Arduino Set-up and the first micro-ros code

Overview: We need to now upload our code to ESP32 for publishing or subscribing to any message or data. We need to have the following three things installed in our system for ESP part.

- Install Arduino IDE from https://www.arduino.cc/en/software/
- Install ESP32 board in Arduino IDE
- Install Micro-ros Arduino library for your ROS version from https://github.com/micro-ROS/micro_ros_arduino

First ESP32-Micro-ROS program using WiFi

Include the following libraries including micro ros, ros client, int32 msg and so on.

```
#include <micro_ros_arduino.h>
#include <Arduino.h>
#include <rcl/rcl.h>
#include <rclc/rclc.h>
#include <rclc/executor.h>
#include <std_msgs/msg/int32.h>
#include <WiFi.h>
```

Setting up WiFi credentials and network configuration.

```
// WiFi credentials
char* ssid = "*******";
char* password = "********";

// micro-ROS agent IP and port
char* ip = "192.168.220.75";
const uint32_t agent_port = 8888;
```

Declaring node, LDR publisher and other ROS resources.

```
// ROS node and publisher
rcl_node_t node;
rcl_publisher_t pub_ldr;
rclc_executor_t executor;
rcl_allocator_t allocator;
rclc_support_t support;
```

Defining sensor pin, message and checking connection (optional)

```
// LDR Pin
#define LDR_PIN 34
// Message
std_msgs__msg__Int32 ldr_msg;

// Check agent connection function
bool check_agent_connection() {
  return rmw_uros_ping_agent(100, 1) == RMW_RET_OK;
}
```

Setting up all things including pinMode, WiFi, wifi transport, allocator, node initialization, sensor data publisher and the executor.

```
void setup() {
  pinMode(LDR_PIN, INPUT);

  // Connect to WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
  }
  // Setup micro-ROS WiFi transport
  set_microros_wifi_transports(ssid, password, ip, agent_port);
  delay(2000);
  // Wait for agent
  while (!check_agent_connection()) {
    delay(1000);
  }

  allocator = rcl_get_default_allocator();
  // Initialize ROS support
  rclc_support_init(&support, 0, NULL, &allocator);
  // Create node
  rclc_node_init_default(&node, "esp32_single_sensor_node", "", &support);
  // Create publisher for LDR sensor
  rclc_publisher_init_default(&pub_ldr, &node, ROSIDL_GET_MSG_TYPE_SUPPORT(std_msgs,
msg, Int32), "ldr_value");
  // Create executor
  rclc_executor_init(&executor, &support.context, 1, &allocator);
}
```

Publishing the data continuously in loop

```
void loop() {
  if (check_agent_connection()) {
    ldr_msg.data = analogRead(LDR_PIN);
    rcl_publish(&pub_ldr, &ldr_msg, NULL);
  }
  delay(500);
}
```

Arduino Sketch: Set up your respective credentials and upload the above full code to ESP32.

Execution: Run the following command with your respective PC or ROS IP.

```
ros2 run micro_ros_agent micro_ros_agent udp4 -i 192.168.220.75 -p 8888
```