



Decoder selbst bauen, Teil1: Digitalprotokolle

Digitalprotokolle

Mit diesem ersten Teil starten wir eine Artikelreihe über den Selbstbau von Decodern für die Modellbahn. Ziel ist, Sie, liebe Leser, in die Lage zu versetzen, eigene Decoder nach Ihren eigenen Bedürfnissen zu bauen. Bevor es jedoch ans Lötkolbenschwingen gehen kann, sind eine Reihe von Grundlagen zu klären, so z.B.: „Wie funktioniert ein Digitalprotokoll und wie entlocke ich ihm die enthaltenen Informationen?“

In diesem ersten Artikel gibt der Autor einen Überblick über die gebräuchlichsten Digitalprotokolle: Wie sind sie grundsätzlich aufgebaut? Was haben sie gemeinsam? Was ist verschieden?

Im Folgeartikel wird detailliert auf das Selectrix Protokoll eingegangen, damit wir für dieses System selbst mit Hilfe eines Arduinos und der Sprache C einen Decoder entwickeln können.

Die Hardware zum Steuern von Weichen und Signalen wird Thema im dritten Artikel sein: Was können/müssen wir ändern, um die von Ihnen bevorzugten Antriebe einsetzen zu können? Was gibt es für weitere Möglichkeiten?

Hier wird nun nicht mehr der Arduino verwendet, sondern ein preiswerter AVR Prozessor (ATtiny2313).

Im vierten und vorerst letzten Teil befassen wir uns intensiv mit der Software: Eine frei verfügbare Software-Bibliothek sorgt für die Dekodierung des SX-Signals. Auf dieser Basis können wir uns dann auf die Anpassung der Ausgabe konzentrieren und eigene Funktionalitäten entwickeln.

Zu einigen der Selbstbauprojekte sind fertige Platinen vorhanden. Mit ihnen gelingt der Einstieg in den Selbstbau leichter. Aber geben wir nun Michael Blank das Wort:

DIGITALPROTOKOLLE IN DER ÜBERSICHT

Ich beschränke mich hier auf die derzeit gängigsten Systeme. Dies sind DCC (Stichworte „Lenz“ und „NMRA“), Märklin Digital (auch oft „Motorola“ genannt) und Selectrix (auch kurz „SX“). Dabei setze ich den Schwerpunkt zunächst auf die ursprünglichen Protokolle, um die Dinge nicht allzusehr zu verkomplizieren. Auf Erweiterungen bzw. Fortentwicklungen wie Railcom, mfx, SX2, RMX ... gehen wir später ein.

Zunächst ein Blick auf die Gemeinsamkeiten: Diese sind größer, als man

vielleicht annehmen mag und erst sie machen ein Konzept wie einen „Multi-protokoll-Decoder“ möglich.

Alle diese Systeme verwenden ...

- ... auf dem Gleis ein digital moduliertes Signal mit wechselnder Polarität, also ein „Wechselstrom“-Signal von etwa +/-20 Volt. In diesem sind die Informationen über Geschwindigkeit der Lokomotiven, die Stellung der Weichen und Signale codiert.
- ... einen oder mehrere Bussysteme, über den/die Steuergeräte wie z.B. Handregler, Melder, Zubehördecoder mit der Zentrale gekoppelt werden (siehe Schwerpunkt Dimo 4/2012).

Bei einigen Systemen können Decoder zur Ansteuerung von Weichen etc. sowohl direkt mit dem Gleissignal als auch über einen Bus gesteuert werden.

Achtung: Eine Normung (z.B. DCC) suggeriert, dass alles DCC-Equipment mit allem DCC-Equipment verwendet werden kann. Die Normung bezieht sich aber nur auf das Gleissignal. Das heißt: Alle Loks mit DCC-Standarddecoder sollten mit jeder DCC-Zentrale fahren können. Jedoch ist der Handregler von Hersteller 1 mit grosser Wahrscheinlichkeit nicht an einer Zentrale von Hersteller 2 zu verwenden. Auch die herstellerabhängigen Erweiterungen der jeweiligen Standards, die in vielen Fällen nicht offengelegt sind, verhindern die freie Kombination von Komponenten verschiedener Hersteller. Wir haben im Modellbahnbereich leider noch immer eine Situation, wie sie etwa bei den Personal-Computern war, bevor normierende Kräfte aufgrund der Marktmacht der zusammenarbeitenden Firmen faktisch unumgängliche Standards setzten. (Die älteren unter den Lesern werden sich noch an diese Zeiten erinnern, während die jüngeren sich wahrscheinlich gar nicht vorstellen können, dass man zu manchen Zeiten eine PCI-Grafikkarte eines beliebigen Herstellers nicht in einen PC eines anderen Herstellers stecken konnte.)

ADRESSEN UND ZUSTÄNDE

Doch kommen wir zurück zum Wichtigsten, was auf der Modellbahn ferngesteuert wird, den Lokomotiven, Weichen und Signalen. Neben diesen

gibt es natürlich noch fast unbegrenzt viele andere Dinge, von Drehscheiben über Haus- und Fahrzeugbeleuchtungen bis hin zu Riesenrädern, die man fernsteuern kann oder könnte.

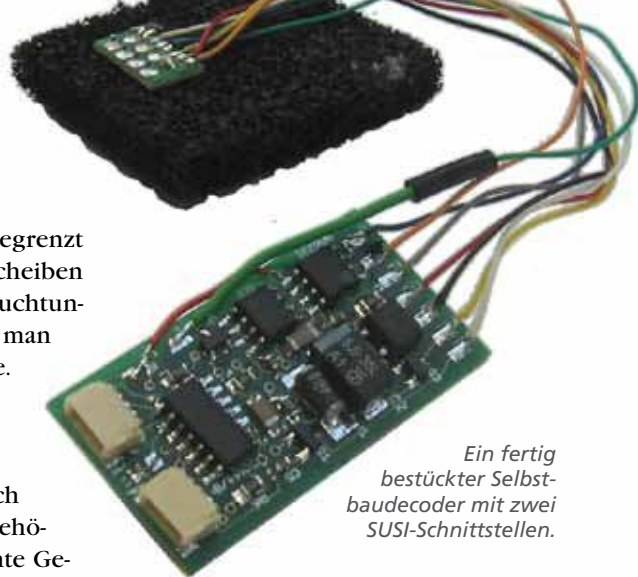
Alle diese zu steuernden Einheiten auf der Modellbahn haben jeweils eine (feste) Adresse und einen Zustand, der sich ändern kann. Zum Zustand gehören zum Beispiel die gewünschte Geschwindigkeit, die Fahrtrichtung und der Lichtzustand einer Lok.

Die Grundstruktur der meisten Digitalsteuerungen ist identisch: Jedem zu steuernden Gerät (Lok, Weiche etc.) wird eine Adresse zugewiesen und für jede dieser Adressen wird ein digitaler Zustand über das Gleis übertragen. Ausserdem wird der Strom für die Motoren der Loks und für sonstige Verbraucher über das Gleis übertragen. Über das Gleis werden also Daten UND Energie geleitet.

Wie man aus der Tabelle der Gemeinsamkeiten und Unterschiede der Systeme ersehen kann, sind nicht nur die beiden Grundprinzipien

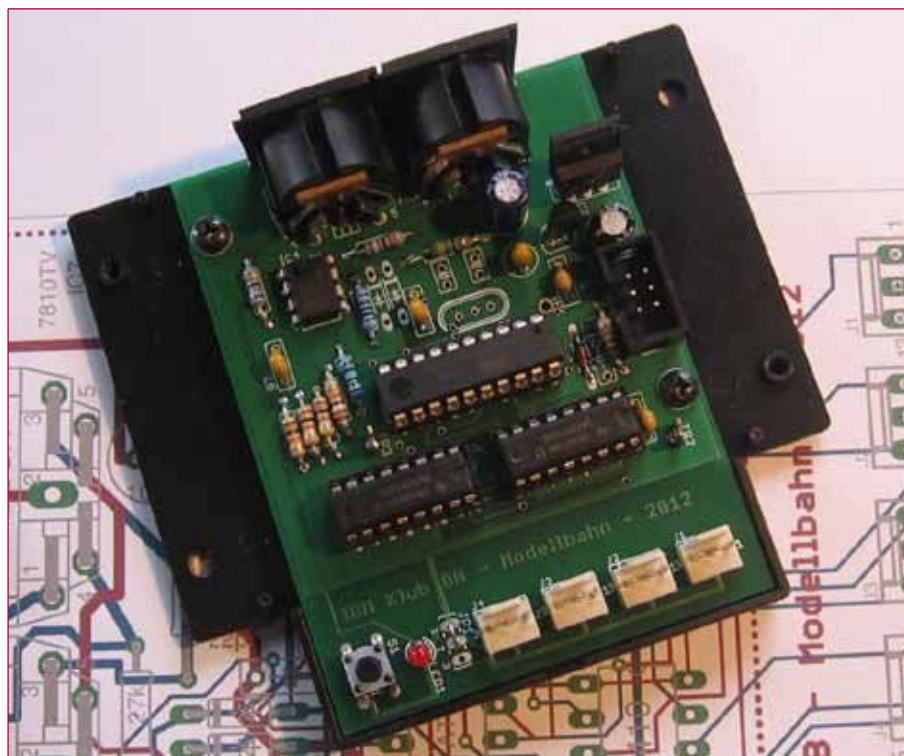
- jedes Gerät hat eine Adresse und bekommt spezifische Daten

Wie ein solcher Weichen- und Signaldecoder selbst gebaut wird, ist u.a. Thema der Artikelserie. Die gezeigte Ausführung ist für das SX-System vorgesehen. Wer die „Decoder selbst bauen“-Serie mitverfolgt, wird jedoch auch in der Lage sein, Anpassungen für DCC und MM zu machen.



Ein fertig bestückter Selbstbaudecoder mit zwei SUSI-Schnittstellen.

Basis sind solche Platinen. Die Bauteile sind noch von Hand lötbar, die einseitige Bestückung macht die Verarbeitung leichter.



SYSTEM	GLEISSIGNAL	ÜBERTRAGUNG	FREQUENZ	ANZAHL FS
DCC	baugrößen-abhängig; genormt von 14 – 22 V; meist einstellbar	Pakete, asynchron	5 – 10 kHz	14, 27, 28, 126
MM	typisch +/- 22 V	Pakete, asynchron	5 – 10 kHz	14, 27
SX	genormt; +20 V/0 V/-20 V	Fester Rahmen, synchron	20 kHz	31

- es werden sowohl Daten wie auch Energie in einem übertragen ziemlich identisch, sondern auch einige andere Parameter. Ein Unterschied ist die Verbreitung der verschiedenen Systeme. In der USA hat sich DCC komplett durchgesetzt, es gibt dort fast nichts anderes mehr. In Deutschland haben Märklin-Systeme eine hohe Verbreitung und es gibt darüberhinaus viele Selectrix-Anlagen, gerade bei den mit Computer gesteuerten. Warum das so ist, werden wir noch sehen.

GRUNDLEGENDER AUFBAU EINES GLEISSIGNALS

DCC, ursprünglich eine Erfindung der Firma Lenz aus den '80ern (die ersten Decoder gab es 1989 für Arnold-N, später auch für Märklin-Gleichstrom), wurde von der NMRA (Nordamerikanische Modellbahnerorganisation) etwas erweitert und 1993 zur Norm erklärt. Die Standards S-9.1 (elektrisches Signal) und S-9.2 (Kommunikation) der NRMA beschreiben das Signal auf dem Gleis. Es besteht aus kurzen Pulsen: 58 µsec die eine Polarität, dann 58 µsec die andere Polarität, siehe Grafik. Eine kurze Pulsfolge ist eine logische „1“, eine lange (118 µsec +, dann 118 µsec -) ist als logische „0“ definiert. Ein Synchronisationssignal (um im Datenstrom erst mal einen „Anfang“ zu finden) besteht aus mindestens 10 „1-bits“.

Bei DCC haben die einzelnen Datenpakete unterschiedliche Längen. Dadurch ist das Protokoll sehr leicht erweiterbar. Es natürlich darf nirgendwo eine Folge mit zehn „1-bits“ in den Nutzdaten enthalten sein und auch sonst müssen die Kommandos unterscheidbar sein.

Ursprünglich gab es die Lokadressen „1“ bis „111“, diese wurden später

auf vierstellige Zahlen erweitert, was gut mit den meist vierstelligen Vorbild-Loknummern in den USA harmoniert. Details finden sich in der Norm. In den den NMRA-Festlegungen entsprechenden europäischen Normen finden sich die zugehörigen Festlegungen in den NEM 670 und 671. Hier wird ein Beispiel-Datenpaket gezeigt, jeweils bestehend aus Synchronisation (10 x „1“) und drei Datenbytes (1 Byte = 8 Bit). Die einzelnen Bytes werden dabei durch ein „Paket-Startbit“ („0“) getrennt.

Das in der Tabelle aufgeführte Beispiel zeigt, wie an die Lok mit der Adresse 55 die Geschwindigkeit „11“ und die Fahrtrichtung „vorwärts“ übertragen wird. In den Datenpaketen kommt das höchstwertige Bit zuerst, die Wertigkeit der einzelnen Bits ist also 128, 64, 32, 16, 8, 4, 2 und 1.

Die einzelnen Datenpakete werden so häufig wiederholt, wie es möglich ist. Die Gesamtzahl der zu übertragenden Datenpakete hängt ab von der Zahl der angesteuerten Loks und Funktionsdecoder ab, daher ist die Wiederholungszeit für ein Paket nicht festgelegt. Die Norm verlangt eine Wiederholung mindestens alle 30 Millisekunden.

Jede DCC-Zentrale verwendet einen eigenen Algorithmus für die Anordnung bzw. Wiederholreihenfolge der einzelnen Datenpakete. Ziel ist es dabei, das Ansprechverhalten der Triebfahrzeuge zu optimieren. Es ist z.B. wichtiger, dass eine Lok schnell auf eine Veränderung am Fahrregler reagiert, als dass z.B. das Licht in einem Funktionsdecoder zeitgenau geschaltet wird.

SELECTRIX

Bei Selectrix (oft auch einfach „SX“ genannt) ist die Zahl der möglichen

Adressen grundsätzlich auf 0 bis 111 beschränkt. Diese Adressen werden als 16 „Päckchen“ mit jeweils 7 Kanälen gesendet ($7 \cdot 16 = 112$), so dass mit jedem Sendezyklus alle Adressen angesprochen werden.

Der SX-Bus (siehe auch Dimo 4/2012 Seite 50) überträgt das Taktsignal („T0“ genannt), das Datensignal („T1“, gesendet von der Zentrale) und das „Rücksignal“, welches zB die Handregler auf den Bus legen. Der Takt hat eine feste Frequenz von 20 kHz. Das heisst, alle 50 µsec wird ein neues Datenbit auf den Bus gelegt, die Übertragungsrate ist etwas höher als bei DCC.

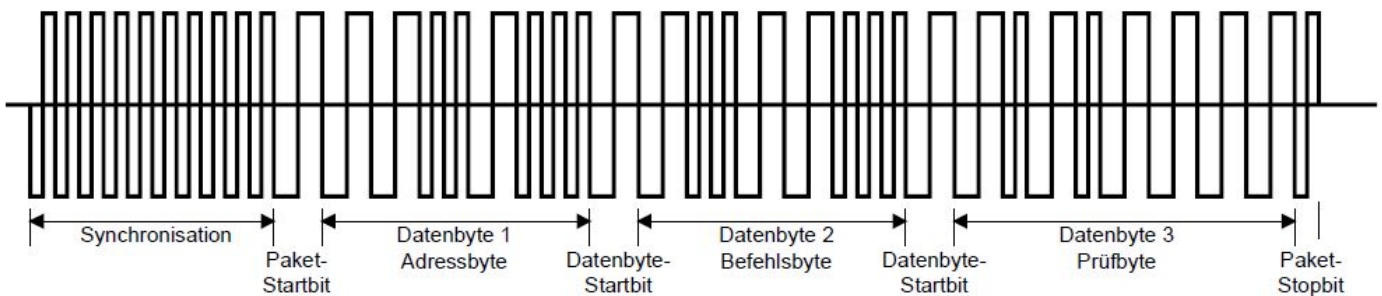
Auch auf dem Gleis ist der Takt erkennbar. Im Gegensatz zu DCC wird jedes Bit mit einem 10 µsec langen spannungslosen Zeitraum begonnen. Erst danach wird für 40 µsec auf die positive oder negative Gleisspannung geschaltet. Es gibt also drei mögliche Zustände auf dem Gleis. Dies ist auch die Erklärung dafür, dass sich reine DCC-Booster nicht für Selectrix eignen. Typischerweise beträgt die Gleisspannung 20 V.

Eine Änderung von +20 V nach -20 V oder umgekehrt (jeweils mit Takt-Zwischenschritt 0 V) bedeutet dabei eine logische „1“. Bei einer logischen „0“ im Datenstrom wird nach dem kurzen Taktimpuls mit 0 V wieder die gleiche Spannung ans Gleis gelegt, wie vor dem Taktimpuls.

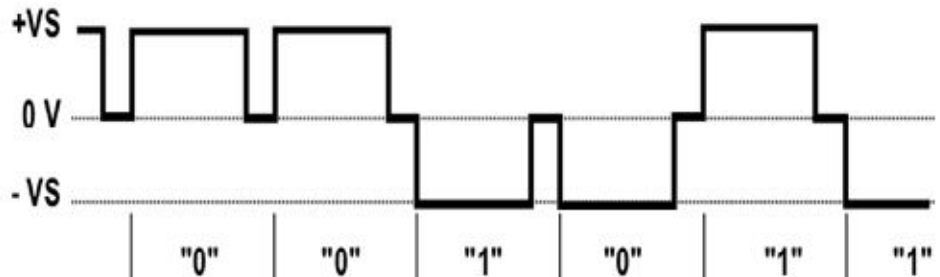
Bei Selectrix werden nach jedem zweiten Datenbit logische „Einsen“ in die Signalfolge eingeschoben, um das Synchronsignal (das hier aus der Signalfolge „0 0 0 1“ besteht) eindeutig erkennen zu können. Nach dem Synchronsignal wird eine von 16 „Gruppenadressen“ übertragen. Es folgen sieben Datensignal-Pakete für diese Gruppe, darauf wieder ein Synchron- und Adressbyte. Dies ergibt $7 \cdot 16 = 112$ Kanäle.

BYTE	DATEN	AUSWERTUNG	BEDEUTUNG
1	00110111	= 32+16+4+2+1=55	0 + (7 bit) Adresse
2	01100111 (im Format 01DCSSSS)	01 = Command 01 ..1 ...0 0111	Lokgeschwindigkeit 28 Fahrstufen Fahrrichtung vorwärts bedeutet nach Tabelle 1 in NEM 671 Geschwindigkeit 11 von 28 (dies muss einer Tabelle entnommen werden – die Zahl ist wegen der Rückwärtskompatibilität von DCC zu früheren Version hier nicht so leicht zu erkennen aus dem Binär- wert)
3	01010000	XOR von Byte 1 und Byte 2	Prüfbyte, das der Fehlererkennung dient - erkennt der Decoder, dass das Prüf- byte für das von ihm empfangene Paket nicht stimmt, so verwirft er das ganze Paket.

Tabellen- und Signaldarstellung eines DCC-Basis-Datenpakets mit drei Datenbytes (Adresse, Befehl, Prüfsumme), kodiert für Adresse 55 und Vorwärtsfahrt mit Fahrstufe 11.



Die SX-Bitdarstellung zeigt den typischen Spannungsverlauf mit dem „Absatz“ oder „Zwischenschritt“ bei 0 V. Diese spannungslosen Zwischenschritte sind der Grund, warum reine DCC-Booster nicht für SX eingesetzt werden können.



Signaldarstellungen aus NEM 671 u. 680

Antwortzeit	Feste Antwortzeit: Nach maximal 80 ms ist jedes Kommando übertragen.	Variable Antwortzeit: Die Zeit, die die Übertragung eines Kommandos braucht, hängt davon ab, wieviele andere Signale gesendet werden.
Erweiterbarkeit	Nicht leicht zu erweitern, beschränkt auf 112 Kanäle	Leicht zu erweitern
Fester Takt	Synchron	Asynchron
Hersteller	Einige Kleinere	Viele Große, viele Kleinere

Der Vergleich SX–DCC zeigt, dass jedes System seine Vor- und Nachteile hat.

signal in den NEM genormt: NEM 680 beschreibt die Bitdarstellung des digitalen Steuersignals von SX, NEM 681 die Datenpakete. Da die Fahrgeschwindigkeit der Modelle bei SX von Beginn an in 31 Fahrstufen aufgelöst, jedoch nur zwei Schaltfunktionen vorgesehen sind, ist der Aufbau eines Steuerungsdatenbytes recht einfach. Es setzt sich aus den acht Bits S0, S1, S2, S3, S4, D, L, F zusammen. Dabei bedeuten S = Speed (Geschwindigkeit), D = Direction (Fahrtrichtung), L = Light (Licht) und F = Function (Funktion).

MM: MÄRKLIN-MOTOROLA

Neben DCC und SX gibt es das sog. Märklin-Motorola-Protokoll, das durch die Mittelleiterfahrer eine große Marktbedeutung bekam. Vor einigen Jahren ließ Märklin dann als Ablösung für MM ein neues moderneres Digitalprotokoll für die Modellbahn entwickeln: mfx. Seither werden alle Märklin-Loks mit Decodern für dieses Format ausgelie-

fert. Weder das ältere MM noch mfx wurden von Märklin offengelegt oder einer Normung unterworfen. Nichtsdestotrotz sind Details beider Protokolle bekannt geworden und können in verschiedenen Publikationen nachgelesen werden. Da wir uns hier auf DCC und SX konzentrieren wollen, verzichten wir auf die Darstellung der entsprechenden Gleissignale.

VOR- UND NACHTEILE DER EINZELNEN PROTOKOLLE

Hierüber wurden in der Vergangenheit fast schon „ideologische Grabenkämpfe“ geführt. Fangen wir deshalb mit einer grundsätzlichen Vorbemerkung an: „Mit allen Systemen kann man Modelllokomotiven fahren.“ Das ist doch schon mal was!

Niemand wird zu einem Systemwechsel gezwungen, wenn er erweiterte Funktionalitäten haben will. Durch diverse Erweiterungen der ursprünglichen Protokolle sind alle Systeme inzwischen in der Lage, fast beliebig viele

Loks zu steuern, mit und ohne Sound, mit und ohne Computer.

Wir beschränken uns hier auf die im „Zweileiterbereich“ vertretenen Protokolle, also einen Vergleich zwischen DCC und SX. Ein grundlegender Unterschied ist die feste Anzahl von übertragenen Datenpaketen je Zeiteinheit (112) bei SX, während die Paketanzahl bei DCC von der Zahl der Lokomotiven und anderen Faktoren abhängt.

SX kann man daher als synchrones System (d.h. mit festem Daten-Takt) bezeichnen, während DCC ein asynchrones System ist (Daten werden nach Bedarf gesendet). Die Vor- und Nachteil-Tabelle fasst dies noch einmal zusammen.

Auch verschiedene aus SX heraus entstandene Erweiterungen (z.B. RMX von Rautenhaus als DCC-SX-Kombisystem) bleiben im SX-Bereich grundsätzlich beim typischen SX-Timing. Allerdings werden dabei die Kanäle gemultiplexed, so dass durch mehr Kombinationsmöglichkeiten weitere Informationen übertragen werden.

Im nächsten Teil des Artikels werden wir uns detaillierter mit dem Selectrix-System beschäftigen und es selbst decodieren mit Hilfe eines Arduinos (www.arduino.cc), der in C programmiert wird.

Michael Blank