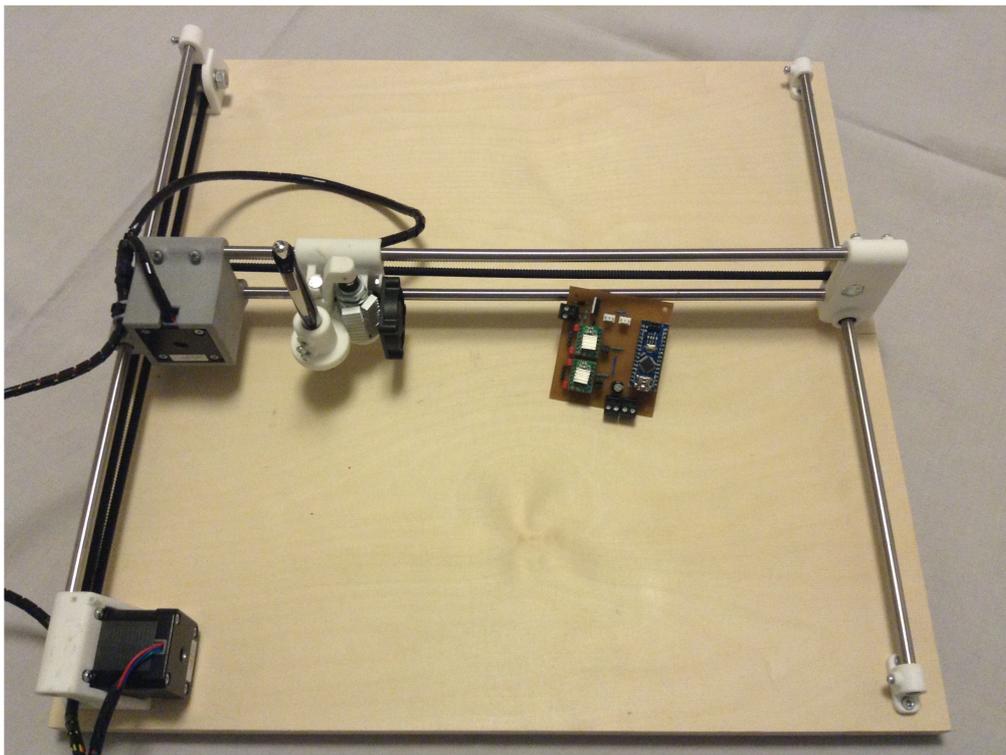


NWT PROJEKT

DOKUMENTATION

Thema:

„Kann man die eigene Handschrift mit Hilfe eines selbstgebauten Zeichen-Plotters imitieren?“



von:

Moritz Bauer, Tobias Bodentien,

Frieder Frank und Claudius Held

Inhaltsverzeichnis

Planungsphase	3
Beschreibung des Projekts.....	3
Vorbereitung.....	3
Arbeitspakete.....	3
Software	4
Ansteuerung des Plotters.....	4
<i>Firmware</i>	4
<i>G-Codes</i>	4
<i>Host-Programm</i>	5
Hauptprogramm.....	5
Elektronik	9
Ansteuern des Hubmagneten.....	10
Entwurf der Platine.....	10
<i>Schematische Zeichnung mit Eagle</i>	10
<i>Erstellen des Platinenlayouts</i>	11
Mechanik	12
Konstruktion.....	12
Aufbau des Plotters.....	13
<i>Bearbeitung der Holzplatte</i>	14
<i>Nachbearbeitung der Plastikteile</i>	14
Inbetriebnahme des Plotters	16
Projektabschluss	17
Mögliche Verbesserungen bis zur Präsentation:.....	17
Kosten.....	17
Danksagung	18
Literatur- und Quellenverzeichnis	19

Planungsphase

Beschreibung des Projekts

„Kann man die eigene Handschrift mit Hilfe eines selbstgebauten Zeichen-Plotters imitieren?“

Unser NWT Projekt befasst sich mit dem Entwickeln, Konstruieren und Bauen eines Zeichen-Plotters und der Erstellung eines geeigneten Software-Arbeitsablaufes, um die eigene Handschrift vom Plotter imitieren zu lassen.

Die Projektphase beginnt am 23.03.2015, die Dokumentation muss am 22.06.2015 abgegeben werden. Ab dem 06.07.2015 finden die Präsentationen und Kolloquien der einzelnen Projektgruppen statt. Ein Gesamtbudget von 20€ darf für das Projekt nicht überschritten werden.

Vorbereitung

Zu Beginn der Projektphase hat sich das gesamte Team mögliche Arbeitspakete überlegt und auf die Mitglieder verteilt. Zudem wurde ein vorläufiger Zeitplan mit mehreren Meilensteinen erstellt.

Die einzelnen Meilensteine sind wie folgt gesetzt:



Arbeitspakete

Die Arbeitspakete sind im Grundlegenden in 3 Kategorien eingeteilt:

- Elektronik → Moritz Bauer, Tobias Bodentien
- Software → Claudius Held
- Mechanik → Frieder Frank

Im Anhang ist ein Ablauf der einzelnen Arbeitspakete zu sehen.

Software

Ansteuerung des Plotters

Zu dem Arbeitspaket Software mussten wir uns zuerst die Frage stellen, wie wir den Plotter ansteuern wollen.

Firmware

Die Firmware ist das Programm, das auf den Arduino geladen wird, und die von uns gesendeten G-Codes (siehe Abschnitt „G-Code“) in Signale für die Stepper-Motoren umwandelt. Hierzu haben wir uns verschiedene Möglichkeiten angeschaut und sind zu dem vorübergehenden Entschluss gekommen, die Firmware „grbl“ zu verwenden. Sie ist komplett Open-Source. D.h. sie ist Quelloffen und kann von jedem frei im Internet heruntergeladen und anschließend verändert werden.

G-Codes

Bei „G-Codes“ handelt es sich um eine Sammlung verschiedener Befehle die von einem Computer an eine Fertigungsmaschine gesendet werden können. Hierbei wird nach einem G oder M Befehl, der angibt was die entsprechende Code-Zeile bewirken soll, zum Beispiel zwei Koordinaten angegeben, zu denen sich das Werkzeug (In unserem Fall ein Stift) bewegen soll. Beispiel:

G71	/Die Maschine bewegt sich im metrischen Koordinatensystem
G1 X100 Y130.5	/Das Werkzeug geht zur Position X=100 und Y=130.5 (mm)
G70	/Die Maschine bewegt sich im imperialen Koordinatensystem
G1 X2 Y5	/Das Werkzeug geht zur Position X=2 und Y=5 (Zoll)

Die Firmware mussten wir nun noch auf unseren Plotter, und vor allem auf unsere Elektronik anpassen. Z.B. mussten die Pins definiert werden, die den Motoren die Richtung angeben usw. Zunächst hatten wir hierbei ein Problem, da wir nicht wussten, wie wir den Magneten, der den Stift absetzen soll, ansteuern sollen. Da die Firmware „grbl“ eigentlich für CNC-Fräsen ausgelegt war, lösten wir das Problem, indem wir den Ausgang für das „Motor-An“ und „Motor-Aus“ für die Frässpindel benutzten, um den Magneten an- und abzuschalten.

Host-Programm

Um erstellte G-Codes an den Arduino zu senden, benötigten wir noch ein sog. Host-Programm. In dieses Programm kann man eine erzeugte G-Code-Datei laden, welche dann in kleinen Paketen an den Arduino gesendet wird.

Für unsere Firmware wurde im Internet das Programm „Universal-Gcode-Sender“ empfohlen. Es ist zudem in Java programmiert, was es Plattformunabhängig macht.

Hauptprogramm

Um den Plotter später zu testen, werden wir sogenannte „Singleline-Fonts“ verwenden. Als „Font“ bezeichnet man im allgemeinen eine Schriftart, die in einer Datei „gespeichert“ ist und mit der man, wenn man sie in ein Textbearbeitungsprogramm (z.B. „Word“) einfügt, schreiben kann. Es gibt verschiedene Arten von Fonts. Wir haben „Outline-Fonts“ und „Singleline-Fonts“/„Midline-Fonts“ betrachtet. Sie unterscheiden sich darin, dass Outline-Fonts die Außenlinien der Buchstaben abbilden und Singleline-Fonts nur eine Linie, die meistens auch die Mittellinie ist.

Wir haben uns für Singleline-Fonts entschieden, da diese mit einem Stift besser dargestellt werden können, und, vom Plotter geschrieben, realistischer aussehen.

ABCabc123

Midline-/Singleline-Font

ABCabc123

Outline-Font

Bei der Software haben sich im Laufe des Projekts einige Probleme und daraus folgende Umstrukturierungen ergeben. Das Programm, welches die G-codes aus den Fonts hätte extrahieren sollen, ist leider nicht komplett kostenlos. Die Komplett-Version würde 99\$ kosten, was unser Budget deutlich überschreiten würde. Auf unsere Nachfrage, ob wir eine Lizenz für schulische Zwecke erhalten könnten, erhielten wir keine Antwort. Somit mussten wir uns etwas neues überlegen und entschieden uns Inkscape mit einer G-Code Erweiterung zu verwenden.

Mit dieser Inkscape Erweiterung lassen sich Texte, die in allen in Windows installierten Schriftarten geschrieben werden können, in G-Codes umwandeln. Hier tritt das nächste Problem auf: In Windows lassen sich keine Singleline-/MidlineFonts installieren. Somit kann man mit diesen in Inkscape auch nicht schreiben.

Wir entschieden uns dieses Problem erst einmal beiseite zu legen und Outline-fonts zu benutzen. Mit Outline-Fonts und Inkscape konnten wir recht leicht die G-Codes jeglicher Schrift erhalten.

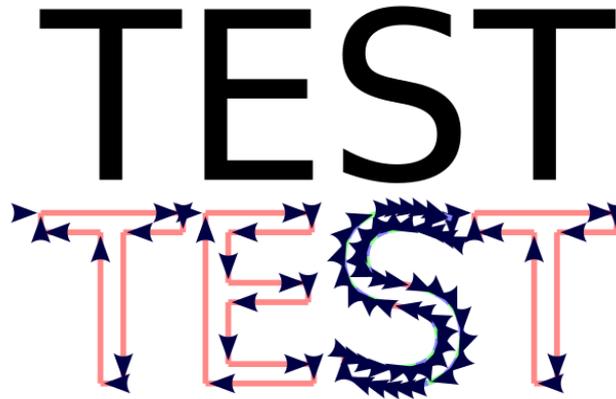


Abbildung 1: Text mit visualisierten Verfahrenswegen

Diese G-Codes haben nur ein Problem: sie sind nicht auf unsere Firmware des Arduinos angepasst. Es fehlen die von uns für das Aufsetzen und Abheben des Stifts benötigten „M3“ und „M4“ Befehle. Inkscape liefert im G-Code aber an manchen Stellen in Klammern stehende Kommentare wie z.B. „(Penetrate)“. Diese werden immer vor und nach dem Code-Abschnitt ausgegeben, der später geschrieben/gezeichnet werden soll. So kann das Einfügen der „M3“ und „M4“ Befehle von Hand gemacht werden, was aber schon bei einzelnen Wörtern sehr lange dauern kann. Somit entschieden wir uns ein Programm zu schreiben, das uns diese Befehle automatisch einfügt.

Das Programm soll auf die erstellte G-Code-Datei zugreifen und nach den von Inkscape in Klammer angegebenen Begriffen suchen. Hierbei gibt es einmal „(Penetrate)“, bei dem der Plotter den Stift auf das Blatt setzen sollte (M4) und „(Change tool to default tool)“ oder „(end cutting Path)“, bei welchem der Stift abgehoben werden sollte (M3). Das Script sucht dabei einfach die einzelnen Zeilen des G-Codes nach diesen Begriffen ab und fügt, wenn einer gefunden wurde, den entsprechenden Befehl ein. Dies funktionierte auf Anhieb recht gut und nachdem ein Fehler behoben war, der dafür sorgte, dass nicht vor jedem „(Start cutting path)“ ein „M4“ Befehl eingefügt wird sogar fehlerfrei.

Somit konnten wir jetzt ganze Sätze in unserer eigenen, oder jeder beliebigen in

Windows installierbaren Schrift schreiben. Das einzige Problem dabei ist, dass dies immer noch Outline-Fonts sind und somit die Schrift nicht sonderlich schön aussieht. Auch im Internet zu Verfügung gestellte, angebliche „Singleline-Fonts“, die sich in Windows installieren lassen, entpuppten sich als sehr dünne Outline-Fonts.

Da unser ursprüngliches Ziel damit aber noch nicht erreicht war versuchten wir eine Alternative zu finden, die gar keine Fonts mehr benutzt. Diese bot uns die Inkscape Erweiterung. Diese wandelt in Inkscape gezeichnete Formen in G-Codes um. Wir zeichneten jeden Buchstaben unserer Handschrift in Inkscape nach und speicherten G-Codes für diesen. Diese Buchstaben setzten wir immer an den Nullpunkt. Somit erhielten wir für jedes einzelne Zeichen, das in unserer Schrift geschrieben werden soll, ein Textdokument mit den zugehörigen G-Codes. Die Überlegung dahinter war, ein Programm zu schreiben, das die jeweiligen G-Codes nach Wunsch zusammenfügt. Somit sollten, wenn man die Buchstaben AB schreiben wollte, die G-Codes des Buchstaben B nach denen des Buchstaben A ausgegeben werden. Hier entstand ein weiteres Problem. Die G-Code mussten, abhängig von der jeweiligen Position, die der Buchstabe einnehmen sollte, verändert werden. Wir hatten 2 Möglichkeiten, die erste war, jeden einzelnen G-Code Befehl auf die Position, an der der Buchstabe starten sollte, aufzuaddieren und dadurch jeden Buchstaben an jeglicher Position schreiben zu können. Die zweite war, nach jedem Buchstaben, der geschrieben wurde an den Punkt zu fahren, an dem der nächste Buchstabe ansetzen sollte und dort den Nullpunkt neu auf die aktuelle Position zu definieren. In Anbetracht des Aufwandes, der mit der ersten Möglichkeit verbunden ist, entschieden wir uns für die Zweite.

Hierzu benötigten wir einen Befehl, der den Nullpunkt neu definiert, welcher nach kurzem Suchen gefunden war. Laut der Internetseite der Firmware macht der Befehl „G92“ genau dies: er definiert die Position des Stiftes neu. Hierbei kann mit den hinter dem „G92“ Befehl angegebenen X und Y Werten bestimmt werden, an welcher „neuen“ Position sich der Stift befinden soll. Für uns sieht dieser Befehl dann so aus: „G92 X0 Y0“. Die aktuelle Position wird als neuer Nullpunkt im Koordinatensystem gesetzt.

Das dazugehörige Programm ist alles in allem recht einfach. Es liest aus einem Textdokument, in welchem die Eingabe des zu schreibenden Textes steht (zB „Hallo“), das erste Zeichen aus und hängt den Inhalt des diesem Zeichen zugeordneten Textdokuments in ein Ausgabe Textdokument an. Nun soll vom Endpunkt des Buchstabens, also der aktuellen Position, ein Stück nach rechts gegangen werden, um den nächsten Buchstaben zu schreiben. An dieser Position wird nun der neue Nullpunkt gesetzt, damit dort der nächste Buchstabe ansetzen kann. Danach wird das erste Zeichen der Eingabe gelöscht. Jetzt setzt das Programm wieder vorne an und fragt den jetzt ersten Buchstaben der Eingabe ab. Dies alles wird solange wiederholt, bis keine Zeichen mehr in der Eingabe vorhanden sind.

Dieses Programm ist bis zum aktuellen Zeitpunkt nicht komplett fertig gestellt, da es ein sehr großer Aufwand ist für jeden einzelnen Buchstaben einen G-Code zu erstellen und diesen in den Programmcode einzufügen. Das Grundgerüst steht und 4 Buchstaben können bisher geschrieben werden. Bis zum endgültigen Präsentationstermin des Projekts sollte aber zumindest eine Handschrift vollständig integriert sein.

Elektronik

Zu Beginn haben wir uns überlegt, welche Motoren für die Ansteuerung des Plotters am geeignetsten sind. Die Wahl war schnell getroffen:

Stepper-Motoren (auch „Schrittmotor“).

Sie werden im Gegensatz zu handelsüblichen DC-Motoren über einen speziellen Treiber (wir haben uns für einen Pololu a4988 entschieden) an den Mikrocontroller angeschlossen, da die Motoren viel mehr Strom brauchen, als die Pins des Mikrocontrollers liefern. Sie wandeln vom Mikrocontroller kommende „High“ und „Low“ Signale in Schritte um. Mit jedem Schritt bewegt sich dann der Stepper Motor um 0.8° und bleibt danach an genau dieser Stelle stehen. Somit haben wir die Möglichkeit die Achsen und somit auch den Stift sehr genau zu bewegen.

Auf einem Breadboard haben wir 2 Schrittmotor-Treiber (a4988), die wir freundlicherweise, zusammen mit einem „Arduino Due“ bereitgestellt bekommen haben, mit einem (später mit zwei) NEMA 17 Schrittmotoren verkabelt. Diese hatte Frieder noch von seinem 3D-Drucker übrig, sodass sie sozusagen nichts gekostet haben. Im Internet fanden wir auch schnell einen Schaltplan zu den verwendeten Bauteilen. Diesen setzten wir dann auf dem Breadboard um:

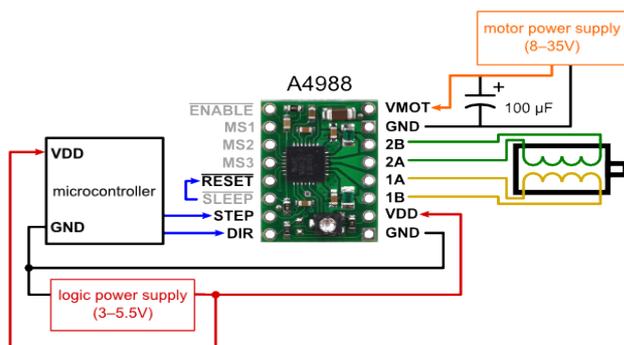


Abbildung 3: Schaltplan für Stepper-Treiber

Quelle: http://fritzing.org/media/fritzing-repo/projects/a/a4988-single-stepper-test/images/Arduino%20A4988%20Single%20Stepper%20Wiring_bb.jpg

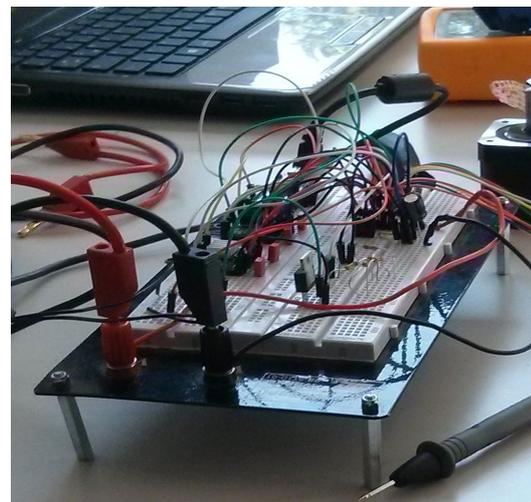


Abbildung 2: Schaltung auf Breadboard

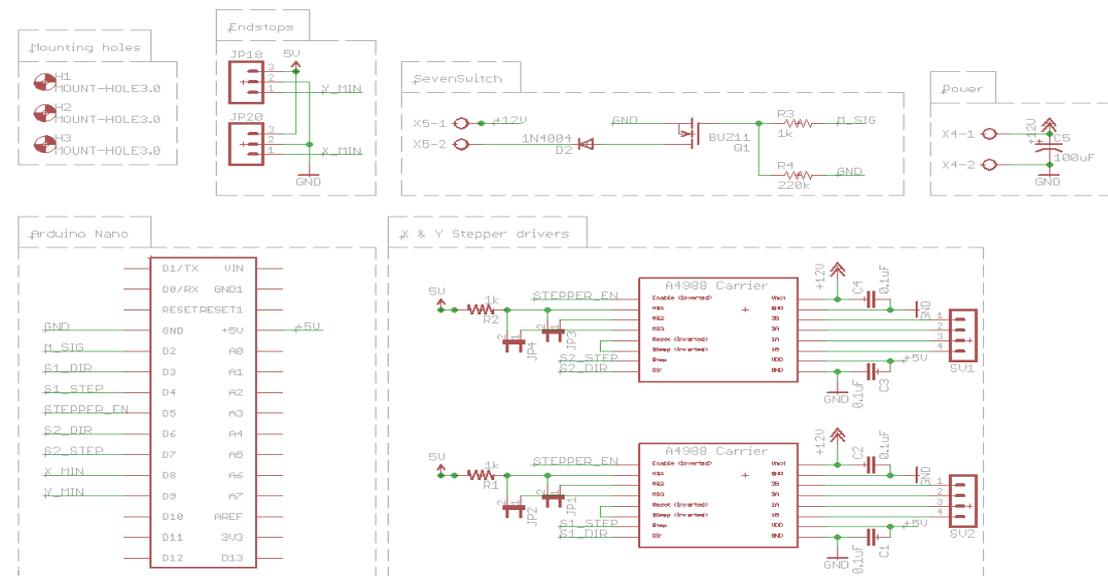
Ansteuern des Hubmagneten

Der Hubmagnet, welcher den Stift beim Schreiben anhebt und absenkt, wird mit einem MOSFET angesteuert. Ein MOSFET ist ein aktives Bauelement mit mindestens drei Anschlüssen (Elektroden): G (gate, dt. *Steuerelektrode*), D (drain, dt. *Abfluss*), S (source, dt. *Quelle*). Wie andere Feldeffekttransistoren wirkt der MOSFET wie ein spannungsgesteuerter Widerstand. Somit konnten wir den Stromkreis des Magneten mit dem Arduino steuern. Wir benutzen den BUZ11, welchen wir von Herrn Lange zur Verfügung gestellt bekamen. Was die Schaltung anging half er uns auch.

Entwurf der Platine

Wir wählten das Programm „Eagle“ um unsere Platine zu entwerfen und machten uns mit diesem vertraut, indem wir diverse „how-to“ Videos zu dem Programm auf Youtube anschauten. Am meisten konnten wir mit dem des Kanals „The Ben-Heck Show“ anfangen und fanden uns somit schnell in Eagle zurecht.

Schematische Zeichnung mit Eagle



Der erste Schritt war, eine schematische Zeichnung der Platine zu entwerfen. Nachdem man alle elektronischen Bauteile, die später auf der Platine sein sollen hinzugefügt hat, muss man die einzelnen Pins der Bauteile benennen. Pins, die später durch Leiterbahnen miteinander verbunden sein sollen, werden mit dem gleichen Namen benannt. In diesem Arbeitsschritt wird ausschließlich die Verbindung der Pins beachtet, um eine funktionierende Schaltung zu gewährleisten. Dies war der kleinere Teil beim Entwurf der Platine.

Erstellen des Platinenlayouts

Nachdem wir dann den Schaltplan fertig gestellt haben, wechselten wir auf die „Board“ Ansicht von Eagle. In dieser Ansicht haben wir die zusammengehörigen Bauteile mit Leiterbahnen verbunden. Dieses Layout schickten wir zu Herrn Juchem, um es fräsen zu lassen. Fräsen hat den großen Vorteil, dass die Fräse sehr genau arbeiten kann und keine Nachbearbeitung nötig ist. Beim Ätzen befürchteten wir, dass wir keine guten Ergebnisse bekommen würden, da wir es selber noch nie gemacht haben (außer einmal im NWT-Unterricht). Zusätzlich gehören beim Ätzen auch das Vorbereiten und Entsorgen der Ätzmittel dazu, sowie das Reinigen der benötigten Werkzeuge. Wir meinen, dass sich Ätzen nur lohnt, wenn man mehrere Platinen herstellt. Als wir die fertige Platine bekommen haben mussten wir nur noch die Bauteile drauf löten.

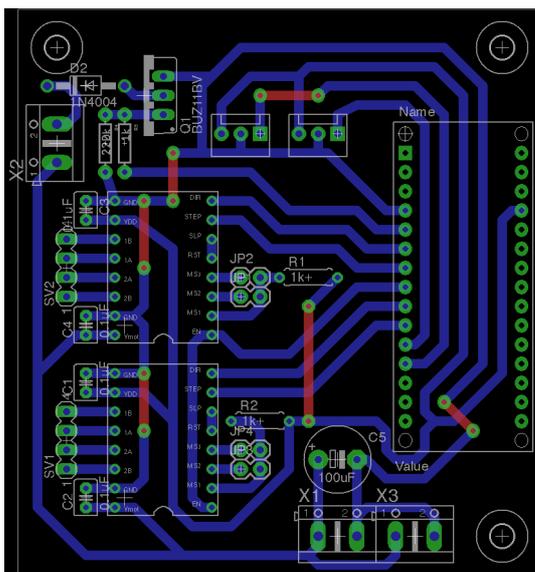


Abbildung 5: Layout der Platine

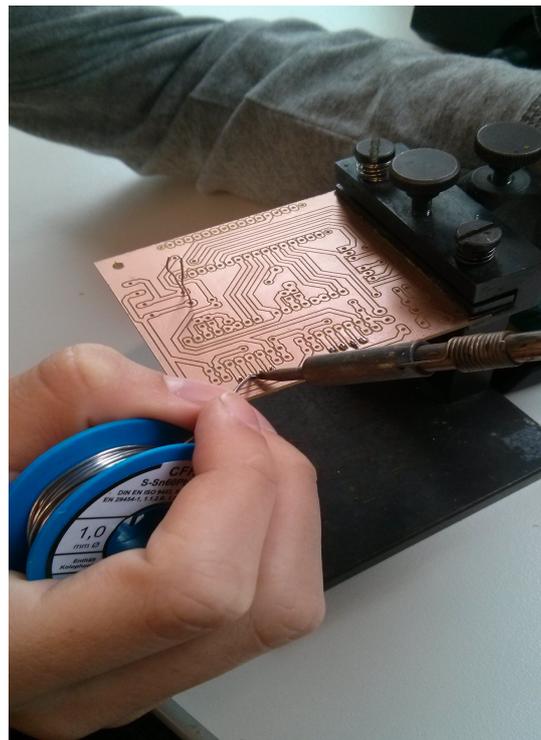


Abbildung 4: Löten der Bauteile

Mechanik

Herkömmliche Plotter haben 2 Achsen, die im 90° Winkel zueinander ausgerichtet sind (kartesische Bauweise). Unseren Plotter wollten wir auf die selbe Weise realisieren, da diese Bauart viele Vorteile mit sich bringt:

- Die Mechanik ist sehr einfach zu durchschauen
- Man benötigt nur 2 Motoren
- Die Ansteuerung und Berechnung der Fahrwege können sehr einfach realisiert werden.

Konstruktion

Die Konstruktion des Plotters soll komplett am Computer statt finden. Mit einem sog. CAD(„Computer Aided Design“)-Programm (in unserem Fall „Solidworks“) können Bauteile virtuell erstellt werden. Des weiteren kann ein Aufbau des gesamten Plotters und jeder einzelnen Schraube simuliert werden. Dadurch erhoffen wir uns eine große Zeitersparnis beim Aufbau und der anschließenden Inbetriebnahme des Plotters. Den Schwerpunkt legen wir hierbei auf eine geringe Größe und wenige, einfach zu verbauende Teile. Dadurch ergeben sich allerdings andere Probleme: Die meisten Teile sind komplett individuell für unseren Plotter und können mit unseren Mitteln nicht hergestellt werden.

Um dieses Problem zu umgehen hat sich die Firma „Jomatik“ aus Tübingen bereiterklärt, die Bauteile mit einem professionellen 3D-Drucker herzustellen. Hierbei wird das SLS-Verfahren (Selektives Laser Sintern) angewendet, welches sehr genaue und stabile Bauteile aus Plastik (PA12) verspricht.

Das Modell des Plotters wurde am 11. Mai 2015 fertiggestellt. Da Frieder vor diesem Projekt noch nicht so viel Erfahrung mit dem CAD-Programm gesammelt hat, konnte er die Zeit, die er für das Modell benötigte nur schwer einschätzen, weshalb der Zeitplan am Ende von ihm nicht eingehalten werden konnte. Des weiteren stellte sich die Konstruktion von mehreren Bauteilen, die zudem noch aufeinander abgestimmt sein müssen als sehr aufwendig heraus.

Aufbau des Plotters

Der Plotter ist wie folgt aufgebaut:

Auf einer Basisplatte aus Holz werden vier Bauteile befestigt. An diesen sind die Führungswellen für die Y-Achse befestigt. Auf einer Seite sind zudem noch Halterungen für einen Motor, bzw. für ein Radial-Kugellager. Sie werden benötigt um später einen Riemen antreiben bzw. umlenken zu können, der die X-Achse (und deren Motor), in Y-Richtung verschiebt. Der X-Motor wird mit zwei Linear-Kugellagern an der Führungswelle geführt. An ihm sind wiederum zwei Führungsachsen für den Stift-Schlitten befestigt. Der Stift wird mithilfe eines Zug-Magneten auf das Papier gedrückt; eine Druckfeder drückt ihn, bei ausgeschaltetem Magneten in seine Ausgangsposition.

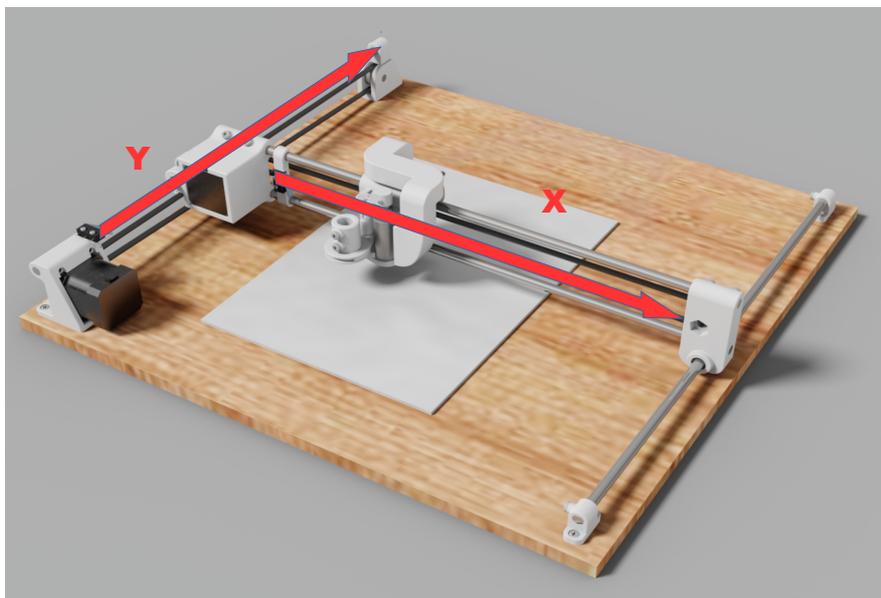


Abbildung 6: Rendering des Plotters

Nachdem das Modell am Computer fertiggestellt wurde, ließen wir die Bauteile innerhalb von 2 Wochen für uns fertigen. Während dieser Zeit kümmerten wir uns um die Basisplatte, auf der später alle Bauteile befestigt werden sollten. Hierbei hat uns der Schreinermeister Uwe Jansen unter die Arme gegriffen. In seiner Werkstatt durften wir alle benötigten Werkzeuge verwenden um höchste Maßhaltigkeit beim Bearbeiten der Holzplatte zu gewährleisten

Bearbeitung der Holzplatte

Auf einer genau zugeschnittenen Birken-Multiplex-Platte mit 18mm Dicke, von welcher wir uns eine sehr hohe Stabilität erhofften, haben wir zunächst mit Hilfe der technischen Zeichnungen, die uns das CAD-Programm ausgab, die Bohrungen angezeichnet. Nach dem Bohren der Löcher versuchten wir sogenannte Gewindeeinsätze in die Löcher zu setzen. Vorteil von ihnen ist, dass man die Bauteile mit Schrauben direkt in dem Holz befestigen kann, ohne z.B. auf der Rückseite Muttern anbringen zu müssen. Außerdem wirkt es optisch schöner und „aufgeräumter“. Zum einsetzen der Gewindeeinsätze schraubten

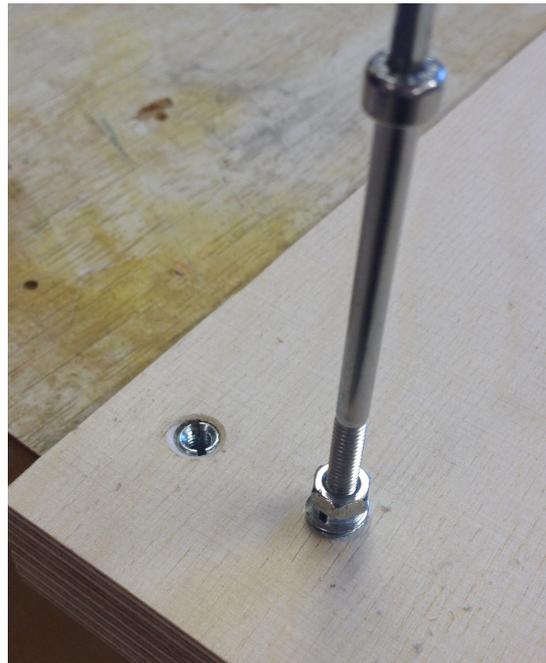


Abbildung 7: Einsetzen der Gewindeeinsätze

wir eine Schraube in das innere Gewinde, konterten mit einer Mutter, um ein raus- oder reindreihen der Schraube zu verhindern, und drehten alles vorsichtig in das Bohrloch. Nach etwa 2 Stunden Arbeit war die Basisplatte fertiggestellt.

Nachbearbeitung der Plastikteile

Nach ca. 1 weiteren Woche waren die gedruckten Bauteile fertig. Sie benötigten allerdings auch noch eine weitere Nachbearbeitung. Im Computer wurden alle Bohrungen für die 8mm-Führungsstangen auch mit 8mm konstruiert, obwohl sonst (zumindest bei diesem Herstellungsverfahren) eine Toleranz von ca. 0,1mm gegeben werden sollte. Somit konnte man in der Nachbearbeitung mit einer Reibahle eine höhere Genauigkeit der Bohrung erreichen, was einen festeren Sitz der Führungsstange zur Folge hatte.



Abbildung 8: Aufreiben der Bohrung mithilfe einer Reibahle

Danach mussten auch bei den Plastik-Bauteilen Gewindeeinsätze eingesetzt werden. Allerdings war hier das Außengewinde nicht für Holz sondern auch metrisch. Die Gewindeeinsätze haben hier den Zweck bei mehrmaligem ein- und ausdrehen der Schrauben einen Verschleiß oder Abrieb des Plastiks zu verhindern.

Beim Einsetzen der Gewindeeinsätze gingen wir vor wie bei den Holzgewindeeinsätzen, mit dem einzigen Unterschied, dass wir ein Gewinde in das Plastik vorgeschritten haben.



Abbildung 9: Gewindeschneiden



Abbildung 10: Gewindeeinsatz einsetzen

Inbetriebnahme des Plotters

Ab dem 13. Juni 2015 haben wir anschließend versucht den Plotter mit all seinen Komponenten in Betrieb zu nehmen. Hierzu verkabelten wir erst die einzelnen elektronischen Bauteile mit der Platine. Als Stromquelle diente uns ein altes Computernetzteil. Dann musste der Arduino mit dem Computer verbunden werden. Da wir allerdings einen Arduino-Klon (also einen Nachbau) besaßen, mussten wir erst einen bestimmten Treiber installieren. Dies ging jedoch ohne Probleme.

Nachdem wir den Plotter einschalteten und den Computer mit dem Plotter verbunden hatten, konnten wir schon die ersten manuellen Bewegungen ausführen. Bei dieser Gelegenheit stellten wir auf den Treibern den benötigten Motor-Strom ein und definierten die „Steps per mm“. Diese werden benötigt, dass die Firmware die Fahrwege richtig auf die Motor-Schritte umrechnen kann. Hierfür ließen wir den Plotter mit folgendem G-Code ein einfaches Quadrat mit Kantenlängen von 50mm zeichnen:

```
G1 X0 Y0 F500 // Plotter an Ursprung
G1 X50 // Plotter auf X-Achse um 50mm nach rechts
G1 Y50 // Plotter auf Y-Achse um 50mm nach oben
G1 X0 // Plotter auf X-Achse um 50mm nach links
G1 Y0 // Plotter auf Y-Achse um 50mm nach unten
```

Nun mussten wir die gezeichneten Kanten messen. Durch den SOLL-IST Vergleich konnten wir die „Steps per mm“ neu einstellen.

Anschließend ließen wir unsere ersten Tests zeichnen (siehe Anhang).

Hierbei fiel uns immer wieder auf, dass der Magnet extrem heiß wurde (anfassen war nur für sehr kurze Zeit möglich). Um dieses Problem zu mindern, klebten wir mit einem Wärmeleitkleber mehrere Aluminium-Kühlkörper an den Magneten. Diese sollten die Kühlleitung verstärken.

Ebenfalls sehr auffällig ist, dass der Stift beim Absetzen und anschließenden Verfahren sehr stark wackelt. Das Bauteil für die Stift-Halterung lässt sich jedoch aus Mangel an Zeit nicht verbessern. Zudem hat die Firma in nächster Zeit keine Möglichkeit solche großen Teile zu drucken. Beim schreiben der Handschrift sollte ein leichtes Wackeln allerdings nicht groß auffallen.

Projektabschluss

„Kann man die eigene Handschrift mit Hilfe eines selbstgebaute Zeichen-Plotters imitieren?“

Schlussendlich können wir sagen, dass das Projekt zum größten Teil erfolgreich durchgeführt wurde. Der Bau des Plotters an sich war zwar eine große Herausforderung, konnte aber, auch wenn der Zeitplan nicht vollständig eingehalten werden konnte, abgeschlossen werden. Die Elektronik hat größtenteils problemlos funktioniert, was auf eine sehr gute Planungs- und Testphase zurückzuführen ist. Kleinere Fehler, z.B. dass der USB Stecker leicht an eine Leiterbahnklemme drückt, beeinflussen nicht die Funktion, mindern aber die Benutzerfreundlichkeit minimal.

Die Software stellte bei dem Projekt eindeutig eines der größten Probleme dar, da es noch so gut wie keine Informationen zu diesem Vorhaben gab. Die Idee zur möglichen Umsetzung kam leider zu spät, sodass keine wirklichen Ergebnisse zur Abgabe der Dokumentation vorgezeigt werden können. Das Arbeitspaket zur Software hatte zudem eine erhöhte Schwierigkeit, da es auf die Ergebnisse der anderen Arbeitspakete/-gruppen zurückgreifen musste, und so erst spät anfangen konnte.

Auch wenn die Handschrift noch nicht so imitiert werden kann, wie wir es uns am Anfang vorgestellt hatten, können wir mit dem Plotter Zeichnungen anfertigen.

Mögliche Verbesserungen bis zur Präsentation:

Software: Einbindung der eigenen Handschrift auf die Art, wie sie im Abschnitt „Software“ erklärt wurde.

Elektronik: Bau eines Elektronik-Gehäuses für ein ordentliches Kabel-Management

Mechanik: Konstruieren einer stärkeren Stift-Halterung für genauere Zeichnungen

Kosten

Die größten Kosten des Projekts sind – wie erwartet – bei der Mechanik entstanden. Allerdings belaufen sich die Kosten auch nur auf 15€ für den Magneten und die Feder, die den Stift nach oben drückt. Der Rest wurde von verschiedenen Personen oder Firmen zur Verfügung gestellt. Somit haben wir das vorgegebene Budget nicht überschritten.

Danksagung

Wir haben für unser Projekt Hilfe von vielen Personen und Firmen bekommen, welchen wir an dieser Stelle unseren Dank aussprechen wollen:

Vielen Dank an...

- **Markus Hitter** (alias „Traumflug“), der uns über das „RepRap“-Forum viele Tipps zu Firmware und Elektronik gegeben hat.
- **Uwe Jansen**, der uns zum einen das Holz und zum anderen seine Werkstatt zur Bearbeitung der Platte zur Verfügung stellte
- Die Firma **Jomatik** und Herrn **Johannes Matheis**, durch die wir die 3D-gedruckten Plastikteile zur Verfügung gestellt bekommen haben.
- **Philipp ten Brink** (alias „Charly148“) der uns (ebenfalls über das „RepRap“-Forum) die beiden Motortreiber und einen Arduino Due zur Verfügung stellte (den wir aber leider nicht verwenden konnten).

Literatur- und Quellenverzeichnis

Für unser gesamtes Projekt haben wir sehr wenige Quellen benutzt. Dies hat folgende Gründe:

- Die meisten Dinge kannte Frieder schon von seinen 3D-Druckern. So hatte er z.B. Kenntnisse über die Elektronik, die Firmware zur Ansteuerung des Plotters und die Mechanik.
- Es gibt keine nützlichen Quellen zur Konstruktion des Plotters. Die einzigen Quellen stellen hier die Technischen Zeichnungen und Datenblätter der Standard-Bauteile dar.
- Claudius musste für das Projekt nichts neues lernen. Sein bisheriges Wissen zur Programmierung verschiedener Programme reichte aus. Außerdem wurde nach unserem Wissen kein derartiges Projekt, das die Handschrift mit einbezieht, veröffentlicht.

Allerdings haben wir uns auf das Wissen mehrerer Personen gestützt:

- Markus Hitter, den wir über das „RepRap“-Forum kennen gelernt haben, hat uns in Sachen Elektronik und Software viele Tipps gegeben. Unter Anderem haben wir sein „Gen7“ Board und die „SevenSwitch“ als Inspiration für unsere Platine verwendet.
- Johannes Matheis, der Geschäftsführer der Jomatik GmbH, hat bei der Konstruktion des Plotters vor allem bei Toleranzen etc. Tipps gegeben.
- www.reprap.org (18.4.2015). HITTER, M.: Generation 7 Electronics.
Eingestellt: 10. Feb 2011. http://reprap.org/wiki/Generation_7_Electronics
- www.reprap.org (18.4.2015). HITTER, M.: SevenSwitch 1.2.
Eingestellt: 8. August 2013. http://reprap.org/wiki/SevenSwitch_1.2
- www.hobby-bastelecke.de (6.6.2015). LEHWALD, M: Richtiges Löten.
<http://www.hobby-bastelecke.de/basteltips/loeten.htm>
- www.autoitscript.com (27.4.2015).
<https://www.autoitscript.com/autoit3/docs/functions.htm>
- www.inkscape.org (18.4.2015). <https://inkscape.org/de/>
- wiki.inkscape.org (18.4.2015).
http://wiki.inkscape.org/wiki/index.php/Extension_repository