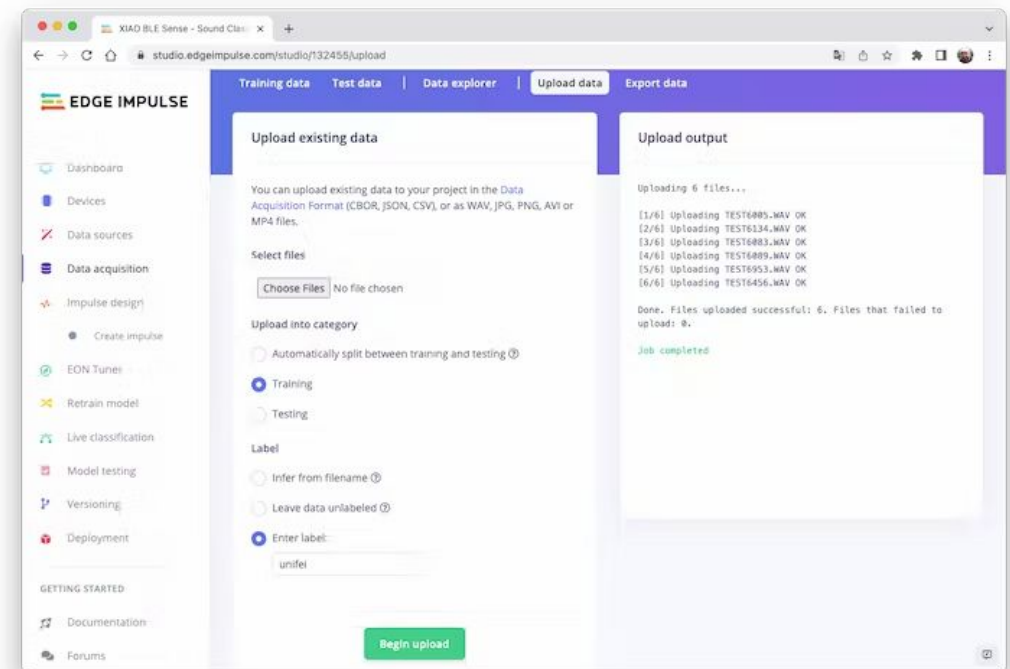
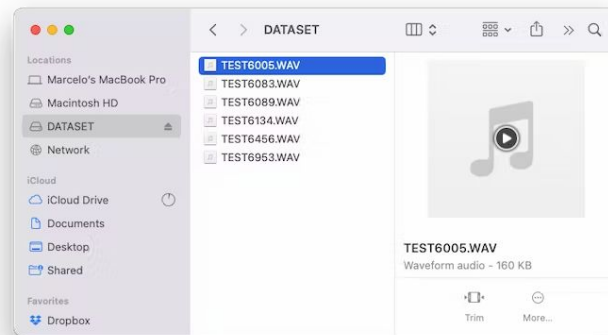
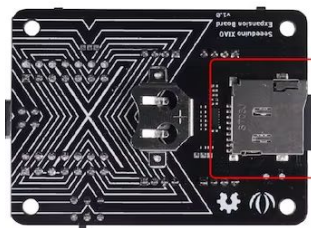
4. Uploading .wav data



TinyML Made Easy: Sound Classification (KWS)
MJRoBot (Marcelo Rovai)

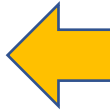
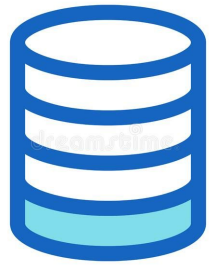


.wav



4. Uploading .jpg data

<https://github.com/YoongiKim/CIFAR-10-images>



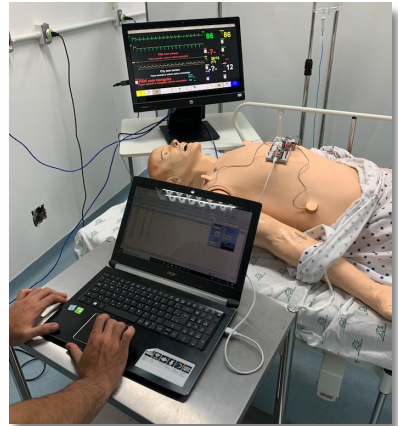
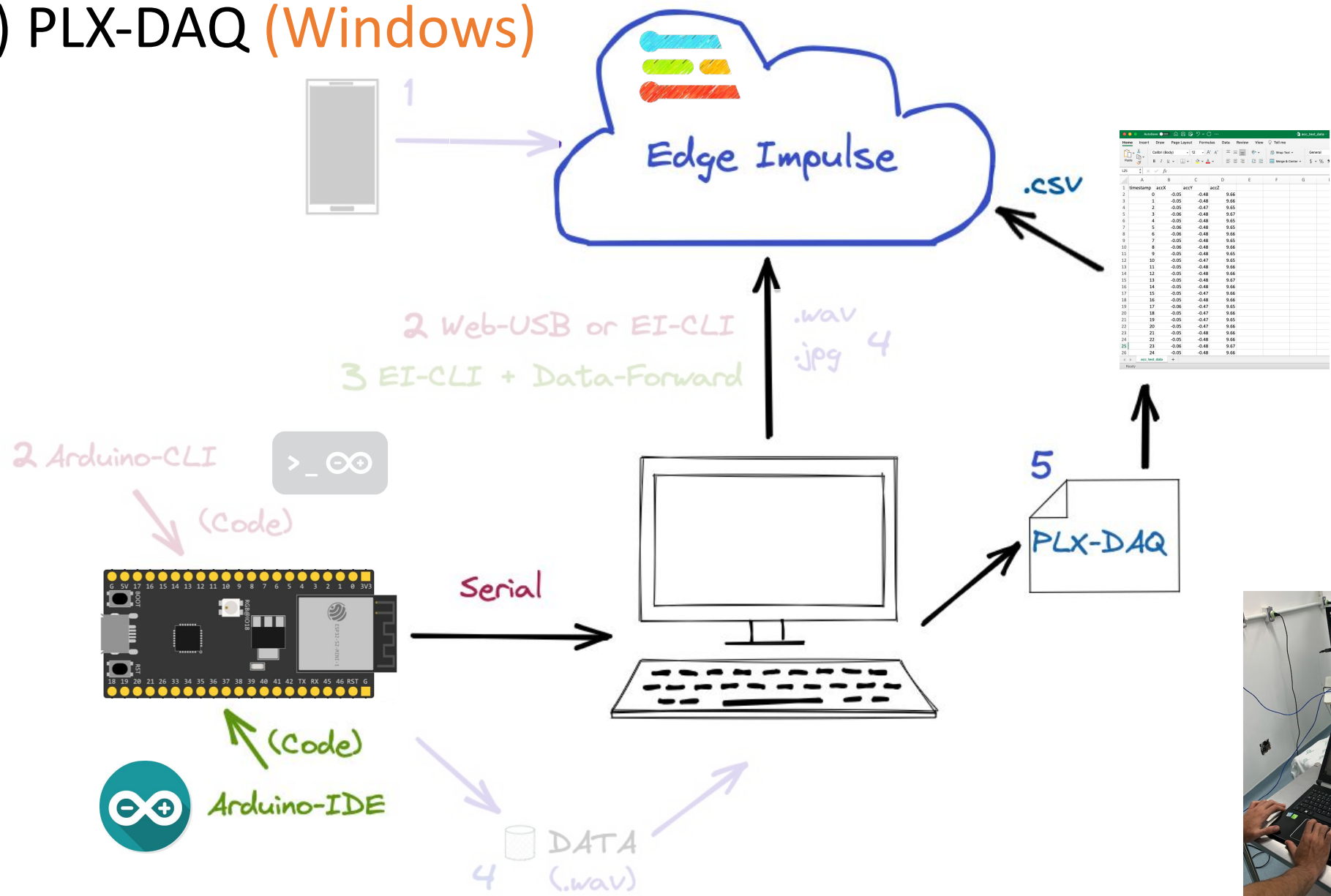
The screenshot shows the GitHub repository page for 'YoongiKim/CIFAR-10-images'. The repository contains files 'test', 'train', and 'README.md'. A 'Code' button is highlighted with an orange box. A 'Clone' dropdown menu is open, showing options for cloning via HTTPS, SSH, or GitHub CLI, and opening with GitHub Desktop or downloading as a ZIP file. The README.md content is visible below, stating 'CIFAR-10-images' and 'CIFAR-10 raw jpeg images'.

The slide features the UNIFEI logo and text: 'Workshop on Scientific Use of Artificial Learning on Low-Power Devices: Applications and Advanced Topics', 'Image Classification (Cifar10) using Convolutions (CNN) and Edge Impulse Studio', and 'Prof. Marcelo José Rovali, UNIFEI - Federal University of Itajubá, Brazil, TinyML4U Academic Network Co-Chair'.

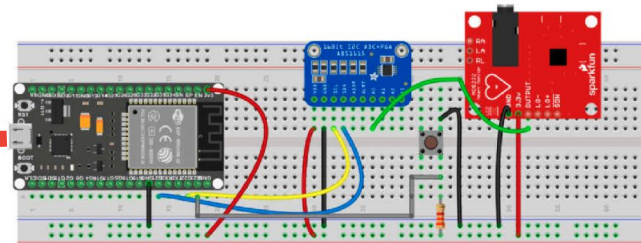
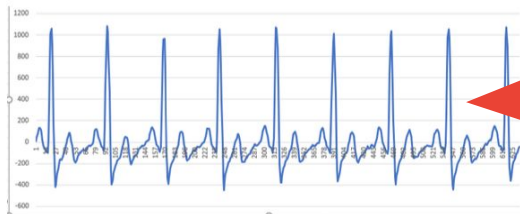
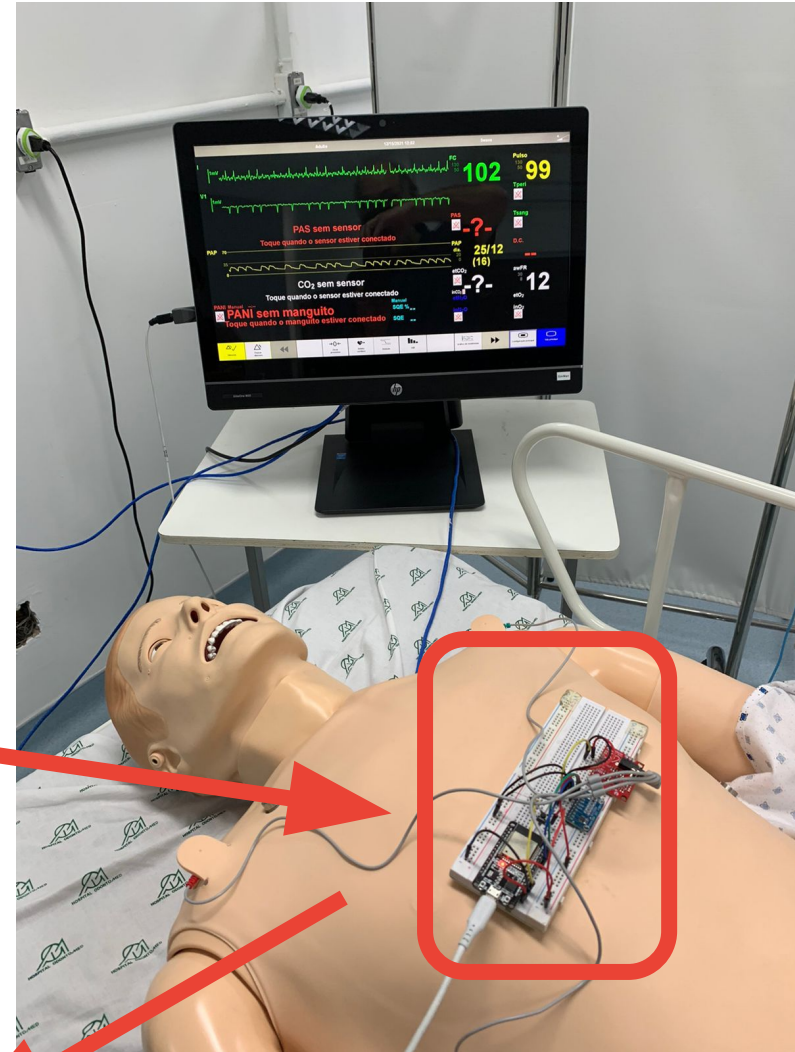
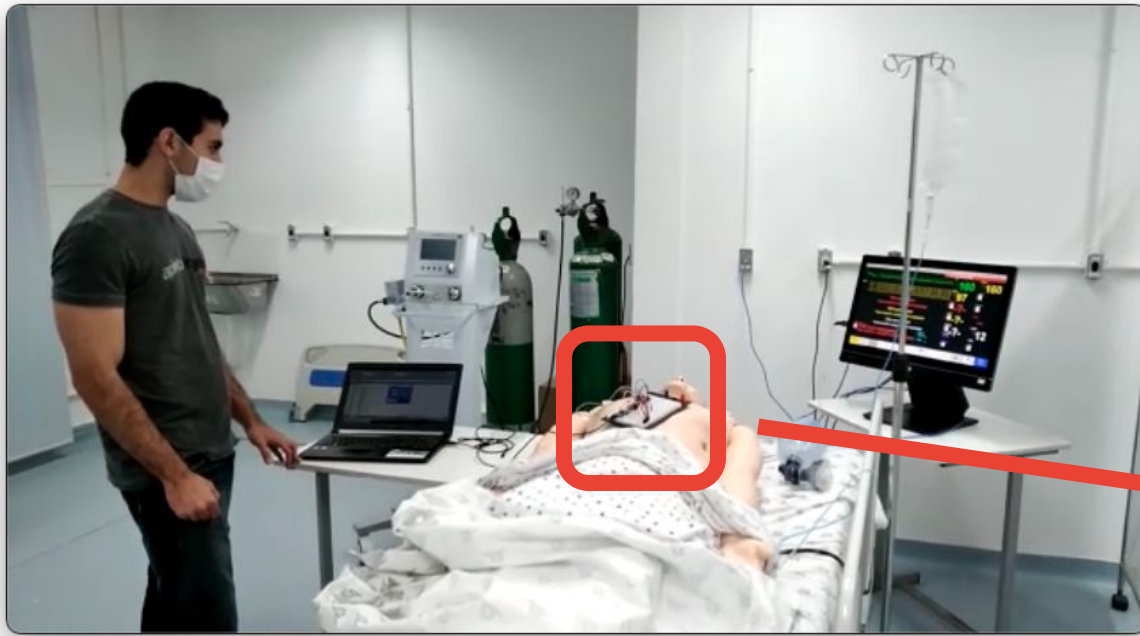


The screenshot shows the Edge Impulse Studio 'Upload Data' interface. The 'Upload existing data' section is active, with 'Choose Files' selected. The 'Upload into category' section has 'Training' selected. The 'Label' section has 'Enter label' selected, with 'dog' entered in the text field. A 'Begin upload' button is visible at the bottom right. In the background, a file explorer window shows a directory structure with 'CIFAR-10-images-master' and 'CIFAR-10-l...s-master.zip', and a list of image files like '0000.jpg' through '0020.jpg'. A preview of a dog image is shown next to the file list.

5. (.CSV) PLX-DAQ (Windows)



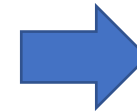
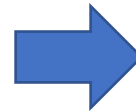
5. (.CSV) PLX-DAQ (Windows)



fritzing

5. Data Ingestion using PLX-DAQ (Windows) => Final Format: .csv

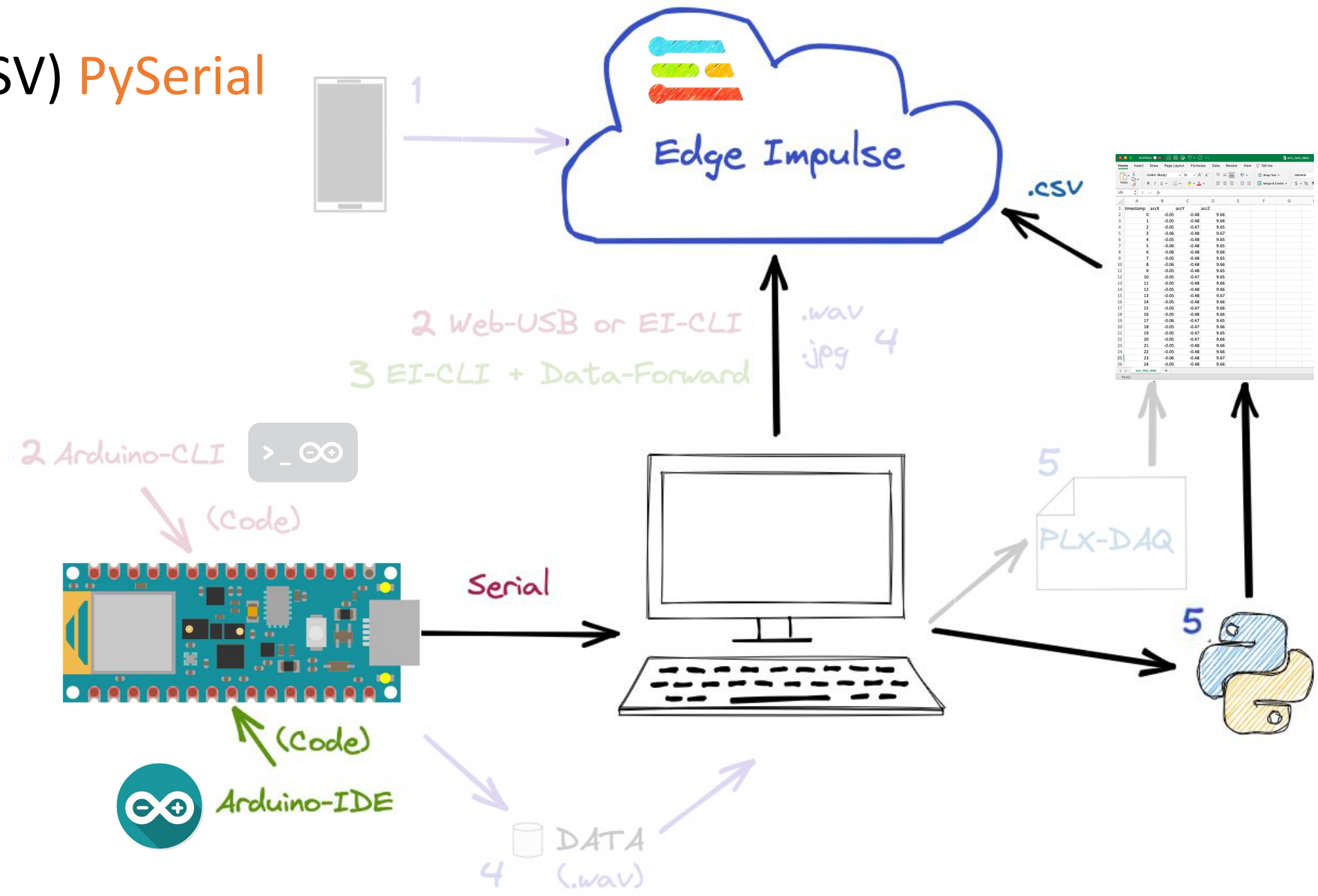
```
Capture_Ardu33_Sense_IMU_Acc
1 #include <Arduino_LSM9DS1.h>
2
3 #define CONVERT_G_TO_MS2 9.80665f
4 #define FREQUENCY_HZ 50
5 #define INTERVAL_MS (1000 / (FREQUENCY_HZ + 1))
6
7 void setup() {
8     Serial.begin(115200);
9     while (!Serial);
10    Serial.println("Started");
11
12    if (!IMU.begin()) {
13        Serial.println("Failed to initialize IMU!");
14        while (1);
15    }
16 }
17
18 void loop() {
19     static unsigned long last_interval_ms = 0;
20     float x, y, z;
21
22     if (millis() > last_interval_ms + INTERVAL_MS) {
23         last_interval_ms = millis();
24
25         IMU.readAcceleration(x, y, z);
26
27         Serial.print(x * CONVERT_G_TO_MS2);
28         Serial.print(',');
29         Serial.print(y * CONVERT_G_TO_MS2);
30         Serial.print(',');
31         Serial.println(z * CONVERT_G_TO_MS2);
32     }
33 }
```



	A	B	C	D	E	F	G	I
1	timestamp	accX	accY	accZ				
2		0	-0.05	-0.48	9.66			
3		1	-0.05	-0.48	9.66			
4		2	-0.05	-0.47	9.65			
5		3	-0.06	-0.48	9.67			
6		4	-0.05	-0.48	9.65			
7		5	-0.06	-0.48	9.65			
8		6	-0.06	-0.48	9.66			
9		7	-0.05	-0.48	9.65			
10		8	-0.06	-0.48	9.66			
11		9	-0.05	-0.48	9.65			
12		10	-0.05	-0.47	9.65			
13		11	-0.05	-0.48	9.66			
14		12	-0.05	-0.48	9.66			
15		13	-0.05	-0.48	9.67			
16		14	-0.05	-0.48	9.66			
17		15	-0.05	-0.47	9.66			
18		16	-0.05	-0.48	9.66			
19		17	-0.06	-0.47	9.65			
20		18	-0.05	-0.47	9.66			
21		19	-0.05	-0.47	9.65			
22		20	-0.05	-0.47	9.66			
23		21	-0.05	-0.48	9.66			
24		22	-0.05	-0.48	9.66			
25		23	-0.06	-0.48	9.67			
26		24	-0.05	-0.48	9.66			

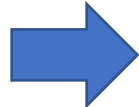
<https://www.youtube.com/watch?v=BwbmNle2CZo>

5. (.CSV) PySerial



5. Data Ingestion using Python (PySerial) => Final Format: .csv

```
Capture_Ardu33_Sense_IMU_Acc
1 #include <Arduino_LSM9DS1.h>
2
3 #define CONVERT_G_TO_MS2 9.80665f
4 #define FREQUENCY_HZ 50
5 #define INTERVAL_MS (1000 / (FREQUENCY_HZ + 1))
6
7 void setup() {
8     Serial.begin(115200);
9     while (!Serial);
10    Serial.println("Started");
11
12    if (!IMU.begin()) {
13        Serial.println("Failed to initialize IMU!");
14        while (1);
15    }
16 }
17
18 void loop() {
19     static unsigned long last_interval_ms = 0;
20     float x, y, z;
21
22     if (millis() > last_interval_ms + INTERVAL_MS) {
23         last_interval_ms = millis();
24
25         IMU.readAcceleration(x, y, z);
26
27         Serial.print(x * CONVERT_G_TO_MS2);
28         Serial.print(',');
29         Serial.print(y * CONVERT_G_TO_MS2);
30         Serial.print(',');
31         Serial.println(z * CONVERT_G_TO_MS2);
32     }
33 }
```

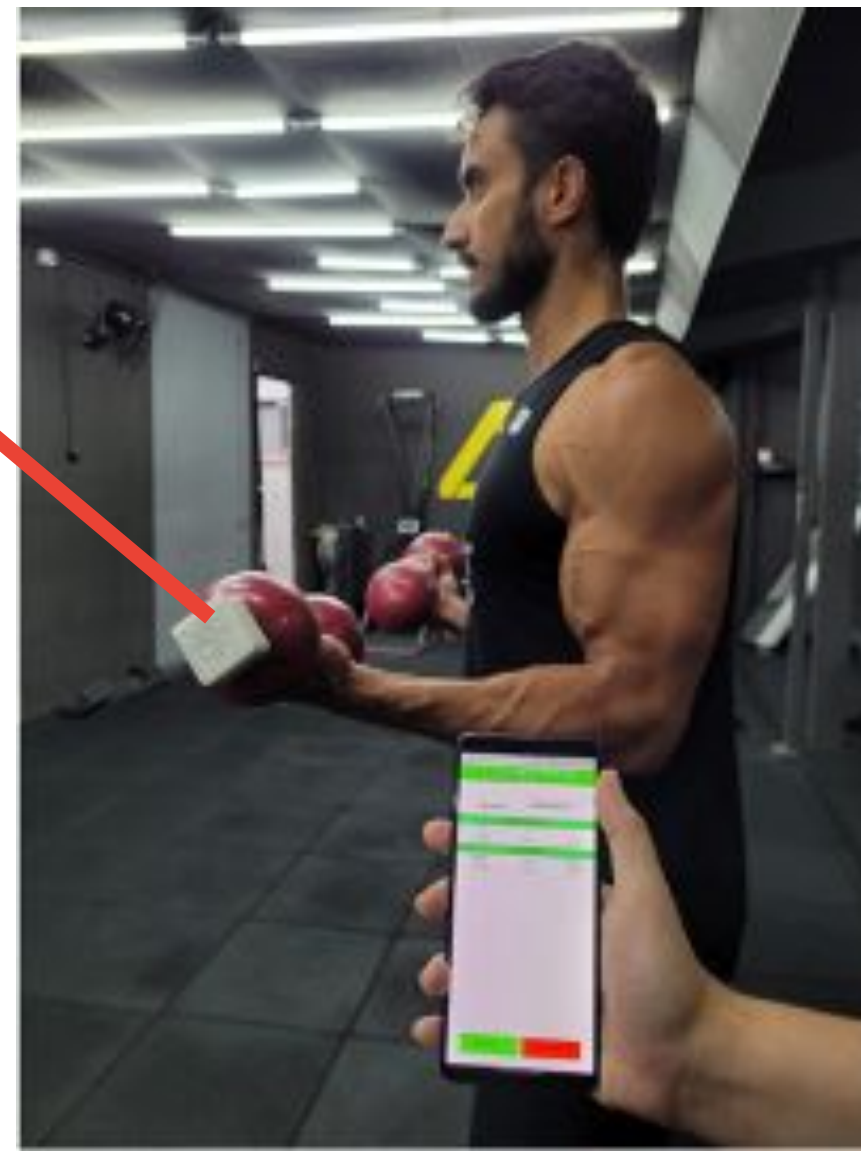
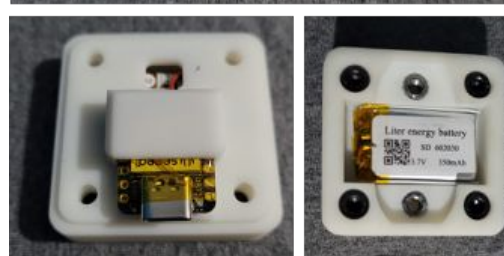


```
1 # Sensor data Logger (CSV)
2 # by Marcelo Rovai @ 13July21
3
4 import serial
5
6 arduino_port = '/dev/tty.usbmodem144301'
7 baud_rate = 115200
8 ser = serial.Serial(port=arduino_port, baudrate=baud_rate)
9
10 fileName = "acc_test_data.csv" # name of the CSV file generated
11
12 first_line = 'timestamp,accX,accY,accZ'
13 file = open(fileName, "w")
14 file.write(first_line + "\n") # write data with a newline
15 file.close()
16
17 Freq_hz = 50
18 num_seconds = 10 # number of seconds collecting data
19 samples = num_seconds * Freq_hz # number of samples to collect
20
21 sample = 0
22 while sample <= samples:
23     getData = str(ser.readline())
24     data = getData[2:][:5]
25     print(data)
26
27     file = open(fileName, "a")
28     file.write(str(sample) + "," + data + "\n")
29     sample = sample+1
30 print("Data collection complete!")
31 file.close()
```

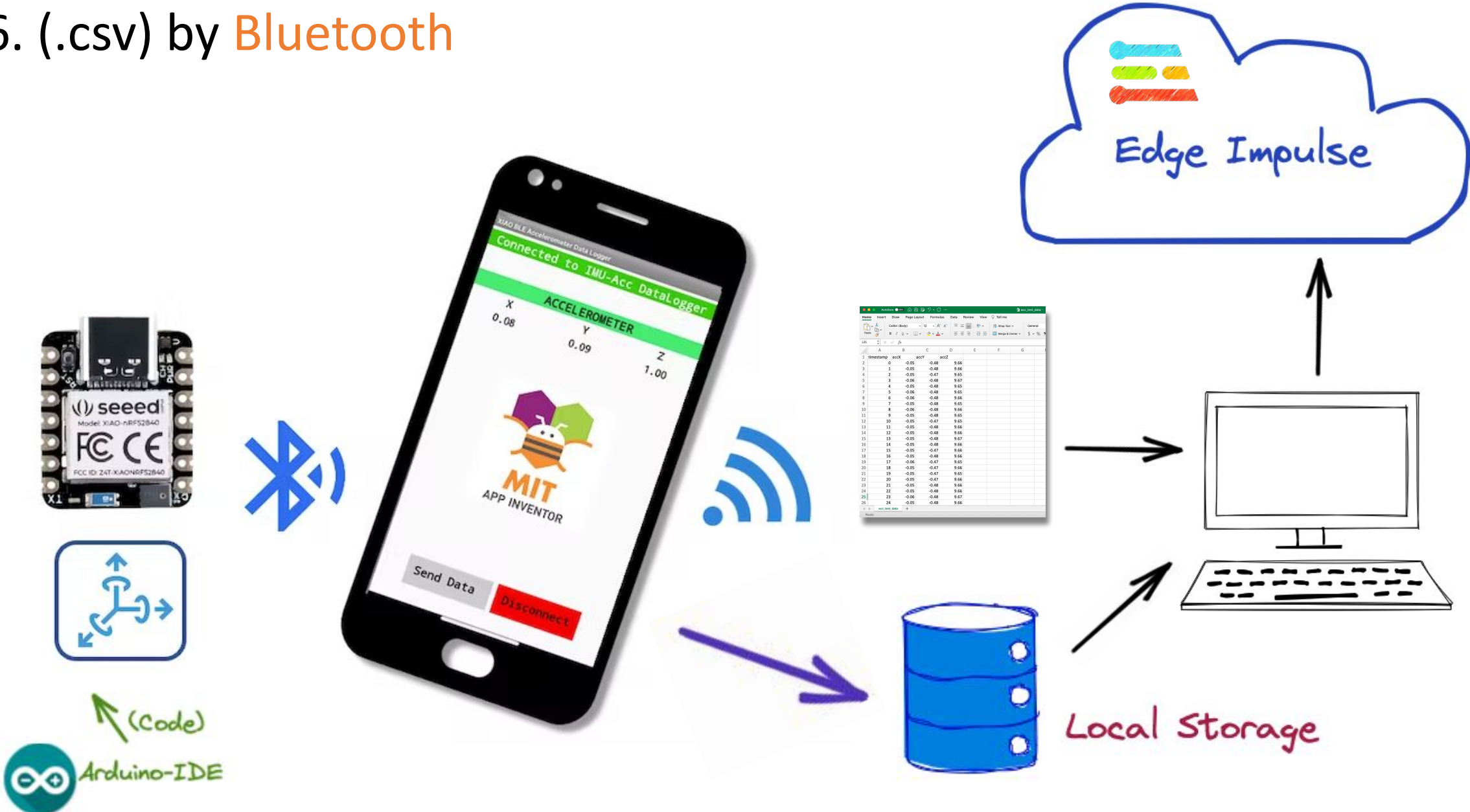


	A	B	C	D	E	F	G	I
1	timestamp	accX	accY	accZ				
2	0	-0.05	-0.48	9.66				
3	1	-0.05	-0.48	9.66				
4	2	-0.05	-0.47	9.65				
5	3	-0.06	-0.48	9.67				
6	4	-0.05	-0.48	9.65				
7	5	-0.06	-0.48	9.65				
8	6	-0.06	-0.48	9.66				
9	7	-0.05	-0.48	9.65				
10	8	-0.06	-0.48	9.66				
11	9	-0.05	-0.48	9.65				
12	10	-0.05	-0.47	9.65				
13	11	-0.05	-0.48	9.66				
14	12	-0.05	-0.48	9.66				
15	13	-0.05	-0.48	9.67				
16	14	-0.05	-0.48	9.66				
17	15	-0.05	-0.47	9.66				
18	16	-0.05	-0.48	9.66				
19	17	-0.06	-0.47	9.65				
20	18	-0.05	-0.47	9.66				
21	19	-0.05	-0.47	9.65				
22	20	-0.05	-0.47	9.66				
23	21	-0.05	-0.48	9.66				
24	22	-0.05	-0.48	9.66				
25	23	-0.06	-0.48	9.67				
26	24	-0.05	-0.48	9.66				

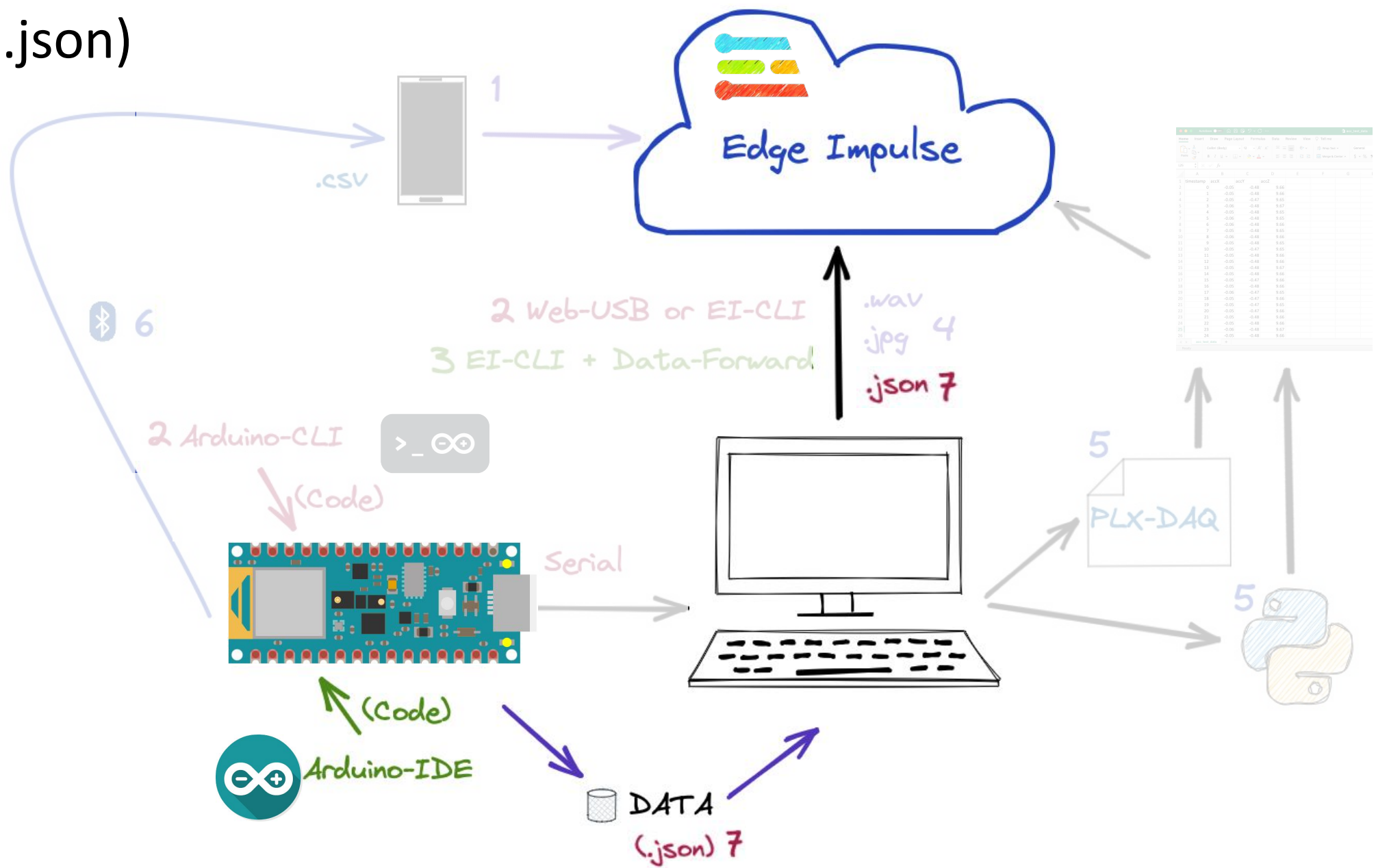
6. (.csv) by Bluetooth



6. (.csv) by Bluetooth

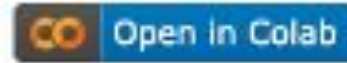


7. (.json)



7. Raw Uploader (.json files)

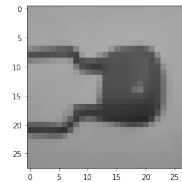
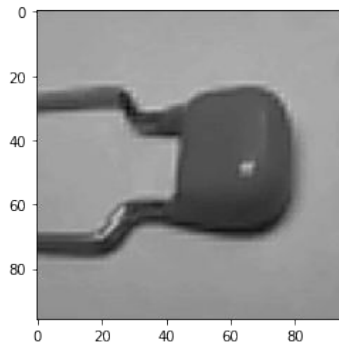
Image Classification: Raw Uploader



Run this notebook to convert images to a single row of raw, normalized values (between 0 and 1) and upload them to Edge Impulse as raw samples. Note that pixel values will be normalized to be between 0 and 1.

Create a folder named "dataset" in the /content directory and upload your images there. The images should be divided into their respective classes, where each class has its own folder with the name of the class. For example:

```
/content
|- dataset
  |- background
  |- capacitor
  |- diode
  |- led
  |- resistor
```

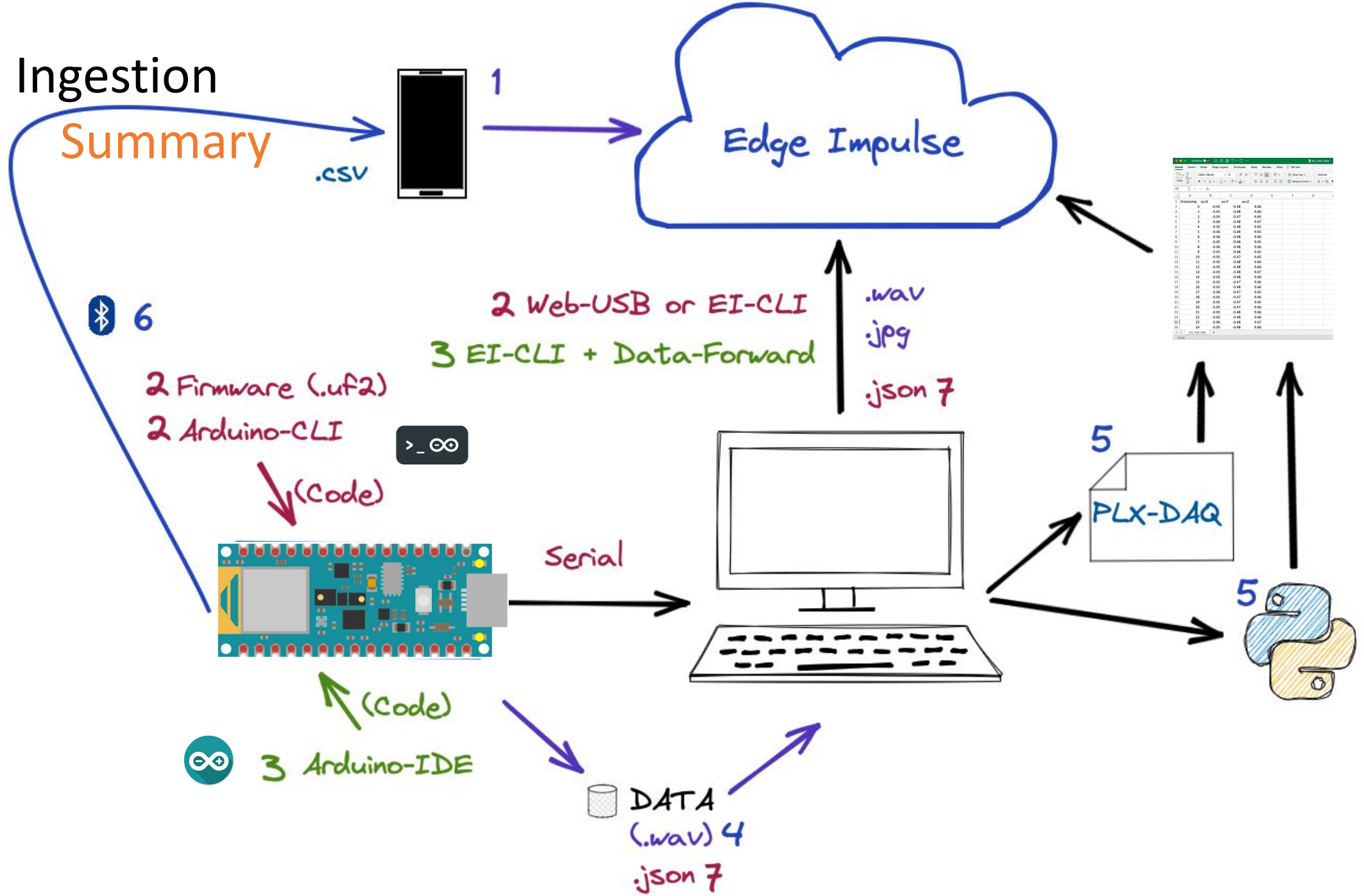


Author: EdgImpulse, Inc.
Date: June 6, 2021
License: [Apache-2.0](#)



TinyML Made Easy: Exploring Regression - White Wine Quality
MJRoBot (Marcelo Rovai)

Data Ingestion



To learn more ...

- IESTI01 TinyML - Machine Learning for Embedding Devices (Videos: Pt)
- WALC 22 – Applied AI - TinyML (Videos in Spanish)
- 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
- "Deep Learning with Python" book by François Chollet
- "TinyML" book by Pete Warden, Daniel Situnayake
- "TinyML Cookbook" by Gian Marco Iodice
- "AI at the Edge" book by Daniel Situnayake, Jenny Plunkett

On the [TinyML4D website](#), You can find lots of educational materials on TinyML. They are all free and open-source for educational uses – we ask that if you use the material, please cite them! TinyML4D is an initiative to make TinyML education available to everyone globally.

Thanks



UNIFEI