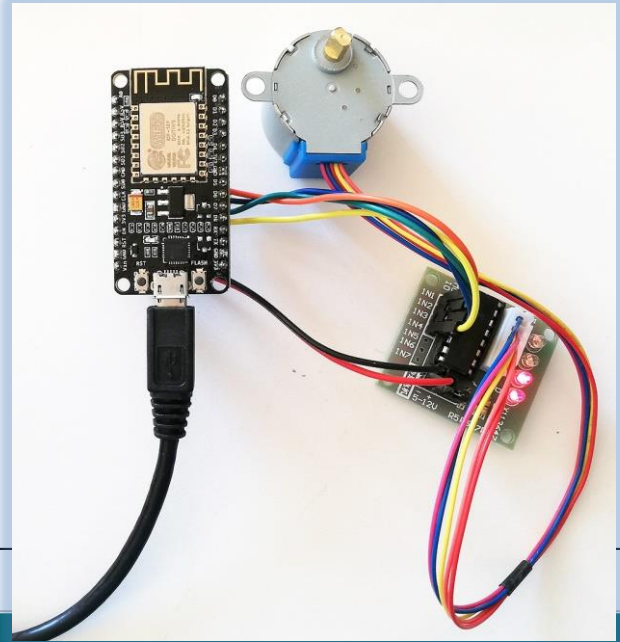


28BYJ-48-Schrittmotor - WLAN-gesteuert über ESP8266-Board und ULN2003-Treiberplatine



```
NodeMcuMotorRadio | Arduino 1.8.12
Datei Bearbeiten Sketch Werkzeuge Hilfe

NodeMcuMotorRadio

#include <Stepper.h>
#include<ESP8266WiFi.h> #include<ESP8266WiFiAP.h> #include<ESP8266WiFiGeneric.h> #include<ESP8266WiFiMulti.h>
#include<ESP8266WebServer.h> #include<ESP8266WiFiSTA.h> #include<ESP8266WiFiType.h>#include<ESP8266WiFiScan.h>
#include<ESP8266WiFi.h> #include<WiFiClient.h> #include<WiFiClientSecure.h>
#include<WiFiServer.h> #include<WiFiServerSecure.h> #include<WiFiUdp.h>
int richtung=1;String Wert="Vor";const int steps=100; Stepper myStepper(steps, 14, 12, 13, 15);

ESP8266WebServer server(80);//Sollwert-Eingabe
const char HTML_ANFANG[]="<!DOCTYPE html><head>"
"<meta name = 'viewport' content = 'width=device-width,initial-scale=1.0,maximum-scale=1.0'"
"<meta http-equiv='refresh'content='20'><title>ESP266 mit Schrittmotor</title></head><body><center>"
"<FORM style='font-family:arial'action='method='post'><h4>Steuern mit dem ESP266</h4>"
"<h3><INPUT type = 'radio' id='dreh' name='message' value='Vor'>Vor<br>"
"<INPUT type = 'radio' id='dreh' name='message' value='Stopp'>Stopp<br>"
"<INPUT type = 'radio' id='dreh' name='message' value='Zurueck'>Zurueck<br>"
"<Button style='font-size:18px'class='Button'name='submit'>Einstellen!</button></P></h3></FORM>";
const char HTML_ENDE[]="</body></center></html>";

String wertTabelle() //Messwert-Ausgabe:
{String tabl="<h2 style='font-family:arial'>Aktueller Wert:</h2>";tabl=tabl+"<table style='font-family:arial'>";
tabl=tabl+Wert;tabl=tabl+"</table>";return tabl;};

void HTML_Server()//Wenn Text vom Client eingetroffen, HTML-Seite neu zusammenbauen:
{if (server.hasArg("message")&& server.arg("message")!=""){Wert=server.arg("message");};
String htm=String(HTML_ANFANG)+wertTabelle()+String(HTML_ENDE);server.send(200,"text/html",htm);}

void setup()//WLAN, Server starten
{WiFi.persistent(false);WiFi.mode(WIFI_STA);delay(100);
WiFi.begin("hegburk","202187130877");Serial.begin(9600);while(WiFi.status()!=WL_CONNECTED){delay(500);Serial.print(".");};
WiFi.mode(WIFI_STA);WiFi.hostname("ESP8266");Serial.println("IP: "+WiFi.localIP().toString());delay(1000);
server.on("/",HTML_Server);server.begin();WiFi.persistent(false);WiFi.mode(WIFI_STA);delay(100);myStepper.setSpeed(15);}

void loop() //Dauerschleife
{String htm=String(HTML_ANFANG)+wertTabelle()+String(HTML_ENDE);
if(Wert=="Vor"){richtung=-1;};if(Wert=="Stopp"){richtung=0;};if(Wert=="Zurueck"){richtung=1;};
for (int i=0; i <= steps; i++){myStepper.step(richtung);delay(2);};delay(100);server.handleClient();delay(100);}
```

1. Nachschauen, ob die IDE für das Board vom Typ „Generic ESP8266 Module“ vorbereitet ist, das ggf. über die „Voreinstellungen“ nachholen.
2. Den Code in das IDE-Fenster hineinkopieren und mit „Überprüfen/Kompilieren“ testen, ob alle Bibliotheken schon in die IDE integriert sind. Die Informationen erhält man in roter Schrift im schwarzen Feld oder im Code-Fenster. Fehlendes über „Sketch => Bibliothek einbinden“ installieren.
3. Wenn das Kompilieren ohne Fehlermeldungen (Warnungen muss man erst einmal nicht beachten)ausgeführt wird, das Board über USB anschließen und den richtigen Port einstellen.
4. Richtiges WLAN mit Namen und Schlüssel in den Code eintragen, Sketch (Code) hochladen.
5. Seriellen Monitor öffnen, Reset-Taster auf dem ESP drücken. Wenn die IP-Adresse im Monitorfenster ausgegeben wird, diese kopieren und in den Browser eingeben. Die Website müsste sich öffnen. Falls im Monitorfenster keine IP-Adresse erscheint – oder die Website sich nicht öffnet, auf Fehlersuche gehen.

NodeMcuMotorRadio.ino (Rot: Eigene Werte eintragen!)

```
#include <Stepper.h>
#include<ESP8266WiFi.h> #include<ESP8266WiFiAP.h> #include<ESP8266WiFiGeneric.h> #include<ESP8266WiFiMulti.h>
#include<ESP8266WebServer.h> #include<ESP8266WiFiSTA.h> #include<ESP8266WiFiType.h> #include<ESP8266WiFiScan.h>
#include<ESP8266WiFi.h> #include<WiFiClient.h> #include<WiFiClientSecure.h>
#include<WiFiServer.h> #include<WiFiServerSecure.h> #include<WiFiUdp.h>
int richtung=1;StringWert="Vor";const int steps=100; Stepper myStepper(steps, 14, 12, 13, 15);

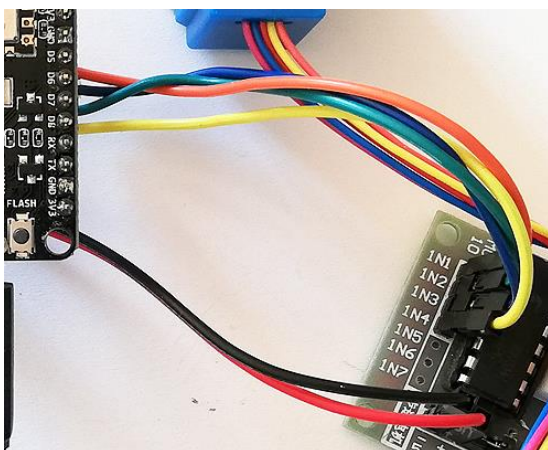
ESP8266WebServer server(80); //Sollwert-Eingabe
const char HTML_ANFANG[]="<!DOCTYPE html><head>"
"<meta name = 'viewport' content = 'width=device-width,initial-scale=1.0,maximum-scale=1.0'>"
"<meta http-equiv='refresh'content='20'><title>ESP8266 mit Schrittmotor</title></head><body><center>"
"<FORM style='font-family:arial'action='method='post'<h4>Steuern mit dem ESP8266</h4>"
"<h3><INPUT type = 'radio' id='dreh' name='message' value='Vor'>Vor<br>"
"<INPUT type = 'radio' id='dreh' name='message' value='Stopp'>Stopp<br>"
"<INPUT type = 'radio' id='dreh' name='message' value='Zurueck'>Zurueck<br>"
"<Button style='font-size:18px'class='Button' name='submit'>Einstellen!</button></P></h3></FORM>";
const char HTML_ENDE[]="</body></center></html>";

String wertTabelle() //Messwert-Ausgabe:
{String tabl="<h2 style='font-family:arial'>Aktueller Wert:</h2>";tabl=tabl+"<table style='font-family:arial'>";
tabl=tabl+Wert;tabl=tabl+"</table>";return tabl;};

void HTML_Server()//Wenn Text vom Client eingetroffen, HTML-Seite neu zusammenbauen:
{if (server.hasArg("message")&& server.arg("message")!=""){Wert=server.arg("message");};
String htm=String(HTML_ANFANG)+wertTabelle()+String(HTML_ENDE);server.send(200,"text/html",htm);}

void setup()//WLAN, Server starten
{WiFi.persistent(false);WiFi.mode(WIFI_STA);delay(100);
WiFi.begin("SSID", "Netzwerkschlüssel");Serial.begin(9600);while(WiFi.status() !=WL_CONNECTED) {delay(500);
Serial.print(".");};
WiFi.mode(WIFI_STA);WiFi.hostname("ESP8266");Serial.println("\IP:" + WiFi.localIP().toString());delay(1000);
server.on("/", HTML_Server); server.begin();WiFi.persistent(false);WiFi.mode(WIFI_STA); delay(100);
myStepper.setSpeed(15);}

void loop() //Dauerschleife
{String htm=String(HTML_ANFANG)+wertTabelle()+String(HTML_ENDE);
if(Wert=="Vor"){richtung=-1;};if(Wert=="Stopp"){richtung=0;};if(Wert=="Zurueck"){richtung=1;};
for (int i=0; i <= steps; i++){myStepper.step(richtung);delay(2);};delay(100); server.handleClient();delay(100);}
```



D5 => IN1 ... D8 => IN4