

Smarte Garagentore

Hauseigene Anlage
mit ESP 8266



Autor -Waldemar Breiling DJ1DZ
Erstellt - April 2026

Inhaltsverzeichnis

1. VORWORT	3
2. PLANUNG	4
3. SCHALTPLAN	5
4. WAS WIRD GEBRAUCHT	6
5. AUFBAU	7
6. TEST PHASEN	8
7. ENDMONTAGE	9
8. PROGRAMM ABLAUFPLAN	10
9. PROGRAMM CODE MIT ARDUINO IDE	11
10 Nachsatz	

1. VORWORT

Ein Garagentor elektrisch zu betreiben ist heute kein großes Problem. Sind es 2 Garagen, sind sie auch getrennt zu bedienen. Dann erhalten sie, wenn möglich, einen Taster, der innerhalb der Garage installiert wird. Eine Fernbedienung ermöglicht das Öffnen und Schließen der Tore. Was fehlt, ist eine Anzeige, die das Tor als geschlossen oder geöffnet darstellt. Wenn ich jetzt noch die Garagentore per Handy bedienen könnte, wäre mir das smart genug.

2. PLANUNG

Welche Voraussetzungen wären wünschenswert?

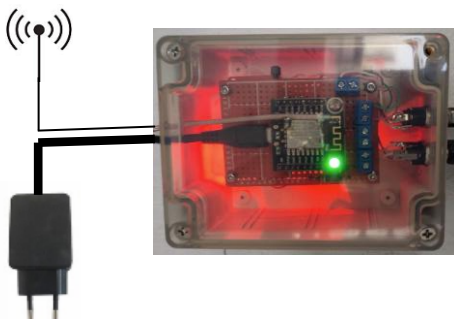
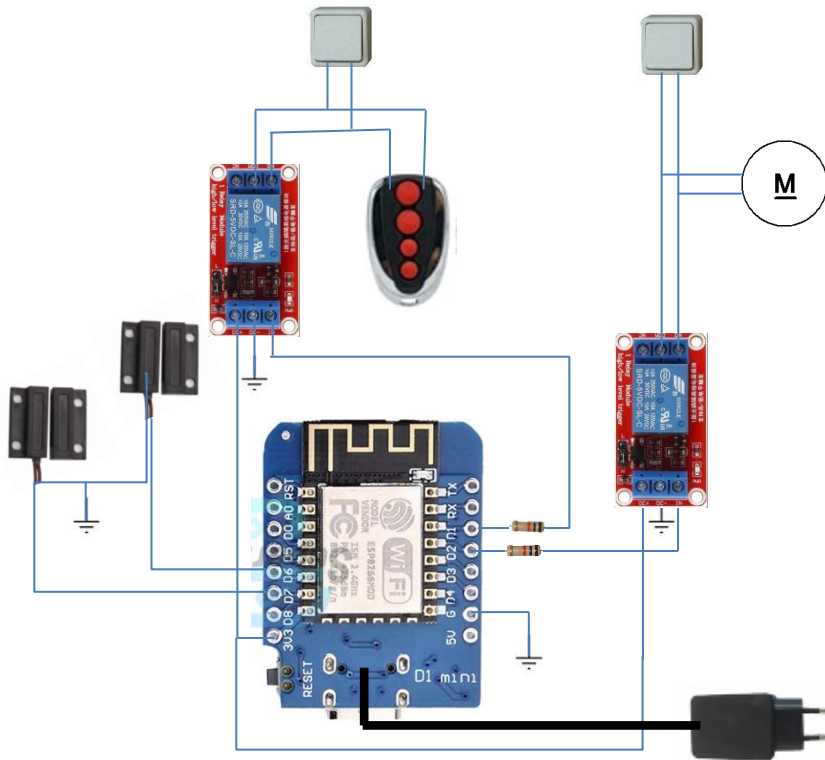
- Benutzung normal mit Fernbedienung
- Der innere Taster ist notwendig
- Bedienung auch per Handy
- Per Handy ist der Status (auf/zu) erkennbar
- Es müssen 2 Tore einbezogen werden.

Nach erster Prüfung wurde festgestellt, dass ein Torantrieb keinen Anschluss für einen Taster hat.

Zweites Problem: Innerhalb der Garage ist kein WLAN vorhanden.

3. SCHALTPLAN

Garagentor Öffner System



4. WAS WIRD GEBRAUCHT

Liste der benötigten Teile:

1. Microprozessor – ESP 8266 1 x



2. Relais 2 x



3. Reed Kontakte – MC 38 2 x



4. Taster über Putz 2 x



5. Fernbedienung – Steel Line Zt07 3 x



6. Lochrasterplat. 7x5cm, 2,25mm 1 x



7. Niedervolt Stecker 5x2,1mm 4 x



8. Netzteil 230V/5V/0,7A 1 x



9. Gehäuse mit Klarsichtdeckel
IP55, 115x90x55 (LBH) 1 x



5. AUFBAU

Der Aufbau sollte so erfolgen, dass bei einer Reparatur oder Revision das Gehäuse allein abgebaut werden kann. Jedes Kabel, das nach außen geführt wird, ist mit ,Stecker am Gehäuse vorgesehen.

Zu dem Problem: Ein Tor hat keinen Tasteranschluss! wird eine Fernbedienung genutzt, bei der die Kontakte der Taste 1 parallel zum Taster benutzt werden. Wird also der Taster betätigt, sendet die Fernbedienung.

Und das Problem – kein WLAN innerhalb der Garage! wird gelöst durch Anschluss eines Antennenkabels, wobei die Antenne auf der Platine durch Trennung abgeschaltet wird.

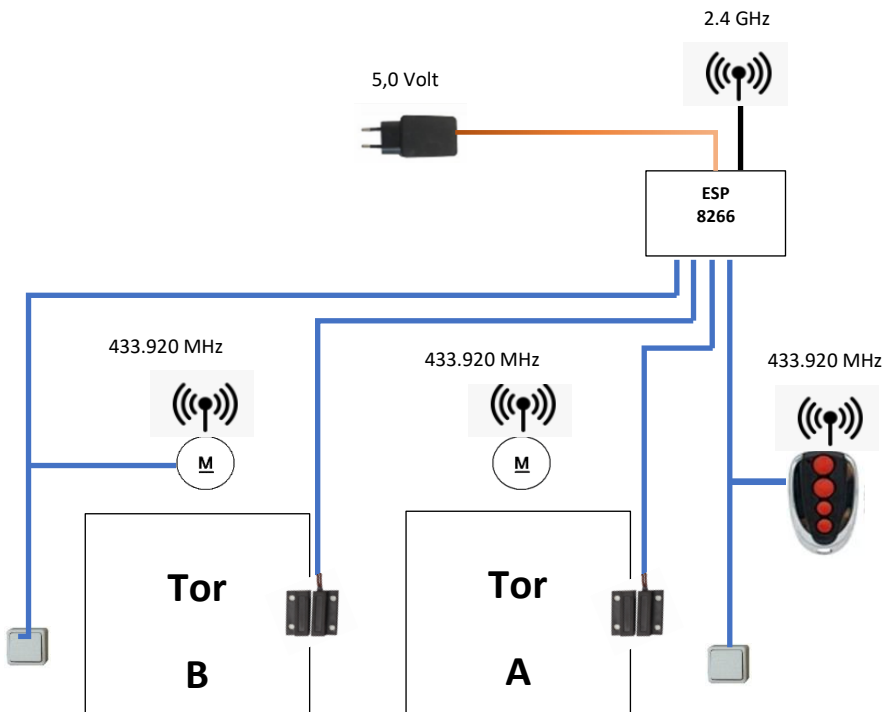
6. TEST PHASEN

Es muss sichergestellt werden, dass der Empfang vor Ort ausreichende Feldstärke liefert. Dazu wird die Antenne auf der ESP 8266 Platine mit einem Cat-Messer getrennt und ein Antennenkabel RG 316, 50 Ohm, dort angelötet. Dies ergab ausreichende Empfangswerte, nachdem die Antenne außerhalb der Garage positioniert wurde.

Jetzt wurden auch die Reed-Kontakte und die Taster getestet.

7. ENDMONTAGE

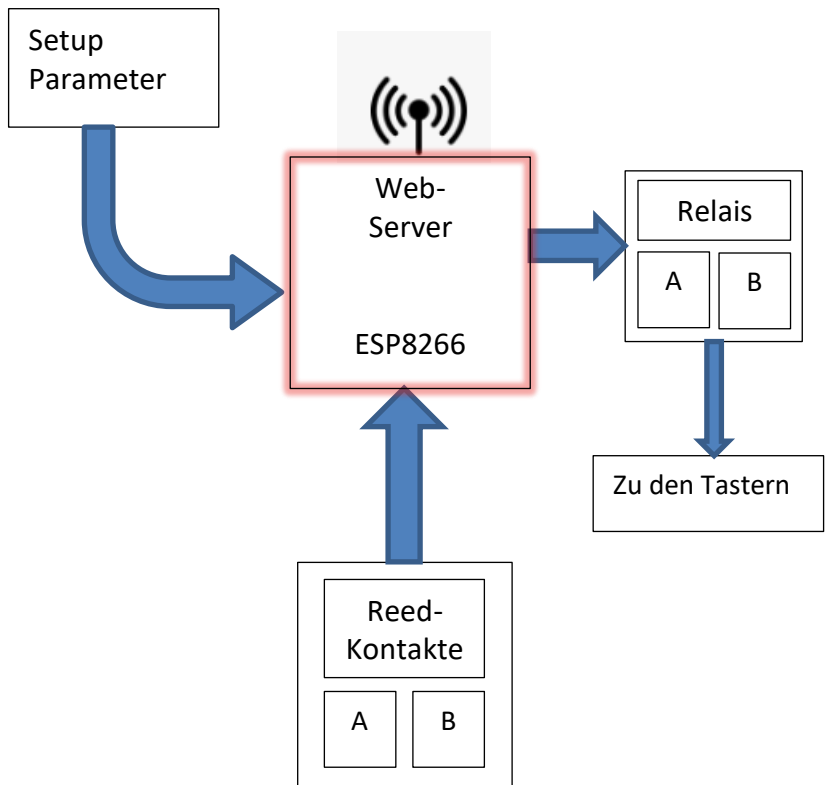
Alle Leitungen werden in Schutzrohre verlegt.
Steckdosen sind sinnvoll angeordnet. Das
Antennenkabel wird durch eine Öffnung unterhalb
der Dachverkleidung nach außen verlegt.
Alle Schalter sind innenliegend.



8. PROGRAMM ABLAUFPLAN

Vereinfacht werden alle notwendigen Parameter gesetzt und der Webserver eingerichtet.

Wird über Wlan ein Befehl gesendet, so wird das entsprechende Relais A oder B angesprochen und der Torzustand wird durch die Reedkontakte ausgegeben



9. PROGRAMM CODE MIT ARDUINO IDE

Die Programmierung wurde mit Hilfe von „KI“ erstellt.

Gesteuert werden 2 Garagentore.

Zeile 7 und 8:

Eigenes Wlan mit zugehörigem Passwort eintragen.

Zeile 11 bis 14:

Hier werden die Pins D1 und D2 für die Taster benutzt. Die Reed-Kontakte liegen auf Pin14 und Pin12 und zeigen an ob das Tor auf oder zu ist.



Zeile 19 bis 49 :

Diese Zeilen dienen zur Erzeugung der Webseite.

Zeile 51 bis 93:

Diese Zeilen beschreiben das Handling der Webseite.

Zeile 94 bis 97:

Hier wird auf Tasteneingabe via Webseite gewartet.

```
1 //Garagentor Steuerung "Garage A-B (1-2)"
2 //w.breiling DJ1DZ (KI)
3 #include <ESP8266WiFi.h>
4 #include <ESP8266WebServer.h>
5
6 // --- WLAN ZUGANGSDATEN ANPASSEN ---
7 const char* ssid = "eigenes Wlan";
8 const char* password = "Passwort zu eigenes Wlan";
9
10 // --- PIN DEFINITIONEN ---
11 const int RELAY_1_PIN = 5; // D1 - Steel-Line (Fernbedienung)
12 const int RELAY_2_PIN = 4; // D2 - Sihao (Direktkontakt)
13 const int REED_1_PIN = 14; // D5 - Sensor Tor 1
14 const int REED_2_PIN = 12; // D6 - Sensor Tor 2
15
16 ESP8266WebServer server(80);
17
18 // Funktion zum Erzeugen der Webseite
19 void handleRoot() {
20     String html = "<html><head><meta name='viewport' content='width=device-
21 width, initial-scale=1.0'>";
22     html += "<style>body{font-family:sans-serif; text-align:center; padding:20px;}
23 .btn{display:block; width:100%; padding:20px; margin:10px 0; font-size:20px;
24 cursor:pointer; border-radius:10px; border:none; color:white;} .btn-
25 1{background:#3498db;} .btn-2{background:#e67e22;} .status{font-weight:bold;
26 margin-bottom:20px;}</style></head><body>";
27
28     html += "<h1>Garagensteuerung</h1>";
29
30     // Status Tor 1
31     html += "<div class='status'>Tor 1 (Steel-Line): " +
32 String(digitalRead(REED_1_PIN) == LOW ? "<span
33 style='color:green'>ZU</span>" : "<span style='color:red'>OFFEN</span>") +
34 "</div>";
35     html += "<button class='btn btn-1' onclick=\"location.href='/trigger1'\">Tor 1
36 Schalten</button>";
37
38     html += "<hr>";
39
40     // Status Tor 2
41     html += "<div class='status'>Tor 2 (Sihao): " + String(digitalRead(REED_2_PIN) ==
42 LOW ? "<span style='color:green'>ZU</span>" : "<span
43 style='color:red'>OFFEN</span>") + "</div>";
```

```
44     html += "<button class='btn btn-2' onclick=\"location.href='/trigger2'\">Tor 2
45     Schalten</button>";
46
47     html += "<p><a href='/\">Status aktualisieren</a></p></body></html>";
48     server.send(200, "text/html", html);
49 }
50
51 // Funktionen zum Schalten der Relais
52 void handleTrigger1() {
53     digitalWrite(RELAY_1_PIN, LOW); // Relais an
54     delay(400); // Kurzer Impuls
55     digitalWrite(RELAY_1_PIN, HIGH); // Relais aus
56     server.sendHeader("Location", "/");
57     server.send(303);
58 }
59
60 void handleTrigger2() {
61     digitalWrite(RELAY_2_PIN, LOW);
62     delay(400);
63     digitalWrite(RELAY_2_PIN, HIGH);
64     server.sendHeader("Location", "/");
65     server.send(303);
66 }
67
68 void setup() {
69     Serial.begin(115200);
70
71     // Relais konfigurieren (HIGH = Aus bei Active-Low Relais)
72     pinMode(RELAY_1_PIN, OUTPUT); digitalWrite(RELAY_1_PIN, HIGH);
73     pinMode(RELAY_2_PIN, OUTPUT); digitalWrite(RELAY_2_PIN, HIGH);
74
75     // Sensoren konfigurieren
76     pinMode(REED_1_PIN, INPUT_PULLUP);
77     pinMode(REED_2_PIN, INPUT_PULLUP);
78
79     // WLAN Verbindung
80     WiFi.begin(ssid, password);
81     Serial.print("Verbinde mit WLAN...");
82     while (WiFi.status() != WL_CONNECTED) {
83         delay(500); Serial.print(".");
84     }
85     Serial.println("\nVerbunden!");
86     Serial.print("IP-Adresse: "); Serial.println(WiFi.localIP());
```

```
87
88     // Webservice Routen
89     server.on("/", handleRoot);
90     server.on("/trigger1", handleTrigger1);
91     server.on("/trigger2", handleTrigger2);
92     server.begin();
93 }
94 // Warten auf Tastendruck
95 void loop() {
96     server.handleClient();
97 }
```

Nachsatz

Ein Problem ist aufgetreten.

Nachdem die Fernbedienungen an der Motoreinheit programmiert wurde und auch die Auf- und Zu-Positionen eingestellt waren, ist nach ca. 5 Minuten das Tor automatisch auf die ZU-Position gefahren. Diese Funktion ist in der Betriebsanweisung nicht beschrieben.

Im Internet wurde aber gezeigt, dass ein RESET durchgeführt werden kann.

1. Stromstecker ziehen
2. P-Taste drücken und Stromstecker einstecken.
3. 10 Sekunden auf LED Blinken warten.
4. P-Taste los lassen.

Garagentor Steuerung:

Led 1 blinkt --Krafteinstellung beim Schließen

Led 2 blinkt --AUF- Programmierung

Led 3 blinkt --Testfahrt

Led 4 blinkt --ZU- Programmierung

Led 6 blinkt --Krafteinstellung beim Öffnen

Led 7 blinkt --Fernbedienung programmieren

Led 5 blinkt --?

Led 8 blinkt --?

1