Die Verwendung der

Rail control library

mit

NewDeal Office / **Breadbox Ensemble** unter **R-BASIC**

Version: 08.11.2018

Ulrich Cordes
Auf der Langenbach 14
D-34317 Habichtswald

Email: ulrich.cordes@gmx.de
Web: www.ulrich-cordes.de

Editorial

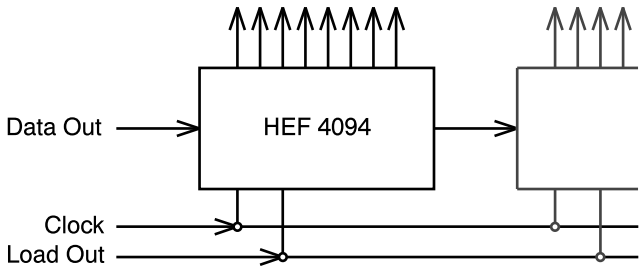
In den 1980er Jahren bot fischertechnik Baukästen an, die darauf abgestimmt waren, computersteuerbare Modelle aufzubauen. Als Zubehör gab es ein Interface, welches am Druckerport der damaligen Heimcomputer angeschlossen werden konnte. Die Technik dieses Interfaces war so einfach und zugleich genial, dass ich diese Technik auch für meine R-BASIC Rail Control Module verwende und dafür eine R-BASIC Library (eine Art Treiber) programmiert habe.

Die Technik des Interfaces

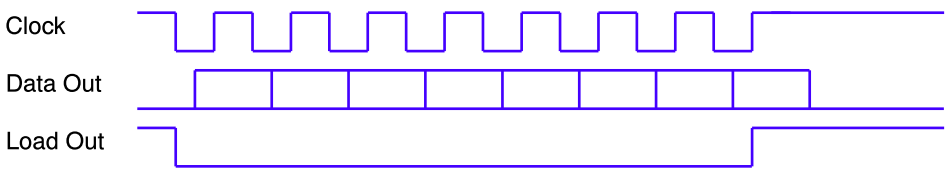
Die damaligen Computer hatten fast alle mindestens einen Anschluss für einen Drucker. Dieser Anschluss bietet jedoch nicht ausreichend Bit-Breite, um damit direkt komplexe Modelle steuern zu können. Aus diesem Grund hat sich fischertechnik bei der Entwicklung des Interfaces für ein serielles Konzept entschieden, .

Digitale Ausgänge

Zum schalten von acht Ausgängen benötigt man nur drei Bits des Druckeranschlusses sowie einen CMOS-IC vom Typ HEF 4094. Dieses IC enthält 8 Ausgangsports, die Daten werden aber seriell (bitweise nacheinander) in das IC „geschoben“.



Wird das **Load Out**- Signal auf low gelegt, übernimmt das IC mit jedem **Clock**-Impuls Daten (0 oder 1) und reicht diese im internen Speicher mit jedem weiteren **Clock**-Impuls eine Position weiter. Sind auf diese Weise alle Daten übertragen worden, kann das **Load Out**-Signal wieder auf high gelegt werden und erst dann liegt auch an den acht Ausgängen des ICs die Information an, die wir zuvor übertragen haben.

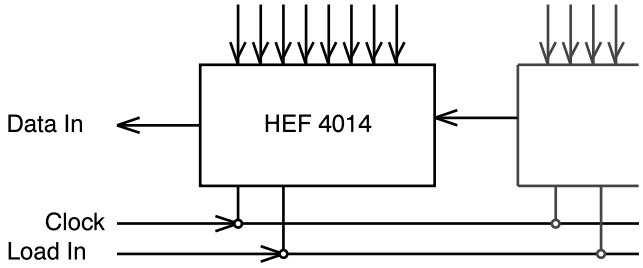


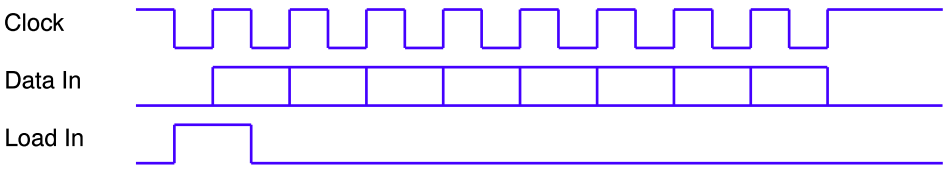
Damit man sich als R-BASIC-Programmierer aber nicht darum zu kümmern braucht, Bits einzeln in ein IC zu schieben, gibt es die **Rail control library**, die als eine Art Treiber funktioniert und diese Bitschiebereien übernimmt.

Um mehr wie acht Ausgangsports zu erhalten, können einfach mehrere ICs vom Typ HEF 4094 hintereinander geschaltet, also kaskadiert werden. Die Anzahl der Ports wird eigentlich nur die Treiber-Software, in diesem Fall durch die R-BASIC-Library begrenzt. Theoretisch ist es möglich, mehrere tausend Ports auf diese Weise zu bedienen. Diese R-BASIC-Library ist auf maximal 256 Ports begrenzt.

Digitale Eingänge

Das Einlesen von Sensor-Informationen funktioniert nach dem selben Prinzip. Mit einem anderen IC vom Typ HEF 4014, dem **Clock**-Signal sowie den beiden zusätzlichen Signalen **Load In** und **Data In** können Informationen eingelesen werden.





Mit dem **Load In**-Impuls werden die außen am IC anliegenden Stati in ein internes Schieberegister übernommen, mit dem **Clock**- und **Data In**-Signal werden die Daten aus dem Schieberegister eingelesen.

Auch hier übernimmt die Library die Aufgabe einzelne Bits zu verschieben und bietet hierfür dem Programmierer einen sehr komfortabelen BASIC-Befehl an. Um mehr wie 8 Signale zu erfassen kann auch der 4014 kaskadiert werden.

Systemvorraussetzungen

- PC mit DOS-Betriebssystem und paralleler Druckerschnittstelle
- GEOS 2.x, NewDeal Office oder Breadbox-Ensemble
- R-BASIC 0.7 oder höher

Wichtig: Diese R-BASIC-Library kann nur auf Rechnern eingesetzt werden, auf denen GEOS unter purem DOS läuft. Überlagerte Windows- bzw. Linux-Betriebssysteme untersagen direkte IO-Zugriffe auf den Druckerport. Solche IO-Zugriffe benötigt die Library jedoch zwingend.

4 - Rail control library

Die Hardware

Die Hardware ist in foolgende einzelne Module aufgeteilt:

1. Basis-Modul
2. Relais-Modul
3. Transformator-Modul
4. Modul für digitale Inputs

Basismodul

Verwenden der R-BASIC-Library

Der erste Schritt ist, dem R-BASIC klar zu machen, dass die **Rail control library** benutzt werden soll. Dies geschieht mit folgendem Befehl ganz am Anfang eines R-BASIC-Programms:

```
INCLUDE "RailControlLibrary"
```

Die Library enthält eine Initialisierungsroutine, mit der die verwendete IO-Basis-Adresse des Druckerports sowie die Anzahl der Eingangs- und Ausgangsports bekannt gemacht wird.

```
RailInit adr AS INTEGER, out_ports AS INTEGER, inp_ports AS INTEGER
RailInit &h378, 32, 8
```

Hiermit wird der Library die bei PCs übliche Druckerport-Adresse &h378 bekannt gemacht. Die Library kann maximal 256 Eingangs- und 256 Ausgangsports verwenden. Wir geben hier an, dass wir zum Beispiel 32 Ausgangsports und 8 Eingangsports verwenden wollen.
Das Schieben der Bits durch die ICs kann etwas Zeit beanspruchen. Daher ist es vielleicht nicht sinnvoll, mit jedem `RailSetOutport` immer auch sofort alle Bits durch die ICs zu schieben.

Mit den folgenden Konstanten kann der Basic-Code „lesbar“ gestaltet werden. Ich verwende in diesem Dokument hauptsächlich diese Konstanten.

```
CONST REL_OFF     0 ' Aus
CONST REL_ON      1 ' Ein
CONST FOREWARD    1 ' Vorwärts
CONST BACKWARD    2 ' Rückwärts
```

Relais

Zum Ansteuern von Relais besitzt die **Rail control library** einen recht einfach aufgebauten Befehl.

```
RailSetOutport portnum AS INTEGER, value AS INTEGER
```

Beispiele:

```
RailSetOutport 3, 1
RailSetOutport 4, REL_OFF

CONST rot = 0
CONST gruen = 1
CONST signal = 13
RailSetOutport signal, gruen
```

Zum Schalten von Weichen oder Form-Signalen werden ebenfalls Relais verwendet. Im Gegensatz zu Licht-Signalen oder Beleuchtungen sollten die Antriebe der Weichen nicht dauerhaft mit Strom versorgt werden. Daher müssen diese Antribe nach ca. einer Sekunde wieder abgeschaltet werden.

```
CONST weiche_1_links = 5
DELAY 60
RailSetOutport weiche_1_links, REL_ON
DELAY
RailSetOutport weiche_1_links, REL_OFF
```

Hinweis: Wird auch das Basis-Modul verwendet, welches eine Sicherheitsabschaltung besitzt, wird das letzte Programm nicht funktionieren. Die Sicherheitsabschaltung überprüft das CLK-Signal. Wird dieses für ca. eine halbe Sekunde **nicht** angesteuert, geht die Sicherheitsabschaltung davon aus, dass das R-BASIC-Programm nicht mehr funktioniert oder die Module nicht mehr mit dem Druckerport verbunden sind. In dem Fall werden alle Ausgänge ausgeschaltet um Zugunfälle oder ein Überhitzen von Magnetspulen zu verhindern. Das bedeutet, es sollte auf die Verwendung des **DELAY**-Befehls verzichtet werden.

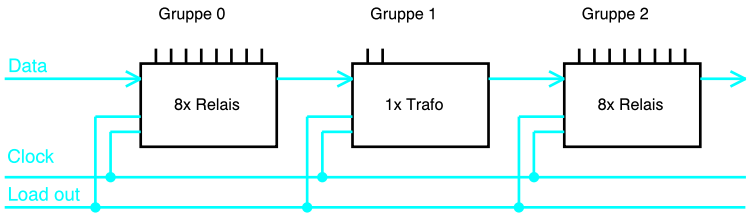
Ein Besseres Beispiel:

```
SUB SchalteWeiche (weiche AS INTEGER)
      DIM t AS DWORD
      t = SysGetCount() + 60
      RailSetOutport weiche, REL_ON
      ' Eine Sekunde warten
      WHILE SysGetCount()<t
            RailUpdate
      WEND
      RailSetOutport weiche, REL_OFF
END SUB
```

Trafo

Jeder kennt ihn, den Transformator auf einer Modellbahnanlage. Ein anderer Begriff hierfür ist der Fahrtregler. Hiermit wird die Fahrgeschwindigkeit auf der Modellbahnanlage eingestellt.

Die R-BASIC Rail control library kann ebenfalls Fahrtregler verwenden. Allerdings benötigt ein Fahrtregler insgesamt 8 Bit Breite, das heißt also ein einzelner Basutein HEF 4094 bedient einen Fahrtregler.



Folgender Befehl stellt eine Fahrstufe ein:

```
RailSetSpeed group AS INTEGER, value AS INTEGER, direction AS INTEGER
```

Dabei entspricht **group** der Nummer der 8er-Gruppe wie im letzten Bild dargestellt.
Ist der Wert **value** 0, dann wird keine Fahrspannung ausgegeben. Ein Wert von 1...64 gibt in Stufen eine Spannung 4.5 bis 13,5V aus.
Die Variable **direction** kann die Werte 0 für Aus, 1 für Vorwärtsfahrt und 2 für Rückwärtsfahr annehmen.

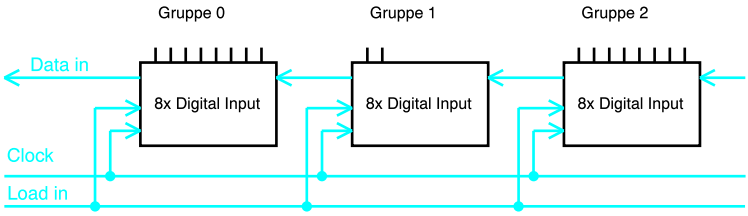
Beispiele:

```
RailSetSpeed 1, 35, 1

CONST TRANSFORMER_1 = 1
CONST TRANSFORMER_2 = 4
RailSetSpeed TRANSFORMER_2, 22, BACKWARD
RailSetSpeed TRANSFORMER_1, 0, REL_OFF
```

Sensoren

Zur Steuerung einer Modellbahn muss R-BASIC auch Kenntnis zur aktuellen Position von einzelnen Zügen haben. Mindestens müssen an End- bzw. Halte-Punkten Sensoren ausgewertet werden. Hier bieten sich zum Beispiel IR-Lichtschranken oder elektronische Lösungen an.



Um einen Sensor einzulesen gibt es den Befehl:

```
RailGetInport (portnum AS INTEGER) AS INTEGER
```

```
DIM v AS INTEGER
v = RailGetInport (20) ' 5. Eingangssignal des 3. Digital-Input-Moduls
```

Der Parameter *portnum* kann einen Wert zwischen 0 und 255 sein, darf aber die Anzahl eingestellten Ausgangsports nicht überschreiten. Der Rückgabewert ist dann entweder 0 oder 1.

Allgemein

Ein paar allgemeine Befehle mit denen sich die Hardware zurücksetzen lässt bzw. man alle Ausgaben für eine Pause anhalten und wieder weiter laufen lassen kann.

```
RailPause       ' Merkt sich den Zustand aller Ausgaben und schaltet diese dann aus
RailContinue    ' Läßt alles wieder weiter laufen
RailReset       ' Setzt alle Ausgaben komplett zurück, z.B. bei Programm-Ende
```

Fehler

„Worum fungdsionierd'n de Schaise hier nich?“ Die Vielfalt möglicher Fehler kennt keine Grenzen. Dabei sind Fehler wie der vergessenen Stecker am Druckerport oder eine fehlende Spannungsversorgung Rail Control Module eher noch leicht zu finden. Schwieriger wird es mit Programmierfehlern, aber es ist noch kein Meister vom Himmel gefallen.