

# Node: ESP32\_IMU\_Node

## Header

Team: Groep 4 RMB

Created: 04.11.2025

name	date	Short description
ESP32_IMU_Node (Wired)	30.10.2025	Micro ROS2 node that receives data from the IMU and publishes it on /imu_data_esp. The publisher published sensor_msgs::msg::Imu messages on the imu_data_esp topic.
ESP32_IMU_Node (Wireless)	04.11.2025	Program that connects to a computer with a UDP server over Wi-Fi, and sends IMU sensor data to it.

## Node description

### Wired

This Micro\_ROS node works like a bridge between the computer and the IMU sensor. The sensor publishes raw data to the ESP32, the ESP32 filters the necessary data out of it and then puts all the data into the IMU msg. And then it publishes the data to the /imu\_data\_esp topic so the computer is able to reach it.

ESP32 > /imu\_data\_esp > IMULifecycleNode > /imu\_data > Database Subscriber

### Wireless

This node establishes a UDP socket server that listens for incoming IMU data packets on a configurable port.

Each packet is expected to contain six comma-separated floating-point values representing angular velocity and linear acceleration.

The node verifies the data for validity, discards zeroed packets, and publishes the readings as ROS2 IMU messages with proper timestamping and frame identification.

ESP32 > (UDP) > IMULifecycleNode > /imu\_data > Database Subscriber

## Node sub-objects and functions (communication objects)

### Wired

<code>publisher : /imu_data_esp32</code> Published by <code>esp32_imu_node</code> Purpose: publishes IMU data received from the sensor on the topic.
Publisher function: <code>rcl_publish(&amp;publisher, &amp;imu_msg, NULL);</code> Publishes the received IMU data on the topic ( <code>/imu_data_esp32</code> ).

### Wireless

<code>sender : UDP port 5005</code> sent by <code>IMUsensor_wireless (ESP32)</code> Purpose: sends IMU data received from the IMU sensor to the computer.
sender functions: <code>udp.beginPacket(host_ip, host_port);</code> <code>udp.write((uint8_t*)buffer, strlen(buffer)); // &lt;-- cast to uint8_t*</code> <code>udp.endPacket();</code> These 3 functions are used to send data over UDP

## Node actions and messages

Messages:

- (Wired) Published Message Type: `sensor_msgs::msg::Imu`
- (Wireless) Sent a UDP message: `char buffer[128]`

Topics:

- `/imu_data_esp`: Raw IMU data from ESP32.

## Custom node functions

### Wired

Void `setup()`

Configures micro ros node, initializes publisher and configures the IMU sensor.

Void `loop()`

Gets data from the IMU sensor and puts it into the `IMU_msg`, and then publishes the data on the topic.

## Wireless

Void setup()

Connects to Wi-Fi with port 5005 and configures the IMU sensor.

Void loop()

Gets data from the IMU sensor and puts it into the buffer to get sent on the UDP server.

## Node dependencies

### Wired

Micro\_ROS\_arduino: Includes dependencies for micro\_ROS on Arduino IDE.

roscpp: Core Micro\_ROS C++ client library.

Adafruit\_BNO055: Library for the IMU sensor.

sensor\_msgs: Defines sensor\_msgs::msg::Imu for IMU data communication.

### Wireless

Adafruit\_BNO055: Library for the IMU sensor.

WiFiUdp: Library for UDP connection over Wi-Fi.