Weichensteuerung / - antrieb 2.0

Mit Funk braucht man nicht so viele Drähte

Um den Fahrweg der Züge zu beeinflussen sind Weichen ein unabdingbares Erfordernis. Will man einen Betrieb mit Rechnersteuerung oder gar einen Automatikbetrieb organisieren, müssen die Weichen von der Steuerungssoftware ansteuerbar sein.

Ein Bild, das Wolke, Zug, Eisenbahn, Maßstabsmodell enthält.

Automatisch generierte BeschreibungWährend über diese Anforderung schnell Konsens erzielbar ist, ist das über die Art, wie die Weichen umgeschaltet werden, nicht ganz so einfach. Wenn man eine kleine Marktrecherche unternimmt, stößt man wohl auf drei verschiedene Ansätze.

Da ist zunächst die Lösung von Märklin. Für ihre Gleise gibt es den Ansatz, ein kleines Metallstück zwischen zwei Magneten flitzen zu lassen, mit dem dann auch die Umschaltung der Weichen realisiert wird. Der Magnet, der das Metall anzieht, bestimmt dann auch die Stellung der Weiche. Der Vorteil dieses Ansatzes ist sicherlich die schlichte und einfache Umsetzung und dass diese Lösung vollständig unter der Weiche verschwindet, also nicht unbedingt eine Öffnung des Bodens unterhalb der Weiche notwendig ist. Nachteile sind sicherlich einmal das deutlich hörbare und für den Bahnbetrieb ziemlich unnatürliche Klacken beim Umschalten sowie die hohe Geschwindigkeit des gesamten Vorgangs.

Diese Nachteile werden beim motorischen Umschalten vermieden.

Da gibt es zunächst die weit verbreitete Servolösung. Dafür wird unterhalb der Weiche ein möglichst kleines Servo in Stellung gebracht. Daran ist ein Draht so angebracht, dass bei Bewegung des Servo die Weiche dann auch umschaltet. Die beschriebenen Nachteile der Märklin-Lösung sind damit auf jeden Fall umgangen. Denn das Servo schnurrt eventuell nur, macht also keine störenden Geräusche und kann den Umschaltvorgang auch beliebig langsam vollziehen. Dieser Ansatz hat aber natürlich auch Nachteile. Da ist zunächst die Öffnung im Boden sowie ein wie auch immer gearteter Halter für das Servo. Vielleicht sind diese Nachteile nicht unbedingt weltbewegend, aber warum nicht eine Lösung wählen, die alle Nachteile, also der Märklin- und der Servolösung nicht aufweist?

# Aufgabenstellung

Also muss eine Lösung her, die leise und in der Geschwindigkeit steuerbar ist sowie unter ein Märklin C-Gleis passt. Da scheidet eine auch verbesserte Servolösung aus, da ich kein Servo gefunden habe, das vollständig unter die Weiche passt. Alle Ansätze haben eine noch nicht erwähnte Gemeinsamkeit. Sie benötigen einen Decoder, der das Signal von der Zentrale bzw. dem Steuerungsprogramm zum Umschalten der Weiche so umsetzt, dass der Weichenantrieb damit zurechtkommt. Denn die Zentrale setzt nur einen kurzen Befehl ab, der dann vom Decoder in längeres Ansteuern umgesetzt wird. Meist sitzt dieser Decoder irgendwo außerhalb der Weiche und ist für die Steuerung mehrerer Weichen zuständig. Das bedingt natürlich wieder die Verlegung diverser Kabel. Ich bin kein Freund dicker Kabelstränge. Deshalb kam die Anforderung hinzu, den Decoder neben dem Weichenantrieb unter der Weiche zu platzieren und als Sahnehäubchen die Verbindung zur Zentrale drahtlos, also über Funk, auszugestalten.

Eine weitere Forderung ist quasi selbstredend, muss aber dennoch erwähnt werden. Die Lösung muss in nicht unbedingt alle, aber die am häufigsten eingesetzten C-Gleisvarianten eingesetzt werden können, das sind die linke und rechte Weiche, die linke und rechte Bogenweiche sowie die Kreuzung.

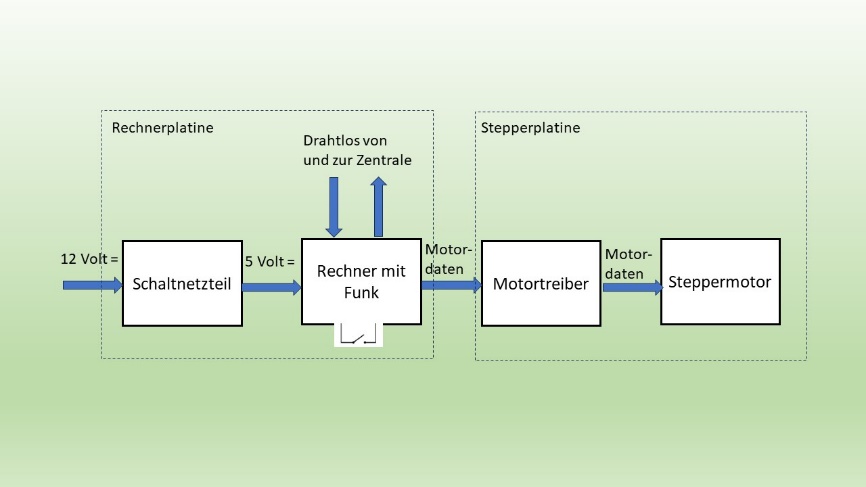
# Lösungsansatz

Wie sieht nun der Ansatz aus, der alle diese Anforderungen umsetzt?

Da ich bei diesem wie auch bei anderen Projekten intensiv meine CNC-Fräse und den 3D-Drucker eingesetzt habe, kam mir relativ schnell der Gedanke, den Servoantrieb analog zu diesen Geräten mit einem Stepperantrieb auszustatten. Denn dort wird der Vortrieb ziemlich exakt mittels eines Steppermotors vollzogen. Weiterhin hatte ich auch in Erinnerung, dass in Fernost Mini-Stepper mit einem 4 mm Durchmesser angeboten werden. Damit war ich der Lösung schon einem Riesenschritt näher geworden. Wie bei allen meinen Entwicklungsansätzen habe ich bewusst daraufgesetzt, möglichst wenig diskrete Bauteile einzusetzen. Stattdessen bevorzuge ich fertige, möglichst kleine Baugruppen. Und das aus mindestens zwei Beweggründen. Einmal spare ich mir damit eine Menge an Entwicklungsarbeit, denn die Module funktionieren in der Regel den Erwartungen entsprechend. Und zweitens suche ich natürlich möglichst kleine Module aus, die dann auch SMD-Bauteile einsetzen, was beim Einsatz von einzelnen SMD-Bauteilen die Lötarbeit zwar spannend aber nicht unbedingt erstrebenswert macht.

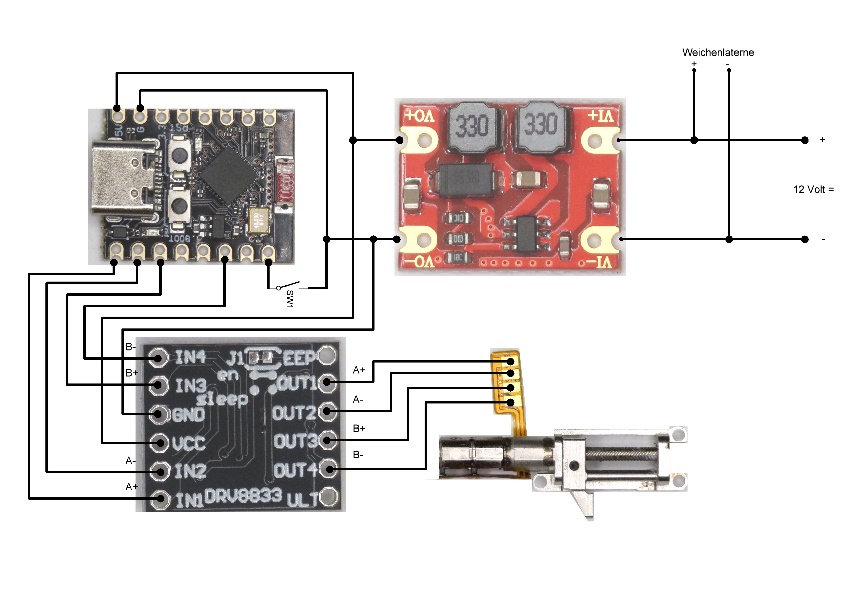
Da alle meine sichtbaren Weichen mit Weichenlaternen ausgestattet werden sollen, gab es die Entscheidung, dass es nur eine gemeinsame Stromversorgung für die Laternen und die Weichensteuerung geben soll. Um die Verlustleistung bei der Anpassung der Eingangsspannung auf die erforderlichen 5 Volt für alle Komponenten zu minimieren, wurde die Spannung auf 12 Volt festgelegt. Damit sind die Laternen ausreichend versorgt und der Energieverlust bei der Reduzierung der Spannung erträglich.

# Die Schaltung

Letzten Endes führte dieser Ansatz zu dem folgenden Blockschaltbild:

Die Eingangsspannung von 12 Volt wird auf ein Schaltnetzteil geführt, das den anderen Komponenten die benötigten 5 Volt zur Verfügung stellt. Die Wahl fiel auf ein Schaltnetzteil und nicht auf einen dreibeinigen Festspannungsregler, weil das gewählte Netzteil einen Wirkungsgrad von über 90 % aufweist und damit die Verlustleistung, die unmittelbar in Wärme umgewandelt wird, minimiert. Und Wärme können wir unter der Weiche überhaupt nicht gebrauchen, denn die kann, zumindest wenn keine Reparaturöffnung vorgesehen ist, nicht abfließen und führt irgendwann unweigerlich zu Problemen. Das Netzteil speist unmittelbar die folgende Rechnerkomponente wie auch den anschließenden Motortreiber, was im Blockschild nicht dargestellt ist.

Der Rechner kann nicht nur rechnen, sondern kann sich auch drahtlos mit der Zentrale unterhalten und empfängt von dort alle Befehle. Er selbst gibt Quittungs- sowie auf Anforderung Verwaltungsinformationen zurück. An diesen Rechner ist ein Taster angeschlossen, mit dem der Steppermotor in seinen Nullpunkt gefahren werden kann. Weiterhin gibt dieses Modul Informationen an den Motortreiber ab, wie dieser den Stepper anzutreiben hat. Dazu werden vier Anschlüsse, die mit A+, A-, B+ und B- bezeichnet sind, benötigt. Der Motortreiber hat keine eigene Logik. Seine Aufgabe besteht darin, die Informationen von dem Rechnermodul an die elektrischen Notwendigkeiten des Steppermotors anzupassen. So gehen auch vom Motortreiber vier Anschlüsse mit identischer Bezeichnung an den Steppermotor.

Nebenstehendes Schaltbild zeigt die konkrete Ausgestaltung mit den Modulen und deren Verdrahtung. Wie man sieht, ist der Schaltungsaufwand durch den Einsatz der Baugruppenmodule auf ein Minimum geschrumpft. Als Netzteil wird ein DM13B-5 eingesetzt. Als Eingangsspannung können bis zu 15 Volt angeschlossen werden, so dass meine Entscheidung für 12 Volt nicht endgültig sein muss. Der Ausgang kann bis 600 mA Strom liefern, die wir bei unserem Aufbau sicherlich in Anspruch nehmen werden. Zudem beeindruckt der kleine Baustein mit einem Wirkungsgrad von bis zu 91%.

Ursprünglich hatte ich eine Eingangsspannung von 9 Volt vorgesehen. Das ist so auch auf der Rechnerplatine aufgedruckt. Die Decoder funktionieren damit auch, aber nur in geringen Stückzahlen. Ganz am Ende des Artikels beschreibe ich einen Testaufbau mit 24 Decodern. Da brach mein 9 Volt Netzteil regelmäßig zusammen. Der Tausch mit einem 12 Volt Netzteil hat dann für Stabilität auch mit 24 Decodern gesorgt.

Der Kern des Systems ist sicherlich der ESP32-C3 Super-Mini, der alles an Bord hat, was wir für unsere Zwecke benötigen. Das ist einmal die WLAN-Schnittstelle für die Kommunikation mit unserer Zentrale und ausreichend viele Schnittstellen, um den Steppermotor anzusteuern und einen Taster für die Nullstellung abzufragen. Die CPU wird mit bis zu 160 MHz getaktet und bietet damit eine ausreichende Leistung, um unsere Anwendung umzusetzen. Ein weiteres wichtiges Auswahlkriterium für dieses SoC (System on Chip) war neben der geringen Größe die externe Antenne, die die Signalstärke verbessert und damit diese drahtlose Anwendung sicher ermöglicht.

Der ESP32-C3 steuert über den Motortreiber den Steppermotor. Dort ist ein DRV8833 mit einer 2 Kanal H-Bridge verbaut. Das Modul leistet bis zu 1.5A, was für unsere Zwecke sicherlich ausreichend ist.

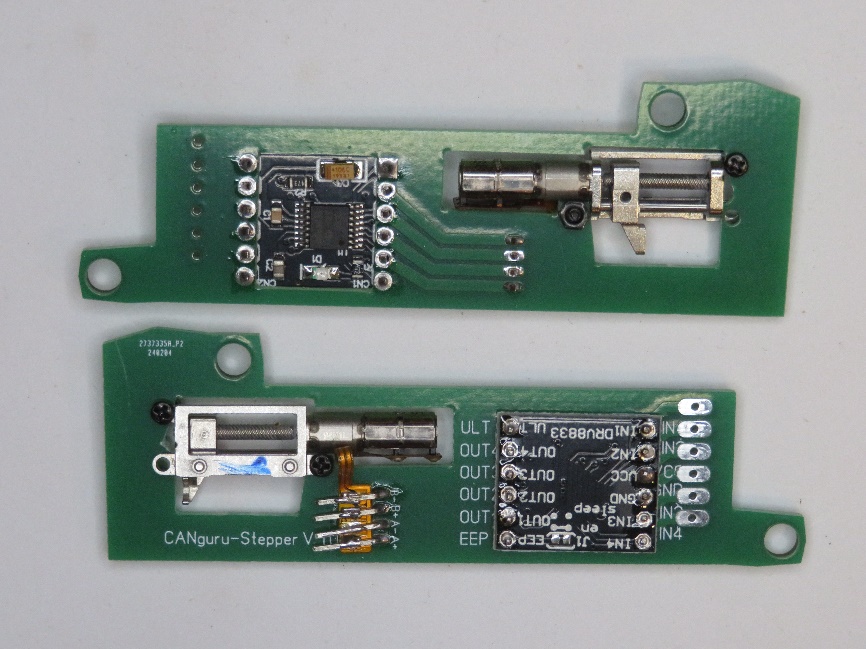
Der Miniatur-Steppermotor dreht über ein Metallplanetengetriebe eine kurze Gewindestange, auf der ein kleiner Auslegerarm sitzt, der ebenfalls ein Gewinde wie eine Mutter aufweist. Wenn sich nun die Gewindestange dreht, bewegt sich der Auslegerarm entsprechend. Durch diese mehrfache Untersetzung verfügt dieses System über ausreichend Kraft und auch einen ausreichend langen Verfahrensweg, um die Weiche sicher zu schalten. Es soll aber ein Nachteil, der mir viel Sorgen bereitet und viele Überlegungen gekostet hat. Es ist dies der elektrische Folienanschluss, für den ich kein brauchbares Gegenstück gefunden habe. Deshalb blieb mir nichts anderes als die hier vorgestellte Lötlösung.

Leider hat der Motor noch einen kleinen Haken. Obwohl man es ihm äußerlich nicht ansieht, gibt es zwei unterschiedliche Typen. Und zwar unterscheiden sie sich in der Laufrichtung. Das ist mir erst aufgefallen, als ich die Weichenantriebe in meiner Kleinserie gefertigt habe. Einige Exemplare verhielten sich einfach umgekehrt wie erwartet. Diesem Phänomen kann man relativ einfach begegnen; man muss es nur kennen.

# Die Platinen

Ein Bild, das Elektronik, Elektronisches Bauteil, Elektrisches Bauelement, passives Bauelement enthält.

Automatisch generierte BeschreibungPrägend für die Gestaltung der Platinen waren zwei Aspekte. Einmal musste das Layout so flexibel ausgelegt werden, dass die Platinen in allen vorgesehenen Weichentypen angewendet werden können. Das führte zu der Entscheidung, dass man diese Anforderung nicht mit einer Platine erfüllen kann. Dafür ist die Aufteilung in zwei Platinen notwendig. Da Märklin die Erstellung eines optimalen Layouts bereits gelöst hat, habe ich mich bei der Gestaltung der Umrisse intensiv daran orientiert.

Die zweite Anforderung war die Beschränkung der Lösung in der Bauhöhe. Das bedeutet, dass an keiner Stelle ein Bauteil über den Rand der Weiche hinausragen darf. Da aber die Summe von Platinenstärke und Modulhöhe die Messlatte der zulässigen Höhe reißt, musste eine andere Lösung her. Die bestand darin, dass die Aufbauten der einzulötenden Module, wie beispielsweise ESP32-C3, in der Platine verschwinden müssen. Dazu wurden Ausschnitte in der Platine vorgesehen, in die dann die Bauteile der Komponenten verschwinden. Entsprechend müssen diese Module so eingelötet werden. Das führt dann dazu, dass die Bauteile auf den Modulen auf die Höhe insgesamt nicht mehr angerechnet werden. Mehr dazu bei der Betrachtung der beiden Platinen.

Am Beispiel der Rechnerplatine kann man erkennen, was damit gemeint ist. Dreht man die bestückte Platine um 90°, wird klar, dass alle Bauteile der eingelöteten Module in den Aussparungen verschwinden. Sicherlich ist eine besondere Verfahrensweise notwendig, um die Module in dieser Art einzulöten. Dazu später mehr.

## Die Stepperplatine

Ein Bild, das Text, Elektronisches Bauteil, Elektronik, Schaltung enthält.

Automatisch generierte BeschreibungDie Stepperplatine trägt den Motortreiber sowie den Steppermotor.

Nun ist das Bestücken der Platine bzw. das Einlöten der Komponenten etwas gewöhnungsbedürftig. Denn die Verbindung zwischen Lötpad auf der Platine und Lötpad auf dem Modul wird wie gewohnt über Pfostenstifte geleistet. Allerdings dürfen die Stifte an beiden Seiten lediglich maximal 1 mm aus der Lötstelle herausragen, damit die Maximalbauhöhe eingehalten wird. Diese Randbedingung erfordert besondere Maßnahmen.

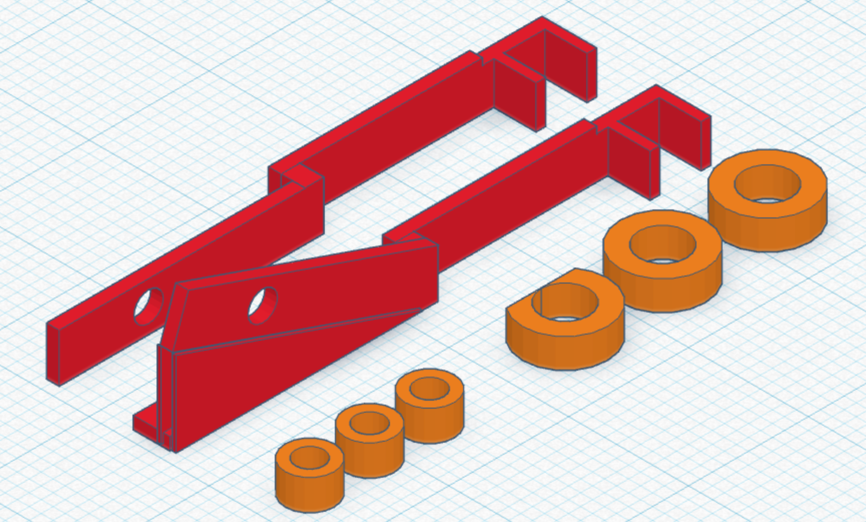
## Die Rechnerplatine

Die Rechnerplatine trägt den ESP32-C3 sowie die Schaltnetzteilplatine.

Ein Bild, das Elektronisches Bauteil, Elektronik, Elektrisches Bauelement, passives Bauelement enthält.

Automatisch generierte BeschreibungDer Ausschnitt für den ESP32-C3 wurde nach vorne offen gestaltet. Damit ist es möglich, den USB-Anschluss für die Erstprogrammierung zu erreichen, auch wenn das Modul mehr oder weniger in der Trägerplatine versinkt. Am oberen Rand erkennt man die beiden Anschlüsse für den Taster, mit dem der Steppermotor auf den Nullpunkt gefahren wird. Am rechten Ende sind zwei 9 Volt Anschlüsse. Einer der beiden dient der Spannungsversorgung, an den anderen kann die Weichenlaterne angeschlossen werden.

# Die 3D-Teile

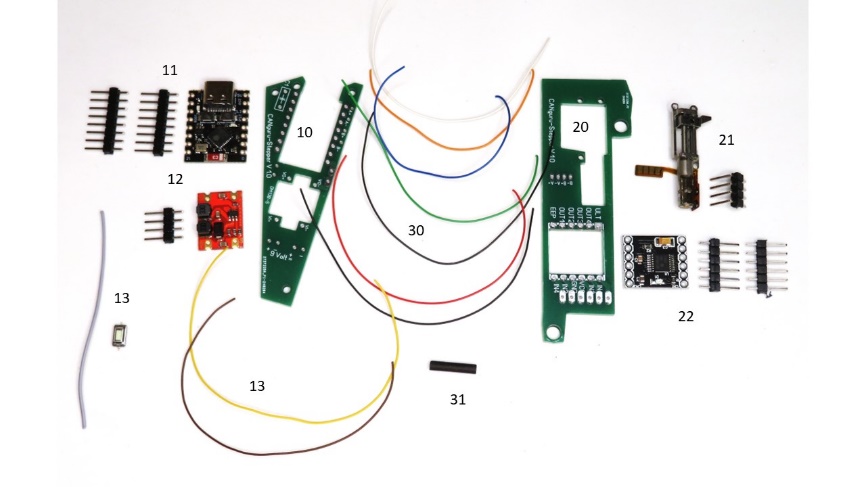
Für den Einbau und die Bewegung der Weiche werden noch einige wenige 3D Teile benötigt. Da ist als wichtigstes Element die Verbindungsschiene zu nennen, die die Verbindung zwischen dem Stepperarm und dem kleinen Bolzen an der Weichenzunge herstellt. Die Verbindungsschiene gibt es in zwei Ausfertigungen, einmal gerade und einmal gebogen. Je nach Weichenart wird die gerade oder die gebogene eingesetzt. Es ist ganz einfach: alle Weichen kommen mit der geraden Schiene zurecht. Lediglich die Kreuzung verlangt nach der gebogene Verbindungsschiene.

Dann gibt es noch verschiedene kleine Abstandshalter. Die kleinen werden auf die drei dünnen Bolzen gesteckt, auf die die Rechnerplatine aufgesetzt werden. Die größeren kommen auf die Halter für die Stepperplatine.

Diese Teile wurden mit dem Werkzeug TinkerCAD, das hier bereits vorgestellt wurde, entworfen und stehen im Unterverzeichnis […/0125-3D-Parts](https://github.com/CANguru-System/20-CANguru-3.0/tree/main/0125-3D-Parts) (siehe Kapitel über die Installation) als stl-Dateien zum Download bereit.

# Stückliste

Um den beschriebenen Platinen Leben einzuhauchen, werden natürlich noch einige Bauteile benötigt. Im folgenden Bild sind alle Zutaten dargestellt und auch so gruppiert wie sie funktional zusammengehören.

Die Teile sind in der anschließenden Stückliste einzeln mit Bezugshinweisen aufgeführt. Die Nummern in der rechten Spalte beziehen sich auf die Nummern im Bild.

Die vielen Einzeldrähte mitten im Bild sind die Verbindungen zwischen den beiden Platinen. Dazu habe ich unterschiedliche Farben benutzt, damit keine Anschlüsse vertauscht werden.

Dabei ist noch zu beachten, dass die beiden Platinen in den fünf Weichenvarianten auch unterschiedlich weit entfernt eingebaut werden. Um die Platinenkombination flexibel in alle Varianten eingebaut werden kann, hat sich bei mir folgende Festlegung bewährt:

Verbindung zwischen

A+: weißer Draht, 8 cm

A-: oranger Draht, 8 cm

B+: blauer Draht, 8 cm

B-: grüner Draht, 8 cm

Spannungsversorgung der Motorplatine

Pluspol: roter Draht, 10 cm

Minuspol: schwarzer Draht, 10 cm

Diese insgesamt sechs Drähte werden mit einem kurzen Stück Schrumpfschlauch zusammengebündelt, um das Kabelgewirr unter der Weiche zu minimieren.

Anschluss Spannungsversorgung 12 Volt an die Rechnerplatine

Pluspol: gelber Draht, 12 cm

Minuspol: brauner Draht, 10 cm

Anschluss Taster für die Motoreinstellung

Grauer Draht, 6 cm

|  |  |  |
| --- | --- | --- |
| Bauteil | Zu beziehen bei |  |
| Platinen | https://github.com/CANguru-System/20-CANguru-3.0/tree/main/0120-Platinen/Weiche-Stepper-ESP32C3 | 10 und 20 |
| Diverse 3D-Teile | https://github.com/CANguru-System/20-CANguru-3.0/tree/main/0125-3D-Parts |  |
| DM13B-5 (Schaltnetzteil) | AliExpress | 12 |
| ESP32-C3 Super-Mini | AliExpress | 11 |
| DRV8833 (Motortreiber) | Amazon | 22 |
| Miniatur-Steppermotor | AliExpress | 21 |
| Stiftleisten, gerade und gewinkelt | Diverse Quellen, auch ebay.de |  |
| Linsenschrauben M1.6 4 mm | Ebay.de |  |
| Sechskantmuttern M1.6 | Ebay.de |  |
| Linsenschrauben, Tellerkopf M2 5 mm | Ebay.de |  |
| Mikrotaster SMD 3x6x2,5mm | Ebay.de | 13 |
| Kupferschaltdraht 0,5 mm verschiedene Farben | Schönwitz | 30 und 13 |
| Schrumpfschlach | Ebay.de | 31 |

Leider werden viele Bauteile ausschließlich in Fernost angeboten. Hierzulande besteht da Nachholbedarf, was die Herstellung dieser Miniaturteile wie auch deren Lieferung anbelangt. Preislich liegt man bei etwa 15 € pro Weichensteuerung, wenn man die günstigen Quellen nutzt.

# Bestücken der Platinen

Zunächst wird das Vorgehen beim Einbau der spezifischen Anteile der Stepperplatine erläutert. Anschließend wird erklärt, wie die Chips eingebaut werden, was für beide Platinen gleichermaßen gilt.

o Dieser Schritt ist bei der Stepperplatine nicht immer notwendig, aber immer dann, wenn der Motor sehr eng in dem Platinenausschnitt liegt. Wegen leichter Toleranzen bei den Steppermotoren wurde der Ausschnitt für die Motoren etwas zu gering festgelegt. Deshalb muss mit einer kleinen Feile eine etwas 1 mm lange Ritze zwischen den beiden Bohrlöchern angebracht werden. Dann kann der Motor ohne Druck in das vorgesehene Loch eingebracht werden.

o Da die Bohrlöcher ziemlich eng am Motorkörper anliegen, kann es Schwierigkeiten beim späteren Festschrauben des Motors geben, und zwar an der auf dem Platinenbild oberen Verschraubung. Das äußert sich so, dass sich zwar die Schraube gut von oben einführen lässt, die Mutter nicht eingedreht werden kann, weil der Motorrahmen zu nah an der Schraube sitzt. Auch hier hilft nur etwas beherztes Schleifen an der beschrieben Problemstelle.

o Wenn alle die Schwierigkeiten gemeistert sind, kann man den Motor einschrauben. Wenn auch 3 Löcher vorgesehen sind, reichen erfahrungsgemäß auch zwei wie auf dem nächsten Bild zu sehen. Stecken Sie zunächst eine Schraube (M1.6) in ein Loch und legen die zugehörige Mutter auf den Tisch. Nun positionieren Sie die Platine mit Motor und Schraube so, dass die Schraube genau über der Mutter steht (man kann zwischen Platine und Motorrahmen etwas hindurchsehen) und drehen dann die Schraube. Mit etwas Glück hat sich dann die Mutter etwas festgedreht. Den Rest erledigen Sie nun von der Rückseite. Das gleiche Prozedere führen Sie mit der zweiten Schraube/Mutter-Kombination aus.

o Nachdem der Motor festsitzt, muss der elektrische Anschluss des Motors fixiert werden. Dazu drücken Sie das bewegliche Pad so auf die Platine, dass die vier nebeneinanderliegenden Anschlüsse recht dicht an die entsprechenden Bohrlöcher, die mit A+ bis B- bezeichnet sind, zu liegen kommen. Falls die Rückseite des beweglichen Pads nicht mehr klebefähig ist, verwenden Sie eine winzige Menge Sekundenkleber. Achten Sie dabei darauf, dass Sie die danebenliegenden Löcher nicht mit Kleber verschließen.

o Wenn der Anschluss verklebt ist, bringen Sie einen kleinen Tropfen Lötzinn auf die 4 Pads des Anschlusses auf.

Ein Bild, das Elektronik, Elektrisches Bauelement, Elektronisches Bauteil, passives Bauelement enthält.

Automatisch generierte Beschreibungo Im nächsten Schritt wird die elektrische Verbindung zwischen dem Motor und der Platine hergestellt. Dazu werden zunächst vier gewinkelte Stiftleisten vereinzelt und von den schwarzen Haltern befreit. Dazu werden die Stiftleisten in die Bohrlöcher neben den Pads gesteckt und dann mit den Pads und den Bohrlöchern verlötet.

* Die Stiftleisten werden auf der Rückseite der Platine mit einem kleinen Seitenschneider abgeknipst

Nun werden die Chips eingelötet. Da sie etwas anders als normalerweise einzubauen sind, werden die einzelnen Schritte im Folgenden aufgeführt. Die Bilder zeigen den Einbau des ESP 32 in die Rechnerplatine. Das Einlöten der beiden anderen Bauteile erfolgt ganz analog.

* Ein Bild, das Elektrisches Bauelement, Elektronisches Bauteil, Elektronik, passives Bauelement enthält.

  Automatisch generierte BeschreibungVorbereiten der geraden Stiftleisten, indem die schwarzen Halter ganz nach unten gezogen werden.
* Einführen der Stiftleisten von der unbedruckten Seite her.
* Umdrehen der Platine und hinlegen.
* Die schwarzen Halter liegen auf einer 1 bis 2 mm starken Unterlage (hier 2 mm Polystyrol).
* Das Ende der Platine liegt und dem 8 mm Holzstück und wird festgeklemmt, so dass die Stiftleisten etwa 1 mm aus der Platine herumschauen.
* Die zweimal 8 Lötstellen werden verlötet.
* Ein Bild, das Elektronik, Elektronisches Bauteil, Elektrisches Bauelement, passives Bauelement enthält.

  Automatisch generierte BeschreibungDie Platine wird umgedreht und von der Rückseite verlötet.
* Ein Bild, das Elektronisches Bauteil, Elektronik, Elektrisches Bauelement, passives Bauelement enthält.

  Automatisch generierte BeschreibungDie Stiftleisten werden mit einem kleinen Seitenschneider abgeknipst.

# Die Installation

Neben dem mechanischen Zusammenbau habe ich mir auch Gedanken über den Prozess, wie die Software auf den Weichenantrieb kommt, Gedanken gemacht. Denn meist hat man nicht nur einige wenige sondern etliche solcher Systeme auf der Anlage, in meinem Falle sind es 30 Stück.. Da macht es schon Sinn, etwas Mühe in ein Installationsprogramm zu stecken.

## Das Installationsprogramm

Die Software, die den Decodern das Leben einhaucht, wird hier als Firmware bezeichnet. Um diese Firmware auf die Decoder, also den ESP32, zu bringen, gibt es mehrere Möglichkeiten. Die naheliegende ist die Nutzung der Programmierumgeben Visual Studio Code (VSC). Wie das geht, wurde bereits an anderer Stelle erläutert. Deshalb wird hier darauf verzichtet. Für diejenigen, die VSC nicht nutzen, wurde ein Installationsprogramm für die Erstinstallation entwickelt. Für alle Decoder ist es so, dass neben dem Aufbringen der Firmware bei der Erstinstallation weitere Informationen auf die Hardware geladen werden müssen. Für diese Erstinstallation wird der ESP32 über den USB-Anschluss mit dem Windowsrechner verbunden. Später werden Softwareupdates über die Luftschnittstelle, also drahtlos, über das heimische WLAN per „Over The Air“ (OTA) vollzogen. Das ist nicht zu verwechseln mit der Drahtloskommunikation ESP-NOW, die für die Datenkommunikation während des Betriebes des ESP32 benutzt wird. Für die OTA-Kommunikation muss der Rechner die IP-Adresse des zu bedampfenden Decoders kennen sowie das Passwort zur Nutzung des WLAN. Beide Informationen werden bei dieser Erstinstallation neben der Firmware auf den ESP32 zur späteren Verwendung geladen.

Um von dem Installationsprogramm Gebrauch machen zu können, müssen neben dem Programm selbst auch die zu installierenden Softwareanteile in einer festgelegten Struktur bereitgestellt werden. Das ist gar nicht so kompliziert wie es sich im ersten Moment anhört. Denn, wenn Sie die Software von der CANguru-System Github-Seite herunterladen und auf Ihrem Rechner entfalten, ist die benötigte Struktur bereits vorhanden.

Ein Bild, das Text, Muster, Screenshot, Quadrat enthält.

Automatisch generierte BeschreibungWenn Sie nebenstehenden QR-Code scannen oder über einen Klick auf den Link https://github.com/CANguru-System/20-CANguru-3.0/tree/main gelangen sie auf die Github-Seite mit allen dafür notwendigen Dateien. Um diese Dateien auf Ihren Rechner herunterzuladen, klicken Sie auf das grüne Feld „Code“ und dann auf „Download ZIP“. Die heruntergeladene ZIP-Datei können Sie an einem Ort Ihrer Wahl abspeichern und dann dort extrahieren. Im nun sichtbaren Verzeichnis 0105-CANguru-Install-All finden Sie die Datei CANguru-Install-All.cmd, die Sie durch einen Doppelklick ausführen. Daraufhin startet das Installationsprogramm.

Nebenstehender Screenshot zeigt dieses Installationsprogramm, mit dem übrigens auch der obige QR-Code erstellt wurde. Wenn Sie einen Link (url) in das Textfeld unten links eingeben und dann auf den Knopf QR-Code klicken, wird die zugehörige Bitmap erstellt und angezeigt.

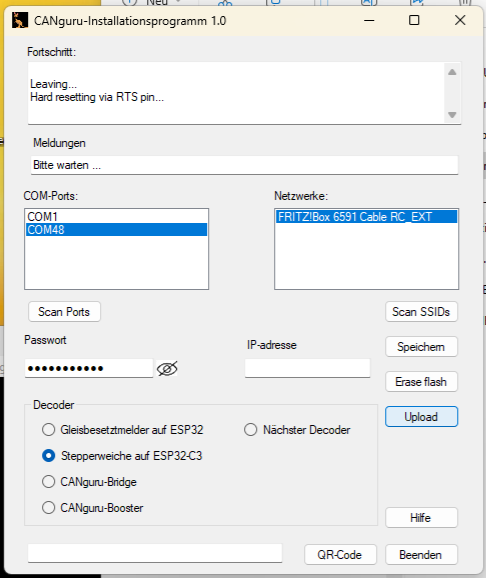
Wie der Installationsprozess abläuft, wird in den nachfolgenden Schritten erläutert.

o Da bei der Erstinstallation alle Informationen über den USB-Port übertragen werden, muss dem Installationsprogramm die Nummer des benutzten Ports bekannt sein. Drücken Sie den Button SCAN, um die belegten Ports zu sehen und wählen Sie den relevanten durch Anklicken aus. Auch für den Fall, dass Sie einen Decoder an Ihren Rechner angeschlossen haben, kann es sein, dass das Programm keinen benutzten Port findet. Das kann daran liegen, dass Sie den notwendigen Treiber der Firma Silicon Labs nicht installiert haben. Im gleichen Verzeichnis finden Sie diesen Treiber, den Sie dann anwenden müssen.

o Da wie bereits an anderer Stelle ausgeführt, für ein späteres OTA-Verfahren die IP-Adresse dieses Decoders hinterlegt werden muss, ist es notwendig, dem Decoder kurzzeitig Zugang zum heimischen WLAN-Netzwerk zu verschaffen, damit der Router die IP-Adresse vergeben kann. Dafür sind dessen SSID (steht für Service Set Identifier und ist der Name eines WLAN-Netzwerkes) und das Passwort notwendig. Wenn Sie nun den Knopf Scan SSIDs drücken, werden nach einer „Denkpause“ alle verfügbaren SSIDs angezeigt. Wählen Sie dann den zutreffenden durch Anklicken aus.

o Anschließend tragen Sie das zugehörige Passwort in das gleichnamige Feld ein.

o Nun wählen Sie die Firmware aus der Liste aus, mit der Sie den Decoder bespielen möchten.

o Mit dem Button *Erase flash* können Sie einen Decoder vollständig löschen. Insbesondere werden alle Einstellungen des angeschlossenen Decoders gelöscht. Das ist immer dann notwendig, wenn im Rahmen einer Softwareänderung Parameter hinzugefügt oder gelöscht wurden. Bei normalen Softwareänderungen ist das nicht notwendig.

o Sie können nun den Button *Upload* drücken, um die Firmware des Decoders aufzuspielen. Dazu wird zunächst eine Software aufgespielt, die alle für das spätere OTA-Verfahren notwendigen Parameter erkundet und auf den Chip speichert, so auch die IP-Adresse. Anschließend wird die Firmware für die Weichensteuerung aufgespielt. Damit ist die Software-Installation fertiggestellt und das Einmessen kann beginnen. Es kommt zwar selten vor, aber es sollte erwähnt werden, dass ein ESP-Chip Probleme mit dem WLAN haben kann. Dann bekommen Sie eine Meldung, dass keine PI-Adresse vergeben werden konnte. Manchmal hilft es, das ganze Prozedere zu wiederholen. Häufig aber leider auch nicht. Um diesen Fall mehr oder weniger auszuschließen, kann man den Aufspielprozess als allerersten, also vor dem Einlöten durchführen. Wenn man dann ein schlechtes ESP32-Exemplar erwischt, dann hat man nur 2€ verloren, aber nicht 1 ganze Stunde Hobbyzeit für ein unnötiges Einlöten.

Falls dieser Vorgang in einem anderen Fehler endet, so liegen meist folgende Fälle vor. Möglicherweise beklagt sich das Programm, dass es den COM-Port nicht finden kann oder er nicht existiert. Eventuell ist der Port durch ein anderes Programm, z.B. ein Terminalprogramm belegt. Abhilfe schafft sofort ein Schließen dieses Programmes. Im anderen Fall ist der ESP32-C3 nicht im Download-Modus. Das wird behoben, indem der Boot-Knopf heruntergehalten wird und währenddessen kurz der Reset-Knopf gedrückt wird.

o Mit dem Button Speichern können Sie den Port, den SSID und das Passwort sichern. Und brauchen diese Daten bei einem Neustart des Programmes nicht erneut ermitteln bzw. eingeben.

## OTA (**O**ver **T**he **A**ir)

Im vorhergehenden Abschnitt haben wir die Firmware über ein USB-Kabel vom PC auf den ESP geladen. Das ist beim ersten Mal auch unumgänglich. Wenn die Weichensteuerung aber erst einmal auf der Anlage verbaut ist, wollen wir sicherlich nicht mehr mit dem USB-Kabel unter der Anlage hantieren, wenn mal die Software ein Update erfahren soll. Da geht es mit OTA drahtlos sicherlich einfacher.

Dafür genügt es, den ESP32-C3 in einen Ladezustand zu bringen und anschließend die ElegantOTA-Seite aufzurufen. In den Ladezustand gelangen wir, indem wir beim CANguru-Server auf dem Reiter Konfiguration (siehe Abschnitt *Anpassen der Weiche*) den betroffenen Decoder auswählen und den Knopf OTA betätigen. Der ESP verbindet sich nun mit dem heimischen Netzwerk. Nun öffnen wir auf dem PC einen Web-Browser, z.B. Google Chrome oder Microsoft Edge und geben in die Adresszeile Ein Bild, das Text, Software, Computer, Computersymbol enthält.

Automatisch generierte Beschreibungdie während des ersten Installationsvorganges oder auch im CANguru-Server angezeigte IP-Adresse gefolgt von einem */update*. Es meldet sich dann der ESP mit dem folgenden ElegantOTA-Bildschirm.

Sie werden nun aufgefordert, zu der ausführbaren Datei zu navigieren, die auf den ESP geladen werden soll. Dabei handelt es sich stets um den gleichen Pfad und den gleichen Dateinamen. Wenn Sie die Dateien von der github-Seite heruntergeladen haben, sieht der Weg so aus ../0104-Weiche-Stepper-ESP32C3/.pio/build/seeed\_xiao\_esp32c3. Dort finden wir u.a. die Datei *firmware.bin*. Diese Datei wählen wir aus und bestätigen. Unmittelbar danach beginnt der Ladeprozess. Der Fortschritt kann mit den Prozentzahlen und Balken beobachtet werden.

Nach einigen Sekunden ist der Prozess beendet und der ESP32-C3 wieder einsatzbereit.

# Das Einmessen

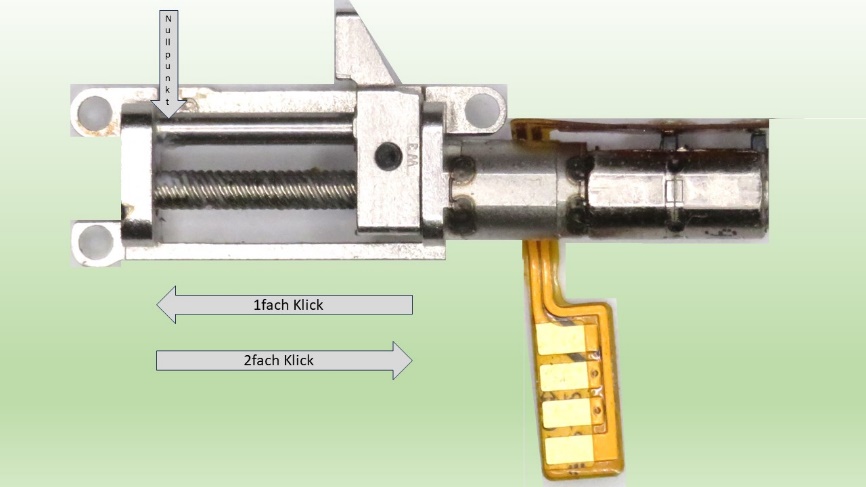
Bevor der Weichenantrieb in Betrieb genommen werden kann, muss der Fahrweg des Motors eingemessen werden. Dadurch wird insbesondere der Nullpunkt festgelegt. Das ist der Punkt, von wo aus das Servo in die linke Stellung läuft bzw. anschließend in die rechte zurückläuft. Dieser Punkt liegt am motorentfernten Ende des Fahrweges. Neben der Festlegung auf diesen Startpunkt kann es sein, dass die Länge des Fahrweges angepasst werden muss.

Die erste Aufgabe besteht also darin, den kleinen Stepperarm an den oben beschriebenen Nullpunkt zu fahren. Dies wird mit dem kleinen Taster, der an den ESP32 angeschlossen ist, vorgenommen. Softwareseitig gibt es vier Signalmöglichkeiten, um den Stepper in die gewünschte Position zu fahren, nämlich das einfache, das doppelte und das dreifache sowie auch das vierfache kurze Drücken des Tasters sowie das lange Drücken (mindestens eine Sekunde).

Im ersten Schritt wird festgelegt, in welche Richtung der Motor laufen soll. Mit dem einfachen Drücken ist dies die Richtung zum gewünschten Nullpunkt, bei einem Doppelklick die entgegengesetzte Richtung, also zum Motor zu. Erst beim nächsten Druck auf den Taster läuft der Motor bzw. der Stepperarm dann wirklich, wie gewünscht los. Der Motor läuft während des Einstellvorgangs bewusst besonders langsam, um den Nullpunkt exakt feststellen zu können.

Man könnte meinen, dass ein Lauf in Richtung des Motors gar nicht gebraucht wird. Das ist prinzipiell auch richtig. Ich habe mir aber angewöhnt, zunächst diese Richtung zu wählen und damit zu sehen, ob der Motor korrekt arbeitet. Und das insbesondere dann, wenn der Stepperarm bereits am Nullpunkt steht.

Dann gibt es noch zwei Signalmöglichkeiten mit dem Taster. Das ist einmal der Dreifachklick. Der wird immer dann gebraucht, wenn man sich bei der Richtungswahl, also dem ersten Drücken vertan hat. Mit diesem schnellen dreifachen Drücken kommt man quasi wieder in die Ausgangsstellung. Der nächste Tastendruck wird wieder also Richtungsfestlegung interpretiert. Und dann gibt es noch den langen Tastendruck. Hiermit zeigt man dem System an, dass man den Stepperarm zum Nullpunkt gefahren hat. Nach dem Loslassen des Tasters fährt dann der Stepperarm von dieser Position auf die entgegengesetzte, also die andere Weichenstellung.

Erweist sich diese Fahrt des Stepperarms als zu kurz, also die Weiche nicht sicher umgeschaltet hat, so kann man über das Konfigurationsmenü im CANguru-Server diese Weglänge korrigieren.

Wie bereits bei der Beschreibung des Steppermotors erwähnt, gibt es einige Exemplare, deren Laufrichtung abweicht. Dies äußert sich darin, dass entgegen der o.a. Abbildung der Stepperarm beim Einfachklick nicht vom Motor weg- sondern zum Motor hinläuft. Das wird dadurch geheilt, indem man vier kurze Klicks eingibt. Dadurch wird die Laufrichtung umgedreht. Anschließend gibt man noch einen Dreifachklick ein, um alles wieder auf Anfang zu setzen, und der Einmessvorgang startet wie oben beschrieben.

Hier die Klickmöglichkeiten im Überblick:

• Einfacher Tastendruck:

Die Laufrichtung wird festgelegt auf das motorferne Ende. Beim nächsten Klick läuft der Motor, beim dritten Klick steht der Motor und ist in die umgekehrte Richtung umgeschaltet.

• Zweifacher Tastendruck:

Die Laufrichtung wird festgelegt auf das motornahe Ende. Beim nächsten Klick läuft der Motor, beim dritten Klick steht der Motor und ist in die umgekehrte Richtung umgeschaltet.

• Dreifacher Tastendruck:

Falls man nach Festlegung der Laufrichtung feststellt, dass die eingestellte Laufrichtung falsch war, kann hiermit die Festlegung widerrufen.

• Vierfacher Tastendruck:

Für Motoren, die hinsichtlich der Laufrichtung umgekehrt reagieren, kann die Laufrichtung dauerhaft umgedreht werden.

• Langer Tastendruck (mindestens 1,5 Sekunden):

Der Motor steht. Mit dem dreifachen Tastendruck wird der Nullpunkt an der aktuellen Stelle festgelegt. Nach Loslassen des Tasters läuft der Motor an das motornahe Ende.

Damit Sie immer auf dem Laufenden sind, in welche Richtung der Motor drehen will, können Sie das USB-Kabel angesteckt lassen. Öffnen Sie ein Terminalprogramm, wie etwa Tera Term, mit dem aktuellen COM-Port und Sie können die Meldungen verfolgen, die der kleine Rechner abgibt. Das geht natürlich nur, solange die Platinen noch nicht eingebaut sind.

Über den Bedarf für das Einmessen hinaus habe ich dem System noch einen fünffachen Tastendruck spendiert. Damit vollzieht der Rechner einen Systemneustart. Obwohl der kleine Rechner ziemlich zuverlässig arbeitet, kann das dann doch manchmal notwendig werden.

# Anpassen der Weiche

Wenn die Firmware auf den ESP32 geladen und der Motor eingemessen ist, sind nur fast am Ende. Es fehlen noch einige Parameter, die Sie im CANguru-Server auf dem Reiter *Konfiguration* einstellen können.

Ein Bild, das Text, Screenshot, Software, Display enthält.

Automatisch generierte BeschreibungDenn um die Weichen auf der Anlage unterscheiden zu können, müssen sie natürlich mit einer frei wählbaren (zwischen 1 und 255), aber eindeutigen Adresse versehen werden. Weiterhin kann es sein, dass Ihnen der Stepperarm bzw. der Umschaltvorgang der Weiche zu langsam oder auch zu schnell läuft. Auch das kann in Grenzen eingestellt werden. Weiterhin gibt es noch die Möglichkeit, die Gesamtzahlzahl der Schritte, die der Steppermotor in einem Umschaltvorgang unternimmt, zu verändern. Diese Zahl kann in den meisten Fällen so stehen bleiben.

Um diese Daten nun dauerhaft im Decoder zu speichern, klicken wir auf eine Decoderzeile in der Listbox.

Weiterhin finden Sie hier auch den Knopf, mit dem Sie den OTA-Prozess starten. Weiter gibt es einen Knopf, der mit *Reset Decoder* benamst ist. Nach einem Drücken dieses Knopfes werden die Parameter des ausgewählten Decoders auf seine Anfangswerte zurückgesetzt. Interessant für die Dokumentation Ihrer Modellbahn ist der Knopf *Decoderdaten speichern*. Hiermit erzeugen Sie eine Liste aller Decoder samt zugehörigen Parametern. Sie finden eine Datei CANguru.txt im Verzeichnis …\0102-CANguru-Server-Version-3.5\CANguru\bin\Release\CANguru, die Sie mit Bordmitteln ausdrucken können.

# Zusammenbau der Weiche

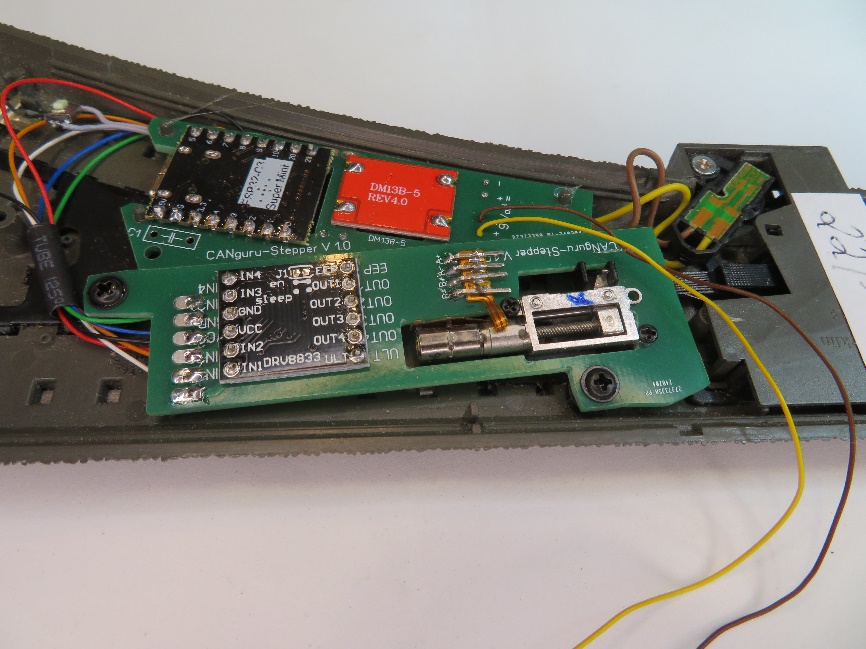
Falls die Weiche mit einer Laterne ausgestattet ist, werden zunächst deren Anschlüsse entsprechend gekürzt und festgelötet. Anschließend kann die CPU-Platine mittels Klebepads unter der Weiche gesichert werden. Alternativ wird die Platine in die kleinen Zapfen gesteckt und mit einem winzigen Tropfen Klebstoff festgehalten. Die Stepperplatine kann nicht direkt aufgesteckt werden. Stattdessen werden zunächst die beiden Abstandshalter, die mit dem 3D-Drucker erzeugt wurden, aufgesteckt und dann wird die Platine mit zwei Linsenschrauben mit Tellerkopf festgeschraubt.

Bei den fünf Varianten werden die Platinen in unterschiedlichen Positionen eingebaut, aber der Hergang ist immer der gleiche.

Die folgenden Bilder zeigen den Einbau in eine Bogenweiche, eine gerade Weiche und eine Kreuzungsweiche.

Ein Bild, das Elektronik, Elektrische Leitungen, Elektronisches Bauteil, Elektrisches Bauelement enthält.

Automatisch generierte Beschreibung



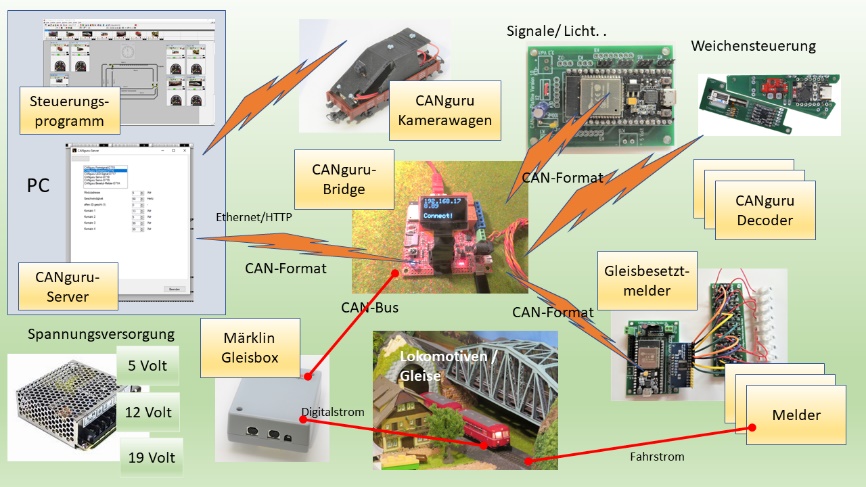
Ein Bild, das Elektronik, Elektronisches Bauteil, Elektrisches Bauelement, passives Bauelement enthält.

Automatisch generierte Beschreibung

# Das CANguru-System

Wenn Sie die Software aus dem Github-Verzeichnis *20-CANguru-3.0* (siehe Kapitel Installationsprogramm) heruntergeladen haben, ist Ihnen möglicherweise aufgefallen, dass dort neben der Weichensteuerung und dem Installationsprogramm weitere Verzeichnisse zu finden sind.

Insgesamt machen diese Verzeichnisse eine vollständige Sammlung von Programmen aus, die zusammen mit dem Programm WinDigiPet die Steuerung einer Modellbahn ermöglichen. Alle Programme basieren auf dem CANguru-System, das ich vor einigen Jahren in dieser Zeitschrift vorgestellt habe. Bei diesen aktuellen Programmen handelt es sich um fortgeschrittene Versionen, in die die Erfahrung von zwei Anlagen eingeflossen sind.

Der Name CANguru rührt daher, dass das System auf dem Märklin-CAN-Bus aufsetzt. Es besteht aus der zentralen CANguru-Bridge, Decodern und dem CANguru-Server. Die CANguru-Bridge basiert auf einem ESP-Modul mit integrierter CAN-Schnittstelle und die Decoder (außer der Weichensteuerung; die hat wie oben beschrieben ein abweichendes Layout) auf einer von mir entwickelten Platine, die jeweils mit dem günstigen ESP32-WROOM bzw. dem ESP32 -C3 für die Weichensteuerung läuft. Diese Platinen können für unterschiedliche Zwecke wie Gleisbesetztmelder oder in Kürze für eine Signalsteuerung eingesetzt werden. Je nach Einsatzzweck werden sie mit anderen Bauteilen bestückt und programmiert. Neben den ESP-Sketches habe ich mit dem CANguru-Server noch eine Erweiterung für PC-Steuerprogramme geschrieben, um die CANguru-Komponenten zu verwalten.

Alle Entwicklungsergebnisse (Platinenlayout, die Software sowie weitere Hinweise wie 3D-Druckdateien) stelle ich auf github.com/ CANguru-System/20-CANguru-3.0 zum Herunterladen zur Verfügung. Auch Weiterentwicklungen lege ich dort stetig ab. Zwei Anlagen, eine kleine und eine mittelgroße, sind bei mir seit einiger Zeit mit den CANgurus zuverlässig in Betrieb. Und das Beste daran ist: Wenn mir irgendetwas nicht gefällt, kann ich es einfach ändern.

## Die CANguru-Bridge

Der Dreh- und Angelpunkt des Systems ist die CANguru-Bridge. Sie stellt die Verbindung zwischen der Steuersoftware der Modellbahn auf einem PC und den Bahnkomponenten her. Alle Befehle, die der Nutzer über das Steuerungsprogramm an die Loks, Weichen oder Signale gibt, werden im Märklin-CAN-Format über ein Ethernetkabel an die CANguru-Bridge und von dort auf den physikalischen CAN-Bus gelegt, wenn es sich um ein Signal für die Loks handelt. Auf diesem Weg erhält die Gleisbox ihre Kommandos. Sie erzeugt das entsprechende Signal und führt es über das Gleis an die Lokomotiven. Andere Kommandos, die für die CANguru-Decoder bestimmt sind, werden nicht über die Gleisbox sondern über ESP-NOW drahtlos direkt an die Decoder weitergeleitet. ESP-NOW ist ein spezielles Funkprotokoll von Espressif, das unabhängig vom heimischen WLAN arbeitet.

Für die Bridge nutze ich das ESP32-Modul Olimex ESP32EVB mit CAN-Schnittstelle. Zusammen mit der Steuerprogrammerweiterung CANguru-Server reicht dies bereits aus, um Loks zu lenken. Die Bridge kann außerdem Informationen, wo sich die Züge befinden, aus dem Bahnsystem aufnehmen und an das Steuerungsprogramm weitergeben. Dazu gibt es Gleisbesetztmelder.

## CANguru-Server

Quasi die Mensch-Maschine-Schnittstelle der CANguru-Bridge ist der CANguru-Server, der die Steuersoftware ergänzt. Das Programm ist in C# geschrieben und läuft derzeit nur auf Windows. Ich nutze als Steuerprogramm das kostenpflichtige WinDigiPet. Mit dem CANguru-Server „verwalte“ ich meine ESP-Module, denn alle Decoder benötigen bei ihrem ersten Einsatz Angaben, wie sie einzusetzen sind. Das ist beispielsweise eine Adresse oder mit welcher Geschwindigkeit die Weichen umgeschaltet werden sollen. Über den CANguru-Server können die Module auch drahtlos (Over The Air, OTA) neu programmiert werden.

Zunächst werden die Bridge und die Decoder durch Anlegen der Betriebsspannung in Betrieb genommen. Dann wird der CANguru-Server auf dem PC gestartet, der eine Verbindung mit der Bridge und dann zwischen der Bridge und den Decodern herstellt. Danach kann das Steuerungsprogramm, also WinDigiPet, gestartet werden. Nun werden alle Befehle wie Lok-Geschwindigkeit, Weichenstellen etc. vom Steuerungsprogramm direkt über das Ethernet-Kabel an die Bridge und von dort an die zuständigen Decoder oder an die Gleisbox gesendet. Der CANguru-Server spielt jetzt nur noch eine dokumentierende Rolle, in dem die auszuführenden Befehle angezeigt werden.

## ESP-NOW als Alternative zum WLAN

Beim CANguru-System wird intensiv vom ESP-NOW Protokoll Gebrauch gemacht. ESP-NOW ist ein drahtloses Kommunikationsprotokoll der Firma Espressif, mit dem mehrere ESP32-Module Daten ohne ein WLAN austauschen können. Um es einzusetzen, müssen die beteiligten Geräte zunächst einen Pairing-Prozess abschließen. Anschließend besteht eine sichere Verbindung und es sind keine weiteren Schritte wie etwa Handshaking nötig. ESPNOW funktioniert als Primary/Secondary-System, mit der CANguru-Bridge als Primary-Controller und den CANguru-Decodern als Secondaries.

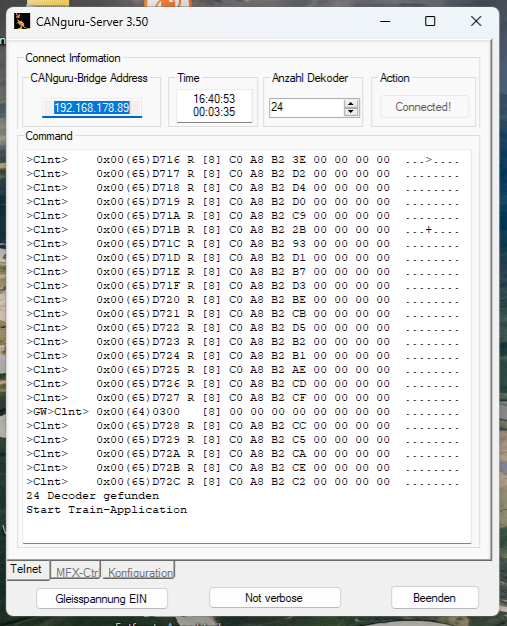
Der Pairing-Prozess läuft in etwa folgendermaßen ab: Alle Decoder senden für bestimmte Zeit ihre eindeutige MAC-Adresse. Die CANguru-Bridge empfängt die Adressen und sendet jeweils eine Quittung. Leider ist die Anzahl der Netzteilnehmer zunächst auf 20 begrenzt. Mit 19 Decodern (zusammen mit der Bridge sind es dann 20 Teilnehmer) kann man aber noch eine mittelgroße Modellbahn ausstatten. Mit dieser Obergrenze werden wir uns später noch beschäftigen. ESP-NOW bietet auch die Möglichkeit, die Kommunikation zu verschlüsseln – dabei wird aber die Anzahl der Module halbiert, die mit der Zentrale kommunizieren können. Da ich die Wahrscheinlichkeit als gering ansehe, dass es Hacker gibt, die meine Modellbahn unbedingt entgleisen lassen wollen, habe ich mich für die 20 Module entschieden.

## Die Decoder

Alle Decoder, außer der hier vorgestellten Weichensteuerung, habe ich auf der CANguru-Platine mit dem ESP32-WROOM als zentralem Baustein aufgebaut. Je nach Einsatzgebiet wird sie entsprechend um wenige Bauteile, wie etwa Spannungsregler, Kondensatoren oder den PCA9685-Controller zur Ansteuerung von LEDs, ergänzt. Für die Spannungsversorgung gibt es einen 5-Volt-Anschluss. Dies ist die einzige eingehende Leitung, sodass beispielsweise ein Servo-Decoder zusätzlich nur Kabel für die einzelnen Servos braucht. Die notwendigen Infos kommen wie beschrieben drahtlos über ESP-NOW.

Was die Software anbelangt, so haben die unterschiedlichen Decoder auch unterschiedliche Software, allerdings mit einem gemeinsamen Kern, einer Art Betriebssystem, der sie befähigt, mit den anderen Komponenten zu kommunizieren und die benötigten Daten bereitzustellen. Die Sketche habe ich in der Programmiersprache C++ in der kostenlose Programmierumgebung Visual Studio Code mit PlatformIO erstellt. Um das Rad nicht jedes Mal neu erfinden zu müssen, nutze ich, wenn möglich, Bibliotheken aus der Arduino- bzw. ESP32-Welt.

# Die Stepper-Weichensteuerung

Die Firmware, die wir mit dem Installationsprogramm auf die Rechnerplatine bzw. den dort verbauten ESP32-C3 geladen haben, besteht im Wesentlichen aus zwei Funktionsblöcken. Da ist einmal der Teil, der nach dem Start des CANguru-Servers für die Verbindungsaufnahme zur CANguru-Bridge zuständig ist; nennen wir ihn den Verwaltungsanteil. Und dann den Betriebsanteil, der ständig horcht, ob ein Weichenbefehl gesendet wurde.

Zum Verwaltungsteil muss nichts mehr gesagt werden. Die notwendigen Erläuterungen wurden bereits in der obigen Beschreibung des CANguru-Systems geliefert.

Aber auch der Betriebsanteil ist recht übersichtlich. Verfolgen wir einen Weichenbefehl. Den Anstoß gibt ein Klick auf eine Weiche in WinDigiPet. Geben wir ihr die Adresse 24 (hexadezimal ist das 0x18). WinDigiPet schickt dazu den Befehl

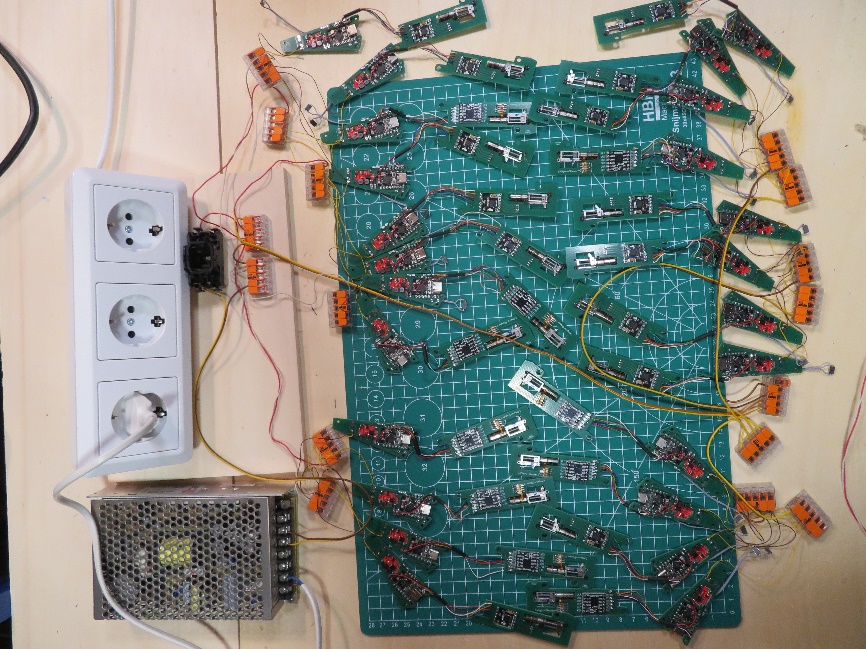
0x00(16)0300 [8] 00 00 30 17 00 01

an die Bridge, die ihn ungeändert an die Decoder verschickt.

Die Adresse wurde von WinDigiPet etwas modifiziert. Es ist die Adresskonstante 0x3000 hinzugekommen, dafür wurde die 0x18 um 0x01 reduziert. Vollständig lautet nun die Adresse 0x3017. Die Weichen wissen, dass Weichenbefehle (erkenntlich an der 0x16) nur an sie (und Signaldecoder) gerichtet sind und überprüfen alle, ob die Adresse 0x18 für sie zutrifft. Der Decoder mit der zutreffenden Adresse wertet nun das Byte hinter der Adresse aus und stellt eventuell fest, dass der Wert dort nicht mit der gespeicherten Stellung der Weiche übereinstimmt und gibt diese Information weiter. Dadurch wird eine neue Zielposition für den Steppermotor festgelegt. Schon beginnt der Stepper zu laufen. Ob er schnell oder auch langsamer läuft, hängt von der Länge der Pausen ab, die in diesen Lauf eingeschoben werden.

An dieser Stelle muss noch auf die ESP-NOW Obergrenze mit 20 Decodern eingegangen werden. Bereits in einer kleinen Anlage ist diese Einschränkung nicht tolerabel, zumal mit der hier beschriebenen Weichensteuerung ein Decoder pro Weiche benötigt wird. Doch Entwarung! Die Obergrenze kann relativ einfach umgangen werden, indem man sie gar nicht überschreitet. Der Trick ist, dass man unabhängig von der tatsächlichen Anzahl Decoder niemals mehr als zwanzig anmeldet. Ich habe sogar nur immer keinen einzigen angemeldet. Immer dann, wenn eine Kommunikation notwendig wird, wird der eine angemeldet, diese Kommunikation abgewickelt und sofort wieder abgemeldet. Der Rückweg ist unproblematisch, da die CANguru-Bridge die MAC-Adressen aller Decoder kennt und deshalb auch richtig zuordnen kann.

In der Theorie alles plausibel. Aber funktioniert das auch wirklich? Denn vertiefende aufschlussreiche Dokumentation über ESP-NOW ist leider nicht verfügbar. Also wurde ein umfangreicher Test mit 24 Decodern gestartet, um das Verhalten auszuprobieren. Der Test hat dann doch die Theorie bestätigt. Alle 24 Weichendecoder lassen sich problemlos ansprechen.

Da Modellbahnen in gewisser Weise auch als kritische Infrastruktur einzustufen sind, habe ich diverse Sicherungsmaßnahmen in die Software eingebaut. Denn es kann fatale Folgen haben, wenn ein Decoder aus meist unerfindlichen Gründen im Betrieb nicht mehr reagiert oder bei der Anmeldung sich erst gar nicht meldet.

Deshalb kann man die Anzahl der Decoder, die in der Anlage verbaut sind, nun im CANguru-Server in der oberen Zeile eintragen. Beim folgenden Anmeldeprozess besteht das System auf die eingestellte Anzahl und macht 10 Versuche, diese Vorgabe zu erfüllen. Erst dann läuft das System weiter und zeigt abschließend, wie viele Decoder tatsächlich gefunden wurden. Falls wirklich ein verbauter Decoder, der beim vorigen Anmeldeprozess gefunden wurde, sich diesmal nicht meldet, wird dies angezeigt. Weiterhin werden alle Decoder im laufenden Betrieb zyklisch abgefragt. Wenn sich einer nicht mehr meldet, wird diese unverzüglich angezeigt.

Alle diese Maßnahmen haben dazu geführt, dass meine kleinen Preiserlein in den D-Zugwagen sich keine Gedanken über falsch gestellte Weichen machen müssen.

Viel Spaß beim Nachbauen!