



Hochschule für Technik,
Wirtschaft und Kultur Leipzig

FAKULTÄT INGENIEURWISSENSCHAFTEN

MODUL 5420 - EMBEDDED SYSTEMS

Dokumentation zur Programmierung eines Pen Plotters

Autor Leona Vogel und Carolin Schneider
Betreuer Professor Andreas Pretschner und M.Sc. Marco Braun

13. März 2024

Inhaltsverzeichnis

Abbildungsverzeichnis	ii
Tabellenverzeichnis	ii
1 Einleitung	1
2 Aufbau des Pen Plotters	1
2.1 Beschreibung des Aufbaus	1
2.2 Materialliste	3
2.2.1 Zusätzliche Materialien	3
2.2.2 Materialien aus dem Makeblock-Kit	4
3 Ansteuerung der einzelnen Komponenten des Pen Plotters	5
3.1 Bauelemente zum Ansteuern der Motoren	5
3.2 Bauelemente zum Auslesen der Endlagenschalter	5
4 Programmierung des Pen Plotters	6
4.1 Controller	6
4.2 Verschaltung	6
4.2.1 Bauelemente der 12 V-Spannungsebene	6
4.2.2 Bauelemente der 5 V-Spannungsebene	7
4.2.3 Bauelemente der 3,3 V-Spannungsebene	7
4.3 Programmierumgebung	7
4.4 Bibliotheken	9
4.4.1 Servo.h	9
4.4.2 AccelStepper.h	9
4.4.3 MultiStepper.h	9
4.4.4 SPI.h	9
4.4.5 SD.h	10
4.5 Programmaufbau	10
4.5.1 Zusätzliche Funktionen	11
5 Aufbau der Datei auf der SD-Karte	12
6 Ausblick	13
Quellenverzeichnis	14

Abbildungsverzeichnis

1	Allgemeiner Aufbau	1
2	Stepper-Motor	2
3	Servo-Motor	2
4	Schrittmotor-Treiber	5
5	RJ25-Adapter	5
6	Raspberry Pi Pico Pinout	6
7	Hinzufügen des Boards in den Voreinstellungen	8
8	Hinzufügen des Boards im Boardverwalter	8
9	Rechteck	12

Tabellenverzeichnis

1	Materialliste der zusätzlichen Materialien	3
2	Materialliste der Materialien aus dem Makeblock-Kit	4
3	Genutzte Funktionen aus der Servo.h-Bibliothek	9
4	Genutzte Funktionen aus der AccelStepper.h-Bibliothek	9
5	Genutzte Funktionen aus der SD.h-Bibliothek	10

Listings

1	Beispiel G-Code	12
---	---------------------------	----

1 Einleitung

Ein Pen Plotter dient dazu, mit Hilfe eines Stiftes, der in eine Halterung eingespannt wird, und Motoren, die den Stift bewegen, Bilder zeichnen zu können.

In dieser Dokumentation soll erläutert werden, wie der Pen Plotter aufgebaut ist und wie die einzelnen Komponenten angesteuert werden können. Zudem werden der Aufbau und die Funktionsweise zusätzlicher Komponenten erläutert. Zusätzliche Komponenten dienen dazu, den Funktionsumfang des Pen Plotters zu erweitern.

2 Aufbau des Pen Plotters

2.1 Beschreibung des Aufbaus

Der Pen Plotter (siehe Abbildung 1) besteht aus einem kartesischen Rahmen mit zwei Achsen. Die Achsen können mit jeweils einem Schrittmotor bewegt werden. In einer Halterung, die sich durch die Bewegung der Achsen ebenfalls bewegt, ist ein Stift eingespannt. Dieser Stift kann mit Hilfe eines Servo-Motors angehoben oder abgesetzt werden. Bei angehobenem Stift wird keine Linie gezeichnet. Bei abgesetztem Stift wird eine Linie gezeichnet. Zudem verfügt der Pen Plotter über vier Endlagenschalter mit denen überprüft werden kann, wann die Achsen an den Begrenzungsrahmen anstoßen, sie sich also in ihrer Maximalstellung befinden.

Die Hardware des Pen Plotters stammt von der Firma Makeblock, die fertige Roboter-Bausätze verkauft. Da Makeblock Komponenten verschiedener Hersteller unter neuen Namen weiterverwendet, ist es oft sehr schwierig, ausführliche Datenblätter zu den einzelnen Komponenten zu finden. Die Datenblätter von Makeblock listen oft nur diejenigen Funktionen auf, die für den Einsatz in ihrem Bausatz notwendig sind.

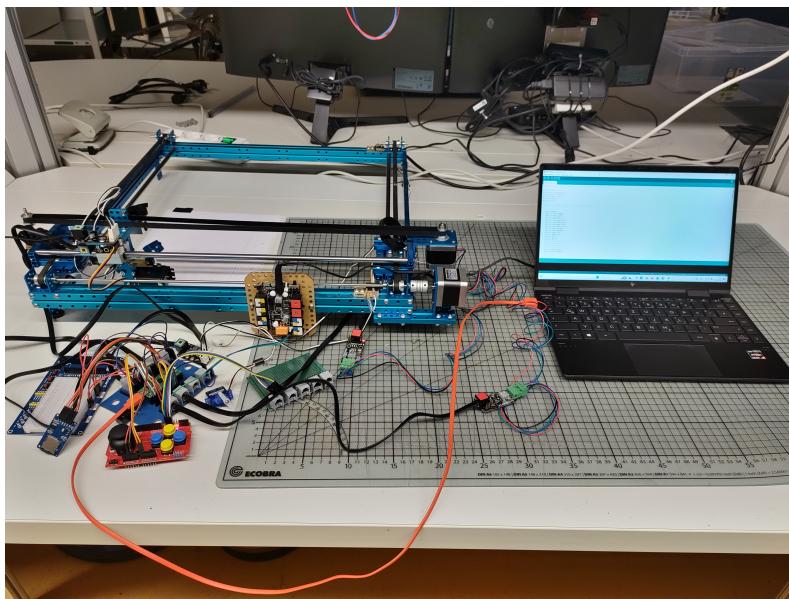


Abbildung 1: Allgemeiner Aufbau des Pen Plotters

Die Schrittmotoren (siehe Abbildung 2) sind vom Typ Makeblock 42BYG. Es handelt sich dabei um einen zweiphasigen Gleichstrom-Schrittmotor mit einer Betriebsspannung von 12 V. In einem Schritt fährt der Motor einen Winkel von $1,8^\circ$ mit einer Genauigkeit von 5%. Das bedeutet, dass der tatsächlich gefahrene Winkel pro Schritt zwischen $1,71^\circ$ und $1,89^\circ$ liegen kann. Der Motor benötigt also ungefähr 200 Schritte für eine volle Umdrehung.[4]

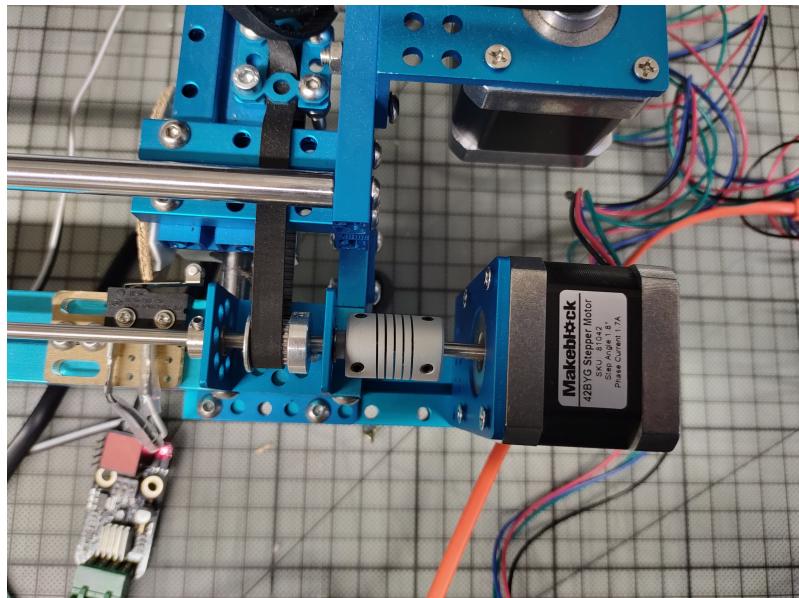


Abbildung 2: Stepper-Motor

In diesem Projekt wird nicht der ursprünglich von Makeblock mitgelieferte Servo-Motor, sondern der Hitec HS 300 RCD Apollo 5, verwendet (siehe Abbildung 3). Dem Servo-Motor kann eine Zielposition mittels einer Pulsweite vorgegeben werden. Die Pulsweite darf sich dabei in einem bestimmten Bereich bewegen, welcher im Datenblatt angegeben ist. Im Falle des Hitec HS 300 RCD Apollo 5“ sind das 900 bis 2100 μ s.[2]

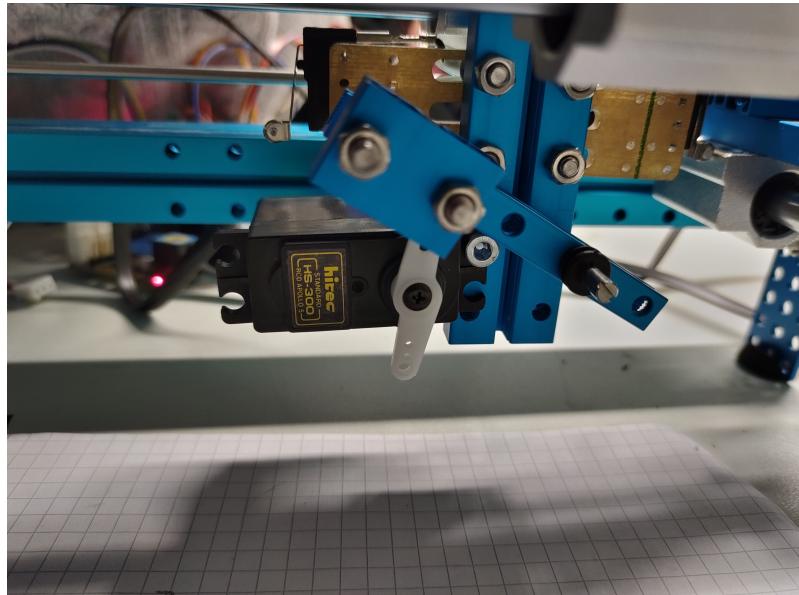


Abbildung 3: Servo-Motor

Für die Programmierung wird anstatt des Me Basebord, welches auf dem Arduino basiert, der Raspberry Pi Pico H verwendet.

Für die Ansteuerung im manuellen Modus wird zusätzlich ein Joystick Shield verwendet, welches über einen Joystick, zwei Tastern und einen Steuerkreuz verfügt. Damit soll im manuellen Modus die Position des Pen Plotters verfahren werden.

Im automatischen Modus soll der Pen Plotter die in einer Datei gespeicherten Bewegungen ausführen. Hierfür wird ein SD-Karten-Modul mit SD-Karte verwendet, auf der dann die Datei gespeichert ist.

2.2 Materialliste

Die Materialliste ist aufgeteilt in zusätzlich verwendete Materialien und solche, die aus dem Makeblock Pen Plotter Kit stammen.

2.2.1 Zusätzliche Materialien

Pos.	Anz.	Name	Beschreibung
1	1	Raspberry Pi Pico H	Microcontroller Raspberry Pi Pico H
2	1	Hitec HS 300 RCD Apollo 5	Servomotor
3	1	JoyStick Shield V1.A	Joystick mit Tastern und Steuerkreuz
4	1	DEBO MICROSD 2	SD-Karten-Modul
5	1	Platine	für den Raspberry Pi mit Spannungswandler und RJ25-Adaptoren

Tabelle 1: Materialliste der zusätzlichen Materialien

2.2.2 Materialien aus dem Makeblock-Kit

Pos.	Anz.	Name	Beschreibung
1	1	Beam 0824-16	Balken für die Stifthalterung
2	4	Beam 0824-48	Balken für die X-Achse
3	1	Beam 0824-80	Balken für die Stifthalterung
4	4	Beam 0824-96	Balken 0824-96
5	2	Beam 0824-112	Balken 0824-112
6	2	Beam 0824-496	Balken für den Rahmen
7	1	Beam 0808-80	Balken für die Stifthalterung
8	2	Beam 2424-504	Balken für den Rahmen
9	2	Bracket 3x3	Halterung
10	5	Plate 3x6	Platte als Füße für den Rahmen
11	5	Bracket U1	Halterung
12	3	Belt Connector	Riemenverbinder
13	2	42BYG Stepper Motor Bracket	Schrittmotorhalterung
14	2	42BYG Stepper Motor	Schrittmotor
15	6	Timing Pulley18T	Zahnräder
16	3	Open-end Timing Belt (1.3m)	Offener Zahnriemen
17	2	D Shaft 4x56mm	D-Welle für das Getriebe
18	1	Threaded Shaft 4x39mm	Gewindewelle
19	10	Shaft Collar 4mm	Wellenkragen
20	1	Flexible coupling 4x4mm	Flexible Kupplung 4x4mm
21	4	Linear Motion Shaft D8X496mm	Lineare Bewegungswelle für die Achsen
22	6	Linear Motion Slide Unit 8mm	Lineare Bewegungsschlitten für die Achsen
23	10	Flange Bearing 4x8x3mm	Flanschlager 4x8x3mm
24	50	Nut M4	Mutter M4
25	10	Countersunk Screw M3x8	Senkkopfschraube M3x8
26	26	Headless Set Screw M3x5	Gewindestift
27	16	Plastic Rivet R4060	Kunststoffniene
28	6	Plastic Rivet R4100	Kunststoffniene
29	20	Plastic Ring 4x7x2mm	Kunststoffring
30	3	Nut M2	Mutter M2
31	3	Cross recessed screw M2×10	Kreuzschlitz-Zylinderkopfschraube M2×10
32	30	Hexagon socket screws M4×14	Innensechskantschraube M4×14
33	28	Hexagon socket screws M4×16	Innensechskantschraube M4×16
34	12	Hexagon socket screws M4×22	Innensechskantschraube M4×22
35	18	Hexagon socket screws M4×30	Innensechskantschraube M4×30
36	36	Hexagon socket screws M4×8	Innensechskantschraube M4×8
37	1	Base Bracket	Basis Halterung
38	4	LS Bracket V2.0	Halterung für die Endlagentaster
39	1	Linear Motion Shaft D4x512mm	Lineare Bewegungswelle für das Getriebe
40	1	Linear Motion Shaft D4x80mm	Lineare Bewegungswelle für die X Achse
41	10	Cross recessed screws-C ST 2.2×9.5	Kreuzschlitz-Zylinderkopf-Schraube
42	5	Cutable Linkage 3	Schneidbare Verbindung
43	30	Nylon cable ties 2*100	Kabelbinder
44	5	Rubber band	Gummiband
45	2	Me Stepper motordriver V1.0	Motortreiber für die Schrittmotoren
46	3	RJ25 Adapter	RJ25 Adapter
47	4	Micro Switch Button	Endlagentaster
48	2	6P6C RJ25 cable-20cm	RJ25 Kabel 20cm lang
49	1	6P6C RJ25 cable-35cm	RJ25 Kabel 35cm lang
50	2	6P6C RJ25 cable-50cm	RJ25 Kabel 50cm lang

Tabelle 2: Materialliste der Materialien aus dem Makeblock-Kit

3 Ansteuerung der einzelnen Komponenten des Pen Plotters

3.1 Bauelemente zum Ansteuern der Motoren

Die Schrittmotoren werden über Motortreiber angesteuert, die von Makeblock speziell dafür entwickelt wurden. Bei dem in Abbildung 4 rot markierten Anschluss handelt es sich um einen RJ25-Anschluss, über den der Motortreiber mit dem Controller verbunden ist. Über den grünen Anschluss ist der Motortreiber mit dem Motor verbunden. Der Motortreiber basiert auf dem Chip A4988 und besitzt ein Potentiometer zum Einstellen des Motorstroms.[1] Für dieses Projekt werden keine hardwareseitigen Einstellungen am Motortreiber vorgenommen.

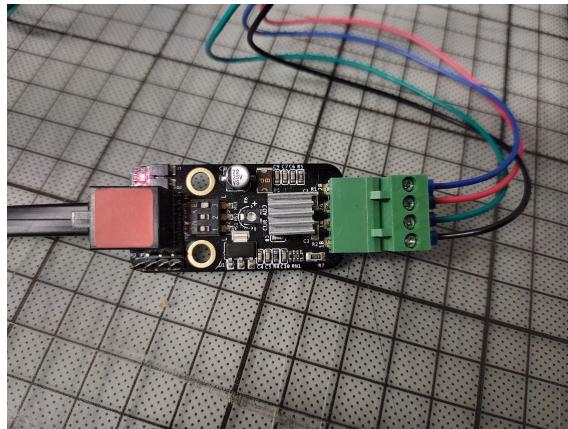


Abbildung 4: Schrittmotor-Treiber

Der Servomotor könnte auch direkt an die 5 V-Spannungsversorgung und den Signalpin des Mikrocontrollers angeschlossen werden. Da die Verbindung vom Controller jedoch einheitlich über RJ25-Leitungen erfolgen soll und der Servomotor sich im Raum bewegt und deshalb ein Kabel aus mehreren Leitungen mechanische Vorteile bietet, gibt es vom Makeblock den ME RJ25 Adapter (siehe Abbildung 5).

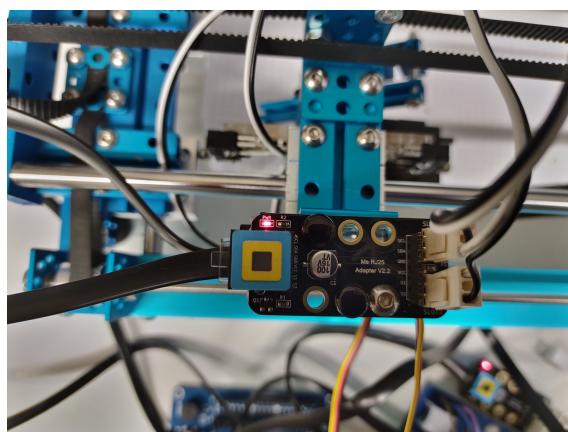


Abbildung 5: RJ25-Adapter für Servo-Motor und Endlagenschalter

3.2 Bauelemente zum Auslesen der Endlagenschalter

Zum Auslesen der Endlagenschalter werden ebenfalls ME RJ25 Adapter wie in Abbildung 5 verwendet. An jeden Adapter können zwei Endlagenschalter angeschlossen werden.

4 Programmierung des Pen Plotters

4.1 Controller

Als Mikrocontroller wird der Raspberry Pi Pico H verwendet. Im Gegensatz zum Raspberry Pi Pico W ist dieser nicht WLAN- und Bluetooth-fähig. Der Mikrocontroller kann mit einer Betriebsspannung zwischen 1,8 V und 5,5 V versorgt werden.[3] Die Spannungsversorgung kann entweder über die Hardware-Pins 38 und 39 (siehe Abbildung 6) oder über den Micro-USB-Anschluss erfolgen. Über den Micro-USB-Anschluss kann der Mikrocontroller auch programmiert werden.

Der Raspberry Pi Pico H verfügt über 26 GPIO-Pins (GP0 bis GP22 und GP26 bis GP28), die alle für digitale Signale und für PWM-Signale verwendet werden können. Zusätzlich besitzen einige der Pins noch eine zweite Funktionalität. GP16 bis GP19 sind die Standard-Pins für die SPI-Kommunikation und GP26 bis GP28 können zusätzlich zum Lesen und Schreiben analoger Signale verwendet werden.

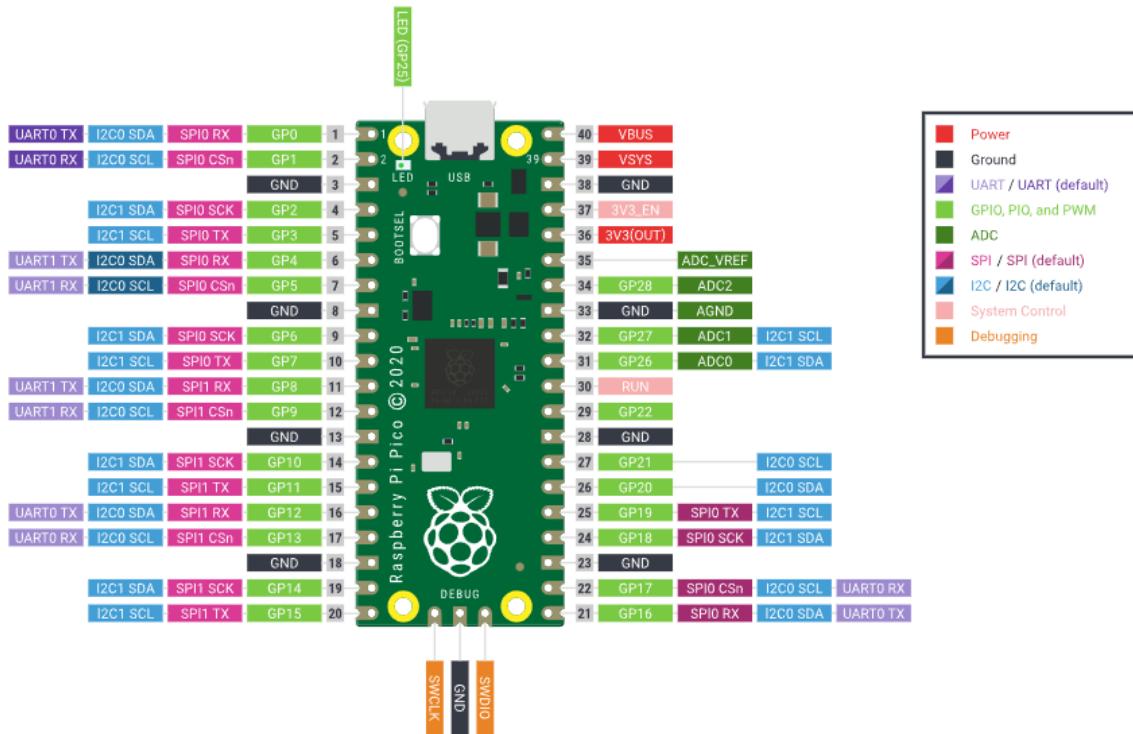


Abbildung 6: Pinout des Raspberry Pi Pico H

Source: <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>

4.2 Verschaltung

Bei der Verschaltung ist vor allem die Spannungsversorgung der Bauteile und die Verbindung zu dem Pin des Raspberry Pi Pico wichtig. Außerdem sollte das Bezugspotential der einzelnen Spannungsebene verbunden werden. Die Verbindungen an den Pins ist ebenfalls im Code gut dokumentiert.

4.2.1 Bauelemente der 12 V-Spannungsebene

An die 12 V-Spannungsebene sind nur die beiden Schrittmotoren mit den jeweiligen Treibern. Dabei ist an den Anschlüssen V M 12 V angeschlossen und an GND 0 V. Bei den Motortreibern

sind darüber hinaus die Pins DIR (Richtung, Direction) und STP (Step) wichtig. Bei dem Motor der X-Achse sind diese an die Pins 6 (STP) und 7 (DIR) angeschlossen und bei dem der Y-Achse an die Pins 8 (DIR) und 9 (STP).

4.2.2 Bauelemente der 5 V-Spannungsebene

Mit der 5 V-Spannungsebene werden der Servomotor, der Joystick und das SD-Karten-Modul betrieben. Hierbei ist die Spannungsversorgung an VCC 5 V und an GND 0 V angeschlossen.

Für den Servomotor wird S1 des RJ25-Adapters mit dem Servo verbunden und auf der anderen Seite mit Pin 4 des Raspberry Pi Picos.

Für den Joystick müssen VRX und VRY an die analogen Pins des Raspberry Pi Picos angeschlossen werden. Das sind bei Pin 26 (VRX) und Pin 27 (VRY).

Bei dem SD-Karten-Modul werden die Pins 16 bis 19 für die SPI-Kommunikation verwendet.

4.2.3 Bauelemente der 3,3 V-Spannungsebene

Mit 3,3 V werden die ganzen Taster betrieben. Das betrifft also die Endlagentaster und die Taster des Joystick-Shields. Dabei muss analog zu der 5 V-Spannungsebene an VCC 3,3 V angeschlossen werden und an GND 0 V.

Die Endlagentaster sind mit dem Signalpin also S1 bzw. S2 des RJ25-Moduls mit den Pins 0 (Y-Richtung unten), 1 (Y-Richtung oben), 2 (X-Richtung rechts) und 3 (X-Richtung links) des Raspberry Pi Picos verbunden.

Das Steuerkreuz auf dem Joystick-Shield ist mit den Pins 10 - 13 verbunden und die zusätzlichen Taster auf diesem Shield, die nachfolgend für die Steuerung des Servos bzw. für die Umschaltung der Betriebsart verwendet werden, sind mit den Pins 20 (Servo) und 21 (Modus) verbunden.

4.3 Programmierumgebung

Zur Programmierung des Raspberry Pi Pico wird die Arduino IDE verwendet. Wie der Name bereits sagt, ist die Arduino IDE ursprünglich zum Programmieren von Mikrocontrollern der Firma Arduino gedacht. Damit mit dieser auch Mikrocontroller von Raspberry Pi programmiert werden können, müssen diese zunächst in der IDE hinzugefügt werden. Dazu wird die Github-Bibliothek von Earle Philhower genutzt.[7]

Um den Raspberry Pi Pico zur Arduino IDE hinzufügen zu können, muss in der Arduino IDE Datei → Voreinstellungen (siehe 7) geöffnet werden. In dem sich öffnenden Fenster muss bei „Zusätzliche Boardverwalter-URLs“ die folgende URL hinzugefügt werden:

https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

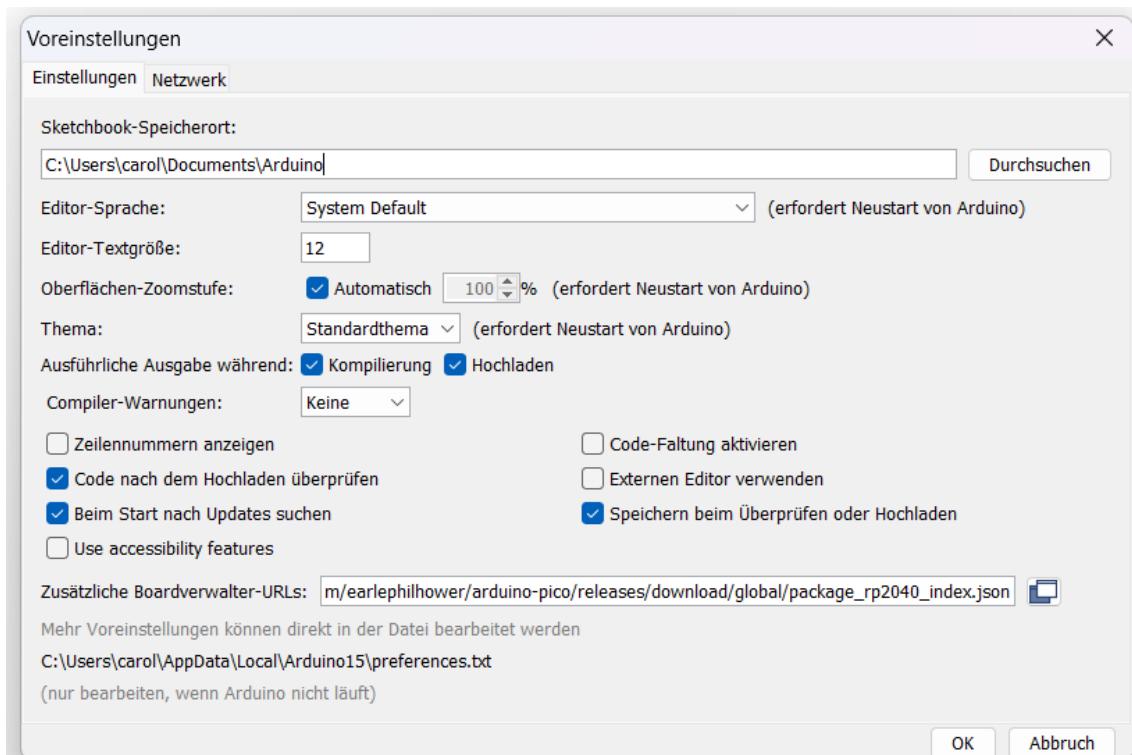


Abbildung 7: Hinzufügen des Raspberry Pi Pico Boards zur Arduino IDE in den Voreinstellungen

Danach kann unter Werkzeuge → Board → Boardverwalter nach dem Raspberry Pi Pico gesucht und dieses Board installiert werden (siehe Abbildung 8).

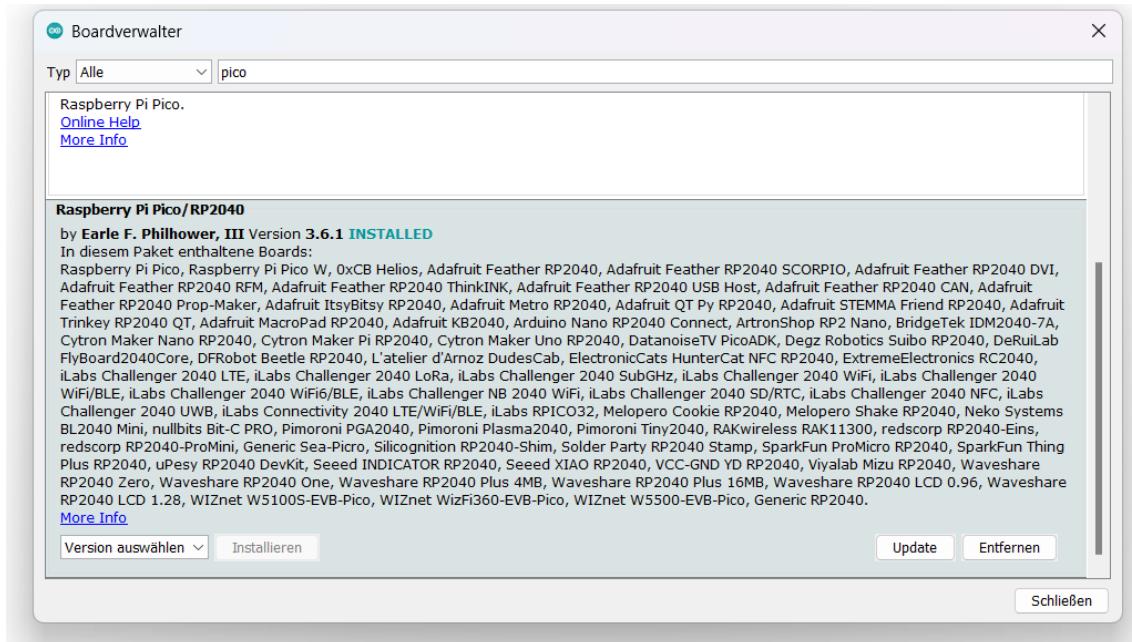


Abbildung 8: Hinzufügen des Raspberry Pi Pico Boards zur Arduino IDE im Boardverwalter

4.4 Bibliotheken

Um die Programmierung des Pen Plotters zu vereinfachen, werden einige fertige Funktionen aus Bibliotheken genutzt. Diese können in der Arduino IDE unter Werkzeuge → Bibliotheken verwaltet gesucht und installiert und danach in ein Programm eingebunden werden. In den folgenden Abschnitten werden die von uns in diesem Projekt genutzten Bibliotheken und die verwendeten Funktionen genauer betrachtet.

4.4.1 Servo.h

Diese Bibliothek erleichtert die Ansteuerung des Servo-Motors. Für diesen können Positionen mittels Pulsweitenmodulation festgelegt werden.

In Tabelle 3 sind die verwendeten Funktionen aus dieser Bibliothek aufgelistet und erläutert.[10]

Funktion	Erklärung
Servo	Constructor, Erstellt Objekt der Klasse Servo
attach	Zuweisung des Pins des Servos zusätzlich können minimale und maximale Pulsweite angegeben werden
writeMicroseconds	Ansteuerung des Servos mit der angegebenen Pulsweite

Tabelle 3: Genutzte Funktionen aus der Servo.h-Bibliothek

4.4.2 AccelStepper.h

Um die Schrittmotoren anzusteuern, werden die in Tabelle 4 aufgelisteten Funktionen der AccelStepper-Bibliothek benutzt. Der Schrittmotor benötigt pro Schritt einen Impuls. Für die Drehrichtung gibt es einen zusätzlichen Pin.[5]

Funktion	Erklärung
AccelStepper	Constructor, Erstellt Objekt der Klasse AccelStepper Angabe der genutzten Pins
run	Bewegung des Motors (Aufruf min. einmal pro Schritt)
setSpeed	gewünschte Geschwindigkeit in Schritte pro Sekunde
stop	Stoppen des Motors so schnell wie möglich
setCurrentPosition	Setzen der derzeitigen Position als 0 Position des Motors
moveTo	Einstellung der gewünschten absoluten Position

Tabelle 4: Genutzte Funktionen aus der AccelStepper.h-Bibliothek

4.4.3 MultiStepper.h

Diese Bibliothek stellt eine Erweiterung von der Bibliothek AccelStepper.h dar, die es ermöglicht, mehrere Schrittmotoren unabhängig von einander zu koordinieren.

4.4.4 SPI.h

Diese Bibliothek ist dafür da, dass die SPI-Kommunikation, also eine Art der seriellen Datenkommunikation, mit einem anderen Gerät hergestellt werden kann. In unserem Projekt ist dieses zweite Gerät das SD-Karten-Modul. [11]

4.4.5 SD.h

Die Bibliothek SD.h ermöglicht das Lesen und Schreiben einer Datei auf der SD-Karte. Dafür werden die in Tabelle 5 erläuterten Funktionen der Bibliothek verwendet.[9]

Funktion	Erklärung
begin	Initialisierung der SD Bibliothek und Karte Nutzung des SPI-Bus an den Pins 16 - 19
open	Öffnet eine Datei auf der SD-Karte
available	Überprüfung, ob noch Bytes zum Lesen vorhanden sind
close	Schließen der Datei
read	Speichern der geschriebenen Daten Lesen des nächsten Zeichens der Datei

Tabelle 5: Genutzte Funktionen aus der SD.h-Bibliothek

4.5 Programmaufbau

Das Programm soll die Ansteuerung der Schrittmotoren und des Servomotors in zwei verschiedenen Betriebszuständen realisieren.

Im manuellen Modus kann der Stift mithilfe eines Tasters abgesetzt und angehoben werden. Außerdem kann die Position des Stiftes wahlweise durch den Joystick gesteuert werden oder wenn dieser in Ausgangslage ist, kann der Stift durch das Steuerkreuz in geraden Linien bewegt werden. Durch die Stärke der Neigung kann bei der Verwendung des Joysticks auch die Geschwindigkeit der Schrittmotoren verändert werden. Der zweite Betriebsmodus ist der automatische SD-Karten-Modus. Bei diesem werden die Motoren automatisch gemäß der Datei auf der SD-Karte angesteuert (mehr dazu in Kapitel 5). Zwischen den Betriebsmodi lässt sich mithilfe eines zweiten Tasters wechseln.

Das Programm ist prinzipiell in die folgenden Abschnitte thematisch unterteilt:

1. Einbindung der benötigten Bibliotheken
2. Anlegen und Initialisierung der globalen Variablen
3. Herstellung der Ausgangsposition in `void setup`
4. Funktionalitäten des SD-Karten-Modus in `void loop`
5. Funktionalitäten des manuellen Modus in `void loop`
6. zusätzliche Funktionen `buttonToggle` und `goHome`

Dabei wird der Code in `void setup` nur einmalig bei jedem Start des Arduinos ausgeführt und der Code in `void loop` wird zyklisch ausgeführt.

4.5.1 Zusätzliche Funktionen

buttonToggle

Diese Funktion dient dazu, dass die Taster des Joystick Shields als Schalter genutzt werden können. Dabei wird sichergestellt, dass einmaliges Drücken auch nur als einmaliges Schalten erfasst, indem eine Flankenerkennung und Entprellen eingebaut ist. Bei diesem Code wird ein Beitrag aus einem Arduino-Forum genutzt[6] und dieser Code als Funktion umgeschrieben, da er an mehreren Stellen für zwei Taster benötigt wird.

goHome

Diese Funktion dient allein dazu den Pen Plotter zurück in die Ausgangsposition zu fahren und die Referenzposition neu zu setzen. Da diese Funktionalität an mehreren Stellen benötigt wird, wird sie in eine Funktionalität ausgelagert.

5 Aufbau der Datei auf der SD-Karte

Das Datenformat der Datei auf der SD-Karte entspricht G-Code, welchen wir im Programm „Inkscape“ erstellt und in eine txt-Datei kopiert wird. G-Code wird hauptsächlich für CNC-Fräsen und 3D-Drucker verwendet. Durch den einfachen Aufbau und die komfortable Erstellung ist er gut für dieses Projekt geeignet. Bei der Auswertung im Programm werden nur die Werte für X- und Y-Richtung genutzt, da der Pen Plotter über keine Z-Achse verfügt. Außerdem wird nach dem Befehl zu Beginn einer Zeile, ob bei der Ausführung dieser Zeile der Stift angehoben oder abgesetzt werden soll. Die Hauptbefehle im G-Code dafür sind G00 und G01. Bei G00 soll sich der Pen-Plotter ohne zu zeichnen bewegen und bei G01 soll der Pen-Plotter einen geraden Strich ziehen.

Das in diesem Projekt verwendete Programm zum Lesen der Datei ist selbst geschrieben, da keine für diesen Pen Plotter zugeschnittene Bibliothek zur Verfügung steht. Das Programm zum Lesen der Datei ist noch nicht vollständig entwickelt und es muss deswegen bei der Erstellung der Datei auf folgende Dinge geachtet werden:

- Es darf keine leeren Zeilen in der Textdatei geben. Diese sind rauszulöschen.
- Nach der Angabe des Y-Werts muss noch mindestens ein Leerzeichen kommen. Bei Zeilen, in denen keine weiteren Angaben stehen (Z-Wert o.ä.), ist also ein Leerzeichen am Ende der Zeile hinzufügen
- In jeder Zeile müssen Werte für X und Y stehen
- Es muss eine zusätzliche letzte (nicht leere) Zeile in der Textdatei geben. Diese wird nicht ausgeführt. Es kann zum Beispiel die erste Zeile der Datei wiederholt werden.

Hier ist als Beispiel der G-Code aufgeführt, den wir benutzt haben, um den das Rechteck mit dem Pen Plotter zu zeichnen.

Listing 1: Beispiel G-Code

```
G00 X29.552237 Y257.104477 \\  
G01 X136.309700 Y257.104477 Z-0.150000 F400.000000 \\  
G01 X136.309700 Y144.067170 Z-0.150000 \\  
G01 X29.552237 Y144.067170 Z-0.150000 \\  
G01 X29.552237 Y257.104477 Z-0.150000\\  
G00 X29.552237 Y257.104477
```

Für den G-Code ist diese Zeichnung das Ergebnis des Pen Plotters.

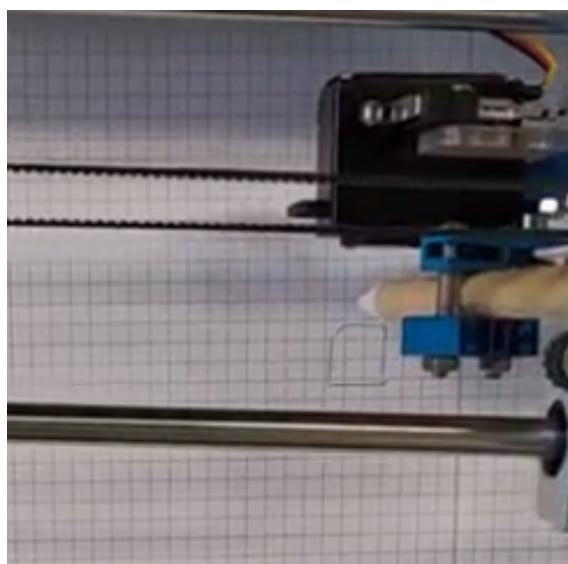


Abbildung 9: Vom Pen Plotter gemaltes Rechteck

6 Ausblick

Innerhalb der Projekts konnten zusammenfassend folgende Funktionalitäten umgesetzt werden.

- Ansteuerung der Schrittmotoren und des Servomotors mit dem Raspberry Pi
- Steuern des Pen Plotters mit Joystick und Steuerkreuz
- Auslesen einer Datei auf einer SD-Karte
- Ansteuern des Pen Plotters nach den G-Code-Koordinaten

Der Pen Plotter bietet weiterhin noch einige Verbesserungsmöglichkeiten, die aufgrund der begrenzten Zeit des Projekts nicht mehr bearbeitet werden konnten. In Rahmen weiterer Arbeiten können noch folgende Herausforderungen gelöst werden.

1. Das größte Problem ist, dass die Umschaltung von manuellen Modus auf SD-Modus nicht fehlerfrei funktioniert. Der Fehler könnte bei der Referenzpunktsetzung liegen. Selbiges Problem besteht genauso, wenn im automatischen Modus an die Kanten angestoßen wird. In dem Fall soll das Programm abgebrochen werden und neu begonnen werden. Dabei gibt es auch den Fehler, dass das Programm nicht wieder von vorne angefangen wird.
2. Das Zeichnen im SD-Modus ist darüber hinaus nicht sehr genau. Dies ist bei dem Zeichnen des Rechtecks deutlich zu erkennen, da eine Lücke in der Zeichnung bleibt und eine Ecke eher rund gezeichnet wird (siehe Abbildung 9), obwohl das im G-Code nicht so vorgesehen ist. Dies liegt zum einen mechanisch an dem Einspannen des Stifts und zum anderen an einer zu ungenauen Ansteuerung der Schrittmotoren.
3. Das Auslesen der SD-Karten-Datei könnte anders gestaltet werden, sodass weniger Anforderungen an das Format der Datei gestellt werden müssen.
4. Eine kleinere Verbesserung ist im Joystick-Modus möglich. Hier stimmt die Richtung in den man den Joystick bewegt nicht genau mit der tatsächlichen Bewegung des Pen Plotters überein.

Quellenverzeichnis

- [1] Reichelt Elektronik. *MB SMD V1 Makeblock - Me Schrittmotor Treiber V1*. URL: <https://www.reichelt.de/de/de/makeblock-me-schrittmotor-treiber-v1-mb-smd-v1-p202130.html?r=1> (besucht am 20. Feb. 2024).
- [2] Hitec. *General Servo Information*. URL: <https://hitecrcd.com/files/Servomanual.pdf> (besucht am 20. Feb. 2024).
- [3] Raspberry Pi Ltd. *Raspberry Pi Pico Datasheet. An RP2040-based microcontroller board*. 2023, S. 4.
- [4] Makeblock. *42BYG Stepper Motor*. URL: https://media.digikey.com/pdf/Data%20Sheets/Makeblock%20PDFs/81042_Web.pdf (besucht am 20. Feb. 2024).
- [5] Mike McCauley. *AccelStepper library for Arduino*. URL: <https://www.airspayce.com/mikem/arduino/AccelStepper/> (besucht am 20. Feb. 2024).
- [6] my_xyprojekt. *Arduino Taster als Schalter*. URL: <https://forum.arduino.cc/t/arduino-taster-als-schalter/677684> (besucht am 7. März 2024).
- [7] Earle Philhower. *Arduino-Pico*. URL: <https://github.com/earlephilhower/arduino-pico> (besucht am 20. Feb. 2024).
- [8] Makeblock Robotics. *How to Make Makeblock XY Plotter V2.0*. URL: <https://www.instructables.com/How-to-make-Makeblock-XY-Plotter-v20/> (besucht am 7. März 2024).
- [9] Arduino SRL. *SD*. URL: <https://www.arduino.cc/reference/en/libraries/sd/> (besucht am 7. März 2024).
- [10] Arduino SRL. *Servo*. URL: <https://www.arduino.cc/reference/en/libraries/servo/> (besucht am 20. Feb. 2024).
- [11] Arduino SRL. *SPI*. URL: <https://www.arduino.cc/reference/en/language/functions/communication/spi/> (besucht am 7. März 2024).