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

# Aufgabe 2

# Bremsassistent, Tempomat und Spurhalteassistent

*Rechtzeitiges Bremsen vor einem Hindernis und eigenständiges Halten der Fahr­spur und der Geschwindigkeit sind wichtige Fahrerassistenzsysteme, die in dieser Aufgabe entwickelt werden. Damit das Fahrzeug bei deinen Tests nicht unkontrolliert davonfährt, wird es zunächst mit einem „Not-Halt“ ausgestattet.*

## Konstruktionsaufgabe

Für diese Aufgabe benötigen wir den Servo-Motor, den Ultraschall-Sensor und die USB-Kamera. Schließe den Ultraschall-Sensor an I1, den Servo-Motor an S1 und die USB-Kamera an USB1 an (siehe Schaltplan). Beim Servo-Stecker auf die richtige Polung achten: das braune Kabel links, das orangene rechts.

**Wichtig**: Wenn der TXT gestartet ist, stellt sich der Servo automatisch auf „Geradeaus-Stellung“. Stecke den Servo-Hebel so auf, dass beide Vorderräder in dieser Position einigermaßen genau nach vorne ausgerichtet sind.

**Achtung**: Wenn du den Servo bei eingeschaltetem TXT mit der Hand bewegst, kannst du ihn beschädigen. Auch darf der Servo keine Position ansteuern, die jenseits des Anschlags des Servo-Hebels liegt – daher sollten für Servo-Befehle nur Positionen zwischen etwa 100 und 400 genutzt werden. Der Lenkbereich unseres Fahrzeugs ist noch etwas kleiner; er liegt ungefähr bei Servo-Werten zwischen 130 und 370.

Mit dem Interface-Test kannst du den Bereich des Lenkeinschlags, in dem sich die Vorderräder noch frei drehen, sehr einfach aus­messen.

Wir werden das Fahrzeug nun mit einem Bremsassistenten, einem Tempomaten und einem Spurhalte­assis­tenten ausstatten.

## Programmieraufgaben

**1. Not-Halt**

Wir beginnen mit einem „Not-Halt“-Mechanismus, bevor wir das Fahrzeug autonom fahren lassen: Bei einem lauten Klatschen soll das Fahrzeug sofort anhalten.

Programmiere den „Not-Halt“ als nebenläufigen Prozess (Thread), der über das Mikrofon der Kamera gesteuert wird. Eine passende Lautstärke-Schwelle, bei der der Motor stoppen soll, kannst du experimentell bestimmen, indem du dir den vom Mikrofon bestimmten Lautstärke-Wert zunächst auf der Konsole anzeigen lässt.

**Beachte**: Der Motor erzeugt selbst ein relativ lautes Geräusch.

**2. Geradeausfahrt**

Du wirst feststellen, dass das Fahrzeug nicht exakt geradeaus fährt. Bevor wir in den Experimentieraufgaben die Lenkung automatisch an der Fahrspur ausrichten, sollst du zunächst versuchen, den Servo einigermaßen auf Geradeausfahrt zu stellen.

Eine erste Näherung erhälst du mit dem Interface-Test. Testen musst du das Ergeb­nis jedoch experimentell: Schreibe ein Blockly-Testprogramm und korrigiere den Wert für die initiale Servo-Einstellung so lange, bis das Fahrzeug über eine Teststrecke von 2 m möglichst exakt die Richtung hält.

Gib‘ in den folgenden Programmen die korrekte Servo-Einstellung in einer Variablen vor und richte den Servo zu Programmbeginn mit diesem Wert aus.

**3. Bremsassistent**

Ein Bremsassistenzsystem soll nun dafür sorgen, dass dein autonom fahrendes Fahrzeug nicht auf ein stehendes (oder auch bewegtes) Hindernis auffährt, wenn der Fahrer unaufmerksam ist oder zu spät reagiert. Dazu soll rechtzeitig eine Bremsung eingeleitet werden, sodass das Fahrzeug dem Hindernis nicht näher als 10 cm kommt.

Den Abstand zu einem Hindernis (oder zu einem vorausfahrenden Fahrzeug) kannst du mit dem Ultraschall-Sensor messen, den du bereits aus dem Robotics TXT 4.0 Base Set kennst.

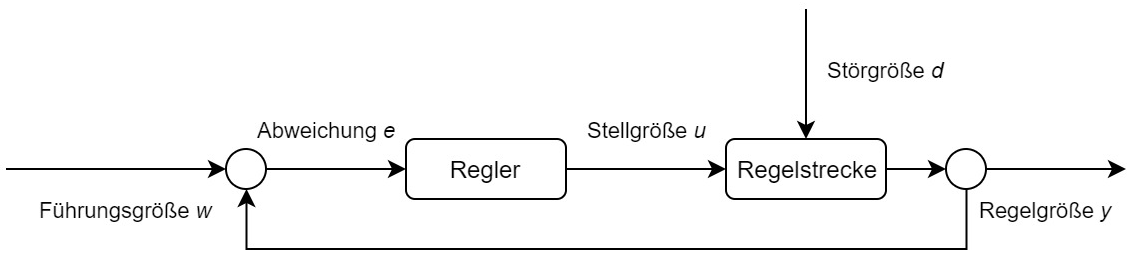
Schreibe ein entsprechendes Blockly-Programm und teste es mit verschiedenen Hindernissen. Nutze den „Not-Halt“-Thread aus der Programmieraufgabe 1, um das Fahrzeug erforderlichenfalls durch Klatschen stoppen zu können.

**Beachte**: Dein Fahrzeug legt auch während der Abstandsmessung eine Fahrstrecke zurück.

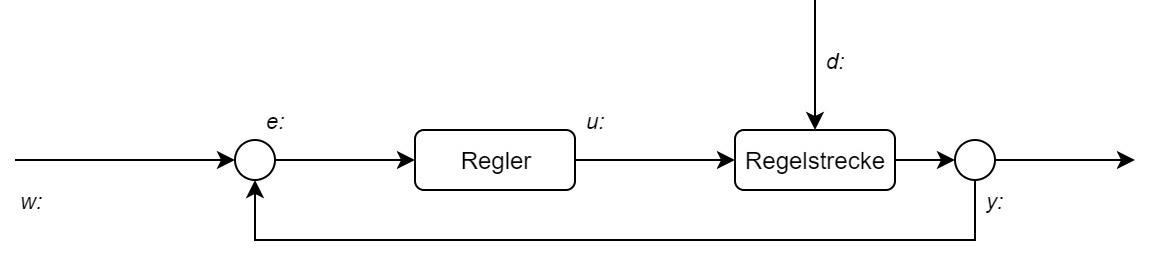
**4. Tempomat**

Ein Fahrerassistenzsystem, das heute in den meisten Mittelklassewagen enthalten ist, ist der *Tempomat*: Ein Geschwindigkeitsregler, der dafür sorgt, dass ein Fahrzeug eine vorgegebene Geschwindigkeit einhält [1].

Betrachte dazu zunächst die folgende Darstellung eines Regelkreises. Du kennst sie bereits aus Aufgabe 8 des Robotics TXT 4.0 Base Set.



4a. Welche Größen entsprechen beim Tempomaten den Parametern *w*, *e*, *u*, *d* und *y*? Beschrifte den folgenden Regelkreis entsprechend:



4b. Programmiere eine Tempomat-Regelung in Blockly. Als Führungs­größe soll die gewünschte Geschwindigkeit in m/h in einer Variablen vorgegeben sein.

**Tipp**: Mit dem „Map“-Kommando kannst du die Führungsgröße sehr elegant in die benötigte Motorspannung umrechnen, wenn du die Geschwindigkeit des Fahrzeugs in m/h kennst.

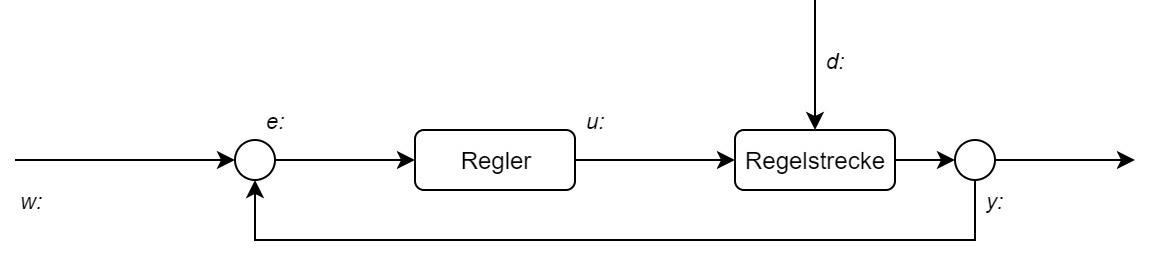
**Beachte**: Bei einen „Not-Halt“ und auch bei einem Halt vor einem Hindernis darf der Tempomat die Geschwindigkeit anschließend nicht wieder hochsetzen. Passe diese beiden Threads in deinem Programm entsprechend an.

## Experimentieraufgabe

**1. Spurhalteassistent mit P-Regler**

Jetzt soll das Fahrzeug lernen, selbstständig einer Fahrspur zu folgen. Dazu soll es sich mit Hilfe der Kamera an der rechten Fahrbahn­begrenzung orien­tieren. Auch dieses Fahrerassistenzsystem sollst du mit einem Proportionalregler (P-Regler) verwirklichen.

1a. Beschrifte zunächst den folgenden Regelkreis:



1b. Erweitere dein Hauptprogramm nun um einen entsprechenden Proportional-Regler, der die Servo-Lenkung abhängig von der Spurabweichung korrigiert.

**Tipp**: Konfiguriere die Linienerkennung der Kamera so, dass sie die Position „0“ liefert, wenn das Fahrzeug in 2 cm Abstand parallel zur rechten Fahrbahnbegrenzung steht.

**Tipp**: Begrenze die Geschwindigkeit deines Fahrzeugs auf 350 und teste den P-Regler zunächst mit einem sehr kleinen Wert (). Erhöhe den Proportionalitätsfaktor so lange, bis der Regler „aufschwingt“, und wähle dann den Wert, bei dem der Regler sich am schnellsten „einschwingt“.

Verwende zum Testen deines Reglers den geraden Straßenabschnitt auf dem beiliegenden Bogen.

1c. Ergänze Dein Programm um eine Textausgabe, die nach jeder Änderung der Position der erkannten Fahrbahnbegrenzung

* die Zeit (in ms), die seit dem Start des Programms verstrichen ist, und
* den Wert der aktuellen Position

durch ein Leerzeichen getrennt auf der Konsole ausgibt.

Kopiere nach jeder Testfahrt mit einem anderen Wert für die Konsolen-Ausgaben in eine Tabellenkalkulation und lass‘ dir die Werte grafisch in einem Diagramm (x: Zeit, y: Position) anzeigen. Wähle den Proportionalitätsfaktor , bei dem sich der Kurven­verlauf am schnellsten „einschwingt“.

**2. Spurhalteassistent mit PD-Regler**

Wie beim Buggy in Aufgabe 8 des Robotics TXT 4.0 Base Set kannst du durch eine Erweiterung des Reglers um ein „D“-Glied (Differential-Anteil), das die Größe der Veränderung der Abweichung der erkannten Fahrbahnbegrenzung von der Position „0“ bei der Lenkkorrektur berück­sichtigt, das Überschwingen weiter dämpfen.

2a. Erweitere den P-Regler aus Experimentieraufgabe 1 zu einem PD-Regler.

2b. Führe Testfahrten mit unterschiedlichen Werten für den Differential-Faktor durch und lass‘ dir die Daten in einem Tabellen­kalku­lationsprogramm grafisch anzeigen.

**Tipp**: Beginne deine Tests mit dem Differential-Faktor und erhöhe dessen Wert so lange in Schritten von 0,005, bis das Überschwingen des P-Reglers gut gedämpft wird.

2c. Wenn du mehrere gerade und gekrümmte Straßenabschnitte aneinander legst, kannst du den Spurhalteassistenten auf einem herausfordernden Straßenparcours testen.

Anlagen

# Aufgabe 2: Bremsassistent, Tempomat und Spurhalteassistent

## 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.
* Fahrbahn mit Markierungen auf beiliegendem Bogen (oder als Ausdruck der Fahrbahn-Datei)
* Hindernis (z. B. ein Buch oder ein Karton)

## Weiterführende Informationen

[1] Jim Meininghaus: [*Die Geschichte des Tempomaten. Wie ein Blinder das Autofahren veränderte*](https://www.motor-talk.de/news/wie-ein-blinder-das-autofahren-veraenderte-t4865108.html). 03.03.2014, motor-talk.de.

[2] Thomas Paulsen: [*Autonomes Fahren: Die 5 Stufen zum selbstfahrenden Auto*](https://www.adac.de/rund-ums-fahrzeug/ausstattung-technik-zubehoer/autonomes-fahren/grundlagen/autonomes-fahren-5-stufen/). 07.11.2018, adac.de.

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