

VII. Arduinos im Robo-Camp

Ausführliche Beschreibung eines Ablaufs zum Einsatz von Arduinos

Themenprojekt – Automatisierung (12 x 45 min.)

Kurz zum Inhalt: Das Projekt befasst sich mit der Automatisierung einer Kaffeebohnenzentrale, für die zwei Robotertypen entwickelt werden. Ein Roboter, der die Bohnen nach Qualität sortiert (Farbsortierung), und ein Roboter, der sie durch das Lager transportiert (Linienverfolgung). Die Schüler wählen aus, mit bzw. an welchem der beiden Roboter sie arbeiten möchten. Danach sollen sie sich in kleineren Gruppen relativ eigenständig mit dem Projekt beschäftigen (empfohlen werden hier Zweiergruppen).

Lernergebnisse: Die Schüler lernen die Anwendung von Inhalten aus früheren Lektionen in einem praktischen Kontext, um konkrete Probleme zu lösen – mit Bezug zu industriellen Arbeitsplätzen im 'wirklichen Leben'. Gleichzeitig erhalten sie einen tieferen Einblick in und Verständnis für die Entwicklung komplexer automatisierter Systeme sowie deren Grenzen.

Material: Im erarbeiteten Material wurde LEGO® als Plattform für die zu entwickelnden Roboter verwendet. Um die Integration der Elektronik in die LEGO-Plattform zu erleichtern, wurden gleichzeitig mehrere hiermit kompatible 3D-Modelle entwickelt. Anleitungen hierfür und für 3D-Modelle werden – wie auch schon zuvor beschriebenes Material – auf der Webseite zur Verfügung gestellt. Gleiches gilt für einen vollständigen Überblick über die im Projekt verwendeten Komponenten, Lösungsvorschläge und Vorschläge zur Weiterentwicklung der Roboter usw. Es ist wichtig zu beachten, dass die Roboter auf viele verschiedene Arten und aus vielen verschiedenen Materialien konstruiert werden könn(t)en, weshalb die entwickelten LEGO-Handbücher nur eine von mehreren Lösungen sind.

Projektkonzept:

Das folgende Projekt basiert auf einer Kaffeebohnenzentrale, in der die Bohnen zunächst nach Qualität sortiert und dann im zugehörigen Lager verteilt werden. Die Bohnen wurden zuvor in der Zentrale von Hand sortiert und anschließend von Hand im Lager verteilt. Dies erfordert anstrengende körperliche und zudem zeitaufwändige Arbeit. Die Zentrale möchte daher diese beiden Prozesse automatisieren, weshalb zwei Roboterarten gebaut und programmiert werden sollen, die jeweils die entsprechenden Funktionen ausführen können.

Sortieren von Kaffeebohnen (Farbsortierung): Die Bohnen werden nach Qualität sortiert, die an ihrer Farbe erkennbar ist. Für dieses Projekt bestehen die Bohnen aus 3D-gedruckten Bauklötzen (rot, grün, blau und gelb). Daher muss der Roboter eine Bohne entgegennehmen, sie einer Farbkategorie zuordnen, vier zugehörige Kartons so drehen, dass der entsprechende

Karton dem Förderband zugewandt ist, und schlussendlich die Bohne in diesen ablegen und die Kartons in ihre Ausgangsposition zurückdrehen können.

Transportieren (Folgen einer Linie): Die sortierten Bohnen werden in Kartons verpackt, die im Rahmen des Projekts aus 3D-gedruckten Kartons (rot, grün, blau und gelb) bestehen. Die Farbe gibt an, an welchen der vier entsprechenden Bereiche im Lager sie geliefert werden sollen. Das Lager wurde mit schwarzen Linien auf dem Boden versehen, nach denen die Roboter navigieren sollen. Zur Vermeidung von Unfällen sollen die Roboter außerdem anhalten, wenn sich vor ihnen ein Gegenstand befindet, der im Weg steht. Dies können Mitarbeiter aus Fleisch und Blut oder andere Roboter sein.

Die Zentrale ist bereits mit einem Bereich ausgestattet, in dem die Sortierung stattfindet. Außerdem haben die Mitarbeiter kürzlich die Arbeit mit dem Aufbringen der Linien abgeschlossen, auf denen die Roboter navigieren sollen, siehe Abb. 22.

Wenn die Schülerinnen und Schüler gewählt haben, mit welchem der beiden Roboter sie arbeiten möchten, können sie das in den nächsten beiden Abschnitten beschriebene Verfahren weiterverfolgen.

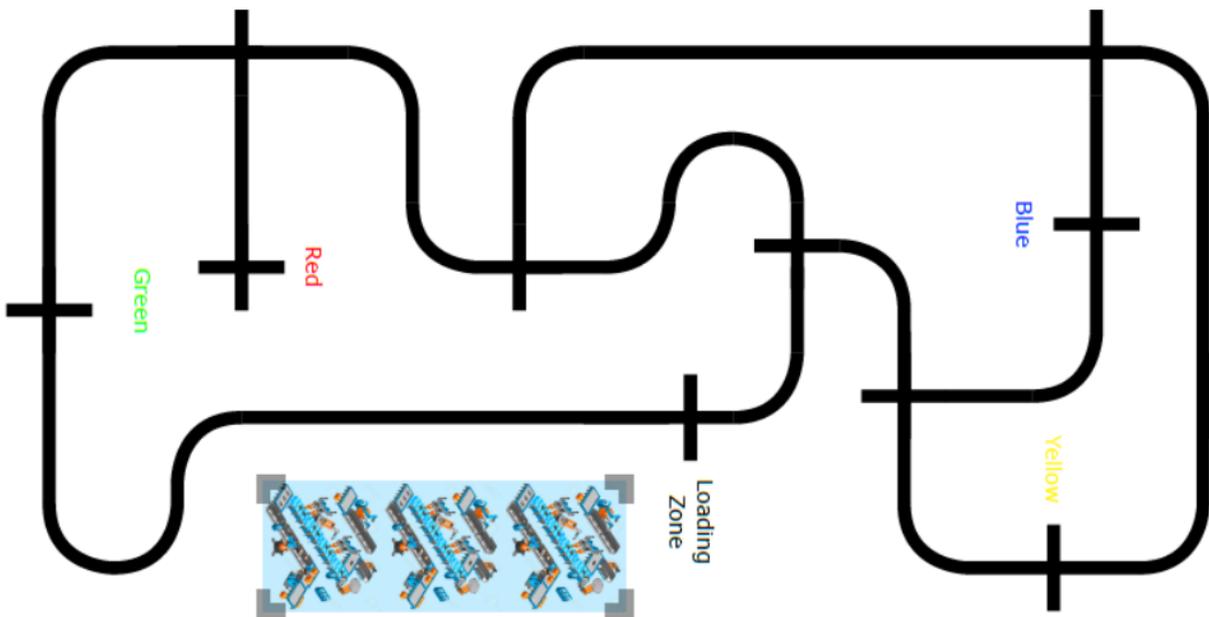


Abb. 22: Übersicht über die Ladezone und Beförderungstrecken des Lagers

Die folgenden Videos zeigen Beispiele aus der Praxis – von Robotern, die für das automatische Sortieren und Transportieren von Kaffeebohnen entwickelt wurden.

Sortieren von Kaffeebohnen: <https://www.youtube.com/watch?reload=9&v=i0nmII-bNLI>

Transportieren: <https://www.youtube.com/watch?v=WlIS3vNSuQ4>

Sortieren von Kaffeebohnen:

Der Roboter besteht aus zwei Hauptteilen, die zu einem zusammengefasst sind. Der erste Teil ist das Förderband, auf das die Kaffeebohnen zuerst gelegt werden. Dann werden sie unter einen Farbsensor verbracht, der ihre Farbe erfasst (rot, grün, blau oder gelb). Anschließend soll der Förderer sie das letzte Stück hin zu einem Karton und in diesen hinein verfrachten. Der zweite Teil ist der Mechanismus zum Drehen der entsprechenden Kartons, sodass die Bohnen, wenn sie das Ende des Förderbandes erreichen, in den richtigen Karton hineinfallen. Siehe nachstehende Abb. 23.

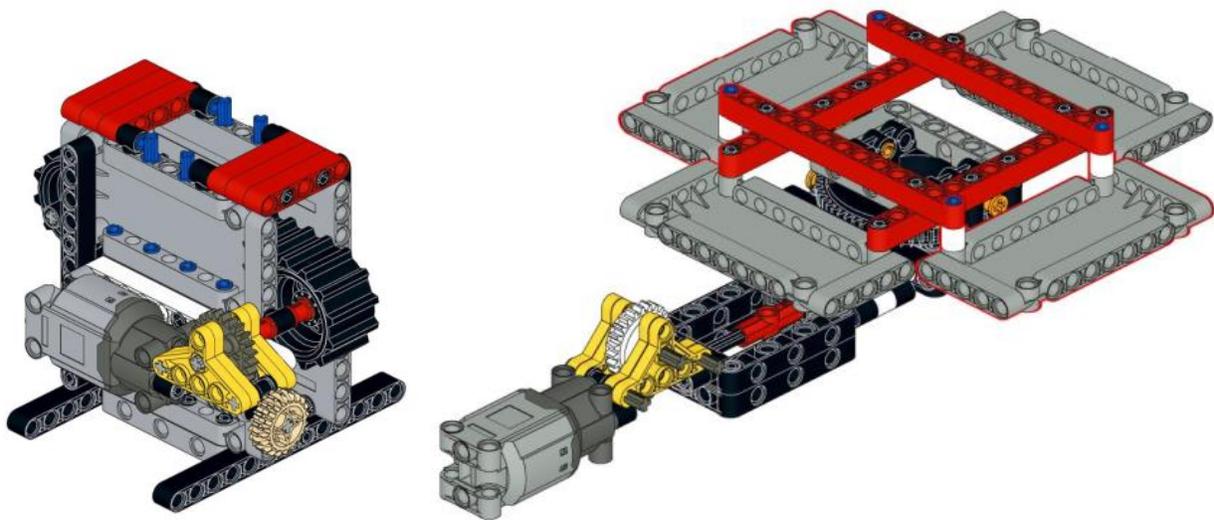


Abb. 23: LEGO-Modelle des Förderbandes und der Kartons

Erste Schritte:

1. Entwickeln des Roboters:

Um eine Arbeitsgrundlage zu haben, ist die erste Aufgabe die Konstruktion des Roboters selbst.

Aufgabe:

- Gehen Sie nach der LEGO-Anleitung "Bean Sorter – Belt" vor und bauen Sie das Förderband.
- Gehen Sie nach der LEGO-Anleitung "Bean Sorter – Rotator" vor und bauen Sie den Rotationsmechanismus für die Kartons.
- Gehen Sie nach der LEGO-Anleitung "Bean Sorter – Combine" vor und bauen Sie die beiden Teile zusammen.
- Weisen Sie nun den Kartons ihre jeweilige Farbe zu, ggf. mit einem Marker, und legen Sie fest, welche standardmäßig dem Förderband zugewandt sein soll.

Material:

- 2 x LEGO Power Functions Motor (Large).

2. Integrieren der Arduino-Plattform:

Jetzt soll der Arduino ins Spiel gebracht werden, also der Mikrocontroller zur Steuerung unseres Roboters, sowie die zugehörige Motorabschirmung, eine Erweiterung des Arduino, mit dem wir die Motoren steuern, und zusätzlich ein Steckbrett.

Aufgabe:

- Gehen Sie nach der Anleitung "Bean Sorter – Integrate the Arduino platform" vor und fügen Sie Arduino, Motorabschirmung, Steckbrett und Akkusatz hinzu.

Material:

- Arduino Uno
- Motorabschirmung R3
- Steckbrett (halbe Größe)
- Akkusatz + Kabel
- Arduinomodul (3D-gedruckt)
- Steckbrettmodul (3D-gedruckt)
- Akkusatzmodul (3D-gedruckt)
- 2 x Stecker/Stecker-Kabel

3. Test der Motoren:

Im folgenden Schritt sollen die Motoren angeschlossen und gleichzeitig einige kleinere Tests durchgeführt werden, um sicherzustellen, dass sie wie vorgesehen funktionieren.

Aufgabe:

- Motoren mit der Motorabschirmung verbinden.
- Förderband 2 Sek. lang vorwärts laufen lassen – mit Höchstgeschwindigkeit.
- Förderband 2 Sek. lang rückwärts laufen lassen – mit halber Geschwindigkeit.
- Kartons 2 Sek. lang im Uhrzeigersinn drehen lassen – mit Höchstgeschwindigkeit.
- Kartons 2 Sek. lang gegen den Uhrzeigersinn drehen lassen – mit halber Geschwindigkeit.

Cheat sheets:

- Aktoren – DC-Motor
- Code – pinMode
- Code – digitalWrite
- Code – analogWrite
- Code – delay

Förderband:

Für diesen Teil empfiehlt sich das Erstellen einer neuen Programmdatei – aber speichern Sie unbedingt die vorherigen aus den *Ersten Schritten*.

1. Einsatz von Funktionen zur Steuerung des Förderbandes:

Funktionen sind eine gute Methode, um einen Codeabschnitt abzugrenzen, damit sie ihn immer wieder verwenden können, ohne dass wir den Code jedes Mal neu schreiben müssen. Gleichzeitig macht es den Code für den/die Anwender*in besser lesbar.

Aufgabe:

- Erstellen Sie eine Funktion "startBelt", die das Förderband startet.
- Erstellen Sie eine Funktion "stopBelt", die das Förderband anhält.
- Erstellen Sie eine Funktion "deliverBean", die die Bohne 'anliefert' (das Förderband startet und soll 5 Sek. lang laufen bis es wieder stoppt).
- Überprüfen Sie, dass alle Funktionen einwandfrei arbeiten.

Cheat sheets:

- Code – Functions

2. Integrieren des Farbsensors:

Vor dem Einsatz des Förderbandes müssen wir den Farbsensor integrieren und prüfen, ob er bestimmungsgemäß funktioniert.

Aufgabe:

- Gehen Sie nach der Anleitung "Bean Sorter – Integrate Color-Sensor" vor und integrieren Sie den Farbsensor.
- Legen Sie eine 'Bohne' auf das Förderband unter den Farbsensor und geben Sie am Bildschirm ein, um welche Farbe es sich handelt. Führen Sie dies für alle Farben aus (Rot, Grün, Blau und Gelb).

Material:

- Farbsensor (TCS3200)
- Farbsensormodul (3D-gedruckt)
- 7 x Stecker/Buchse-Kabel

Cheat sheets:

- Sensors – Color-Sensor (TCS3200)
- Code – Variables
- Code – Serial Print

3. Auf erkannte Farben reagieren:

Wir können jetzt unser Förderband steuern und den Farbsensor einsetzen, um Farben zu erkennen. Bis dahin macht das Programm aber immer nur das Gleiche. Daher möchten wir nun eine Struktur für das Programm erstellen, das es ermöglicht verschiedene Aktionen auszuführen, je nachdem, welche Farbe der Farbsensor erkennt (Rot, Grün, Blau oder Gelb).

Aufgabe:

- Starten Sie das Förderband.

- Wenn der Farbsensor "red" liest:
 - "red" eingeben.
 - Förderband anhalten und die 'Bohne' abliefern.
- Überprüfen, dass es funktioniert – wenn ja, einen "if else" für jede der anderen Farben herstellen (Grün, Blau und Gelb), der die gleiche Funktionalität wie die der Farbe Rot enthält.

Cheat sheets:

- Code – Conditionals.

Die Kartons:

Für diesen Teil empfiehlt sich das Erstellen einer neuen Programmdatei – aber speichern Sie die vorherige zum Förderband.

1. Funktionen zum Rotieren der Kartons einsetzen:

Wie beim Förderband möchten wir auch hier Funktionen zum Abgrenzen des Codes für die Motorsteuerung einsetzen, damit er wiederverwendet werden kann.

Aufgabe:

- Erstellen Sie eine Funktion "startRotation", die die Kartons in eine Drehung gegen den Uhrzeigersinn versetzt.
- Erstellen Sie eine Funktion "stopRotation", die die Drehung der Kartons anhält.
- Erstellen Sie eine Funktion "rotatePosition", die einen int als Parameter nimmt (nennen Sie den Parameter "position") – abgesehen vom Ausgeben des Parameters an das Programm bleibt diese Funktion ansonsten leer.
- Überprüfen Sie, dass die Funktionen einwandfrei arbeiten.

Cheat Sheet:

- Code – Functions (mit Parametern).

2. Schalter integrieren:

Bisher konnten wir die Kartons nur zeitabhängig hin und her rotieren, aber da die Zeit, die zum Rotieren einer Runde benötigt wird, davon abhängt, wie viel Energie noch in den Batterien vorhanden ist oder wie viel Schmutz ggf. in das Getriebe gelangt ist usw., ist dies keine optimale Lösung. Daher wollen wir einen Schalter in unseren Aufbau integrieren, mit dem wir jedes Mal die Rotation der Kartons um eine Position erkennen.

Aufgabe:

- Gehen Sie nach der Anleitung "Bean Sorter – Integrate Switch" vor und integrieren Sie den Schalter.
- Überprüfen Sie durch Anzeigen seines Zustandes an das Programm, dass er funktioniert.

Material:

- Schalter

- Schaltermodul (3D-gedruckt)
- Widerstand 1 K Ω
- 3 x Stecker/Stecker-Kabel.

Cheat Sheet:

- Sensors – Switch.

3. Anzahl der Kartonrotationen überprüfen:

Da der Schalter bei jedem Passieren eines Kartons automatisch gedrückt wird, können wir damit prüfen, um wie viele Positionen sich die Kartons bewegt haben, indem wir die Anzahl des Drückens registrieren. Auf diese Weise können wir immer sicherstellen, dass die Kartons sich um genau die Anzahl der gewünschten Positionen verschieben. Hierzu müssen wir die Funktion "DeliverBean" abgeschlossen haben, damit sie die gewünschte Anzahl an Rotationen ausführt.

Aufgabe:

- In der Funktion "deliverBean":
 - Rotation der Kartons starten.
 - Neue Zählervariable namens "i" erstellen und ihr den Wert 0 zuordnen.
 - Eine while-Schleife mit der Bedingung ($i < \text{position}$) erstellen.
 - Wenn Schalter gedrückt ist:
 - "i" um 1 erhöhen.
 - Ca. 200 ms warten (je nach Rotationsgeschwindigkeit).
 - Kartonrotation anhalten.
- Funktion durch Aufrufen mit verschiedenen Parametern (1, 2, 3 usw.) überprüfen.
- Zählen Sie nun, wie oft die Kartons um eine Position rotieren müssen, bevor sie sich vor dem Förderband befinden, und wie oft sie rotieren müssen, bevor sie wieder an ihrem Ausgangspunkt sind. Schreiben Sie diese Zahlen auf ein Blatt Papier.

Förderband und Kartons kombinieren:

Für diesen Teil empfiehlt sich das Erstellen einer neuen Programmdatei – aber speichern Sie wieder unbedingt die vorherigen Funktionen.

1. Kombinieren des gesamten Aufbaus:

Jetzt ist es Zeit, das Förderband mit den Kartons zu kombinieren, damit der Roboter vollautomatisch funktioniert. Wir haben bereits fast alle dafür erforderlichen Arbeiten durchgeführt, insbesondere durch den Einsatz von Funktionen.

Aufgabe:

- Förderband starten.
- Wenn der Farbsensor "red" liest:
 - "red" an den Bildschirm ausgeben.
 - Förderband anhalten.

- Kartons so rotieren lassen, dass der rote vor dem Förderband steht.
- 'Bohne' abliefern.
- Kartons so rotieren lassen, dass der rote erneut an seinem Ausgangspunkt steht.
- Überprüfen Sie, dass dies alles funktioniert – und erweitern Sie dann das Programm um die übrigen Farben (Grün, Blau und Gelb).

Varianten der Aufgabe:

Sie können die Aufgabe weiter variieren bzw. weiterentwickeln und so die Funktionen des Aufbaus erweitern, z. B. verschiedene LEDs zum Leuchten bringen, um anzuzeigen, welche Aktionen der Roboter gerade ausführt. Eine weitere Möglichkeit wäre, einen Halter zu bauen, der mit Hilfe eines Servomotors nacheinander neue Bohnen auf das Förderband legt.

Transportieren:

Der Roboter ruht auf zwei Vorderrädern (die ihn antreiben) und einem Kugelrad. Siehe nachstehende Abb. 24.

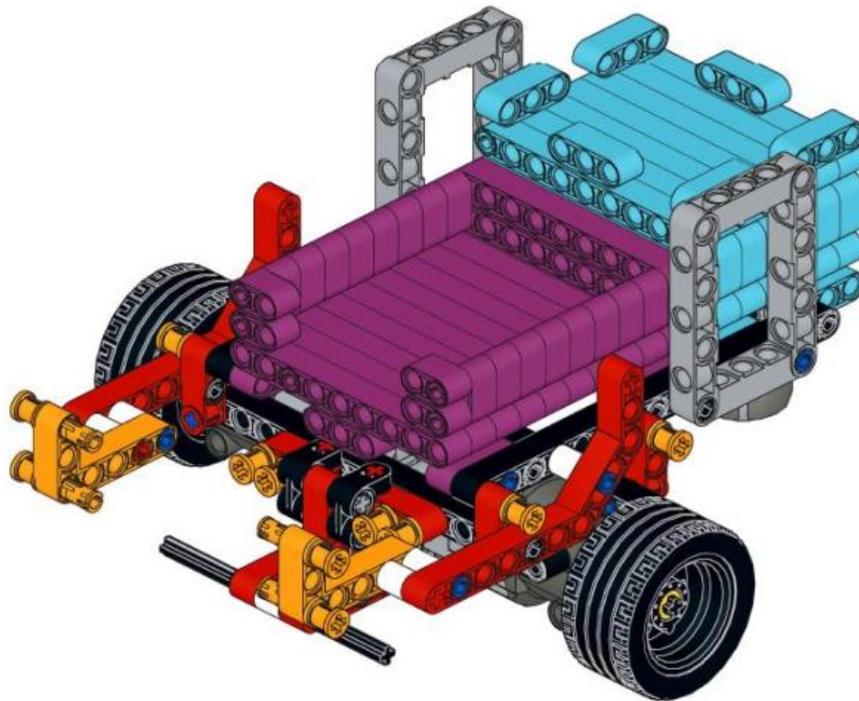


Abb. 24: LEGO-Modell des Förderroboters

Erste Schritte:

1. Entwickeln des Roboters:

Die erste Aufgabe ist die Konstruktion des Roboters selbst, um eine Arbeitsgrundlage zu haben.

Aufgabe:

- Gehen Sie nach der LEGO-Anleitung "Transporter – Transporter" vor und bauen Sie das Förderfahrzeug (Transporter).

Material:

- LEGO + 2 x LEGO Power Functions Motor (L).

2. Integrieren der Arduino-Plattform:

Jetzt soll der Arduino angewendet werden, also der Mikrocontroller zur Steuerung unseres Roboters, sowie die zugehörige Motorabschirmung, eine Erweiterung des Arduino, mit dem wir die Motoren steuern, sowie ein Steckbrett.

Aufgabe:

- Gehen Sie nach der Anleitung "Transporter – Integrate the Arduino platform" vor und fügen Sie Arduino, Motorabschirmung, Steckbrett und Akkusatz hinzu.

Material:

- Arduino Uno
- Motorabschirmung R3
- Steckbrett (halbe Größe)
- Akkusatz + Kabel
- Arduinomodul (3D-gedruckt)
- Steckbrettmodul (3D-gedruckt)
- Akkusatzmodul (3D-gedruckt)
- 2 x Stecker/Stecker-Kabel.

3. Test der Motoren:

Jetzt ist es Zeit, die Motoren anzuschließen und gleichzeitig einige kleinere Tests durchzuführen, um sicherzustellen, dass alles wie gedacht funktioniert.

Aufgabe:

- Motoren mit der Motorabschirmung verbinden.
- Den Transporter 2 Sek. lang vorwärts laufen lassen – mit Höchstgeschwindigkeit.
- Den Transporter 2 Sek. lang rückwärts laufen lassen – mit halber Geschwindigkeit.
- Den Transporter 2 Sek. lang nach links drehen lassen – mit Höchstgeschwindigkeit.
- Den Transporter 2 Sek. lang nach rechts drehen lassen – mit halber Geschwindigkeit.

Cheat sheets:

- Aktoren – DC-Motor
- Code – pinMode

- Code – digitalWrite
- Code – analogWrite
- Code – delay

Nach Linien navigieren:

Für diesen Teil empfiehlt sich das Erstellen einer neuen Programmdatei nachdem die vorherigen Schritte gespeichert wurden.

1. Funktionen zum Steuern des Transporters einsetzen:

Funktionen sind eine gute Methode, um einen Codeabschnitt abzugrenzen, damit sie immer wieder verwendet werden können, ohne dass wir den Code jedes Mal neu schreiben müssen. Gleichzeitig macht es den Code für den/die Anwender*in besser lesbar.

Aufgabe:

- Erstellen Sie eine Funktion "driveForward", die den Roboter vorwärts fahren lässt.
- Erstellen Sie eine Funktion "crossLine", die den Roboter ca. 5 cm vorwärts fahren und ihn dann wieder anhalten lässt.
- Erstellen Sie eine Funktion "turnLeft", die den Roboter eine viertel Drehung nach links ausführen lässt.
- Erstellen Sie eine Funktion "turnRight", die den Roboter eine viertel Drehung nach rechts ausführen lässt.
- Erstellen Sie eine Funktion "uTurn", die den Roboter eine halbe Drehung ausführen lässt – Richtung egal.
- Erstellen Sie eine Funktion "stopRobot", die den Roboter für 1 Minute anhalten lässt.
- Überprüfen Sie, dass alle sechs Funktionen einwandfrei arbeiten.

Cheat sheets:

- Code – Functions.

2. Integrieren der IR-Sensoren:

Bevor wir den Roboter zum Folgen der Bodenlinien bringen können, müssen wir die beiden IR-Sensoren einbauen und überprüfen, ob sie richtig funktionieren.

Aufgabe:

- Gehen Sie nach der Anleitung "Transporter – Integrierte IR-Sensoren" vor und integrieren Sie die IR-Sensoren.
- Zunächst den Wert des linken IR-Sensors ablesen und ans Programm weitergeben; danach die Werte für Weiß und Schwarz notieren. Den Vorgang für den rechten IR-Sensor wiederholen.

Material:

- IR-Sensor (QRE1113, analog).
- IR-Sensormodul (3D-gedruckt)
- 6 x Stecker/Stecker-Kabel

Cheat sheets (Spickzettel):

- Sensors – IR-sensor (QRE1113, analog).
- Code – Variables.
- Code – analogRead.
- Code – Serial Print.

3. Folgen der Linie:

Jetzt können wir den Roboter mithilfe der IR-Sensoren so steuern, dass er einer Linie folgt. Dazu müssen wir für jeden IR-Sensor den Wert berechnen, der genau zwischen dem Wert für Weiß und dem Wert für Schwarz liegt. Wir nennen diesen Wert den Schwellenwert. Um einer Linie zu folgen, lassen wir die Motoren laufen, wenn ihr jeweiliger IR-Sensor auf Weiß sieht, andernfalls müssen sie anhalten. Auf diese Weise korrigiert der Roboter automatisch, wenn er droht vom Kurs abzukommen und dabei die Linie zu überfahren. Auf die gleiche Weise stoppt er auch automatisch, wenn er an eine Kreuzung kommt, weil beide Sensoren jetzt Schwarz registrieren. Beginnen Sie mit der Simulation, indem Sie den Roboter von Hand steuern.

Aufgabe:

- Den linken Motor zum Laufen veranlassen, wenn der linke IR-Sensor weniger als den Schwellenwert liest; sonst soll er anhalten.
- Den rechten Motor zum Laufen veranlassen, wenn der rechte IR-Sensor weniger als den Schwellenwert liest; sonst soll er anhalten.
- Überprüfen Sie, dass der Roboter jetzt der Linie zu folgen vermag – und anhält, wenn er eine Kreuzung erreicht.

Cheat sheets:

- Code – Conditionals.

4. Folgen der Linie – als Funktion:

Bisher stoppt unser Roboter automatisch, wenn er eine Kreuzung erreicht, aber er fährt auch von selbst wieder an, wenn man ihn manuell über die Kreuzung hinauschiebt oder auf eine neue Linie stellt. Wenn wir also unsere in den Funktionen "turnLeft" und "turnRight" gespeicherten Abbiegebefehle anwenden möchten, wenn wir eine Kreuzung erreichen, gibt es ein Problem. Versuchen Sie mal dies mit dem Befehl auszuführen – der Roboter wird sich jetzt einfach nur drehen und der Linie überhaupt nicht mehr folgen. Das liegt daran, dass die Liniencodes, nach denen er der Linie folgt, im Bruchteil einer Sekunde ausgeführt werden, woraufhin der Befehl zum Drehen erneut aufgerufen wird. Wir müssen daher eine eigenständige Funktion erstellen, die bewirkt, dass der Roboter der Linie folgt, bis er eine Kreuzung erreicht, wonach die Funktion endet.

Aufgabe:

- Erstellen Sie eine Funktion "followLine".
 - Verwenden Sie eine while-Schleife mit der Bedingung (true).

- Den linken Motor zum Laufen veranlassen, wenn der linke IR-Sensor weniger als den Schwellenwert liest; sonst soll er anhalten.
- Den rechten Motor zum Laufen veranlassen, wenn der rechte IR-Sensor weniger als den Schwellenwert liest; sonst soll er anhalten.
- Wenn beide IR-Sensoren weniger als den Schwellenwert lesen, halten Sie beide Motoren an und unterbrechen Sie die while-Schleife.
- Überprüfen Sie, dass die Funktion einwandfrei arbeitet.
- Überprüfen Sie sodann, ob sie zusammen mit den Drehfunktionen funktioniert – und zudem die Funktion des Geradeausfahrens an einer Kreuzung –, indem Sie die untenstehenden Funktionen aufrufen und daraufhin beobachten, ob der Roboter die vorgesehene Route fährt. Denken Sie daran, dass nach Ablauf einer Minute die Funktion "stopRobot" zu Ende ist, und der Ablauf (die Sequenz) von vorn beginnt.
 - followLine
 - turnLeft
 - followLine
 - crossLine
 - followLine
 - turnRight
 - followLine
 - uTurn
 - stopRobot

Cheat sheets:

- Code – While loop (mit Unterbrechung).

Nach Umfeld navigieren:

Bevor Sie fortfahren, empfiehlt es sich das erstellte Programm zu speichern und eine Kopie zu erstellen, um anschließend in der Kopie weiterzuarbeiten.

1. Integrieren des Abstandssensors:

Der Roboter kann jetzt gemäß den Linien auf dem Boden des Lagers navigieren, aber was ist, wenn sich ein anderer Roboter vor ihm befindet oder ein Mitarbeiter übersehen hat, dass er direkt auf ihn zusteuert? Um mögliche Probleme und Verletzungen zu vermeiden, bringen wir jetzt einen Abstandssensor ins Spiel, mit dem der Roboter die Entfernung zum nächsten Objekt messen kann. Dadurch können wir im nächsten Schritt den Roboter so programmieren, dass er anhält, falls ein Objekt ihm (oder er dem Objekt) zu nahe gekommen ist.

Aufgabe:

- Gehen Sie nach der Anleitung "Transporter – Ultrasound-Sensor" vor und integrieren Sie den Abstandssensor.
- Geben Sie die Entfernung zum nächsten Objekt an den Bildschirm aus.

Material:

- Abstandssensor (HC-RS04).
- Abstandssensormodul (3D-gedruckt)

- 4 x Stecker/Stecker-Kabel.

Cheat sheets:

- Sensors – Ultrasonic-Sensor (HC-RS04).

2. Beim einem im Weg stehenden Objekt Roboter anhalten:

Unsere Funktion "followLine" wollen wir nun so aufwerten, das der Roboter stoppt, wenn der Abstand zu einem Objekt vor ihm 20 cm unterschreitet.

Aufgabe:

- Funktion "followLine" wie folgt aufwerten:
 - Zuerst wird in der while-Schleife der Abstand zum nächsten Objekt gelesen.
 - Werten Sie die Bedingungen für erlaubtes Motorlaufen so auf, dass die Bedingung jetzt Folgendes vorsieht:
 - Wenn linker IR-Sensor unter Schwellenwert und Abstand über 20 cm, dann linken Motor laufen lassen, sonst anhalten.
 - Wenn rechter IR-Sensor unter Schwellenwert und Abstand über 20 cm, dann rechten Motor laufen lassen, sonst anhalten.

Karton mit Bohnen 'anliefern':

Bevor Sie fortfahren, empfiehlt es sich auch hier das erstellte Programm zu speichern und eine Kopie zu erstellen, um anschließend in der Kopie weiterzuarbeiten.

1. Integrieren des Servomotors:

Mitarbeiter können jetzt einen Karton auf dem Roboter platzieren und ihn so programmieren, dass er einer Route folgt – aber er kann den Karton bei Ankunft noch nicht selbst abliefern. Wir wollen ihn daher mit einem Servomotor ausstatten, der die Kartons von der Ladefläche kippen kann, woraufhin der Roboter zurückfährt, um einen neuen Karton abzuholen.

Aufgabe:

- Gehen Sie nach der Anleitung "Transporter – Servo-Motor" vor und integrieren Sie den Servomotor.
- Überprüfen Sie den Servomotoren auf ordnungsgemäße Funktionalität, indem Sie ihn in verschiedene Positionen versetzen.

Material:

- Servomotor.
- 6 x Stecker/Stecker-Kabel.

Cheat sheets:

- Effectors – Servo-Motor.

2. Karton 'anliefern':

Jetzt muss der Roboter nur noch den Karton freigeben, wenn er den vorgesehenen Bestimmungsort erreicht hat.

Aufgabe:

- Erstellen Sie eine Funktion "deliverBox", die mittels des Servomotors die Ladefläche in Kippposition bringt, damit der Karton herunterfällt, woraufhin der Servomotor in seine Ausgangsposition zurückkehrt.
- Überprüfen Sie, dass es entsprechend funktioniert und der Roboter nunmehr einem Streckenverlauf folgen kann, auf dem er am richtigen Ort den beförderten Karton ablädt.

Varianten der Aufgabe:

Sie können sich gern eigene Ideen für mögliche Erweiterungen ausdenken, z. B. verschiedene LEDs, die anzeigen, welche Aktionen der Roboter derzeit ausführt. Oder der Einsatz eines LDRs (Light Dependent Resistor) zwecks Feststellung, ob sich ein Karton auf der Ladefläche des Roboters befindet.