```
#include <WiFi.h>
#include  <WiFiUdp.h>

// === Wi-Fi Access Point Config ===
const char* ssid = "PicoReceiver";
const char* password = "ReceiverPass123";

// === UDP Receiver ===
WiFiUDP udp;
const int port = 4210;
const int bufferSize = 512;
uint8_t  udpBuffer[bufferSize];

// === Audio Memory ===
#define MAX_AUDIO_SIZE  64000
uint8_t  audioData[MAX_AUDIO_SIZE];
size_t audioIndex = 0;
bool receiving = true;

// === PWM Output Config ===
const int pwmPin = 15;          // GPIO 15
const int sampleRate = 8000;    // 8 kHz playback
unsigned long lastSampleMicros = 0;
volatile size_t playIndex = 0;
bool playbackStarted = false;

void setup() {
   Serial.begin(115200);
   delay(2000);

  // Set up PWM pin
   pinMode(pwmPin, OUTPUT);
  analogWriteFreq(31250);        // Fast PWM frequency
  analogWriteResolution(8);      // 8-bit resolution
  analogWrite(pwmPin, 127);      // Mid-level
  // Start Wi-Fi AP
  WiFi.mode(WIFI_AP);
   WiFi.softAP(ssid, password);
   delay(500);

   IPAddress IP = WiFi.softAPIP();
   Serial.println("===== Pico W UDP Audio Receiver =====");
   Serial.print("Access Point IP Address: "); Serial.
println(IP);
   Serial.print("SSID: "); Serial.println(ssid);
   Serial.print("Password: "); Serial.println(password);

   udp.begin(port);
   Serial.print("Listening on UDP port "); Serial.println(port
}

void loop() {
  // === Audio Receive Section ===
   int packetSize = udp.parsePacket();
  if (packetSize && receiving) {
    int len = udp.read(udpBuffer, bufferSize);
    if (len > 0) {
      if (len == 3 && strncmp((char*)udpBuffer, "END", 3) ==
0) {
        receiving = false;
        Serial.println("=== Transfer Complete ===");
        Serial.print("Total bytes received: ");
        Serial.println(audioIndex);

        Serial.println("Starting playback...");
        playIndex = 0;
        playbackStarted = true;
        lastSampleMicros = micros();
      }
      else if (audioIndex + len < MAX_AUDIO_SIZE) {
        memcpy(&audioData[audioIndex], udpBuffer, len);
        audioIndex += len;

        Serial.print("Received: ");
        Serial.print(len);
        Serial.print(" bytes | Total: ");
        Serial.println(audioIndex);
      }
      else {
        Serial.println(" Buffer full! Stopping reception.");
        receiving = false;
      }
    }
  }

  // === Audio Playback Section ===
  if (playbackStarted) {
    unsigned long now = micros();
    if (now - lastSampleMicros >= (1000000UL / sampleRate)) {
      lastSampleMicros = now;

      if (playIndex < audioIndex) {
        analogWrite(pwmPin, audioData[playIndex++]); // 8-bit
value
      } else {
        Serial.println("=== Playback Finished ===");
        playbackStarted = false;
      }
    }
  }
}
```