

Ein über WLAN gesteuertes Robot-Fahrzeug

Bluetooth und WLAN erscheinen oft wie konkurrierende Systeme.

Für den mobilen Einsatz außerhalb von Heimnetzwerken hat sich Bluetooth etabliert, weil darüber Geräte direkt miteinander verbunden werden können. Eine typische Anwendung ist die Freisprecheinrichtung in Kraftfahrzeugen, die eine Verbindung zwischen Handy und dem Audio-System herstellt und damit der Fahrerin bzw. dem Fahrer Ferngespräche während der Fahrt ermöglicht.

Innerhalb von Heimnetzwerken ist demgegenüber WLAN meist die bessere Wahl. Geräte, die sich in ein WLAN-Netz einwählen, bekommen eine IP-Adresse zugewiesen, über die sie mittels Internet-Browser angesprochen und gesteuert werden können.

Hier ein Beispiel, das zeigt, wie ein Robot-Fahrzeug über WLAN ferngesteuert werden kann.

Benötigte Hardware: Robino-I-Fahrgestell, NodeMCU

Programmcode:

```
#include <ESP8266WiFi.h>
const char* ssid = "Repeater";
const char* password = "39090083131170350355";
int mot1 = D1; // Motor 1
int mot2 = D2; // Motor 2
WiFiServer server(80);
void setup() {
  Serial.begin(115200);
  delay(10);
  pinMode(mot1, OUTPUT);
  pinMode(mot2, OUTPUT);
  digitalWrite(mot1, LOW);
  digitalWrite(mot2, LOW);
  // Mit dem WLAN-Netzwerk verbinden
  Serial.println();
  Serial.println();
  Serial.print("Verbinde mit ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("Mit dem Router verbunden");
  Serial.println();
  // Server starten
  server.begin();
  Serial.println("Server gestartet");
  Serial.println();
  Serial.println();
  // IP-Adresse für den Aufruf des ESP-Servers
  Serial.print("Kopiere die nachstehende IP in die Adresszeile deines Browsers,");
  Serial.println();
  Serial.print("um die Verbindung herzustellen: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");
}
void loop() {
  // Verbindungsprüfung
  WiFiClient client = server.available();
  if (!client){
```

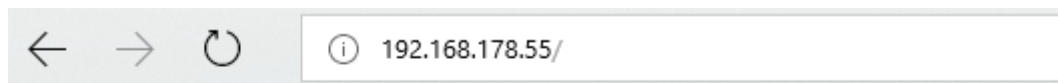
```

    return;
}
// Aktivitäten vom Server im Monitor anzeigen
Serial.println("neue Aktivitaet ");
while (!client.available()) {
    delay(100);
}
// Erste Zeile der Eingabe lesen
String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();
// LED entsprechend der Eingabe an- oder ausschalten
int value1 = HIGH;
int value2 = HIGH;
if (request.indexOf("/mot1an") != -1) {
    digitalWrite(mot1, HIGH);
    value1 = HIGH;
}
if (request.indexOf("/mot1aus") != -1) {
    digitalWrite(mot1, LOW);
    value1 = LOW;
}
if (request.indexOf("/mot2an") != -1) {
    digitalWrite(mot2, HIGH);
    value2 = HIGH;
}
if (request.indexOf("/mot2aus") != -1) {
    digitalWrite(mot2, LOW);
    value2 = LOW;
}
// Auf die Eingabe reagieren
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("");
client.println("<!DOCTYPE HTML>");
client.println("<html>");

    client.println("Klicke <a href=\" /mot1an\"><b>hier</b></a> um Motor-1
<b>an</b>zuschalten - - ");
    client.println("Klicke <a href=\" /mot1aus\"><b>hier</b></a> um Motor-1
<b>aus</b>zuschalten.<br><br>");
    client.println("Klicke <a href=\" /mot2an\"><b>hier</b></a> um Motor-2
<b>an</b>zuschalten - - ");
    client.println("Klicke <a href=\" /mot2aus\"><b>hier</b></a> um Motor-2
<b>aus</b>zuschalten");
    client.println("</html>");
    delay(100);
    Serial.println("Verbindung getrennt");
    Serial.println("");
}

```

Nach Übertragen des Codes im seriellen Monitor die IP-Adresse der NodeMCU kopieren und in die Adresszeile eines Webbrowsers einfügen, was zu folgender Anzeige führt:

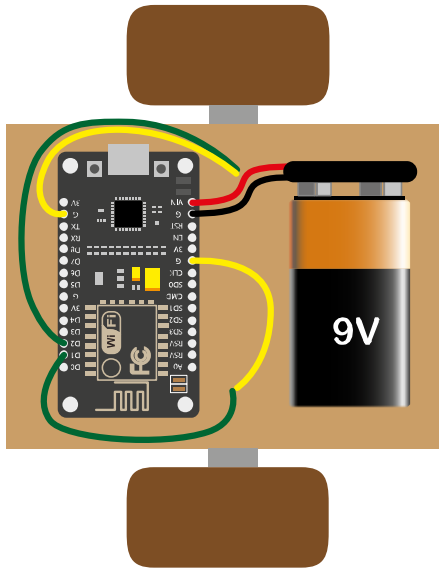


Klicke [hier](#) um Motor-1 **anzuschalten** - - Klicke [hier](#) um Motor-1 **auszuschalten**.

Klicke [hier](#) um Motor-2 **anzuschalten** - - Klicke [hier](#) um Motor-2 **auszuschalten**

Beide Motoren an bedeutet Geradeausfahrt, linker Motor aus, rechter an, Linkskurve, rechter aus, linker an, Rechtskurve.

Und so sieht der Aufbau aus:



Details zur Konstruktion des Fahrgestells und die Anbringung der Motoren kann der [Anleitung](#) zum Bau des Robino-I entnommen werden. Getriebemotoren, Clip zum Anschluss der Batterie, Kabel und NodeMCU gibt es z.B. im [Eckstein-Shop](#). 9V Blocks sind günstig bei Aldi und Lidl.

Der 9V Block übernimmt die Stromversorgung. Der Pluspol ist an Vin und der Minuspol mit GND der NodeMCU verbunden.

Die Motoren werden wie folgt angeschlossen: Je ein Anschluss der beiden Motoren wird GND verbunden, die übrigen Anschlüsse mit D1 und D2.

Die Drehrichtung der Motoren kann durch Umpolung verändert werden.