Modellbahnsteuerung

Wie bereits beim Vergleich der diversen Digitalzentralen erwähnt, verwende ich für meine Modellbahn das 2L Gleichstrom System und die Fahrzeuge werden mit DCC gesteuert. Signale, Weichen und alle Rückmelder für Gleisbesetztmeldungen und Weichenstellungen werden über ein anderes Bussystem gesteuert und gemeldet. Hier habe ich lange überlegt, welches System ich verwenden möchte. Die meisten Digitalzentralen sind Multiprotokollzentralen für alle möglichen Gleisformate zur Steuerung der Fahrzeuge und einige Digitalzentralen haben auch mehrere Bussysteme für Rückmeldungen und Handregler. Von Anfang ist aber auch ein Aspekt der Selbstbau von Elektronik für die Modellbahnsteuerung. Dafür ist in der Regel eine gute Dokumentation des Datenprotokolls erforderlich. LocoNet ist z.B. weit verbreitet, aber eine vollständige Protokollspezifikation gibt es nicht, nur eine sehr alte Personal Edition. XPressNet[1)] und BiDiB sind recht gut dokumentiert.

Natürlich gibt es auch andere Möglichkeiten wie dem Verzicht auf ein Bussystem. Weichen und Signale können auch über einfache Schalter gesteuert werden, Belegtmeldungen können mit LEDs signalisiert werden. Auch rein mechanisch können Weichen und Signale über Hebelwerke gestellt werden. Ohne ein Bussystem ist der Verdrahtungsaufwand nicht zu unterschätzen und von eine mechanische Stellmöglichkeit ist für wenige Weichen und Signale interessant, aber bei mehreren Komponenten sehr aufwändig.

Nach einigen Experimenten mit der Z21PG und DCC-Ex tendierte ich Richtung fertiger Zentrale, wie z.B. die mc² oder Intellibox. Letztendlich bin ich wieder auf den Pfad der Tugend gelangt 😉

Ich würde aber gerne alles, was ich für meine Modellbahn baue, hier auf der Website veröffentlichen - Schaltungen, Quelltexte usw. - einfach alles. Das möglichst unter einer freien Lizenz, wie z.B. der GPL. Nach einiger Recherche bin ich bei der Model Electronic Railway Group (MERG) fündig geworden. Für den vollen Zugriff ist zwar eine Mitgliedschaft erforderlich, die aber mit 2,30€ im Monat moderat ausfällt. Dafür bekommt man Zugang zum Forum & Wiki und bekommt die vierteljährlich erscheinende Clubzeitschrift. Die Quelltexte für alle Bausätze sind bei Github erhältlich oder auch im umfangreichen Wiki. So ziemlich alles wird unter der CC BY-NC-SA 4.0 oder der GPL veröffentlicht. Unter der Lizenz kann alles völlig frei verwendet werden, solange auf die Quelle verwiesen wird, die gleiche Lizenz verwendet wird und es nicht kommerziell ist. Perfekt!

Nun ist eine freie Lizenz für den Selbstbau und die Mischung von fertige erhältlichen Geräten (beim MERG nur als Bausatz) mit Eigenbau das eine, die Verwendbarkeit für die Modellbahn, bzw. für die eigenen Ansprüche das andere. Also habe ich mich eine Weile mit dem Konzept der Steuerung mit dem CBUS[2)] und dem abwärtskompatiblen VLCB beschäftigt. Es wird eine Philosophie verfolgt, die ich mit der Unix-Philosophie vergleiche, in Hardware und Software. Baue kleine Module, die nur eine oder wenige Funktionen haben, aber diese gut machen. Ein anderer Aspekt bei dieser Modellbahnsteuerung ist der, ich bin unabhängig von Hersteller, von Release-Zyklen, kann die Geräte selbst reparieren und selbst bauen. Ganz wie es mir gefällt.

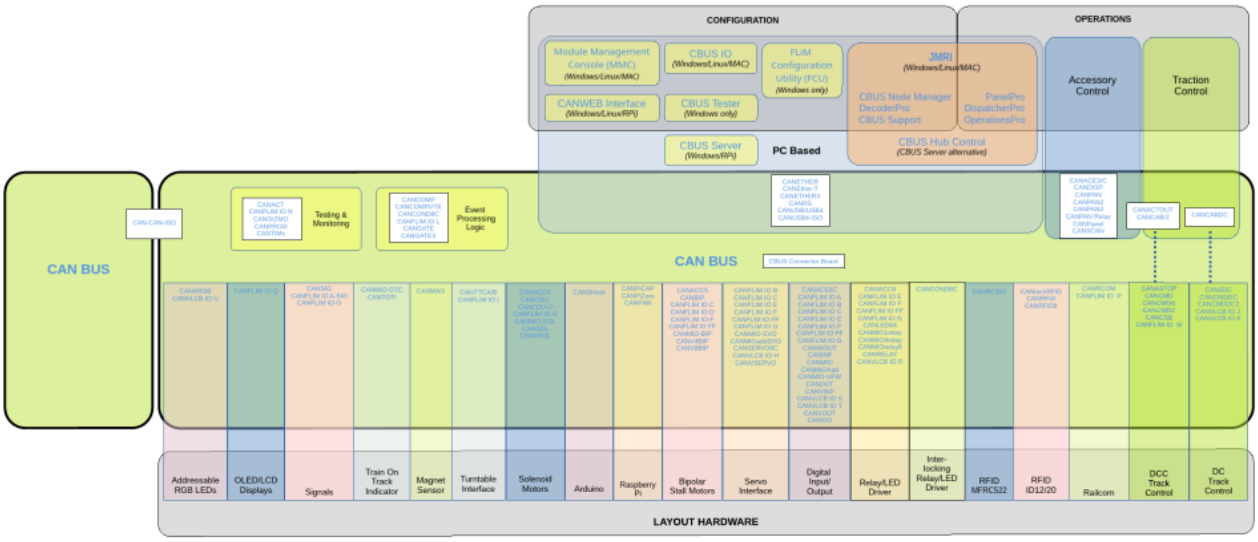
So habe ich mich entschlossen, die im Bau befindliche Modellbahn mit CBUS/VLCB zu steuern.

CBUS / VLCB

Der CBUS[3] und auch der abwärtskompatible VLCB[4] , was im Prinzip eine Erweiterung des CBUS ist und gemischt auf dem gleichen Datenbus betrieben werden kann, basieren beide auf CAN - Controller Area Network. Der CAN Bus ist aus der Automobilbranche bekannt und wurde 1983 von Bosch entwickelt. Heute ist dieser Bus aus keinem neuen Auto mehr wegzudenken. Es handelt sich dabei um einen sogenannten Producer/Consumer oder auch Multi-Master Bus. Die Übertragungsrate beträgt beim CBUS und VLCB 125kBit/s (CAN geht bis 1MBit/s). Dabei wird mit Standard-CAN, also 8 Byte Nutzdaten und 11-Bit Identifier gearbeitet.

Wenn ein Modul zum Beispiel ein Belegtmelder für einen Gleisabschnitt im Schattenbahnhof ist, werden die Ereignisse "Gleis frei" oder "Gleis besetzt" auf den Bus gegeben. Es wird dabei keine Gegenstelle adressiert, weil der Bus nur mit Events (Ereignisse, die ein Modul auch lernen kann) arbeitet. Das jeweilige Modul, was mit der Meldung "Gleis belegt" etwas anfangen kann, wird dieses Ereignis verarbeiten. Das Ereignis kann auch von mehreren Modulen verarbeitet werden, z.B. könnte ein Schaltmodul die Weichen schalten, so das kein weitere Zug in das Gleis einfahren kann und ein anderes Modul schaltet die LEDs im Stellpult zur Visualisierung der Belegtmeldung an. Die Anzeige kann z.B. ein kleines Stellpult mit visualisiertem Gleisplan mit Taster für Weichen und Signale und LEDs für die Anzeige sein. Ein Taster für das Umlegen einer Weiche würde dann ein Ereignis auslösen, "Weiche gerade" und das Modul für diese Weiche würde auf das Ereignis reagieren.

Es ist somit keine Zentrale Steuereinheit nötig, kein Computer an der Modellbahn. Auch die Digitalzentrale muss nicht mehr als DCC und den CAN Bus können. Alles wird über den Bus abgewickelt. Vom MERG gibt es dazu eine ganze Reihe an Modulen, mit Baubeschreibung, freier Software und für viele Module auch Leiterplatten oder ganze Bausätze. Eine schematischen Überblick zeigt diese Grafik[5] :



Folgende Bausätze gibt es zur Zeit:

Current CBUS Kits

Kit: link to page	Description	Module Name: link to software page
11	8 channel relay/accessory driver & 8 feedback or switch inputs	CANVOUT
12	16 channel feedback or switch inputs	CANVINP
13	8 channel servo driver & 8 feedback or switch inputs	CANVSERVO
14	4/8 channel constant motor driver & 8 feedback or switch inputs	CANV4BIP
28	8 channel output relay/accessory driver module	CANOUT
29	8 channel input module for feedback or switch inputs	CANINP
80A	USB Interface	CANUSB4
90	CBus Activity Monitor	CANACT
93A	CBUS RJ22 Connector kit with horizontal or vertical sockets (revised kit)	CAB Plug-in Point
93S	Single CBUS RJ22 Connector board with horizontal or vertical sockets (new offering)	CAB Plug-in Point
94	4ch Pulse (Solenoid) Motor Driver	CANSOL
94A	4ch Pulse (Solenoid) Motor Driver	CANSOL
97	12V Control Panel Module (32 switch input, 32 LED output)	CANPAN
97A	12V Control Panel Module with variable brightness (32 switch input, 32 LED output) Under development	CANPAN-VAR
98	Multi-Input-Ouput Module	CANMIO
	8 channel RFID interface	CANRFID8
	8 channel servo drive (CANMIO Servo version)	CANSERVO8
	MAS colour light signal control	CANSIG
98B	Multi-Input-Ouput Module using Rev K board Under development	CANMIO
101A	CBUS Beginners Pack A (New 12v Modules)	CANINP CANOUT LCB CONNECTOR BOARD
102B	CBUS Beginners Pack B (New 12v Modules)	CANVSERVO CANUSB4 CANGizmo
110	ARDUINO CAN Shield	Arduino CBUS Shield
193	CANACTOUT RJ22 & Activity Monitor	CANACTOUT RJ22 & Activity monitor
391	MioADD CANMIO Daughter Board	MioADD CANMIO Daughter Board
392	MioADDSvo CANMIO Daughter Board	MioADDSvo CANMIO Daughter Board
491	New LCB Experimenter's Kit - Switches and LEDs	
492	CBUS Module Bench Tester and Network Monitor	CANGizmo
493	New LCB Experimenter's Kit - BI-Polar	
497	Test module specifically designed to test and set up CANPAN Kit 97	CANTM7
498	Voltage Detector Dongle Kit for CBUS	

CAN-ETHER - Computer interface for Ethernet is being considered for inclusion in the Kit Locker.

Mit diesen Bausätzen ist alles vorhanden was man als Basis für die Steuerung einer Modellbahn benötigt, von einer DCC Zentrale mit CBUS zu Weichen- und Signal Modulen, Multi I/O Modulen und Rückmeldern. Ein Video zu Modulen und der Funktion gibt es hier. Eine kleine Modulübersicht gibt es auch auf der öffentlichen Webseite der MERG.

Die meisten Module lassen sich über den CAN Bus aktualisieren. Die Module haben dazu einen Bootloader und mit der entsprechenden Software auf einem PC oder Raspberry kann eine neue Firmware in das Modul geladen werden. Aktuell wird der CBUS von JMRI und Rocrail unterstützt. Aus dem Rocrail Umfeld gibt es auch eigene CBUS Module. Mit dem node.js basierenden MMC-Server und MMC-Client steht auch ein browserbasierendes Tool für den CBUS bereit, für die Konfiguration und Aktualisierung der Module.

Mittlerweile hat sich auch noch eine weitere Software dazu gesellt: Traintastic. Reinder Feenstra hat sehr schnell die Basis Kommunikation für CBUS/VLCB ins Programm aufgenommen, akt. aber nur in einem eigenen Branch, weil es noch in der Entwicklung ist. Ich habe dazu die ersten Tests mit der Hardware durchgeführt, mit dem CANUSB4-ISO und einem CANVOUT Node vom MERG. Das ist ein Node mit 8 Eingängen und 8 Ausgängen. Über das Stellwerk von Traintastic lassen sich diese bereits schalten. Es ist aber noch einiges an Arbeit erforderlich. Aktuell ist nicht vorgesehen die Konfiguration das Lernen von Events im Programm umzusetzen, dafür gibt es MMC, was dieses sehr gut kann und alle Nodes inkl. Firmware-Update unterstützt. Traintastic ist ein einfaches Steuerungsprogramm, nicht

überladen und mit Lua kann man Steuerungsaufgaben auch als Script integrieren. Aktuell habe für Testzwecke noch die Zentrale CANCMDB und den Handregler CANCAB beim MERG bestellt. Ziel ist, eine vollständige CBUS/VLCB Unterstützung in Traintastic zu implementieren, so dass eine einfach zu bedienende Soft- und Hardware als Open Source Lösung mit vollständiger Dokumentation zur Verfügung steht.



Mittlerweile wird CBUS/VLCB von Traintastic unterstützt und befindet sich im Master Branch. Die Development Builds und die AppImages enthalten alle CBUS/VLCB

Steuerung für meine Anlage

Für meine Anlage sind diverse Module geplant. Als erstes ist natürlich das Interface für einen PC oder Raspberry zu nennen - CANUSB4-ISO. Die Schaltung und Software habe ich weitestgehend übernommen, aber die Schaltung um ein ISO Interface und ein paar Details erweitert. Durch das USB ISO Interface besteht eine galvanische Trennung zwischen der Modellbahnelektronik und einem PC, was Probleme vermeidet. Der Schattenbahnhof soll automatisiert oder halbautomatisiert laufen. Dies will ich, wie schon erwähnt, mit Traintastic auf einem Raspberry 4 realisieren, der über ein CANUSB4 Interface mit den Modulen am CBUS kommunízieren kann.

Weitere geplante Module:

- Rückmelder für Weichenstellung (CANVINP)
- Rückmelder für Belegtkennung von Gleisen (CANVINP)
- Schalt- und Servomodule für Weichen und Signale (CANVSERVO)
- Multi-I/O (CANMIO)
- DCC Zentrale (CANCMD)
- Handregler (CANCAB)



Die Beschreibungen, Links, Dokumentationen werden hier relativ regelmässig dem Entwicklungsstand angepasst, also einfach ab und zu mal schauen, was es Neues gibt!

1)
Ich konnte keine Spezifikation auf der Website von Lenz finden, es gab aber mal ein Dokument XpressNet.pdf für die 4.0 Version.
2)
CBUS protocol documents are a copyright of Mike Bolton and Gil Fuchs
3)
CBUS Spezifikation https://github.com/cbus-traincontrol/cbus-traincontrol.github.io
4)
VLCB Spezifikation https://github.com/Versatile-LCB/VLCB-documents
5)
Grafiken & Photos stammen von der MERG und dürfen frei veröffentlicht werden: https://www.merg.org.uk/resources/copyright

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

Permanent link:
https://isnix.de/doku.php?id=modellbahn:elektronik:steuerung

Last update: **2026-05-15 18:14**