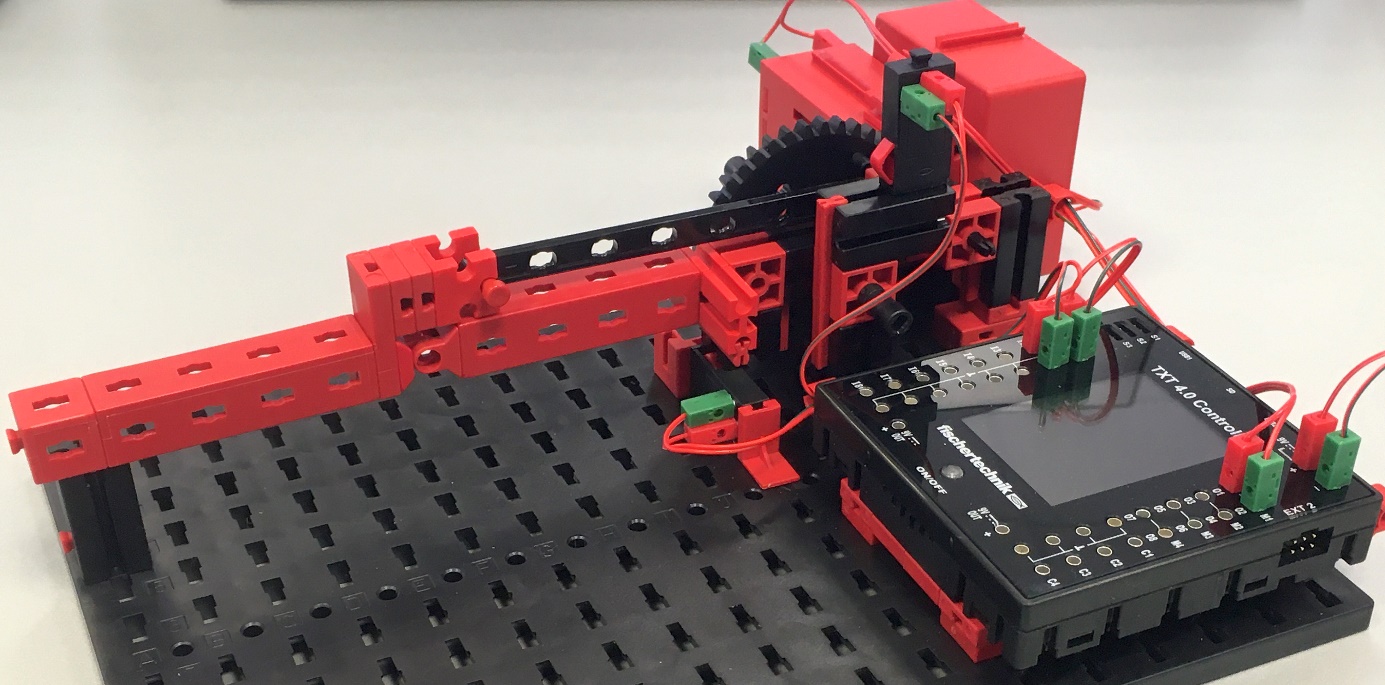
Aufgabe 2:

# Parkhausschranke

## Konstruktionsaufgabe

Konstruiere die erste Version der Schranke gemäß der Bauanleitung. Schließe den Motor und die Taster an den TXT an (siehe Schaltplan). Prüfe mit dem Interface-Test, in welche Richtung sich der Motor jeweils dreht (Achtung: geringe Geschwindigkeit wählen!). Die beiden Endlagentaster sollten so an die Eingänge I1 (untere Endlage) und I2 (obere Endlage) angeschlossen werden, dass sie den (digitalen) Wert „1“ liefern, wenn sie gedrückt sind.



## Programmieraufgaben

**1. Schranke mit Endlagentastern**

Die Schranke soll alle 10 Sekunden automatisch öffnen, in geöffneter Position fünf Sekunden „verharren“ und dann wieder schließen.

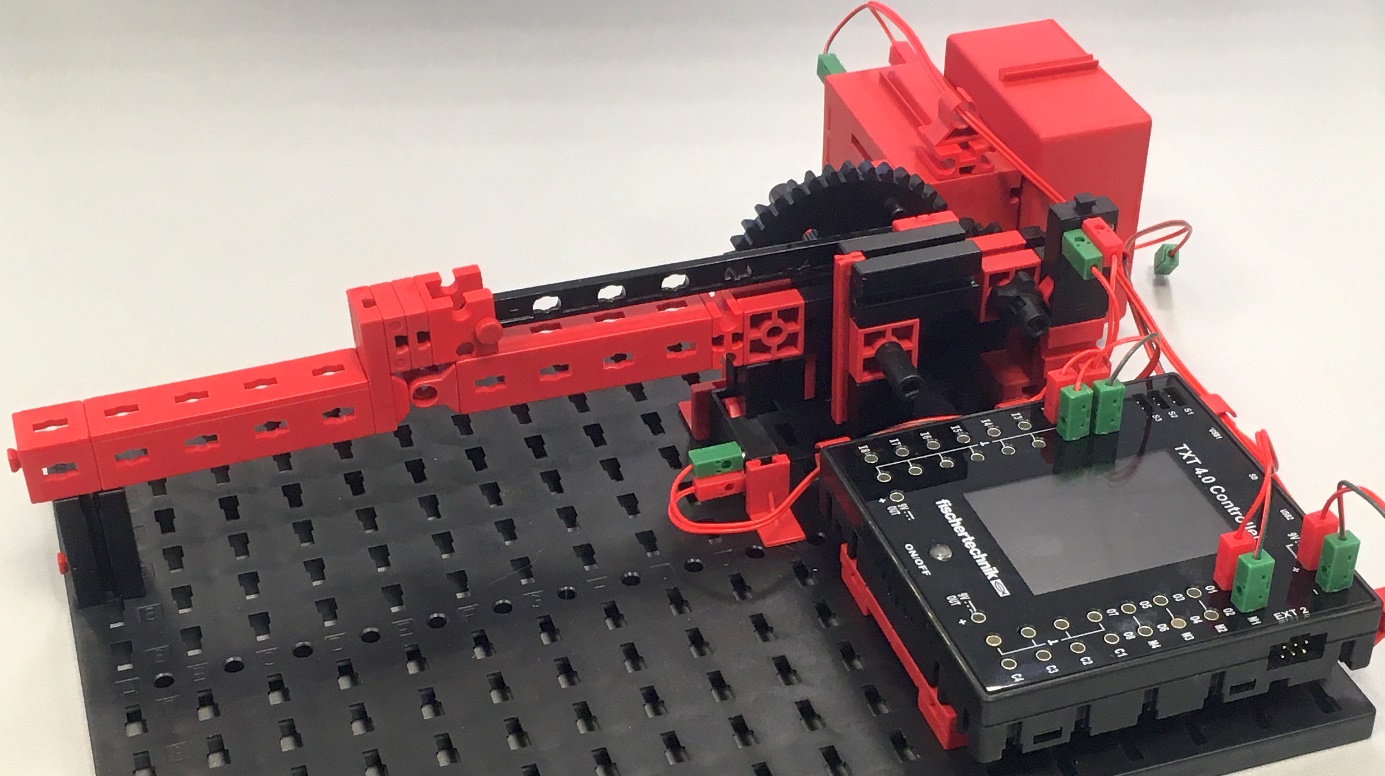
1a. Zeichne zunächst ein Zustands­übergangsdiagramm zur Aufgabenstellung, bevor du mit der Program­mierung beginnst.

Beachte: Die Schranke ist beim Starten des Programms möglicherweise in keinem eindeutigen Zustand – in welchen Zustand deines Zustandsübergangsdiagramms sollte dein Programm aus dem Startzustand übergehen?

1b. Schreibe ein entsprechendes Programm und teste es. Bleibe bei der Program­mierung nah an deinem Zustandsübergangs­diagramm.

**2. Schranke mit Impulsrad**

Baue die Schranke zu Version 2 um. Der obere Endlagentaster (für „Schranke oben“) entfällt; stattdessen wird dieser Taster nun von einem Impulsrad betätigt. Schließe diesen Impulstaster an C1 an (einen der schnellen Zählereingänge).



2a. Berechne zunächst die Anzahl der Impulse, die beim Schließen bzw. Öffnen der Schranke mit dem Taster gemessen werden können.

2b. Passe dein Zustandsübergangsdiagramm an die geänderte Konstruktion der Schranke an. Einen Endlagentaster (für „Schranke unten“) benötigst du auch hier, damit die Schranke in einem beliebigen Zustand gestartet werden kann und zunächst einmal schließt.

2c. Modifiziere anschließend dein Programm aus Programmieraufgabe 1 ent­sprechend, sodass die Schranke beim Öffnen nicht durch Betätigung des Endlagen­tasters, sondern nach einer bestimmten Zahl von Impulsen des Impulsrads stoppt.

**3. Schranke mit Magnetencoder**

Der Motor besitzt einen Encoder, der je Achsumdrehung 63,9 Impulse liefert. Damit kannst du die Schranke auch ohne das Impulsrad präzise öffnen. Entferne das Impulsrad und den Impulstaster und schließe das Encoder-Kabel des Motors (wie im Schaltplan der Bauanleitung) am TXT an C1 an.

3a. Berechne zunächst, nach wie vielen Impulsen deine Schranke geöffnet bzw. geschlossen ist. Überprüfe den Wert mit dem Interface Test, indem du die Schranke manuell schließt, den Zähler zurücksetzt und die Schranke dann öffnest.

3b. Passe im Programm die Impulszahl entsprechend an.

## Experimentieraufgaben

**1. Schranke mit Bedarfstaster**

Bisher öffnet die Schranke automatisch, nicht erst bei Bedarf. Das wollen wir nun durch einen „Bedarfsmelder“ ändern.

Da einer der beiden Taster nun frei ist, kannst du ihn als „Bedarfsmelder“ verwen­den. Montiere ihn an einem Baustein 30 vor der Schranke, damit er von einem Autofahrer erreicht werden kann. Schließe ihn am TXT an den Eingang I2 an. Wenn er gedrückt wird, soll die Schranke drei Sekunden später öffnen.

1a. Passe dein Zustandsübergangsdiagramm so an, dass die Schranke drei Sekunden nach Betätigung des Bedarfstasters öffnet. Zu Beginn des Programms soll die Schranke zunächst automatisch schließen, wenn sie nicht bereits geschlossen ist.

1b. Ändere dein Programm entsprechend.

**2. Schranke mit Durchfahrtskontrolle**

Nun schließt die Schranke allerdings noch automatisch nach einer festen Zeiteinheit (fünf Sekunden). Wenn zwei Fahrzeuge schnell sind, könnten beide hindurchfahren, und wenn ein Fahrzeug sehr langsam ist, könnte es von der schließenden Schranke beschädigt werden. Beides möchten wir verhindern.

Die Schranke soll erst (mit einer Sekunde Verzögerung) schließen, nachdem das Fahrzeug (oder ein Fußgänger) sie passiert hat. Ergänze die Schranke daher um eine Lichtschranke (Konstruktion entsprechend Version 3 in der Bauanleitung, Anschluss des Fototransistors an I3, LED an 9V-Spannungsausgang, siehe Schaltplan). Beachte, dass der rot markierte Anschluss des Fototransistors mit dem mit „I3“ gekennzeich­neten Kontakt verbunden werden muss. Überprüfe den Sensor mit dem „Interface Test“.

2a. Erweitere dein Zustandsübergangsdiagramm so, dass die Schranke erst schließt, wenn die Lichtschranke nicht mehr unterbrochen ist.

2b. Ergänze das Programm entsprechend.

**3. Parkhausschranke mit Durchfahrt-Zähler**

Jetzt möchten wir noch wissen, wie viele Fahrzeuge die Schranke passiert haben.

3a. Erweitere dein Zustandsübergangsdiagramm um einen Zähler.

3b. Führe in deinem Programm eine Variable ein, die die erfolgten Durchfahrten zählt und auf dem Display des TXT anzeigt.

## Ergänzungsaufgabe

(Gruppenaufgabe, erfordert zwei Bausätze)

Aufgabe für zwei Schranken mit Bedarfstaster, Lichtschranke und Zähler.

**1. Parkhausschranken mit Belegungskontrolle**

Mit einer Schranke für die Ein- und einer zweiten für die Ausfahrt kannst du aus den Zählerständen der Ein- und Ausfahrten die Anzahl der freien Parkplätze des Park­hauses berechnen und auf dem Display anzeigen. Wenn keine Parkplätze frei sind, soll die Einfahrtschranke sperren und auf dem Display anzeigen, dass das Parkhaus voll ist. Die maximale Anzahl der Parkplätze des Parkhauses soll für den Test des Programms 10 betragen.

Schließe beide Schranken geeignet an einen TXT an und erweitere dein Programm entsprechend.

**Tipp**: Steuere eine der beiden Schranken in einem nebenläufigen Prozess (Thread).

Anlagen

# Aufgabe 2: Parkhausschranke

## Erforderliches Material

* PC für Programmentwicklung, lokal oder über Web-Schnittstelle.
* USB-Kabel oder BLE- bzw. WLAN-Verbindung für die Übertragung des Programms auf den TXT4.0.

## Weiterführende Informationen

[1] Wikipedia: [*Endlicher Automat (Zustandsautomat)*](https://de.wikipedia.org/wiki/Endlicher_Automat)

[2] Ferdinand Wagner, Ruedi Schmuki, Thomas Wagner, Peter Wolstenholme: [*Modeling Software with Finite State Machines. A Practical Approach*](http://is.ifmo.ru/download/modelingsoftwarewithfinitestatemachinesapracticalapproach.pdf). Auerbach Publications, 2006.

[3] Online-Diagrammeditor zur Erstellung von Zustandsübergangsdiagrammen (Format drawio): <https://www.diagrammeditor.de/>