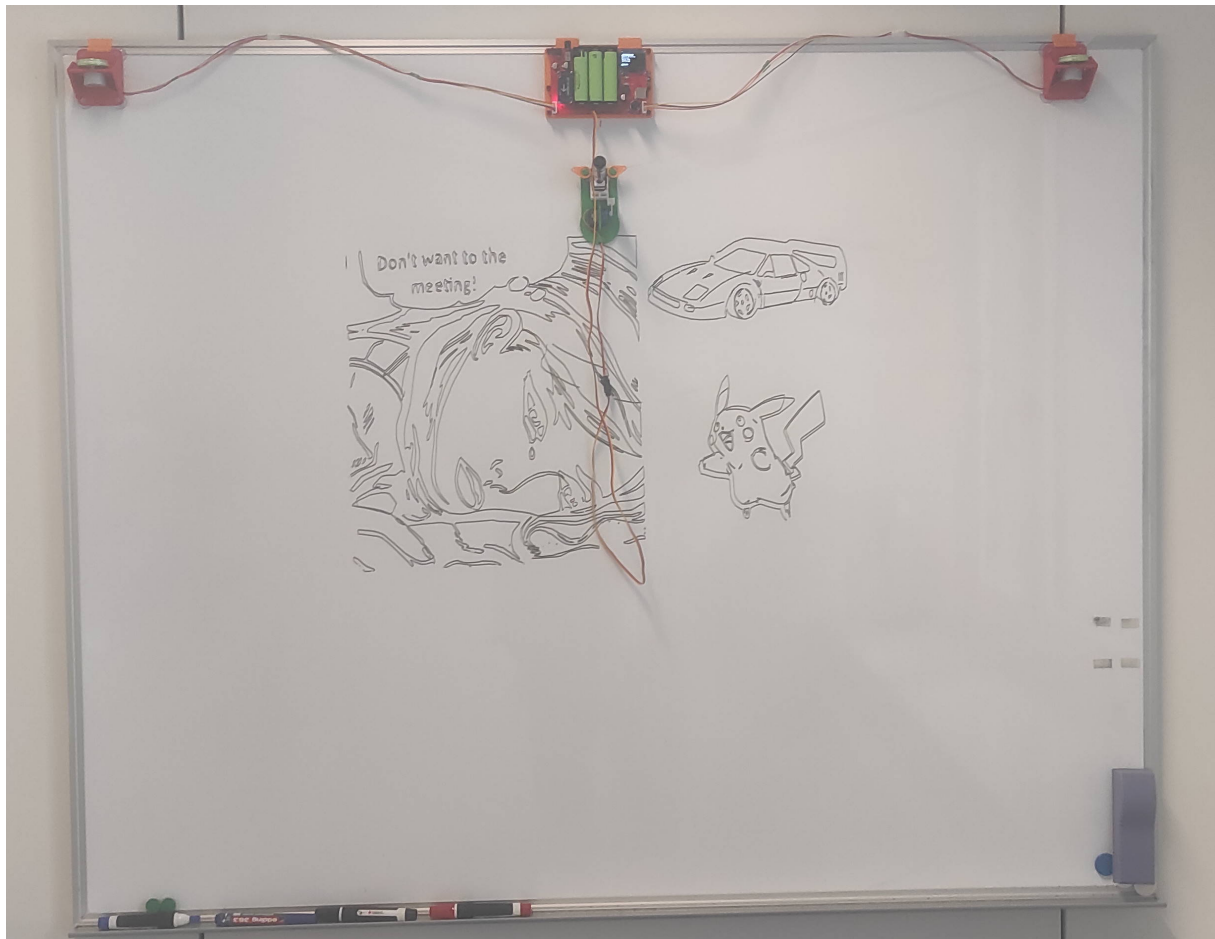


Wandplotter / Polargraph mit UNO oder ESP32 für Whiteboards

Von Thorsten Hartwig, August 2022



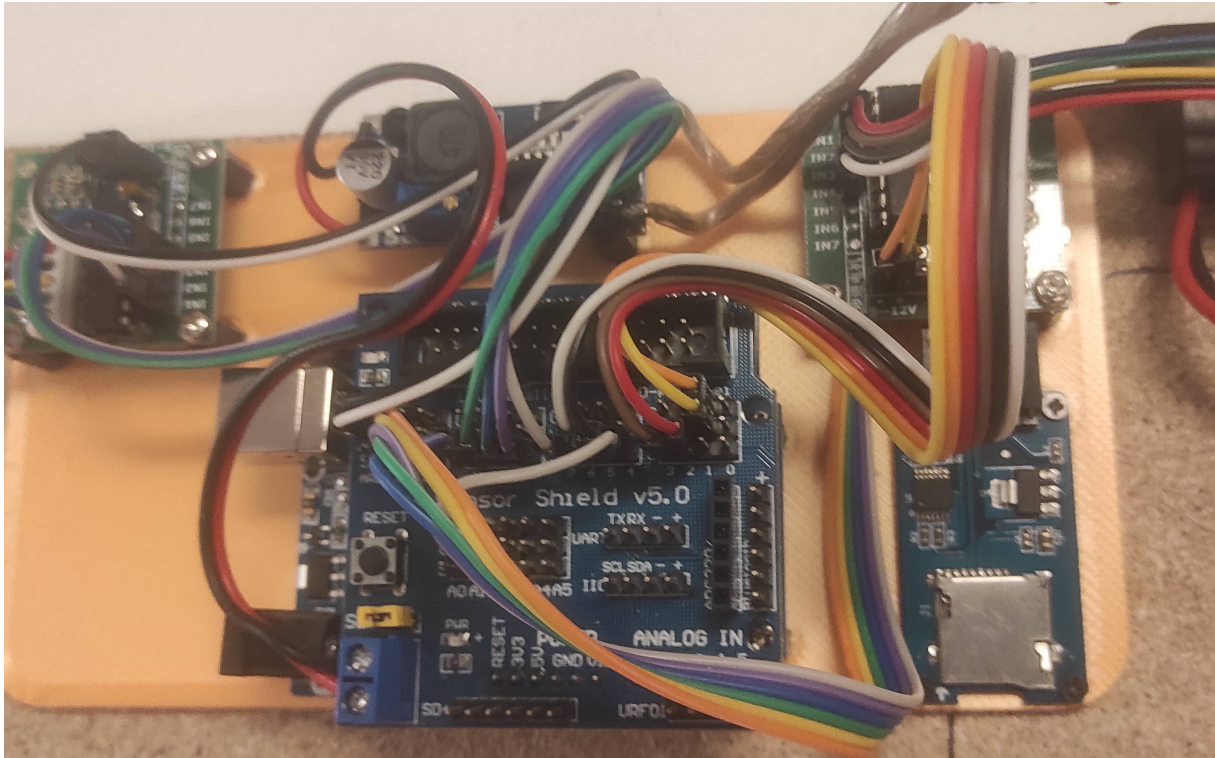
Die Idee eines Wandplotters, der einen Stift an zwei Bindfäden über eine Fläche bewegt, ist nun schon etwas älter. Bekannt sind solche Projekte unter den Namen Makelangelo, Polargraph, Wall Drawer, Wall Plotter, etwas Googeln findet viele Projekte. Schaut die Videos im Projekt an, da steckt nicht so viel hinter.

Einen solchen Plotter gibt es als Holz-Bausatz auf den bekannten asiatischen Shopping-Seiten, sogar auf ebay findet man einen „Arduino Polargraph“ für 69€. Irgendwie taugt das alles nur so halb. Schon das malen an einer senkrechten Fläche geht damit nur schlecht. Eine Konstruktion, die man einfach ans nächste Whiteboard klemmt und laufen lässt, habe ich jedenfalls nirgends gesehen. Das geht alles besser und viel günstiger. Deshalb habe ich mir diesen Bausatz vorgenommen und die Teile für einen 3D-Druck konstruiert.

Das Projekt mit dem UNO fand ich arg primitiv und unhandlich, da habe ich eine Variante mit einem ESP32 entwickelt. Dennoch zeige ich hier auch die UNO-Variante für den Fall, dass jemand keine Elektronikbasterei mag.

Plotter mit Arduino UNO

Wem der UNO schon genügt – bitte sehr: 3D-Dateien ausdrucken, UNO etc. nach Bild aufbauen, verdrahten, Code downloaden und los geht's. Die Stückliste, Fotos und Code gibt's auf GitHub unter <https://github.com/Pontifex42/Walldraw>

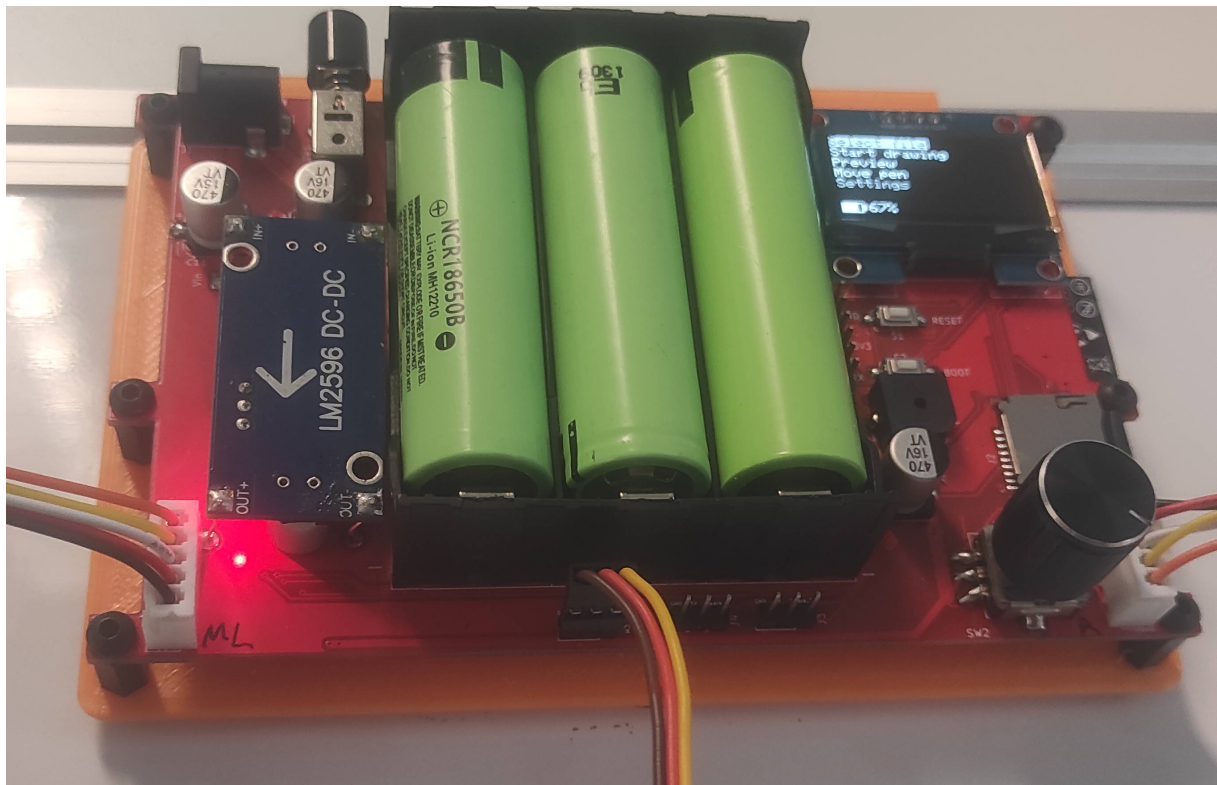


Die Nachteile der UNO-Variante:

- 1) Eine schöne Anleitung oder Schaltplan habe ich dafür nicht. Sollte aber kein großes Problem sein, die Pinbelegung ist im Code zu finden, der Rest sollte sich aus den Fotos erschließen.
- 2) Eine Stromversorgung geht erstmal nur mit einem 5V-Steckernetzteil. Die Zuleitung liegt dann ziemlich im Weg. Und wenn keine Steckdose in der Nähe ist, ist eh kein Kunstpreis zu gewinnen. Eine Versorgung z.B. mit einem LiPo-Akku erfordert nur einen DC-DC-Wandler und ein paar XT-60 Verbindungsstecker, vorzugsweise noch einen XT-60 Schalter. Siehe Fotos.
- 3) Auf der SD-Karte muss das Bild als Datei namens „1.nc“ existieren, die bereits für das passende Papierformat skaliert wurde. Wenn man ein anderes Bild zeichnen möchte oder die Größe ändern möchte trägt man die SD-Karte erst mal zum nächsten PC. Immerhin: um eine Bild-nc-Datei auf ein Papierformat anzupassen habe ich ein DOS-Kommandozeilen-Tool geschrieben „WallDrawResizer.exe“, ebenfalls im Projekt zu finden. Ansonsten nimmt man Inkscape und lernt das erstmal zu bedienen.
- 4) Der Code muss angepasst werden, um z.B. die Auf/-Ab-Winkel des Servos (Wichtig!), die Größe des Plotters (Default: 1000mm), die Drehrichtung der Motoren, die Geschwindigkeit, und weiteres einzurichten. Das Ganze ist eigentlich nur darauf ausgelegt, fest auf ein Brett montiert zu werden. Was man auf den Bildern auch sieht.
- 5) Das Programm fängt nach dem Einschalten an, um die Mitte des Malbereichs herum zu zeichnen. Mehr kann es nicht.

Plotter mit ESP32

Für die, die eine kleine Herausforderung mögen oder einfach gern Elektronik basteln, gibt es eine selbstgemachte Platine und einen „fetten“ Prozessor. Ein ESP32 hat genug Rechenleistung, um eine nc-Datei selbst auf ein gewünschtes Papierformat umzurechnen. Mit einem Display kann man Dateien auswählen, Einstellungen flexibel vornehmen, anstatt im Code zu ändern, die Zeichnung im Malbereich etwas verschieben und manches mehr. Und vor allem: ein paar Akkus machen das Ganze überall einsetzbar. Häuserwände, Büro-Whiteboards, was auch immer.



Wer schon immer mal ausprobieren wollte, eine Platine fertigen zu lassen (sollte für knapp über 10€ bestellbar sein) und SMD-Bauteile zu löten – nur zu! Das ist gar nicht so schwer, wenn man sich vorher ein paar YT-Tutorials ansieht. Die meisten Teile hat man als Elektroniker in den üblichen Sammelsurium-Kits bereits in der Schublade. Allerdings wird es etwas teurer, wenn man noch keine halbwegs eingerichtete Elektronik-Bastelecke hat.

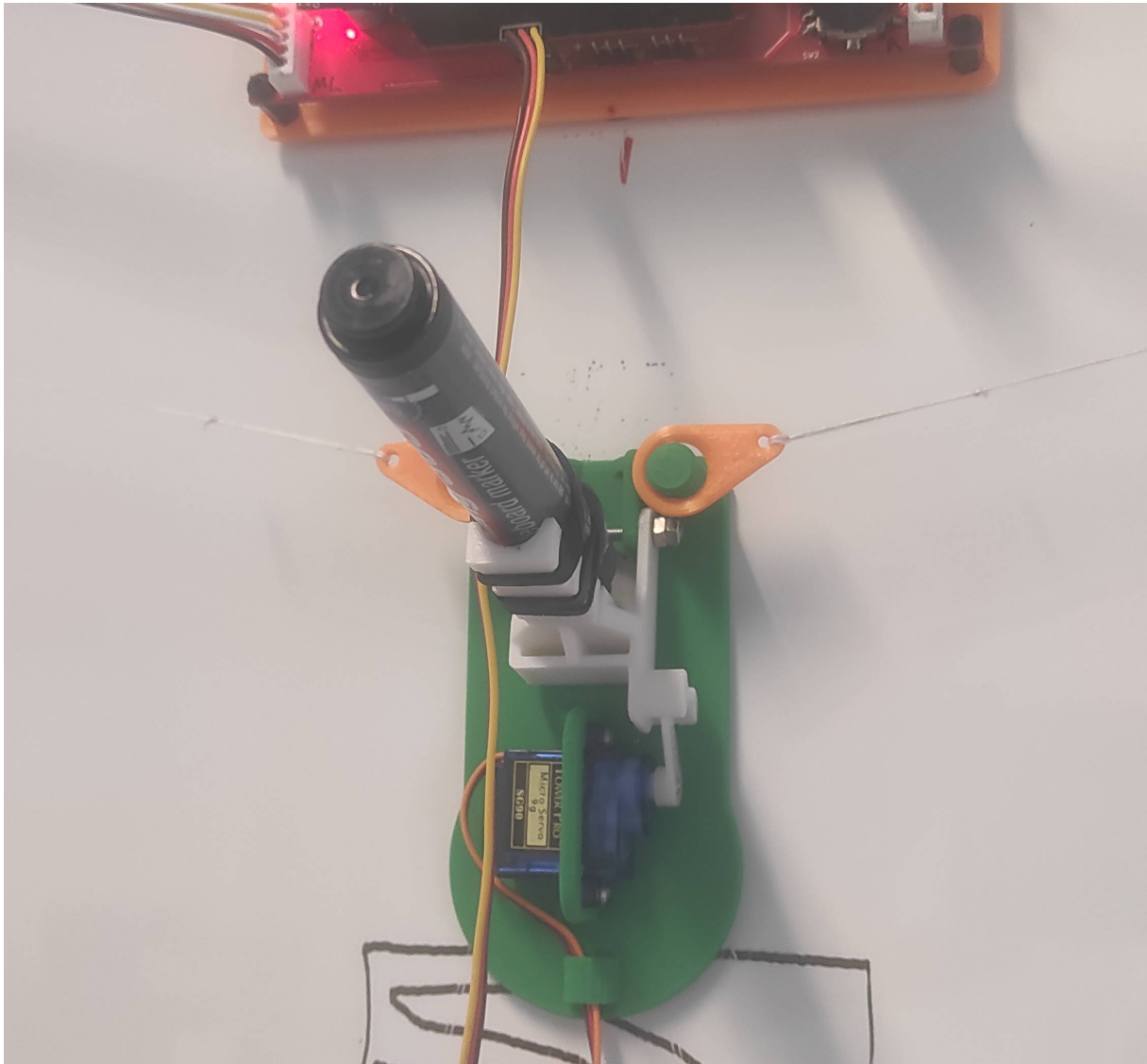
Die 3D-Dateien und Aufbau der Teile

Die Archivdateien für Fusion360 sind bei den Downloads enthalten, nur für den Fall, dass jemand eine Konstruktion anpassen oder verbessern möchte. Ansonsten: ab mit den STL-Dateien in den Slicer und flux den Drucker angeworfen. Alle STL-Dateien sind unkompliziert zu drucken. Stützstruktur kann man bei den Stepper-Haltern einschalten, geht aber auch ohne. Alles andere braucht weder Brim noch Supports.



Die „Hooks“ dienen dazu, die Teile einfach an die Oberkante eines Whiteboards zu hängen. Die lassen sich – absichtlich – etwas schwer einklinken. Alternativ schraubt man das Ganze auf ein Brett. Die Gummisauger lassen sich mit einer Zange und leichter Kraft einsetzen, etwas Übung hilft. Muss man aber nicht haben, etwas Klebeband gegen verrutschen der Motoren tut es auch.

Die Löcher für die Motorachsen sind absichtlich etwas knapp bemessen, damit es stramm sitzt. Je nach Kalibrierung des Druckers muss man ggf. eine Winzigkeit feilen. Für die Schrauben des Servos gilt gleiches. Der Stifthalter wird mit M3 Schrauben oder einer M3 Gewindestange drehbar am Schlitten befestigt. Wenn man keine selbstsichernden Muttern hat sollte man Kontermuttern verwenden.



Die Bindfäden lassen sich zum Aufwickeln in den Kerben der Spulen befestigen. Das andere Ende der Fäden durch eines der kleinen Löcher fädeln, zu einer Schlaufe knoten und entweder an die Fadenringe oder direkt am Schlitten befestigen. Rechter und linker Motor nutzen natürlich das linke und rechte Loch. Wenn man die Stepper-Halter nicht auf ein Brett schraubt und nach hinten einen Luftspalt hat, sollte man die Löcher auf der Unterseite verwenden, das erleichtert später das Balancieren des Stiftes.

Wenn Kabel rumhängen, können die CableClamps helfen, ansonsten braucht man die nicht. Von den Montageplatten druckt man nur die Variante, die man bauen möchte.

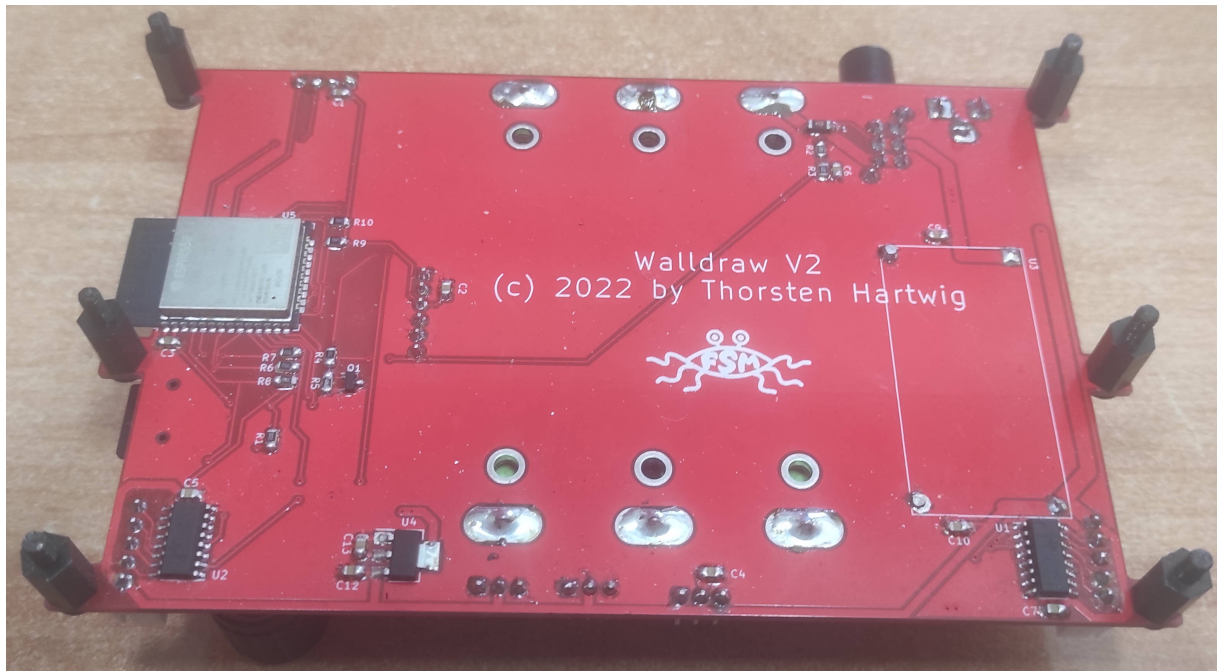
Der Rest sollte sich aus den Bildern erschließen.

Die Platine mit dem ESP32

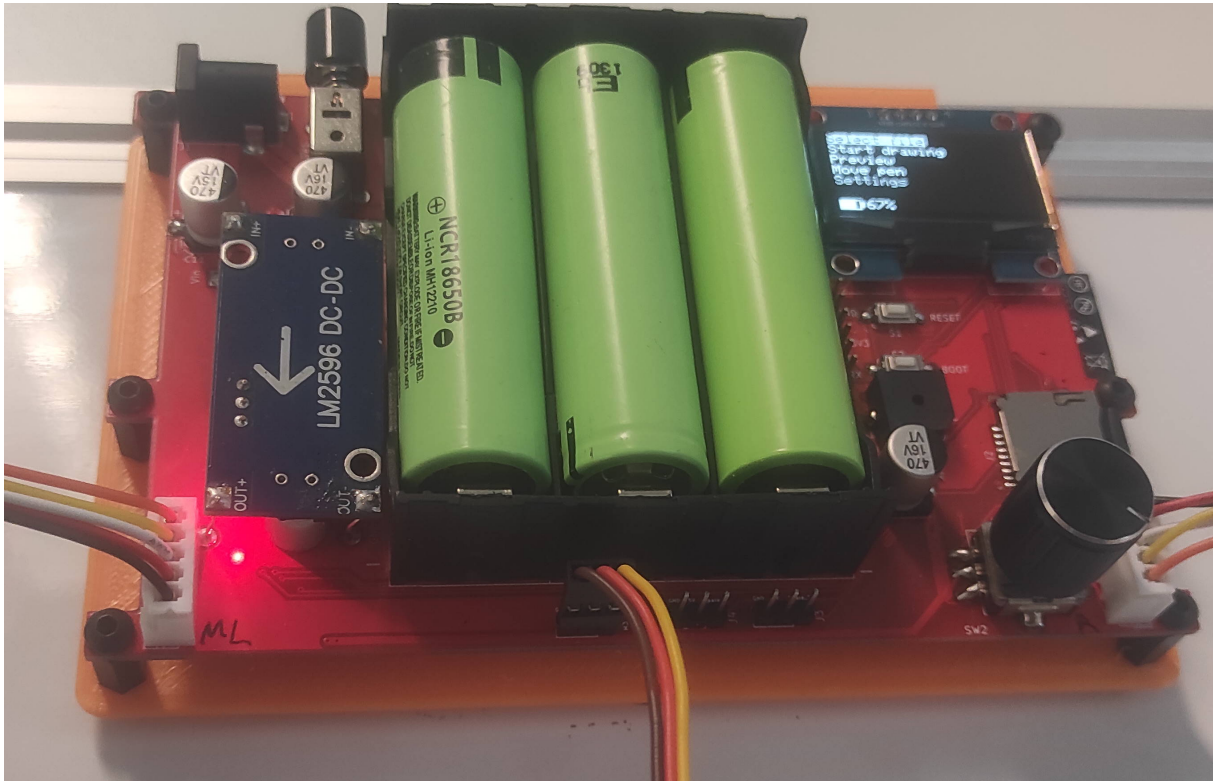
Die Gerber-Dateien zur Fertigung der Platine liegen im Projekt, können aber auch mit KiCad selbst erstellt werden. Die Zip-Datei kann man z.B. bei JLCPCB oder PCBWAY hochladen und den Bestellprozess zügig durchklicken. In etwa 12 Tagen liegen die Teile dann im Briefkasten.

Die Datei ibom.html enthält die Stückliste der Platine mit Links zu möglichen Lieferanten. Ich denke, ich habe da nichts Kompliziertes oder Außergewöhnliches verwendet. Bei Fragen bitte direkt anschreiben, möglicherweise kann auch meine Bastelkiste aushelfen. Sogar überzählige Platinen würde ich innerhalb Europas verschicken. Wer fragt, kann nicht verlieren: walldraw (at) hartwig.org

Zuerst bestückt man die Unterseite. Das Löten des ESP32 Moduls mag eine kleine Hürde sein, ich selbst trage Lötpaste mit einem Stencil auf und lege das auf eine Heizplatte. Ansonsten: nicht sparsam mit Flussmittel sein, dann funktioniert das schon.



Der DC-DC-Wandler wird kopfüber mit Drähten montiert, damit nachher niemand an der Einstellschraube drehen kann und damit das ganze zerstört. Also nicht vergessen: vorher auf 5,0V einregeln.



Manche Bauteile sind nicht unbedingt notwendig. Der Piepser mit Transistor wird eigentlich nur zur Batteriewarnung gebraucht, die Elkos sind eher zur Sicherheit da, es müsste auch ohne gehen. Die „Spare“-Pins sind überflüssig. Der FTDI-Anschluss wird von manchen so gemacht, dass man das FTDI-Modul direkt in die Löcher steckt, und die Taster drückt man vermutlich nur ein einziges mal zum Download der Firmware, da reicht auch ein Stück Draht. Die 5,5mm DC-Buchse braucht man natürlich nicht, wenn man einen Akku als Stromversorgung nimmt. Umgekehrt gilt das für die Akku-Halterung. Und wer braucht schon eine Sicherung? Man könnte auch den SD-Cardreader sparen, wenn man ein Webinterface programmiert und die Bilder per Browser/Smartphone in den Speicher des ESP32 lädt, da sind immerhin 3MB (gibt auch Module mit 8/16MB) frei. Die Mühe habe ich mir nicht gemacht, das ist was für mehr engagierte Bastler.

Die Software

Damit auch Nicht-Programmierer den Code besser nachvollziehen können habe ich auf Objektorientierung verzichtet. Denke, das Ganze ist durch die Ansätze von „Software-Architektur“ halbwegs lesbar. Der Code lässt sich mit der Arduino-IDE 2.x compilieren und laden. Für die ESP-Variante braucht man die OLED-Lib Adafruit_SH110X (mit Adafruit_GFX), für die UNO-Variante die Lib TinyStepper. Ich selbst nutze das Microsoft Visual Studio mit dem Plugin „VisualMicro Arduino“, deshalb ist eine .sln und .vcxproj Datei dabei. Das Plugin kostet nach der 45-Tage-Testzeit für Bastler und Studenten 12\$ im Jahr.

Für den UNO ist damit schon alles gesagt. Für ESP32 gibt es noch zwei Fallstricke: es sind Boardmanager in der Version 1.0.6 und 2.x im Umlauf. Bitte auf die 2.x updaten. Außerdem liefert die Arduino-IDE die lib „SD.h“ mit. Die muss man aus dem Lib-Verzeichnis entfernen, damit die „SD.h“ lib des ESP32 verwendet wird – die heißt dort nämlich genauso. Vorsicht: wenn man den FTDI-Adapter auf 5V statt auf 3,3V eingestellt hat darf man sich eine neue Platine löten. Glückwunsch.

Zum Download, wer es nicht kennt: FTDI Adapter anschließen, dann die beiden Tasten „Boot“ und „Reset“ drücken und zuerst Reset, dann Boot loslassen. Dann ist der ESP32 im Download-Modus.

Die Einrichtung des ESP32

Wenn der Plotter aufgebaut ist, stellt man zuerst die Auf- und Ab-Winkel des Servos im Menü „Settings – Servo angles“ ein. Wenn der Servo mit Kraft gegen Widerstand in die Ab-Position drückt, brennt er nach kurzer Zeit durch. Deshalb in der Ab-Position etwas Spiel lassen.

Als nächstes misst man den Abstand der beiden Fadenlöcher und gibt dies unter „Settings-Plotter dimensions-Set plotter width“ ein. Für die Höhe des Zeichenbereichs in „Settings-Plotter dimensions-Set plotter height“ genügt eine Schätzung. Danach probiert man die Drehrichtung der Stepper unter „Move pen-Move vertical“ aus. Negative Y-Werte müssen nach unten gehen. Wenn es andersrum dreht, entweder die Spulen wenden oder „Settings-Reverse Motors“ wählen.

Alle Einstellungen bleiben nach dem Ausschalten gespeichert.

Da der Stift nicht waagrecht zwischen den Fadenlöchern stehen kann, sondern etwas darunterbleiben muss, ist die Oberkante des Zeichenbereichs festgelegt als ein Achtel der Breite unterhalb der Fadenlöcher, also bei 1500mm Breite etwa 19 cm tiefer. Der Stift muss vor Beginn des Zeichnens durch Drehen der Spulen in diese Höhe gebracht werden, in die Mitte zwischen den Fadenlöchern, das ist die Start- bzw. Null-Position. (Beim UNO ist es die Mitte des Zeichenbereichs.)

Etwas fummelig ist das richtige Einlegen eines Stiftes. Mal ist er zu hoch und zeichnet nie, mal zu tief und zeichnet immer, da probiert man einen Augenblick, bevor man den Schlitten an die Fäden hängt. Die Kerben dienen dazu, den Schlitten auszubalancieren, das hängt vom Gewicht des Stiftes ab.

Die Bedienung

Solange kein weiterer Offset eingegeben wird, ist der obere Zeichnungs- bzw. Papierrand etwa 1/6 der Plotterbreite unter dem oberen Rand (Beim UNO: in der Mitte des Plotterbereichs). Bei einem 1000mm Abstand der Fadenlöcher also 17cm unterhalb der Löcher, horizontal in der Mitte zwischen den Löchern. Dort muss der Stift zu Beginn durch Drehen der Spindeln positioniert werden. Nun kann man per „Select file“ eine Datei von der SD-Karte auswählen, per „Preview“ auf dem Display anzeigen lassen und mit „Start drawing“ loslegen. Eine Datei „Calibrate.nc“, die ein Rechteck zum Testen zeichnet, ist im Projekt.

Mit „Speed“ kann man etwas die Geschwindigkeit ändern. Stufe 10 könnte die Motoren außer Tritt bringen, Stufe 9 und 8 erzeugen etwas unpräzise Striche. Die kleineren Stufen dienen dazu, eine Zeichenoperation zu Demozwecken auf den ganzen Tag auszudehnen.

Eine Akkuladung dürfte für mehr als fünf Stunden Malerei genügen. Vorsicht mit tiefentladenen Batterien: Wenn „Battery Power Low“ angezeigt wird, baldmöglichst ausschalten. Es ist keine Abschaltung des Prozessors implementiert, nur die Motoren werden abgeschaltet. Wenn die Batterien tiefentladen werden, sind sie kaputt. Der DC-Wandler braucht auch im Leerlauf etwas Strom.

Unter „Settings-Paper size“ kann man eine Papiergröße auswählen. Die Zeichnung wird so dimensioniert, dass ein Rand von 1cm verbleibt. Die Einstellung „Native“ zeichnet in der Größe, die von der Datei vorgegeben wird. Mit „Settings-Draw offsets“ kann man die Zeichnung im Zeichenbereich platzieren, um z.B. mehrere Zeichnungen auf ein Blatt zu machen.

Zeichnungen

Was noch fehlt, sind Zeichnungen. Ich hadere etwas damit, meine Zeichnungen zu publishen, da könnte ich in Konflikt mit Urheberrechten kommen. Deshalb eine kurze Anleitung zum Selbermachen von Dateien:

- Hübsche Zeichnung im Internet suchen. Die besten Ergebnisse bekommt man in Skizzen ohne Graustufen oder Farben. Googelt z.B. nach „Einstein Strichzeichnung“ oder einfach „Zeichnung“
- Software „Inkscape“ downloaden und installieren, ist kostenlos.
- Auf YT ein Tutorial anschauen, zu finden z.B. mit der Searchphrase „Inkscape gcode erstellen“
- Die erstellte Datei muss bei meiner Implementierung die Dateiendung „.nc“ haben, also die .ngc-Dateien aus Inkscape umbenennen.
- Für den UNO die Datei mit „WallDrawResizer.exe“ auf gewünschte Größe ändern und umbenennen in „1.nc“
- Ab auf die SD Karte damit und dann etwas mit den Menüs des Plotters beschäftigen.
- Los geht's.

Hab ich was vergessen? Kommentare und Mails sind willkommen. Verbesserungen der Software sowieso.

walldraw (at) hartwig.org