



University of Applied Sciences

HOCHSCHULE  
EMDEN-LEER

## *Service- und Entwicklerdokumentation*

---

### Demonstrator für einen Schrittmotor

Autor:	Maik Theilmann
Matrikel-Nr.:	7021861
Autor:	Jannik Grönke
Matrikel-Nr.:	7022006
Autor:	Enno Stein
Matrikel-Nr.:	7021960
Autor:	Lars Hanneken
Matrikel-Nr.:	7021910
Autor:	Jan ter Veen
Matrikel-Nr.:	7022071
Studiengang:	Maschinenbau und Design
Erstprüfer:	Prof. Dr. Elmar Wings
Abgabedatum:	27. Mai 2024



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>i</b>
<b>Abbildungsverzeichnis</b>	<b>v</b>
<b>Tabellenverzeichnis</b>	<b>vii</b>
<b>Liste des Listings</b>	<b>ix</b>
<b>Acronyms</b>	<b>xi</b>
<b>1 Projektbeschreibung</b>	<b>1</b>
1.1 Aufgabenstellung . . . . .	1
1.2 Herausforderungen . . . . .	1
1.3 Lösungsansätze . . . . .	1
<b>2 Hardwarebeschreibung des Arduino</b>	<b>3</b>
2.1 Aufbau des Arduinos . . . . .	3
2.2 Integrierte Sensorik . . . . .	4
2.2.1 9-Achs-IMU für die Bewegungserkennung (LSM9DS1)	4
2.2.2 Näherungs-,Umgebungslicht-, Farb- und Gestensen-	
sor (APDS9960) . . . . .	4
2.2.3 Barometrischer Drucksensor (LPS22HB) . . . . .	5
2.2.4 Digitales Mikrophon (MP34DT05) . . . . .	5
2.2.5 DC-DC-Wandler (MPM3610) . . . . .	6
2.3 Beschreibung der Schnittstellen . . . . .	6
2.3.1 I2C . . . . .	6
2.3.2 USB: . . . . .	7
2.3.3 Bluetooth®5 . . . . .	7
2.3.4 Weitere Kommunikationsschnittstellen . . . . .	8
2.3.5 Digitale Ein- und Ausgangspins . . . . .	9
2.3.6 Analoge Eingangspins . . . . .	9
2.3.7 Weitere Pins . . . . .	9
2.3.8 LED-Lampen . . . . .	9
2.4 Schaltplan . . . . .	10
2.5 Hinweis Arduino 33 BLE Sense Lite . . . . .	10
2.6 Bezugsquellen . . . . .	10

<b>3</b>	<b>Weitere Hardware</b>	<b>11</b>
3.1	OLED-Display . . . . .	11
3.2	Signalleuchte . . . . .	12
3.3	Spannungswandler . . . . .	12
3.3.1	Netzteil SNT RD 50A . . . . .	12
3.3.2	Spannungswandler ASM1117 . . . . .	12
3.4	Linearführung . . . . .	13
3.5	Drehwinkel-Encoder . . . . .	13
3.6	Schrittmotorsteuerung . . . . .	13
<b>4</b>	<b>Schaltplan</b>	<b>15</b>
<b>5</b>	<b>Materialliste</b>	<b>17</b>
<b>6</b>	<b>Beschreibung der Software IDE</b>	<b>21</b>
6.1	Installation der Arduino IDE . . . . .	21
6.2	Beschreibung der Entwicklungsumgebung . . . . .	21
6.3	Erste Schritte in der Entwicklungsumgebung . . . . .	22
6.3.1	Auswahl des Mikrocontrollers . . . . .	22
6.3.2	Bibliotheken einbinden . . . . .	23
6.4	Programmierung . . . . .	25
6.4.1	header . . . . .	25
6.4.2	setup() . . . . .	25
6.4.3	loop() . . . . .	25
6.5	Erster Programmtest . . . . .	25
<b>7</b>	<b>Beschreibung des Programms auf dem Arduino</b>	<b>27</b>
<b>8</b>	<b>Beschreibung des Schrittmotors</b>	<b>29</b>
8.1	Beschreibung eines Schrittmotors . . . . .	29
8.1.1	Aufbau und Funktionsweise eines Schrittmotors . . . . .	29
8.1.2	Schrittmotor Bauformen . . . . .	30
8.1.3	Betriebsarten unipolar und bipolar . . . . .	31
8.2	Beschreibung des verwendeten Schrittmotors . . . . .	31
<b>9</b>	<b>Schrittverluste Verhindern</b>	<b>33</b>
9.1	Auswahl des Schrittmotors . . . . .	33
9.2	Betriebsart . . . . .	33
9.2.1	Start-Stopp-Betrieb . . . . .	33
9.2.2	Beschleunigung und Rampenprofil (Trapezförmig) . . . . .	34
9.3	Externe Ereignisse . . . . .	36
9.3.1	Lastrückkopplung . . . . .	36
9.3.2	Erhöhung der Nutzlast mit der Zeit . . . . .	36
<b>10</b>	<b>Testdurchläufe</b>	<b>39</b>

Inhaltsverzeichnis	iii
<b>11 Offene Punkte</b>	<b>41</b>
<b>12 Literatur</b>	<b>43</b>



# Abbildungsverzeichnis

2.1	Arduino Nano 33 BLE Sense . . . . .	3
2.2	Arduino Nano Pinboard . . . . .	7
4.1	Schaltplan . . . . .	15
6.1	Hauptoberfläche in der Arduino IDE . . . . .	22
6.2	Installation des Boards . . . . .	23
6.3	Herunterladen von Bibliotheken . . . . .	24
9.1	Start-Stopp Frequenz [ <b>FaulhaberDriveSystems.2020</b> ] .	34
9.2	Trapezförmiges Geschwindigkeitsprofil [ <b>FaulhaberDriveSystems.2020</b> ]	35





# Tabellenverzeichnis

5.1	Materialliste . . . . .	19
9.1	Ursachen und Lösungen: Motor läuft nicht an [ <b>FaulhaberDriveSystems.2020</b> ] . . . . .	34
9.2	Ursachen und Lösungen: Motor beendet die Beschleunigungsrampe nicht [ <b>FaulhaberDriveSystems.2020</b> ] . . . . .	35
9.3	Ursachen und Lösungen: Motor beschleunigt bis zur Enddrehzahl und bleibt stehen, sobald eine konstante Drehzahl erreicht ist. [ <b>FaulhaberDriveSystems.2020</b> ] . . . . .	36



# Liste des Listings



# Abkürzungen



# 1 Projektbeschreibung

## 1.1 Aufgabenstellung

Die Aufgabe besteht darin, mithilfe eines Arduino Nano 33 BLE Sense Lite einen Demonstrator für einen Schrittmotor zu entwickeln. Mittels einer Konstruktion, die über einen Schrittmotor verfügt, welcher einen Riemen antreibt, soll ein Schlitten auf einer Linearführung verfahren werden. Es sollen unterschiedliche Bewegungscharakteristiken demonstriert werden. Durch die Programmierung des Arduino, wird der Schrittmotor gesteuert und wird in die gewünschte Richtung und Position bewegt. Dies beinhaltet eine korrekte Programmierung, sowie die korrekte Ausgabe durch den Schrittmotor und eine Konstruktion, die mit allen Teilen funktionsfähig ist. Nachfolgend wird die Hardware beschrieben, sowie die Entwicklung der Software und die Konstruktion und Fertigung des Demonstrators.

## 1.2 Herausforderungen

Das Projekt lässt sich in 3 verschiedene Teilprojekte unterteilen, die zum Abschluss der Aufgabe führen. Zuerst ist da die Programmierung und die Arbeit mit der Arduino IDE. Außerdem die Konstruktion des Demonstrators und zuletzt die hierfür benötigte Auswahl der Hardware-Teile. Ebenfalls wichtig ist der Umgang mit den elektronischen Bauteilen, dass diese nicht durch den elektrischen Strom beschädigt werden und das der Demonstrator für den Transport geeignet ist.

## 1.3 Lösungsansätze

Durch die Ausarbeitung der Herausforderungen, können Lösungsansätze hierfür entwickelt werden. Indem vorab eine Skizze für die Konstruktion angefertigt wird, kann darauf basierend die Suche nach Teilen durchgeführt werden. Um den Demonstrator mit all seinen Teilen und seiner unhandlichen Größe trotzdem transportfähig zu konstruieren, werden die elektronischen Bauteile mit auf dem Alu-Profil montiert. Außerdem spielt es für die Programmierung eine entscheidende Rolle, wie die verschiedenen Stufen angefahren werden sollen. Hierfür wurde sich von vornherein überlegt, wie die Stufen auszuwählen sind und in welche Positionen und in welchen Geschwindigkeiten verfahren wird.





## 2 Hardwarebeschreibung des Arduino

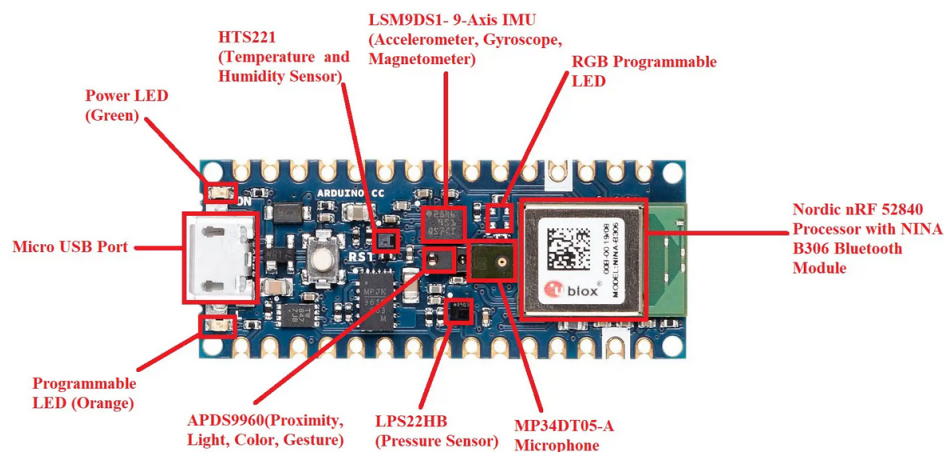


Abbildung 2.1: Arduino Nano 33 BLE Sense  
[eTech.2021]

### 2.1 Aufbau des Arduinos

[Ard24a]

Der Arduino Nano 33 Bluetooth Low Energie (BLE) Sense Lite ist ein Mikrocontroller basierend auf dem Nordic nRF52840-SoC (System-on-a-Chip). Trotz seiner kompakten Bauweise mit einer Größe von 45 x 18 mm verfügt es über mehrere verschiedene integrierte Sensoren, Aktoren und konnektive Schnittstellen (siehe Abbildung 2.1). Zudem hat der Arduino-Prozessor einen 1 MB Flash-Speicher und 256 KB RAM. Bei dem Prozessor handelt es sich um einen Arm®Cortex-M4F Prozessorkern, welcher mit einer Taktrate von 64 MHz arbeitet und für Mikrocontroller Anwendungen optimiert ist. Dieser bietet eine gute Leistung bei geringen Stromverbrauch und ist auch für Echtzeitverarbeitungsaufgaben geeignet. Außerdem besitzt der Prozessor eine Floating Point Unit (FPU), wodurch sich Datenverarbeitungsoperationen effizient verarbeiten lassen.[Arm]

Mithilfe einer integrierten Steckverbindung (engl. „header“) kann der Mikrocontroller direkt auf ein Board gesteckt werden.

## 2.2 Integrierte Sensorik

Der Arduino Nano 33 BLE-Sense Lite verfügt über eine Vielzahl von integrierten Sensoren und Aktoren mit denen verschiedene Umwelteigenschaften detektiert werden können. Dazu gehören die in den folgenden Abschnitten beschriebenen Sensoren und Aktoren.

### 2.2.1 9-Achs-IMU für die Bewegungserkennung (LSM9DS1)

Die Trägheitsmesseinheit LSM9DS1 ist ein System-in-Package, d.h. auf engstem Raum werden mehrere elektronische Komponenten oder Chips in einem Paket miteinander kombiniert.[Wag13] Die IMU verfügt über einen 3D-Linearbeschleunigungsmesser, ein 3D-Gyroskop und einen 3D-Magnetometer. Außerdem beinhaltet das System eine serielle I2C (Inter-Integrated-Circuit)-Bus Schnittstelle, die einen Standard und einen Fast Mode (100/400 kHz) bereitstellt, zudem eine serielle SPI-Standardschnittstelle. Der Sensor hat einen linearen Beschleunigungsmesser (Accelerometer) mit wählbarer Skala von  $\pm 2/\pm 4/\pm 8/\pm 16$  g, es misst die lineare Beschleunigung in drei Achsen (x, y, z) und ermöglicht die Erfassung von Änderungen der Geschwindigkeit und Position. Das Magnetfeld ist mit einer wählbaren Skala von  $\pm 4/\pm 8/\pm 12/\pm 16$  Gs ausgestattet, damit misst es das magnetische Feld in den drei Achsen und ermöglicht die Bestimmung der Ausrichtung relativ zur Erdmagnetfeldrichtung. Das 3D-Gyroskop ist mit wählbarem Skalenendwert:  $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000 \frac{\text{Grad}}{\text{s}}$  und ist dafür zuständig, die Winkelgeschwindigkeit bzw. Drehbewegung um die drei Achsen zu erfassen.[STM15][Ard24a] Mögliche Anwendungsbereiche sind zum Beispiel eine Bewegungssteuerungen (Drohnensteuerung, Robotik und industrielle Automatisierung), Schwingungsüberwachung und -kompensation, Antennen, Plattformen, optische Bild- und Objektivstabilisierung.

### 2.2.2 Näherungs-,Umgebungslicht-, Farb- und Gestensensor (APDS9960)

Hierbei handelt sich um einen sehr vielseitigen Sensor. Er dient zur Gestenerkennung, Farberkennung, Abstandsmessung und Umgebungslichtmessung. Die Gestenerkennung nutzt vier gerichtete Fotodioden, um reflektierte Infrarot-Energie (die von der integrierten LED stammt) zu erfassen, um die physischen Bewegungsinformationen (d.h. Geschwindigkeit,

Richtung und Entfernung) in digitale Informationen zu übersetzen. Die Näherungserkennung ermöglicht die Messung der Entfernung durch die Erkennung der reflektierten Infrarot-Energie, (von der integrierten LED) mit Hilfe einer Fotodiode. Erkennungsereignisse sind interruptgesteuert und treten immer dann auf, wenn das Näherungsergebnis die oberen und/oder unteren Schwellenwerte überschreitet. Der Sensor verfügt zudem über Offset-Einstellregister, um den System-Offset zu kompensieren, der durch unerwünschte Infrarot-Energier reflexionen am Sensor entsteht. Die Infrarot-LED-Intensität ist werksseitig eingestellt, so dass eine Kalibrierung der Endgeräte aufgrund von Bauteilschwankungen nicht erforderlich ist. Die Farb- und Umgebungslichterkennung liefert Daten zur roten, grünen, blauen Farbspektrum und Daten zum klaren Lichtintensität. Jeder der Kanäle R, G, B, C hat einen UV- und Infrarot-Sperrfilter und einen speziellen Datenkonverter, der gleichzeitig 16-Bit-Daten erzeugt. Diese Architektur ermöglicht Anwendungen eine genaue Messung des Umgebungslichts zu messen und die Farbe zu erkennen, was es den Geräten wiederum ermöglicht, die Farbtemperatur zu berechnen und die Hintergrundbeleuchtung des Displays dementsprechend anzupassen.[Ava15][Ard24a]

### 2.2.3 Barometrischer Drucksensor (LPS22HB)

Der LPS22HB ist ein sehr kompakter Absolutdrucksensor, der auf dem piezoresistiven Prinzip basiert. Er fungiert als Barometer und verfügt über einen digitalen Ausgang. Zudem ist in diesem Drucksensor ein Temperatursensor integriert mit welchem Druckmessungen zusätzlich kompensiert werden können. Das Gerät besteht aus einem Sensorelement und einer Integrated-Circuit (IC)-Schnittstelle, die eine Kommunikation zwischen der Sensoreinheit und der Anwendung über Inter-Integrated Circuit (I2C) oder Serial Peripheral Interface (SPI) ermöglicht. Das Sensorelement erfasst den absoluten Druck und besteht aus einer speziell hergestellten, aufgehängten Membran. Der LPS22HB ist in einem vollständig vergossenen Land-Grid-Array-Gehäuse untergebracht, das kleine Löcher aufweist. Durch diese Öffnung gelangt der externe Druck auf das Sensorelement. Der Betrieb des Sensors ist über einen Temperaturbereich von  $-40^{\circ}\text{C}$  bis  $+85^{\circ}\text{C}$  gewährleistet. [STM17][Ard24a] Anwendungsbereiche für diesen Sensor sind zum Beispiel: Wetterstationen, Höhenmesser, Luftdrucküberwachung in industriellen Prozessen oder tragbare smarte Geräte, wie Sportuhren oder Smartphones.

### 2.2.4 Digitales Mikrofon (MP34DT05)

Das MP34DT05-A ist ein kompaktes, stromsparendes, omnidirektionales, digitales Mikrofon mit einem kapazitiven Sensorelement und einer IC-Schnittstelle. Das Sensorelement, das in der Lage ist, akustische Wellen

zu detektieren, wird mit einem speziellen Silizium-Mikrobearbeitungsverfahren für die Herstellung von Audiosensoren hergestellt. Die IC-Schnittstelle wird in einem speziellen Halbleiterherstellungsverfahren (CMOS-Verfahren) hergestellt, das die Entwicklung einer speziellen Schaltung ermöglicht, die ein digitales Signal im PDM (Pulsdichtenmodulation) -Format extern bereitstellen kann. Das Mikrofon hat einen Signal-Rausch-Abstand von 64 dB und eine Empfindlichkeit von  $-26 \text{ dBFS} \pm 3 \text{ dB}$ . Sein maximaler Schalldruckpegel liegt bei 122,5 dB SPL. Der MP34DT05-A ist in einem SMD-kompatiblen, EMI-geschirmten Top-Port-Gehäuse erhältlich und arbeitet zuverlässig über einen erweiterten Temperaturbereich von  $-40 \text{ }^{\circ}\text{C}$  bis  $+85 \text{ }^{\circ}\text{C}$ . Abmessungen des Gehäuses HCLGA-4 LD sind  $3 \times 4 \times 1$  (in mm). Anwendung findet das Modul beispielsweise in mobilen Endgeräten, im Bereich der Spracherkennung sowie in Laptops und Notebooks.[STM21][Ard24a]

### 2.2.5 DC-DC-Wandler (MPM3610)

Das MPM3610-Modul ist ein integriert Gleichspannungs- zu Gleichspannungs-Wandler. Dieser kann Eingangsspannungen von bis zu 21 V verarbeiten mit einem Wirkungsgrad von mindestens 65% bei minimaler Last. Bei einer Eingangsspannung erhöht sich der Wirkungsgrad auf 85%.[Ard24a]

## 2.3 Beschreibung der Schnittstellen

Der Arduino Nano 33 BLE-Sense Lite bietet eine Vielzahl von Schnittstellen, die das Board mit anderen Komponenten und Geräten einfach verbinden lassen. In diesem Abschnitt sind einige Details zu den wichtigsten Schnittstellen des Boards erläutert.

### 2.3.1 I2C

Der Inter-Integrated-Circuit (I2C) -Bus ermöglicht die Vernetzung von Mikrocontrollern, Sensoren, LED-Bildschirmen, Echtzeituhren und vielen anderen Bausteinen. Das Kommunikationsprotokoll basiert auf dem Master-Slave-Prinzip. Der Master initiiert und beendet die Kommunikation, stellt den Takt zur Synchronisation bereit und löst Kommunikationsprobleme. Jeder Slave hat eine eindeutige Adresse, mit einer Länge von 7 Bit und ermöglicht die Adressierung von bis zu 128 Slaves. Geräte mit unterschiedlichen Adresslängen können im selben Bus koexistieren. Bevor der Master Daten überträgt oder empfängt, muss er einen Slave mit einer vorher vereinbarten Adresse ansprechen.[MS18][STM15] Sind ihre Adressen in Übereinstimmung kann im nächsten Schritt ein einzelnes Bit gesendet werden, mit dem dann festgelegt wird, ob der Master

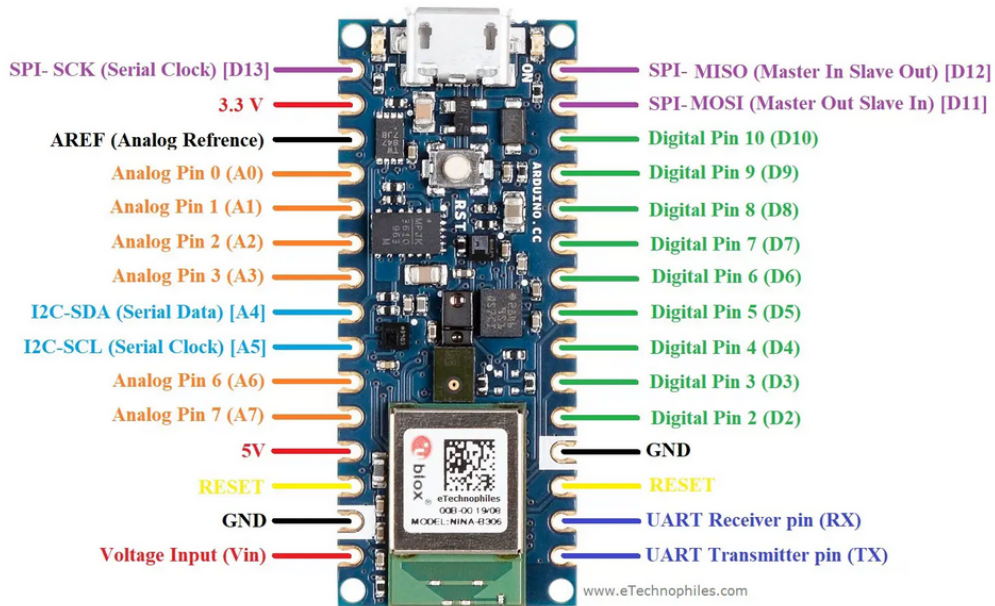


Abbildung 2.2: Arduino Nano Pinboard

[Arduino2023]

Daten an dem Slave übertragen (Binär:1) oder auslesen möchte (Binär:0). Dieser Datenaustausch wird danach bestätigt, sodass weitere Daten ausgetauscht werden können. Zur Beendigung des Datenaustausches wird ein Stop-Signal gesendet.[GW22] In der minimalen Konfiguration werden Master und Slave über die bidirektionalen Busleitungen SDA und SCL verbunden, die über Pull-up-Widerstände an die Versorgungsspannung angeschlossen sind. Weitere Geräte können durch Verbindung ihrer SDA- und SCL-Anschlüsse mit den entsprechenden Busleitungen an den Bus angeschlossen werden.[MS18] In diesem Projekt dient der Arduino als Master und ein zusätzlich angeschlossenes OLED-Bildschirm als Slave.

### 2.3.2 USB:

Das Board kann über einen Micro-Universal-Serial-Bus (Micro-USB) mit einem Computer verbunden werden, um es zu programmieren oder Daten zu übertragen. Die Datenübertragungsrate beträgt dabei 12 Mbit/s. Außerdem kann der Arduino über diesen USB-Port Strom beziehen.

### 2.3.3 Bluetooth®5

Die Bluetooth-Verbindung kann als drahtloses Kommunikationsweg eingesetzt werden. Dieses Bluetooth-Protokoll hat eine Übertragungsrate

von 2 Megabit pro Sekunde (Mbps) und eine Sendeleistung von +8 Dezibel Milliwatt (dBm). Die Empfindlichkeit beträgt dabei -95 dBm. Des weiteren verbraucht diese Verbindung im Sendebetrieb 4,8 mA und 4,6 mA im Empfangsbetrieb. Das Bluetooth Modul ist kompatibel mit mehreren Protokollen unter anderem mit dem *Thread-Protokoll* und dem *Zigbee-Protokoll*. [Ard24a][NrdSem3.2024]

### 2.3.4 Weitere Kommunikationsschnittstellen

[Ard24a][NrdSem3.2024]

- **NFC-A-Tag:** Near Field Communication (NFC) ist eine zusätzliche Funktion zur drahtlosen Kommunikation über kurze Distanzen. Zudem besitzt der NFC-A-Tag die Funktionen sich in einen Bereitschaftsmodus versetzen zu lassen, dass durch ein NFC-fähiges Gerät dann initiiert werden kann. Außerdem unterstützt es *touch-to-pair*, diese Funktion ermöglicht eine Kopplung mit anderen NFC-fähigen Geräten durch Berührung.
- **Arm CryptoCell CC310 Security Subsystem:** Für die Durchführung kryptografischer Operationen und Sicherheitsaufgaben. [Nor24b]
- **QSPI/SPI/TWI/I<sup>2</sup>S/PDM/QDEC:** Verschiedene weitere serielle Kommunikationsschnittstellen, die für den Datenaustausch verwendet werden können.
- **EasyDMA:** Direkt Memory Access (DMA) ist für die Übertragung von Daten zwischen verschiedenen Speicherbereichen, ohne dabei die CPU zu belasten. [GW22]
- **Analog-Digital-Wandler (ADC):** Wandelt analoge Eingangssignale in digitale Daten um. Der Wandler hat eine Auflösung von 12 Bit und eine maximale Abtastrate von 200 Kilosamples pro Sekunde (ksps).
- **128-Bit-AES/ECB/CCM/AAR-Co-Prozessor:** Co-Prozessor für kryptografische Operationen, der auf dem Advanced Encryption Standard (AES) basiert. Dieser unterstützt verschiedene Betriebsmodi wie Electronic Codebook (ECB), Counter with CBC-MAC (CCM) und Automatic Address Recognition (AAR). [Nor24a]
- **Quad-SPI-Schnittstelle 32 MHz:** SPI-Schnittstelle, die eine maximale Taktrate von 32 MHz unterstützt. Quad-SPI ermöglicht es, Daten schneller als die herkömmliches SPI zu übertragen, indem es vier Datenleitungen verwendet. [Nor23]

### 2.3.5 Digitale Ein- und Ausgangspins

Das Board verfügt über 14 digitale Ein- und Ausgangspins. Die digitalen Pins können nur zwei Zustände, nach dem Binär-System lesen: wenn ein Spannungssignal vorliegt und wenn kein Signal vorhanden ist (0 oder 1). Einige der Pins sind zudem zur Pulsweitenmodulation fähig (D3, D5, D6, D9, D10). Außerdem sind die digitalen Pins D11 und D12 als Master-Output-Slave-Input (MOSI) und als Master-Input-Slave-Output (MISO), in einer Serial-Peripheral-Interface (SPI) Kommunikation einsetzbar.[Ard24a]

### 2.3.6 Analoge Eingangspins

Die Platine hat zusätzlich 8 analoge Eingangspins (A0-A7) die wiederum als Analog-Digital-Wandler (ADC) verwendet werden können. Außerdem sind diese Pins als digitale Ein-/Ausgangspins konfigurierbar. Der Pin A0 kann zudem als Digital-Analog-Wandler (DAC) konfiguriert werden. Die beiden Pins A4 und A5 können außerdem für die I2C-Kommunikation verwendet werden. Dabei fungiert A4 als Datenleitung (SDA), während A5 als Taktleitung (SCL) fungiert.[Ard24a]

### 2.3.7 Weitere Pins

- **+3,3 V:** Erzeugt interne Stromquelle im Gerät und wird als Referenzspannung verwendet.
- **VIN:** Stromversorgung
- **5V:** Gibt 5V an die externen Komponenten ab.
- **RST-Pin:** Dient zum Zurücksetzen des Arduinos.
- **AREF-Pin:** Liefert die Spannungsreferenz, die der Mikrocontroller zur Zeit verwendet.  
[Ard24a]

### 2.3.8 LED-Lampen

Im Arduino selbst sind 3 LED's verbaut, die auch alle programmiert werden können. Diese sind vor allem für die Überprüfung der Sensorik oder Softwareprogrammen nützlich. Zu den LED-Lampen gehören:

- Programmierbare Power-LED (grün): Zeigt an, dass das Arduino-Board eingeschaltet ist.
- Programmierbare LED (orange)

- Programmierbare RGB-LED  
[Ard24a]

## 2.4 Schaltplan

fehlt noch...

## 2.5 Hinweis Arduino 33 BLE Sense Lite

Der Arduino Nano 33 BLE Sense Lite ist eine komprimierte Variante vom ursprünglichen Arduino Nano 33 BLE Sense, welcher zusätzlich noch über einen Temperatur- und Feuchtigkeitssensor verfügt. Der Lite hat stattdessen einen Drucksensor integriert, über welchem auch die Temperatur gemessen werden kann, jedoch nicht die Feuchtigkeit.[PA22]

## 2.6 Bezugsquellen

Als Bezugsquellen dienen vor allem Datenblätter der Hersteller. Die meisten Informationen konnten dem Datenblatt des Arduino entnommen werden, jedoch ist dazu anzumerken, dass sich dieses Datenblatt auf das Arduino 33 BLE Sense bezieht und nicht auf das Arduino 33 BLE Sense *Lite*. Speziell zum *Lite* gibt es jedoch kein Datenblatt, so wurde das Datenblatt vom Arduino 33 BLE Sense herangezogen. Dies stellt sonst kein Problem dar, da sich, wie in 2.5 beschrieben, nur um eine komprimierte Version des Arduino 33 BLE Sense handelt.



## 3 Weitere Hardware

In den nachfolgenden Kapiteln folgt eine Aufzählung und Erläuterung der wesentlichen, zusätzlich benötigten Hardware. Dazu zählen Aktoren zur Darstellung des Betriebszustandes, Schalter und weitere Komponenten, die für den Aufbau des Schrittmotor-Demonstrators nötig waren.

### 3.1 OLED-Display

Zur Feststellung des aktuellen Betriebszustandes und zur Ausgabe des eingestellten Bewegungsstufe des Schlittens dient ein 1,3 Zoll OLED-Display (*Organic Light Emitting Diode*). Bei OLED-Displays sind jeweils drei OLEDs für einen Pixel zuständig und benötigt keine Hinterleuchtungen oder LC-Zellen, dadurch ist eine scharfe Kontrastierung möglich. Die OLED-Displays zeichnen sich durch die Reaktionsschnelligkeit aus. Die Helligkeit eines Pixels wechselt in weniger als einer Mikrosekunde.[Die19] Insgesamt ist das Modul  $36\text{ mm} \times 34\text{ mm} \times 36\text{ mm}$  groß mit einer Bildschirmdiagonalen von 1,3 Zoll. Das Display besteht aus  $128 \times 64$  weißen OLED-Bildpunkten. Durch die I2C-Kommunikation kann das Display mit dem Arduino verbunden werden. Zur Stromversorgung wird 3,3 V Gleichstromspannung, bei einem Stromverbrauch von weniger als 11 mA, benötigt. Mithilfe von 4 Steckbrückenkabel ist es mit dem Arduino verbunden, um es mit der nötigen Spannung zu versorgen und die I2C-Verbindung herzustellen (Verdrahtung detailliert in Kapitel 4 Schaltplan). Die zur I2C-Kommunikation nötige Adresse ist 0x3F. Zusätzliche Spezifikationen des Displays sind:

- Pixelgröße:  $0.21\text{ mm} \times 0,21\text{ mm}$
- Pixelabstand:  $0.23\text{ mm} \times 0,23\text{ mm}$
- Anzeigemodus: Passive Matrix
- Pixelfarbe: Weiß

[AD24]

## 3.2 Signalleuchte

Zusätzlich zum OLED-Display ist eine SMD-LED-Signalleuchte zur Zustandserkennung verbaut. Sie soll Störungen und Fehler erkenntlich machen. SMD steht für *Surface Mount Device* und bedeutet, dass die Signalleuchte für Oberflächenmontagen konzipiert ist. Die Leuchte ist dabei mit einem einfachen Stecksystem am Gehäuse befestigt. Dazu wird sie in einer 8 mm Montagebohrung des Gehäuses gesteckt und über die 10 mm Gehäuseblende der Signalleuchte fixiert. Die Leuchte hat eine 3 mm im Durchmesser große LED und emittiert ein rotes Licht. Die Betriebsspannung liegt zwischen 2 bis 2,4 V und der Betriebsstrom bei 20 mA. Angeschlossen wird sie über vier Litzen am Arduino (verweis auf Schaltplan).[Men24]

## 3.3 Spannungswandler

Für den Aufbau des Demonstrators sind mehrere Spannungswandler nötig. Folgende Wandler wurden in der Schaltung verwendet.

### 3.3.1 Netzteil SNT RD 50A

Dieser Wechselspannung-/Gleichspannungswandler (AC/DC-Wandler) wandelt die 230 V Wechselspannung des Netzanschlusses in 12 V Gleichspannung um. Weitere Spezifikationen des Wandlers sind:

- **Bauteilabmessung** ( $L \times B \times H$ ): 99 mm  $\times$  97 mm  $\times$  36 mm
- **Ausgangsstrom**: 2 A
- **Leistung**: 54 W
- **Wirkungsgrad**: 79 %

[Mea19]

### 3.3.2 Spannungswandler ASM1117

Dieser Spannungswandler wandelt die 12 V vom Netzteil SNT RD 50A in jeweils 3,3 V und 5 V um. Die 3,3 V werden für den Arduino, Schrittmotorsteuerung, LC-Display und dem Drehwinkel-Encoder verwendet. Weitere Technische Daten des Wandlers sind:

- **Bauteilabmessung** ( $L \times B \times H$ ): 40 mm  $\times$  40 mm  $\times$  20 mm
- **Ausgangsstrom**: 800 mA

[Adc]

## 3.4 Linearführung

Zur Demonstration einer linearen Bewegung wird eine kompakte Linearführung verwendet. Die Führung ist 400 mm lang und 12 mm breit. Diese Linearführung wird vor allem in Fused Deposition Modeling (FDM)-Drucker verwendet, wo es auf hohe Präzision bei niedrigen Toleranzen ankommt. Deshalb eignet sich diese Führung auch gut für dieses Projekt, wo es nicht darum geht, große Lasten zu bewegen, sondern möglichst genau die Schrittauflösung auf eine Millimeterskala zu übertragen.

## 3.5 Drehwinkel-Encoder

Der Demonstrator soll mehrere Programme fahren können, deswegen wurde ein Drehwinkel-Encoder der Steuerung hinzugefügt. Dieser wird über fünf Pins am Arduino angeschlossen. Durch drehen des Drehschalters werden nacheinander zwei Kontakte geschlossen oder geöffnet. Dieser dadurch entstehende Signalfuss, bestehend aus zwei um 90 Grad versetzte Sinus bzw. Cosinus Schwingungen werden ausgewertet. Daraus wird bestimmt, in welcher Richtung (im oder gegen Uhrzeigersinn) und wie weit (inkrementell) gedreht wurde. Mithilfe dieser Logik kann durch ein Menü eine Bewegungsstufe ausgewählt werden, die der Schrittmotor fahren soll.[Bas16] Bei einer Drehung im Uhrzeigersinn wird im Menü eine Bewegungsstufe höher und bei einer Drehung gegen den Uhrzeigersinn eine Bewegungsstufe niedriger ausgewählt. Zusätzlich zum Drehwinkel-Encoder ist auch noch ein Schaltfunktion im Bauteil selbst integriert. Durch eindrücken des Encoders wird ein Taster betätigt, durch den der eingestellte Wert bestätigt und an den Arduino zur weiteren Verarbeitung weitergegeben wird. Zur Besseren Handhabung des Drehwinkel-Encoders wurde noch ein Drehgriff angefertigt und auf dem Drehgeber montiert. Weitere Details:

- **Abmessungen** ( $b \times l \times h$ ): 18 mm  $\times$  31 mm  $\times$  30 mm
- **Betriebsspannung**: 3,3 - 5 V [Sim19]

## 3.6 Schrittmotorsteuerung

Für eine einfachere Bedienung des Schrittmotors wurde Schrittmotortreiber mit integriertem Übersetzer verbaut. Bei einer Ausgangskapazität von bis zu 35 V und  $\mp 2$  A ist der DEBO DRV A4988 für den Betrieb von bipolaren Schrittmotoren ausgelegt. Auf dem Treiber ist ein Stromregler mit fester Ausschaltzeit integriert, in dem zwischen einer langsamen oder gemischten Abklingmodi gewählt werden kann. Durch eine Impulseingabe wird der Motor um einen Mikroschritt angetrieben. Der Vorteil

der Schrittmotorsteuerung liegt darin, dass keine Phasensequenztabellen, Hochfrequenz-Steuerleitungen oder komplexe Schnittstellen programmiert werden müssen. Im Schrittbetrieb wählt der A4988 automatisch den Abklingmodi, langsam oder gemischt. Im gemischten Modus wird im Abklingvorgang zwischen Schnellen und langsamen Modus gewechselt. Dabei führt der gemischte Modus zu einer Verringerung der hörbaren Motorgeräusche, einer höheren Schrittgenauigkeit und einer geringeren Verlustleistung. Weitere Details:

- **Abmessungen** ( $b \times l \times h$ ):  $5\text{ mm} \times 5\text{ mm} \times 0,9\text{ mm}$
- **Steuerspannung:** 3,3 - 5 V
- **Maximale Ausgangskapazität:** bis 35 V und  $\mp 2\text{ A}$
- **Schutzschaltungen:** Thermische Abschaltschaltung, Schutz vor Masseschluss, Schutz vor kurzgeschlossenener Last, Überkreuzungsstromschutz [All22]

## 4 Schaltplan

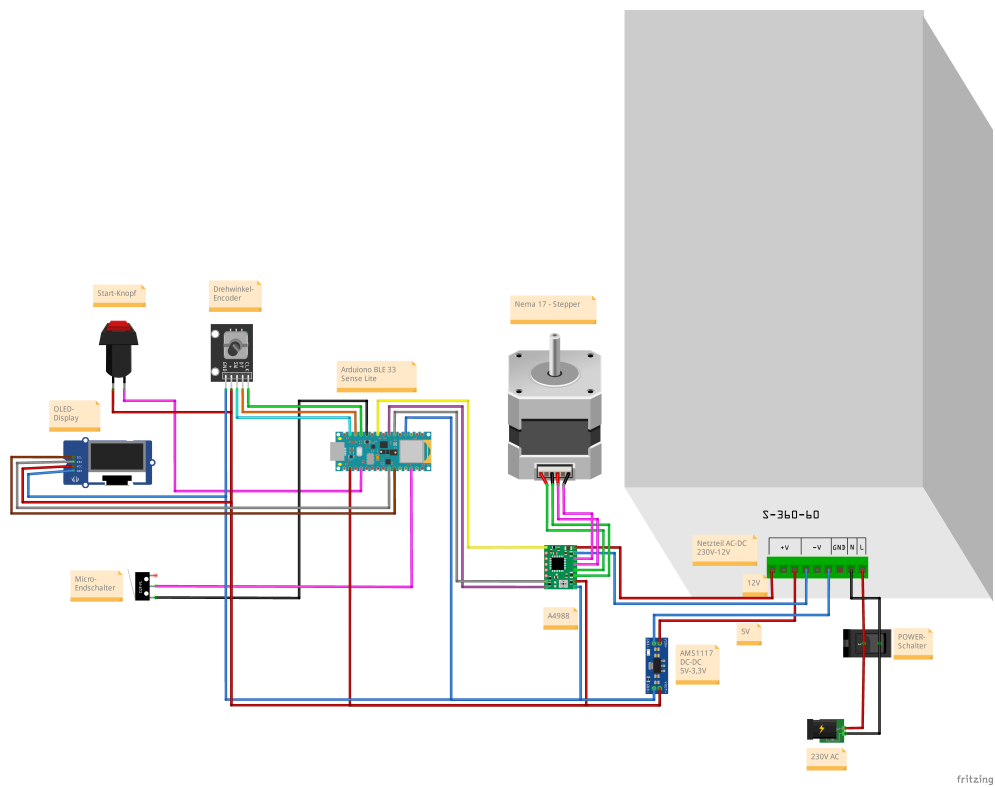


Abbildung 4.1: Schaltplan





## 5 Materialliste

Pos.	Stk.	Bezeichnung	Artikel-Nr.	Preis[€]	Bestelladresse
1	1	Arduino Lern-Kit: - Arduino Nano Sense BLE 33 BLE Lite - USB-A-Micro Ver- bindungskabel - Klemmboard	ARD KIT TINYML	52,40	www.reichelt.de
2	1	CR-10 Nema 17 Schrittmotor 34 mm 42-34	RBS12536	14,25	www.roboter- bausatz.de
3	1	1 m Aluprofil 20 × 20 I-Typ Nut 5	RBS12105	11,81	www.roboter- bausatz.de
4	1	Schaltnetzteil, ge- schlossen, 50 W, 5/12 V, 6 A	SNT RD 50A	17,10	www.reichelt.de
5	1	Spannungswandler Power Adapter Modul 12 V auf 3.3 V/5 V/12 V; AMS1117	RBS16534	1,68	www.roboter- bausatz.de
6	1	GT2 Riemenscheibe 20 Zähne 5 mm Boh- rung für 6 mm Riemen	RBS10277	1,15	www.roboter- bausatz.de
7	1	GT2 Riemenscheibe 20 Zähne 5 mm Boh- rung mit Dual Kugel- lager	RBS12569	1,85	www.roboter- bausatz.de
8	1	yourDroid PLA Fila- ment Grau 1.75 mm 1 kg	RBS13428	15,70	www.roboter- bausatz.de
9	1	Linearführung MGN12H 450 mm	RBS12916	32,86	www.roboter- bausatz.de
10	1	yourDroid GT2 Zahn- riemen offen; 6 mm; 1 m faserverstärkt	RBS10161	2,79	www.roboter- bausatz.de
11	30	M3 Hammermutter T- Schlitz Nut 6	RBS11229	0,25	www.roboter- bausatz.de
12	30	M3 × 20 mm Sechs- kantschraube DIN912 Edelstahl	RBS14063	0,10	www.roboter- bausatz.de
13	1	BMI Lineal 963050030 Maßstab 0,5 m Edel- stahl	823983 - 62	9,49	www.conrad.de
14	1	65 Jumper Wire Kabel im Set	RBS10023	1,59	www.roboter- bausatz.de
15	1	40 Pin Dupont / Jumper Kabel Buchse- Buchse 20 cm	RBS101136	1,09	www.roboter- bausatz.de
16	1	Netzkabel, Schutzkon- taktstecker gew, 1,8 m, schw, C13 gew	NKSK 200 SW GEW	3,00	www.reichelt.de
17	1	Artillery Sidewinder X2 Wippschalter AC- 01	RBS15072	4,19	www.roboter- bausatz.de
18	1	Entwicklerboards - Drehwinkel-Encoder	KY-040	2,20	www.reichelt.de
19	1	Kontermutter für Po- tentiometerknöpfe	P4-Mutter	0,22	www.reichelt.de
20	1	Drucktaster Button Schalter Grün 12 mm; 250 V; 1 A	RBS13869	0,85	www.roboter- bausatz.de
21	1	Mikroschalter / Minia- tur Endschalter	RBS 10856	0,49	www.roboter- bausatz.de
22	1	Entwicklerboards - Schrittmotorsteue- rung, A4988	DEBO DRV A4988	5,80	www.reichelt.de



Pos.	Stk.	Bezeichnung	Artikel-Nr.	Preis[€]	Bestelladresse
23	2	JST - Buchsengehäuse, 1 × 4-polig - EH	JST EH4P BU	0,09	www.reichelt.de
24	1	Signalleuchte, SMD- LED, ø3 mm, Kunst- stoff	MEN 2660.8301	4,60	www.reichelt.de
25	1	0.96ÖLED Display I2C/IIC/TWI 128x64 Pixel für Arduino	RBS10350	4,05	www.roboter- bausatz.de
26	1	Einpressmuttern M3 × 4 × 4 100 Stück für Voron 2.4	RBS15553	3,14	www.roboter- bausatz.de

Tabelle 5.1: Materialliste



# 6 Beschreibung der Software IDE

## 6.1 Installation der Arduino IDE

Um Mikrocontroller von Arduino, wie den verwendeten Arduino Nano 33 BLE Sense Lite, programmieren zu können, wird die Software namens Arduino IDE in der Version 2.3.2 verwendet. Alternativ dazu könnten auch andere Entwicklungsumgebungen, wie Qt verwendet werden, die eine Programmierung in der Programmiersprache C++ ermöglicht. Vorteil der Arduino IDE sind vor allem ihre Funktionen, die das Einbinden der Software auf dem Mikrocontroller vereinfachen. Der Bezug der Software Arduino IDE erfolgt über die offizielle Webseite von Arduino, welche unter [www.arduino.cc/software](http://www.arduino.cc/software) erreichbar ist. Auf dieser Plattform stehen diverse Versionen, für unterschiedliche Betriebssysteme, wie Windows, Linux und Mac OS X, sowie in verschiedenen Dateiformaten, zur Auswahl bereit [Ard24b]. Um die Anwendung zu installieren, muss die heruntergeladene Datei ausgeführt und den Installationsanweisungen gefolgt werden. Danach ist die korrekte Konfiguration des verwendeten Entwicklungsboards sowie die Installation entsprechender Bibliotheken erforderlich.

## 6.2 Beschreibung der Entwicklungsumgebung

Nach dem Start der installierten Anwendung öffnet sich die Hauptoberfläche. Oben in der Hauptoberfläche befindet sich eine Menüleiste, die verschiedene Menüs wie *Datei*, *Bearbeiten* und *Sketch* enthält. Diese Menüs ermöglichen das Bearbeiten und Öffnen von Sketches, also Programmen in der IDE, das Kompilieren und Hochladen von Code sowie das Verwalten von Bibliotheken. Außerdem bietet die Menüleiste ein Hilfe-Menü, das bei Fragen und Problemen hilfreich ist. Die Symbolleiste auf der linken Seite der Hauptoberfläche bietet Schaltflächen für häufig genutzte Funktionen wie das Verwalten von Sketches, Boards und Bibliotheken. Im Zentrum der Hauptoberfläche befindet sich ein Code-Editor, der die Bearbeitung der Sketches ermöglicht. Direkt darunter erscheint nach dem ersten Kompilieren eines Sketches ein Ausgabefenster, das den Kompilierungs- und Hochladeprozess sowie eventuelle Fehler während

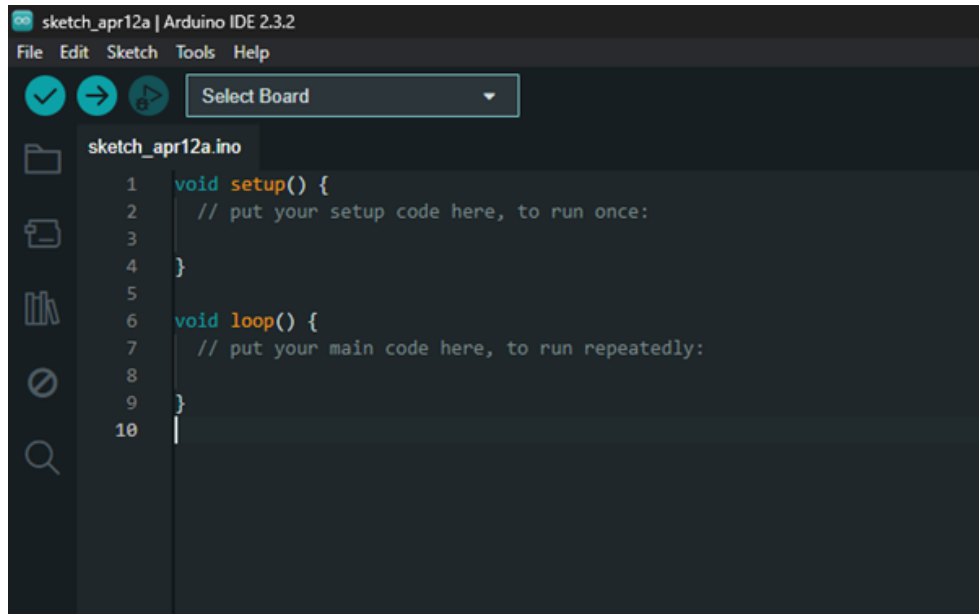


Abbildung 6.1: Hauptoberfläche in der Arduino IDE

der Kompilierung anzeigt. Dadurch können Probleme im Code schnell identifiziert werden.

## 6.3 Erste Schritte in der Entwicklungsumgebung

### 6.3.1 Auswahl des Mikrocontrollers

In der Entwicklungsumgebung können sofort Sketche geöffnet und geschrieben werden. Um den Sketch aber kompilieren zu können, muss das korrekte Board ausgewählt werden. Dafür muss zunächst das passende Board installiert werden. Dies erfolgt über den *Boards-Manager*. Im *Boards Manager* kann in diesem Fall die Datei Arduino Mbed OS Nano in der aktuellen Version 4.1.1 heruntergeladen und installiert werden. Danach kann unter *Select Board* das Board Arduino BLE Sense 33 ausgewählt werden [Ard24d]. Alternativ kann das Board über ein USB-Kabel angeschlossen werden. Unter *Select Board* wird bereits das korrekte Board und der entsprechende COM, also in diesem Fall der USB-Port, angeboten. Werden diese ausgewählt, sind die Installationsanweisungen zu befolgen, um das Board zu installieren.

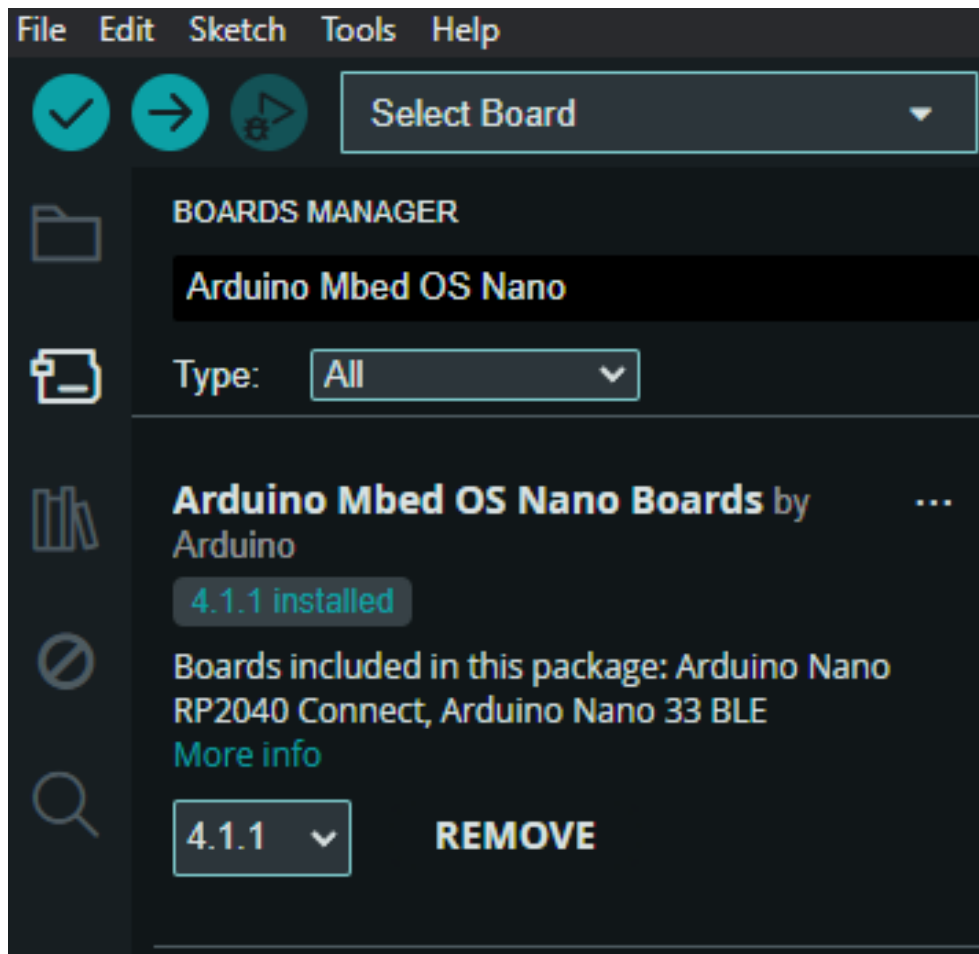


Abbildung 6.2: Installation des Boards

### 6.3.2 Bibliotheken einbinden

Von Arduino gibt es bereits sehr viele offiziell unterstützte Bibliotheken. Will man diese in einem Sketch nutzen, muss man diese zunächst installieren. Dafür kann über den *Library Manager* die nötige Bibliothek heruntergeladen und installiert werden. Die Suchfunktion hilft dabei, die korrekte Bibliothek zu finden [Ard24c]. Nachdem die Bibliothek installiert ist, kann sie im header in den Sketch eingebunden werden. Bei Verwendung von Bibliotheken, die nicht im *Library Manager* zu finden ist, kann diese über *Sketch->Include Libraries->Add .ZIP Library...* eingebunden werden.

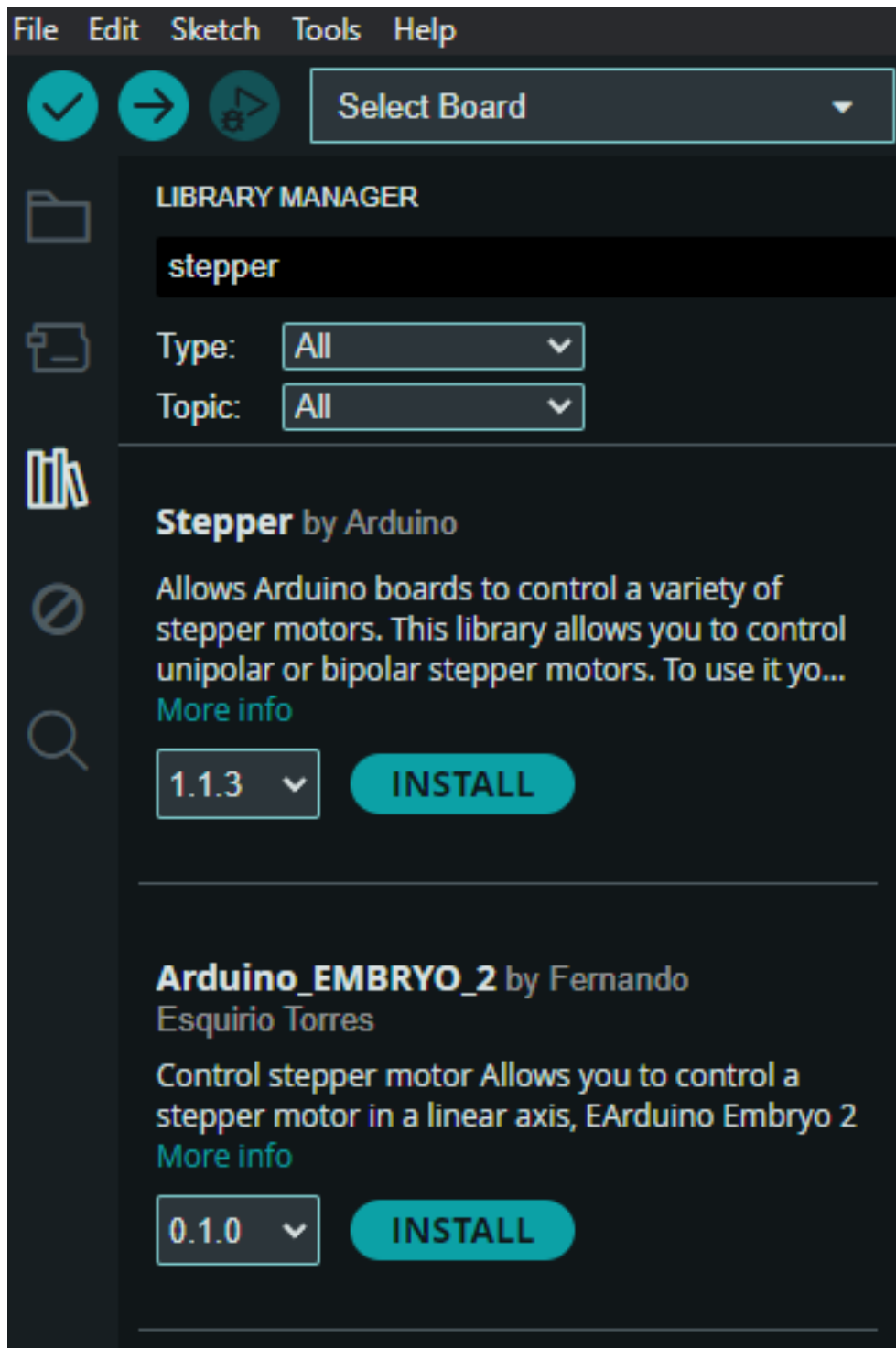


Abbildung 6.3: Herunterladen von Bibliotheken

## 6.4 Programmierung

Wird ein neuer Sketch geöffnet, sind bereits die Funktionen *void setup()* und *void loop()* hinterlegt.

### 6.4.1 header

Im *header* werden vor allem alle nötigen Bibliotheken hinterlegt und initialisiert. Gleichzeitig können hier aber auch Adressen von Peripheriegeräten hinterlegt und erste wichtige Variablen definiert werden.

### 6.4.2 setup()

Die *void setup()*-Funktion wird bei jedem Neustart oder Reset einmal ausgeführt [Arduino.2024d]. Hier wird zum Beispiel die serielle Kommunikation gestartet, aber auch verwendete Pins initialisiert und zugewiesen.

### 6.4.3 loop()

In der *void loop()*-Funktion findet der größte Teil des Programms statt. Der Inhalt dieser Funktion wird dauerhaft wiederholt, bis entweder kein Strom mehr an dem Arduino anliegt, oder der Reset-Knopf gedrückt wird und somit zunächst erst die *void setup()*-Funktion wieder gestartet wird [Ard24e].

## 6.5 Erster Programmtest

Um die Funktion eines Systems zu testen, wird oft zunächst ein sehr simples Programm oder eine sehr grundlegende Funktion getestet. In diesem Fall kann hier über *Datei -> Beispiele -> 01.Basics -> Blink* ein Sketch geöffnet werden, in dem die auf dem Arduino Nano 33 BLE Sense Lite aufgebrachte LED in einem fest gelegten Takt blinkt, wodurch die korrekte Auswahl des Boards und die Übertragung des Sketches auf den Mikrocontroller getestet werden können.

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(1000);  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(1000);  
}
```

In der *void setup()*-Funktion wird in diesem Beispiel über den Ausdruck *pinMode(LED BUILTIN, OUTPUT)* die auf dem Mikrocontroller verbaute LED in dem Sketch aufgerufen und eingebunden. In *void loop()* wird durch den Befehl *digitalWrite()* die LED eingeschaltet, indem eine Spannung an sie angelegt wird. Nach einer kurzen Verzögerung durch den gleichen Befehl wird die LED dann wieder ausgeschaltet, indem die anliegende Spannung gesenkt wird. Die Verzögerung kann mit dem *delay()*-Befehl eingestellt werden, indem der Wert in der Klammer angepasst wird. Dabei wird der Wert in Millisekunden angegeben.



## **7 Beschreibung des Programms auf dem Arduino**



## 8 Beschreibung des Schrittmotors

In diesem Kapitel folgt eine grundsätzliche Beschreibung eines Schrittmotors. Darauf folgt die Erläuterungen zum Aufbau und Funktionsweise eines Schrittmotors sowie die Aufzählung verschiedener Grundbauarten.

### 8.1 Beschreibung eines Schrittmotors

Ein Schrittmotor ist ein Elektromotor, der sich für präzise Positionierungsaufgaben eignet. Im Gegenteil zu anderen Elektromotoren wird bei einem Schrittmotor keine Positionsmessung oder Positionsregelung benötigt. Mit diesem Motor ist eine erreichbare Positioniergenauigkeit von  $0,1^\circ$  möglich. Diese Art von Elektromotor wird beispielsweise in Druckern oder Scanner, aber auch im Kraftfahrzeugbereich verwendet. Im Kraftfahrzeugbereich werden die Schrittmotoren zur Spiegelverstellung sowie der Sitzverstellung verwendet. Das maximal erreichbare Drehmoment eines Schrittmotors liegt bei zwei Newtonmeter und die maximal erreichbare Drehzahl bei ca. 2000 Umdrehung pro Minute. Der Vorteil eines Schrittmotors ist die Wartungsfreiheit, da der Rotor keine Wicklungen hat. [Bab23][Hag21]

#### 8.1.1 Aufbau und Funktionsweise eines Schrittmotors

Der Schrittmotor besteht aus einem Stator, einem Rotor und einer Steuerelektronik. Diese Steuerelektronik setzt sich zusammen aus einer Treiberstufe und der eigentlichen Steuerung. Bei dem Stator handelt es sich um den feststehenden äußeren Teil und bei dem Rotor um den beweglichen inneren Teil. Im Stator sind Spulen verbaut, die von einem Strom durchflossen werden. Hierdurch entsteht ein magnetisches Feld. Da der Rotor magnetisch ist, folgt er dem Magnetfeld des Stators. Soll eine Bewegung hervorgerufen werden, werden einzelne Wicklungsstränge ein- aus- oder umgeschaltet. Durch diesen Vorgang wird ein rotierendes Magnetfeld erzeugt. Das erzeugte Magnetfeld zieht den Rotor an, wodurch dieser sich in kleinen Schritten bewegt. Die Anzahl der Polpaare im Stator geben die Anzahl der Schritte vor. Die Drehzahl und die Drehrichtung hängt von der Reihenfolge und der Häufigkeit der Stromimpulse ab. Es gibt drei verschiedene Betriebsarten, die abhängig von der Genauigkeit und

der Drehzahl sind. Im Vollschrittbetrieb werden alle Polpaare bestromt. In dem Halbschrittbetrieb wird die Schrittzahl des Motors verdoppelt, wodurch sich die Positionsauflösung im Gegensatz zum Vollschrittbetrieb verdoppelt. Allerdings wird in diesem Betrieb das Drehmoment reduziert. In dem Mikroschrittbetrieb bewegt sich der Rotor in sehr kleinen Schritten. Hierdurch wird eine hohe Positioniergenauigkeit und ein ruhiger Lauf erreicht, da die Ströme und das Drehmoment in kleineren Schritten verändert werden. [Hag21][Ber18]

### 8.1.2 Schrittmotor Bauformen

Wie bei anderen Elektromotoren gibt es auch bei dem Schrittmotor verschiedene Grundbauarten.

- Permanentmagneterreger-Schrittmotor (PM-Schrittmotor)
- Reluktanzschrittmotor (VR-Schrittmotor)
- Hybridschrittmotor (HY-Schrittmotor)

Der **permanentmagnetische Schrittmotor** hat einen Permanentmagneten in dem Rotor verbaut. Hierbei stellt sich der permanentmagnetische Rotor immer so, dass der Nordpol des Rotors dem Nordpol des Statorfeldes gegenüber liegt und der Südpol des Rotors dem Südpol des Statorfeldes. In dieser Ausrichtung ziehen sich die Pole gegenseitig an. Die Drehrichtung des Rotors hängt von der Fließrichtung des Stromes ab. Der permanentmagnetische Schrittmotor entwickelt im ausgeschalteten Zustand ein Drehmoment zur Selbsthaltung. Dies ist aufgrund des permanentmagnetischen Rotors möglich. Bei diesem Drehmoment handelt es sich um das höchste Drehmoment, das auf die Welle des Motors übertragen werden kann, ohne dass diese sich in eine rotierende Bewegung versetzt. Zu dieser Art von Schrittmotoren gehören beispielsweise der Klauenpol-Schrittmotor und der Scheibenmagnet-Schrittmotor. Bei der zweiten Bauart handelt es sich um den **Reluktanzschrittmotor**. Bei dieser Bauart besteht der Rotor aus einem weichmagnetischen Material und besitzt eine gezahnte Form. So lange der Schrittmotor von keinem Strom durchflossen wird, entsteht kein Magnetfeld. Sobald der Motor in Betrieb genommen wird, entsteht ein magnetischer Fluss innerhalb des Rotors. Wird nun eine Wicklung erregt, wird der nächste Zahn des Rotors angezogen. Dadurch, dass die Rotorzähne ungleich der Polteilung sind, kann das System unendlich lange fortgesetzt werden. Die Anzahl der Schritte und die Genauigkeit des Reluktanzschrittmotors ist abhängig von der Anzahl der Zähne auf dem Rotor. Aus technischer Sicht sind mit dieser Bauart Schrittwinkel unter  $1^\circ$  möglich. Damit die Drehrichtung verändert werden kann, sind mindestens zwei Strangwicklungen nötig. Bei den **Hybridschrittmotoren**, auch bekannt als HY-Schrittmotoren

handelt es sich um eine Kombination aus dem Reluktanzschrittmotor und dem Permanentmagneterreger-Schrittmotor. Durch diese Kombination aus den beiden Schrittmotoren werden die Vorteile aus der kleinen Schrittweite, dem hohen Drehmoment und dem Selbsthaltungsmoment genutzt. Bei dem HY-Schrittmotor besteht der Rotor aus zwei um eine halbe Zahnteilung versetzten weichmagnetischen Polrädern, die eine zahnförmige Form haben. Bei den beiden Polrädern bildet das eine Polrad den Nordpol und das zweite Polrad den Südpol. Zwischen den beiden Polrädern befindet sich ein Permanentmagnet. Anders als bei anderen Schrittmotoren wird der Rotor bei dieser Bauart axial magnetisiert. Damit ein kleiner Schrittwinkel möglich ist, haben die Statorpole ebenfalls eine zahnförmige Form. Die Ausrichtung des Rotors ist abhängig von der Stromrichtung und wird durch den minimalen Widerstand bestimmt, der sich aus dem Stromfluss durch die einzelnen Stränge ergibt. Wird ein besonders kleiner Schrittwinkel benötigt, kann dies durch Erhöhung der Zähnezahl erreicht werden. [Sch13] [Hag21] [Bab23]

### 8.1.3 Betriebsarten unipolar und bipolar

Neben den oben bereits genannten Betriebsarten, kann außerdem zwischen dem Unipolarbetrieb und dem Bipolarbetrieb unterschieden werden. Der große Unterschied zwischen den beiden Betriebsarten besteht darin, dass in dem Unipolarbetrieb der Strom in eine Richtung fließt. Bei dem Bipolarbetrieb hingegen fließt der Strom in beide Richtungen. Dies ist möglich, da jeder Wicklungsstrang über eine Vollbrücke gespeist wird. Ein weiterer Unterschied besteht in der Schaltung der Zweige, durch die ein Gleichstrom fließt. In dem Unipolarbetrieb werden die beiden Zweige in Reihe geschaltet. Jeder Wicklungsstrang wird mit zwei Drähten parallel gewickelt. Sind die beiden Wicklungsstränge in dem Bipolarbetrieb parallel gewickelt, müssen die Zweige parallelgeschaltet werden. Im Bipolarbetrieb kann ein höherer Wirkungsgrad erzielt werden, wo hingegen der Unipolarbetrieb eine deutlich einfachere Schaltung aufweist. [Sch13]

## 8.2 Beschreibung des verwendeten Schrittmotors

Für das Automatisierungsprojekt wird ein Nema 17 Schrittmotor der Firma Creality3D verwendet. Dieser Schrittmotor wird im 3D-Druck sowie in CNC-Maschinen eingesetzt. Der Motor arbeitet im Bipolarbetrieb und es sind zwei Spulen verbaut. Er hat einen Schrittwinkel von  $1,8^\circ$  und benötigt so für eine Umdrehung 200 Schritte. Die Wellenlänge beträgt 20 mm und der Wellendurchmesser 5 mm. Der ausgewählte Schrittmotor arbeitet mit einer Nennspannung von 5 Volt. Dadurch, dass er mit einer

niedrigen Spannung arbeitet, kann kein hohes Drehmoment erzeugt werden. Allerdings erzielt der Schrittmotor eine hohe Positioniergenauigkeit durch den Schrittwinkel von 1,8 °. Des weiteren arbeitet der Motor mit 0,84 Amper pro Phase. Weitere Vorteile des Schrittmotors sind die kompakte Bauform mit den Abmessung 42 x 42 x 34 mm sowie das leichte Gewicht von 0,22 kg. Betrieben werden kann der Schrittmotor in einer Umgebungstemperatur von -20 °C bis +50 °C.[Jko]

## 9 Schrittverluste Verhindern

Um mögliche Schrittverlusten einzugrenzen und oder sie zu verhindern, wird in diesem Kapitel beschrieben, wie die Ursachen für Schrittverluste oder Stillstand methodisch zu ermitteln sind. [FaulhaberDriveSystems.2020]

### 9.1 Auswahl des Schrittmotors

Zunächst muss ein passender Motor für die Anwendung gewählt werden. Dabei sollten folgende grundlegende Regeln erfüllt sein:

- Motorauswahl durch Höchstwerte für Drehmoment und Drehzahl (Worst-Case-Szenario)
- Verwendung eines Sicherheitsaufschlag von 30 % auf die Drehmoment-Drehzahl-Kennlinie(Kippmoment)
- Sicherstellen, dass externe Ereignisse die Anwendung nicht blockieren können

Sind die geforderten Drehmomente bei den jeweiligen Drehzahlen, den Motorspezifikation entsprechend, dann sind keine Probleme zu erwarten. Ist der Motor zu schwach gewählt und die Anwendung fordert mehr Leistung als der Motor abgeben kann, so bleibt der Motor stehen. Der nächste Schritt ist die Durchführung von Testdurchläufen. Es soll im Betrieb überprüft werden, ob Schrittverluste auftreten. Schrittmotoren verlieren konstruktionsbedingt nicht nur einen einzigen Schritt. Bei geringen Drehzahlen verliert der Motor ein Vielfaches von vier Schritten.[FaulhaberDriveSystems.2020]

### 9.2 Betriebsart

In diesem Abschnitt werden je nach Betriebsart mögliche Ursachen erläutert, falls der Schrittmotor bei den Tests versagt.

#### 9.2.1 Start-Stopp-Betrieb

Der Motor ist mit der Last fest verbunden und wird mit konstanter Drehzahl betrieben. Innerhalb des ersten Schrittes muss der Motor auf die vorgegebene Frequenz beschleunigen.[FaulhaberDriveSystems.2020]

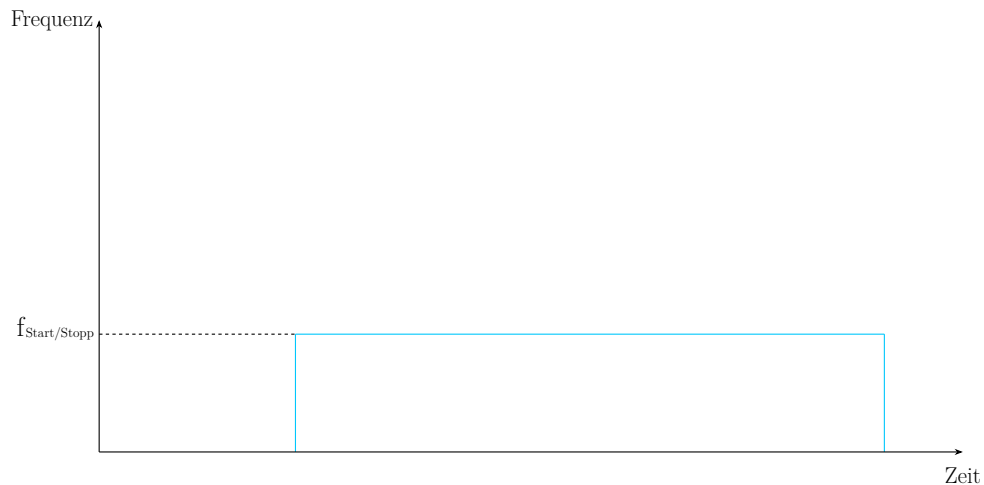


Abbildung 9.1: Start-Stopp Frequenz [FaulhaberDriveSystems.2020]

### Fehlerbild: Motor läuft nicht an

Ursachen	Lösungen
Last zu Hoch	Falscher Motor, größeren Motor wählen
Frequenz zu hoch	Frequenz reduzieren
Pendelt der Motor von Links nach Rechts	Es könnte eine Phase unterbrochen oder nicht angeschlossen sein, dies muss repariert werden
Phasenstrom passt nicht	Phasenstrom erhöhen

Tabelle 9.1: Ursachen und Lösungen: Motor läuft nicht an [FaulhaberDriveSystems.2020]

## 9.2.2 Beschleunigung und Rampenprofil (Trapezförmig)

Der Motor kann mit einer im Controller vorgegebenen Beschleunigungsrate bis auf die Maximalfrequenz beschleunigen. [FaulhaberDriveSystems.2020]

**Fehlerbild: Motor läuft nicht an** (siehe Ursachen und Lösungen aus Tabelle 9.1 Ursachen und Lösungen: Motor läuft nicht an [FaulhaberDriveSystems.2020])

**Fehlerbild: Motor beendet die Beschleunigungsrampe nicht**



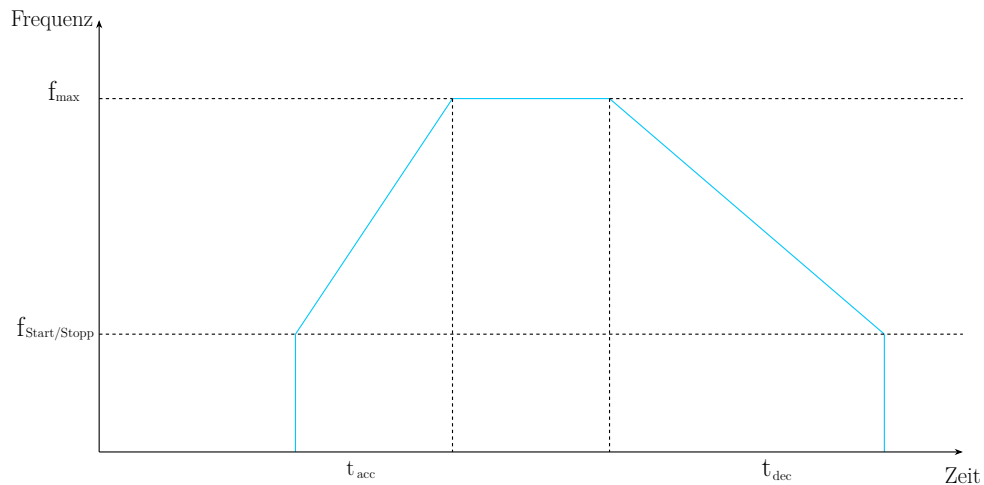


Abbildung 9.2: Trapezförmiges Geschwindigkeitsprofil  
[FaulhaberDriveSystems.2020]

Ursachen	Lösungen
Motor bleibt bei der Resonanzfrequenz hängen	<ul style="list-style-type: none"> <li>• Beschleunigung erhöhen, um die Resonanzfrequenz schneller zu durchlaufen</li> <li>• Start-Stopp Frequenz über dem Resonanzpunkt wählen</li> <li>• Halbschritt- oder Mikroschrittbetrieb verwenden</li> <li>• Mechanische Dämpfung vorsehen</li> </ul>
Falsche Einstellung von Versorgungsspannung oder Strom zu gering	<ul style="list-style-type: none"> <li>• Spannung oder Strom erhöhen</li> <li>• Motor mit geringerer Impedanz testen</li> <li>• Stromregelung verwenden</li> </ul>
Maximaldrehzahl zu hoch	<ul style="list-style-type: none"> <li>• Maximaldrehzahl reduzieren</li> <li>• Beschleunigungsrampe abflachen</li> </ul>
Schlechte Vorgabe der Beschleunigungsrampe durch Elektronik	<ul style="list-style-type: none"> <li>• Anderen Controller verwenden</li> </ul>

Tabelle 9.2: Ursachen und Lösungen: Motor beendet die Beschleunigungsrampe nicht [FaulhaberDriveSystems.2020]

**Fehlerbild: Motor beschleunigt bis zur Enddrehzahl und bleibt stehen, sobald eine konstante Drehzahl erreicht ist.**

Ursachen	Lösungen
Motor wird an Leistungsgrenze betrieben und bleibt stehen aufgrund zu hoher Beschleunigung	<ul style="list-style-type: none"> <li>• Ruckeln verringern durch geringere Beschleunigungsrate oder durch unterschiedliche Beschleunigungsrampen, erst steil dann flacher</li> <li>• Drehmoment erhöhen</li> <li>• Motor im Mikroschrittbetrieb betreiben</li> <li>• Mechanische Dämpfung vorsehen</li> </ul>

Tabelle 9.3: Ursachen und Lösungen: Motor beschleunigt bis zur Enddrehzahl und bleibt stehen, sobald eine konstante Drehzahl erreicht ist. [FaulhaberDriveSystems.2020]

## 9.3 Externe Ereignisse

### 9.3.1 Lastrückkopplung

*"Manchmal wird der vom Motor angetriebene Mechanismus/Last während der Bewegung „aufgezogen“ und gibt diese Energie wieder an den Motor zurück, wenn die Ströme ausgeschaltet werden. Der Mechanismus könnte z.B. ein Untersetzungsgetriebe sein."* [FaulhaberDriveSystems.2020]

Wird diese Energie an den Motor zurück geleitet, kann es passieren, dass der Motor sich um einen Winkel verdreht, der mehr als einem Schritt entspricht. Dabei kann es passieren, dass der Motor nicht ausreichend Drehmoment entwickelt und nicht oder erst nach 4 Vollschritten anläuft. [FaulhaberDriveSystems.2020]

#### Lösungen:

- Die Kommutierung so programmieren, dass Wert und Polarität vor dem Abschalten gespeichert und beim Wiedereinschalten verwendet werden kann.
- Nicht vollständig den Strom abschalten, sondern bei Motorstillstand einen reduzierten *Stand-By* Strom aufrechterhalten

### 9.3.2 Erhöhung der Nutzlast mit der Zeit

*"Manchmal läuft der Motor für eine lange Zeit störungsfrei und viel später treten die ersten Schrittverluste auf. In diesem Fall ist es sehr wahrscheinlich, dass die Last, die der Motor „sieht“, sich geändert hat. Das kann auf Verschleiß der Motorlager oder ein externes Ereignis zurückzuführen sein."* [FaulhaberDriveSystems.2020]

#### Lösungen:

- 
- Prüfen ob ein externes Ereignis durch Veränderung des Mechanismus vorliegt.
  - Prüfen ob Lagerverschleiß vorhanden ist. Verwendung von Kugellager erhöhen die Lebensdauer des Motors.
  - Verwendung von Schmiermittel um Reibung zu verhindern.



## **10 Testdurchläufe**



# 11 Offene Punkte

Hier werden offene Punkte für die spätere Bearbeitung eingefügt.

Die Headers von den .tex Dateien müssen von jedem Autor geändert werden. vlt Schutzhaube als Klemmsicherung





# 12 Literatur

bdbds

