Projekt-Titel: **Automatisiertes Kommissionierungssystem**

Projektleiter: Lukas GREGOR

weitere Teilnehmer: Philip PLEVA

Projektbetreuer: Ahmet KILIC

Verzeichnis

- Verzeichnis
- Projektbeschreibung
- Arbeitsumfang
  - Lineares Bewegungssystem
    - Schrittmotor Treiber DRV8825
      - Modi
      - Implementierung
    - Schrittmotor NEMA 17
      - Eigenschaften
      - Implementierung
    - Endschalter
    - Winkel
    - Software
      - Bibliotheken
      - Programmablauf
  - User Interface

Projektbeschreibung

Das **Automatisierte Automatisiertes Kommissionierungssystem** verfolgt das Ziel, den einfachen Schritt der Paketkommissionierung zu automatisieren und dadurch Geld und Zeit zu sparen. Die Größe des fertigen Projektes ist aufgrund von Ressourcen beschränkt aber es wird mit dem Hintergedanken der Skalierbarkeit entwickelt.

Kommissionierung mit der Hardware zu automatisieren ist nur eine von vielen Bereichen in dem Schrittmotor getriebene Maschinen eingesetzt werden können und der Bedarf nach solchen Systemen ist stehts wachsend.

Arbeitsumfang

Dieser Teil der Diplomarbeit beschäftigt sich mit der Entwicklung eines Schrittmotor getriebenen **Bewegungssystems**, dass eine Plattform entlang der Schiene bewegen kann. Auf dieser Plattform befindet sich ein QR-Code Scanner und ein Drücker die für die Erfassung und Ausgabe der Pakete zuständig sind.

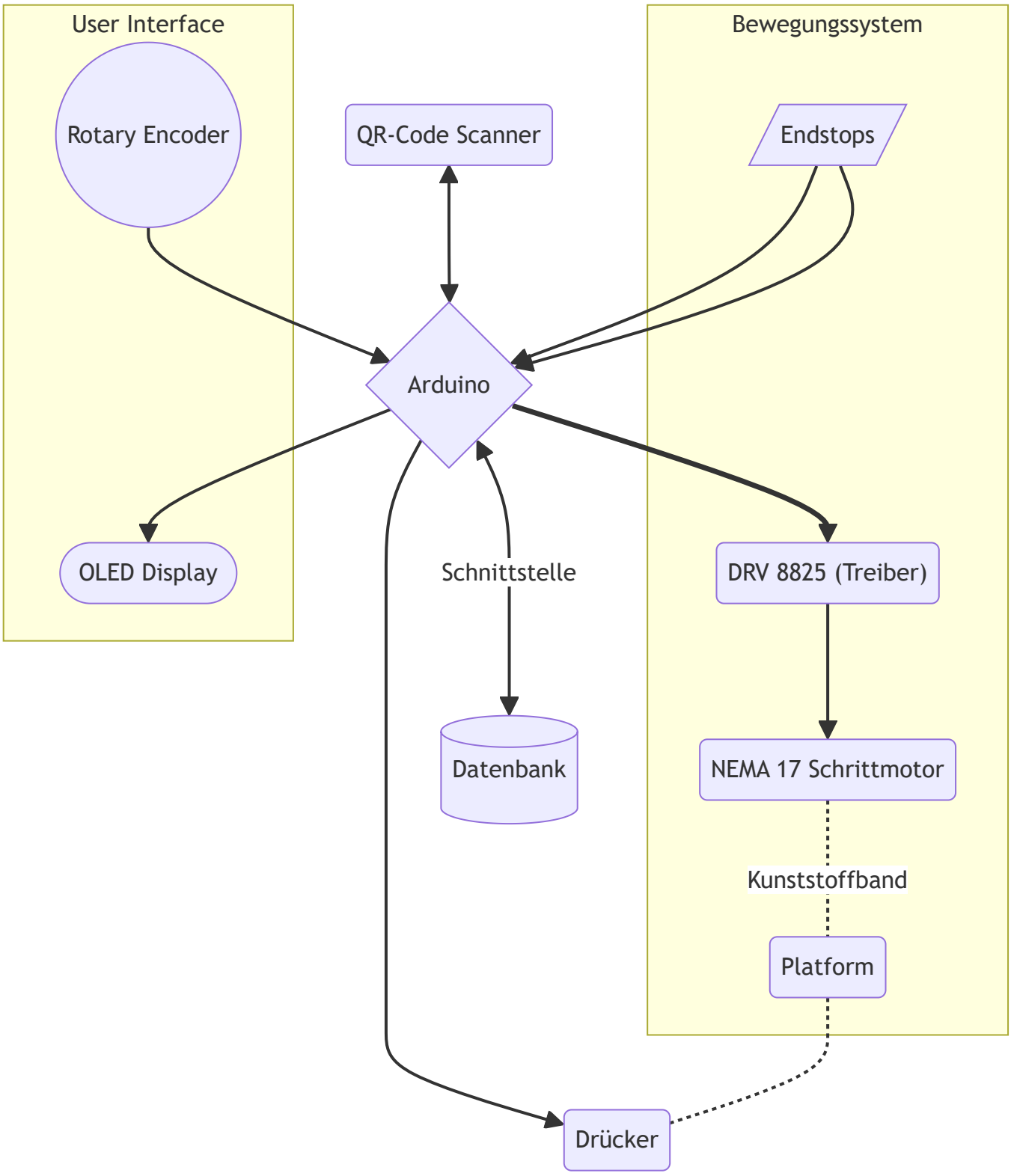

Abbildung 1: Blockdiagramm

Lineares Bewegungssystem

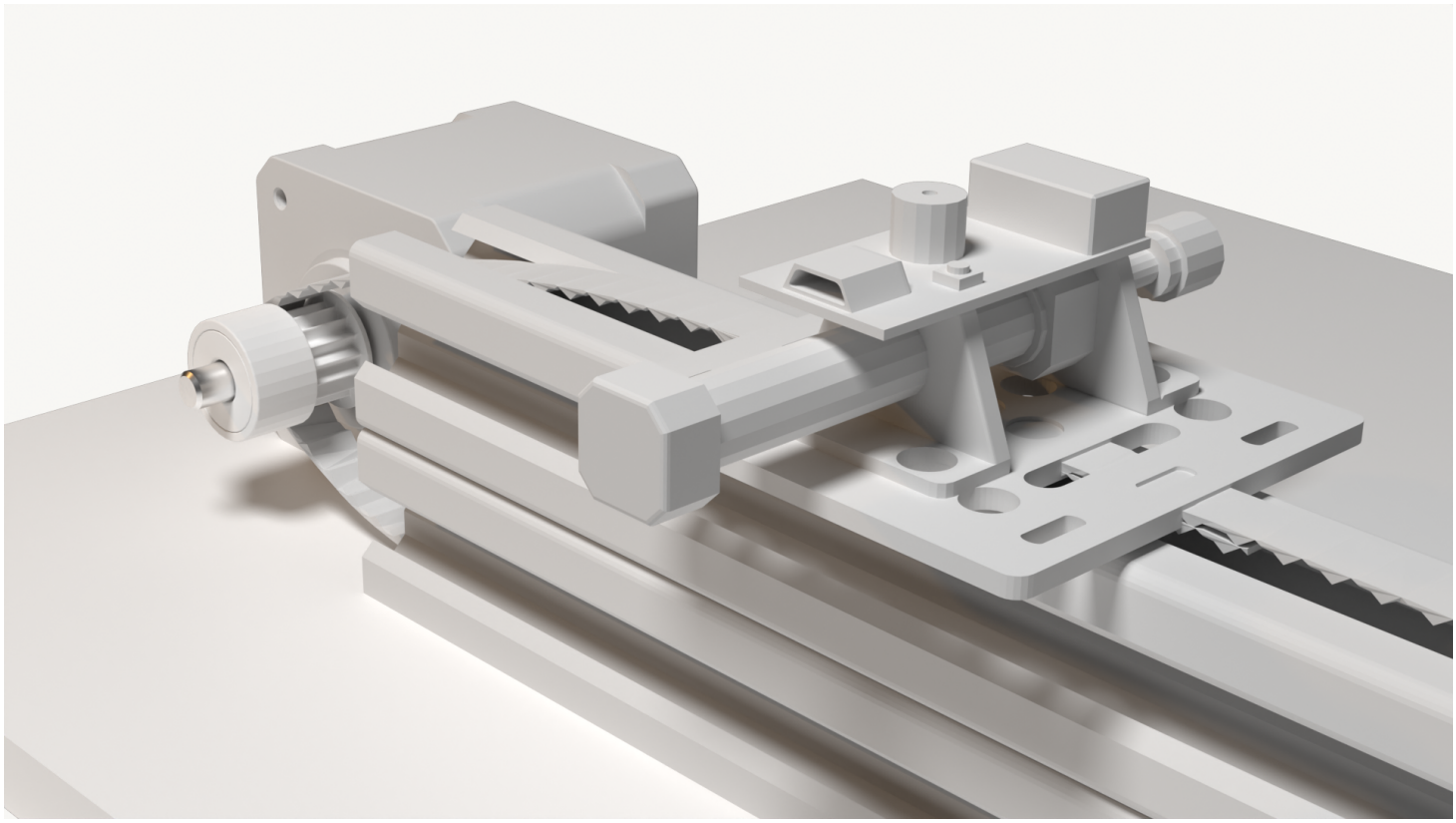

Abbildung 2: Bewegungssystem

Schrittmotor Treiber DRV8825

DRV8825 Schrittmotor Treiber Datenblatt

Der DRV8825 bietet eine Lösung für die Ansteuerung von bipolaren Schrittmotoren. Der Treiber ist in der Lage, einen Strom von 2,5A zu steuern.
Der DRV8825 IC ermöglicht über nur 2 Anschlusspins (**DIR, STEP**) eine vereinfachte Kommunikation mit dem Schrittmotor.
Durch Verwendung weiterer Pins lassen sich verschiedene Modi des Treibers konfigurieren.
Ein weiterer Vorteil den der Motor Treiber liefert ist eine eingebaute Strom-Begrenzungs-Schaltung die in der Prototypenphase versagen des Treibers verhindert.

Modi

PIN	BESCHREIBUNG
DIR	Gibt die Richtung an, in die sich der Motor bewegt
STEP	Eine positive Taktflanke löst den nächsten Schritt aus
nENABLE	Schaltet den Treiber aus. Immer HIGH, außer es ist eine Bewegung des Motors erforderlich.
M0	Steuerung des Schrittmotor-Betriebes
M1	Steuerung des Schrittmotor-Betriebes
M2	Steuerung des Schrittmotor-Betriebes

Über den Zustand des **DIR**-PIN wird die Richtung, in die sich der Motor dreht gesteuert. Eine positive Taktflanke auf dem **STEP**-PIN erziehlt einen Schritt in die jeweilige Richtung. Die Drehgeschwindigkeit ergibt sich dadurch aus der Frequenz, mit der positive Taktflanken auf dem **STEP**-PINs geschehen.

Von den Modi **nSLEEP** und **nRESET** wird kein Gebrauch gemacht. Sie müssen allerdings permanent auf HIGH gesetzt werden, um die Funktion des Treibers zu ermöglichen.

Durch Verwendung Pins(**M0, M1, M2**) lässt sich die Steuerung des Stepper-Betriebes konfigurieren wodurch sich ein optimierter Betriebsmodus auswählen lässt.

In der unveränderten, für den Steppermotor geschriebene Software wird ausschließlich der **32 microstep-Betrieb** und der **1/4 step-Betrieb** verwendet. Der **32 microstep-Betrieb** ermöglicht eine möglichst präzise Positionsauflösung bei Kalibrierung des Bewegungssystems. Die Geschwindigkeit ist dabei nicht von Priorität.

Der **1/4 step-Betrieb** ist optimal für den regulären Betrieb des Bewegungssystems da eine hohe Geschwindigkeit erzielt werden kann, ohne Fehler zu riskieren, die bei zu niedrigen Schrittmotor-Betrieben vorkommen können und den Motor zum Halten verleiten.

Implementierung

Beim Anschaltvorgang kann es vorallem bei langen Leitungen zu hohen Spitzenspannungen kommen welche gefährlich für die auf dem eingebauten Keramikkondesatoren sind. Als Sicherheitsmaßnahme wird so knapp wie möglich an der Versorgungsspannung ein ELKO eingebaut der laut Hersteller mindestens 47uF haben soll.

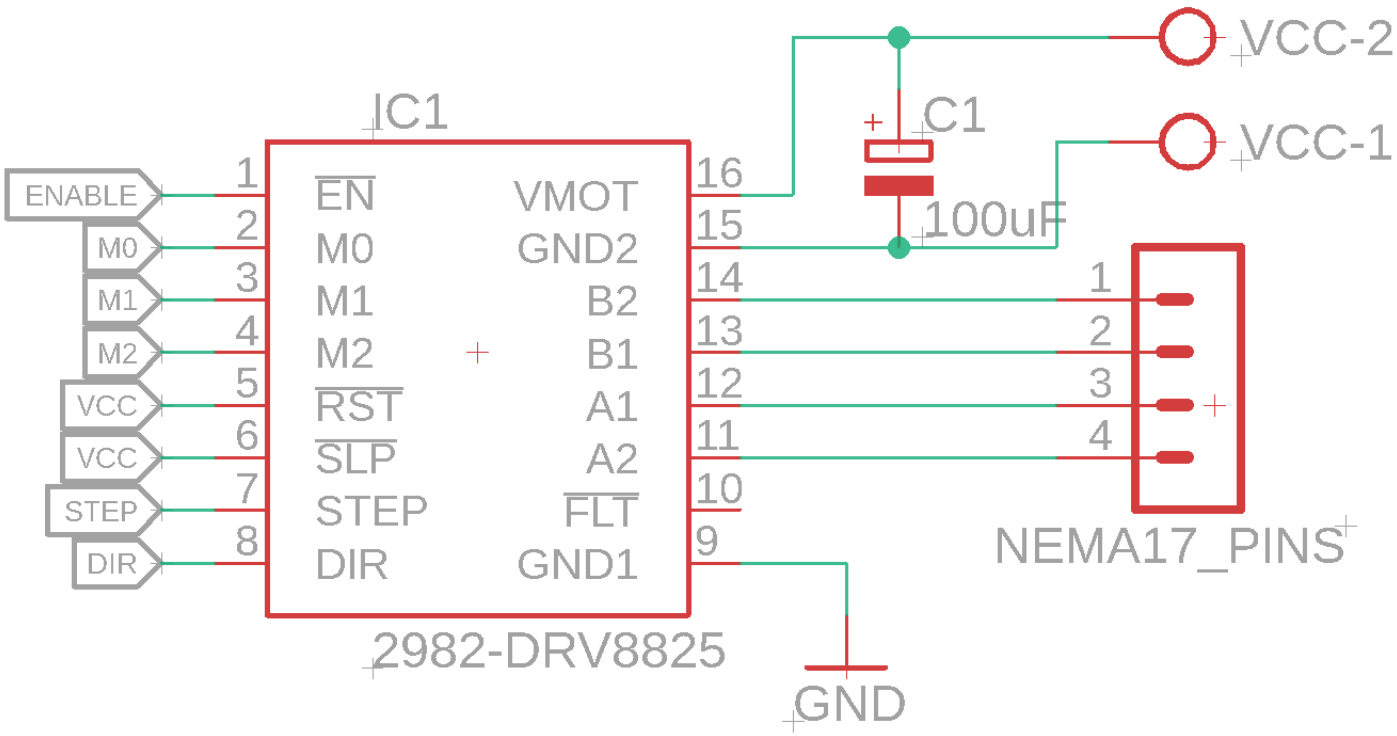

Abbildung 4: DRV8825 Treiber

MODE0	MODE1	MODE2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

Abbildung 5: Betriebsmodi Tabelle

Die Tabelle zeigt alle wählbaren Betriebsmodi für den Steppermotor und welche Ausgänge angesteuert werden müssen um die gewünschte Konfiguration einzustellen.

Schrittmotor NEMA 17

Der NEMA 17 ist ein kompakter Schrittmotor der für gewöhnlich in 3D-Druckern, CNC-Maschinen und anderen Bewegungssystemen verwendet wird. Im Fullstep-Betrieb braucht der Motor 200 Steps um eine Umdrehung durchzuführen.

$$Schrittwinkel = \frac{360^\circ}{200steps} = 1.8^\circ/step$$

Diese Auflösung ist für unsere Anwendung ausreichend. Über eine Veränderung im Betriebsmodus lässt sich auf Bedarf eine noch bessere Auflösung von $Schrittwinkel = \frac{360^\circ}{6400steps} = 0,05625^\circ/step$ bei 32-microstepping erzielen.

Eigenschaften

Spezifikation	Wert
Strom	2A
Innenwiderstand	1,3Ω
Drehmoment	0.55Nm

Die Welle des Stepper-Motors über ein Kunststoffband mit der auf der Schiene laufenden Plattform verbunden. Der Motor hat 2 Phasen und somit 4 Anschlussdrähte über die er mit dem DRV8825 verbunden ist.

Implementierung

Beim Beschleunigen auf hohe Umdrehungszahlen ist es möglich, dass der Schritte übersprungen werden. Der Grund dafür könnte ein nicht ausreichendes Motor-Drehmoment für den nächsten Schritt oder oder die Trägheit des Motors sein. Dieses Problem lässt sich verhindern, indem man die Spannungsversorgung erhöht oder die Strecke die der Motor für einen Schritt benötigt(Schrittwinkel) durch die Änderung des Schrittbetriebes veringert.

Endschalter

Das Bewegungssystem hat 2 Endschalter, die auf beiden Winkeln installiert sind und den Bereich vorgeben, in dem sich die Plattform frei bewegen darf. Die Endschalter kommen eigentlich nur bei der Kalibrierung zu Einsatz aber behalten ihre Funktion über den ganzen Programmablauf bei um sicher

zu stellen, dass bei Fehlern die Plattform nicht über die physikalischen Grenzen fährt und Teile des Systems oder sich selbst beschädigt.
Die Endschalter werden in der NC (Normaly Closed) Konfiguration angeschlossen um vorbeugend Defekt bei Drahtbruch zu verhindern.

Winkel

Der Winkel der als Verbindung zwischen Boden und der Schiene dient wurde selbständig designed und gefertigt um Anforderungen zu treffen und Endschalter darauf installieren zu können.

Software

Bibliotheken

Das Projekt verwendet über 2. Wesentliche Bibliotheken.
Die erste ist die **<Adafruit_SSD1306.h>** Bibliothek, die die Kommunikation mit dem OLED-Display vereinfacht und Unterprogramme enthält, die das designen und programmieren des **User-Interface** erleichtern. Die für die Ansteuerung des Motors zuständige Bibliothek **"LukiStepper.h"**, ist eine speziel für die Anwendung modifizierte Version der **SpeedyStepper.h**-Bibliothek. Sie ist auf dem Real Time Stepper Motor Linear Ramping Algorithmus aufgebaut.

Programmablauf

```
// Standart Bewegungsprofil
stepper.setSpeedInStepsPerSecond(16000); //V_max = 16000steps/s
stepper.setAccelerationInStepsPerSecondPerSecond(250000); // A_max = 250000steps/s^2
```

Zuerst wird das optimierte Bewegungsprofil, welches pragmatisch ermittelt wurde eingestellt.
Höhstgeschwindigkeit: **16000steps/s**.

$$Drehzahl...n = \frac{16000 steps/s}{800 steps} = 120 U/min$$

Beschleunigung = 250000steps/s^2

$$Beschleunigungszeit...t_a = \frac{16000 steps/s}{250000 steps/s^2} = 64 ms$$

```
// Microstepping-Betrieb
digitalWrite(M2,HIGH);
```

Mit nur einem Befehl lässt sich die Betriebskonfiguration von 1/4 auf 1/32 umstellen. Siehe *Abbildung 5*.

```
// Kalibrierungssequenz
digitalWrite(nEnable, LOW);
stepper.calibration(-1, 16000, 200000, S1, true);
stepper.calibration(1, 16000, 200000, S2, false);
digitalWrite(nEnable, HIGH);
```

Mit `digitalWrite(nEnable, LOW)` wird der DRV8825 in Betrieb gesetzt. und es wird die Kalibrierungssequenz gestartet. Nachdem die Kalibrierung abgeschlossen ist, wird der PIN *nEnable* wieder HIGH gesetzt.

User Interface