

Inhaltsverzeichnis

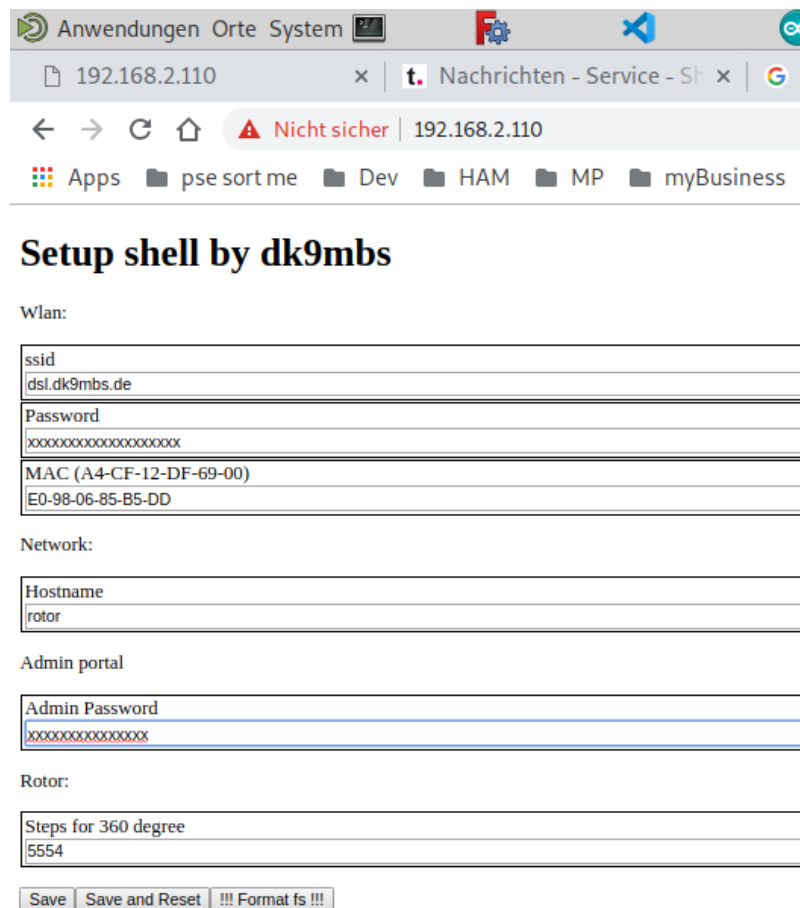
Einrichtung.....	2
Setup.....	2
Protokoll.....	4
Api.....	4
Neue Position setzen.....	4
Aktuelle Position abfragen.....	4
Text auf dem Display ausgeben.....	4
Firmware (DK9MBS-Rotor-OS).....	5
Diagramm.....	5
Quellcode.....	5
Bibliotheken.....	5
Einstellungen Arduino IDE.....	6
Mainboard.....	7
Anschlüsse.....	7
Elektronische Komponenten.....	9
NodeMCU mit ESP8266.....	9
Verfügbare GPIO Pin's.....	9
Verwendung der GPIO Pins.....	10
Pinbelegung.....	10
Motortreiber DRV8825.....	11
Pinbelegung.....	11
Konfiguration der Schrittgröße.....	11
Signal am Step Eingang.....	12
Ausgabe.....	12
LCD Display mit I2C Adapter.....	13
3D Druck.....	14
Explosionszeichnung.....	14
Dateien.....	14
Stückliste.....	15
Elektronik / Mechanik.....	15
3D Teile.....	15
Verbesserungen in den nächsten Versionen.....	16
Verbesserungen.....	16
Quellen.....	17

Einrichtung

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. Im AP Modus ist der Rotor unter der IP Adresse 192.168.4.1 erreichbar.

Setup

In der Setup Oberfläche können alle für den Betrieb des Rotors notwendigen Parameter hinterlegt.



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
xxxxxxxxxxxxxxxx

Rotor:

Steps for 360 degree
5554

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

WLAN	
ssid	Die WLAN ssid ID
Password	WLAN Passwort
MAC	Hier kann die MAC Adresse der ESP WLAN Netzwerkschnittstelle ändern.
Network	
Hostname	Der Hostname unter dem sich der Rotor am DHCP Server registriert.
Admin Portal	
Admin Password	Passwort für das Setup. Dieses Passwort wird auch für den AP Modus verwendet.
Rotor	
Steps for 360 degree	Anzahl Schritte die der Motor für eine komplette Umdrehung benötigt (von Endschalter max. zu Endschalter min.). Durch löschen dieses Eintrags werden beim nächsten Neustart des Rotors die Schritte automatisch neu ermittelt.

Protokoll

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).

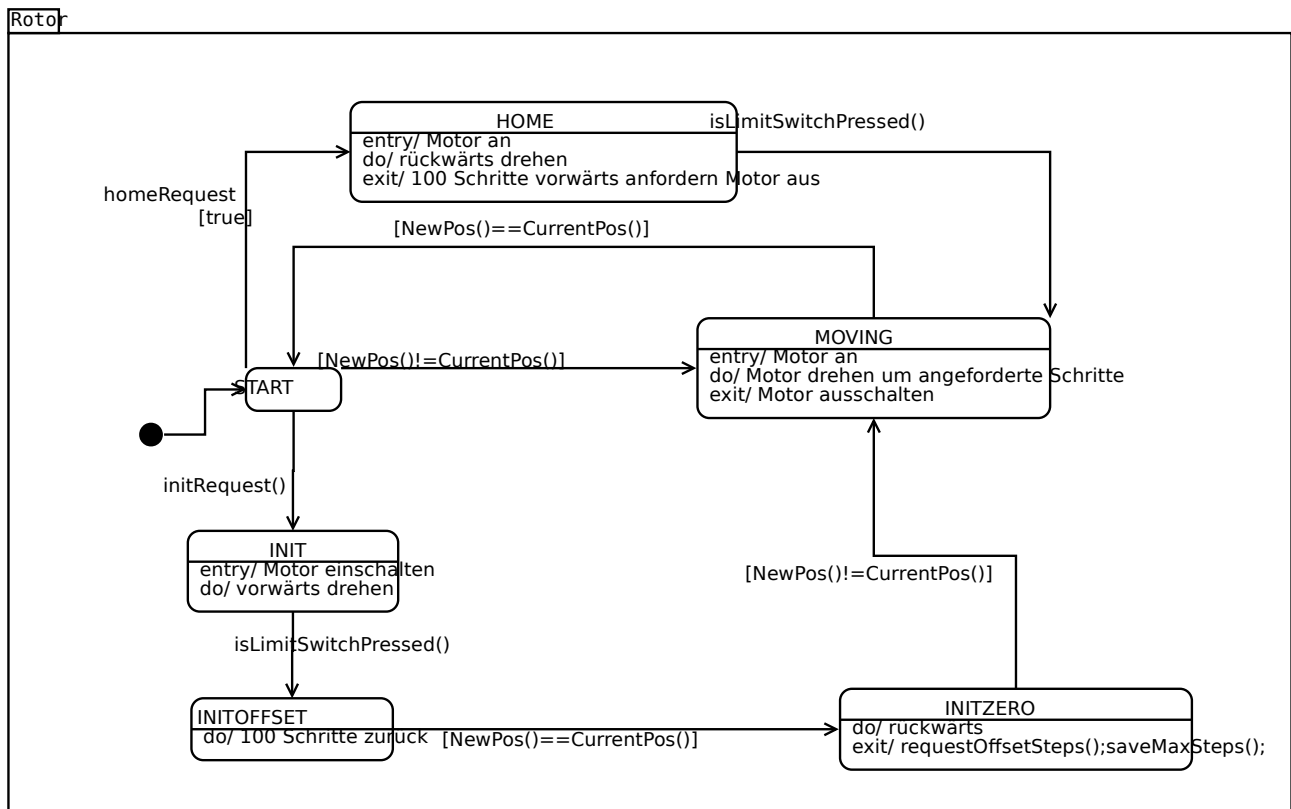
```
wget -O - "http://[IP Adresse des Rotors]/api?command=SETDISPLAY&line0=Hallo"
```

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

Firmware (DK9MBS-Rotor-OS)

Die Architektur der Firmware basiert auf der „finite state machine theory“. In dieser Theorie befindet sich die Firmware immer in einem definierten Zustand. Über Bedingungen kann ein Wechsel des Zustandes erzwungen werden.

Diagramm



Quellcode

Der Quellcode kann unter Github öffentlich eingesehen werden. Die Adresse lautet:

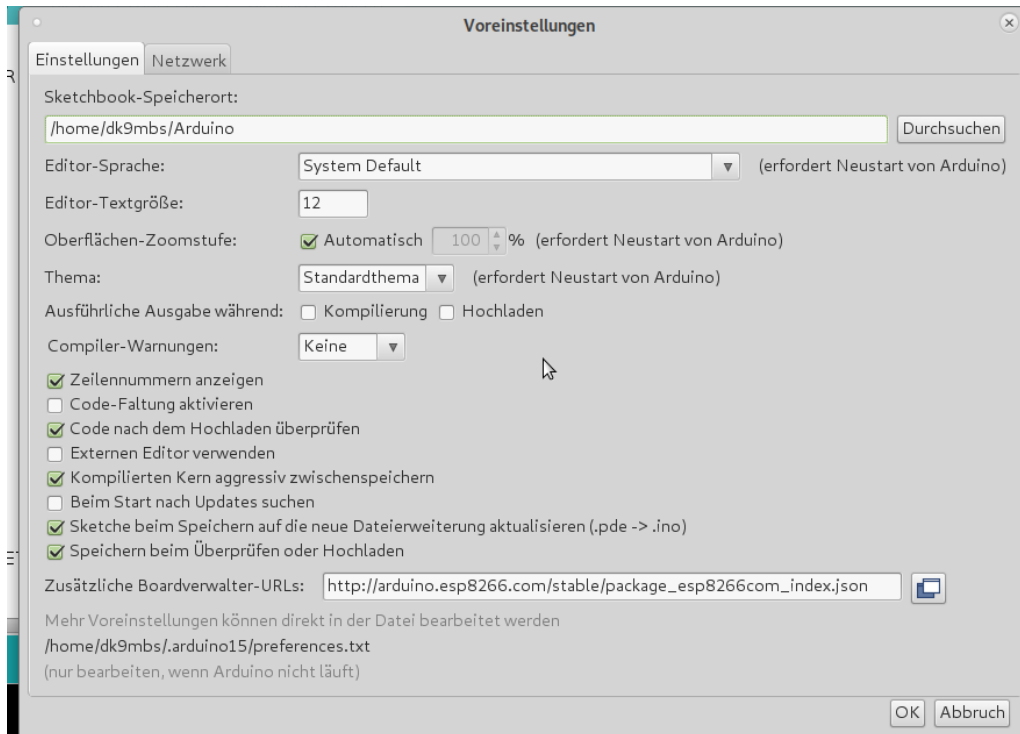
<https://github.com/dk9mbs/Rotor/blob/dev/fw-8266-arduino/fw-8266-arduino.ino>

Bibliotheken

LiquidCrystal_I2C	https://github.com/lucasmaziero/LiquidCrystal_I2C	Manuell installieren

Einstellungen Arduino IDE

Zunächst muss das ESP Board der Arduino IDE bekanntgemacht werden. Dazu muss die URL der Board Definition unter Datei → Voreinstellungen als Zusätzliche Boardverwalter-URL hinterlegt werden [4].



Die zu hinterlegende URL lautet:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

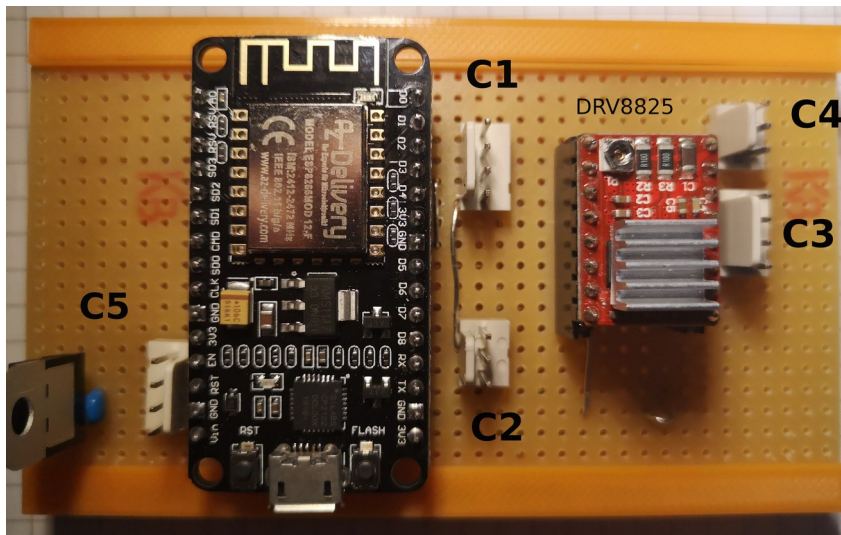
Jetzt kann unter Werkzeuge das richtige Board ausgewählt werden: NodeMCU 0.9 (ESP-12 Module).

Mainboard

Auf dem Rotor Mainboard befinden sich sowohl NodeMCU mit ESP8266, DRV8825 als auch sämtliche Anschlüsse und die Stromversorgung (12V und 5V).

Anschlüsse

Die Anschlüsse sind mit C1 – C5 gekennzeichnet.



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)

Elektronische 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

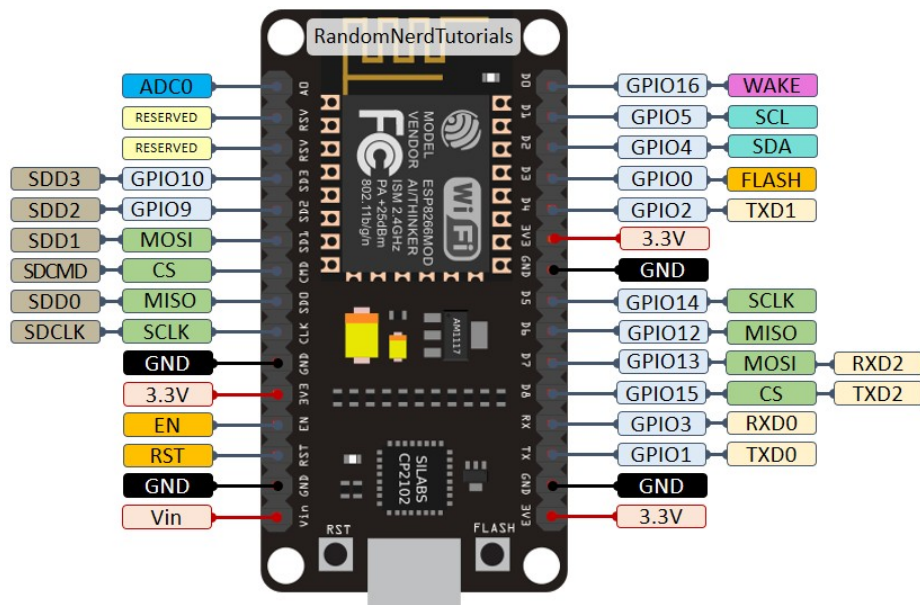
Quelle: [1]

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.

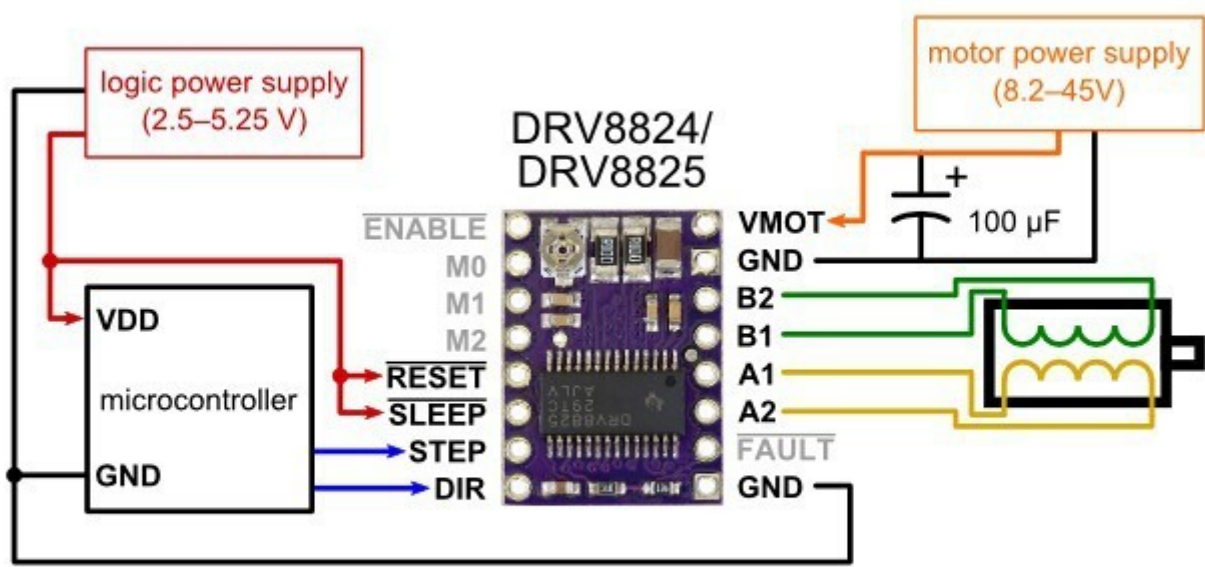


Motortreiber DRV8825

Bei dem DRV8825 handelt es sich um einen intelligenten Schrittmotortreiber. Dieser Treiber benötigt vom Mikrocontroller lediglich Schritt Impulse. Die eigentliche Ansteuerung des Schrittmotors übernimmt der Treiber selbst.

Pinbelegung

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



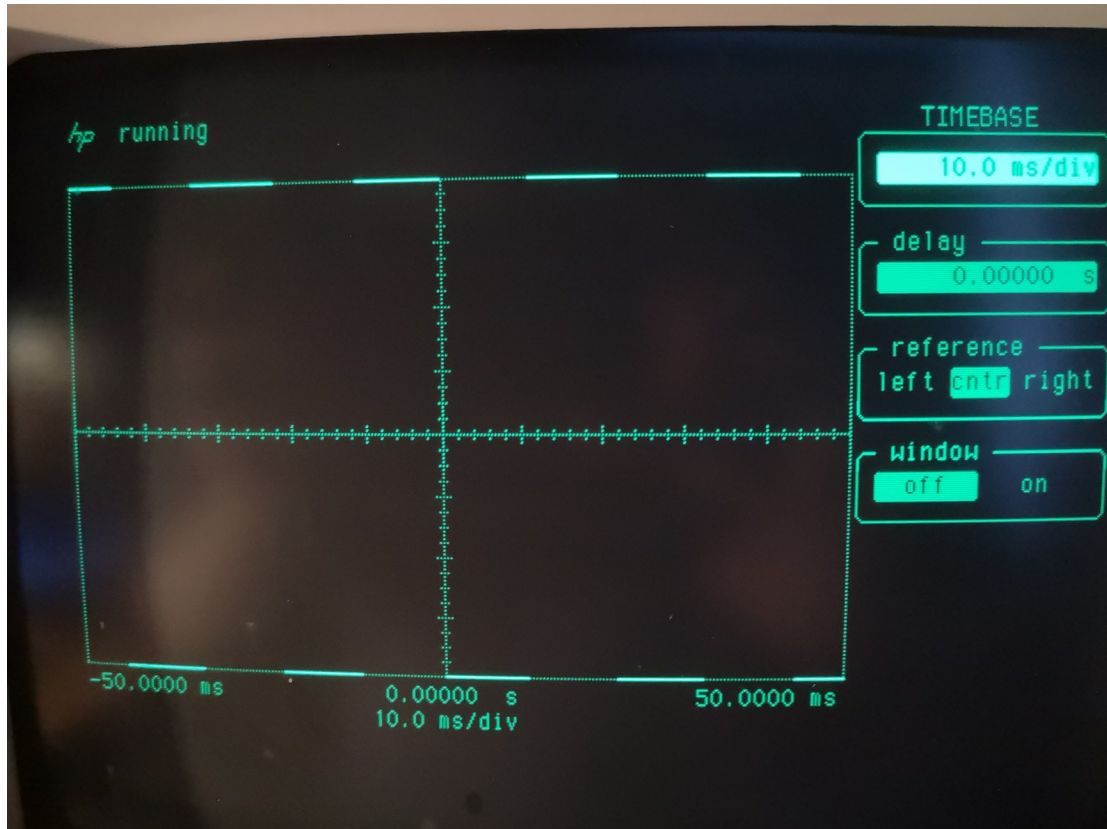
Konfiguration der Schrittgröße

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

Signal am Step Eingang

Am Step Eingang des DRV8825 ist beim Bewegen des Rotors ein 50Hz (20ms Periodendauer) Signal zu erkennen.



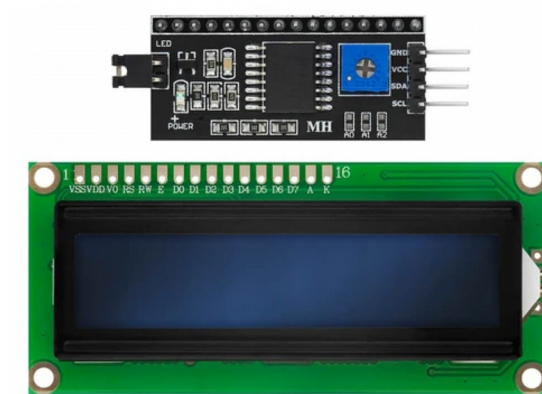
Ausgabe

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

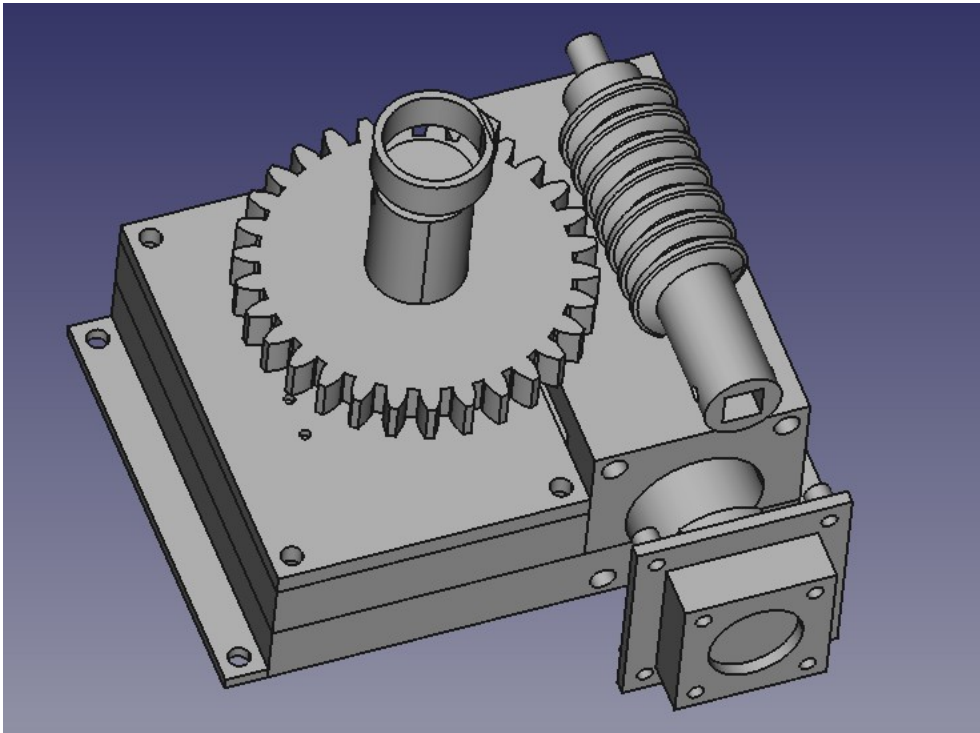


3D Druck

Die Mechanik des Rotors besteht lediglich aus 3D druckbaren Komponenten und einigen wenigen Standardbauteilen die in jedem Baumarkt verfügbar sind.

Explosionszeichnung

Der Rotor besteht aus einzelnen Bauteilen. Als Druckmaterial wird PLA empfohlen.



Dateien

Die Dateien stehen im Freecad Format zur Verfügung. Die Druckdauer beträgt je nach Einstellung und Geschwindigkeit des Druckers ca. 60 Stunden. Die Freecad Dateien können hier runter geladen werden:

<https://github.com/dk9mbs/Rotor/tree/master/3D>

Eine kurze Beschreibung der einzelnen Teile befindet sich in diesem Handbuch im Kapitel „Stückliste“

Stückliste

Elektronik / Mechanik

Menge	Teil
1	DRV8825
1	NodeMCU mit 8266
1	LCD Display 16X2 Zeichen
1	LCD I2C Adapter
1	Kugellager 47mm Außendurchmesser
1	Kugellager 22mm Außendurchmesser
1	Endschalter (Taster Schließer)
1	M3 Gewindestange
4	M4 Schrauben
4	M4 Muttern
8	M3 Muttern
8	M3 Unterlegscheiben

3D Teile

Dateiname	Baugruppe
Rotor_Antrieb.FCStd	Zahnrad mit Antriebsachse
Rotor_Endabschalter.FCStd	Endschalter Ring
Rotor_Gehäuse.FCStd	Gehäuse mit Deckel und Nema Befestigung
Rotor_Schnecke_9mm.FCStd	Antriebsschnecke
Rotor_Schnecke_Nema14_Adapter.FCStd	Adapter Motor - Schnecke

Verbesserungen in den nächsten Versionen

In einer nächsten Versionen sind Verbesserungen sowohl an der Firmware, der Elektronik und der Mechanik geplant.

Verbesserungen

Problem	Lösung	Prio (1-9)
Update	Over the Air Updates	9
Fehlende manuelle Bedienung (keine freien GPIO Pin's)	Anschluss I2C Port Expander	1
Laute Motorgeräusche	Dämpfung zwischen Motor und Gehäuse	2
Drehmoment nicht bekannt	Messen mit Kofferwaage	1
2. Rotor		

Quellen

- [1] <https://lastminuteengineers.com/drv8825-stepper-motor-driver-arduino-tutorial/>
- [2] <https://tttapa.github.io/ESP8266/Chap04%20-%20Microcontroller.html>
- [3] https://de.wikipedia.org/wiki/Endlicher_Automat
- [4] <https://m.heise.de/make/artikel/ESP-Boards-mit-der-Arduino-IDE-programmieren-4130861.html?seite=all>
- [5] <https://github.com/dk9mbs/Rotor/tree/master/3D>