ESP32 Internet-Controlled Car with LCD Text Display This report includes full explanation and the working code used for the ESP32-based joystick-controlled car with LCD text display. --- CODE BELOW ---

```
#include <WiFi.h>
#include <WebServer.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Wi-Fi credentials
const char* ssid = "YOUR_WIFI";
const char* password = "YOUR_PASSWORD";

// Motor pins
#define IN1 19
#define IN2 21
#define IN3 22
#define IN4 5

// LCD setup
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27

// Web server
WebServer server(80);
String lcdText = "";

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

  // Motor pins
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  // LCD
  lcd.init();
  lcd.backlight();
  lcd.clear();

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

  // Web routes
  server.on("/", handleRoot);
  server.on("/F", [](){ forward(); server.send(200); });
  server.on("/B", [](){ backward(); server.send(200); });
  server.on("/L", [](){ left(); server.send(200); });
  server.on("/R", [](){ right(); server.send(200); });
  server.on("/S", [](){ stopCar(); server.send(200); });
  server.on("/addText", [](){
    if (server.hasArg("text")) {
      lcdText = server.arg("text");
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print(lcdText.substring(0,16)); // max 16 chars per line
    }
    server.sendHeader("Location", "/"); // redirect to main page
    server.send(303);
  });
  server.on("/clearText", [](){
    lcdText = "";
    lcd.clear();
    server.sendHeader("Location", "/");
    server.send(303);
  });

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

```
  server.handleClient();
}

void handleRoot() {
  server.send(200, "text/html", R"rawliteral(
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
  body { text-align:center; font-family:Arial; background:#2c3e50; color:white;}
  h1 { margin-top:20px;}
  .btn { width:70px;height:70px;margin:5px;border-radius:50%;border:none;background:#e74c3c;color:wh
  .btn:active { box-shadow:0 2px #c0392b; transform:translateY(3px);}
  .row { display:flex; justify-content:center; margin:5px;}
  input[type=text] { width:60%; padding:5px; font-size:16px; margin:10px 0;}
  .actionBtn { padding:10px 20px; font-size:16px; margin:5px; background:#3498db; color:white; borde
  .actionBtn:active { background:#2980b9; }
</style>
</head>
<body>
<h1>ESP32 Car Controller</h1>

<!-- Joystick -->
<div class="row"><button onclick="fetch('/F')" class="btn">■</button></div>
<div class="row">
  <button onclick="fetch('/L')" class="btn">■</button>
  <button onclick="fetch('/S')" class="btn">■</button>
  <button onclick="fetch('/R')" class="btn">➡</button>
</div>
<div class="row"><button onclick="fetch('/B')" class="btn">■</button></div>

<!-- Text input -->
<form action="/addText" method="get">
  <input type="text" name="text" placeholder="Enter text for LCD" maxlength="32">
  <br>
  <button type="submit" class="actionBtn">Add Text</button>
  <button type="button" onclick="fetch('/clearText')" class="actionBtn">Clear Text</button>
</form>

</body>
</html>
  )rawliteral");
}

// Movement functions
void forward()  { motor(1,0,1,0); }
void backward() { motor(0,1,0,1); }
void left()     { motor(0,1,1,0); }
void right()    { motor(1,0,0,1); }
void stopCar()  { motor(0,0,0,0); }

void motor(int a, int b, int c, int d) {
  digitalWrite(IN1, a);
  digitalWrite(IN2, b);
  digitalWrite(IN3, c);
  digitalWrite(IN4, d);
}
```