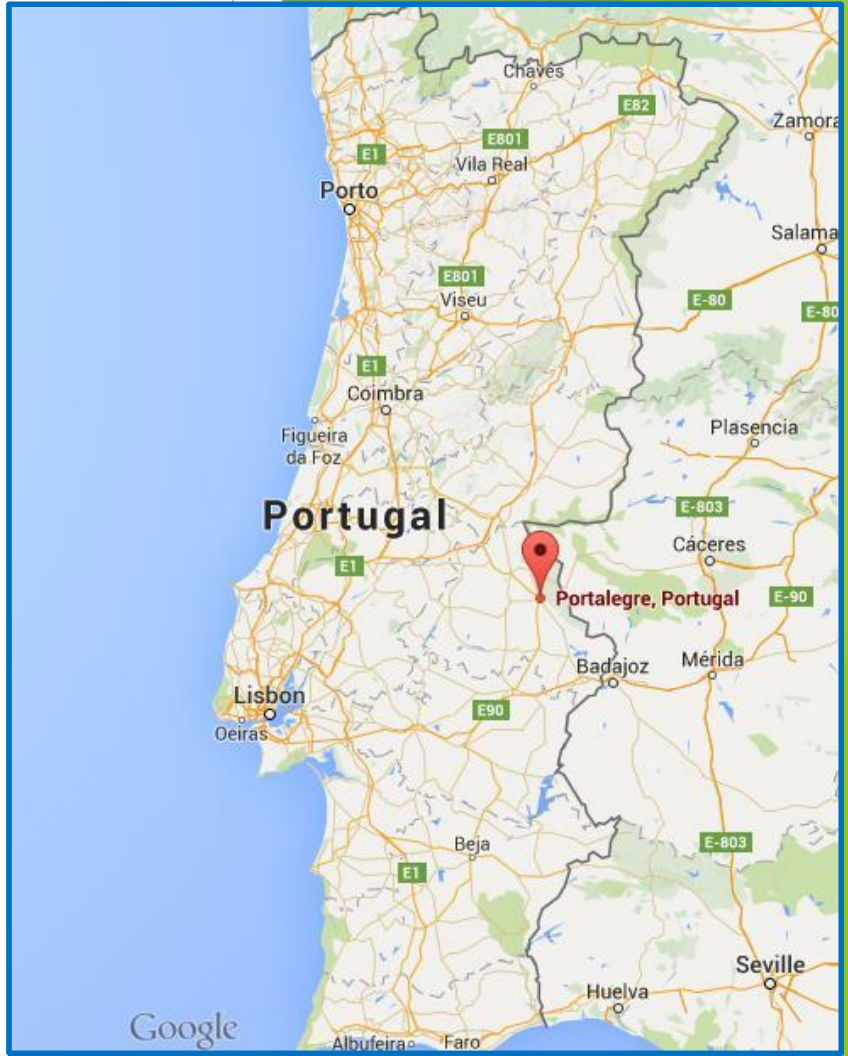
Prof. Sérgio D. Correia
Portalegre Polytechnic University
Portugal



Talk

Inertial Measurement Units (IMUs) Data Acquisition and Filtering

“Tangible interfaces for VR applications”
International Week 2025

Let’s think about the Hardware

The IMU and the ESP32

What is it? How to get it to work?

1. Components of an IMU

Accelerometer

Measures linear acceleration along three axes. Detects changes in velocity and orientation relative to gravity.

Gyroscope

Measures angular velocity. Tracks rotational movements around each axis.

Magnetometer

Measures magnetic field strength. Optional component that provides compass functionality.

3 DOF → Accelerometer

6 DOF → Accelerometer + Gyroscope

9 DOF → Accelerometer + Gyroscope + Magnetometer

More DoF = Better motion tracking and orientation accuracy

1. Components of an IMU

Acceleration (Linear Acceleration)

Definition: Acceleration is the rate of change of velocity over time. It measures how quickly an object speeds up or slows down.

Measured in: m/s² (meters per second squared).

IMU Sensor Used: Accelerometer.

Axes: Typically measured along X, Y, and Z axes.

Example:

A car speeding up from 0 to 60 km/h experiences acceleration.

A person jumping experiences acceleration due to gravity (≈9.81 m/s² on Earth).

1. Components of an IMU

Angular Velocity

Definition: Angular velocity is the rate at which an object rotates around an axis.

Measured in: rad/s (radians per second) or °/s (degrees per second).

IMU Sensor Used: Gyroscope.

Axes: Typically measured along X, Y, and Z (roll, pitch, and yaw).

Example:

A spinning top has angular velocity.

A drone tilting forward has angular velocity in the pitch axis.

1. Components of an IMU

Magnetic Field

Definition: The magnetic field is the force field around magnetic objects or electric currents. It helps determine an object's orientation relative to Earth's magnetic poles.

Measured in: µT (microtesla) or Gauss.

IMU Sensor Used: Magnetometer.

Example:

A compass aligns with Earth's magnetic field.

Smartphones use a magnetometer for navigation apps.

2. Common IMU Degrees of Freedom (DoF)

The degree of freedom (DoF) of an IMU refers to the number of independent motion parameters it can measure. IMUs typically come in different DoF configurations, depending on the number of sensors included:

3-DoF IMU

Contains only accelerometers or only gyroscopes.

Can measure linear acceleration or angular velocity in three axes (X, Y, Z).

6-DoF IMU

Contains 3 accelerometers + 3 gyroscopes.

Measures both linear acceleration and angular velocity in all three axes.

2. Common IMU Degrees of Freedom (DoF)

The degree of freedom (DoF) of an IMU refers to the number of independent motion parameters it can measure. IMUs typically come in different DoF configurations, depending on the number of sensors included:

9-DoF IMU

Contains 3 accelerometers + 3 gyroscopes + 3 magnetometers.

Adds magnetometer data for absolute orientation using Earth's magnetic field.

Helps with heading correction to reduce gyroscope drift.

10-DoF or Higher IMU

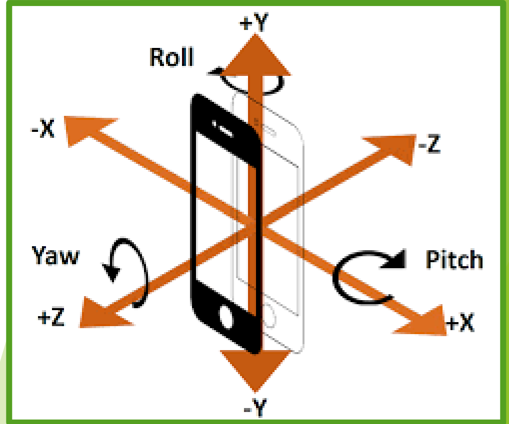
Sometimes includes a barometer (for altitude measurement).

Can improve positioning accuracy, especially in drones and navigation systems.

2. Common IMU Degrees of Freedom (DoF)

Summary Table

Measurement	Definition	Sensor in IMU	Units
Acceleration (Linear)	Rate of velocity change (movement in X, Y, Z directions)	Accelerometer	m/s^2
Angular Velocity	Rotation speed around an axis (roll, pitch, yaw)	Gyroscope	rad/s or °/s
Magnetic Field	Strength and direction of Earth's magnetism	Magnetometer	µT or Gauss



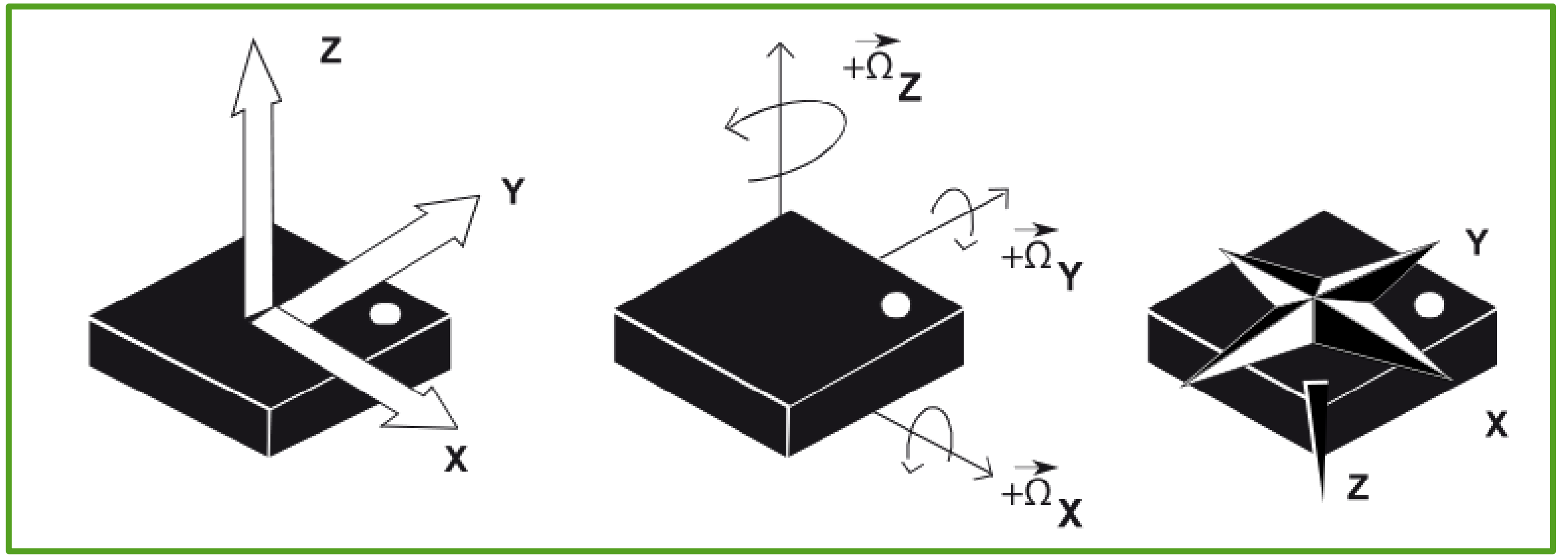
https://blogs.sas.com/content/sgf/2018/09/26/accelerometer-driving-profile/

3. Where can we find IMUs?

- **Smartphones & Tablets** – For screen orientation, step counting, and motion tracking.
- **Drones & UAVs** – For stabilization and navigation.
- **Robotics** – To assist in motion control and self-balancing.
- **Automobiles** – In vehicle airbags, anti-lock braking systems (ABS), and autonomous driving.
- **Gaming & VR** – Motion tracking in controllers and headsets.
- **Wearable Devices** – Fitness trackers and smartwatches use IMUs for step counting and activity tracking.
- **Aerospace & Aviation** – Used in aircraft and spacecraft for navigation and control.
- **Industrial Applications** – Robotics, machinery, and automation systems use IMUs for motion analysis.

4. The Hardware and What will we use?

The IMU will have the form of an integrated circuit, typically Surface Mount Technology (SMT)



In: https://docs.longan-labs.cc/1011021/

ADXL345

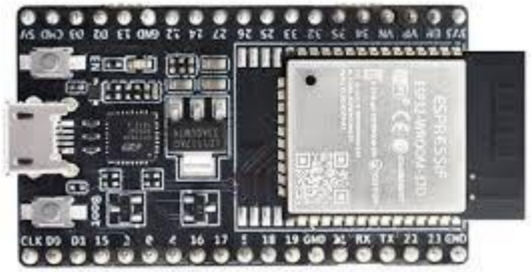
5. Popular Embedded Modules

ARDUINO UNO

ARDUINO MEGA 2560

ARDUINO DUE

NODEMCU ESP8266



ESP32

ESP32-CAM

5. Popular Embedded Modules

Specifications	NodeMCU (ESP8266) 	Arduino DUE 	ESP32 
Microcontroller	ESP8266	AT91SAM3X8E	ESP32-WROOM-32
CPU Core	Tensilica Xtensa LX106	ARM Cortex-M3	Tensilica Xtensa LX6
Clock Speed	80 MHz	84 MHz	160 MHz
Flash Memory	128 KB	512 KB	4 MB
SRAM	4MB	96 KB	520 KB
Digital I/O Pins	16	54	32
Analog Input Pins	1x 10-bit ADC	12x 12-bit ADC	18x 12-bit ADC
Analog Output Pins	-	2x 12-bit DAC	2x 8-bit DAC
Connectivity	IEEE 802.11 b/g/n FTDI USB UART	Native USB FTDI USB UART	802.11 b/g/n Bluetooth v4.2 BR/EDR and BLE FTDI USB UART

6. Popular IMU Modules

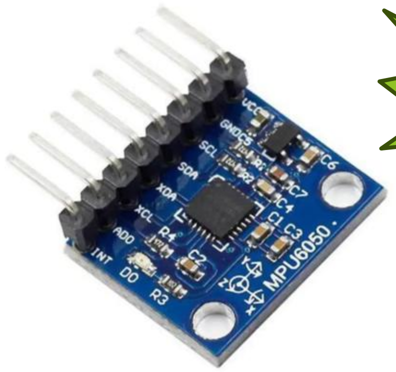
ADXL345

3DOF

The ADXL345 is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 g. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2C digital interface.

MPU6050

6DOF



The MPU6050 is a 6 DOF IMU used to read acceleration and angular velocity in all three dimensions. The MPU6050 object represents a connection to the device on the Arduino hardware I2C bus.

LSM9DS1

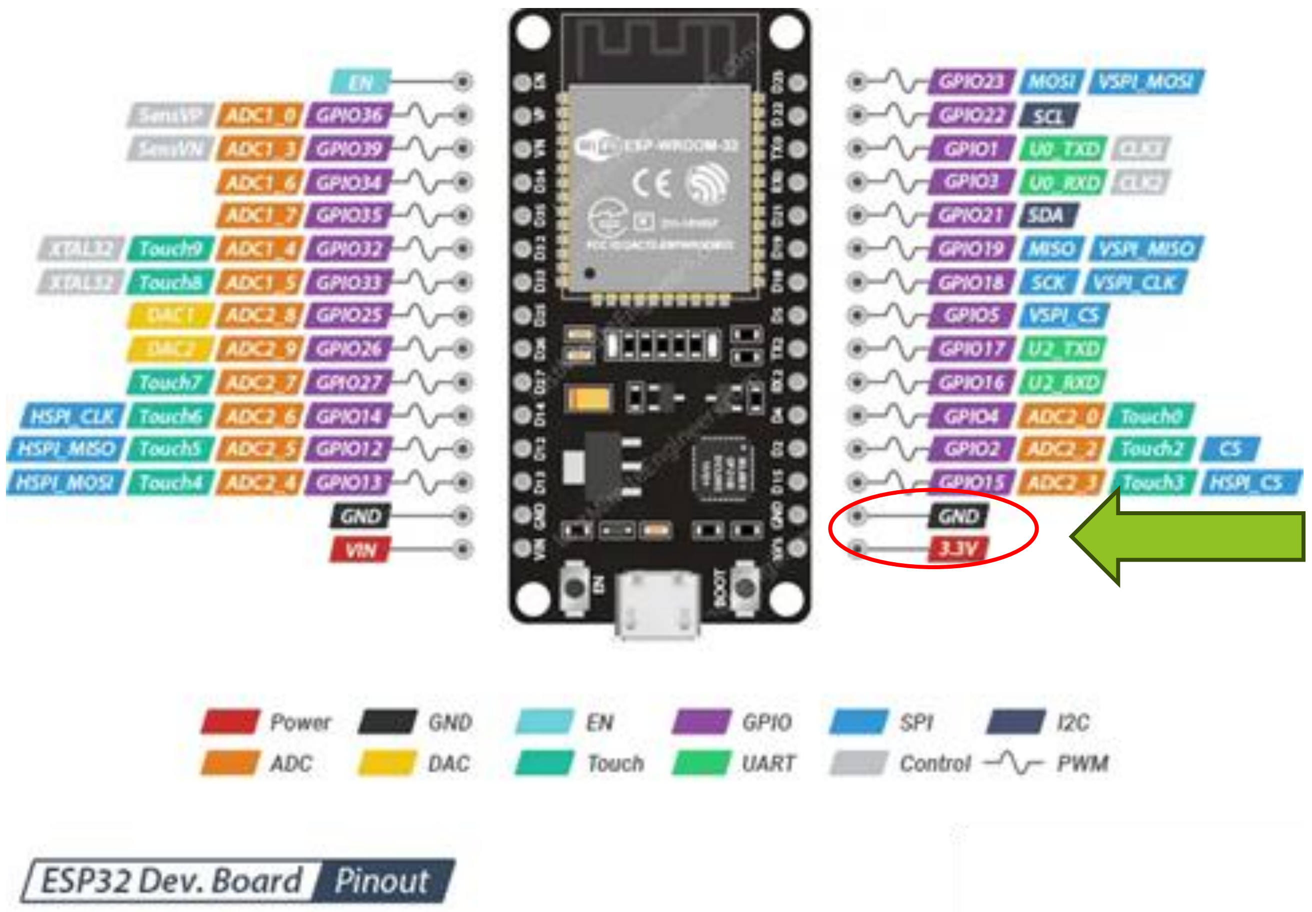
9DOF

Arduino Nano BLE Sense

The LSM9DS1 is a 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor. The LSM9DS1 has a linear acceleration full scale of ±2g/±4g/±8/±16 g, a magnetic field full scale of ±4/±8/±12/±16 gauss and an angular rate of ±245/±500/±2000 dps.

Using more DOF implies applying sensor fusion techniques, thus, more complex algorithms for decision making

7. Connecting an IMU to an ESP32



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

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

Used to power the IMU module

7. Connecting an IMU to an ESP32

ARDUINO

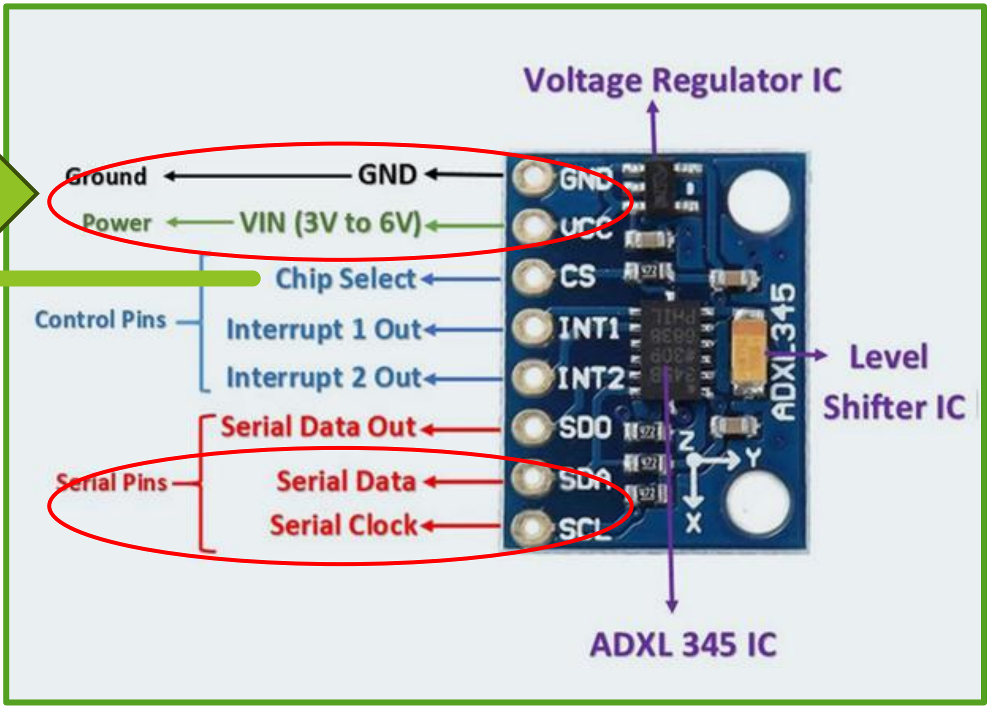
Power supply

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

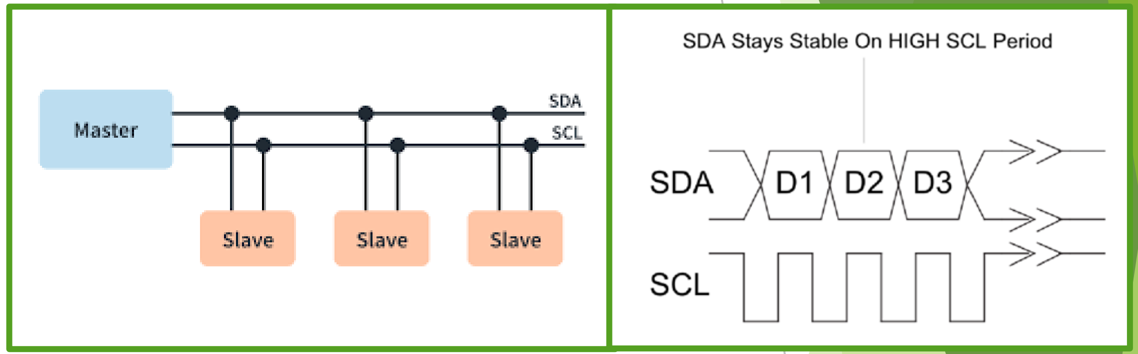
SDO (Serial Data Out): Serial Data Output (SPI 4-Wire)/Alternate I2C Address Select (I2C).

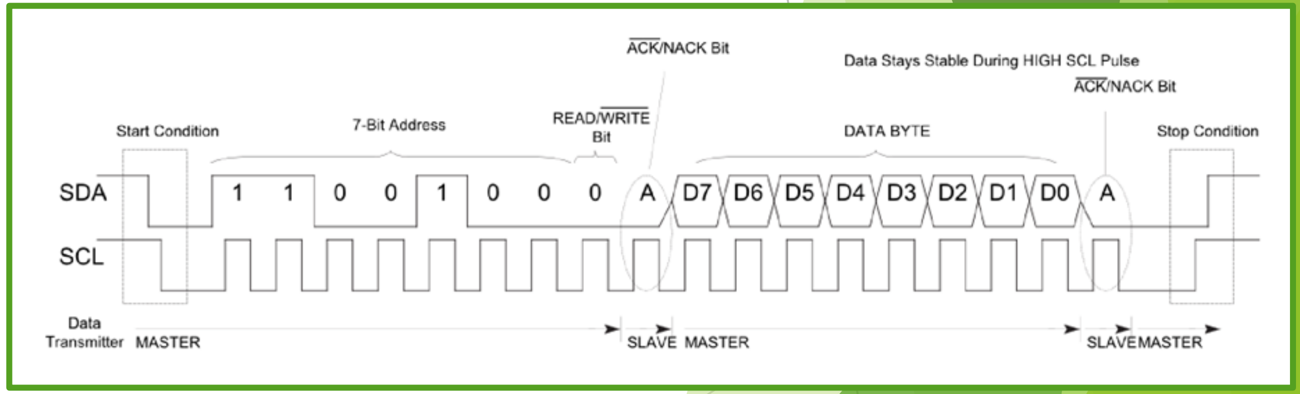
SDA (Serial Data): Serial Data (I2C)/Serial Data Input (SPI 4-Wire)/Serial Data Input and Output (SPI 3-Wire).

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

8. What is an I²C Bus?

I2C (Inter-Integrated Circuit) BUS



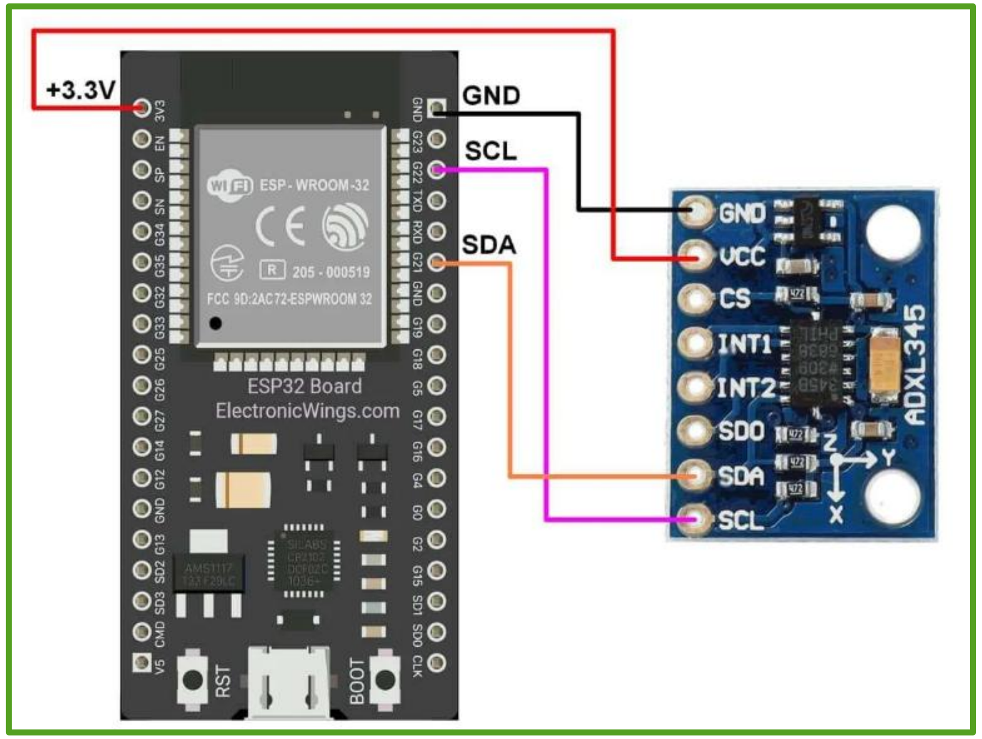


https://www.analog.com/en/resources/technical-articles/i2c-primer-what-is-i2c-part-1.html

I²C (Inter-Integrated Circuit) is a communication protocol used to connect multiple devices using just two wires:

- SDA (Serial Data) – Transfers data between devices.
- SCL (Serial Clock) – Synchronizes data transmission.

9. Physical Connections



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

10. 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

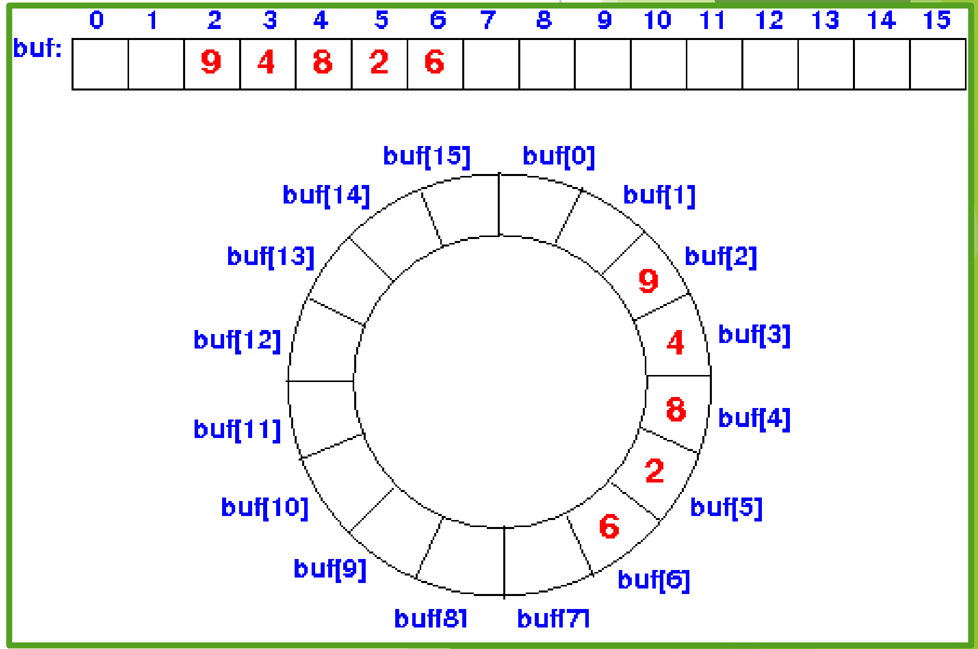
The Circular Buffer, and The Moving Average Filter

What is it? How to get it to work?

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

A data structure that employs a single, fixed-size buffer as if it were connected end to end is known as a circular buffer, circular queue, cyclic buffer, or ring buffer in computer science. Data streams can be readily buffered using this structure.

The buffer is implemented with an array of the data structure, its length, and one or two pointers, depending on the possibility of overlapping.



11. 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.

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

```c
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

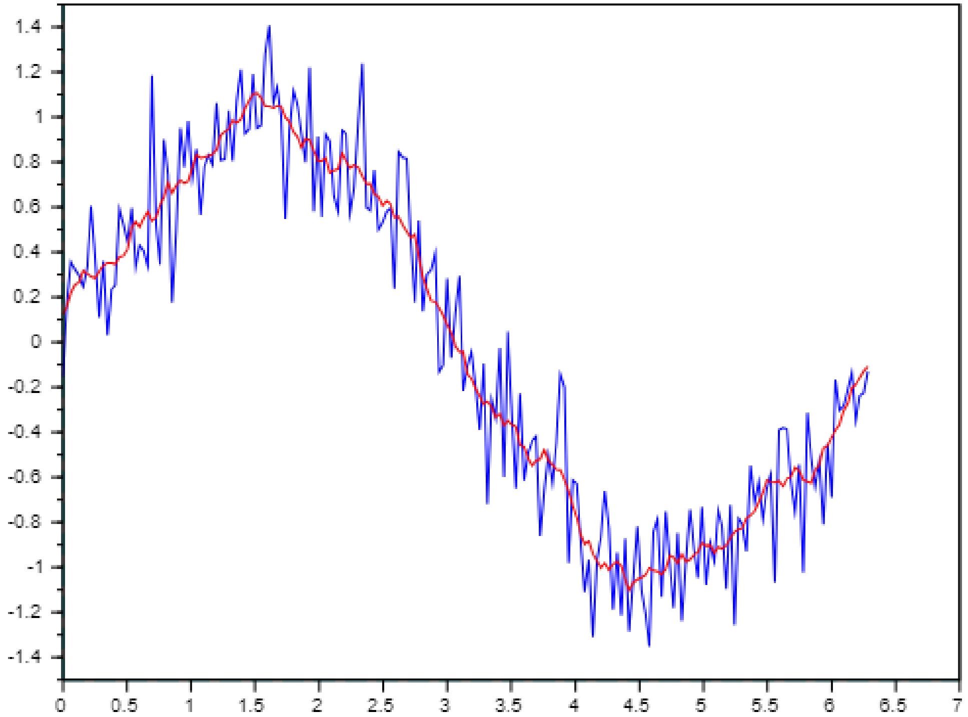
```c
int main()
{
    CircularBuffer buffer;

    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

12. The Moving Average Filter

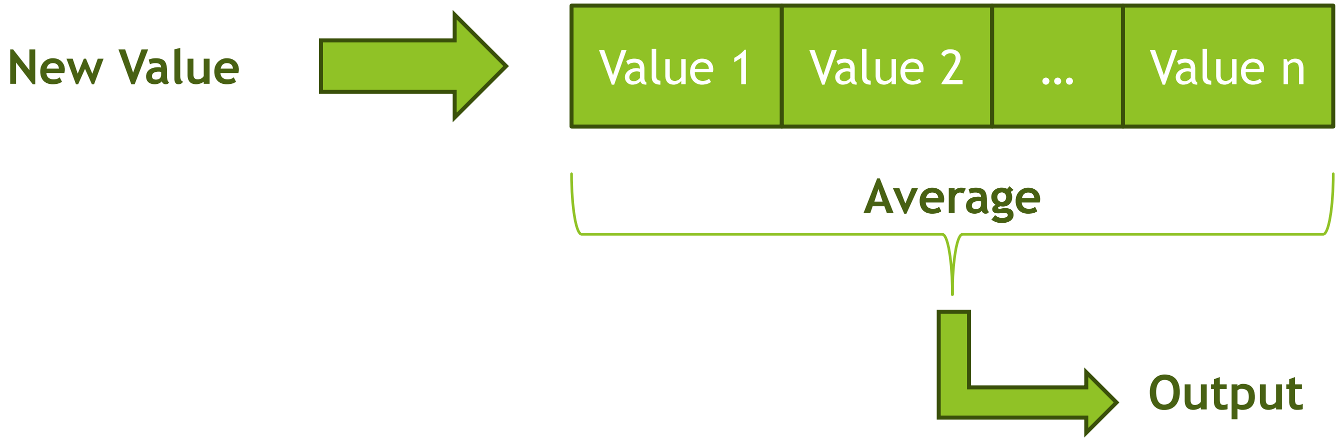
A moving average, also known as a rolling average or running average, is a statistical computation that is used to examine data points by averaging several successive choices of the entire data set. Another name for it is a rolling or moving mean.



When dealing with time series data, a moving average is frequently used to highlight longer-term trends or cycles and smooth out short-term volatility. The moving average's parameters will be adjusted based on the application, which determines the threshold between short- and long-term.

12. The Moving Average Filter

This kind of filter keeps several samples in an overlapping first-in, first-out (or circular) buffer. Every time the filter is executed, a new value from the input is added to the buffer, and the oldest value is removed. Subsequently, the filter computes the mean of all the recorded values, which subsequently serves as the updated filter output.



12. The Moving Average Filter

The simple moving average (SMA) can be calculated by summing all values and dividing by the number of samples (or windows). In a weighted moving average, an average has multiplying factors to give different weights to data at different positions in the sample window. More importance can be given to newer providing a low ranking to older samples.

$$SMA = \frac{p_{n-k+1} + p_{n-k+2} + \cdots + p_n}{k} = \frac{1}{k} \sum_{i=n-k+2}^{n+1} p_i$$

12. The Moving Average Filter

When calculating the next mean $SMA_{k,next}$ with the same sampling width k, the range from $n-k+2$ to $n+1$ is considered. A new value p_{n+1} comes into the sum and the oldest value p_{n-k+1} drops out. This simplifies the calculations by reusing the previous mean $SMA_{k,prev}$.

$$
\begin{aligned}
SMA_{k,next} &= \frac{1}{k}\sum_{i=n-k+2}^{n+1} p_i \\
&= \frac{1}{k}(p_{n-k+2} + p_{n-k+3} + \cdots + p_n + p_{n+1} + \underbrace{p_{n-k+1} - p_{n-k+1}}_{=\,0}) \\
&= \frac{1}{k}(p_{n-k+1} + p_{n-k+2} + \cdots + p_n) - \frac{p_{n-k+1}}{k} + \frac{p_{n+1}}{k} \\
&= SMA_{k,prev} + \frac{1}{k}(p_{n+1} - p_{m-k+1})
\end{aligned}
$$

This means that the moving average filter can be computed cheaply on real time data with a FIFO / circular buffer and only 3 arithmetic steps.

12. The Moving Average Filter

```
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.

```
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,