

DCC Zentrale

Die Digitalzentrale ist das Steuerungselement einer digitalen Modellbahn. Die Digitalzentrale bedient diverse Handregler und versorgt die Fahrzeuge mit den nötigen Informationen. Bei 2L= hat sich das DCC - Digital Command Control - durchgesetzt. Dabei gibt die Zentrale ein DCC Signal aus, eine Bitfolge als Datenstrom. Dieses Rechtecksignal wird von einem [Booster](#) als Gleissignal verstärkt und auf die Gleise gegeben. In dem Signal befinden sich also die Daten und gleichzeitig werden die Fahrzeuge darüber mit Spannung versorgt. Durch den Takt von fast 10kHz und einer Bitfolge auch im Ruhezustand (d.h. es werden gerade keine Daten übertragen) ergibt sich ein Effektivwert am Gleis, mit dem die Dekoder in den Fahrzeugen betrieben werden, aus denen die Motoren, die Beleuchtung ihre Spannung bezieht.

Neben den käuflichen Digital-Zentralen für die Modellbahn gibt es auch eine Reihe Selbstbauprojekte. Die meisten Selbstbauprojekte beruhen auf den 8 Bit Controllern von Microchip, entweder AVR oder PIC. Ein sehr umfangreiches Beispiel ist die [Z21PG](#) von Philipp Gahtow oder eine etwas einfachere Variante die [DCC-Ex](#). Nicht unerwähnt lassen möchte ich die Arbeit von W. Kufer mit dem [OpenDCC](#) Projekt, eine Quelle der Informationen mit vielen guten Beschreibungen, der Protokolle und der Hardware.

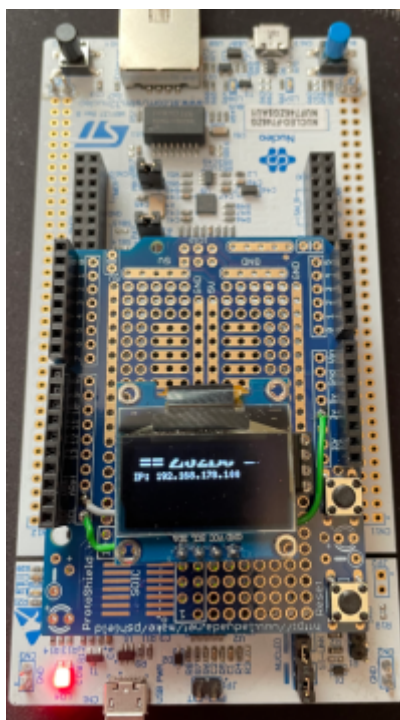
Die 8 Bit AVR Controller sind zwar leistungsfähig, aber gerade in Bezug auf den Speicher und der Netzwerkfähigkeit kommen diese schnell an ihre Grenzen. Die Z21PG und DCC-Ex sind beide Open Source, alle Quelltexte sind veröffentlicht. Auch ich habe erst mit einer Z21PG auf Basis eines ATmega2560 begonnen und wollte mir eigentlich auch für eine Selbstbau-Zentrale eine Leiterplatte machen. Mehr oder weniger per Zufall bin ich in einer meiner vielen Schubladen über den Antennenanalyzer nach EU1KY gestolpert, den ich fürs andere Hobby mal gebaut hatte. Der besteht im wesentlichen aus einem [ST Discovery Modul](#) mit einem STM32 ARM Controller.

So habe ich mich etwas näher mit diesen Controllern beschäftigt und auch die Entwicklungsumgebung dazu installiert, die aus der STM32CubeIDE, dem STM32CubeProgrammer und STM32CubeMX besteht. Alle Tools sind kostenlos auf der Website von STMicroelectronics zu bekommen. Ich habe mich dann bei meinen Experimenten für ein Nucleo-144 Board mit dem STM32F746ZG Controller entschieden. Mittlerweile gibt es einen Nachfolger davon, aber das schöne bei den STM32 Controllern ist die Skalierbarkeit, vom kleinen 8 Pin Controller bis zum F746 mit 204 Pins. Das ist ähnlich wie bei den bekannten AVR Controllern. Mit wenig Aufwand kann man mit seiner Software auf einen leistungsfähigeren Controller umziehen. Nach den ersten Schritten mit CubeIDE bin ich dann auf Visual Studio Code mit PlatformIO und STM32Duino gewechselt. Das hat mich aber gar nicht überzeugt.

Da die Boards gleich Ethernet haben, I/O Pins ohne Ende, tolle AD und DA Wandler, mehrere serielle Schnittstellen, I²C, SPI usw. usf. (einfach mal ins [Datenblatt](#) schauen), sind die für diese Anwendung wirklich gut geeignet. Dazu kommt noch der viele Speicher mit 32Bit Breite, die hohe Verarbeitungsgeschwindigkeit der ARM Controller und der extrem günstige Preis der Boards. Aktive Boards mit Ethernet wären diese hier und es ist im Prinzip egal, welches man wählt, die haben alle den gleichen Formfaktor und das gleiche Layout, um diese auf einer Trägerplatine mit Schnittstellen zu integrieren.

<input type="checkbox"/>	NUCLEO-F207ZG	ACTIVE	STM32F207ZG	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B
<input type="checkbox"/>	NUCLEO-F439ZI	ACTIVE	STM32F439ZI	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B
<input type="checkbox"/>	NUCLEO-F756ZG	ACTIVE	STM32F756ZG	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Mini-B
<input type="checkbox"/>	NUCLEO-F767ZI	ACTIVE	STM32F767ZI	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B
<input type="checkbox"/>	NUCLEO-H563ZI	ACTIVE	STM32H563ZI	Arduino Uno, ST Morpho, ST Zio, USB Type-C FS	100Mbps (IEEE 1588)	ST-LINK V3E, USB Type-C
<input type="checkbox"/>	NUCLEO-H723ZG	ACTIVE	STM32H723ZG	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V3E, USB Micro-B
<input type="checkbox"/>	NUCLEO-H753ZI	ACTIVE	STM32H753ZI	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B
<input type="checkbox"/>	NUCLEO-H755ZI-Q	ACTIVE	STM32H755ZI	Arduino Uno, ST Morpho, ST Zio, USB Micro-AB FS OTG	100Mbps (IEEE 1588)	ST-LINK V2-1, USB Micro-B


Erste Versuche liefen bisher erfolversprechend. Hier ein erstes Foto des Nucleo-144 Boards mit OLED Display und EEPROM. Ich habe hier ein EEPROM für die Speicherung von Konfigurationsdaten vorgesehen, auch wenn man im Flash des Controllers speichern könnte. Nur ist der Flashspeicher nicht beliebig oft wieder beschreibbar und bei den Controllern liegt der Wert bei 10.000x im Gegensatz zum EEPROM mit > 100.000x. Das die Anzeige vom Display so seltsam aussieht, liegt an der niedrigen Wiederholrate.



So eine Softwareentwicklung, auch nur die Umsetzung von bestehenden Code auf einen anderen Controller, kann sehr zeitaufwändig sein. So sollte man immer überlegen, ob sich das lohnt, diesen Aufwand zu betreiben. Denn egal ob Z21PG oder DCC-Ex, das ist erst mal nur Software. Dazu muss man sich die Hardware anschaffen, evtl. gar noch Leiterplatten machen lassen, vorher die Schaltpläne zeichnen und die Software umsetzen, auch wenn vieles auf vorhandenen Arduino Libraries basiert,

wie NMRADCC und LocoNet zum Beispiel.

Daher sollte die Überlegung folgen, ob man nicht doch besser eine Zentrale kauft und seine evtl. knappe Freizeit lieber in den Bau einer Modellbahnanlage und vielleicht kleineren Projekten, z.B. einem Booster oder der Steuerung kleinere Gimmicks auf der Anlage, investiert. Denn abgesehen vom Zeitaufwand entstehen auch noch Materialkosten, wenn es auch nicht viel ist, aber mit der Fertigung von Leiterplatten können schnell 100-200€ über den virtuellen Ladentisch gehen. Dazu kommen noch solche Dinge wie Gehäuse und Netzteil, womit sich die Differenz zu einer käuflich zu

erwerbenden Zentrale auf 150-300€ reduziert und die funktioniert dann einfach nur (hoffentlich ). Auch sollte man sich zumindest mit Grundlagen der Elektronik auskennen, wissen an welchem Ende der LötKolben heiß wird und sich schon mal mit der Arduino IDE beschäftigt haben. Wenn man wirklich Interesse an der Software hat, es gibt mindestens eine Zentrale auf dem Markt, bei der vom Hersteller die Firmware unter der [GPLv2](#) zur Verfügung steht.

Nachteilig bei den am Markt angebotenen Zentralen finde ich die zum Teil langwierigen Release Zyklen der Software. Bei einem Eigenbau kann man auch einen Fehler kurzfristig selbst beheben. Bei einer gekauften Zentrale ist man davon abhängig, wann der Hersteller ein Update herausbringt.

Einen Funktionsvergleich einiger Zentralen habe ich [hier](#) mal aufgestellt. Leider sind nicht alle Daten seitens der Hersteller in einem Produktdatenblatt aufgeführt, es ist zum Teil mühsam, sich einen Überblick zu verschaffen.

Fazit

Mittlerweile ist einiges an Zeit vergangen, ich habe vieles gelesen, vor allem über die Bussysteme für das Schalten und Rückmelden. Aktuell unterstützen die gängigen Zentralen, bis auf die mc² von Tams, nur die älteren Bussysteme, zum Teil über 25 Jahre alt. Nicht das die schlecht sein müssen, nur weil sie älter sind, aber die Technik geht weiter und es gibt neue Entwicklungen. Auch sind in dem vergangenen Jahr, seit dem es die erste Version von diesem Text gab, ein paar, wenn auch nur wenige neue Produkte am Markt erschienen. So habe ich meine Anforderungen an die Modellbahnsteuerung genauer definiert:

- Fahren nur mit DCC, eine Multiprotokoll Zentrale ist nicht erforderlich
- Schneller Datenbus für das Schalten & Rückmelden
- lokale Stellpulte am Ort des Geschehens (mit Display oder als Stellpult mit Tastern & LEDs)
- Steuerung mit Handreglern, nicht in einer "Box" die an einer Stelle der Anlage steht
- offenes System in Bezug auf Hard- und Software
- Selbstbau von Komponenten möglich
- gute Dokumentation

Daraus hat sich ergeben, dass ich den Selbstbau einer Zentrale ausschließe, wie oben schon geschrieben, auch auf Grund des Zeitaufwandes (und für den Fall der Fälle, bzw. für mein Testgleis habe ich noch den Nachbau der [Z21PG](#)). Für das Schalten & Rückmelden kommt weder DCC, CAN, LocoNet und XpressNet in Frage. DCC eignet sich nur zum Schalten, für die Rückmeldungen wäre ein weiterer Bus erforderlich. LocoNet ist proprietär, schlecht dokumentiert und auf Grund des Alters ohne wesentliche Weiterentwicklungen das langsamste Bussystem (16.6kBaud). CAN ist zwar ein Industriestandard, aber wird im Bereich Modellbahn bei 2L= kaum verwendet. XpressNet ist gut dokumentiert, aber eigentlich kein Rückmeldebus, sondern für Handregler und durch den

Industriestandard mit RS485 auch sicher, was gerade in den letzten Jahren durch die Zunahme von elektromagnetischen Störungen (Schaltnetzteile, Wechselrichter von Solaranlagen usw.) ein wichtiger Aspekt ist.

Konzept

Ich werde einen Raspberry (4 oder 5) mit Rocrail einsetzen, was praktisch die Zentrale darstellen wird. Bei meinem Konzept wird aber nicht zwingend ein Display erforderlich sein, denn es genügt, wenn der Rocrail Server läuft, nachdem die Konfiguration abgeschlossen ist. Alternativ gibt es aber eine Menge an Angeboten für Touch-Displays mit Gehäuse für den Raspberry, von 2,8" bis 16". Mit einem im Handel erhältlichen 10" Touch-Display mit 1920x1200px, bei dem im Gehäuse auch Platz für den Raspberry vorgesehen ist, wäre das schon ein Vielfaches mehr, als jede am Markt erhältliche Zentrale mit Display bietet.

An dem Raspberry werde ich das [BiDiB-IFnet](#) von Fichtelbahn betreiben. Das wird über Netzwerk (TCP/IP) gesteuert und stellt mit BiDiB den Datenbus für das Schalten & Rückmelden dar. Ein eingebauter DCC Generator liefert über BiDiB das DCC Signal für die Booster. Zusätzlich hat das Interface einen XpressNet Anschluss für Handregler (Lenz LH101, Roco Multimaus oder Eigenbau).

Warum BiDiB?

Als ITler mit Jahrzehnten Erfahrung in der industriellen Steuerung halte ich [BiDiB](#) für das modernste, schnellste und einfachste Konzept für Schalten & Rückmelden bei der Modellbahn. BiDiB arbeitet mit RS485 bei 500kbaud Übertragungsrate. Durch die Verwendung von Hubs ist es nicht auf max. 32 Knoten beschränkt. Die Konfiguration der Module ist Plug&Play, nach dem Pairing muss man nicht mehr mit Adressen hantieren, auch nach einem Umbau nicht. Die Anbindung über TCP/IP vermeidet Probleme mit USB. Ein Raspberry mit Rocrail unter Linux ist stabil und Remote bedienbar (von einem anderen PC, von einem Tablet) oder lokal mit einem Touch-Display. Somit ist das System frei konfigurierbar, durch fertige Komponenten wird es nur zusammengesteckt. Mit der Rocrail App steht auch der Verwendung eines Smartphones oder Tablet nichts im Wege.

Der bisher nicht in mein Konzept passende Ansatz einer [zentralen](#) Instanz wurde durch die fachlich fundierten Antworten im [OpenDCC](#) Forum (nur für angemeldete Benutzer, deswegen hier kein Deep-Link) entkräftet. Denn genau genommen ist jede Zentrale ein zentraler Ansatz, daher auch der Name



Bei dem geplanten Konzept übernimmt die Funktion der Zentrale der Raspberry mit Rocrail als Software, die über BiDiB die Modellbahn mit dem DCC Signal, dem Schalt- und Rückmeldebus auf einer Leitung versorgt und an dem mobile Bedienteile über XpressNet angeschlossen werden können.

Ich halte dieses Konzept für eine moderne Möglichkeit zur Steuerung einer Modellbahnanlage mit freien Fahren und Überwachung durch die zentrale Software, in der alles zusammenläuft. Darum habe ich mich letztendlich für dieses Konzept entschieden und nicht zum Kauf einer Multiprotokollzentrale.

Die weiter oben schon verlinkte [Übersichtstabelle](#) von gängigen Zentralen werde ich schon aus eigenem Interesse an der Technik weiterhin pflegen.

[BiDiB Wiki Arbeitskreis BiDiB Einstieg BiDiB](#)

From:
<https://isnix.de/> - **It's boring when it works!**

Permanent link:
<https://isnix.de/doku.php?id=modellbahn:zentrale>

Last update: **2025-03-30 18:50**

