```
/*
  Tank30_Level_MQTT.ino
  Water level monitor for 30" tank using probes + ESP32
  Date: 2025-09-24 (PDT)

  Features:
   - 5 probes (Empty, 1/4, 1/2, 3/4, Full)
   - Debounced probe reads
   - Reports current tank level to MQTT
   - Auto Home Assistant discovery

  Wiring:
   - Bottom ground probe -> ESP32 GND
   - Empty probe -> GPIO22
   - 1/4 probe  -> GPIO22
   - 1/2 probe  -> GPIO19
   - 3/4 probe  -> GPIO18
   - Full probe -> GPIO5
*/

#include <WiFi.h>
#include <PubSubClient.h>

// --- Wi-Fi credentials ---
const char* ssid = "SSID";
const char* password = "PASSWORD";

// --- MQTT Broker (Home Assistant / Mosquitto) ---
const char* mqtt_server = "192.168.1.210";   // Change to your broker IP
const int mqtt_port = 1883;
const char* mqtt_user = "USER";          // MQTT username
const char* mqtt_pass = "PASSWORD";          // MQTT password

// --- WiFi & MQTT objects ---
WiFiClient espClient;
PubSubClient client(espClient);

// ---------- Tank probe pins ----------
#define PROBE_EMPTY 22
#define PROBE_QUARTER 21
#define PROBE_HALF 19
#define PROBE_THREEQ 18
#define PROBE_FULL 5

// ---------- MQTT topics ----------
const char* tankTopic = "homeassistant/sensor/tankA_level/state";
const char* configTopic = "homeassistant/sensor/tankA_level/config";

// ---------- Helper: debounced read ----------
bool readProbe(int pin) {
  int lowCount = 0;
  for (int i = 0; i < 5; i++) {
    if (digitalRead(pin) == LOW) {
      lowCount++;
```

```
    }
    delay(20);
  }
  return (lowCount >= 4);
}

// ---------- WiFi connect ----------
void setup_wifi() {
  delay(10);
  Serial.println("Connecting to WiFi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
}

// ---------- MQTT reconnect ----------
void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (client.connect("Tank30Client", mqtt_user, mqtt_pass)) {
      Serial.println("connected");

      // --- Publish Home Assistant discovery config ---
      String payload = "{";
      payload += "\"name\":\"Tank A Level\",";
      payload += "\"uniq_id\":\"tank_a_level\",";
      payload += "\"stat_t\":\"" + String(tankTopic) + "\",";
      payload += "\"ic\":\"mdi:water\"";
      payload += "}";
      client.publish(configTopic, payload.c_str(), true);

    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5s");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);

  pinMode(PROBE_EMPTY, INPUT_PULLUP);
  pinMode(PROBE_QUARTER, INPUT_PULLUP);
  pinMode(PROBE_HALF, INPUT_PULLUP);
  pinMode(PROBE_THREEQ, INPUT_PULLUP);
  pinMode(PROBE_FULL, INPUT_PULLUP);

  setup_wifi();
```

```
  client.setServer(mqtt_server, mqtt_port);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  // Read probes
  bool emptyWet = readProbe(PROBE_EMPTY);
  bool quarterWet = readProbe(PROBE_QUARTER);
  bool halfWet = readProbe(PROBE_HALF);
  bool threeqWet = readProbe(PROBE_THREEQ);
  bool fullWet = readProbe(PROBE_FULL);

  String tankLevel = "EMPTY";
  if (fullWet) {
    tankLevel = "FULL";
  } else if (threeqWet) {
    tankLevel = "3/4";
  } else if (halfWet) {
    tankLevel = "1/2";
  } else if (quarterWet) {
    tankLevel = "1/4";
  } else if (emptyWet) {
    tankLevel = "NEAR EMPTY";
  }

  Serial.print("Tank A Level: ");
  Serial.println(tankLevel);

  client.publish(tankTopic, tankLevel.c_str(), true);

  delay(2000);
}
```