

Inhaltsverzeichnis

Einrichtung.....	2
Setup.....	2
Protokoll.....	3
Grundlagen.....	3
Api.....	3
Neue Position setzen.....	3
Aktuelle Position abfragen.....	3
Text auf dem Display ausgeben.....	3
Firmware.....	4
Konzeption.....	4
Statemachine.....	4
Quellcode.....	4
Mainboard.....	5
Anschlüsse.....	5
Komponenten.....	7
NodeMCU mit ESP8266.....	7
Verfügbare GPIO Pin's.....	7
Verwendung der GPIO Pins.....	7
Pinbelegung.....	8
Motortreiber DRV 8825.....	9
Pinbelegung.....	9
Konfiguration der Schrittgröße.....	9
Ausgabe.....	10
LCD Display mit I2C Adapter.....	10
Quellen.....	11

Einrichtung

Setup

Die Setup Oberfläche der Rotors ist über einen beliebigen Webbrowser erreichbar. Bei gültiger WLAN Konfiguration meldet sich der Rotor automatisch am WLAN an und gibt die vom Router zugewiesene IP Adresse via Display und Serieller Schnittstelle aus (siehe dazu Kapitel Ausgabe). Beim ersten Start des Rotors oder bei ungültiger WLAN Konfiguration (WLAN am Standort nicht verfügbar, Passwort wurde geändert etc.) spannt der Rotor automatisch das WLAN „rotor.dk9mbs.de“ auf und zeigt das Passwort am LCD Display für ca. 2 Sekunden an.

Anwendungen Orte System

192.168.2.110 Nachrichten - Service - St x G a

Nicht sicher 192.168.2.110

Apps pse sort me Dev HAM MP myBusiness

Setup shell by dk9mbs

Wlan:

ssid
dsl.dk9mbs.de
Password
xxxxxxxxxxxxxxxxxxxx
MAC (A4-CF-12-DF-69-00)
E0-98-06-85-B5-DD

Network:

Hostname
rotor

Admin portal

Admin Password
xxxxxxxxxxxxxxxxxxxx

Rotor:

Steps for 360 degree
5554

Save Save and Reset !!! Format fs !!!

Protokoll

Grundlagen

Zur Rotor Steuerung mittels PC kommt ein generisches HTTP Protokoll zum Einsatz. Beispielsweise kann der Rotor mit „wget“ auf einfache Weise gesteuert werden:

```
wget -O – "http://[IP Adresse des Rotors]/api?command=[Kommando]&[Weitere Parameter]"
```

Api

Neue Position setzen

Führt den Rotor an eine neue Position. Als Argument „deg“ sind numerische Werte zwischen 0 und 360 (Grad) zulässig.

```
wget -O – "http://[IP Adresse des Rotors]/api?command=MOVE&deg=45"
```

Der HTTP Request arbeitet nicht blockierend – d.h. Nach Annahme des neuen Wertes liefert der Rotor sofort den HTTP Code 200 zurück. Erst danach startet der Motor. Eine Positionsänderung ist auch während laufenden Motor möglich.

Aktuelle Position abfragen

Liefert die aktuelle Position des Rotors.

```
wget -O – "http://[IP Adresse des Rotors]/api?command=GETCURRENTPOS"
```

Text auf dem Display ausgeben

Gibt einen Text auf dem Display in der obersten Zeile aus (max. 16 Zeichen).

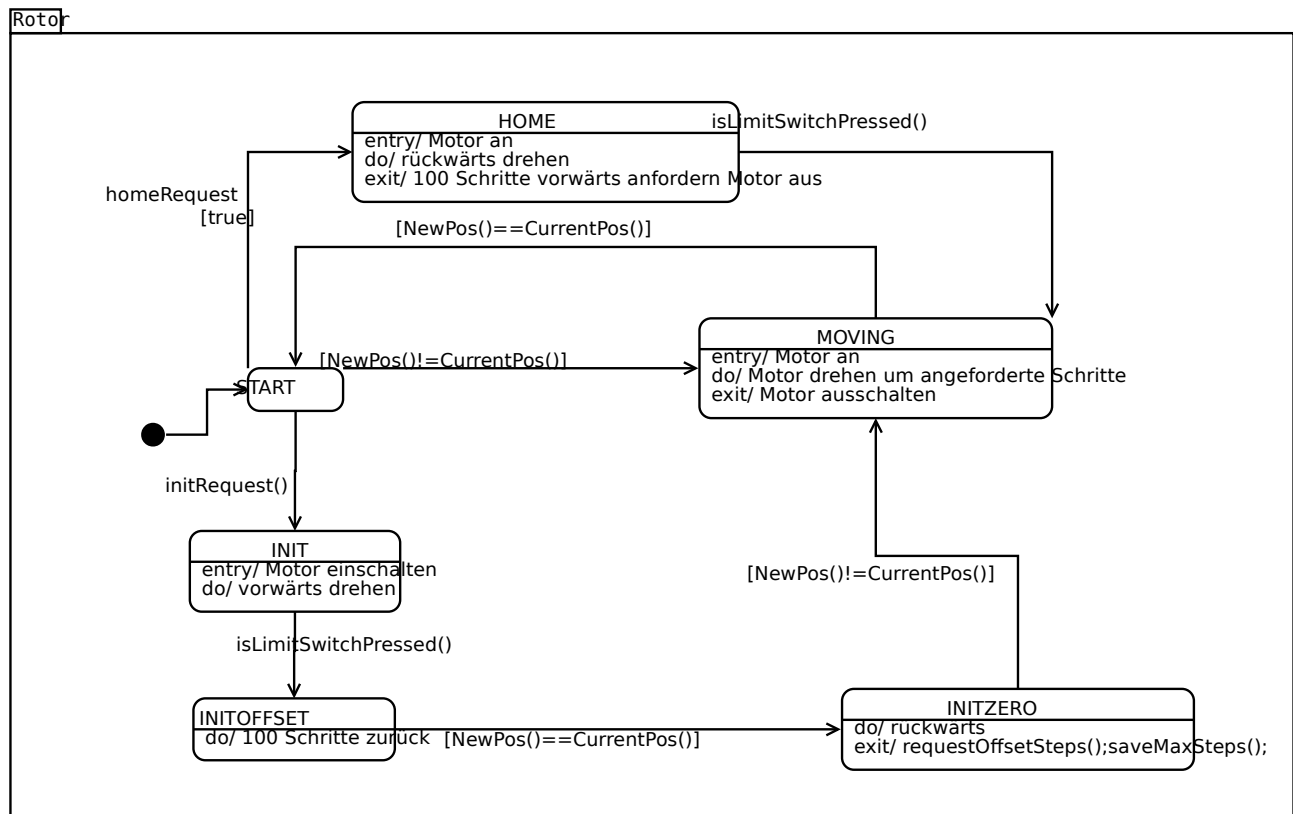
```
wget -O – "http://[IP Adresse des Rotors]/api?command=SETDISPLAY&line0=Hallo"
```

In diesem Beispiel wird „Hallo“ in der ersten Zeile des LCD Displays angezeigt.

Firmware

Konzeption

Statemachine

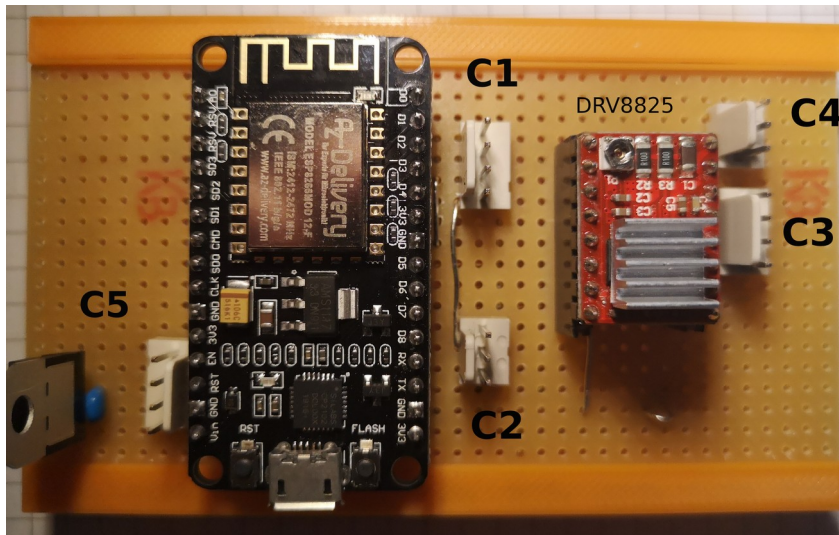


Quellcode

<https://github.com/dk9mbs/Rotor/blob/dev/fw-8266-arduino/fw-8266-arduino.ino>

Mainboard

Anschlüsse



Connector	Pin (von oben nach unten)	Beschreibung
C1		I2C Bus
	1	SCL
	2	SDA
	3	GND
	4	+5V (Vin)
C2		Endschalter
	1	nc
	2	Taster
	3	GND
C3		Stepper
	1	B2
	2	B1
	3	A1
	4	A2
C4		Power (12V)
	1	nc
	2	+12V in
	3	GND
C5		Nicht benutzt

	1	
	2	
	3	
	4	+ 5V (Vin)

Komponenten

NodeMCU mit ESP8266

Verfügbare GPIO Pin's

Nicht alle Pins können frei verwendet werden. Beispielsweise darf Pin 2 beim Booten nicht auf Masse gezogen sein.

GPIO	Function	State	Einschränkungen
0	Boot mode select	3.3V	No Hi-Z
1	TX0	-	Not usable during Serial transmission
2	Boot mode select TX1	3.3V (boot only)	Don't connect to ground at boot time Sends debug data at boot time
3	RX0	-	Not usable during Serial transmission
4	SDA (I ² C)	-	-
5	SCL (I ² C)	-	-
6-11	Flash connection	x	Not usable, and not broken out
12	MISO (SPI)	-	-
13	MOSI (SPI)	-	-
14	SCK (SPI)	-	-
15	SS (SPI)	0V	Pull-up resistor not usable
16	Wake up from sleep	-	No pull-up resistor, but pull-down instead Should be connected to RST to wake up

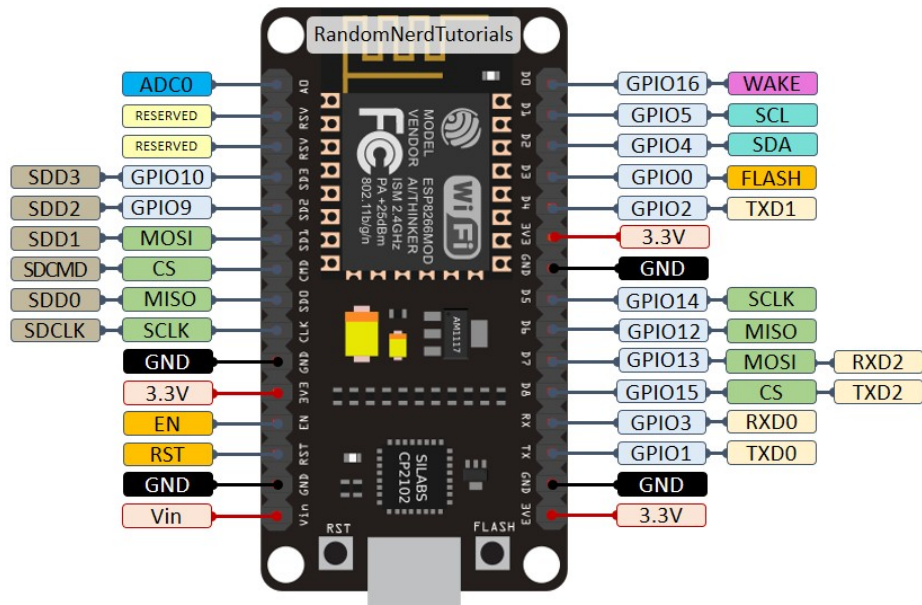
Quelle: [1]

Verwendung der GPIO Pins

Pin	Verwendung	Bemerkung
12	8825 - Step	
13	8825 - Direction	
14	8825 - Enable	
4	LCD (SDA)	10K Pulldown
5	LCD (SCL)	
2	Endschalter	4,7K Pullup Schließer

Pinbelegung

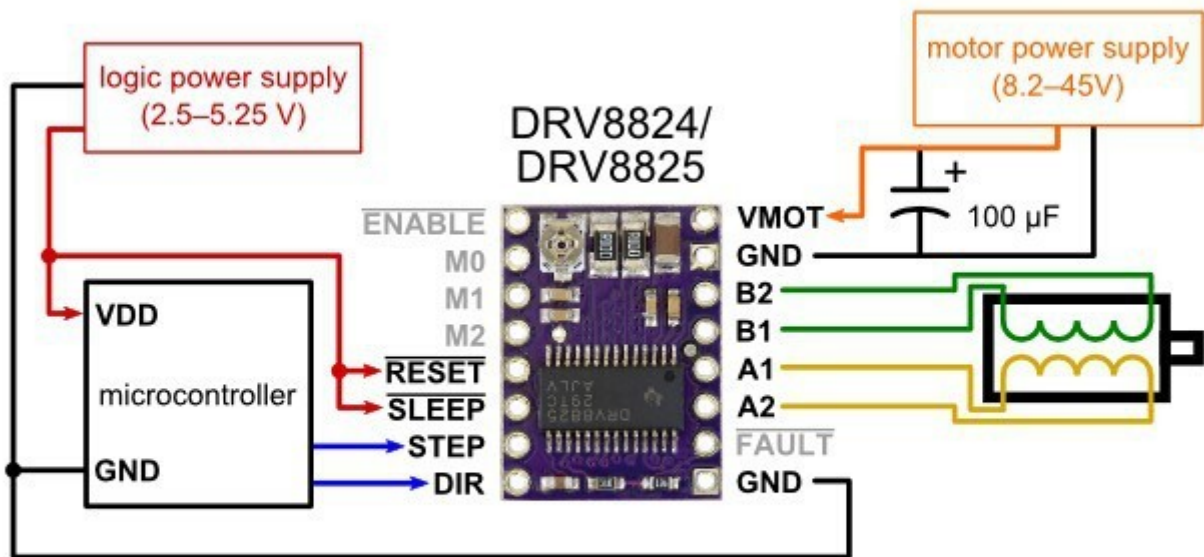
In diesem Manual wird immer von den GPIO Bezeichnungen gesprochen; nicht von den auf dem Board aufgedruckten Aliassen.



Motortreiber DRV 8825

Pinbelegung

Vor dem Betrieb mit einem neuen Motor muss der maximale Strom mit dem Potentiometer laut Herstellerangabe eingestellt werden.



Konfiguration der Schrittgröße

Über die Anschlüsse „M0“ bis „M2“ kann die Schrittgröße konfiguriert werden. Nach Änderung der Schrittgröße muss im Setup Modus des Rotors diese auf 0 gesetzt werden. Nach dem nächsten Neustart ermittelt der Rotor dann selbstständig die Anzahl der Schritte die für eine 360 Grad Drehung benötigt werden und speichert diese dauerhaft.

M0	M1	M2	Schritt Auflösung	Anzahl Schritte für eine Umdrehung
0	0	0	1	200
1	0	0	1/2	400
0	1	0	1/4	800
1	1	0	1/8	1600
0	0	1	1/16	3200
1	0	1	1/32	6400

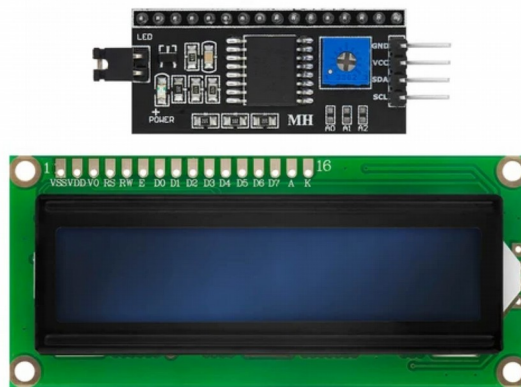
Ausgabe

Die Ausgabe von Meldungen erfolgt bei dem Rotor auf 2 Wegen:

- Über den Seriellen Port mit 115200 Baud
- Über ein am I2C Bus angeschlossenes LCD Display

Das LCD Display zeigt im Betrieb die IP Adresse des Rotors im Netzwerk und die aktuelle Position in (Alt) Grad an.

LCD Display mit I2C Adapter



Quellen

<https://lastminuteengineers.com/drv8825-stepper-motor-driver-arduino-tutorial/>

<https://tttapa.github.io/ESP8266/Chap04%20-%20Microcontroller.html>