Lösungsblatt

# Aufgabe 2: Parkhausschranke

Das Stoppen des Motors beim Öffnen und Schließen der Schranke wird in drei unterschiedlichen Varianten umgesetzt: mit Endlagentastern, einem Impulsrad und einem Encoder. Darüber lernen die Schüler die Funktion eines Winkel­änderungs­sensors (Drehwinkelgeber) kennen. Die Aufgabe ist außerdem gut zum Verständnis der Bedeutung und Funktionsweise Endlicher Automaten geeignet, daher sollte vor Beginn der Programmierung die Lösung zunächst als Zustandsübergangsdiagramm dargestellt werden.

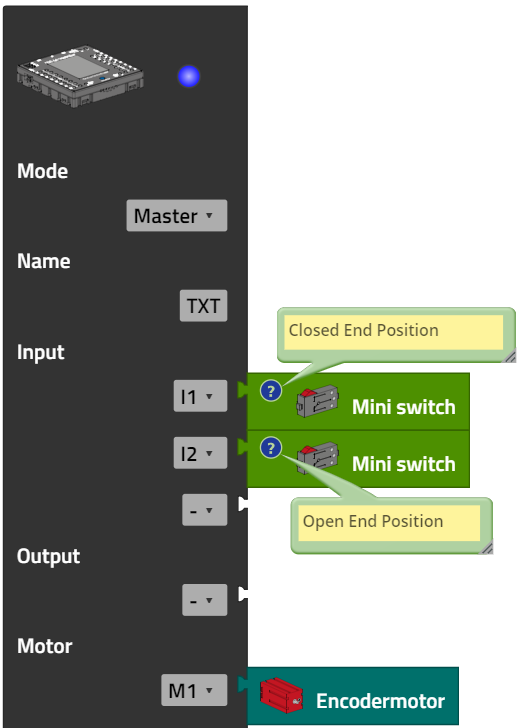
## Konstruktionsaufgabe

Siehe Bauanleitung (Knickschranke, Version 1).

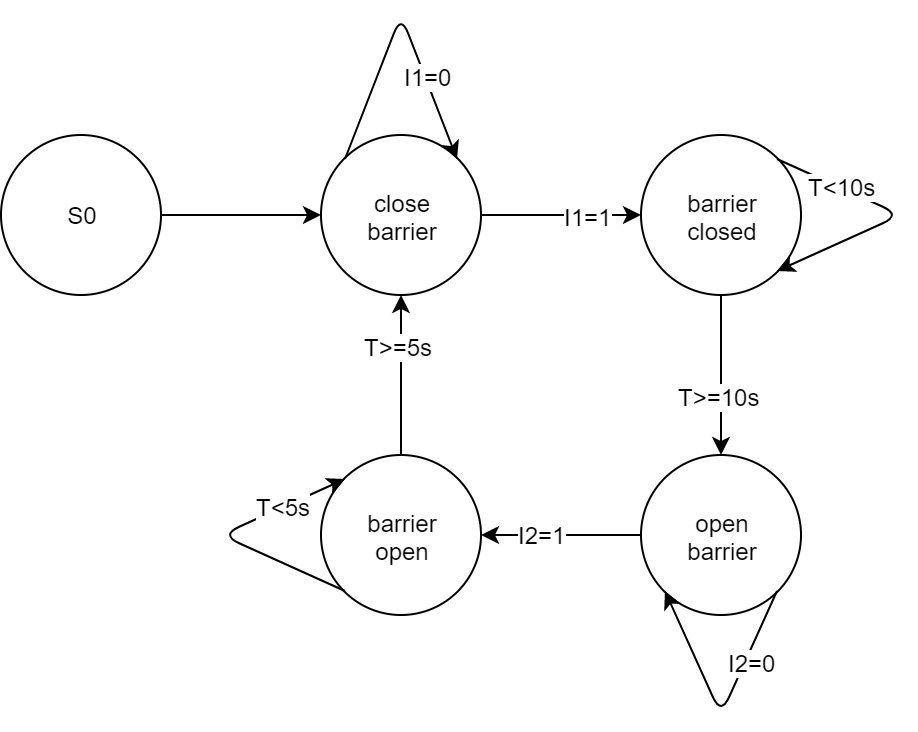
## Programmieraufgaben

**1. Schranke mit Endlagentastern**

Konfiguration der Sensoren und Aktoren:

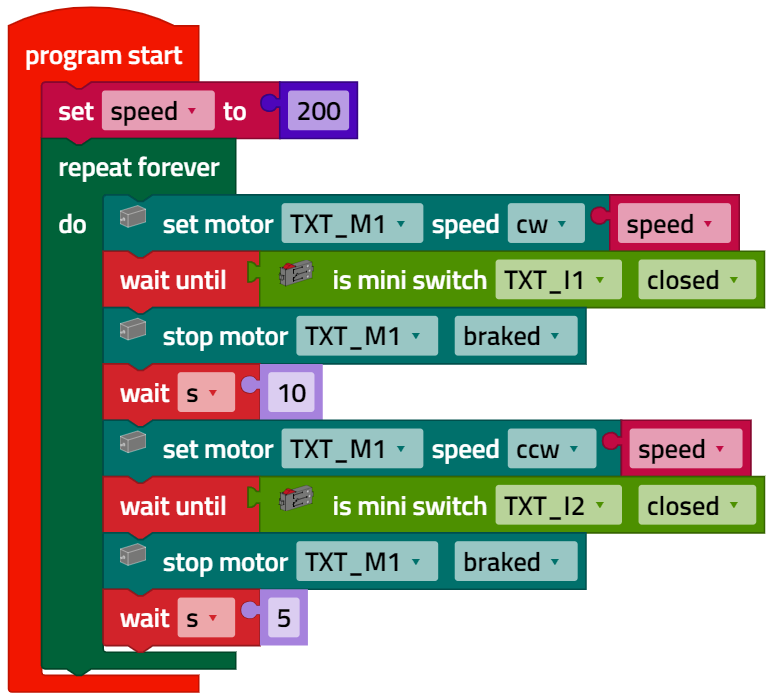


1a. Zustandsübergangsdiagramm:



*State-Transition\_Diagramm\_Boom\_Gate\_with\_Stop\_Position\_Switch.drawio*

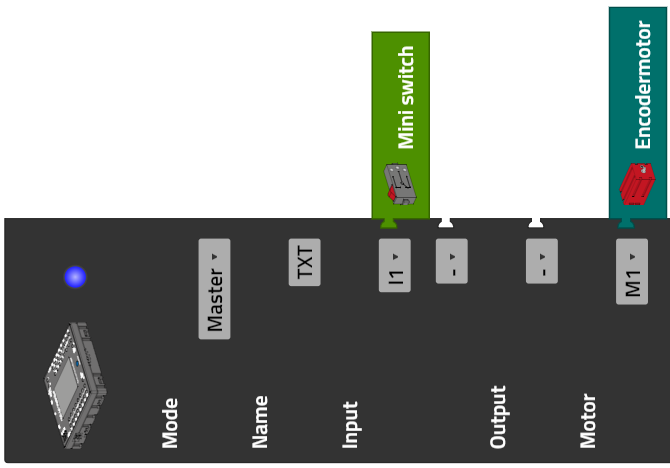
1b. Programm (Beispiel):



*Boom\_Gate\_with\_Stop\_Position\_Switch.ft*

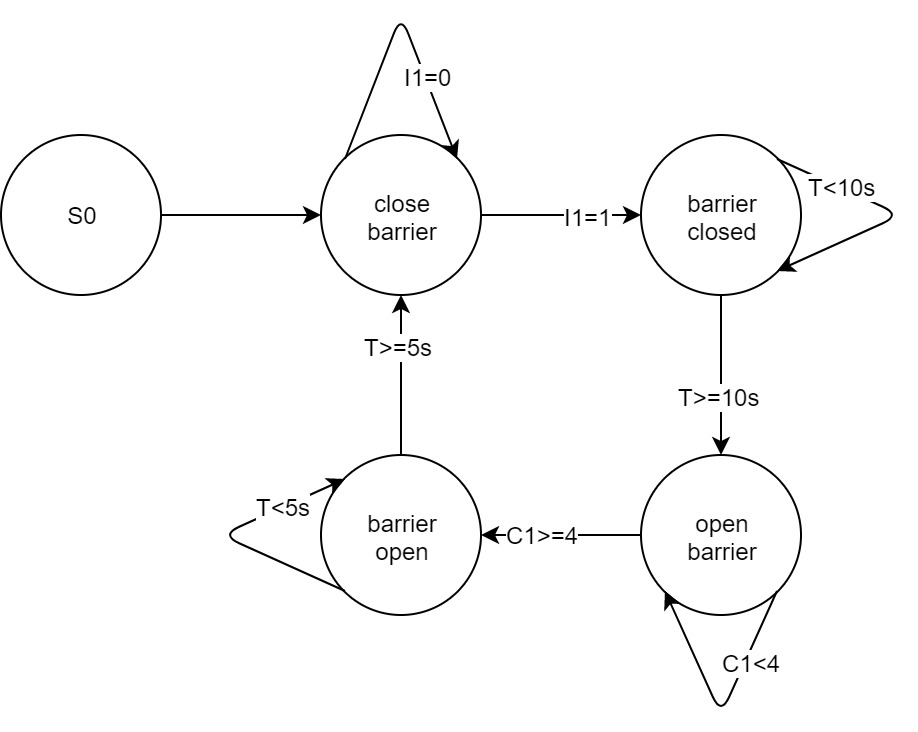
**2. Schranke mit Impulsrad**

An C1 wird der frei gewordene obere Endlagentaster angeschlossen, der nun vom Impulsrad betätigt wird (siehe Bauanleitung, Knickschranke Version 2).



2a. Es sind vier Impulse erforderlich (eine Achsumdrehung des Motors = 4 Impulse, Untersetzung 1:4, also 16 Impulse je Umdrehung der Schrankenachse, für einen Viertelkreis also vier).

2b. Angepasstes Zustandsübergangsdiagramm:

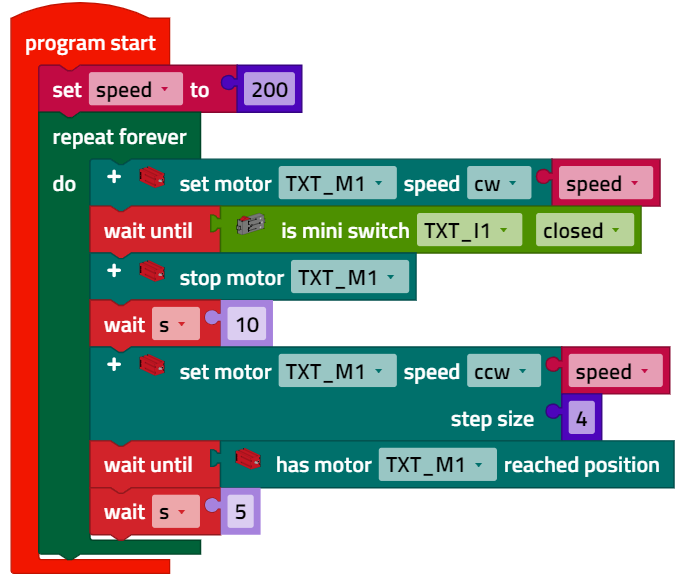


*State-Transition\_Diagram\_Boom\_Gate\_with\_Pulse\_Wheel.drawio*

2c. Der vom Impulsrad betätigte Taster wird als „Encoder Counter“ für den Motor M1 genutzt, damit der Motor mit einem Schrittweiten-Befehl über vier Impulse gesteuert werden kann.

Bei der Programmierung ist zu beachten, dass für die Motoransteuerung nur die Encoder-Motor-Befehle verwendet werden (rotes Motor-Icon).

Programm (Beispiel):



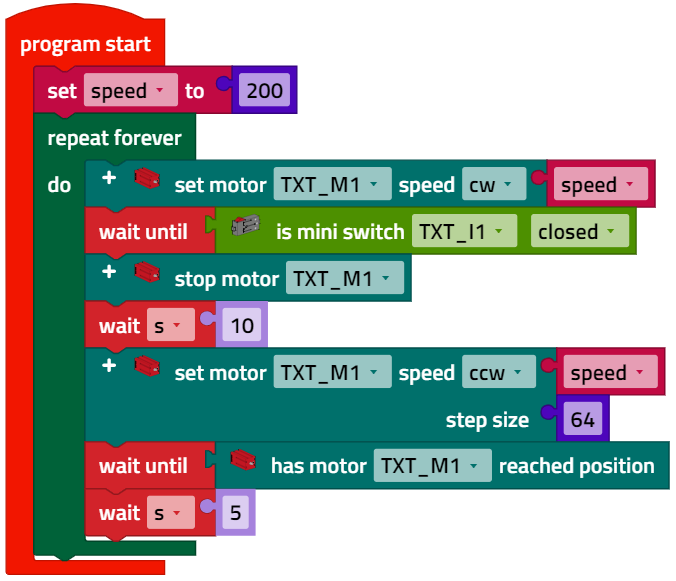
*Boom\_Gate\_with\_Pulse\_Wheel.ft*

**3. Schranke mit Magnetencoder**

Statt des Impulsrad-Tasters wird der Datenausgang des Encoder-Motors an C1 angeschlossen (siehe Bauanleitung, Knickschranke Version 3). Die Konfiguration des Controllers muss nicht geändert werden.

3a. Die Rechnung ist einfach: Impulse.

3b. Programm (Beispiel):

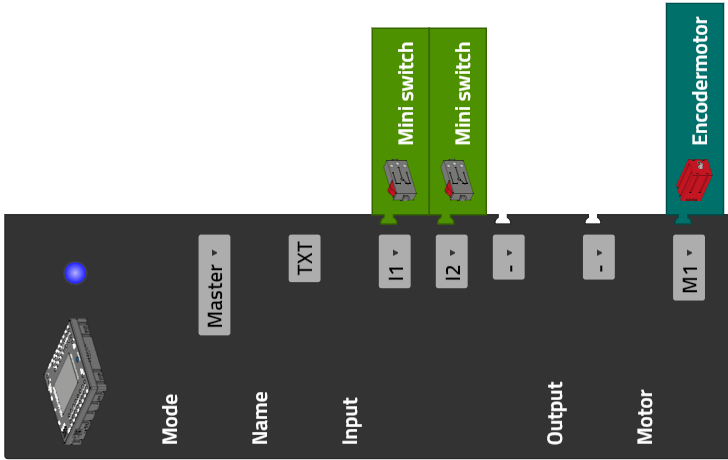


*Boom\_Gate\_with\_Encoder.ft*

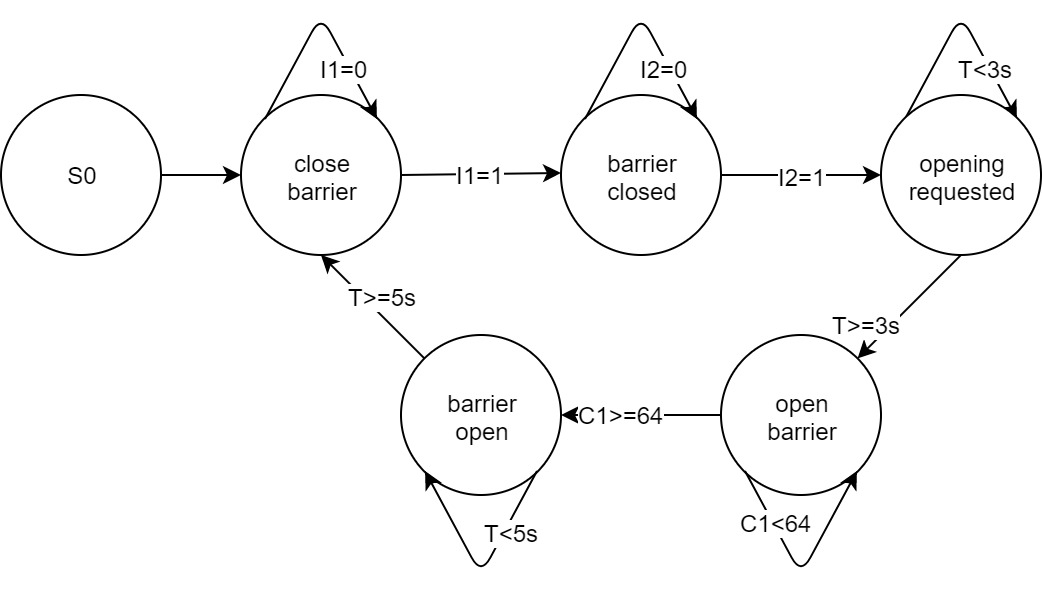
## Experimentieraufgaben

**1. Schranke mit Bedarfstaster**

Der Bedarfstaster wird mit I2 verbunden:

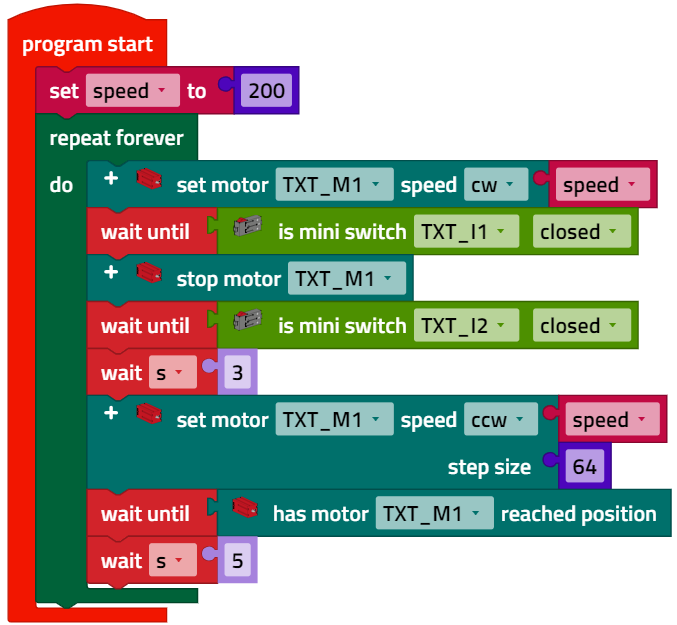


1a. Angepasstes Zustandsübergangsdiagramm:



*State-Transition\_Diagram\_Boom\_Gate\_with\_Pushbutton\_Switch.drawio*

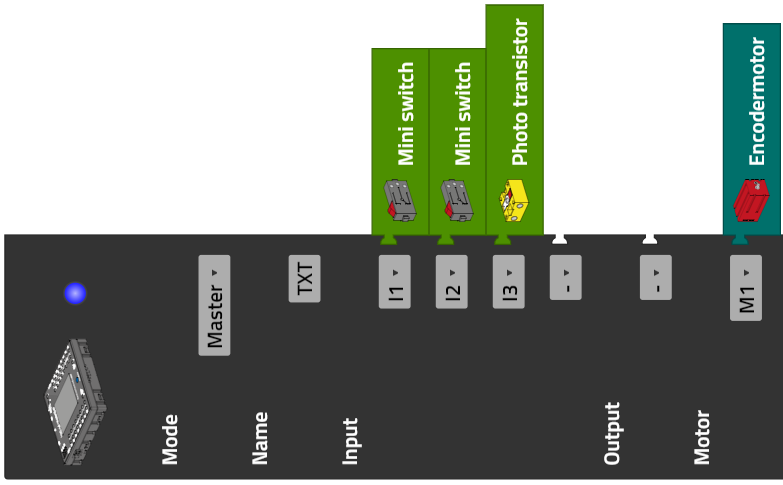
1b. Programm (Beispiel):



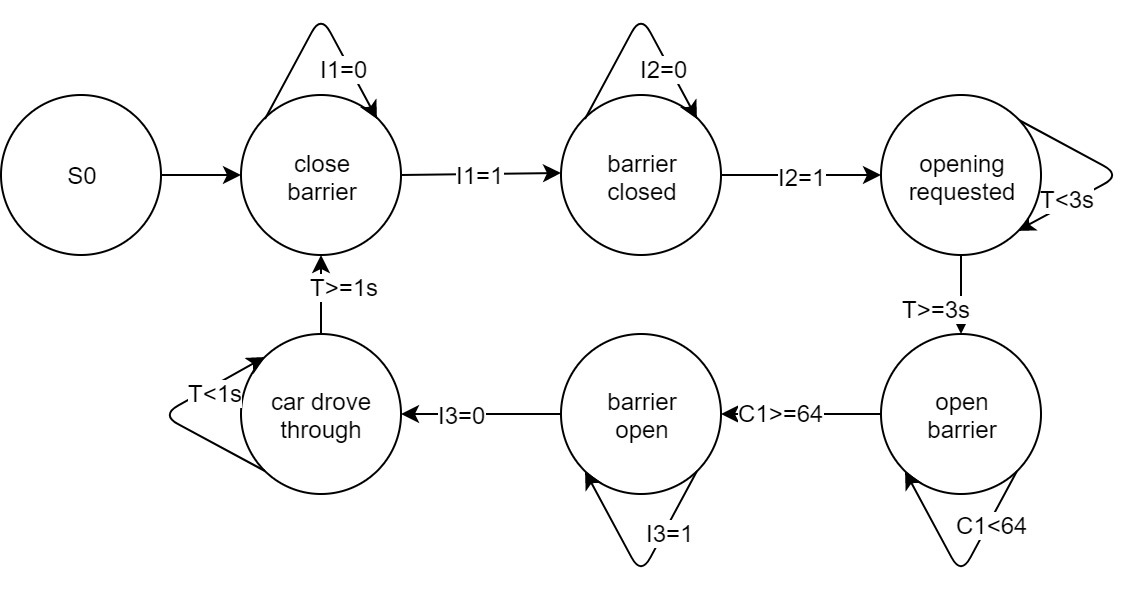
*Boom\_Gate\_with\_Pushbutton\_Switch.ft*

**2. Schranke mit Durchfahrtskontrolle**

Konfiguration des Fototransistors:

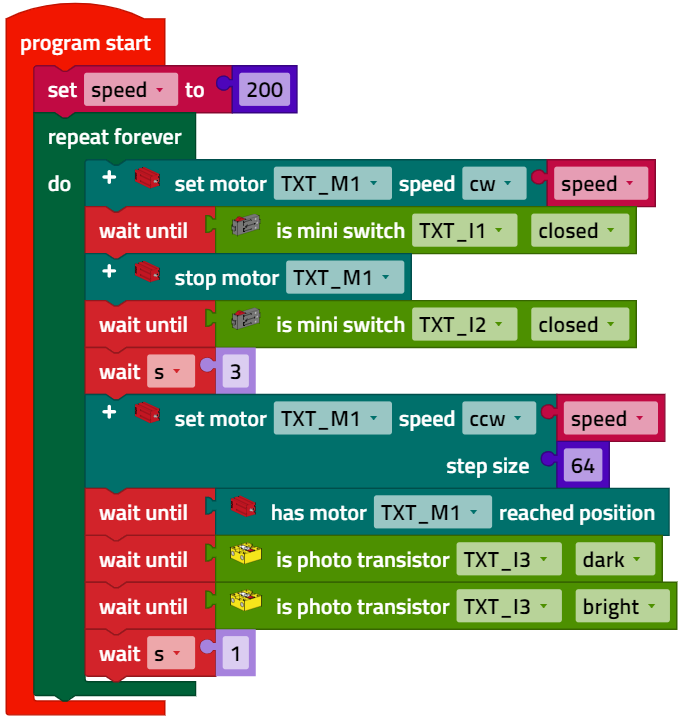


2a. Angepasstes Zustandsübergangsdiagramm:



*State-Transition\_Diagram\_Boom\_Gate\_with\_Pushbutton\_Switch\_and\_Lightbeam.drawio*

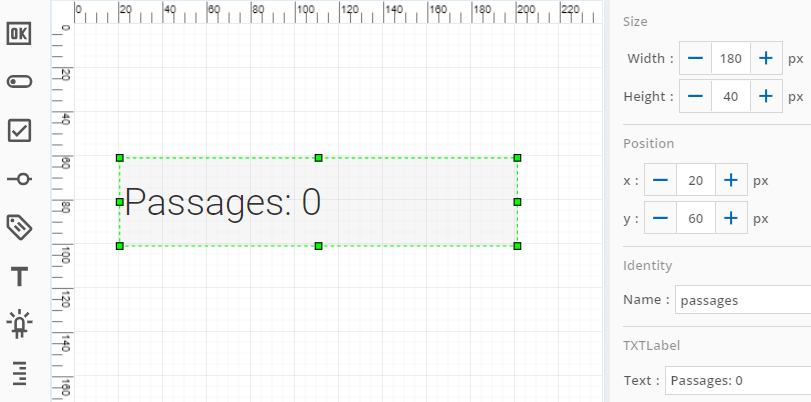
2b. Programm (Beispiel):



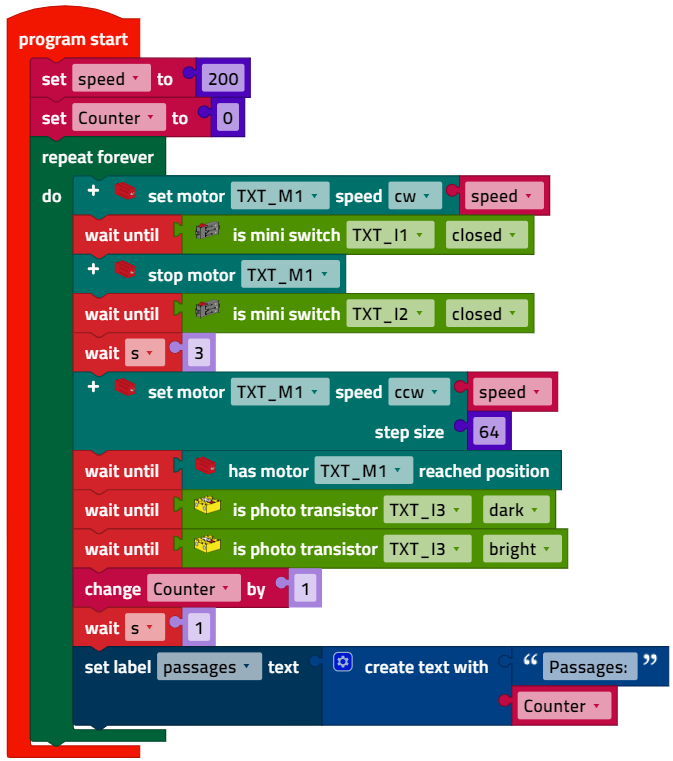
*Boom\_Gate\_with\_Pushbutton\_Switch\_and\_Lightbeam.ft*

**3. Parkhausschranke mit Durchfahrt-Zähler**

3a. Konfiguration der Display-Ausgabe:



3b. Programm (Beispiel):



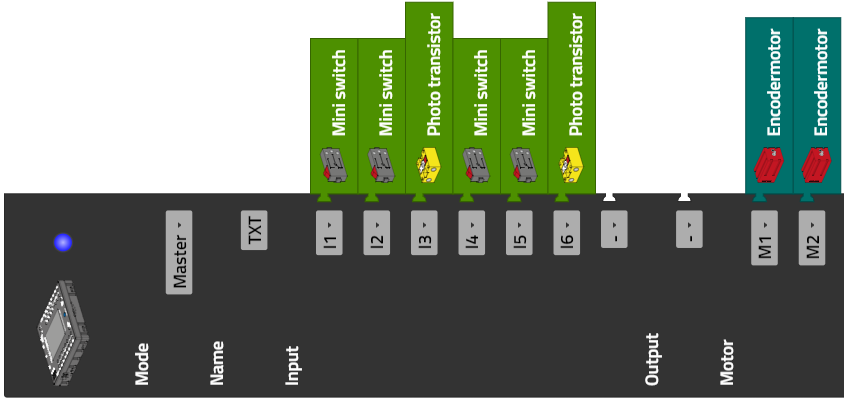
*Boom\_Gate\_with\_Pushbutton\_Switch\_Lightbeam\_and\_Counter.ft*

## Ergänzungsaufgabe

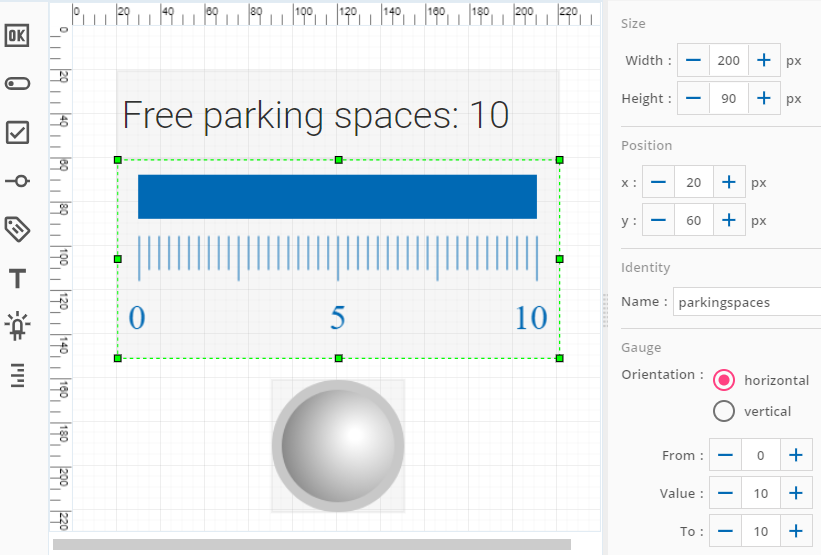
**1. Parkhausschranken mit Belegungskontrolle**

Die Sensoren und Aktoren der zweiten Schranke werden ebenfalls mit dem TXT verbunden: Sensoren an I4 bis I6 (statt I1 bis I3), Motor an M2, LED der Lichtschranke an 9V-Spannungsausgang, Encoder von M2 an C2.

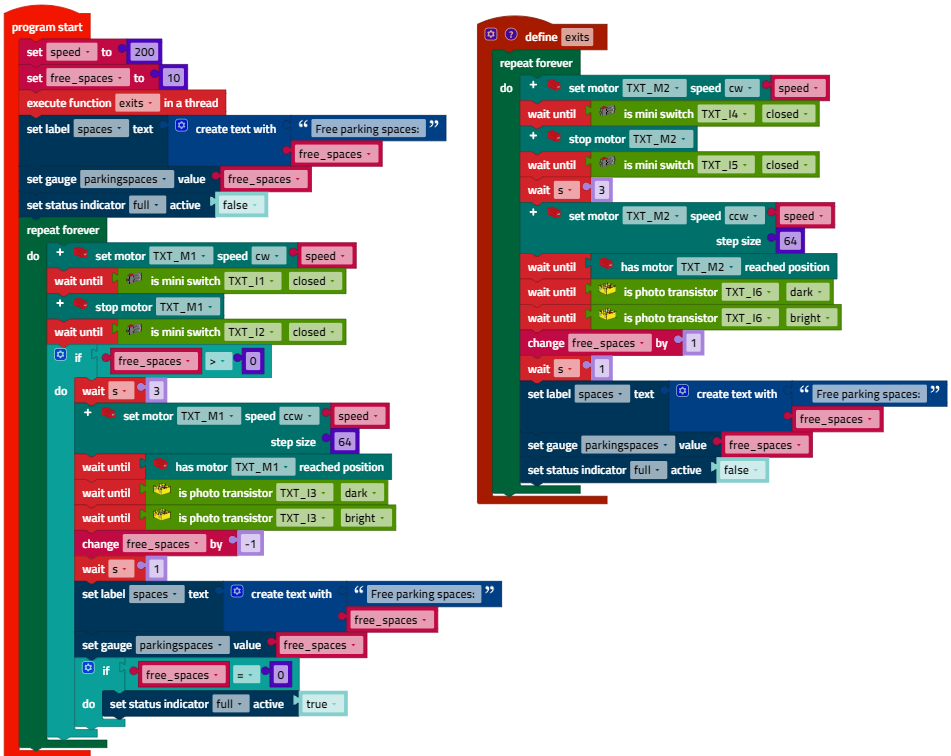
Konfiguration der Sensoren und Aktoren:



Konfiguration des Displays:



Programm (Beispiel):



*Carpark\_Monitoring\_with\_two\_Boom\_Gates.ft*

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/>