# Sender Esp32 Code

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

const char* ssid = "Galaxy M026656";
const char* password = "768484823";

const char* serverIP = "192.168.43.201";  // IP of Receiver ESP32 (static IP recommended)
const int pirPin = 14;
const int ldrPin = 34;

void setup() {
  Serial.begin(115200);
  pinMode(pirPin, INPUT);

  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("\nConnected! IP: " + WiFi.localIP().toString());
}

void loop() {
  int ldrValue = analogRead(ldrPin);
  int motion = digitalRead(pirPin);

  // Send LDR data
  sendData("/ldr?value=" + String(ldrValue));

  // Send motion detection
  if (motion == HIGH) {
    sendData("/pir?motion=1");
  } else {
    sendData("/pir?motion=0");
  }

  delay(2000);  // Delay to avoid flooding
}

void sendData(String endpoint) {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    String url = "http://" + String(serverIP) + endpoint;
    http.begin(url);
    int httpCode = http.GET();
    http.end();
    Serial.println("Sent to: " + url + " | Code: " + String(httpCode));
  }
}
```

# Reciever Esp32 Code

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

const char* ssid = "Galaxy M026656";
const char* password = "768484823";

WebServer server(80);

#define RELAY1 16
#define RELAY2 17
#define RELAY3 18
#define RELAY4 19
#define RELAY5 21
#define RELAY6 22

int ldrThreshold = 1500; // Adjust based on environment

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

  pinMode(RELAY1, OUTPUT);
  pinMode(RELAY2, OUTPUT);
  pinMode(RELAY3, OUTPUT);
  pinMode(RELAY4, OUTPUT);
  pinMode(RELAY5, OUTPUT);
  pinMode(RELAY6, OUTPUT);

  digitalWrite(RELAY1, LOW);
  digitalWrite(RELAY2, LOW);
  digitalWrite(RELAY3, LOW);
  digitalWrite(RELAY4, LOW);
  digitalWrite(RELAY5, LOW);
  digitalWrite(RELAY6, LOW);

  WiFi.begin(ssid, password);
  Serial.print("Connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  Serial.println("\nConnected to WiFi! IP: " + WiFi.localIP().toString());

  // Handle LDR
  server.on("/ldr", HTTP_GET, []() {
    if (server.hasArg("value")) {
      int ldrValue = server.arg("value").toInt();
      Serial.println("LDR: " + String(ldrValue));
      if (ldrValue < ldrThreshold) {
        digitalWrite(RELAY1, HIGH); // Turn ON light
      } else {
        digitalWrite(RELAY1, LOW);  // Turn OFF light
      }
    }
```

```
    server.send(200, "text/plain", "OK");
  });

  // Handle PIR
  server.on("/pir", HTTP_GET, []() {
    if (server.hasArg("motion")) {
      int motion = server.arg("motion").toInt();
      Serial.println("Motion: " + String(motion));
      digitalWrite(RELAY2, motion == 1 ? HIGH : LOW);
    }
    server.send(200, "text/plain", "OK");
  });

  // Web page for Relay 3-6 control
  server.on("/", HTTP_GET, []() {
    String html = "<h2>Relay Control</h2>";
    for (int i = 3; i <= 6; i++) {
      html += "<p>Relay " + String(i) + ": <a href='/relay?ch=" + String(i) + "&state=1'>ON</a> | <a href='/relay?ch=" + String(i) + "&state=0'>OFF</a></p>";
    }
    server.send(200, "text/html", html);
  });

  server.on("/relay", HTTP_GET, []() {
    if (server.hasArg("ch") && server.hasArg("state")) {
      int ch = server.arg("ch").toInt();
      int state = server.arg("state").toInt();
      int pin = getRelayPin(ch);
      if (pin > 0) {
        digitalWrite(pin, state == 1 ? HIGH : LOW);
        server.send(200, "text/plain", "Relay " + String(ch) + (state == 1 ? " ON" : " OFF"));
      } else {
        server.send(404, "text/plain", "Invalid Relay");
      }
    } else {
      server.send(400, "text/plain", "Missing Params");
    }
  });

  server.begin();
}

int getRelayPin(int ch) {
  switch (ch) {
    case 3: return RELAY3;
    case 4: return RELAY4;
    case 5: return RELAY5;
    case 6: return RELAY6;
    default: return -1;
  }
}

void loop() {
  server.handleClient();
}
```