Node: Lifecycle node and subscriber

Header

Team: Groep 4 RMB

Created: 04.11.2025

name	date	Short description
LifecycleNodeSubscriber	04.11.2025	ROS2 node that subscribes to IMU messages published on /imu_data, logs them to the console, and stores the measurements in a MariaDB database for persistent recording and later analysis.
LifecycleManager	30.10.2025	ROS2 node that manages the lifecycle state of the IMU lifecycle node. It monitors incoming IMU data on /imu_data_esp and automatically transitions the target imu_lifecycle_node between lifecycle states (configure, activate, deactivate) depending on data availability and connection status.
IMULifecycleNode	30.10.2025	Lifecycle-enabled ROS2 node that subscribes to IMU data from an ESP32 device on /imu_data_esp or connects to a socket UDP connection from the ESP32 and republishes it to /imu_data for downstream nodes such as database loggers. The node’s behavior depends on its lifecycle state, allowing the lifecycle manager to control when it is active or inactive.

Node description

Combined subscriber system for IMU data acquisition, lifecycle management, and database storage. This structure includes both lifecycle control and standard subscriber functionality — receiving IMU data from an ESP32, processing it through a lifecycle-managed node, and finally storing it in a MariaDB database.

The subscriber system consists of three nodes:

1. IMULifecycleNode – a lifecycle-controlled node that manages IMU data from the ESP32 wired (/imu_data_esp) or wireless (UDP) and republishes it to /imu_data.
2. LifecycleNodeSubscriber – a subscriber node that listens to /imu_data and stores measurements in a MariaDB database.
3. LifecycleManager – an external lifecycle controller that triggers transitions (configure, activate, deactivate) for the lifecycle node.

Wired:
ESP32 > /imu_data_esp > IMULifecycleNode > /imu_data > Database Subscriber

Wireless:

ESP32 > (UDP) > IMULifecycleNode > /imu_data > Database Subscriber

Node sub-objects and functions (communication objects)

publisher : /imu_data IMULifecycleNode Provides clean IMU data for downstream nodes (e.g., database subscribers).
Publisher function: esp32Callback() Publishes the received IMU message if node is active.

subscriber : imu_data_esp Subscribed by IMULifecycleNode. Purpose: Receives IMU data published by the ESP32 device.
Bind function: esp32Callback() Logs and republishes the message to /imu_data if the node is in the active lifecycle state.

Wall_timer: timer_ Started by IMULifecycleNode. Purpose: Periodically polls the UDP socket for new IMU data based on the parameter timer_period_ms.
bind function: receive_data()
Called at each timer tick to check for incoming UDP packets and handle data parsing/publishing.

Connection : port 5005
Connected by IMULifecycleNode.
Purpose: Connects to port 5005 to wireless receive data over UDP.

Bind function: connect_socket()
Connects to port 5005 and checks if connection is succeeded, when the node is in the configuring state.

receiver : UDP port 5005
Received by IMULifecycleNode.
Purpose: Receives IMU data published by the ESP32 device.

Receiver function: receive_data()
Receives the data if the node is in the active lifecycle state.

subscriber : /imu_data
Subscribed by LifecycleNodeSubscriber.
Purpose: Receives processed IMU messages and stores them in the MariaDB database.

Bind function: imuCallback()
Converts ROS2 timestamps to standard C++ time points, extracts linear acceleration and angular velocity, and inserts the data into the database.

Node actions, messages and services

Subscribed Message Type: sensor_msgs::msg::Imu

Received Message Type: char buffer[256]

Published Message Type: sensor_msgs::msg::Imu

Topics:

/imu_data_esp: Raw IMU data from ESP32.

/imu_data: Processed IMU data for storage.

Lifecycle services (for IMULifecycleNode):

/imu_lifecycle_node/get_state

/imu_lifecycle_node/change_state

Managed by lifecycle manager to control activation and configuration.

Custom Node functions

IMULifecycleNode::on_configure()

Configures lifecycle node, initializes publisher and subscriber and connects to socket.

IMULifecycleNode::on_activate() / on_deactivate()

Activates or deactivates the node’s publisher.

IMULifecycleNode::connect_socket()

Connects to the socket

IMULifecycleNode::receive_data()

Polls the UDP socket for available packets, parses comma-separated IMU data, validates it, and calls esp32Callback() if valid.

IMULifecycleNode::is_all_zero()

Returns true if all IMU values are below the configured tolerance_. Used to filter out zero or invalid readings.

IMULifecycleNode::esp32Callback()

Receives and republishes IMU messages when active.

LifecycleNodeSubscriber::imuCallback()

Parses and logs IMU data, converts timestamps, and inserts data into MariaDB using ImuDatabase::addMeasurement().

Node implementation (main)

IMULifecycleNode main:

Uses a SingleThreadedExecutor to manage lifecycle transitions and topic callbacks.

LifecycleNodeSubscriber main:

Implements standard single-threaded execution for data reception and database storage.

Difference to standard implementation:

Combines a standard ROS2 subscriber with a lifecycle-controlled data relay node.

The lifecycle pattern allows controlled startup and shutdown of IMU data flow, enabling robust synchronization with other system nodes or external managers.

Node dependencies

rclcpp: Core ROS2 C++ client library.

rclcpp_lifecycle: Provides lifecycle management (states, transitions, lifecycle publishers).

sensor_msgs: Defines sensor_msgs::msg::Imu for IMU data communication.

lifecycle_msgs: Provides lifecycle state/transition message types.

MariaDB / ImuDatabase: Custom C++ wrapper for database interaction (ImuDatabase.hpp).

Note:
The database subscriber node requires MariaDB and custom headers to establish database connections (ImuDatabase).

The lifecycle node depends on a lifecycle manager to be fully functional (for state transitions).