

## RaspberryPi Projekt: Fahrstraßen Steuerung für Märklin Spur 1 Anlage

Auf meiner Märklin Spur 1 Anlage setze ich Weichenantriebe Art.Nr. 5625 von Märklin ein. Die Antriebe werden einzeln über zwei Stellpulte angesteuert. Bei dieser Art der Steuerung muss, um eine bestimmte Fahrstraße wie z.B. ein Abstellgleis zu erreichen, eine entsprechende Kombination von Einzeleinstellungen vorgenommen werden.

Viel einfacher wäre es, wenn das Ziel ausgewählt werden könnte und eine Logik die entsprechende Einstellung der Weichen vornehmen würde. Diese Art der Ansteuerung nennt man Fahrstraßensteuerung.

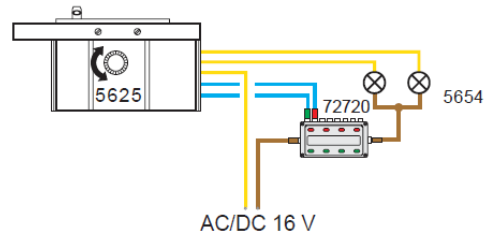


Abbildung 1 Weichenantrieb Ansteuerung über Stellpult

Für eine analog gesteuerte Anlage, wäre dazu eine Diodenmatrix ein geeignetes Mittel. Die Diodenmatrix ist eine altbewährte, weit verbreitete Technik für die Fahrstraßensteuerung eines Bahnhofs. Sie eignet sich gut für kleine bis mittlere Bahnhöfe. Sie ermöglicht es, mehrere Weichen zu einer Fahrstraße zusammenzufassen und mit einem einzigen Tastendruck in die gewünschte Lage zu steuern. Dadurch ergibt sich gegenüber der Einzelsteuerung eine große Verbesserung im Bedienungskomfort und in der Sicherheit.

Die Diodenmatrix hat den Nachteil, dass sie einmal gelötet nicht mehr geändert werden kann. Die Ansteuerung wird fest verdrahtet. Dazu kommt, dass ich Informatiker bin und kein E-Techniker. Mir liegt die Idee näher, in dieser analogen Anlage Computer Logik einzusetzen.

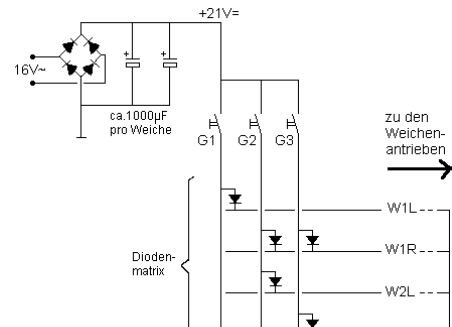


Abbildung 2 Beispiel für Diodenmatrix

Mit dem RaspberryPi steht dafür eine Plattform zur Verfügung, die wenn auch deutlich teurer als eine Schaltung mit elektronischen Bauelementen, immer noch erschwinglich ist.

Eine Programmsteuerung über den RaspberryPi löst auch noch ein anderes elektrisches Problem, das sich im Zusammenhang mit einer Fahrstraßensteuerung stellt.

Ein großes Problem der Diodenmatrix ist, dass alle Weichen einer Fahrstraße gleichzeitig angesteuert werden. Dadurch ist ein Stromstoß erforderlich, der die meisten Stromversorgungen in die Knie zwingt. Das bedeutet, die Weichen schalten infolge zu wenig Strom nicht richtig. Auch dafür gibt es Lösungen, z.B. indem man mit einem Brückengleichrichter den Wechselstrom des Trafos gleichrichtet und glättet und dabei die Spannung und den Strom erhöht. Allerdings müssen die Kabel und Taster so dimensioniert sein, dass sie den Summenstrom verkraften.

Angeblich zieht der Märklin Doppelspulenantrieb 0,8-1A. Da in meinem Gleisstellbild im ungünstigsten Fall bis zu 5 Weichenantriebe gestellt werden müssen, würde der Trafo mit einer maximalen Leistung von 2A nicht genügend Strom liefern können. Auch hier gibt es bewährte elektronische Methoden. Man kann versuchen Stromstöße zu verzögern oder die Weichenantriebe über einen Schaltverstärker ansteuern, bei dem der Strom aus einem vorher aufgeladenen Kondensator entnommen wird.

Dies Problem lässt sich mit einer Programmsteuerung ganz einfach lösen, indem eine Verzögerung zwischen den einzelnen Schaltvorgängen programmiert wird.

## Das Gleisstellbild

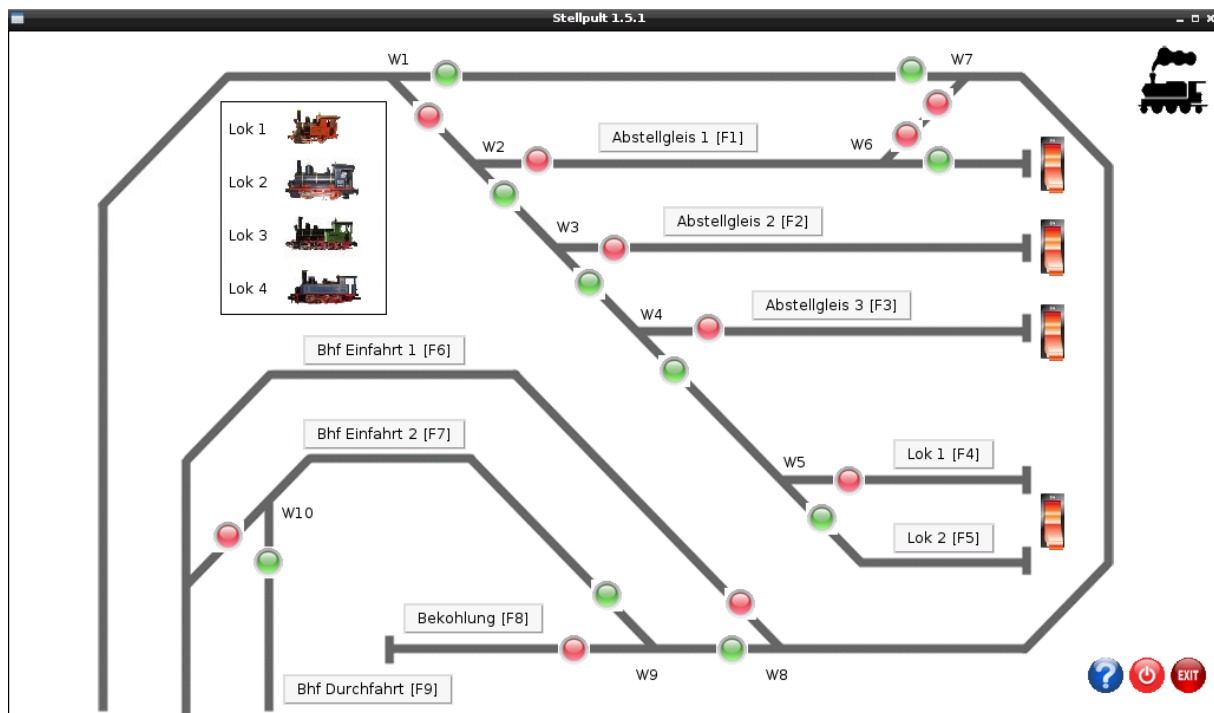


Abbildung 3 Gleisstellbild

Wie Abbildung 3 Gleisstellbild zeigt, gibt es 10 Weichen (W1-W10) die zu stellen sind. Für jede Weiche müssen zwei Leitungen geschaltet werden (Abbildung 1). Weiterhin enthält das Gleisstellbild für jede Weiche zwei Taster mit denen für jede Weiche die Richtung geschaltet werden kann. Die Farbe des Schalters (Grün oder Rot) gibt den aktuellen Schaltzustand an.

Im Programm werden die Taster entweder durch Maus Klicks oder durch Tastatur Eingaben (z.B. w5r stellt Weiche 5 auf rund). Für jedes Abstellgleis und den Lokschuppen gibt es einen Ein/Aus Schalter, um die Gleise stromlos zu machen.

Weg (gerade=0, rund=1, n=neutral)	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
Abstellgleis 1	1	1	n	n	n	0	n	n	n	n
Abstellgleis 2	1	0	1	n	n	n	n	n	n	n
Abstellgleis 3	1	0	0	1	n	n	n	n	n	n
Lokschuppen 1	1	0	0	0	1	N	n	n	n	n
Lokschuppen 2	1	0	0	0	0	n	n	n	n	n
Bhf Einfahrt 1	0	n	n	n	n	n	n	1	n	n
Bhf Einfahrt 2	0	n	n	n	n	n	n	0	1	0
Bhf Durchfahrt	0	n	n	n	n	n	n	0	1	1
Bekohlung	0	n	N	n	n	n	n	0	0	n

Tabelle 1 Weichenschaltung im Gleisstellbild

Tabelle 1 zeigt wie die Weichen im Gleisstellbilde geschaltet werden müssen, um die Weichen für die jeweilige Fahrstraße zu stellen. „0“ steht dabei für geradeaus und „1“ für rund, also die Abbiegung. „n“ bedeutet neutral, die Weiche muss für diesen Weg nicht geschaltet werden.

Für die Einfahrt in den Bahnhof gibt es z.B. zwei Fahrstraßen. In der Variante 1, der äußeren Umfahrung der Bekohlungsanlage wird Weiche 8 auf rund gestellt. In der Variante 2, der inneren Umfahrung, wird Weiche 8 auf geradeaus, Weiche 9 auf rund und Weiche 10 auf geradeaus gestellt.

Weiche 7 spielt hier keine Rolle, da sie als einzige Weiche entgegen der Fahrtrichtung geschaltet wird und nur bei Rückwärtsfahrt geschaltet werden muss.

## Weichenansteuerung mithilfe eines RaspberryPi (RPI)

Der Vorteil des RPi sind seine GPIO Pins. Die aktuell neueste Version, das Modell B+ verfügt über 40 „General Purpose Input/Output“ Pins, kurz GPIOs genannt.

Wie Tabelle 2 zeigt, sind einige der GPIO Pins frei programmierbar, während andere eine definierte Spannung abgeben. Die programmierbaren Pins können entweder für Output konfiguriert werden und liefern dann entweder 0V oder 5V oder als Input Pins, bei denen abgefragt werden kann, ob eine Spannung anliegt. Es gibt insgesamt 26 programmierbare Pins (GPIOxx). Das ist mehr als genug, um je zwei Kontakte von 10 Weichen zu stellen, genügt aber nicht um weitere 2x10 Taster abzufragen. Der Befehl zum Stellen der Weichen kann also nicht über Taster erfolgen, wie sie in analogen Schaltungen üblich sind.

Es gibt noch weitere Einschränkungen, die zu berücksichtigen sind. Der RPi liefert an einem GPIO Kontakt maximal 3,3V. Die Weichenantriebe benötigen aber ca. 16V, einmal abgesehen davon, dass der RPi, der mit 1,2 A betrieben wird, die für den Schaltvorgang eines einzelnen Antriebs erforderlichen 1A nicht liefern kann. Die Weichenantriebe müssen also über Relais angesteuert werden.

Weiterhin ist zu beachten, dass die über Magnetspulen angetriebenen Weichenantriebe keinen konstanten Strom, sondern einen Stromstoß erfordern. Das Relais kann also nach dem Schalten sofort wieder zurückgestellt werden.

## Der Weichenantrieb

Ich setze Märklin Doppelspulen Weichenantriebe mit Endabschaltung ein (Märklin Art. Nr. 5625). Dieser Antrieb kann durch Taster an Stellpullten geschaltet werden. Es ist also nur ein kurzer Stromimpuls erforderlich um die Weiche wieder abzuschalten. Die Eingänge 1 und 2 steuern die Weiche auf gerade oder rund, an L liegt die Masse an und die Eingänge 3 und 4 sind für Rückmeldesignale vorgesehen. Zur Steuerung eines Weichenantriebs sind also 2 Relais erforderlich.

## Hinweise zur Verwendung des Relais

Das von mir verwendete Relais (SainSmart 8 Channel DC 5V Relay Module) zieht an, wenn der Schalteingang (IN1 oder IN2) gegen Masse gezogen wird (GPIO-Pin am RPi auf 0). Da die Steuer- und Versorgungsspannung über Optokoppler galvanisch getrennt ist, ist nicht zu befürchten, dass z.B. durch die Spulen

### Raspberry Pi B+

Pin Nummer		
3.3V	1	2 5V
GPIO2	3	4 5V
GPIO3	5	6 GND
GPIO4	7	8 GPIO14
GND	9	10 GPIO15
GPIO17	11	12 GPIO18
GPIO27	13	14 GND
GPIO22	15	16 GPIO23
3.3V	17	18 GPIO24
GPIO10	19	20 GND
GPIO9	21	22 GPIO25
GPIO11	23	24 GPIO8
GND	25	26 GPIO7
DNC	27	28 DNC
GPIO5	29	30 GND
GPIO6	31	32 GPIO12
GPIO13	33	34 GND
GPIO19	35	36 GPIO16

Tabelle 2 GPIO Pins des RPi B+

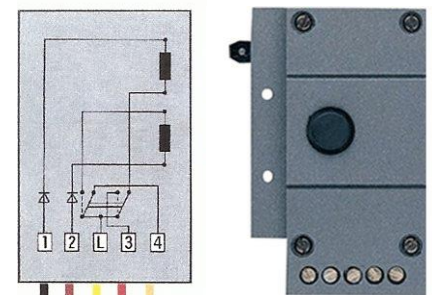


Abbildung 4 Märklin 5625 Weichenantrieb

induzierte Spannungsimpulse den RPi beschädigen. In diesem Fall wäre der Einsatz einer Darlington Schaltung erforderlich gewesen (z.B. durch einen IC UDN2803)

## Umsetzung

Für die Ansteuerung der 10 Weichenantriebe werden 20 Relais benötigt. Die unten abgebildete Relaisplatine gibt es in Ausführungen mit 2, 4, 8 und 16 Relais. Ich habe für die Ansteuerung der Weichenantriebe zwei 8 und eine 4 Kanal Platine eingesetzt. Eine weitere 4 Kanal Platine wird für das Ein/Aus Schalten der Spannung auf den drei Abstellgleisen und dem Lokschuppen verwendet. Die in Europa von SaintSmart vertriebene Platine ist ein Multi-Kanal 5V Relais-Interface-Board. Damit können sehr einfach verschiedene Geräte mit großen Strom / großer Last gesteuert werden. Optokoppler sorgen für eine sichere Trennung der Steuer und Wirkströme. Daher können Mikrocontroller wie der Raspberry Pi direkt angeschlossen werden. Jedes Relais zieht 15-20 mA Steuerstrom. Falls die vom RPi gelieferte Stromstärke einmal nicht ausreichen sollte, können Steuer- und Versorgungsspannung der Relais voneinander getrennt werden, indem der Jumper entfernt wird und an die Pins JC-VCC eine externe 5V Spannungsquelle und an den Pin VCC 3,3V vom RPi angeschlossen werden. Der Pin GND muss mit GND (0V) der Spannungsquelle verbunden werden. In meinem Fall ist das nicht notwendig, da immer nur ein Relais für einen kurzen Impuls angesteuert wird. Mit einem 1200 mA Netzteil war es ohne weiteres möglich 8 Relais gleichzeitig ohne externe Spannungsquelle anzusteuern.



Abbildung 5 8 Kanal 5V Relay Modul

Jedes Relais zieht 15-20 mA Steuerstrom. Falls die vom RPi gelieferte Stromstärke einmal nicht ausreichen sollte, können Steuer- und Versorgungsspannung der Relais voneinander getrennt werden, indem der Jumper entfernt wird und an die Pins JC-VCC eine externe 5V Spannungsquelle und an den Pin VCC 3,3V vom RPi angeschlossen werden. Der Pin GND muss mit GND (0V) der Spannungsquelle verbunden werden. In meinem Fall ist das nicht notwendig, da immer nur ein Relais für einen kurzen Impuls angesteuert wird. Mit einem 1200 mA Netzteil war es ohne weiteres möglich 8 Relais gleichzeitig ohne externe Spannungsquelle anzusteuern.

Ich verwende ein Netzteil mit 5V, 2000 mA. Lastprobleme treten nicht auf, da die Weichenantriebe nur einen Stromimpuls benötigen, der ca. 0,2 Sekunden dauern muss. Im Steuerungsprogramm wird das erreicht, indem der entsprechende GPIO Pin erst auf 0 gezogen wird, dann ein Sleep Timer von 0,2 Sekunden abläuft und dann der Pin wieder auf 1 gesetzt wird. Da dies immer im selben Unterprogramm abläuft, wird immer nur ein Relais geschaltet. Die vier Ein/Aus Schalter stellen kein Problem dar, da der RPi mit dem 2000 mA Netzteil mehr als genug Strom zur Daueransteuerung für vier Relais liefert.

## Verschaltung der RPI GPIOs mit den 8 bzw. 4 Kanal Relais Modul

BCM	Name	Relais
-	+5V	R1-VCC
2	SDA.1	R1-IN1
3	SCL.1	R1-IN2
-	GND	R1-GND
4	GPIO.7	R1-IN3
17	GPIO.0	R1-IN4
27	GPIO.2	R1-IN5
22	GPIO.3	R1-IN6
10	MOSI	R1-IN7
9	MISO	R1-IN8

BCM	Name	Relais
-	+5V	R2-VCC
11	SCLK	R2-IN1
-	GND	R2-GND
5	GPIO.21	R2-IN2
6	GPIO.22	R2-IN3
13	GPIO.23	R2-IN4
19	GPIO.24	R2-IN5
26	GPIO.25	R2-IN6
14	TxD	R2-IN7
15	RxD	R2-IN8

BCM	Name	Relais
-	+5V	R3-VCC
-	GND	R3-GND
18	GPIO.1	R3-IN1
23	GPIO.4	R3-IN2
24	GPIO.5	R3-IN3
25	GPIO.6	R3-IN4
8	GPIO.24	R4-IN1
7	GPIO.26	R4-IN2
12	GPIO.32	R4-IN3
16	GPIO.36	R4-IN4

## Projekt 7" Touchscreen

Um das Gleisstellpult anzeigen zu können, habe ich zunächst ein altes Netbook verwendet, auf das ich ein MintLinux installiert habe. Von dort aus wird über ssh das Display des Stellpults auf das Display des Netbooks übertragen.

Inzwischen gibt es aber Touchscreens, die direkt an den Pi angeschlossen werden können.

Ich habe mir den 7" Touchscreen der Raspberry Pi Foundation und ein Gehäuse gekauft (<https://www.rasppishop.de/Raspberry-Pi-7-Touchscreen-Display>)

Das aktuelle Raspbian Jessie unterstützt den Touchscreen dank Kernel Treiber out of the box. Ich musste das Display nur noch um 180° drehen und dazu in `/boot/config.txt` den folgenden Eintrag vornehmen:

```
lcd_rotate=2
```

Die Variable `display_rotate` dreht zwar auch die Bildschirmausgabe, aber leider nicht den Touchscreen Input.

Da der Bildschirm mit 800x480 eine geringere Auflösung hat, als das Netbook musste ich die Koordinaten der Bildelemente anpassen.

Mithilfe des Tkinter Window Attributes

```
Root.attributes('-fullscreen', 1)
```

wird die volle Bildschirmgröße für die Ausgabe genutzt.

## Projekt Schaltzustand der Weichenantriebe anzeigen

*(noch nicht umgesetzt)*

Bis auf die zwei verbliebenen GPIO Ports BCM 20 und 21, die von mir nicht benutzt werden, sind alle verfügbaren GPIO Ports in Verwendung. Damit verbleibt gegenüber einem analogen Stellpult ein Problem: Die Farben der Taster zeigen die Schaltzustände der Weichen nach der Programmlogik. Wenn eine Weiche einmal manuell umgestellt wird, oder blockiert ist, stimmt die Anzeige des Stellpults nicht mehr mit der Wirklichkeit überein. Um dies zu realisieren müsste der Schaltzustand der Weichen über die Rückmeldekontakte (siehe Abbildung 1) der Weichenantriebe ausgelesen und entsprechend dargestellt werden. Dazu sind noch einmal 2x10 GPIO Pins erforderlich. Auf meinem RPi B+ stehen aber nur noch zwei Pins zur Verfügung.

Nach längerem Suchen habe ich bei der Firma AB Electronics UK ein Erweiterungsboard für den RPi B+ mit der Bezeichnung IO PI Plus gefunden, das auf den RPi aufgesteckt werden kann und diesen dann um weitere 32 GPIOs erweitert. Dieses Board benutzt die I2C Pins für die Ansteuerung des Erweiterungsboards. Alle anderen Kontakte werden durchgeschleift.



## Einrichtung des Raspberry Pi+ für das IO Pi Plus Erweiterungsboard

Als Root:

```
# apt-get update
# apt-get upgrade
# apt-get install python-smbus
# apt-get install i2c-tools
```

In der Datei `/etc/modprobe.d/raspi-blacklist.conf` die beiden Zeilen

```
#blacklist spi-bcm2708
#blacklist i2c-bcm2708
```

Auskommentieren (# am Anfang der Zeile)

In der Datei `/etc/modules` am Ende den folgenden Eintrag anhängen:

```
i2c-dev

# adduser pi i2c
```

Zum Schluss das System neu starten:

```
# reboot
```

Danach kann der RPi I2C unterstützen.