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

# Lösungsblatt Aufgabe 3

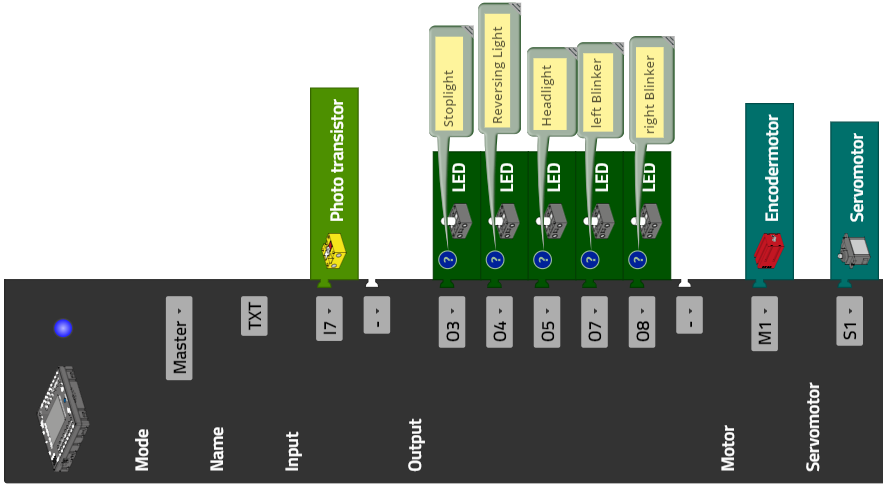
# Lichtautomatik

*Die Threads fragen jeweils einen „Trigger“ ab: den auslösenden Zustand wie die Geschwindigkeit bzw. die Drehrichtung des Motors oder den Einschlagwinkel des Servos. Eine gute Übung für den Nutzen von nebenläufigen Prozessen: Die Steuerung der Beleuchtung wird von der Motorsteuerung vollständig entkoppelt; dadurch wird das Programm übersichtlicher und ist weniger anfällig für Programmierfehler.*

## Konstruktionsaufgabe

In den folgenden Teilaufgaben werden lediglich die LED, der Fototransistor, der Motor und der Servo-Motor benötigt.

Konfiguration der Sensoren:



## Programmieraufgaben

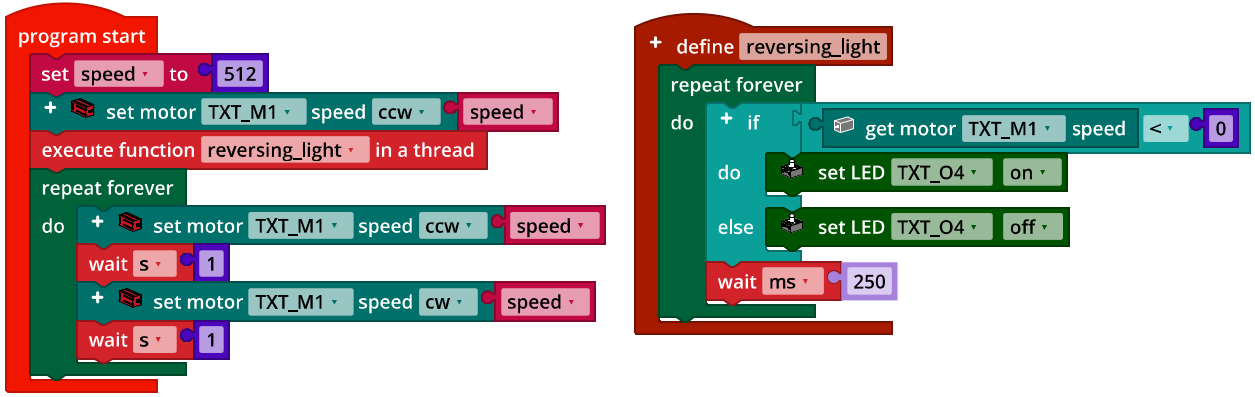
Der für die Aktivierung der LEDs jeweils maßgebliche Zustand wird direkt oder über Variablen (als Semaphore) im jeweiligen Thread abge­fragt. Das Hauptprogramm enthält in den folgenden Programmbeispielen einfache Testroutinen, mit denen die Funktion der Lichtautomatik überprüft werden kann.

Das Programm „*Automatic\_Ligthing.ft*“ enthält alle Threads für die Beleuchtung zusammengefasst.

**1. Rückfahrlicht**

Im Fall des Rückfahrlichts muss die Richtung der Motorumdrehung ausgewertet werden.

Programm (Beispiel) Rückfahrlicht mit Testprogramm:



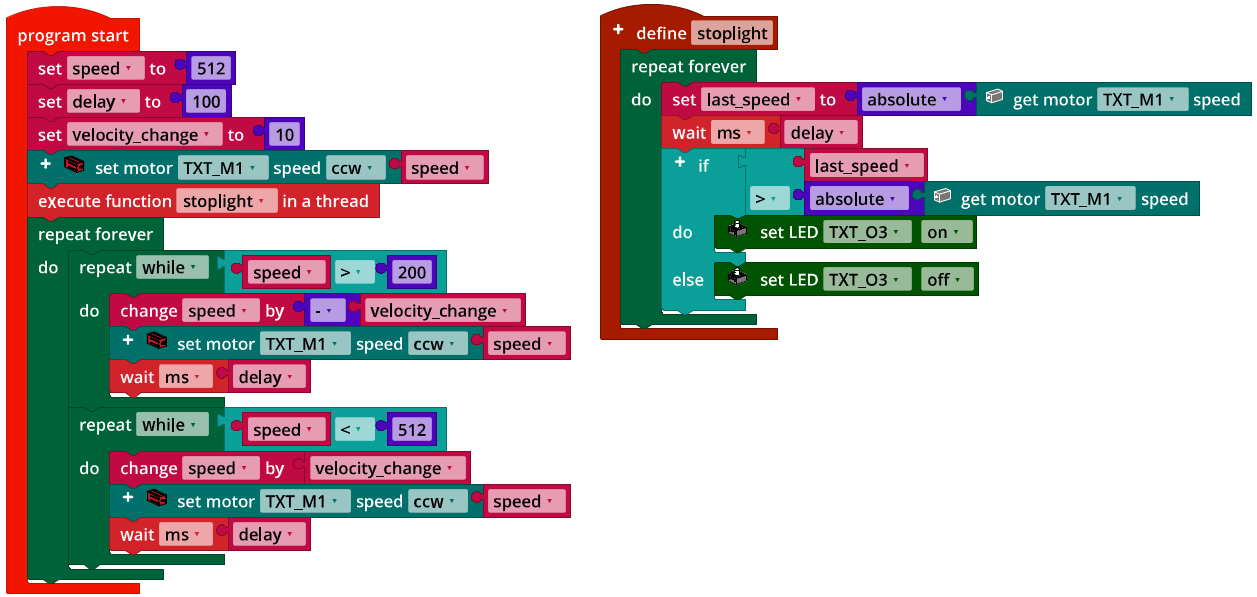
*Reversing\_Light.ft*

**2. Bremslicht**

Damit das Bremslicht leuchten kann, wenn der Motor verzögert, ist die Messung auf­einanderfolgender Geschwindigkeiten des Motors erforderlich.

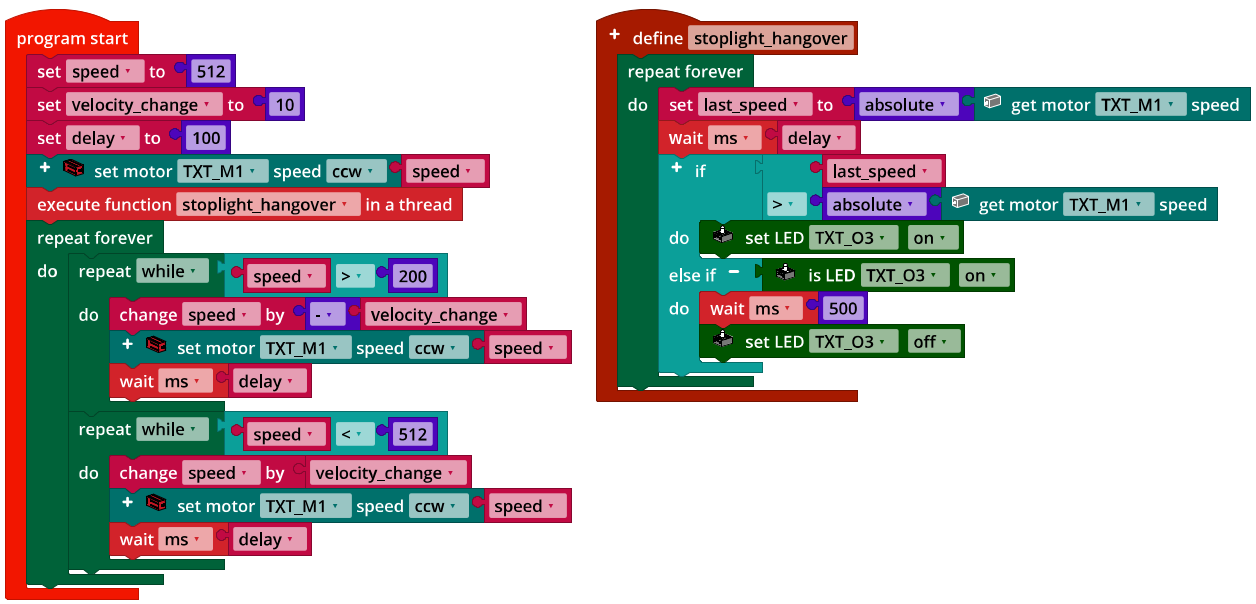
Zu beachten ist: Damit das Bremslicht auch bei einer Rückwärtsfahrt korrekt auf­leuchtet, müssen die Beträge der Geschwin­digkeiten verglichen werden, um eine Verzögerung festzustellen.

2a. Programm (Beispiel) Bremslicht mit Testprogramm:



*Stoplight.ft*

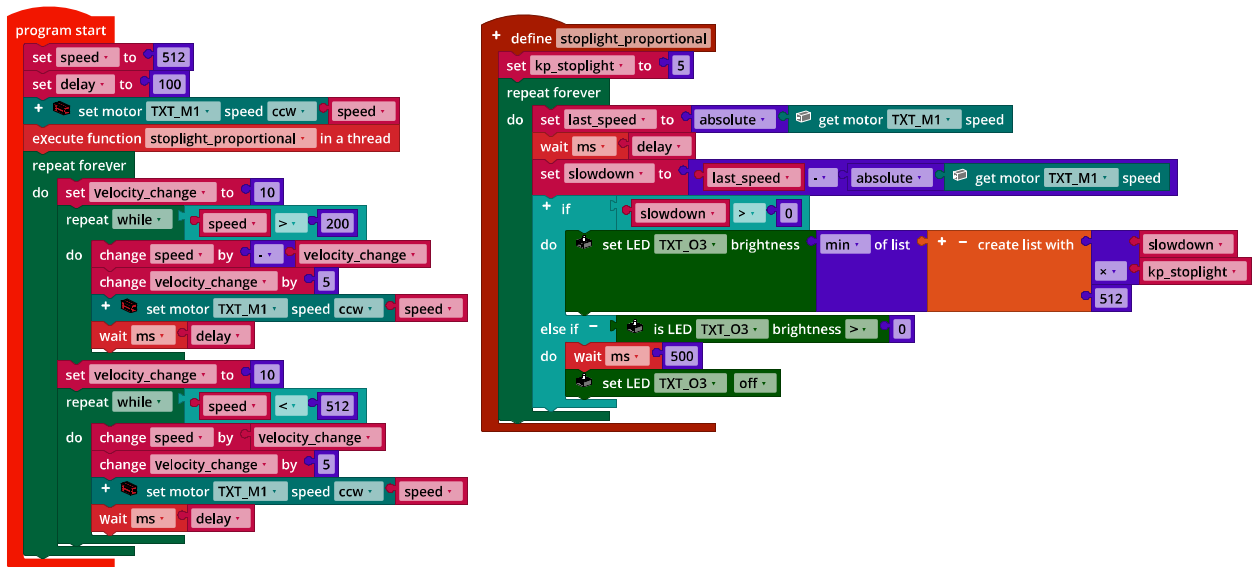
2b. Programm (Beispiel) Bremslicht mit Nachleuchten und Testprogramm:



*Stoplight\_hangover.ft*

2c. Die Messdauer im Thread „stoplight“ hat erheblichen Einfluss auf die Größe der Geschwindigkeitsänderung; mit diesem Wert muss ein wenig experimentiert werden, um eine passende Reaktion bei einem realistischen Bremsverhalten abzubilden. Mit einem Proportionalitätsfaktor () kann der Einfluss der Geschwindigkeitsänderung auf die Helligkeit festgelegt werden.

Programm (Beispiel) Bremslicht mit proportionaler Helligkeit und Testprogramm:



*Stoplight\_proportional.ft*

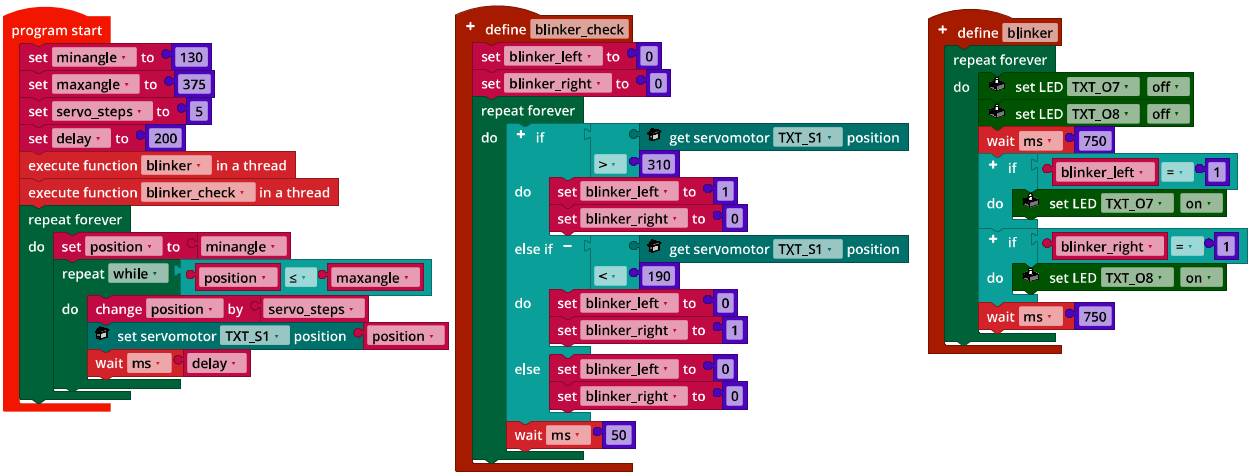
**3. Blinker**

Das Funktionsprinzip eines Blinkers ist bereits aus Aufgabe 3 des Robotics TXT 4.0 Base Set bekannt.

Für den Blinker werden mehrere Zustands­variablen benötigt: eine je Blinker, die abhängig vom Lenk­ein­schlag (also der Position des Servo-Motors) angibt, ob der rechte oder der linke Blinker aktiviert ist, und ein Zustand, der unterscheidet, ob die Blinkleuchte (falls der Blinker aktiviert ist) ein- oder ausgeschaltet ist.

Das folgende Pro­grammbeispiel löst die Aufgabe mit zwei Threads. Vorteil dieses Lösungsansatzes: Er kann sehr einfach um einen Warnblinker erweitert werden.

Programm (Beispiel) Blinker mit Testprogramm:



*Blinker.ft*

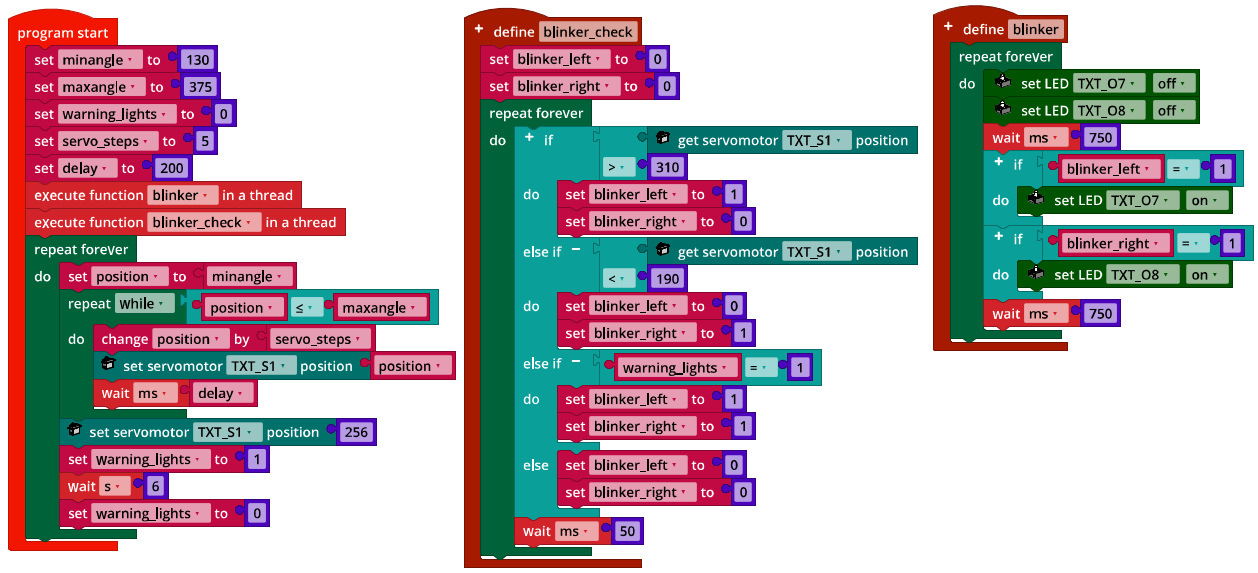
Die Status-Variablen „blinker\_left“ und „blinker\_right“ dienen als Semaphore zwischen dem Thread „blinker\_check“ und „blinker“.

## Experimentieraufgaben

**1. Warnblinker**

Der Status des Warnblinkers kann dem Thread über eine Variable als Semaphor signalisiert werden. Dann genügt es, den Blinker-Prüf-Thread zu erweitern: Wenn der Warnblinker eingeschaltet ist, werden beide Blinker aktiviert und im Blinker-Thread ein- und ausgeschaltet.

Programm (Beispiel) Warnblinker mit Testprogramm:

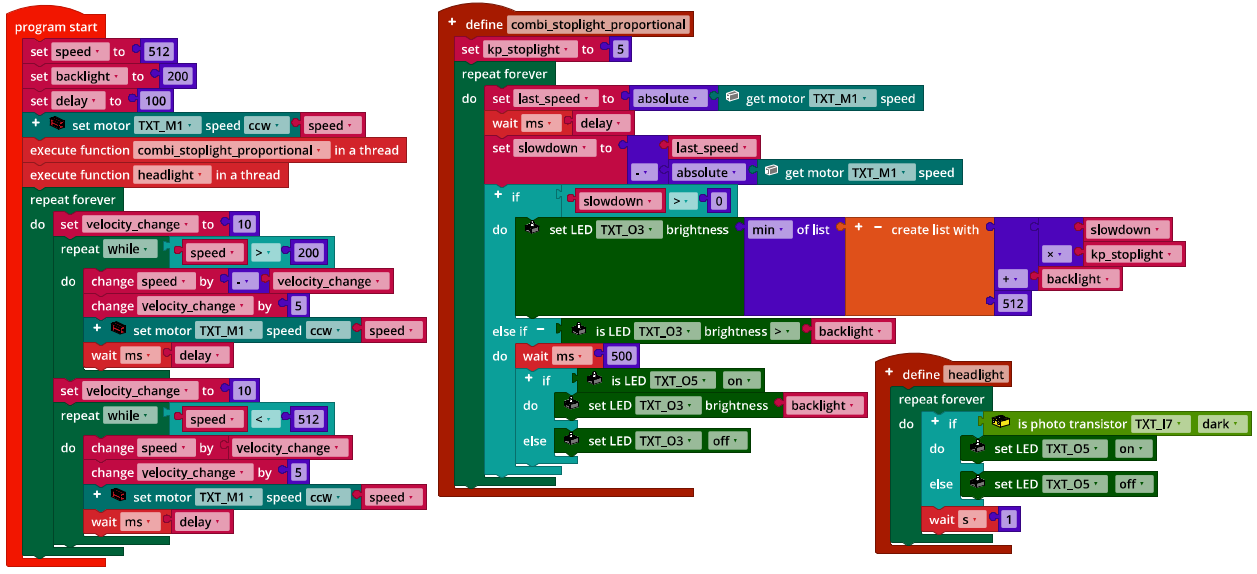


*Blinker\_with\_Warning\_Lights.ft*

**2. Abblendlicht**

Die Helligkeit des Rücklichts darf nur geändert werden, wenn die LED nicht gleich­zeitig als Bremslicht genutzt wird. Nutzt man im Thread für das Bremslicht den Status der Frontscheinwerfer als Indikator dafür, ob das Abblend­licht aktiviert ist, dann kann die Steuerung des kombinierten Rück- und Bremslichts in demselben Thread erfolgen.

Programm (Beispiel) Abblendlicht (Testprogramm):



*Headlight.ft*

Anlagen

# Aufgabe 3: Lichtautomatik

## 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: [*Nebenläufigkeit*](https://de.wikipedia.org/wiki/Nebenl%C3%A4ufigkeit).