Prof. Sérgio D. Correia
Portalegre Polytechnic University
Portugal

Workshop

Inertial Measurement Units (IMUs) Data Acquisition and Filtering

“Tangible interfaces for VR applications”
International Week 2025

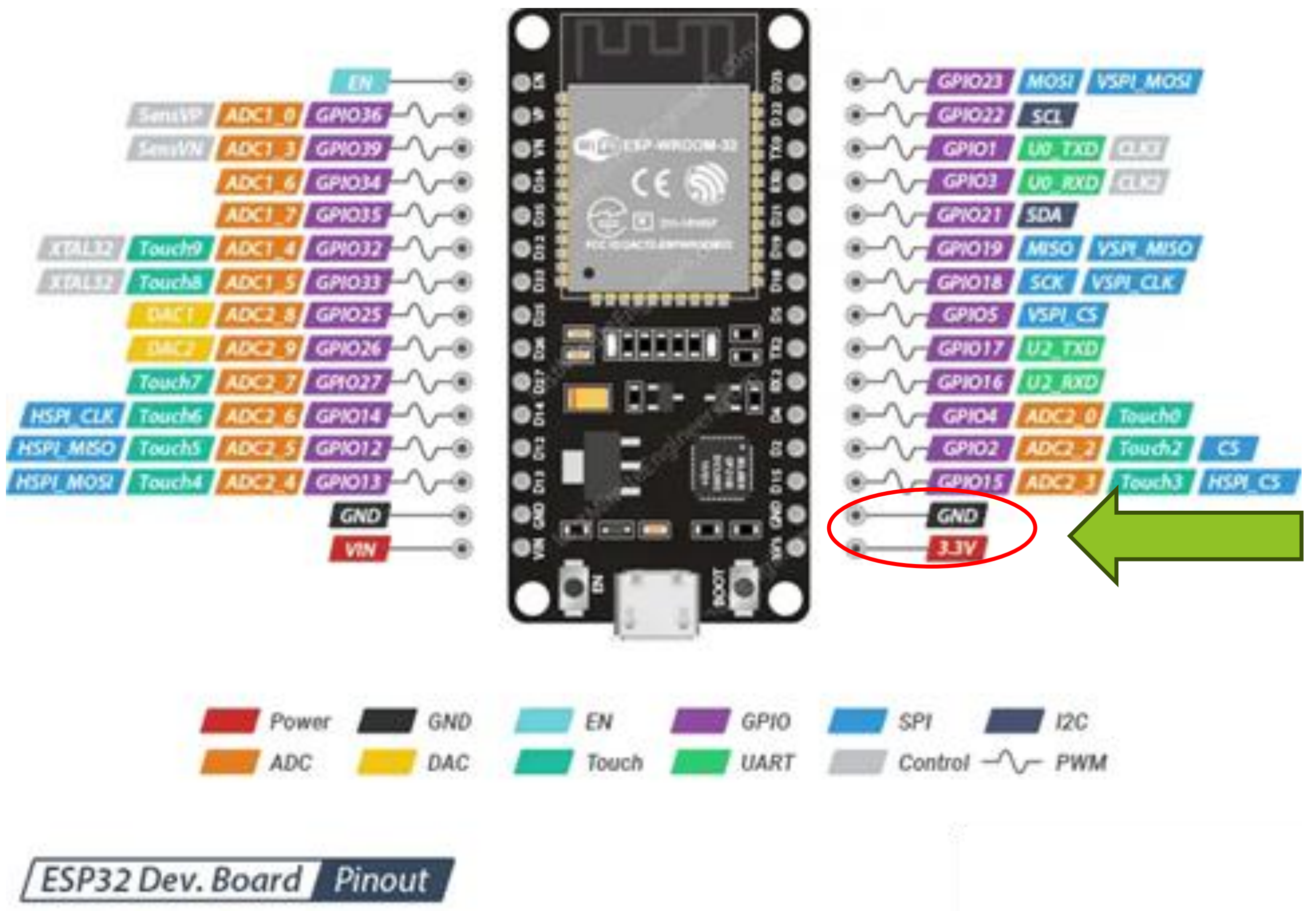
HELHa Haute École Louvain en Hainaut · isec Politécnico de Coimbra · Wrocław University of Science and Technology · PORTALEGRE POLYTECHNIC UNIVERSITY | POLITÉCNICO DE PORTALEGRE · Erasmus+ · EUCLIDES

Let's think about the Hardware

The IMU and the ESP32

What is it? How to get it to work?

1. Connecting an IMU to an ESP32



We need two digital lines to be responsible for the data transfer through the I2C BUS

Used to power the IMU module

https://lastminuteengineers.com/esp32-pinout-reference/

1. Connecting an IMU to an ESP32

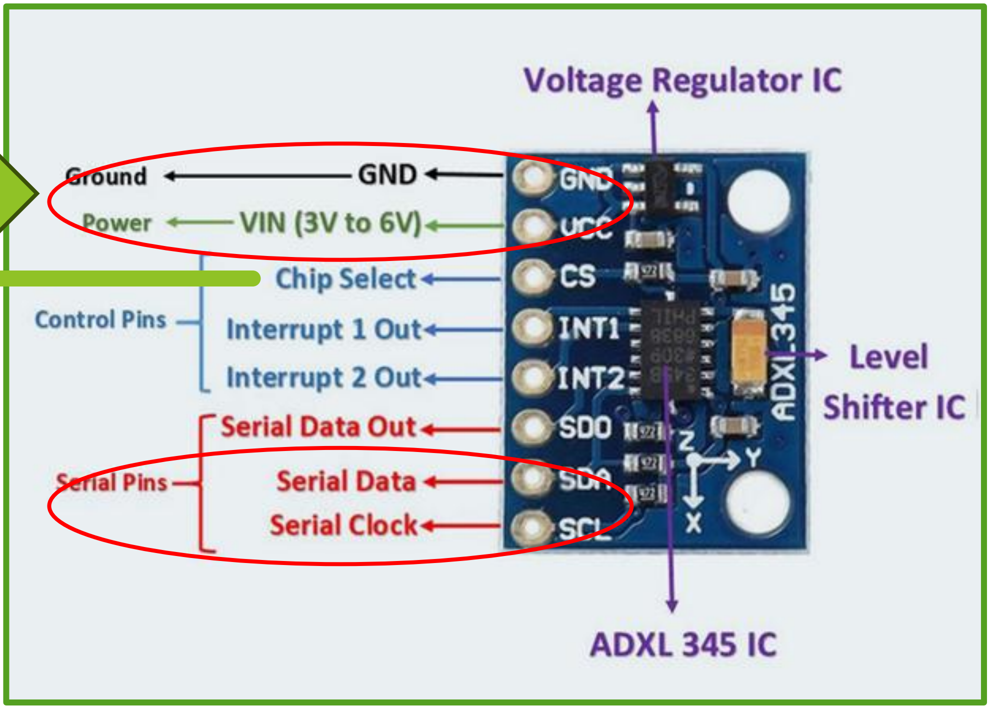
Power supply

Connected to VCC, so the chip “is always selected”

We will not use the interrupt lines

The I2S BUS will be controlled through the DAS and SCL lines

“Serial Data Out” is used for SPI BUS, thus, not for our project



https://www.theengineeringprojects.com/2025/02/adxl345-3-axis-digital-accelerometer.html

ADXL345 Module Pin Description

VCC: Power supply pin connects in the range of 3 to 5.5V DC.

GND: Connect to Supply ground.

CS (Chip Select): Chip Select Pin.

INT1 (Interrupt 1): Interrupt 1 Output Pin

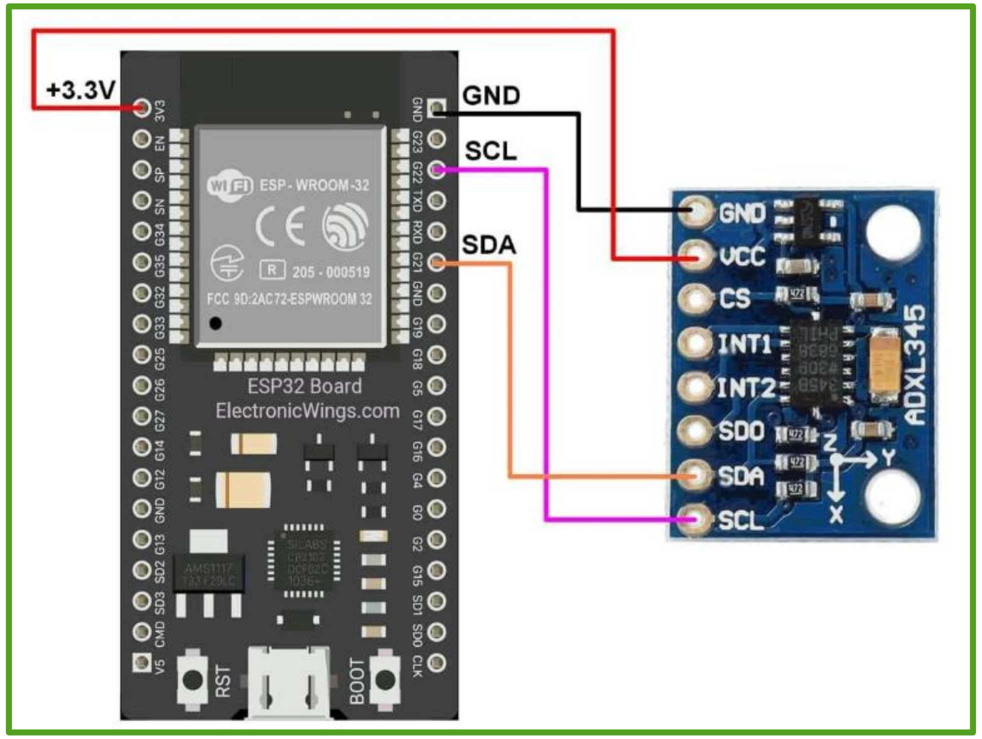
INT2 (Interrupt 2): Interrupt 2 Output Pin

SDO (Serial Data Out): Serial Data Output (SPI 4-Wire)/Alternate I2C Address Select (I2C).

SDA (Serial Data): Serial Data (I2C)/Serial Data Input (SPI 4-Wire)/Serial Data Input and Output (SPI 3-Wire).

SCL (Serial Clock): Serial Communications Clock. SCL is the clock for I2C, and SCLK is the clock for SPI.

2. Physical Connections



Note that different manufacturers may have slightly different pinouts. You should always carefully check your board pinout.

3. Software Requirements

More details will be given in our Workshop

1. **Arduino IDE Setup**

   Install ESP32 board package via Boards Manager. Select appropriate board from the list.

2. **Library Installation**

   Add Adafruit ADXL345 and Adafruit Unified Sensor libraries. Include Wire.h for I2C communication.

3. **Driver Configuration**

   Install CP210x or FTDI drivers if needed. These enable USB communication with the ESP32.

4. **Test Connection**

   Upload a simple sketch to verify the development environment works properly.

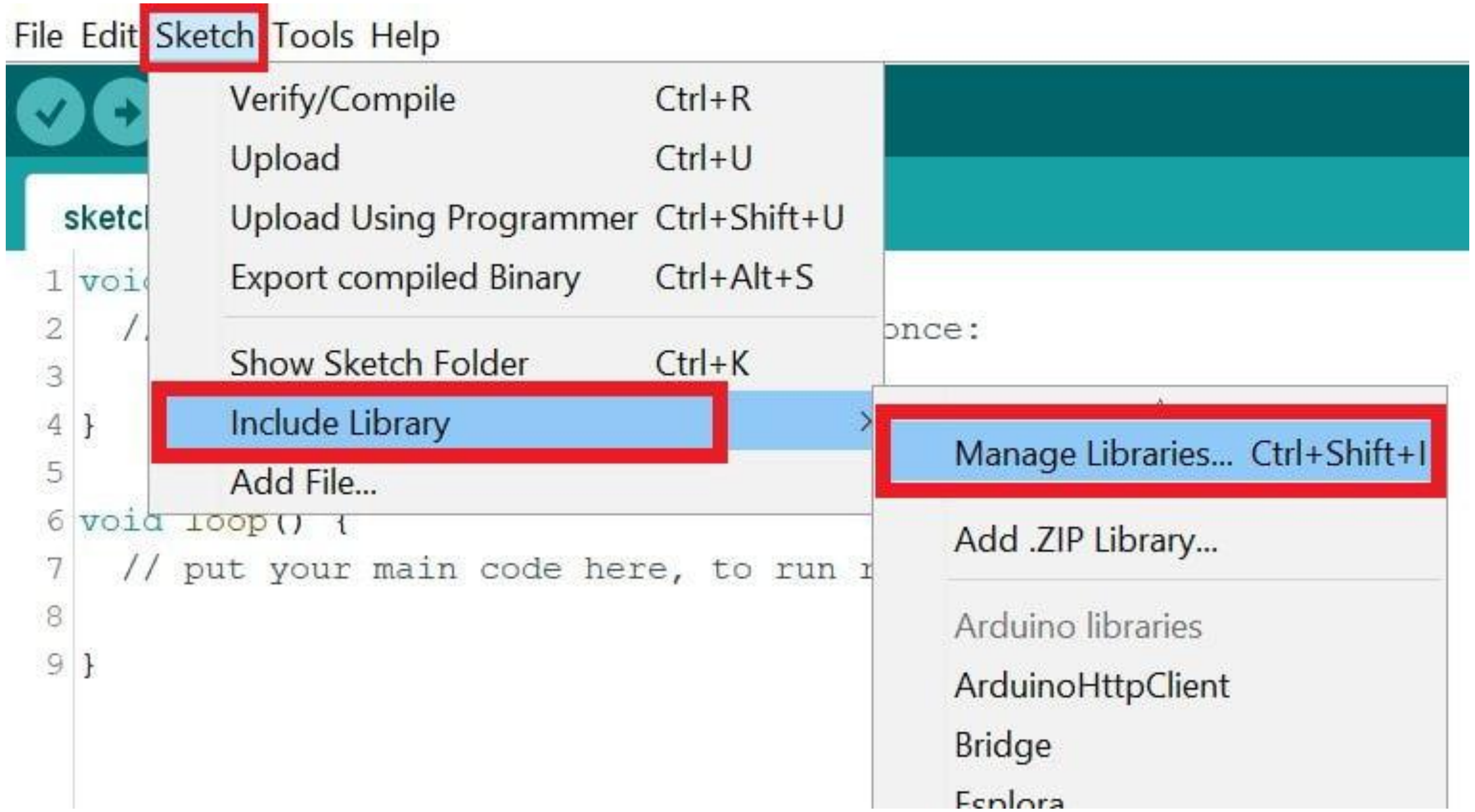
Let’s think about the Software

Connecting and Testing

Let’s code!

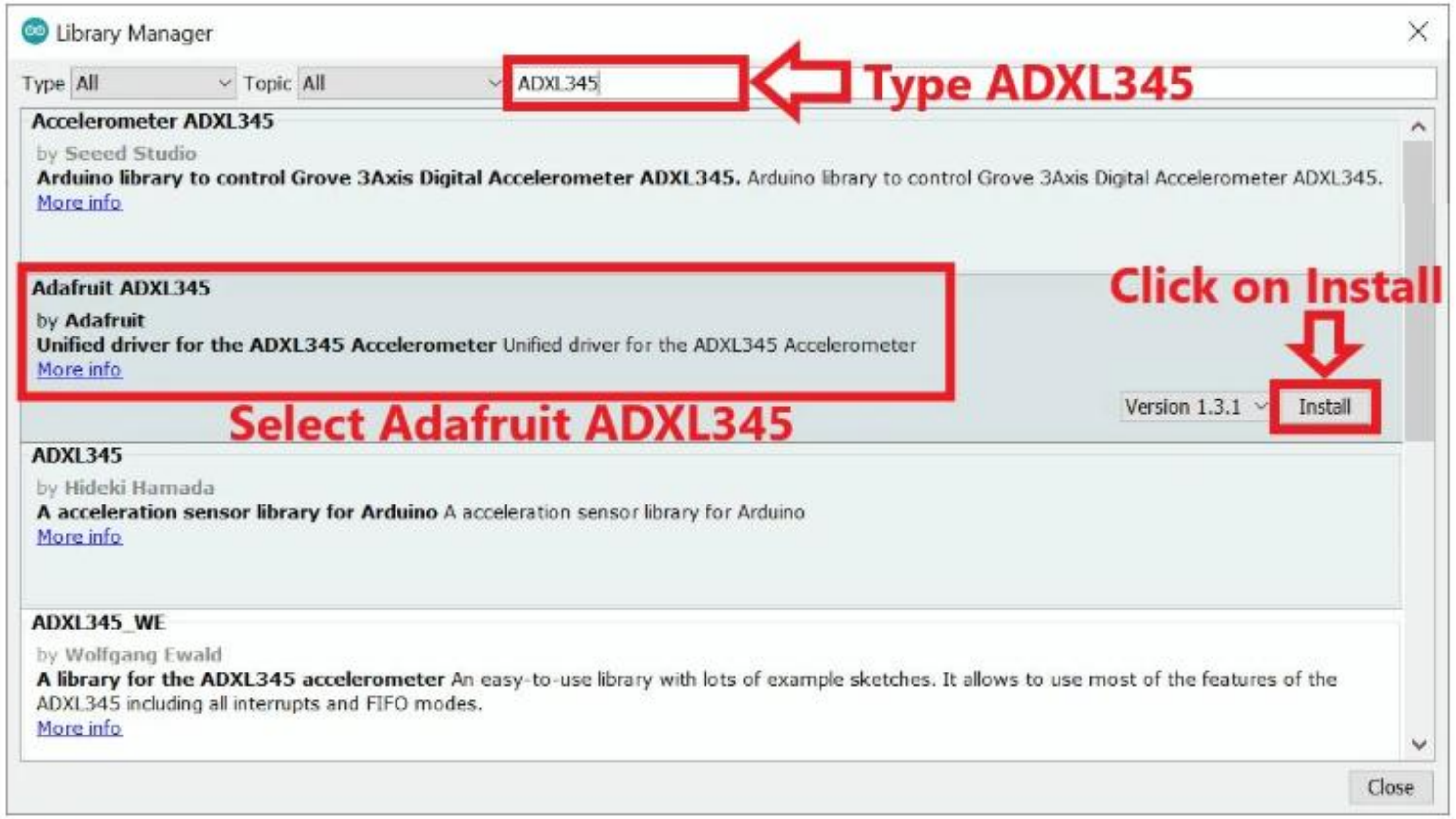
3. Instaling Dependencies

1



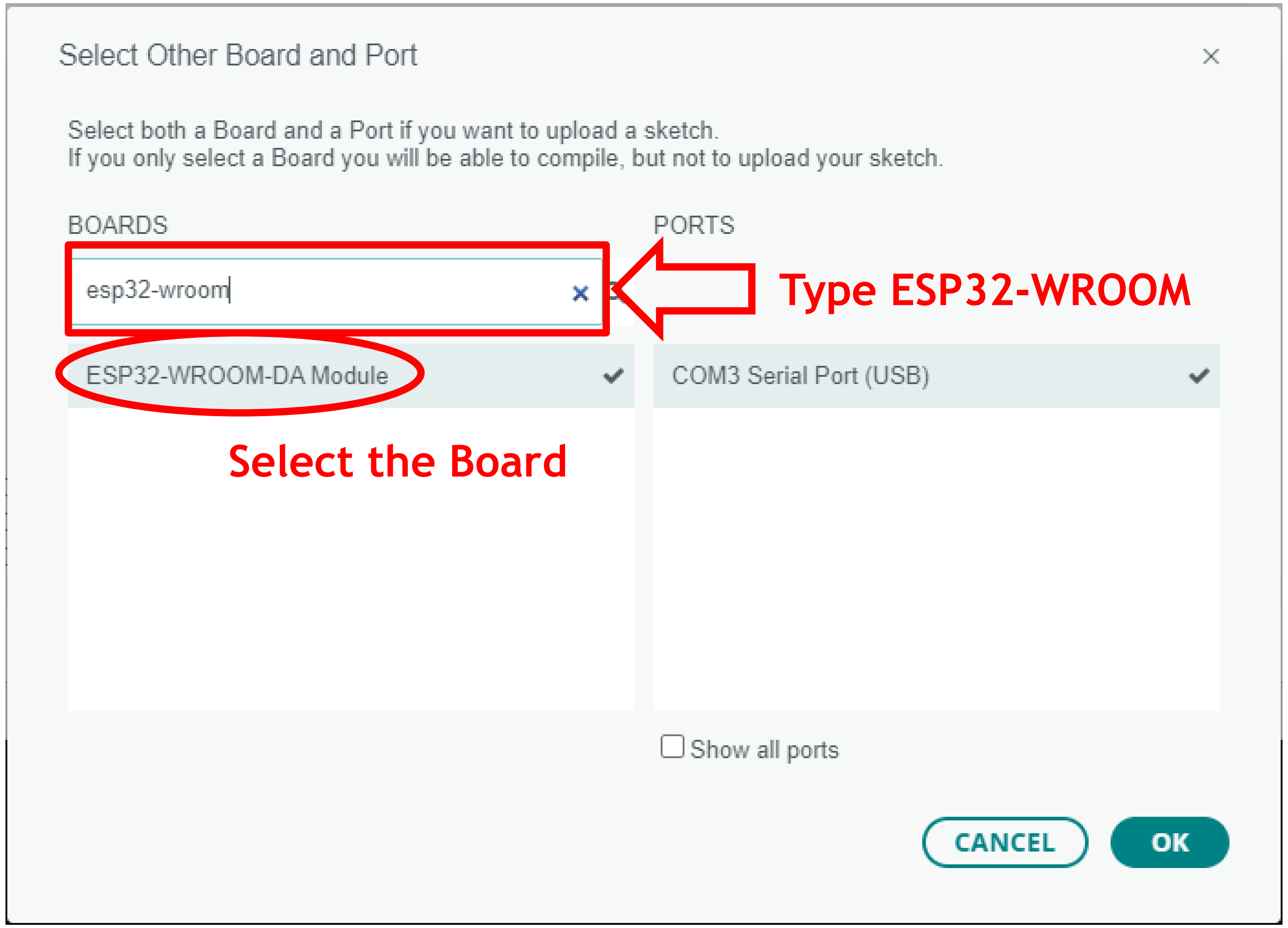
3. Instaling Dependencies

2



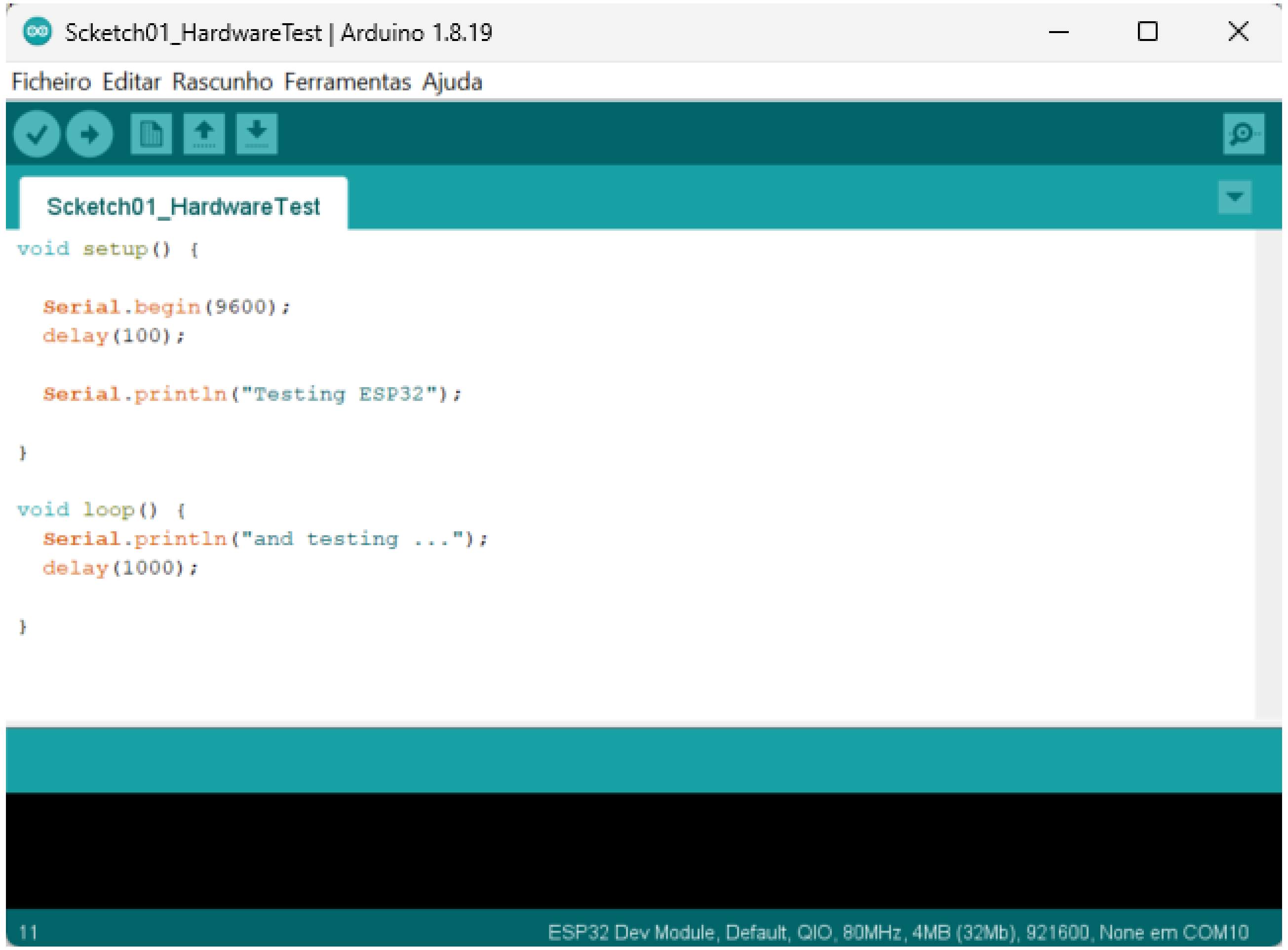
3. Instaling Dependencies

3



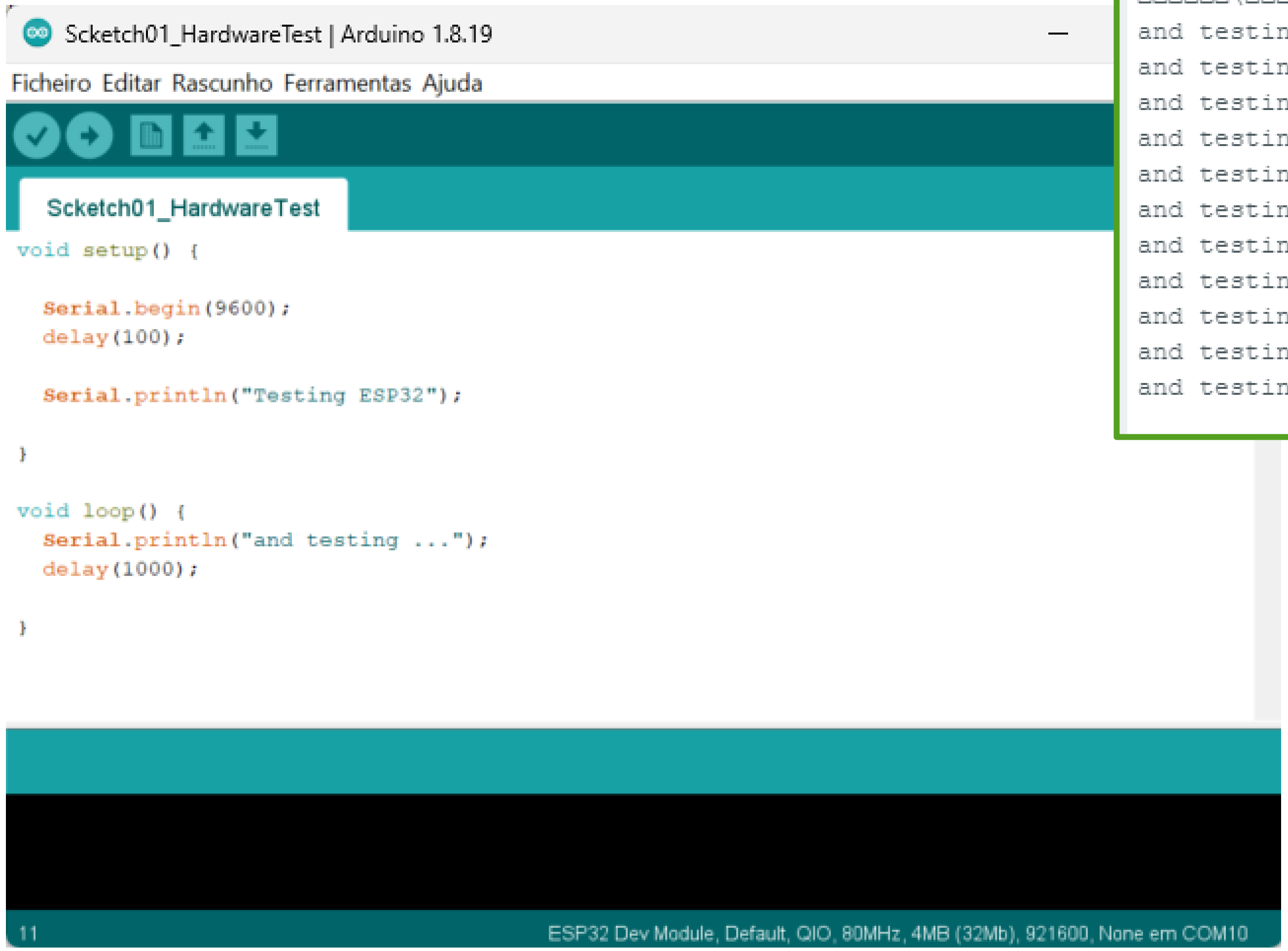
4. Testing the Hardware

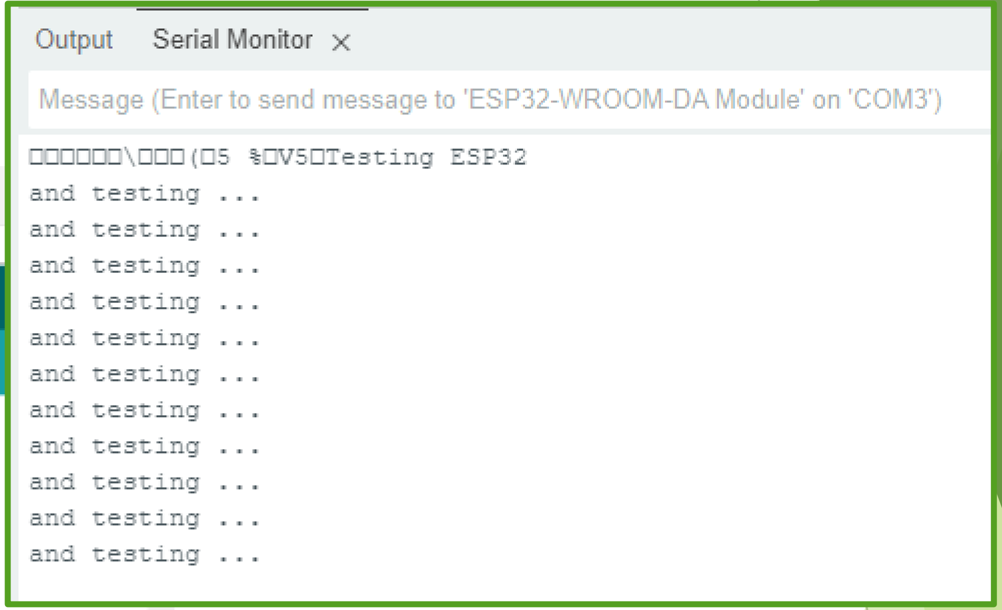
1



4. Testing the Hardware

1





4. Testing the Hardware

2

Upload the project
Scketch01_SensorTest

Output Serial Monitor ×

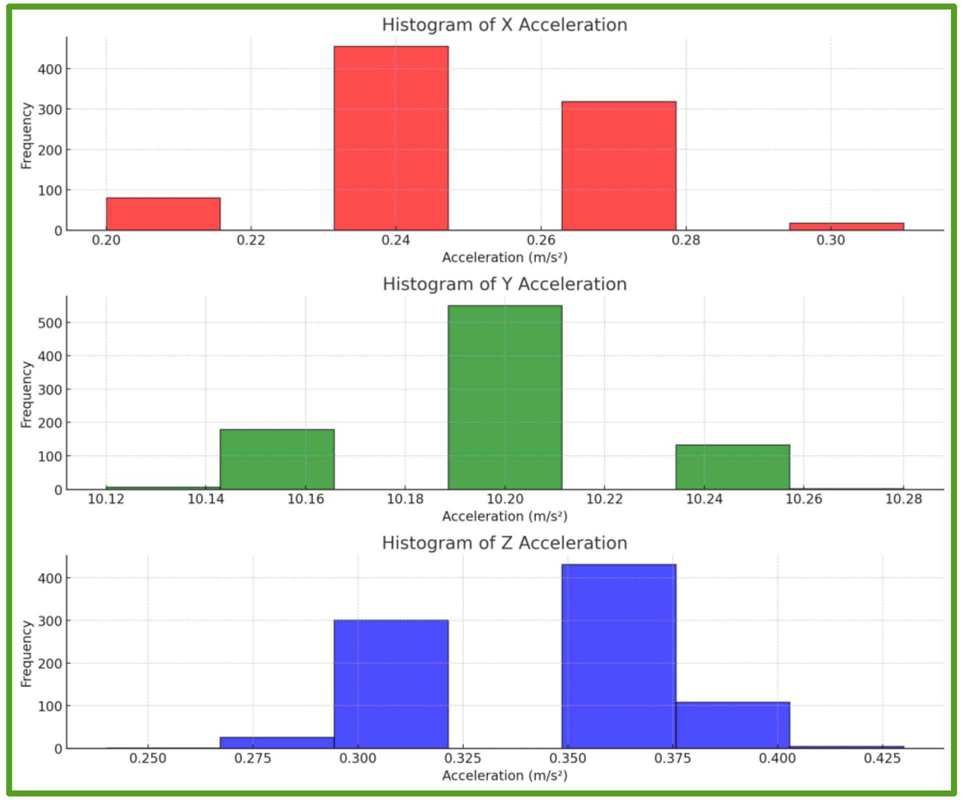
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM3')

```
X: 0.08  Y: -0.67  Z: 9.85  m/s^2
X: 0.08  Y: -0.67  Z: 9.81  m/s^2
X: 0.12  Y: -0.67  Z: 9.77  m/s^2
X: 0.08  Y: -0.67  Z: 9.89  m/s^2
X: 0.08  Y: -0.67  Z: 9.85  m/s^2
X: 0.08  Y: -0.71  Z: 9.85  m/s^2
X: 0.12  Y: -0.67  Z: 9.81  m/s^2
X: 0.12  Y: -0.67  Z: 9.81  m/s^2
X: 0.08  Y: -0.67  Z: 9.81  m/s^2
X: 0.08  Y: -0.67  Z: 9.81  m/s^2
X: 0.08  Y: -0.67  Z: 9.85  m/s^2
X: 0.16  Y: -0.71  Z: 9.85  m/s^2
X: 0.12  Y: -0.63  Z: 9.89  m/s^2
X: 0.12  Y: -0.67  Z: 9.81  m/s^2
X: 0.08  Y: -0.67  Z: 9.92  m/s^2
```

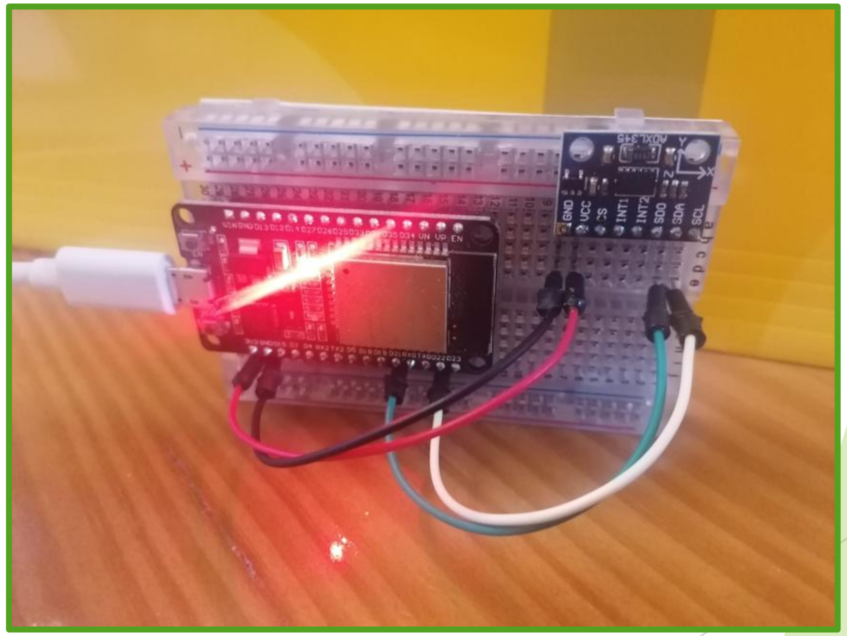
4. Testing the Hardware

3

Let’s see the results



How was the board positioned?



Let’s think about the Software

The Circular Buffer, and The Moving Average Filter

Let’s do it!

5. The Circular Buffer or FIFO (First-In-First-Out)

The buffer size should be a power of 2, less one unit (2^n-1), so that the maximum size is easily detected with one logic operation.

```c
#define BUFFER_SIZE 0xF

typedef struct{
    int data[BUFFER_SIZE];
    int ptr;
} CircularBuffer;


void initializeBuffer(CircularBuffer* buffer){
    buffer->ptr = 0;
}
```

The data structure is to be adapted to the application problem. It can have a timestamp or other data. Since we are considering overlapping, that is, when the buffer is full a new value overlaps the oldest one.

The only major initialization is due to the writing pointer. In case of avoiding a transient state with the first fulfillment of the buffer, the buffer itself could also be initialized. A usual procedure is to fill all the buffer with the first read value.

5. The Circular Buffer or FIFO (First-In-First-Out)

```
void enqueue(CircularBuffer* buffer, int value){

    buffer->data[buffer->ptr] = value;
    buffer->ptr = (buffer->ptr + 1) & BUFFER_SIZE;
}
```

Write the new value in the position that is pointed by the ptr attribute member.

Updates the value of ptr adding one position. Also, checks for the buffer attaining the maximum position value.

00001111 (0x0F)
& 00000011 (Position 3)
00000011

00001111 (0x0F)
& 00010000 (Position 16)
00000000 Reset

TESTING

```
int main()
{
    CircularBuffer buffer;

    for(int i=0; i<20; i++){
        enqueue(&buffer, i);
    }

    return 0;
}

```

Watches

Function arguments	
Locals	
buffer	
data	
[0]	16
[1]	17
[2]	18
[3]	19
[4]	4
[5]	5
[6]	6
[7]	7
[8]	8
[9]	9
[10]	10
[11]	11
[12]	12
[13]	13
[14]	14
ptr	4

5. The Moving Average Filter

```c
void enqueue(CircularBuffer* buffer, int value){

    sum -= buffer->data[buffer->ptr];
    sum += value;

    buffer->data[buffer->ptr] = value;
    buffer->ptr = (buffer->ptr + 1) & BUFFER_SIZE;
}
```

Subtracts the previous value before updating the buffer.

```c
int main()
{
    CircularBuffer buffer;

    for(int i=0; i<20; i++){
        enqueue(&buffer, i);
    }

    float mean = buffer.sum / BUFFER_SIZE;

    return 0;
}
```

To calculate the moving average, the sum is divided by the number of samples, 1

Implement 3 buffer, one for each axis, and 3 moving average filters

Let's do it!

2

- ❑ Notice that when in a stable position, you will only have acceleration on the Z axe (the gravity acceleration).
- ❑ Looking at the other axes, you will be able to understand the plate slope.

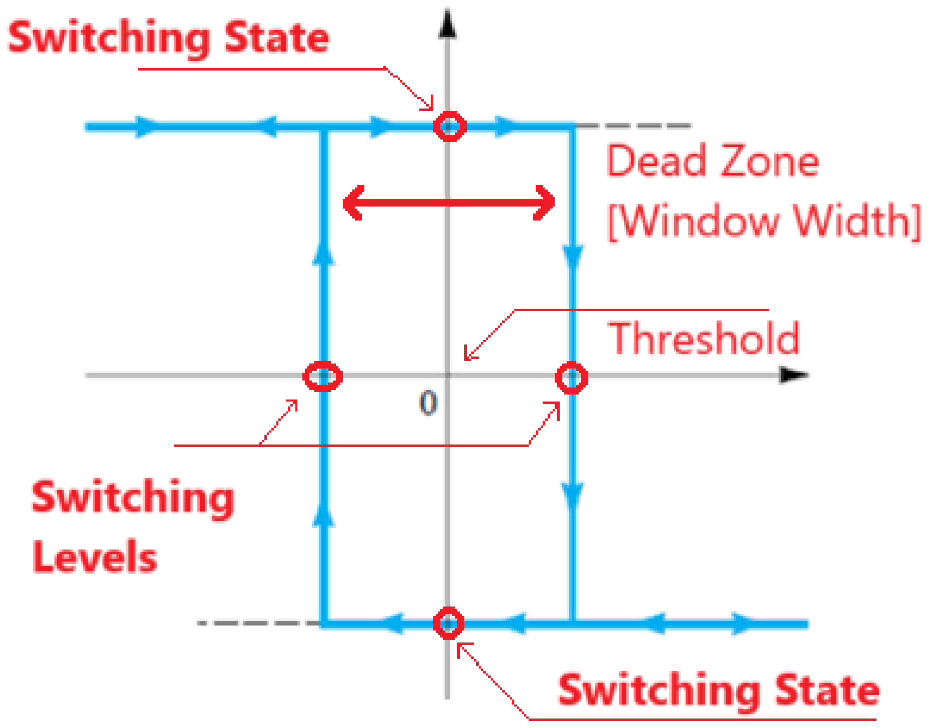
Create a firmware that write on the serial channel that an axis has changed, above a certain value (pre-defined).

What happens when you are on the edge of the threshold value?

How to solve this?

6. Hysteresis

Hysteresis is a phenomenon where the output or state of a system doesn't only depend on the current input, but also on its recent history. Essentially, it introduces a dead zone or buffer zone where small fluctuations in input do not cause frequent state changes.

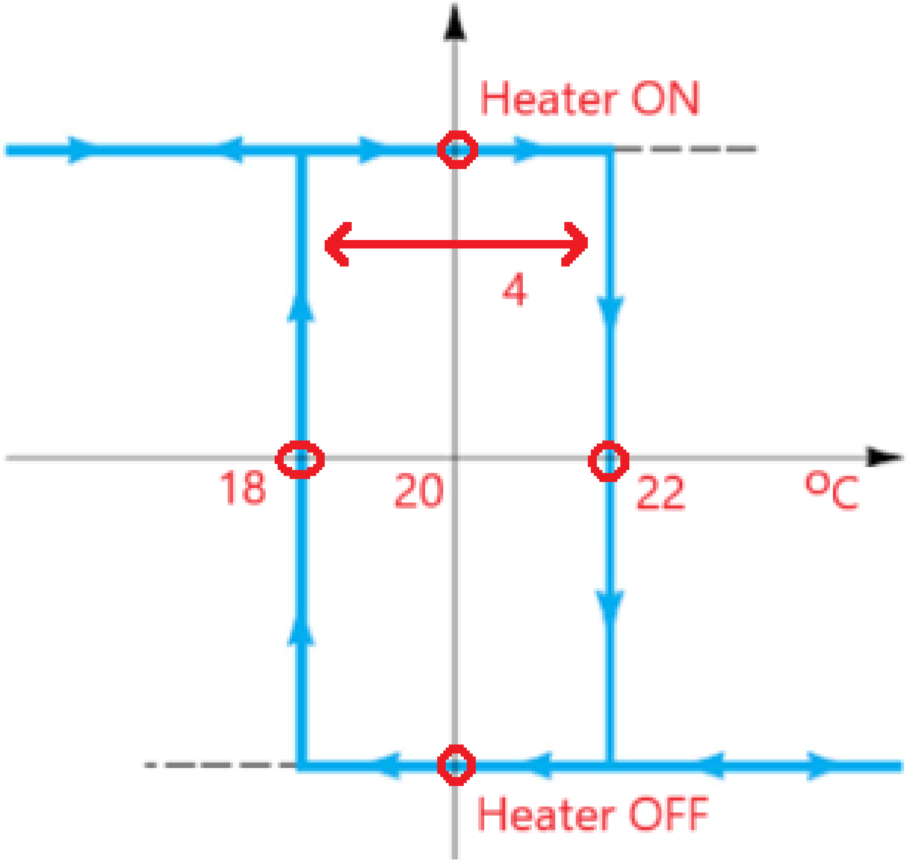


6.1. Hysteresis (Classic example)

Imagine a thermostat:

- Turns ON the heater when temperature drops below 18°C.
- Turns OFF the heater when temperature rises above 22°C.

This prevents the heater from rapidly switching ON/OFF due to small temperature oscillations around 20°C.



6.2. Why use Hysteresis with a Microcontroller reading analog values (e.g., IMU accelerometer)?

Analog sensors, like IMUs (Inertial Measurement Units), often have small fluctuations or noise in their readings, even if the actual physical condition (e.g., acceleration) is stable.

Problem:

If you make decisions (like triggering an event) based on raw accelerometer readings, small variations might cause the system to switch states erratically (false positives/negatives).

6.3. How to apply Hysteresis with IMU Accelerometer data

Scketch03_Histeresis

Scenario

Let's say you want to trigger an event when the X-axis acceleration exceeds a threshold (for example, indicating a sudden movement or tilt), but you don't want small jitters to trigger or reset the event unnecessarily.

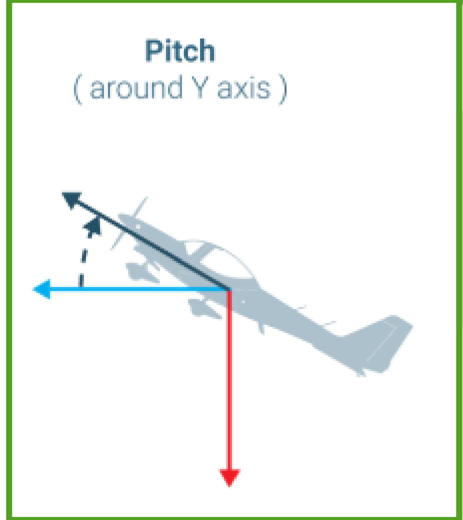
Benefits of Hysteresis in this IMU application

- Stability: Avoids false triggers caused by sensor noise.
- Controlled transitions: No rapid toggling between states.
- Useful for motion detection, tilt sensing, or gesture recognition.

7.1. Euler Angles - What is "Pitch"?

Pitch is the angle of rotation around the Y-axis (tilting forward or backward).
Imagine holding the sensor:

- Pitch = 0° → sensor flat, facing up.
- Positive pitch → front of the sensor tilted upward.
- Negative pitch → front tilted downward.



https://support.pix4d.com/hc/en-us/articles/202558969

$$\text{Pitch} = \arctan 2\left(-a_x, \sqrt{a_y^2 + a_z^2}\right) \times \left(\frac{180}{\pi}\right)$$

Where:

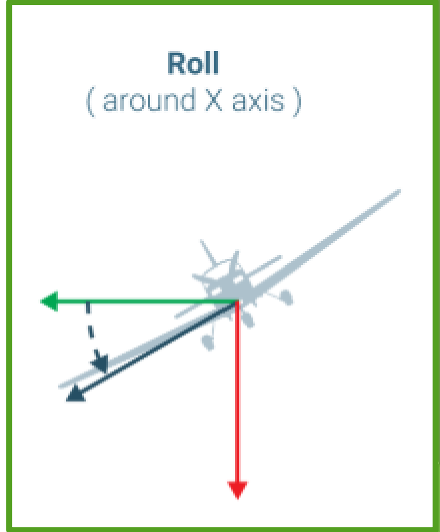
- ax, ay, az = accelerometer readings (after filtering).
- atan2 is used to calculate the correct angle considering the quadrant.
- Result is converted to degrees.

7.2. Euler Angles - What is “Roll"?

Roll is the angle of rotation around the X-axis (tilting left or right).
Imagine holding the ADXL345 sensor:

- Roll = 0° → Sensor flat, level.
- Positive roll → Right side of the sensor tilted upward (like turning a steering wheel clockwise).
- Negative roll → Left side tilted upward (counterclockwise).

$$\mathrm{Roll} = \arctan 2(a_y, a_z) \times \left(\frac{180}{\pi}\right)$$



https://support.pix4d.com/hc/en-us/articles/202558969

Where:

- ay, az = accelerometer readings (after filtering).
- atan2 is used to calculate the correct angle considering the quadrant.
- Result is converted to degrees.

7.4. Calculate Euler Angles from Acceleration

Scketch04_Euler

$$\text{Roll} = \arctan 2(a_y, a_z) \times \left(\frac{180}{\pi}\right)$$

$$\text{Pitch} = \arctan 2\left(-a_x, \sqrt{a_y^2 + a_z^2}\right) \times \left(\frac{180}{\pi}\right)$$

Where

ax, ay, az = filtered acceleration values in m/s^2

Note

- Output in degrees for easy interpretation
- Pitch/Roll assume the device is stationary or moving slowly, as dynamic accelerations can distort the readings

7.4. Calculate Euler Angles from Acceleration

Scketch04_Euler

$$\text{Roll} = \arctan 2(a_y, a_z) \times \left(\frac{180}{\pi}\right)$$

$$\text{Pitch} = \arctan 2(-a_x, \sqrt{a_y^2 + a_z^2}) \times \left(\frac{180}{\pi}\right)$$

Where

ax, ay, az = filtered acceleration values in m/s²

Note

- Output in degrees for easy interpretation
- Pitch/Roll assume the device is stationary or moving slowly, as dynamic accelerations can distort the readings

Your Challenge Now!

Create a Firmware that sends filtered Euler Angles (pitch and roll) over Bluetooth to your Mobile Phone

Let’s put it to work. Good luck!

Serial Bluetooth Terminal