

Bahnübergang mit Andreaskreuzen für die Modelleisenbahn Spur N - Steuerung mit RPi-Pico

Dipl.-Ing. Ulrich Otto Romahn

Grundlagen: Online-Workshops PicoBello¹ und Elektronik-Kompendium - Raspberry Pi Pico²

Meine Modelleisenbahn wird mit mehreren Bahnübergängen, die neben Schranken immer auch Andreaskreuze haben, ausgestattet. In die Andreaskreuze sind rote Blinklichttafeln integriert, die mit roten SMD-LED bestückt werden.



Bildquelle: Fa. Faller - Andreaskreuze und Warnbaken 272913 (Andreaskreuz mit Blinklicht für unbeschränkte oder halbbeschränkte Bahnübergänge)

Klassisch werden im Eisenbahnmodellbau Reedkontakte verwendet, die durch Magnete ausgelöst werden, die unter dem rollenden Material verbaut werden. Einerseits sind die Probleme mit Reedkontakten bekannt und andererseits schränkt diese Vorgehensweise den unabhängigen ggf. automatisierten Zugbetrieb erheblich ein, da bei der Zusammenstellung der Züge immer auch 2 Fahrzeuge mit Magneten dabei sein müssen. Aus diesem Grund soll auf diese Lösung verzichtet werden.

Die Auslösung des Blinkvorgangs soll unabhängig von der Länge des Zuges sein, egal ob nur eine Lok fährt oder ein sehr langer Zug. Weiterhin soll sie auch funktionieren, wenn der Bahnübergang mehrgleisig ist. Der Auslösevorgang soll über „Lichtschranken“ realisiert werden.

¹ Quelle: <https://www.elektronik-kompendium.de/service/events/>

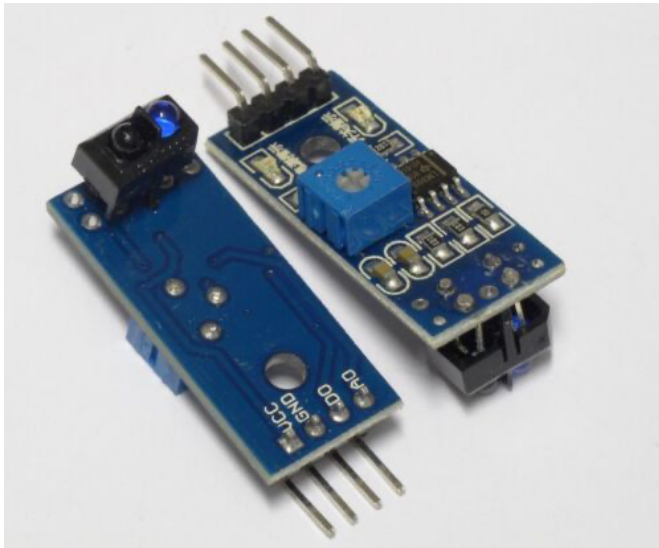
² Quelle: <https://www.elektronik-kompendium.de/sites/raspberry-pi/pico.htm#a1>

Die ersten Versuche wurden mit klassischen analogen Schaltungen in Transistortechnik durchgeführt. Hier zeigte sich schnell, dass das keine zielführende Lösung ist, da das Löten auf Lochrasterplatinen viel Zeit in Anspruch nimmt und Anpassungen, z.B. der Blinkfrequenz, immer aufwendig ist.

Aus diesem Grund sollte ursprünglich ein alter Raspberry PI 3B genutzt werden. Bei der intensiven Befassung bin ich dann auf den RPi Pico und Micropython gestoßen. Ein Buch zum Einstieg und erste Versuche mit Blinkschaltungen und Lichtschranken zeigten deutlich, dass diese Technologie gut für den Modellbahnbetrieb genutzt werden kann.

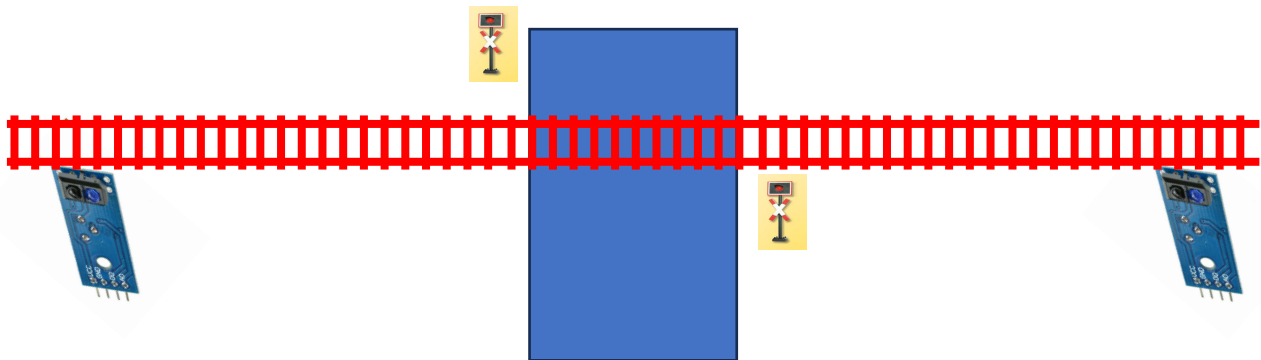
Im Laufe weiterer Recherchen bin ich dann auf die PicoBello-Online-Workshops und die „unendlichen“ Anregungen des Elektronik-Kompendiums aufmerksam geworden.

Im Folgenden möchte ich zeigen, wie ich unter Nutzung des Erlernten einen Bahnübergang mit Blinklicht und dessen Steuerung mit Lichtschranken von Typ TCRT5000 realisiert habe.



Quelle: eBay

Teststrecke Bahnübergang



Logik-Bahnübergang

- Abfrage der Statie der reflektierenden IR Lichtschranken TCRT5000
 - Status „0“ = Objekt erkannt
 - Status „0“ = Blinklicht wird eingeschaltet
 - Status „1“ = Blinklicht wird ausgeschaltet
- Oder-Verknüpfung der Statie der TCRT5000
 - Status „0“ = Blinklicht bleibt eingeschaltet
 - Status „1“ = Blinklicht wird ausgeschaltet

Die ausgegebenen Statie des TCRT5000 sind bei meinen Bauteilen invertiert.

Hinweis:

Die nachfolgenden Ausführungen basieren auf Informationen von Elektronik-Kompendium³ und den PicoBello-Workshops⁴. Sie sind deshalb so ausführlich, um zu zeigen, wie einfach es auch für ungeübte Programmierer ist, durch Kombination der Programme Lösungen für individuelle Anwendungen zu erzeugen.

Objekterkennung mit TCRT5000⁵

Mit dem Infrarot-LED-Sensor TRCT5000 wird eine einfache Objekterkennung (Zug) aufgebaut. In diesem Fall wird ein Objekt erkannt, zu dem sich der Abstand verringert. Der TCRT5000 dient hier als Näherungssensor. Diese Eigenschaft wird für die Zugererkennung genutzt. Die beiden verwendeten TRCT5000 haben hierbei einen festen Abstand zum Gleis. Mit den beiden Bauteilen wird so eine optische Objekterkennung realisiert.

Mit dem nachstehenden Programmcode soll einerseits geprüft werden, ob mit dem TRCT5000 die Annäherung eines Objekts erkannt und detektiert werden kann und andererseits soll der Sensor mit Hilfe des Potentiometers so kalibriert werden, dass eine sichere Objekterkennung – Zug - möglich ist. Weiter soll erprobt werden, welche

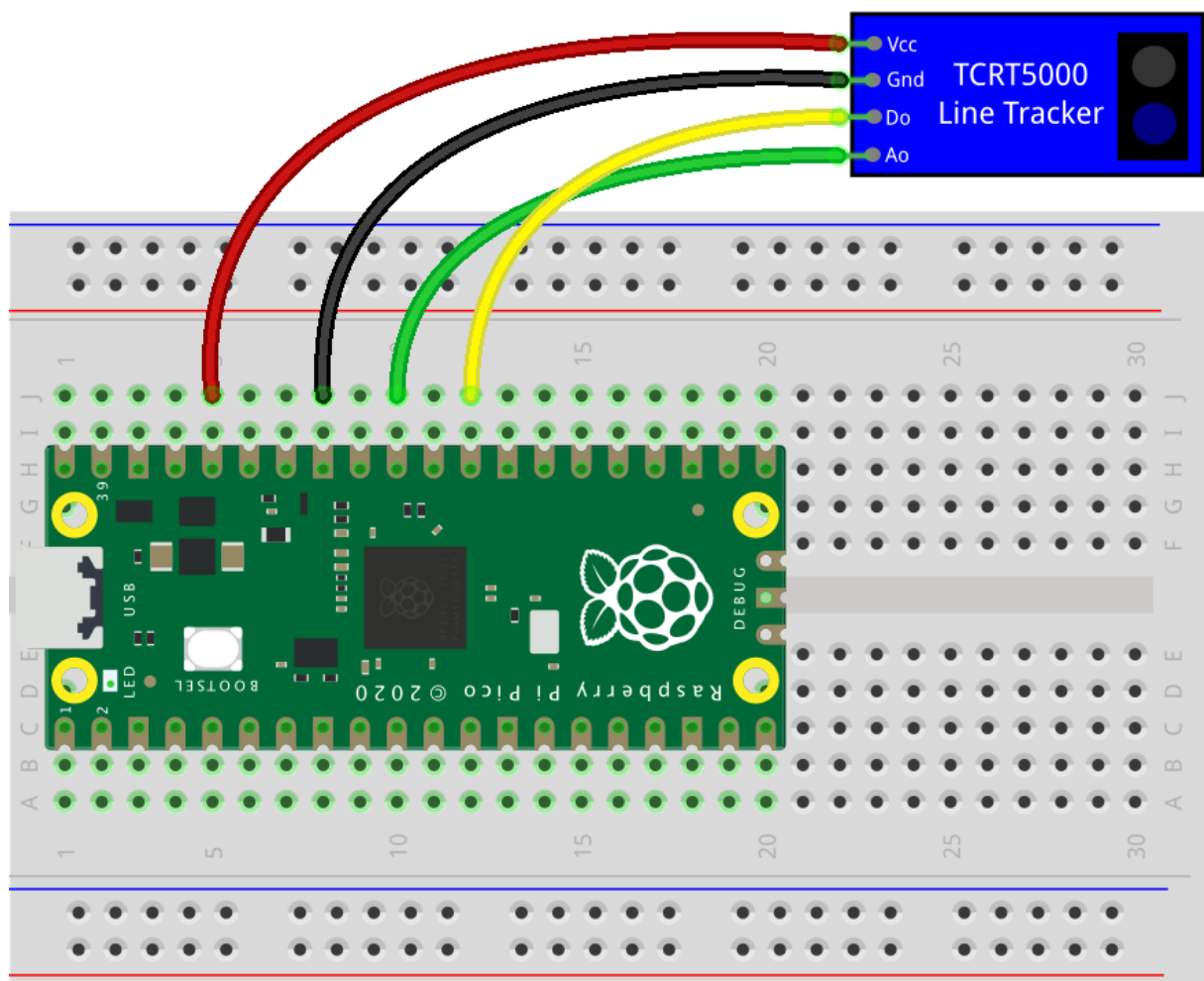
³ Quelle: Elektronik-Kompendium.de; Patrick Schnabel; Droste-Hülshoff-Str. 22/4; D-71642 Ludwigsburg

⁴ Quelle: Dozent Dipl.-Ing. Dieter Carbon; Arbeitskreisleiter „Internet-Sicherheit“ beim VDI Rheingau Bezirksverein

⁵ Quelle: <https://www.elektronik-kompendium.de/sites/raspberry-pi/2801211.htm>

Lichteinflüsse auf einer Modelleisenbahn die sichere Erkennung beeinflussen können. Dies vor dem Hintergrund, dass das Bauteil auch für weitere funktionale Abläufe auf der Modelleisenbahn genutzt werden soll. Beispielhaft sind hier Signalsteuerungen, Automatisierung von Blockstrecken, Zugpendelverkehr auf eingleisigen Strecken zu nennen.

Hilfreiche Hinweise über die Eigenschaften und Möglichkeiten zur Kalibrierung des TCRT5000 finden sich bei Elektronik-Kompendium.



Bildquelle: Elektronik-Kompendium

Raspberry Pi Pico		TCRT5000
Pin 36	3V3 OUT	VCC
Pin 33	GND	GND
Pin 29	GPIO 22	DOUT
Pin 31	GPIO 26 (ADC0)	AOUT

Bildquelle: Elektronik-Kompendium

Hinweis: GPIO 26 wird bei der Bahnübergangssteuerung nicht verwendet.

Programmcode zur Auswertung des digitalen Ausgangs

Der digitale Ausgang des TCRT5000, der mit einem digitalen GPIO-Eingang des Picos verbunden ist, wird in einer Endlos-Schleife nach „0“ ausgewertet. In diesem Fall hat der Sensor ein Objekt erkannt und die Onboard-LED leuchtet auf. Man könnte den Sensor natürlich auch nach „1“ auswerten und damit das Entfernen eines Objekts erkennen.

Die Werte „0“ und „1“ werden kontinuierlich auf der Kommandozeile ausgegeben.

```
# Bibliotheken laden
from machine import Pin
from time import sleep

# Initialisierung: Onboard-LED
led_onboard = Pin (25, Pin.OUT, value=0)

# Initialisierung: GPIO22 als Eingang
sensor_d = Pin(22, Pin.IN)

# Wiederholung (Endlos-Schleife)
while True:
    value_d = sensor_d.value()
    print(value_d)
    if value_d == 0:
        led_onboard.on()
    else:
        led_onboard.off()
    sleep(0.2)
```

Quelle: Elektronik-Kompendium

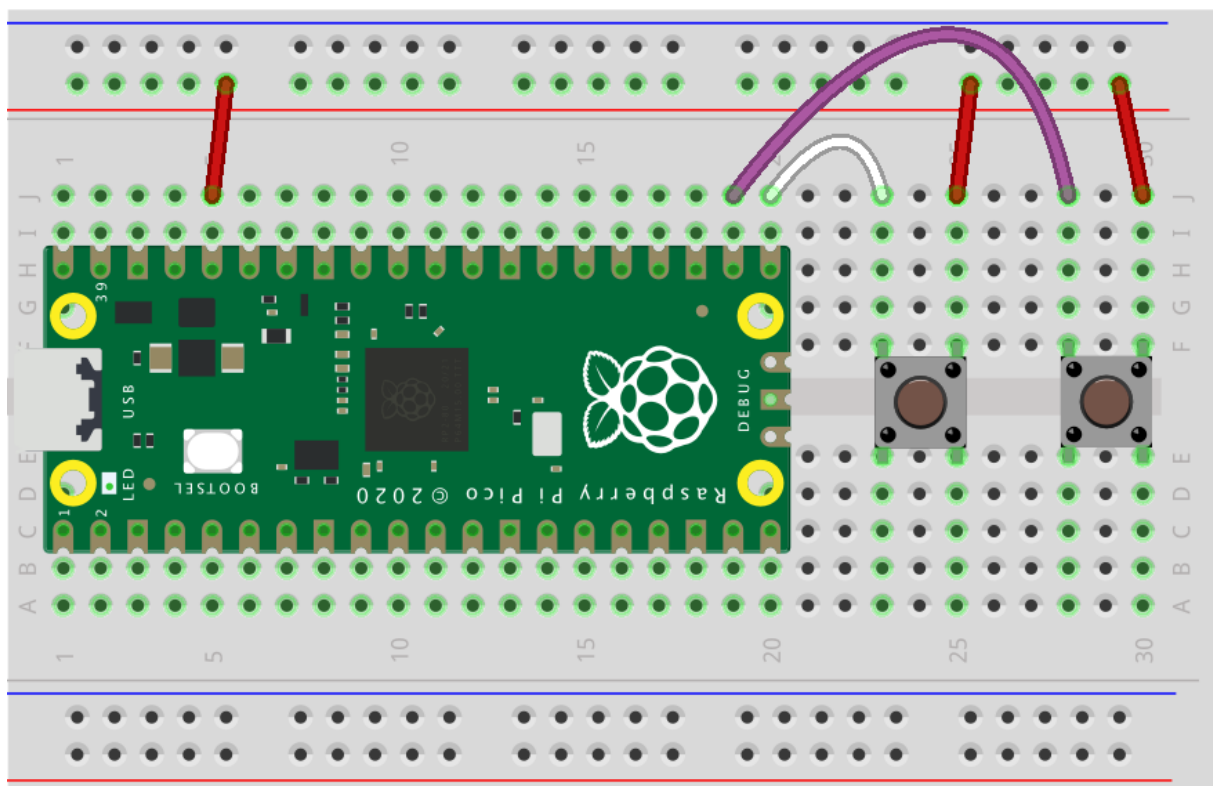
ODER-Funktion mit zwei Tastern⁶

Die Verknüpfung der Ausgangssignal der beiden TCRT5000 erfolgt mittels einer ODER-Funktion. Im Nachfolgenden erfolgt die Erprobung zunächst mit 2 Tastern, statt der TCRT5000, siehe hierzu Fußnote 6.

Eine ODER-Funktion (OR) ist eine logische Funktion, die Teil einer Bedingung ist. WENN „Bedingung“ erfüllt, DANN „Funktion“.

Die Idee dieser Bedingung ist, dass eine Ausführung im Programmablauf nur dann erfolgt, wenn einer der beiden Taster gedrückt wird. Die Bedingung ist auch erfüllt, wenn beide Taster zusammengedrückt werden. In diesem Fall leuchtet beispielhaft eine Leuchtdiode.

Im Weiteren werden die Taster durch 2 TCRT5000 ersetzt.



fritzing

Bildquelle: Elektronik-Kompodium

⁶ <https://www.elektronik-kompodium.de/sites/raspberry-pi/2701181.htm>

Programm zur Umsetzung der Oder-Funktion

Im Programm werden die LED und die Taster initialisiert. Anschließend wird in einer Endlosschleife die Bedingung überprüft, ob die Taster nach der ODER-Funktion betätigt sind. Nur dann leuchtet die Onboard-LED.

```
# Bibliotheken laden
from machine import Pin
from time import sleep

# Initialisierung der Onboard-LED
led_onboard = Pin(25, Pin.OUT, value=0)

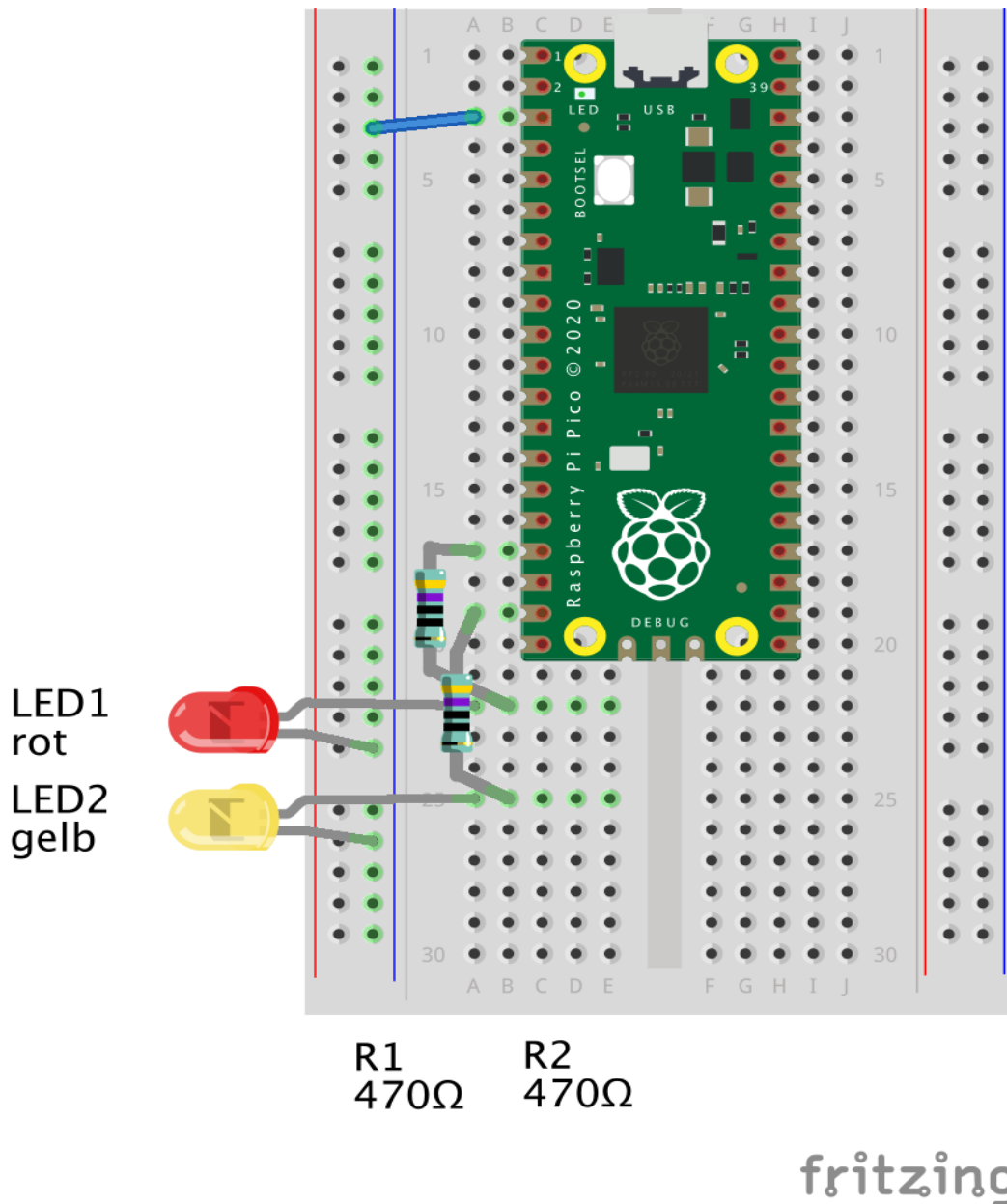
# Initialisierung der GPIOs als Eingang
btn1 = Pin(16, Pin.IN, Pin.PULL_DOWN)
btn2 = Pin(17, Pin.IN, Pin.PULL_DOWN)

# Wiederholung (Endlos-Schleife)
while True:
    # Logische ODER-Verknüpfung (OR)
    if btn1.value() == 1 or btn2.value() == 1:
        led_onboard.on()
    else:
        led_onboard.off()
```

Quelle: Elektronik-Kompendium

LED-Wechselblinker programmieren⁷

Bereits im PicoBello1 blinkte die erste (Onboard-) LED. In PicoBello2 dann eine externe. 2 LEDs abwechselnd Blinken zu lassen, sollte nicht problematisch sein. Über das Programm wird realisiert, dass die eine LED eingeschaltet und die andere ausgeschaltet wird und dies dann im gewünschten Rhythmus.



Bildquelle: Elektronik-Kompodium

⁷ Quelle: <https://www.elektronik-kompodium.de/sites/raspberry-pi/2612081.htm>

Einzelne LED-Steuerung

Von den dargestellten Varianten (siehe Fußnote 7) wurde die folgende Lösung gewählt. Hier werden zwei LEDs über je einen eigenen GPIO unabhängig voneinander angesteuert und so abwechselnd ein- und ausgeschaltet.

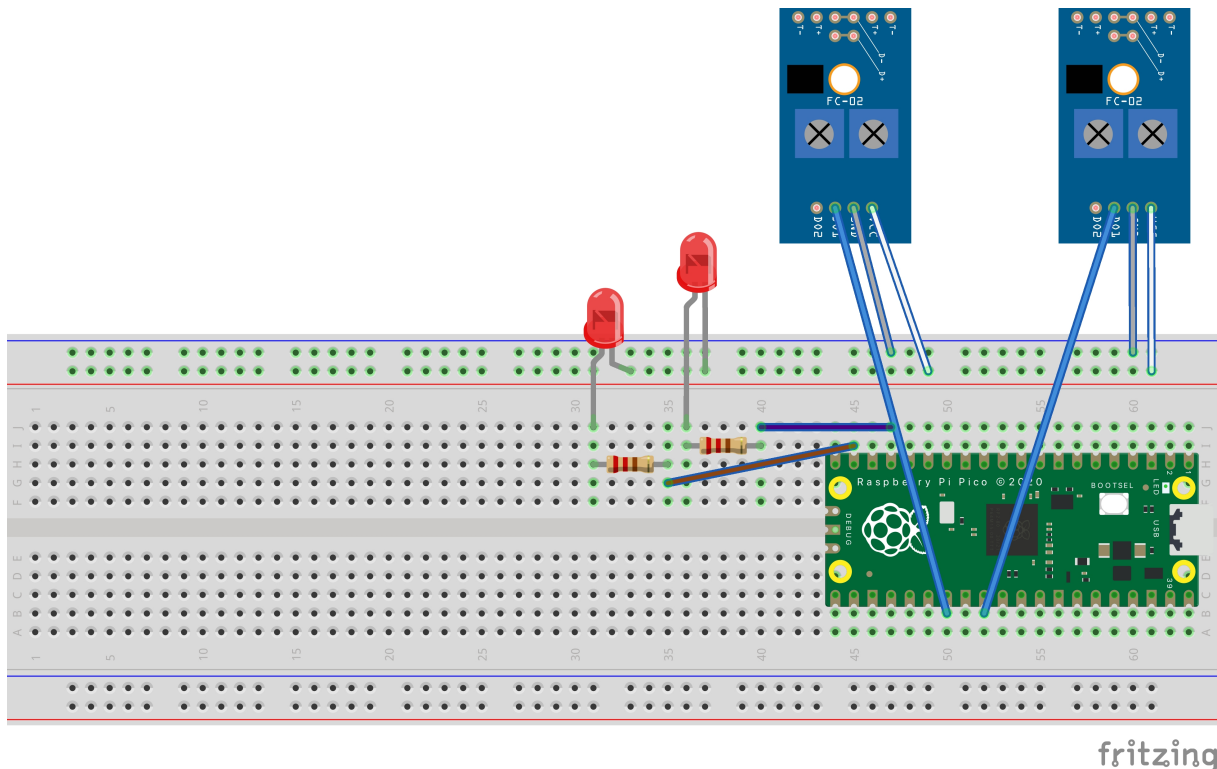
```
# Bibliotheken laden
from machine import Pin
from time import sleep

# Initialisierung von GPIO13 als Ausgang
led1 = Pin(13, Pin.OUT)
led2 = Pin(14, Pin.OUT)

# Wiederholung einleiten (Schleife)
while True:
    # LEDs: EIN/AUS
    led1.on()
    led2.off()
    # 1 Sekunde warten
    sleep(1)
    # LED: AUS/EIN
    led1.off()
    led2.on()
    # 1 Sekunde warten
    sleep(1)
```

Quelle: Elektronik-Kompendium

Steuerung der Signalanlage des Bahnübergangs



Schaltung für die Steuerung der Signalanlage des Bahnübergangs

Aus den vorgenannten Programmen wurde letztendlich das nachstehende Programm (20231212_läuft_Bahnübergang_blinkend_Steuerung_2_Lichtschraken.py) entwickelt, um die Steuerung umzusetzen und zu erproben.

```
# Bibliotheken laden
from machine import Pin
from time import sleep
```

```
# Initialisierung: Onboard-LED
led1 = Pin(13, Pin.OUT, value=0)
led2 = Pin(14, Pin.OUT, value=0)
```

```
# Initialisierung: GPIO21 und 22 als Eingang für die 2 TRCT5000
btn1 = Pin(21, Pin.IN)
btn2 = Pin(22, Pin.IN)
```

```
# Wiederholung (Endlos-Schleife)
while True:
    #value_d = sensor_d.value()
    # print(value_d)
    #if value_d == 0:
    if btn1.value() == 0 or btn2.value() == 0:
```

```
# LED einschalten
led1.on()
led2.on()
# halbe Sekunde warten
sleep(0.5)
# LED ausschalten
led1.off()
led2.off()
# 1 Sekunde warten
sleep(0.5)

else:
    led1.off()
    led2.off()
```

An einer einfachen Testgestellung (siehe Bild Teststrecke Bahnübergang) wurden die Schaltung und das Programm überprüft. Eine Erweiterung auf 2 parallele Gleise ist ohne Probleme möglich. Beleuchtete Zügeinheiten (Streulicht) stellen kein Problem für den sicheren Betrieb dar.