Name: \_\_\_\_\_\_\_\_\_\_ Klasse: \_\_\_ Datum: \_\_\_\_\_\_\_\_

# Aufgabe 4

# Einparkhilfe

*Wir statten das Fahrzeug nun mit einem weiteren wichtigen Fahrerassistenzsystem aus: einer Einparkhilfe. Damit soll es selbstständig eine ausreichend große Parklücke finden und in diese rückwärts in zwei Zügen einparken.*

## Konstruktionsaufgabe

Montiere den Ultraschallsensor so an deinem Fahrzeug, dass er den Abstand des Fahrzeugs zu Objekten am rechten Straßenrand messen kann.

## Programmieraufgaben

Konfiguration der Sensoren:



**1. Parklücke finden**

Nun sollst du ein Programm entwickeln, das im Abstand von etwa 2 cm an der rechten Fahr­bahnbegrenzung entlang steuert und dabei mit Hilfe des Ultraschallsensors eine ausreichend große Parklücke sucht. Die Parklücken kannst du durch Objekte (bspw. Kartons) begrenzen. Am Ende der ersten aus­reichend großen Lücke soll das Fahrzeug an der Stelle halten, an der es mit dem Einparkvorgang beginnen kann.

Bestimme dazu zunächst manuell, wie groß (also wie lang und wie breit) eine Park­lücke mindestens sein muss, damit dein Fahrzeug rückwärts einparken kann. Berück­sichtige dabei die Streuung des Ultraschall-Signals: Wann erkennt der Sensor den Anfang und das Ende einer Parklücke, wenn diese durch Objekte begrenzt sind, die sich im Abstand von etwa 2 cm rechts neben der Fahrbahnbegren­zungslinie befinden?

Bestimme schließlich, wie weit das Fahrzeug hinter dem Ende der Parklücke zum Stehen kommen muss, damit es mit dem Einparkvorgang beginnen kann.

1a. Veranschauliche den Einparkvorgang und deine Messungen mit einer Zeichnung.

1b. Skizziere ein Zustandsübergangsdiagramm.

1c. Programmiere die Parklückensuche. Verwende dabei den Spurhalteassistenten aus Aufgabe 2.

Teste dein Programm auf dem Fahrbahnabschnitt des beiliegenden Bogens mit in unterschiedlichen Abständen aufgestellten Hindernissen rechts neben der Fahrbahn­begrenzung.

**2. Einparkmanöver**

Im zweiten Schritt soll das Fahrzeug, nachdem es eine Parklücke gefunden hat, rückwärts in zwei Zügen einparken.

2a. Bestimme zunächst wieder manuell (Interface Test), wie viele Impulse das Fahr­zeug erst mit Lenkeinschlag nach rechts und anschließend mit Lenkeinschlag nach links zurück­setzen muss.

2b. Wie viele Impulse werden dann noch benötigt, damit das Fahrzeug in der Mitte (des zum Einparken benötigten Teils) der Parklücke zum Stehen kommt?

2c. Programmiere den Einparkvorgang und teste ihn auf dem Fahrbahnabschnitt.

## Experimentieraufgaben

**1. Einparkassistent**

Erweitere deinen Einparkassistenten nun um die in Aufgabe 3 entwickelten Funk­tionen, damit er beim Einparken alle Anforderungen der Straßenverkehrsordnung erfüllt (Bremslicht, Rückfahrlicht, Blinker, Abblendlicht). Ergänze das Programm um ein „Erfolgssignal“, das nach Abschluss des Einpark­vorgangs ertönt, und lass‘ dabei den Warnblinker fünf Mal aufleuchten.

**2. Berechnung des Einparkvorgangs**

Den Einparkvorgang kann man auch berechnen: Die Mitte des Differenzials beschreibt zwei Kreisabschnitte, deren Radien du aus dem Lenkeinschlagswinkel ableiten kannst.

2a. Zeichne die Phase des Einparkvorgangs, in der das Fahrzeug rückwärts fährt.

2b. Berechne aus den Winkeln und einigen Setzungen über die richtigen Abstände zur Seitenlinie die dafür benötigten Impulse.

Vergleiche das Ergebnis mit den von dir in der Programmieraufgabe experimentell bestimmten Werten.

Anlagen

# Einparkhilfe

## 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.
* Test-Fahrstrecke

## Weiterführende Informationen

[1] VDA: [*Automatisierung: Von Fahrerassistenzsystemen zum automatisierten Fahren*](https://www.vda.de/dam/vda/publications/2015/automatisierung.pdf). September 2015.

[2] Nina Tetzlaff: [*Autonome Straßenfahrzeuge*](https://www.dpma.de/docs/dpma/veroeffentlichungen/erfinderaktivitaeten/ea_2018_2019.pdf). Erfinderaktivitäten 2018/2019, DPMA, S. 62-81.

[3] VW: [*Park Assist Steering 2.0*](https://www.vdveer-engineering.nl/en/information/vag-ssp/ssp-vag/ssp-vag-en/331-ssp-494-park-assist-steering-20/download). Design and function. Service Training, 2021.