



SENSE Roboter





SENSE Roboter

© Alle Rechte vorbehalten SES Ltd.

Das Material in diesem Buch darf nicht kopiert, dupliziert, gedruckt, übersetzt, herausgegeben oder verbreitet werden ohne vorherige schriftliche Genehmigung von SES.

Inhaltsverzeichnis

Kapitel 1 – Kontrolle und Roboter	1
1.1 Roboter	1
1.2 Kontrollsysteme	2
1.3 Der SENSE-Roboter	3
1.4 RobocklySense	4
1.5 RobocklySense Installation	4
1.6 RobocklySense-Programm starten	5
Experiment 1.1 – Direkter Modus	6
1.1.1 SENSE-Bewegung	8
1.1.2 Die SENSE-Sensoren	9
Experiment 1.2 – Erste Programme	11
1.2.1 Erstes Programm – vorwärts	13
1.2.2 Programm Download	16
1.2.3 Vorwärts und rückwärts	17
1.2.4 Drehen nach links und nach rechts	18
1.2.5 Übungen – Bewegung in einem Quadrat	20
Experiment 1.3 – Interaktive Programme	21
1.3.1 Speicher und Variablen	22
1.3.2 Wait-until-Befehl	25
1.3.3 Endlos-Schleufe	26
1.3.4 Bewegung zwischen zwei Linien	27
1.3.5 Übung – Zwischen einer Wand und einer Linie (I)	27
Experiment 1.4 – Abläufe als neue Befehle	28
1.4.1 Programme und Abläufe	30
1.4.2 Definitionen	32
1.4.3 Übungen – Zwischen einer Wand und einer Linie (II)	33
1.4.4 Bewegung entlang einer schwarzen Linie	34
1.4.5 Übung – Entlang einer komplexen schwarzen Linie	35

Experiment 1.5 – Bedingungen und Entscheidungen	36
1.5.1 If-then-Befehl	36
1.5.2 OFF und ON mit verschiedenen Werten	41
1.5.3 AND-Bedingung	42
1.5.4 OR-Bedingung	44
1.5.5 Übung – Entlang von zwei Linien	45
1.5.6 Bewegung entlang einer Wand	45
1.5.7 Übungen – Entlang von Wänden	47
Kapitel 2 – Brain Units	48
2.1 Brain units	48
2.2 NeuLog-Sensoren als Brain-Units	49
2.3 ID der Brain-Unit ändern	50
Experiment 2.1 – Geräuschsensor	51
2.1.1 Übung – Warten auf ein Geräusch	53
Experiment 2.2 – Bewegungssensor	54
2.2.1 Übung – Bewegung in einem Distanzbereich	57
Experiment 2.3 – Sucher-Einheit	58
2.3.1 IR-Geber	58
2.3.2 Sucher-Einheit	58
2.3.3 Übung – Mit einem Roboter einem IR-Geber folgen	61
Kapitel 3 – Übungen zu selbstfahrenden Fahrzeugen	62
3.1. Selbstfahrende Fahrzeuge	62
3.2 Programmiersprachen	62
Übung 3.1 – Entlang schwarzer Linien	63
Übung 3.2 – Automatic guided Vehicle (AGV)	63
Übung 3.3 – AGV zwischen Positionen	64
Übung 3.4 – Entlang einem Gebäudeblock	64
Übung 3.5 – Entlang einem Gebäudeblock und um einen Wagen herum ..	65
Übung 3.6 – Entlang einem Gebäudeblock mit einem Stoppzeichen	65
Übung 3.7 – Entlang einem Gebäudeblock mit einem Stopp für einen Fussgänger	66
Übung 3.8 – Wächter beim Gebäudeblock	67
Übung 3.9 – Zwei Wächter beim Gebäudeblock	67
Übung 3.10 – Taxifahrer	68
Übung 3.11 – Taxifahrer mit Passagier	68

Kapitel 1 – Kontrolle und Roboter

1.1 Roboter

Die heutige Welt ist eine Welt der eingebetteten Computersysteme. Wir finden diese in Media-Systemen, Uhren, Telefonen, Fernbedienungen, Autos und vielem mehr. Noch vor wenigen Jahren konnten wir uns nichts vorstellen unter Begriffen wie «tragbare Computer» oder das «Internet von Gegenständen».

Jeden Tag erscheint ein neues überraschendes Produkt oder eine neue Anwendung und schon Monate später können wir uns nicht mehr vorstellen, wie wir ohne diese gelebt haben.

Die Roboter-Systeme, die in Computersystemen eingebettet sind, führen unabhängige Aktivitäten aus wie Suche, Manipulation, Identifikation, Aktivierung, Schutz usw.

Viele Systeme kombinieren eine gewisse Art von künstlicher Intelligenz beim Funktionieren und Kommunizieren zwischen Maschinen.

Jedes Roboter-System beinhaltet einen Controller, der erlaubt, dass das System mit verschiedenen Programmen läuft. Der Programmentwickler schreibt diese Programme auf dem Computer und überträgt diese auf den Controller.

Das Roboter-System beinhaltet den Controller, Bausteine, Räder, Steuerungen, Motoren, Sensoren und mehr.

Ein Roboter-System zu bauen, bildet eine Herausforderung, um Kenntnisse in den verschiedenen Gebieten der Technik zu erlangen (Elektronik, Computer, Mechanik, Elektrizität usw.)

Es gibt viele verschiedene Arten von Robotern: Armroboter, mobile Roboter, Gehroboter und mehr.

Die SENSE-Roboter sind Roboter und «Hirn-Einheiten» für den Bildungsbereich. Damit lassen sich Programme bauen für viele Anwendungen von Robotern.

Die SENSE-Roboter sind autonom und erlauben uns, verschiedene Anwendungen und Funktionen zu programmieren. Jeder Roboter bietet zusätzliche Programmier-Herausforderungen wie Bewegung auf einer Linie, entlang einer Wand, suchen, anhalten, gehen usw.

1.2 Kontrollsysteme

Ein Roboter ist ein computerbasiertes Kontrollsystem

Ein Kontrollsystem kann definiert werden als eine Gruppe von Komponenten, die zusammen verschiedene Variablen kontrollieren und so das Verhalten des Systems kontrollieren.

Beispiele:

- Eine Klimaanlage kontrolliert die Raumtemperatur.
- Ein Kontrollsystem für ein Gewächshaus kontrolliert die Temperatur, die Luftfeuchtigkeit, das Licht und die Bewässerung.
- Ein Geschwindigkeits-Kontrollsystem unterhält eine konstante Motor-
geschwindigkeit, unabhängig von der Drehzahl des Motors.
- Ein Lichtsystem kann eine konstante Helligkeit garantieren, unabhängig von der Stärke des vorhandenen Sonnenlichts. Das System schaltet Lampen ein und aus je nach verlangter Lichtmenge.

Drei Grundeinheiten gibt es in jedem computerbasierten Kontrollsystem:

1. **Eingangseinheit:** Diese Einheit liest die Sensoren des Systems wie Temperatur, Licht, Abstand, Berührungsschalter usw. und gibt diese Informationen an die Kontrolleinheit weiter.
2. **Kontrolleinheit:** Das «Hirn» des Kontrollsystems enthält in seinem Speicher das Systemprogramm. Dieses führt die Programmbefehle aus und verarbeitet die erhaltenenen Daten.
3. **Ausgangseinheit:** Diese Einheit steuert die Komponenten wie Motoren, Lampen, Pumpen oder Ventilatoren gemäss der gemessenen Werte der Sensoren und der «Entscheidungen» des Programmes.



Die Kontrolleinheit ist verbunden mit einem Computer für die Programmierung und um ein Programm vom Computer in den Flash-Speicher der Kontrolleinheit herunterzuladen.

Wenn die Kontrolleinheit vom Computer getrennt und mit einer Stromquelle oder einer Batterie verbunden wird, erhält man ein unabhängiges System.

1.3 Der SENSE-Roboter



Der SENSE-Roboter ist ein Roboter für Anwendungen wie:

- Bewegung entlang einer schwarzen oder weissen Linie
- Bewegung entlang von Wänden oder in einem Labyrinth
- Selbstfahrender Wagen: Bewegung in einem Labyrinth, wenn sich andere Wagen ebenfalls dort bewegen.
- Einem bewegten Körper folgen mit einem Infrarot-Sender und einem Spur-Modul
- Umgebungsbeobachtungen und Messungen mit den NeuLog-Sensoren

Der SENSE beinhaltet folgende Komponenten:

- Grundeinheit
- 3 Steckplätze für NeuLog-Sensoren oder «Brain-Unit»
- 5 Infrarot (IR)-Sensoren
- 1 Liniensensor
- Drehrad
- 2 Motoren mit Rädern
- 1 Controller, der die Sensoren, die Motoren und die unabhängige Bewegung steuert
- Flash-Speicher für die Programme des Benutzers
- USB-Stecker für die Verbindung zu einem PC oder MAC

Der SENSE wird mit einem Adapter für eine externe Batterie geliefert. Eine solche Batterie kann eine Standard-Powerbank mit einem USB-Anschluss sein.

Vielleicht verfügen Sie auch über das NeuLog Batterie-Modul BAT-202, das direkt in einen der Steckplätze für Sensoren eingesteckt werden kann.



Wenn Sie eine Batterie verwenden und den SENSE vom PC trennen, wird der Roboter unabhängig und funktioniert gemäss dem Programm in seinem Flash-Speicher.

1.4 RobocklySENSE

RobocklySENSE ist ein Programm-Editor mit Bausteinen. Diese Bausteine werden zusammengesetzt, um ein Programm zu bauen anstatt codierte Texte zu schreiben.

RobocklySENSE benutzt spezielle Blöcke, welche die Eingangsdaten lesen, die Ausgangsdaten steuern und die Daten von NeuLog-Sensoren einlesen.

RobocklySENSE ist anwenderfreundlich und einfach, um Roboterprogramme zu schreiben und anzuwenden.



1.5 RobocklySENSE Installation

Die Software muss installiert werden, bevor irgendwelche Module mit dem Computer verbunden werden.

Laden Sie das Programm von der Website www.neulog.com/software herunter. Installieren Sie die Software auf Ihrem Rechner.

Der Installationsprozess läuft in zwei Schritten ab: Installation der RobocklySENSE-Software und die Installation der USB-Treiber. Nach dem Installations-Vorgang ist die Software betriebsbereit.

Das Icon von RobocklySENSE  erscheint auf dem Desktop.


Hinweis:

Die Software kann jederzeit aktualisiert werden. Das Update ersetzt nur die wichtigen Dateien. Die Deinstallation der Software vor dem Update ist nicht notwendig.

Während der Aktualisierung der Software kann die USB-Treiberinstallation übersprungen werden, indem Sie die Schaltfläche **Abbrechen** drücken.

1.6 RobocklySENSE-Programm starten

Sie finden das RobocklySENSE-Icon  auf Ihrem Desktop.


Klicken Sie auf das RobocklySENSE-Icon , um das Programm zu starten.

Das Programm wird in einem Browser gestartet und der folgende Bildschirm erscheint.



Diese Anzeige ist für den direkten Modus (erklärt im Experiment 1.1)

Das Programm wird in zwei Schritten beendet:

1. Schliessen Sie das Browser-Fenster.
2. Klicken Sie unten auf das RobocklySENSE-Icon  und schliessen Sie das offene Fenster.



Experiment 1.1 – Direkter Modus

Ziele:

- Die SENSE-Einheiten und die Komponenten kennenlernen.
- Die SENSE-Einheiten steuern im direkten Modus.

Benötigtes Material:

- Computer
- RobocklySENSE-Software
- SENSE-Roboter
- USB-Verbindungskabel

Diskussion:

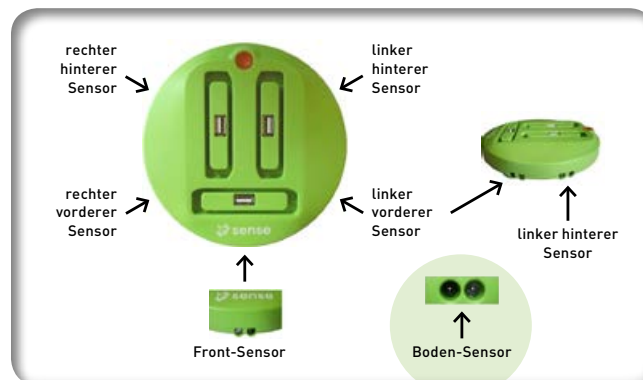
Die RobocklySENSE-Software ermöglicht die direkte Steuerung des SENSE-Controllers.

Sie lernen, wie Sie die SENSE-Motoren steuern und wie Sie die Sensoren des SENSE direkt lesen können, ohne zu programmieren.

Ablauf:

1. Beobachten Sie den SENSE.
2. Suchen Sie den Stecker für das Verbindungskabel.
3. Halten Sie den SENSE und drehen Sie ihn um.
4. Der SENSE beinhaltet:
 - 2 Motoren mit Rädern
 - Ein Drehrad
 - 5 Sensoren mit je 1 Infrarot-LED und einem Fototransistor
 - 1 Linien-Sensor (schwarz oder weiss) mit einer Infrarot-LED und einem Fototransistor

Suchen Sie diese.



5. Verbinden Sie den SENSE mit dem PC mit Hilfe des USB-Kabels.
6. Starten Sie die RobocklySENSE-Software.
Der Bildschirm für die direkte Steuerung erscheint:



Mit dem direkten Modus können Sie das System und seine Verbindungen ausprobieren, bevor Sie programmieren und die Programme starten. Dies erspart Ihnen viele Frustrationen in der Entwicklung.

Das Fenster des **direkten** Modus ist gedacht für die manuelle Kontrolle und den Test der optionalen Roboter-Einheiten: **Sense, Robo, Robo Ex, Brain Arm, Brain Servo, Brain Motor** oder **NeuLog-Sensor**.

Der **Direkt**-Bildschirm ändert sich je nach gewähltem Gerät. Jedes Gerät hat seine eigene ID-Nummer. Der Benutzer kann diese ID-Nummer ändern. In dieser Anleitung werden wir immer auf die Standard-ID-Nummer 1 zurückgreifen.

1.1.1 SENSE-Bewegung

Der SENSE hat 9 Bewegungsbefehle

Forward	beide Räder drehen vorwärts
Stop	beide Räder stehen still
Backward	beide Räder drehen rückwärts
Left deviate	das rechte Rad dreht schnell vorwärts und das linke Rad dreht langsam vorwärts
Left turn	das rechte Rad dreht schnell vorwärts und das linke Rad steht still
Left rotate	das rechte Rad dreht schnell vorwärts und das linke Rad schnell rückwärts
Right deviate	das linke Rad dreht schnell vorwärts und das rechte Rad dreht langsam vorwärts
Right turn	das linke Rad dreht schnell vorwärts und das rechte Rad steht still
Right rotate	das linke Rad dreht schnell vorwärts und das rechte Rad dreht schnell rückwärts

Jeder Befehl hat eine Pfeiltaste auf dem **Direkt**-Bildschirm

Es gibt drei Schaltflächen, um die Geschwindigkeit zu steuern:



- Suchen Sie die Pfeiltasten.
- Halten Sie den SENSE in der Hand.
- Klicken Sie auf die Pfeiltasten und beobachten Sie die Reaktion des SENSE.

10. Stellen Sie den SENSE auf den Tisch.

11. Klicken Sie erneut auf die Pfeiltasten und beobachten Sie die Reaktion des SENSE.

1.1.2 Die SENSE-Sensoren

Der SENSE hat 5 Entfernung-Sensoren auf der Seitenfläche und einen Linien-Sensor auf der Unterseite. Sie werden folgendermassen benannt:

- Bottom sensor Boden-Sensor
- Front sensor Front-Sensor
- Right front sensor rechter vorderer Sensor
- Right back sensor rechter hinterer Sensor
- Left front sensor linker vorderer Sensor
- Left back sensor linker hinterer Sensor

Jeder Sensor hat eine Infrarot-LED (Leuchtdiode) und einen Fototransistor (Lichtsensor), der nach aussen gerichtet ist.

Wenn der SENSE einen Befehl erhält, einen der Entfernungssensoren einzulesen, leuchtet dessen LED und misst die Intensität des empfangenen Lichtes.

Die Entfernungssensoren basieren auf Infrarot-Licht, um den Einfluss des Umgebungslichtes zu reduzieren.

Eine weisse Oberfläche reflektiert sehr viel mehr Licht als eine schwarze Oberfläche. Der Linien-Sensor auf der Unterseite basiert auf diesem Effekt.

Hinweis:

Die Entfernungssensoren sind nicht kalibriert. Die gemessenen Werte zeigen die Intensität des reflektierten Infrarot-Lichtes an. Für die gleiche Distanz erhalten Sie möglicherweise unterschiedliche Werte von den verschiedenen Sensoren.

Beachten Sie die seitlichen Entfernungssensoren. Sie stehen alle in einem Winkel von 45° . Das Experiment «Eine Wand abtasten» (Experiment 1.4.) erklärt die Gründe dafür.

12. Beobachten Sie den **Direct**-Bildschirm und lesen Sie die Werte ab rund um das SENSE-Bild



Diese Werte sind die abgelesenen Werte der SENSE-Sensoren

13. Platzieren Sie den SENSE auf einer weissen Oberfläche.
Der Wert des Bottom-Sensors sollte über 500 sein.
14. Platzieren Sie den SENSE auf einer schwarzen Oberfläche.
Der Wert des Bottom-Sensors sollte stark sinken.

Beachten Sie, dass die Werte auf verschiedenen schwarzen Flächen sehr unterschiedlich sein können.

15. Platzieren Sie den SENSE in verschiedenen Distanzen von einer Wand und beobachten Sie die Auswirkungen auf die 5 Entfernung-Sensoren.

Hinweis:

Der abgelesene Wert jedes Sensors für die gleiche Distanz kann sich von Sensor zu Sensor unterscheiden.

Experiment 1.2. – Erste Programme

Ziele

- Anleitungen befolgen um einen Prozess aufzubauen.
- Ein Programm auf den Controller herunterladen und ausführen.

Benötigtes Material:

- Computer
- RobocklySENSE Software
- SENSE-Roboter
- USB-Verbindungskabel
- Batterie-Modul oder Powerbank mit Adapter

Diskussion:

Ein Computerprogramm ist zusammengesetzt aus Befehl-Ketten gemäss der Befehlsstruktur der Programmiersprache. Es gibt verschiedene Programmiersprachen mit unterschiedlichen Befehlsstrukturen und verschiedenen Programmtypen.

Wir müssen dem Computer sagen, welcher Befehl der erste in einer Kette ist. Der Computer wird diesen Befehl ausführen und weitergehen zum nächsten Befehl in der Kette.

Das Programm kann Befehle beinhalten, die den Computer zu anderen Befehlen weiterführen als zum nächsten in der Kette.

Das Programm kann Befehle beinhalten, die den Computer in eine andere Kette führen, je nachdem, ob eine Bedingung erfüllt ist oder nicht. Das folgende Experiment beschreibt diese Optionen.

Es gibt viele Typen von Programmiersprachen. Jede hat ihre eigene Syntax und ihre eigene Befehlsstruktur.

Der RobocklySENSE ist ein Editor mit Programmbausteinen. Diese werden verbunden, um ein Programm zu bauen. Dadurch kann man sich auf eine Problemstellung konzentrieren und muss nicht Codier-Texte in einer bestimmten Programmiersprache schreiben.

RobocklySENSE ist sehr anwenderfreundlich und es ist einfach, Programme zu bauen und auszuführen. Es ist geeignet für Robotik-Programme und ist die beste Software für Anfänger.

Ablauf:

1. Verbinden Sie den SENSE mit dem PC mit Hilfe eines USB-Kabels.
2. Starten Sie die **RobocklySENSE**-Software.

Der Direct-Bildschirm erscheint:

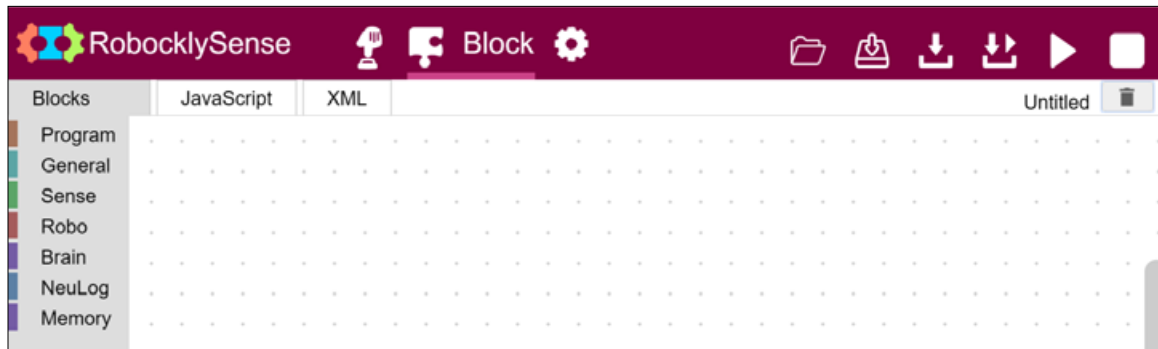


Mit dem direkten Modus können Sie das System und seine Verbindungen ausprobieren, bevor Sie programmieren und die Programme starten. Dies erspart Ihnen viele Frustrationen in der Entwicklung.

3. Probieren Sie die SENSE-Motoren aus gemäss der Beschreibung in Experiment 1.1.

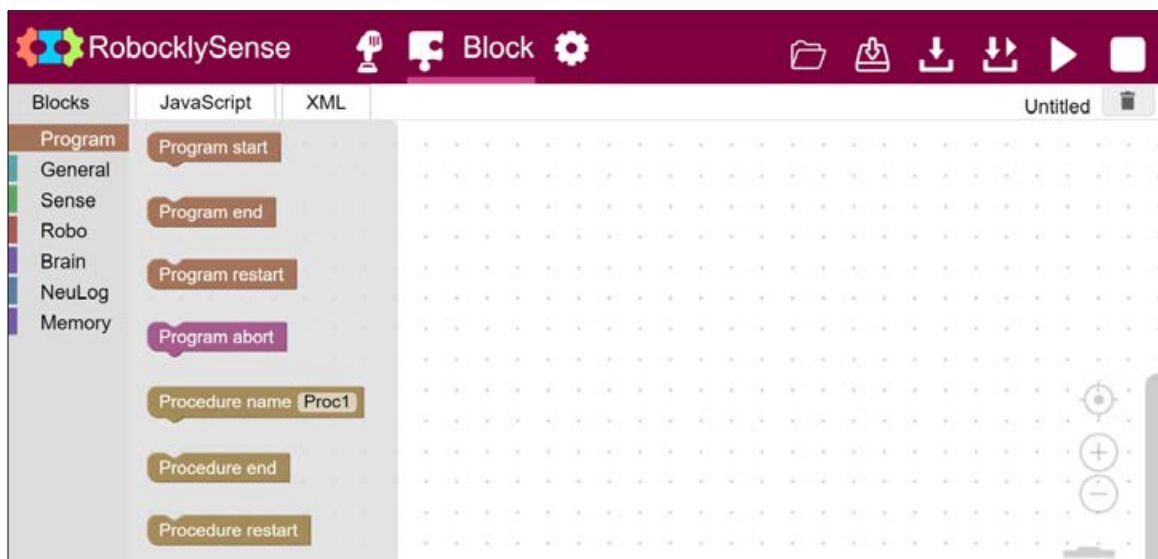
1.2.1 Erstes Programm - vorwärts

4. Wechseln Sie in den **Block-Modus**  .

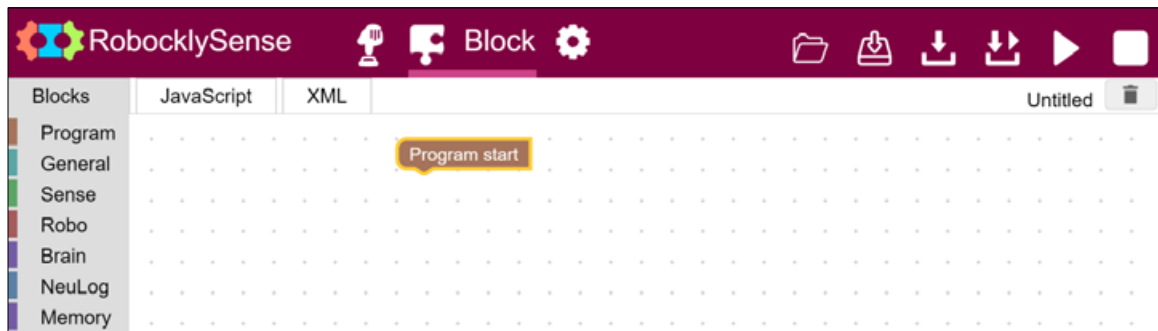


Ein Computerprogramm besteht aus Befehl-Ketten.

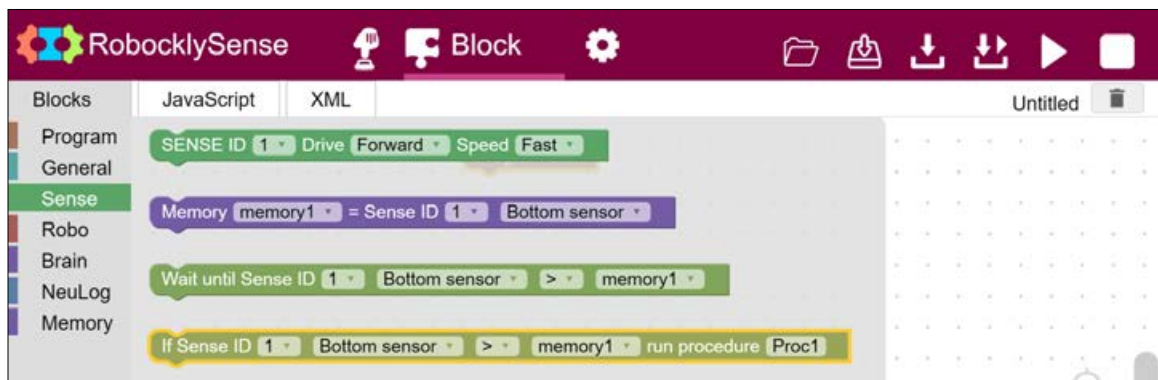
5. Klicken Sie auf die Schaltfläche **Program** und eine Liste mit Programm-Befehlen erscheint.



6. Klicken Sie auf den Befehls-Block **Program start** und ziehen Sie ihn nach rechts.



7. Klicken Sie auf die Schaltfläche **Sense** und eine Liste mit Sense-Befehlen erscheint.

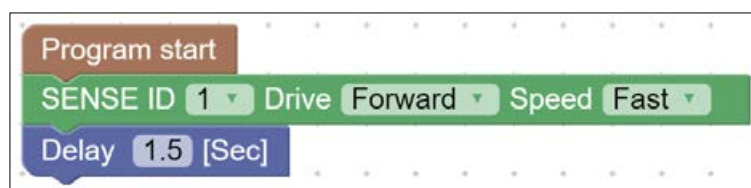


8. Klicken Sie auf den Befehlsblock **SENSE Drive**, ziehen Sie ihn nach rechts und passen Sie ihn ein in den Befehlsblock **Program start**.



9. Klicken Sie auf die Schaltfläche **General** und wählen Sie den Block **Delay**.

Ziehen Sie diesen Block nach rechts und passen ihn ein in den Block **Drive**.

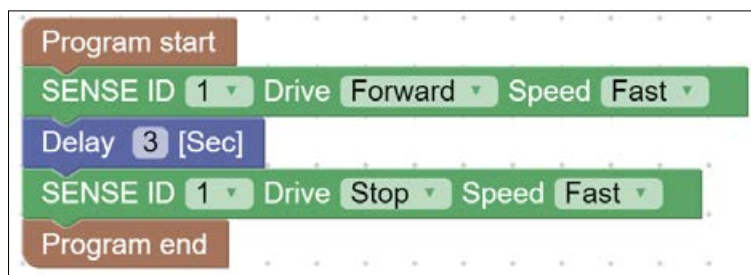


10. Ändern Sie den Wert im Block **Delay** auf 3 Sekunden.
11. Klicken Sie auf die Schaltfläche **Sense** und wählen Sie den Befehlsblock **Stop**. Ziehen Sie diesen BLock unter den **Delay**-Block.



Der **Stop**-Befehl wird 3 Sekunden nach dem **Drive-Forward**-Befehl ausgeführt

12. Klicken Sie auf die Schaltfläche **Program** und wählen Sie den Block **Program end**. Ziehen Sie diesen unter den Block **Drive Stop**.
13. Überprüfen Sie, dass Sie das folgende Programm haben:



14. Klicken Sie auf die Schaltfläche **Save**  und speichern Sie das Programm unter dem Namen **PROG1**.

1.2.2 Programm Download

15. Der SENSE Controller ist auch ein Computer mit einem Speicher.

Wir können das Programm in diesen Speicher herunterladen.

16. Klicken Sie auf die Schaltfläche **Program Download**  .

Das Programm wird nun vom PC zum Flash-Speicher des SENSE gesendet. Es ersetzt ein früheres Programm, falls ein solches existiert.

17. Klicken Sie auf die Schaltfläche **Run**  .

Der SENSE bewegt sich nun während 3 Sekunden vorwärts und hält dann an.

Während das Programm läuft, ändert die Menuleiste wie unten gezeigt mit einer Schaltfläche **Stop** auf der rechten Seite.



18. Um das Programm im SENSE-Speicher zu starten, können Sie auch die orange **Start/Stop**-Taste oben auf dem SENSE drücken.

Wenn Sie diese Taste drücken, wird der SENSE sich 3 Sekunden vorwärts bewegen und dann anhalten.

19. Der SENSE wird mit einem Adapter für eine externe Batterie geliefert. Ein solche Batterie kann eine Standard-Powerbank mit einem USB-Anschluss sein.

Vielleicht verfügen Sie auch über das NeuLog Batterie-Modul BAT-202, das direkt in einen der Steckplätze für Sensoren eingesteckt werden kann.



Wenn Sie eine Batterie verwenden und den SENSE vom PC trennen, wird der Roboter unabhängig und funktioniert gemäss dem Programm in seinem Flash-Speicher.

Verbinden Sie die Batterie mit dem Stecker am SENSE und trennen Sie dann den SENSE vom PC.

Die Menüleiste verändert sich wie folgt:



20. Drücken Sie die **Start/Stop**-Taste und Sie sehen, dass sich der Roboter 3 Sekunden vorwärts bewegt und dann anhält.

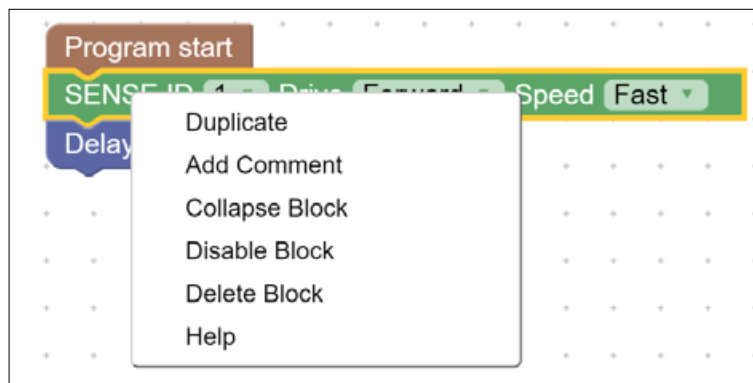
Die SENSE-LED blinkt während dieses Vorgangs.

21. Verbinden Sie den SENSE wieder mit dem PC und warten Sie, bis die Menüleiste wieder folgendermassen aussieht:



1.2.3 Vorwärts und rückwärts

22. Klicken Sie mit der rechten Maustaste auf den Block **Drive**.



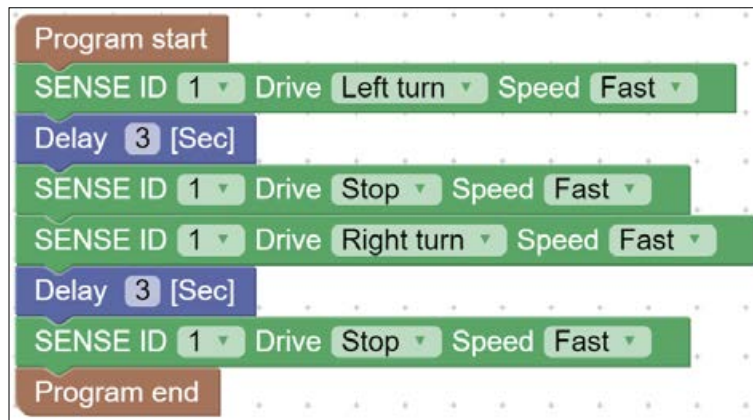
23. Wählen Sie **Duplicate** und ein neuer **Drive**-Block erscheint. Passen Sie diesen in den **Stop**-Block ein.

24. Ändern Sie die Befehle im neuen **Drive**-Block auf **Backward**.

Es ist besser, wenn der Motor gestoppt wird, bevor er die Richtung wechselt.

25. Duplizieren Sie die Befehlsblöcke **Delay** und **Stop**.

26. Ziehen Sie die Befehls-Blöcke und bauen Sie das folgende Programm:



27. Klicken Sie auf die Schaltfläche **Save**  und sichern Sie das Programm unter dem Namen **PROG2**.

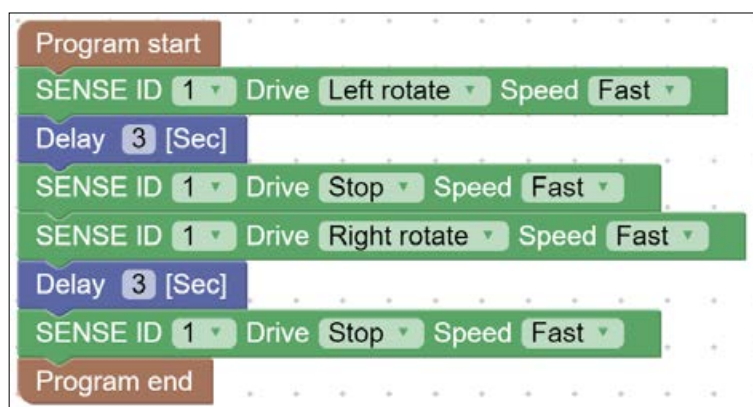
28. Klicken Sie auf die Schaltfläche **Program Download** .

29. Klicken Sie auf die Schaltfläche **Run** .

Der SENSE bewegt sich 3 Sekunden lang vorwärts und 3 Sekunden lang rückwärts und hält dann an.

1.2.4 Drehen nach links und nach rechts

30. Ändern Sie das Programm wie folgt:



Beachten Sie besonders die **Drive**-Befehle.

31. Klicken Sie auf die Schaltfläche **Save**  und sichern Sie das Programm unter dem Namen **PROG3**.

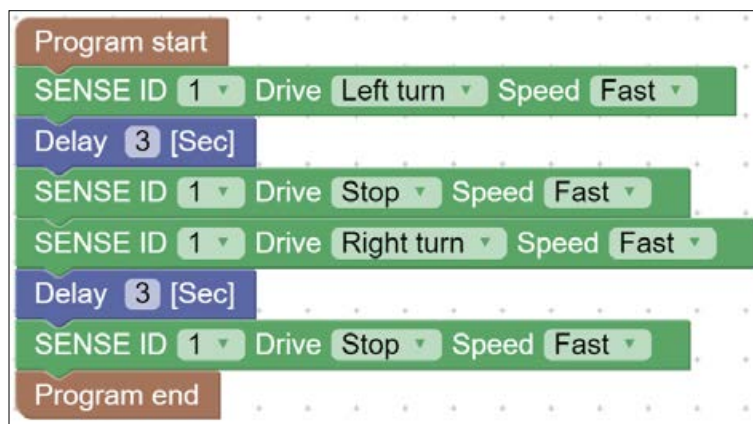
32. Klicken Sie auf die Schaltfläche **Program Download**  .

33. Klicken Sie auf die Schaltfläche **Run**  .

Der SENSE dreht sich 3 Sekunden lang nach links, dreht 3 Sekunden lang nach rechts und hält dann an.

34. Ändern Sie die Zeit beim **Delay**, um genügend Zeit zu haben für die 90°-Drehung.

35. Ändern Sie das Programm wie folgt:



36. Klicken Sie auf die Schaltfläche **Save**  und sichern Sie das Programm unter dem Namen **PROG4**.

37. Klicken Sie auf die Schaltfläche **Program Download**  .

38. Klicken Sie auf die Schaltfläche **Run**  .

Der SENSE dreht sich 3 Sekunden lang nach rechts, dreht 3 Sekunden lang nach links und hält dann an.

Was ist der Unterschied im Verhalten bei den beiden Dreh-Programmen?

39. Ändern Sie die Zeit beim **Delay**, um genügend Zeit zu haben für die 90°-Drehung.

1.2.5. Übungen – Bewegung in einem Quadrat

Aufgabe 1:

Ändern Sie die **Drive**-Befehle im Programm in **Deviate**-Befehle.

Beobachten Sie die Bewegungen des SENSE.

Aufgabe 2:

Bauen Sie ein Programm, dass der SENSE ein Quadrat von 30x30 cm fährt und zum Ausgangspunkt zurückkehrt.

Benutzen Sie die **Rotate**-Befehle für die Drehungen.

Aufgabe 3:

Bauen Sie ein Programm, dass der SENSE ein Quadrat von 30x30 cm fährt und zum Ausgangspunkt zurückkehrt.

Benutzen Sie die **Turn**-Befehle für die Drehungen.

Experiment 1.3 – Interaktive Programme

Ziele:

- Das Programm reagiert auf die Sensoren
- Gebrauch von Variablen
- Bewegung des SENSE zwischen Linien
- Bewegung des SENSE zwischen Linie und Wand

Notwendiges Material:

- Computer
- RobocklySENSE-Software
- SENSE-Roboter
- USB-Verbindungskabel
- Batterie-Modul oder Powerbank mit Adapter

Diskussion:

In diesem Experiment bewegt sich der Sensor zwischen zwei schwarzen Linien. Die positione der Linien begrenzt seine Bewegung. Der Roboter ändert seine Richtung, wenn er an die schwarze Linie stösst. Dies ist ein Beispiel eines Systems, das man Manipulator nennt

Sie lernen in diesem Experiment, wie das Lesen und die Reaktion auf die Linie und auf die vorderen Entfernungssensoren gesteuert wird.

Ein geschlossenes Regelkreis-System ist ein System, das auf Sensoren und Schalter reagiert.

Ein automatisches Beleuchtungssystem, das einen Lichtsensor beinhaltet, ist ein Beispiel für ein solches System. Das Regelkreis-System wird die Lampe einschalten, wenn es dunkel ist, und die Lampe ausschalten, wenn es hell ist. Dieses System wird automatisch der Sommerzeit (wenn die Nacht kurz ist) und der Winterzeit (wenn die Nacht lang ist und früh beginnt) angepasst.

Das Programm eines Regelkreises enthält Entscheidungs-Befehle wie:

Wait-until; stay while; if-then

Die Programme in diesem Experiment benutzen den **Wait-until**-Befehl.

1.3.1 Speicher und Variablen

Im Delay-Befehl wählen wir eine Zahl, welche die Dauer in Sekunden bestimmt.

Wir nennen eine Zahl, die Teil eines Befehls ist, **Konstante**. Wenn wir diese Zahl ändern wollen, müssen wir den relevanten Befehl suchen.

Im Programmieren verwenden wir bevorzugt keine Konstanten, sondern Variablen.

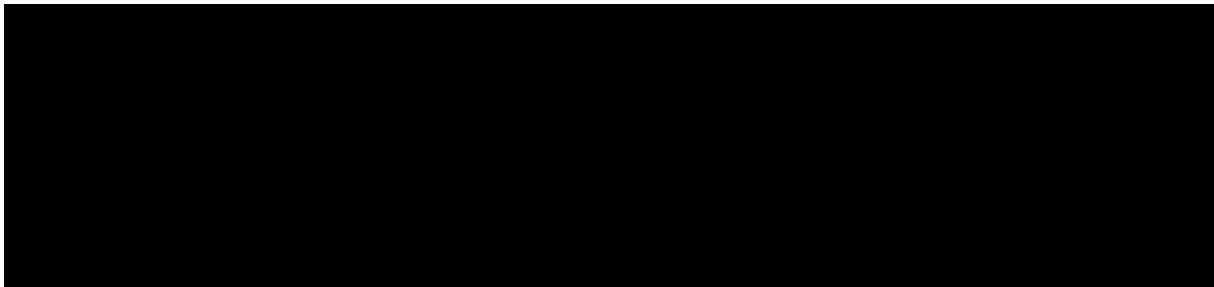
Eine **Variable** ist eine Speicherzelle mit einem Namen. Im Befehl verwenden wir den Namen der Speicherzelle.

Wir können den Wert der Variablen an einer bestimmten Stelle des Programmes einsetzen, was uns die Suche nach Befehlen erspart.

Die Software enthält spezielle Anweisungen für veränderliche Variablen-Werte gemäss den Programmbedingungen.

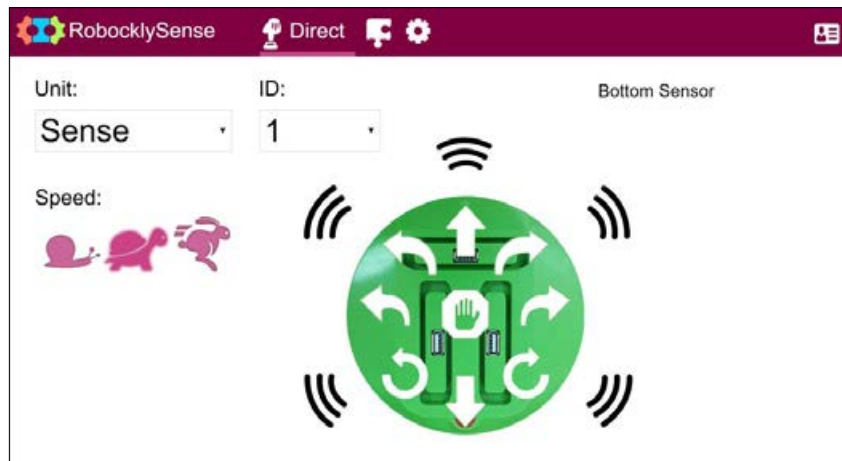
Im **RobocklySENSE** benutzen wir Speicher (Speicher 1 bis Speicher 10) als Variablen.


Bevor Sie mit dem Experiment beginnen, drucken Sie zwei schwarze Linien aus, wie unten abgebildet:

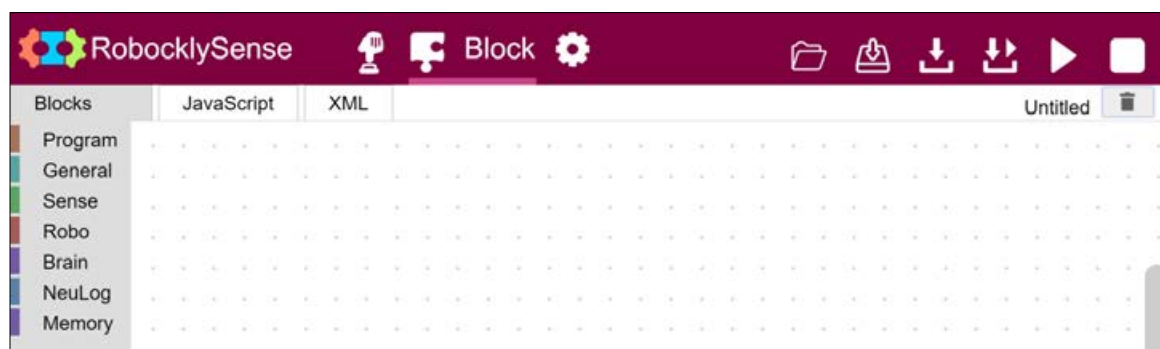


Ablauf:

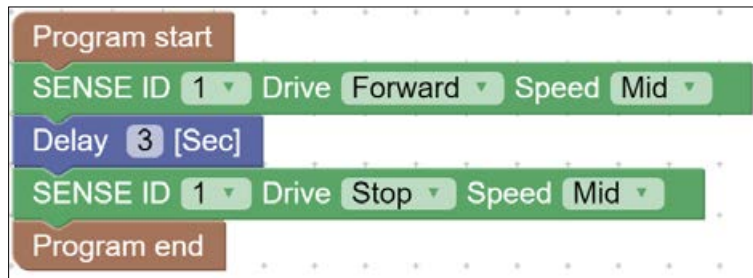
1. Verbinden Sie den SENSE mit dem PC mit Hilfe des USB-Kabels.
2. Starten Sie die **RobocklySENSE**-Software.






3. Überprüfen Sie die SENSE-Motoren, wie in Experiment 1.1. beschrieben.
4. Überprüfen Sie den Bottom-Sensor, wie in Experiment 1.1 beschrieben.
5. Zeichnen Sie die Werte des Bottom-Sensors auf, wenn der SENSE auf einer weissen Oberfläche steht. Wir nennen diesen Wert den «White Value».
6. Zeichnen Sie die Werte des Bottom-Sensors auf, wenn der SENSE auf einer schwarzen Linie steht. Wir nennen diesen Wert den «Black Value».
7. Wechseln Sie in den **Block**-Modus .



8. Bauen Sie das folgende Programm, welches den SENSE für 3 Sekunden vorwärtsbewegt und dann stoppt.

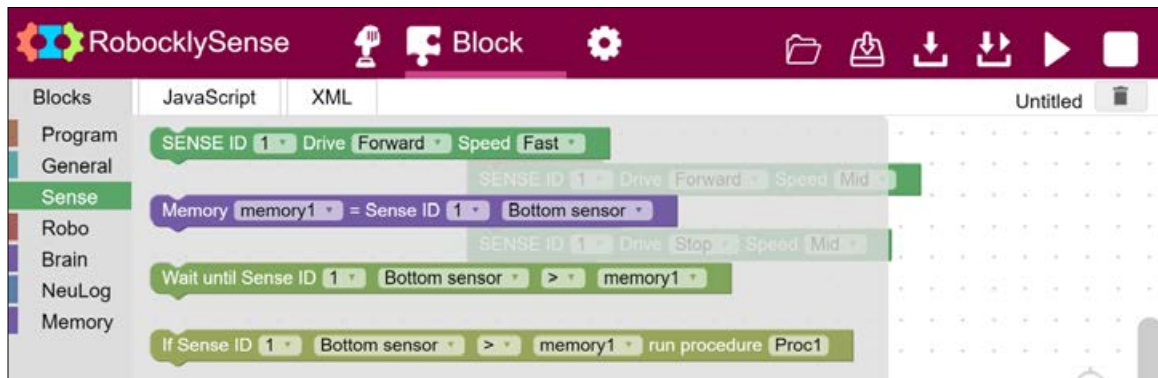


9. Klicken Sie auf die Schaltfläche **Save**  und sichern Sie das Programm unter dem Namen **CART1**.
10. Klicken Sie auf die Schaltfläche **Program Download**  .
11. Stellen Sie den SENSE auf den Tisch oder auf den Boden und klicken Sie auf die Schaltfläche **Run**  .
- Der SENSE bewegt sich 3 Sekunden vorwärts und stoppt.
12. Trennen Sie den SENSE vom PC und stecken Sie das Batterie-Modul ein.
13. Drücken Sie die orange Taste am SENSE und Sie werden sehen, dass er sich 3 Sekunden vorwärts bewegt und dann stoppt.
- Die SENSE-LED blinkt während dieser Bewegung.
14. Verbinden Sie den SENSE wieder mit dem PC.

1.3.2 Wait-until-Befehl

Wir ersetzen den Delay-Befehl durch den Befehl Wait-until.

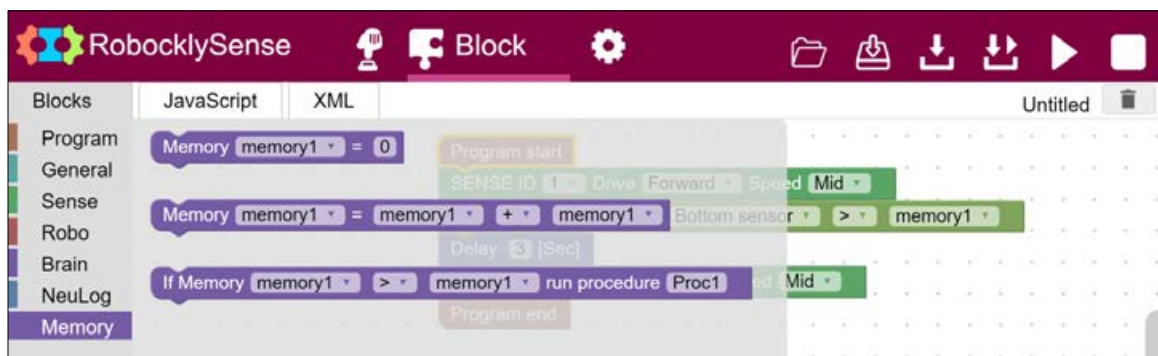
15. Klicken Sie auf die Schaltfläche **Sense** und eine Liste von Eingabe-Befehlen erscheint.



16. Klicken Sie auf den Block
 «**Wait until Sense ID 1 Bottom sensor > [memory1]**» und ziehen Sie ihn ins Programm oberhalb des **Delay**-Befehls.

17. Ändern Sie das Prüfzeichen > (grösser als) in < (kleiner als).

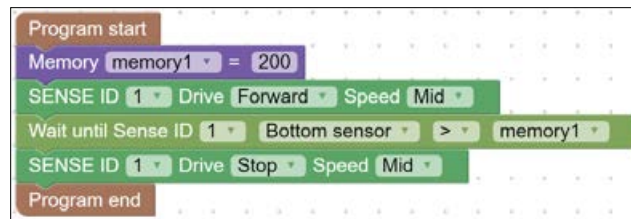
18. Klicken Sie auf die Schaltfläche **Memory** und die folgende Liste der Eingabe-Befehle erscheint:



19. Klicken Sie auf den Block «**Memory [memory1] = 0**» und ziehen Sie ihn an die erste Stelle des Programms.

20. Ändern Sie die Zahl im Befehl auf 50 über dem **Black Value**.

21. Löschen Sie den **Delay**-Block und prüfen Sie, dass Sie nun das folgende Programm haben:

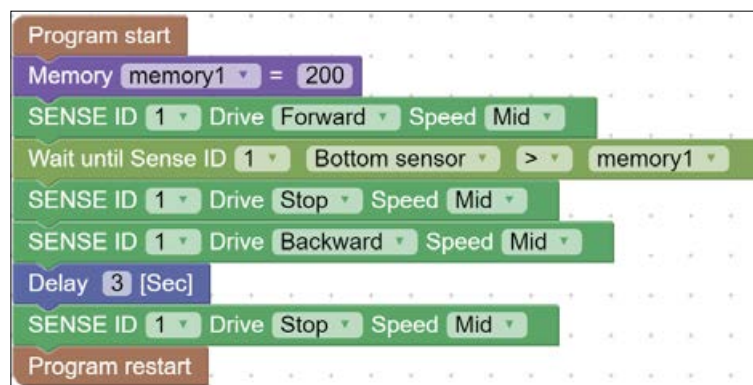


22. Platzieren Sie den SENSE auf einer weissen Oberfläche mit einer schwarzen Linie.
23. Starten Sie das Programm.
Der SENSE sollte sich vorwärts bewegen und anhalten, wenn er die schwarze Linie erreicht.


1.3.3 Endlos-Schleufe

Die meisten Computer-Programme laufen in einer Endlos-Schleufe.

24. Ändern Sie das Programm so, dass der SENSE vorwärts fährt und anhält, wenn er die schwarze Linie erreicht, dann während 3 Sekunden rückwärts fährt und dann wieder vorwärts.
25. Ersetzen Sie den Block **Program end** durch den Block **Program restart** aus der Liste **Program**-Befehle.

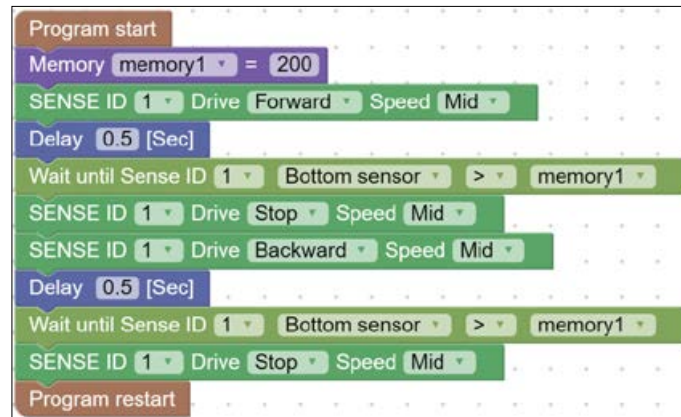


26. Platzieren Sie den SENSE auf einer weissen Oberfläche mit einer schwarzen Linie darauf.
27. Starten Sie das Programm.
Der SENSE sollte sich vorwärts bewegen, bis er die schwarze Linie erreicht, dann stoppen, dann 3 Sekunden lang rückwärts fahren und dann wieder vorwärts.

28. Stoppen Sie das Programm, indem Sie entweder den orangen Knopf auf dem SENSE drücken oder auf die Schaltfläche **Stop** , wenn der SENSE mit einem Computer verbunden ist.

1.3.4 Bewegung zwischen zwei Linien

29. Ändern Sie das Programm so, dass sich der SENSE zwischen zwei Linien bewegt.



Beachten Sie die **Delay**-Befehle. Sie sind deshalb eingefügt, damit sichergestellt ist, dass der Roboter von der ersten schwarzen Linien wegfährt und dann wartet, bevor er zur anderen schwarzen Linie fährt.

30. Speichern Sie das Programm unter dem Namen **CART2**.
31. Platzieren Sie den SENSE zwischen zwei schwarzen Linien und starten Sie das Programm.
Der SENSE sollte sich jetzt rückwärts und vorwärts bewegen zwischen den Linien.
Wenn man die Distanz zwischen den Linien vergrößert, passt sich der Weg des SENSE an.

1.3.5 Übung – Zwischen einer Wand und einer Linie (I)

Aufgabe 1: Verändern Sie das Programm **CART2** so, dass der SENSE sich zwischen einer Wand vorne und einer schwarzen Linie hinten bewegt.

Hinweis:

Sie müssen den Front-Sensor benutzen für die Vorwärtsbewegung. Beachten Sie die Vergleichszeichen (> oder <).

Experiment 1.4 – Abläufe (Procedures) als neue Befehle

Ziele:

- Benutzung von Vorgehensweisen in einem Programm
- Reaktion auf Sensoren

Notwendiges Material:

- Computer
- RobocklySENSE-Software
- SENSE-Roboter
- USB-Verbindungskabel
- Batterie-Modul oder Powerbank mit Adapter

Diskussion:

Ein Computerprogramm besteht aus einer Kette von Befehlen.


Anstatt alle Befehle in einer Kette aneinanderzureihen, können wir das Programm aufteilen in Abläufe, die mehrere kurze Ketten bilden. Jede Kette hat einen eigenen Namen.

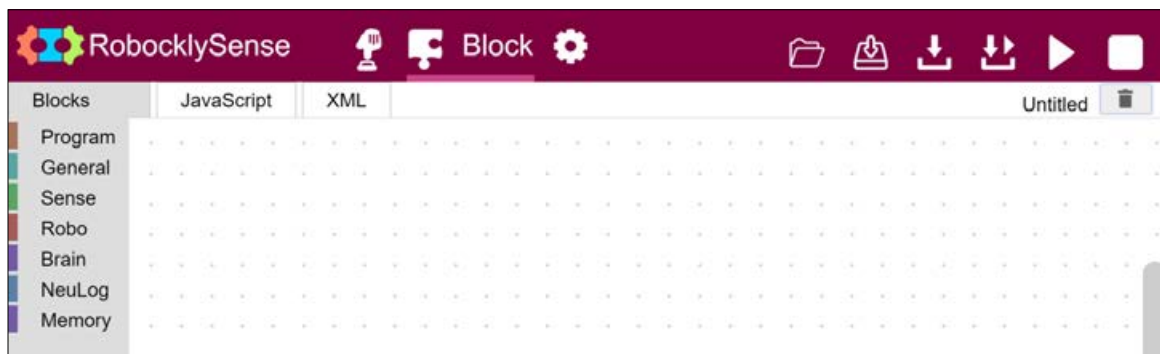
In einem Programm gibt es ein Hauptprogramm und mehrere Abläufe. So weiss der Computer, wo er anfangen muss, wenn das Programm gestartet wird.


In diesem Experiment werden wir Bewegungsabläufe bauen und benutzen.

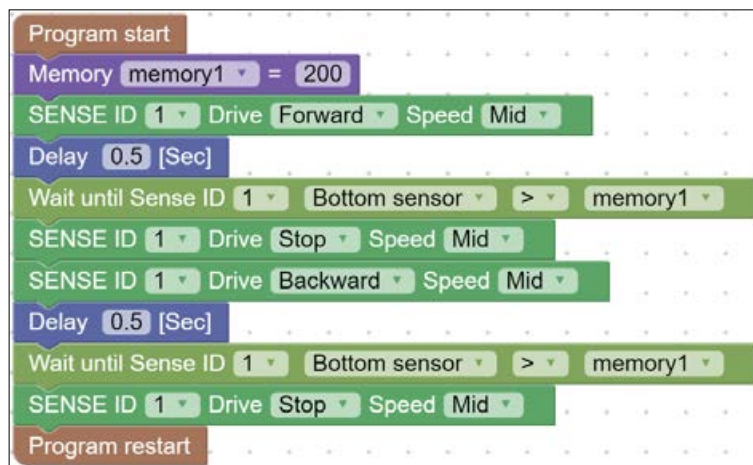
Am Ende des Programmes müssen wir einen Befehl einbauen, der das Programm an den Anfangspunkt zurückbringt (wenn wir wollen, dass das Programm immer wieder neu startet), oder das Programm stoppt und zum Betriebssystem zurückkehrt.

Ablauf:

1. Verbinden Sie den SENSE mit dem PC mit Hilfe des USB-Kabels.
2. Starten Sie die **RobocklySENSE**-Software.
3. Wechseln Sie zum **Block-Modus**  .



4. Klicken Sie auf **Open**  und öffnen Sie das Programm **CART2**.
5. Überprüfen Sie, dass Sie das folgende Programm haben:



Wenn nicht, bauen Sie dieses Programm und speichern Sie es unter dem Namen **CART2**.

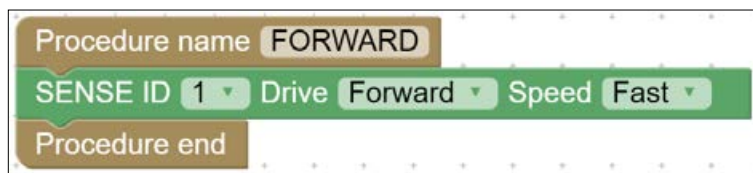
6. Platzieren Sie den SENSE zwischen zwei schwarzen Linien.

Wenn Sie die Distanz zwischen den beiden Linien vergrößern, passt sich die Bewegung des SENSE automatisch an.

7. Stoppen Sie das Programm.
8. Klicken Sie auf das Symbol **Papierkorb**, um den Bildschirm zu leeren.

1.4.1. Programme und Abläufe

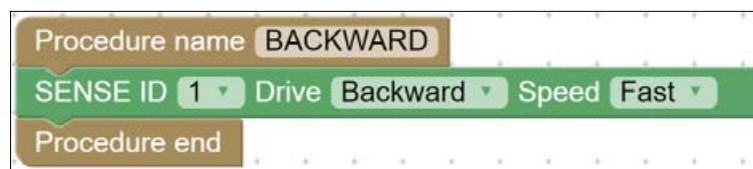
9. Klicken Sie auf die Schaltfläche **Program** und die Liste der allgemeinen Befehle erscheint.
10. Klicken Sie auf die Schaltfläche **Procedure name** und ziehen Sie diese nach rechts.
11. Der **Procedure-name**-Befehl hat ein Feld für den Ablauf-Namen. Schreiben Sie in dieses Feld den Namen **FORWARD**.
12. Bauen Sie den folgenden **FORWARD**-Ablauf.



Hinweis:

Ein Ablauf endet mit **Procedure end** oder **Procedure start**.

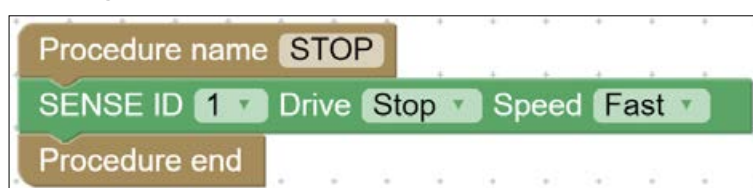
13. Bauen Sie den folgenden **BACKWARD**-Ablauf.



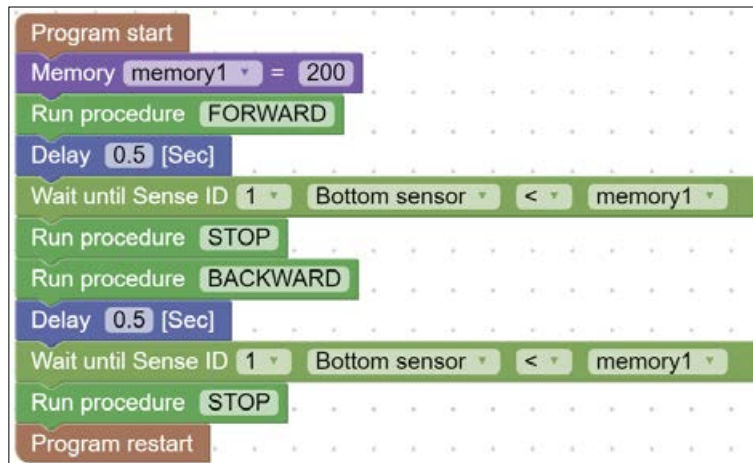
Hinweis:

Um eine Befehlsstruktur zu duplizieren, klicken Sie auf den ersten Befehl, bewegen ihn ein wenig mit der Maus mit gedrückter Maustaste und drücken dann **CTRL+c**. Dann drücken Sie **CTRL+v**. Dies wird die Befehlsstruktur duplizieren.

14. Bauen Sie den folgenden **STOP**-Ablauf.



15. Bauen Sie das folgende Hauptprogramm:



Hinweis:

Im Hauptprogramm benutzen wir anstelle der **Drive**-Befehle den **Run-procedure**-Befehl aus der Liste der **General**-Befehle. Dies eröffnet mehr Möglichkeiten, wie wir später sehen werden.

16. Speichern Sie dieses Programm unter dem Namen **CART3**.

Das Hauptprogramm und die Abläufe sind unter diesem Namen gespeichert.

Das Hauptprogramm trifft die Entscheidungen und benutzt neue Befehle wie vorwärts, rückwärts und stop.

Die Funktionen des Programmes sind die gleichen wie im Programm **CART2**.

17. Platzieren Sie den Sense zwischen zwei schwarzen Linien und starten Sie das Programm.

Der SENSE sollte jetzt vorwärts und rückwärts fahren zwischen den beiden Linien.

Wenn die Distanz zwischen den Linien vergrößert wird, passt sich der Weg des SENSE automatisch an.

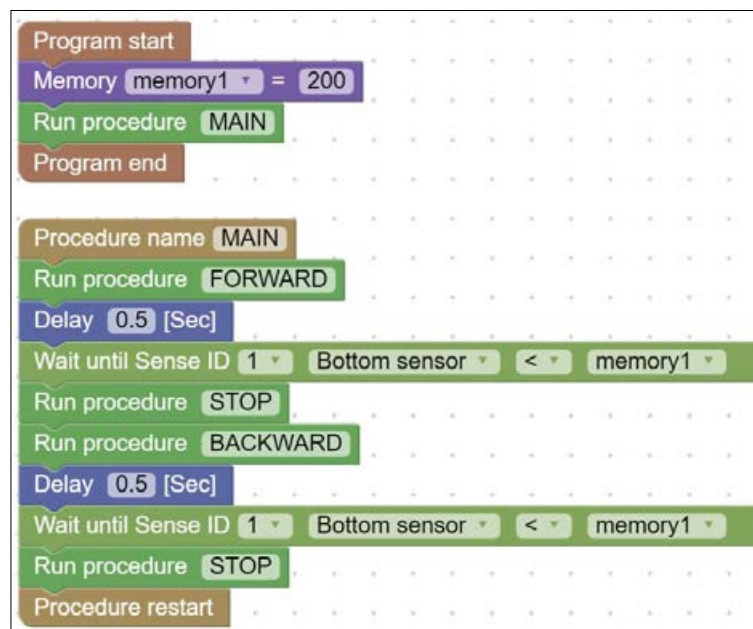
18. Stoppen Sie das Programm.

1.4.2 Definitionen

Die Definition von **memory1** wird in jeder Runde des Programmes ablaufen, obwohl wir diese Definition nur einmal eingeben müssen. Die verkürzt auch die Zeit des Programmzyklus.

Um diese Definition nur einmal machen zu müssen, bauen wir einen Haupt-Ab-
lauf ohne Definitionen, der in einer Endlosschleife abläuft. Die Definitionen
sind in diesem Programm festgelegt (unterhalb des Blockes **Start program**).
Nach der Festlegung startet das Programm den Hauptablauf.

19. Ändern Sie das Programm in eine Programm, das einen **MAIN**-Ablauf bein-
haltet:



20. Speichern Sie das Programm unter dem Namen **CART4**.

21. Das Programm funktioniert gleich wie das Programm **CART3**.

22. Platzieren Sie den SENSE zwischen zwei schwarzen Linien und starten Sie das Programm.

Der SENSE sollte sich vorwärts und rückwärts bewegen zwischen den schwarzen Linien.

23. Stoppen Sie das Programm.

1.4.3 Übungen – Zwischen einer Wand und einer Linie (II)

Aufgabe 1:

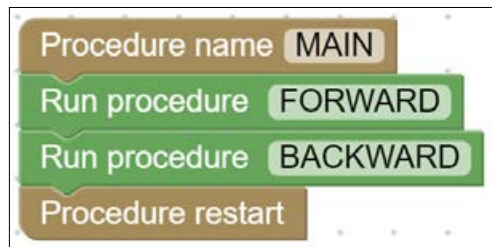
Verbessern Sie das **CART4**-Programm so, dass der SENSE sich zwischen einer Wand vorne und einer schwarzen Linie hinten bewegt.

Starten Sie das Programm und überprüfen Sie das.

Aufgabe 2:

Verbessern Sie das **CART4**-Programm so, dass der sich der SENSE wie in Aufgabe 1 zwischen einer Wand vorne und einer schwarzen Linien hinten bewegt.

Der **MAIN**-Ablauf sollte dabei folgendermassen aussehen:



Passen Sie die Abläufe **FORWARD** und **BACKWARD** entsprechend an.

1.4.4 Bewegung entlang einer schwarzen Linie

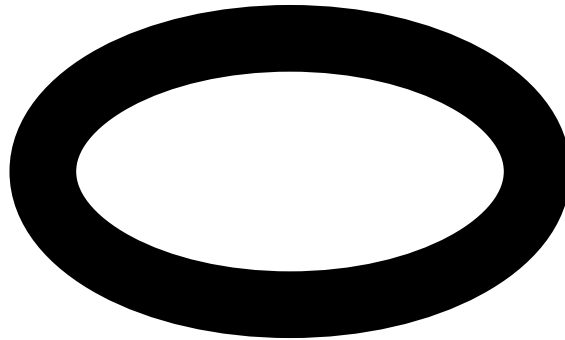
Um den SENSE einer schwarzen Linie entlang zu bewegen, benutzen wir Slow-turn-Abläufe des SENSE.

In slow-turns dreht das eine Rad und das andere Rad stoppt. So bewegt sich der SENSE immer noch vorwärts, während er dreht.

Im Hauptprogramm definieren wir die Bewegung nach folgender Überlegung:

Nach links drehen, bis der SENSE eine schwarze Fläche findet, und dann nach rechts drehen, bis der SENSE eine weisse Fläche findet.

24. Drucken Sie eine schwarze Linie auf einer ganzen Seite aus, wie abgebildet.



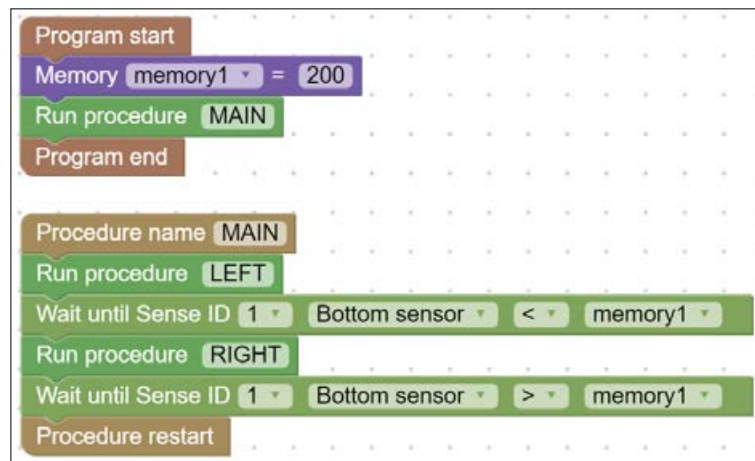
Die Linie sollte mindestens 4 cm breit sein.

25. Löschen Sie den Bildschirminhalt.

26. Bauen Sie die Abläufe **LEFT** turn und **RIGHT** turn folgendermassen:



27. Bauen Sie das folgende Hauptprogramm, das einen **MAIN**-Ablauf beinhaltet.



Hinweis:

Beachten Sie die Vergleichszeichen (< und >).

28. Speichern Sie das Programm unter dem Namen **CART5**.

29. Platzieren Sie den SENSE in der Nähe der schwarzen Linie.

30. Starten Sie das Programm und überprüfen Sie die Bewegung des SENSE.

31. Verändern Sie den Wert von **memory1**, um eine sanfte Bewegung des SENSE zu erreichen.

1.4.5. Übung – Entlang einer komplexen schwarzen Linie

Aufgabe 1:

Erstellen Sie verschiedene schwarze Linien für den SENSE und beobachten Sie sein Verhalten. Verbessern Sie wenn nötig die Programme.

Beispiel einer komplexen Linie:



Experiment 1.5 – Bedingungen und Entscheidungen

Ziele:

- Der If-Befehl
- OR-Bedingung
- AND-Bedingung

Notwendiges Material:

- Computer
- RobocklySENSE-Software
- SENSE-Roboter
- USB-Verbindungskabel
- Batterie-Modul oder Powerbank mit Adapter

Diskussion:

1.5.1 If-then-Befehl

Im vorherigen Experiment haben wir den **Wait-until**-Befehl kennengelernt. Dies ist einer der Bedingungs-Befehle.

Der **If**-Befehl ist der wichtigste Bedingungs-Befehl. Er ist zusammengesetzt aus einer Bedingung und dem Ablauf, der folgt, wenn die Bedingung erfüllt ist.


Die Bedingung wird geprüft, wenn der Bedingungs-Befehl ausgeführt wird.

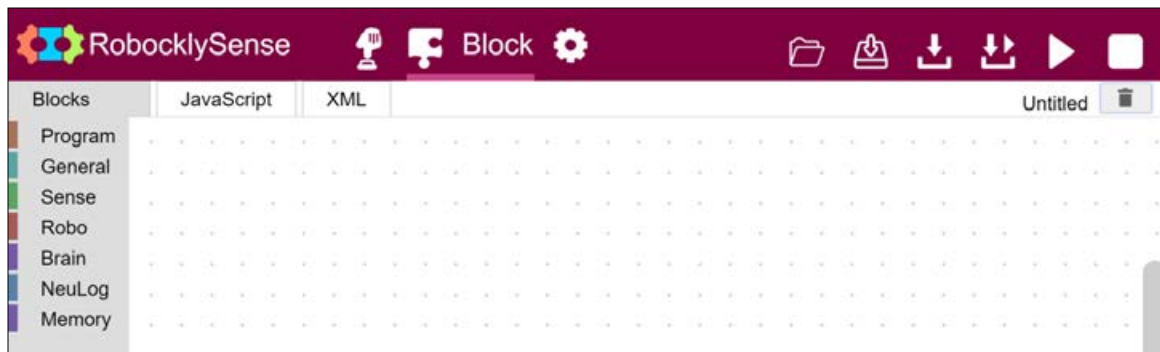
Das folgende Programm ist der wichtigste Ablauf für die Bewegung entlang einer schwarzen Linie unter Benutzung eines **If**-Befehls




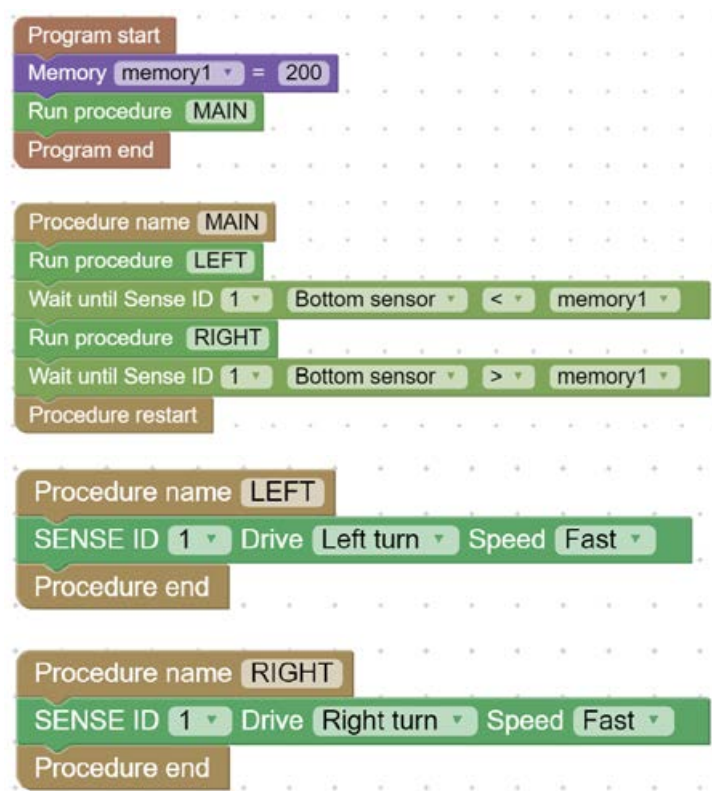
Wir können komplexere Bedingungen zusammenstellen. Diese werden **AND**-Bedingungen und **OR**-Bedingungen genannt. Sie werden weiter unten erklärt.

Ablauf:

1. Verbinden Sie den SENSE mit dem PC mit Hilfe des USB-Kabels.
2. Starten Sie die **RobocklySENSE**-Software.
3. Wechseln Sie zum **Block-Modus**  .

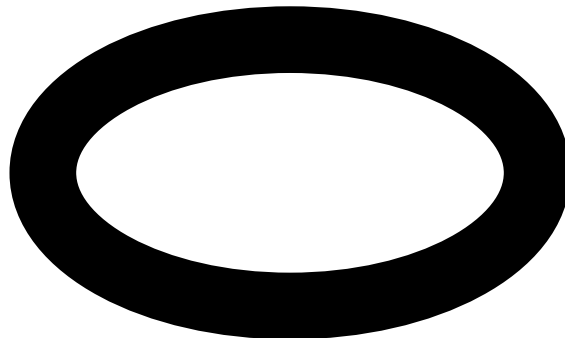


4. Klicken Sie auf **Open**  und öffnen Sie das Programm **CART5**.
5. Überprüfen Sie, dass Sie das folgende Programm haben:



Wenn nicht, bauen Sie dieses Programm und speichern Sie es unter dem Namen **CART5**.

- Platzieren Sie den SENSE auf der schwarzen Kreislinie und starten Sie das Programm.



Der SENSE sollte sich entlang der schwarzen Linie bewegen.

- Stoppen Sie das Programm
- Ändern Sie das Hauptprogramm und den **MAIN**-Ablauf wie folgt:

```

Program start
Memory memory1 = 200
Run procedure MAIN
Program end

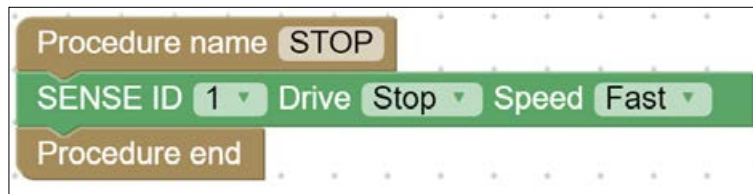
Procedure name MAIN
If Sense ID 1 Bottom sensor > memory1 run procedure LEFT
If Sense ID 1 Bottom sensor < memory1 run procedure RIGHT
Procedure restart

```

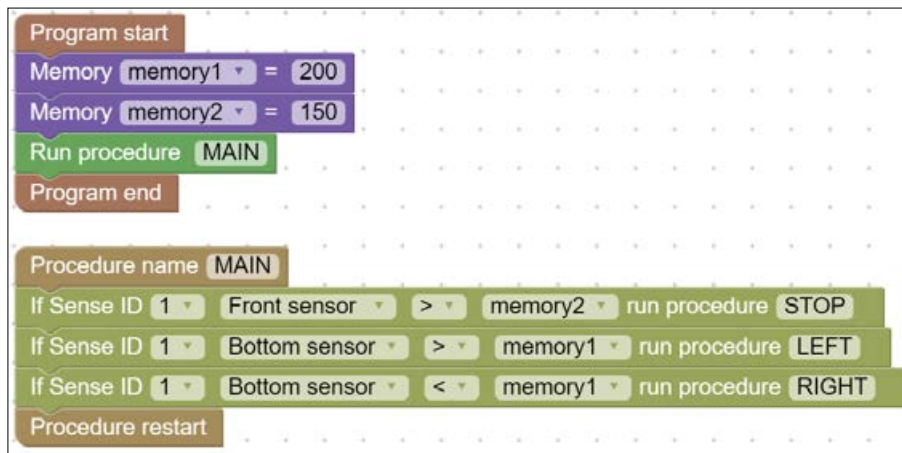
Der **If-Sense**-Befehl ist in der **Sense**-Befehl-Liste.

- Speichern Sie das Programm unter dem Namen **CART6**.
- Stellen Sie den SENSE in die Nähe der schwarzen Linie.
- Starten Sie das Programm und beobachten Sie die Bewegung des SENSE.
- Ändern Sie den Wert in **memory1**, sodass sich der SENSE sanfter bewegt.
- Wir werden das Programm **CART6** verbessern, sodass der SENSE anhält, wenn Sie Ihre Hand vorne hinhalten.

13. Bauen Sie den folgenden **STOP**-Ablauf:



14. Ändern Sie das Hauptprogramm und den **MAIN**-Ablauf wie folgt:



Wir haben zum Frontsensor eine neue Variable hinzugefügt.

In jedem Zyklus überprüft der **MAIN**-Ablauf nun die Distanz von der Wand und ruft den **STOP**-Ablauf auf, wenn der SENSE nahe an der Wand ist.

15. Stellen Sie den SENSE in die Nähe der schwarzen Linie.

16. Starten Sie das Programm und beobachten Sie die Bewegung des SENSE.

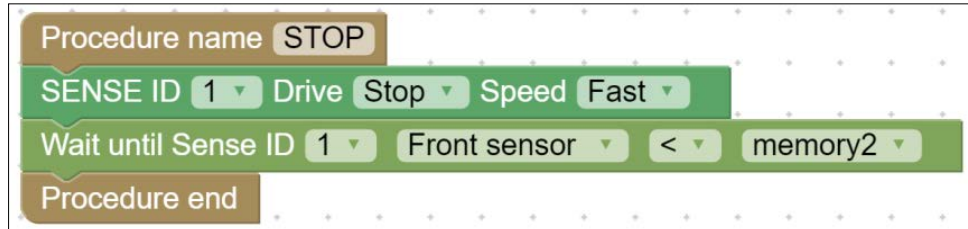
17. Halten Sie Ihre Hand vor den SENSE, während er sich bewegt.

Anstatt zu stoppen, reduziert der SENSE nur die Geschwindigkeit.

Überlegen Sie, weshalb.

18. Wir müssen den **STOP**-Ablauf so verbessern, dass er wartet, bis Sie Ihre Hand wegnehmen.

Ändern Sie den **STOP**-Ablauf wie folgt:



19. Speichern Sie das Programm unter dem Namen **CART7**.

20. Stellen Sie den SENSE in die Nähe der schwarzen Linie.

21. Starten Sie das Programm und beobachten Sie die Bewegung des SENSE.

22. Halten Sie die Hand vor den SENSE, während er sich bewegt.

Ändern Sie die Werte der **memorys**, bis der SENSE sich wie gewünscht bewegt.

1.5.2 OFF und ON mit verschiedenen Werten

In Kontrollsystemen möchten wir normalerweise, dass der OFF-Bedingungswert anders ist als der On-Bedingungswert. Der Grund dafür liegt darin, dass wir den Neustart des Systems vermeiden wollen.

23. Ändern Sie das Programm und die Abläufe wie folgt:

```

Program start
Memory memory1 = 200
Memory memory2 = 150
Memory memory3 = 130
Run procedure MAIN
Program end

Procedure name MAIN
If Sense ID 1 Front sensor > memory2 run procedure STOP
If Sense ID 1 Bottom sensor > memory1 run procedure LEFT
If Sense ID 1 Bottom sensor < memory1 run procedure RIGHT
Procedure restart

Procedure name STOP
SENSE ID 1 Drive Stop Speed Fast
Wait until Sense ID 1 Front sensor < memory3
Procedure end

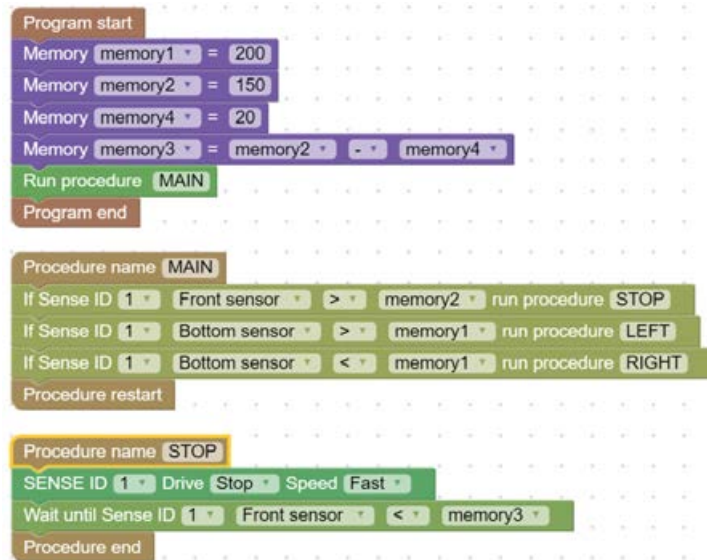
```

24. Der **STOP**-Wert ist höher als der **WAIT**-Wert.

25. Starten Sie das Programm und testen Sie es.

26. Wenn wir den **STOP**-Wert ändern, müssen wir auch den **WAIT**-Wert ändern.

Das folgende Programm stellt sicher, dass der **WAIT**-Wert (memory3) immer tiefer ist als der **STOP**-Wert (memory2).



27. Starten Sie dieses Programm und testen Sie es.

1.5.3 AND-Bedingung

Wir möchten den SENSE nur stoppen, wenn er nahe an der Wand ist **und** wenn er auf der schwarzen Linie ist.

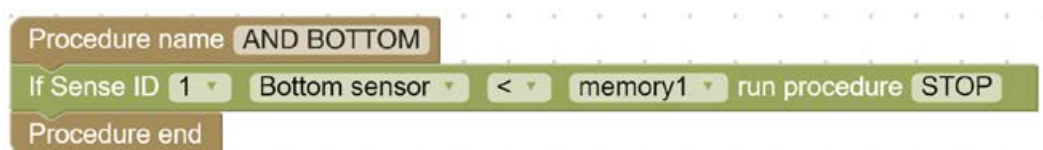
Um dies zu erreichen, brauchen wir die **AND**-Bedingung.

In einzelnen Programmier-Programmen kann ein **If**-Befehl zwei Bedingungen mit einer **AND**-Bedingung dazwischen enthalten.

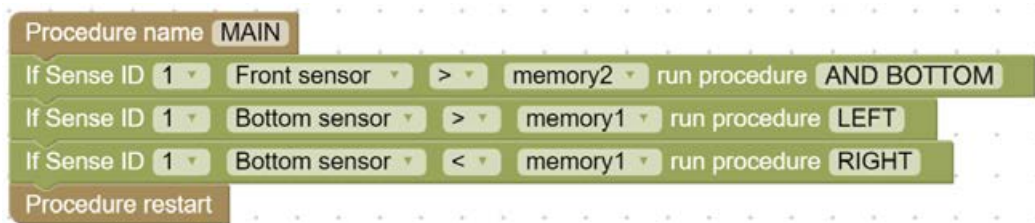
Das Programm RobocklySENSE hat diese Option nicht.

Der **AND**-Vorgang kann erreicht werden, indem man einen neuen Ablauf erstellt.

28. Erstellen Sie den **AND-Bottom**-Ablauf mit einem einzigen Befehl.



29. Ändern Sie den **MAIN**-Ablauf wie folgt:



30. Analysieren Sie das Programm und den Ablauf.

31. Speichern Sie das Programm unter dem Namen **CART8**.

32. Stellen Sie den SENSE in die Nähe der schwarzen Linie.

33. Starten Sie das Programm und beobachten Sie die Bewegung des SENSE.

34. Halten Sie Ihre Hand vor den SENSE, während er sich bewegt.

Beobachten Sie das Verhalten des SENSE.

1.5.4 OR-Bedingung

Anstatt vor der Wand zu stoppen, möchten wir, dass der SENSE umdreht und in der anderen Richtung der schwarzen Linie folgt.

Der SENSE wird nach links drehen, wenn er auf einer weissen Fläche ist oder wenn er nahe an der Wand oder beides.

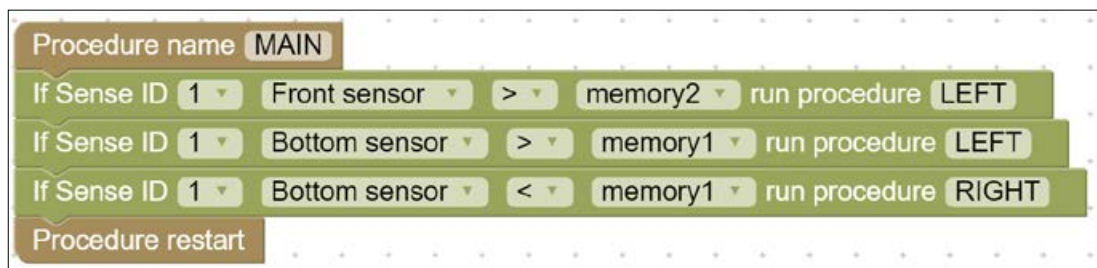
Um dies zu erreichen, brauchen wir die **OR**-Bedingung.

In einzelnen Programmier-Programmen kann ein **If**-Befehl zwei Bedingungen mit einer **OR**-Bedingung dazwischen enthalten.

Das Programm RobocklySENSE hat diese Option nicht.

Der **OR**-Vorgang kann erreicht werden, indem man die beiden **If**-Abläufe hintereinander schreibt.

35. Ändern Sie den **MAIN**-Ablauf wie folgt:



36. Analysieren Sie das Programm und die Abläufe.

37. Speichern Sie das Programm unter dem Namen **CART9**.

38. Stellen Sie den SENSE in die Nähe der schwarzen Linie.

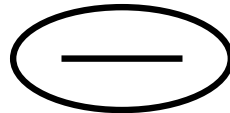
39. Legen Sie einen Block auf die schwarze Linie.

40. Starten Sie das Programm und beobachten Sie die Bewegung des SENSE.

1.5.5. Übung – Entlang von zwei Linien

Aufgabe 1:

Erstellen Sie zwei schwarze Linien wie folgt:



Legen Sie ein Hindernis auf die innere Linie und lassen Sie den SENSE entlang dieser Linie laufen.

Wenn er auf das Hindernis trifft, wechselt er auf die äussere Linie und bewegt sich entlang der äusseren Linie.

1.5.6 Bewegung entlang einer Wand

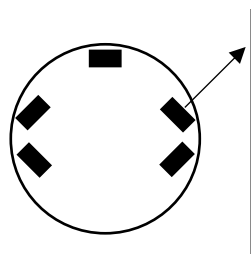
Damit sich der SENSE entlang einer Wand bewegt, benutzen wir den gleichen Algorithmus wie bei der Bewegung entlang einer schwarzen Linie. Wir benutzen den Slow-turn-Ablauf.

Im Hauptprogramm benutzen wir den **If**-Befehl, um die Bewegung gemäss dem folgenden Algorithmus zu steuern:

- Linksdrehung, wenn der SENSE zu nahe an der Wand ist.
- Rechtsdrehung, wenn der SENSE zu weit von der Wand entfernt ist.

Für die Bewegung entlang einer Wand auf der rechten Seite benutzen wir den Frontside-Sensor.

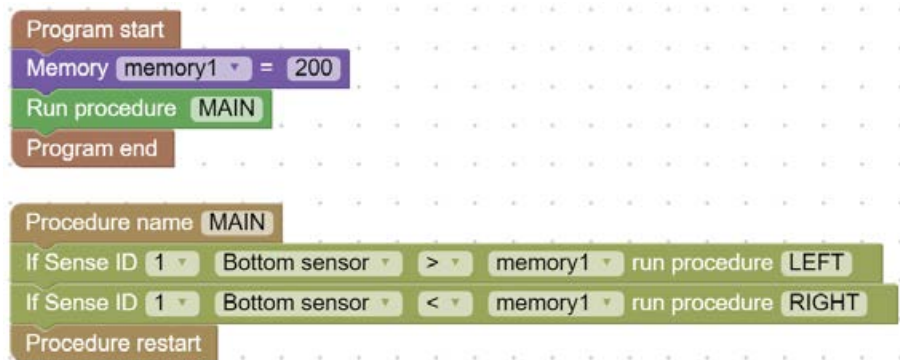
Die Sensoren auf der Seite sind in einem 45° -Winkel zur Basis des SENSE angeordnet.



Wenn der SENSE nach rechts dreht, ist die gemessene Distanz kleiner als wenn er nach links dreht.

Überlegen Sie, was passiert, wenn der Sensor parallel zur Wand steht.

41. Nehmen Sie eine runde Schachtel (Zylinder). Diese benutzen Sie als erste Versuchswand.
42. Das Basisprogramm für die Bewegung entlang einer schwarzen Linie ist **CART6**.

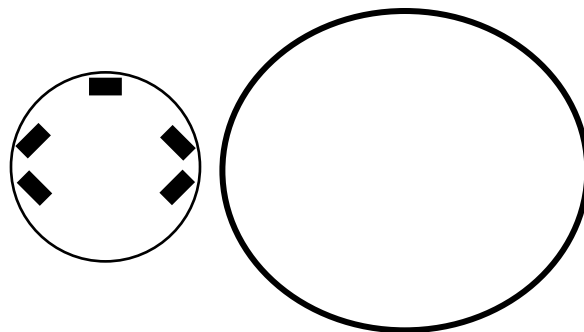


43. Ändern Sie den **MAIN**-Ablauf von **CART6** so, dass der **Right-Front**-Sensor benutzt wird anstelle des **Bottom**-Sensors.

Ändern Sie den Wert von **memory1** auf eine Distanz von 4 cm von der Wand.

44. Speichern Sie das Program unter dem Namen **WALL1**.

45. Stellen Sie den SENSE in die Nähe der runden Schachtel und starten Sie das Programm.



Überprüfen Sie, dass der SENSE sich um die Schachtel herum bewegt.

1.5.7 Übungen – Entlang von Wänden

Aufgabe 1:

Verbessern Sie das Programm **WALL1** so, dass der SENSE sich vorwärts bewegt, falls er auf der rechten Seite keine Wand spürt.

Der SENSE stoppt, wenn er eine Wand berührt, dreht nach links und bewegt sich entlang dieser Wand.

Speichern Sie dieses Programm unter dem Namen **WALL2**.

Aufgabe 2:

Verbessern Sie das Programm **WALL2** so, dass der SENSE sich um eine viereckige Schachtel bewegt.

Legen Sie Hindernisse in den Weg des SENSE und verbessern Sie das Programm so, dass er die Hindernisse umfährt.

Kapitel 2 – Brain-Units

2.1 Brain-Units

Einige Eingabegeräte haben ihr eigenes «Gehirn». Die NeuLog-Sensoren sind solche Brain-Units. Sie senden an die Kontrolleinheit bei Bedarf verarbeitete Daten wie Temperatur (°C oder °F), Lichtintensität in Lux, Distanz in Metern usw.

Die Ausgabegeräte können ebenfalls Brain-Units sein. Z. B die Einheit, welche die Geschwindigkeit und die Richtung des Motors kontrolliert, die Helligkeit einer Lampe, den Winkel eines Servo-Motors usw.

Diese Brain-Units sind in einer Kette verbunden mit der Kontrolleinheit, welche mit ihnen über Meldungen kommuniziert.

Jede Brain-Unit hat eine ID-Nummer. Jede Meldung von der Kontroll-Einheit beginnt mit einer ID-Nummer. Nur die Brain-Unit mit dieser ID-Nummer interpretiert diese Meldung und führt sie aus.

Diese System-Konstruktion ist die Art, wie moderne Systeme aufgebaut sind. Dies hat entscheidende Vorteile:

1. Dieses System braucht viel weniger Kabel. Die Kabel gehen von einem Modul zum nächsten und nicht von jedem Modul zur Kontroll-Einheit.
2. Dieses System kann einfach verändert und ausgebaut werden und hängt nicht ab von der Kapazität an Eingaben und Ausgaben der Kontroll-Einheit.

Die Experimente in diesem Kapitel benutzen die folgenden Brain-Units:

- Geräuschsensor
- Bewegungssensor
- Lichtsensor
- Magnetsensor
- Tracking-Unit
- Greifarm

Wenn Sie diese Sensoren nicht zur Verfügung haben, können Sie sich im Kapitel 3 darüber informieren.

Im Kapitel 3 sind Experimente beschrieben mit dem SENSE-Roboter und dem Batterie-Modul.

2.2 NeuLog-Sensoren als Brain-Units



NeuLog-Sensoren sind auch Brain-Units. Jeder Sensor beinhaltet einen Mini-Computer, der Daten sammelt, speichert und verarbeitet. Jeder Fühler, der mit dem Sensor verbunden ist, wird ab Werk kalibriert und muss nicht mehr geeicht werden.

Die vom Sensor übermittelten Daten sind digital verarbeitet. Der Sensor hat verschiedene Messbereiche. Änderungen des Messbereiches oder der Art der Verarbeitung kann einfach am Computer gemacht werden mit Hilfe der NeuLog-Software.

Die Sensoren werden zusammengesteckt und es gibt fast keine Grenzen in der Zusammenstellung und in der Anzahl der Sensoren.

NeuLog hat 50 verschiedene Sensoren. Einzelne Sensoren funktionieren auch als zwei- bis dreifache Sensoren.

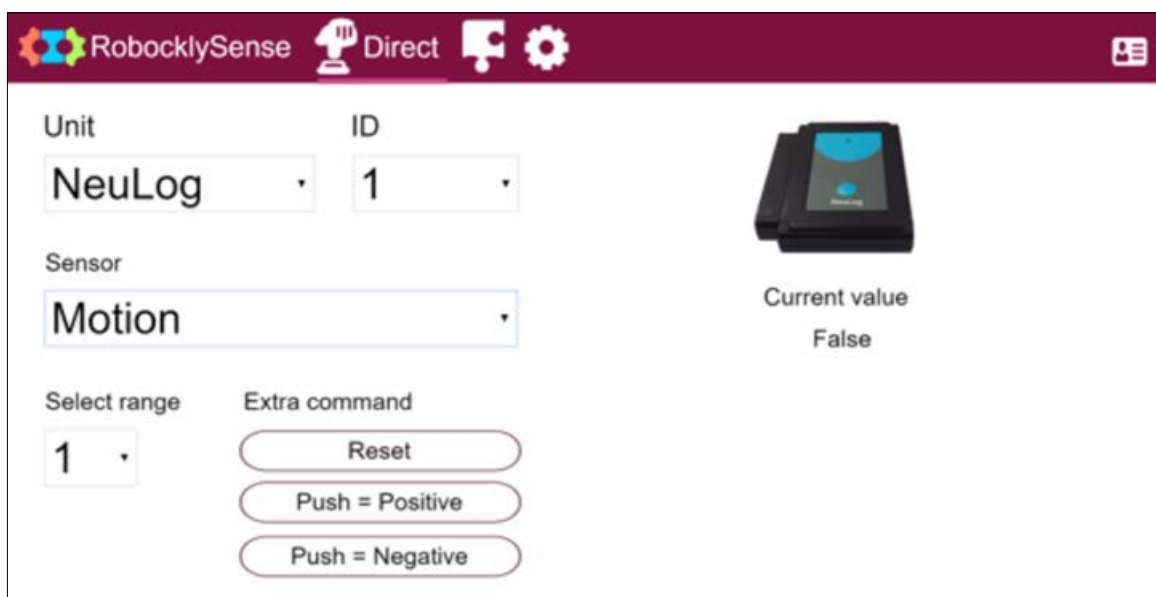
Der SENSE hat drei Steckplätze für NeuLog-Sensoren.


2.3 ID der Brain-Unit ändern

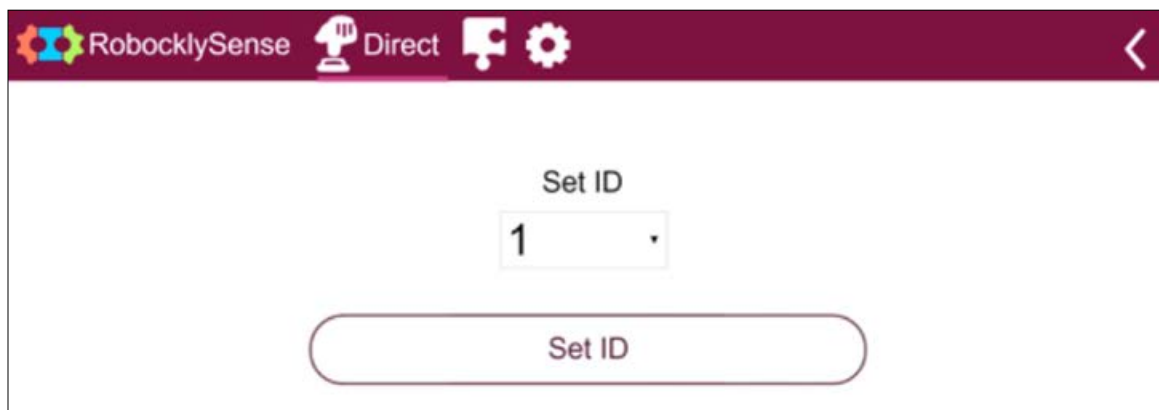
Wie weiter oben erwähnt, hat jede Brain-Unit ihre eigene ID-Nummer. Diese ID-Nummer erlaubt es, dass bis zu 9 Einheiten der gleichen Art verwendet werden. Sie müssen aber darauf achten, dass jede Einheit eine andere ID-Nummer hat.

Im Direkt-Modus gibt es ein spezielles Icon für die Änderung der ID-Nummer jener Unit.

Der folgende Bildschirm zeigt die Werte des Bewegungssensors



Wenn Sie auf das Icon einfügen  rechts oben klicken, sehen Sie den folgenden Bildschirm:



Wenn Sie die ID-Nummer ändern wollen, müssen Sie diese Einheit mit dem PC verbinden. Dann wählen Sie die entsprechende ID im Dropdown-Menü und klicken auf die Schaltfläche **SET ID**.

Experiment 2.1 – Geräuschsensor

Ziele:

- Der Geräuschsensor
- Den SENSE steuern mit einem Geräusch

Notwendiges Material:

- Computer
- RobocklySENSE-Software
- SENSE-Roboter
- USB-Verbindungskabel
- Batterie-Modul oder Powerbank mit Adapter
- NeuLog Geräusch-Sensor

Diskussion:

Der Geräuschsensor benutzt ein eingebautes Mikrofon mit einer Serie von Schaltungen, Filtern und Verstärkern, um eine spezifische Geräuschquelle zu isolieren. Die Schallwellen kommen herein durch ein Loch oben auf dem Plastikgehäuse des Sensors. Für den besten Empfang sollten Sie dieses Loch auf die Schallquelle richten.

Der Geräuschsensor hat zwei Betriebs-Modi:


1. **Analoge arbiträre Einheit (Arb):** Eine arbiträre Einheit zeigt Wellen, Frequenzen und Perioden an. In diesem Modus wird das Geräusch als Signal aufgezeichnet und wieder aufgebaut.
2. **Dezibel (dB):** Die Messeinheit zeigt die Intensität des Geräusches (Lautstärke) an. Beachten Sie, dass dies eine logarithmische Einheit ist.

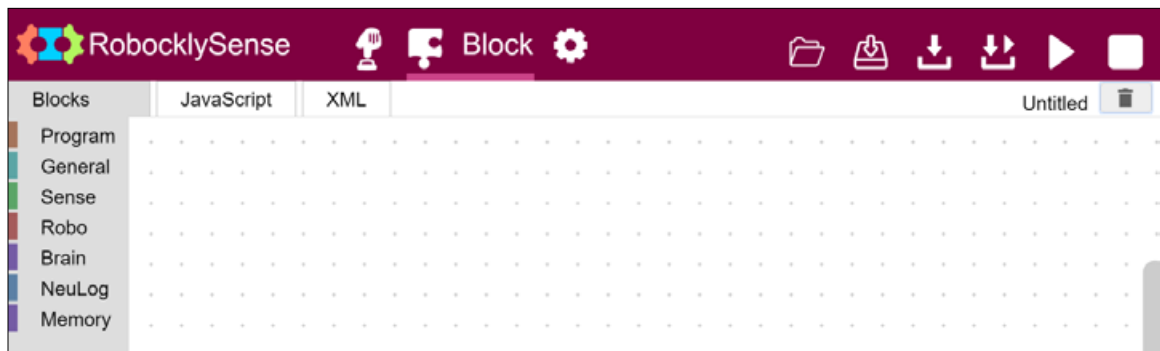
In diesem Modus wird die Schallwelle aufgezeichnet und die durchschnittliche Intensität wird vom Sensor umgerechnet in einen dB-Wert. 40 dB entsprechen Ruhe.

In diesem Experiment werden wir den Sensor im dB-Modus verwenden. Wir weisen dem Sensor die ID 1 als Standard-ID zu.

Die Auswahl des Messbereiches sollte mit der NeuLog-Software vorgenommen werden.

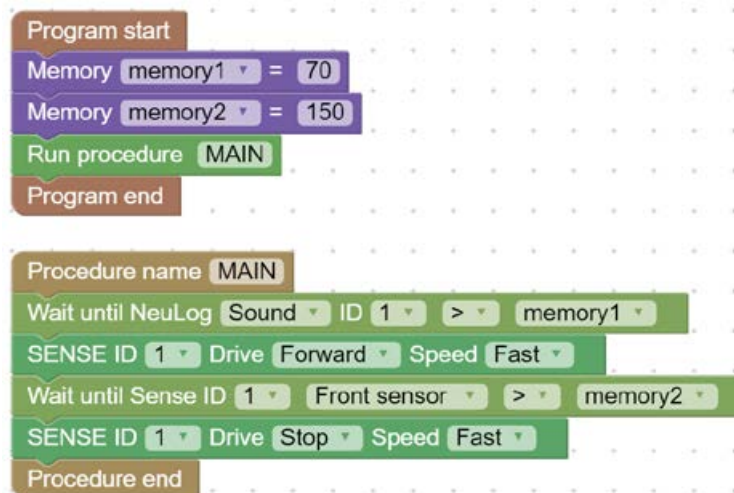
Ablauf:

1. Stecken Sie den Geräuschsensor in einen der Steckplätze des SENSE.
2. Verbinden Sie den SENSE mit dem Computer mit Hilfe eines USB-Kabels.
3. Starten Sie die **RobocklySENSE**-Software.
4. Wechseln Sie in den **Block-Modus**  .



Erstellen Sie ein Programm, das auf ein Geräusch mit 70 dB wartet, den SENSE zu einer Wand bewegt und dann anhält.

5. Bauen Sie das folgende Programm und seine **MAIN**-Abläufe:



6. Studieren Sie das Programm und vergewissern Sie sich, dass Sie alle Befehle verstanden haben.

7. Platzieren Sie den SENSE vor einer Wand und starten Sie das Programm.

Der SENSE sollte sich nicht bewegen.

8. Klatschen Sie in die Hände oder erzeugen Sie ein lautes Geräusch.

Der SENSE sollte sich gegen die Wand bewegen und anhalten.

2.1.1 Übung – Warten auf ein Geräusch

Aufgabe 1:

Verbessern Sie das Programm so, dass ...

- a. ... der SENSE auf ein Geräusch wartet, das lauter ist als 70 dB, dann vorwärts fährt bis an eine Wand und dann anhält.
- b. ... der SENSE wiederum auf das Geräusch wartet, dann rückwärts fährt, bis er eine schwarze Linie erreicht, und dann anhält.
- c. ... zurückkehrt zum Anfang.

Experiment 2.2 – Bewegungssensor

Ziele:

- Der Bewegungssensor als Distanzsensoren
- Bewegung des Roboters gemäss Bewegungssensor

Notwendiges Material:

- Computer
- RobocklySENSE-Software
- SENSE-Roboter
- USB-Verbindungskabel
- Batterie-Modul oder Powerbank mit Adapter
- NeuLog Geräusch-Sensor

Diskussion:

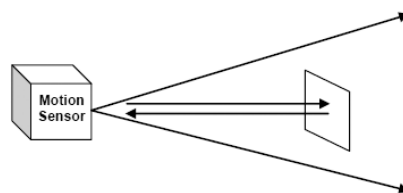
Der Bewegungssensor benutzt einen Ultraschall-Fühler, um sowohl Ultraschallwellen auszusenden als auch das Echo der Wellen zu messen. Objekte im Umkreis von 0,15–10 m können genau gemessen werden, um Distanz, Geschwindigkeit und Beschleunigung anzuzeigen.

Der Bewegungssensor kann Daten in den folgenden Einheiten messen:

- **Meter (m):** Die SI-Distanz-Einheit (internationales Einheiten-System)
- **Meter pro Sekunde (m/s):** Die SI-Geschwindigkeits-Einheit, welche die zurückgelegte Distanz in einer bestimmten Zeit misst
- **Meter /Sekunde² (m/s²):** Die SI-Beschleunigungs-Einheit, welche die Änderung der Geschwindigkeit in einer bestimmten Zeit misst.

Der Bewegungssensor hat zwei Arbeitsbereiche: einer von 0,2–10 m und eine von 0,15–2 m.

Ultraschallwellen werden vom Sensor gesendet und werden in einem trichterförmigen Muster verteilt, ca. 15° rund um den Referenzpunkt.



Der Ultraschall-Wandler ist ein Gerät, das elektronische Impulse in Ultraschall-Impulse umwandeln kann. Das Gerät sendet die Ultraschall-Impulse und empfängt deren Echo. Entweder das Gerät selber oder ein ähnliches Gerät wandelt diese Echo-Impulse wieder um in elektronische Impulse.

Der Ultraschall-Wandler basiert auf einem keramischen Kristall, der auf eine bestimmte Weise geschnitten ist und der zwischen zwei Metallplatten liegt. Der Kristall ist charakterisiert durch den piezoelektrischen Effekt. Das elektrische Feld ändert sich zwischen den Platten und erzeugt eine mechanische Vibration im Kristall.

Der Kristall hat eine bestimmte Resonanz-Frequenz. Die mechanischen Vibrationen und die elektrischen Reaktionen hängen von dieser Resonanz-Frequenz ab.

Wenn der Kristall vom Ultraschall-Wandler Impulse erhält, beginnt er zu vibrieren und übermittelt diese Impulse als akustisches Signal. Diese Töne kann man nicht hören, weil sie über der Hörfrequenz liegen (normalerweise 40 KHz).

Der akustische Ton kann wiederum konvertiert werden in elektronische Impulse, entweder durch einen anderen Ultraschall-Wandler oder durch den Übermittler, wenn die Übermittlung aufhört. Die akustischen Impulse lassen den Wandler vibrieren und diese Vibrationen werden umgewandelt in Stromimpulse.

Die Geschwindigkeit einer Ultraschallwelle beträgt ca. 300 m/s, weil es eine Schallwelle ist.

Für Distanzmessungen wird eine Impulsfolge gesendet und das System misst die Zeit zwischen dem Senden und dem Empfangen.


$$S = 300 \cdot t$$

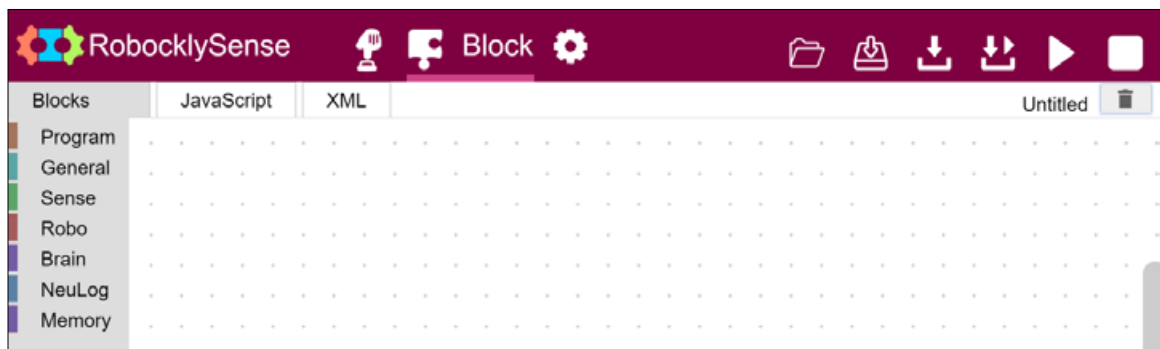
Die Geschwindigkeit wird berechnet, indem die Differenz zweier aufeinanderfolgender Distanzmessungen dividiert wird durch die Zeit zwischen den Abtastungen (abhängig von der Abtastrate).

Der Bewegungssensor benützt ein hochentwickeltes System, das ihm erlaubt, lange Distanzen mit schwachen Impulsen zu messen.

In diesem Experiment benutzen wir den Sensor als Distanzmesser und wir weisen dem Sensor die ID 1 als Standard-ID zu. Die Wahl des Messbereiches sollte in der NeuLog-Software vorgenommen werden.

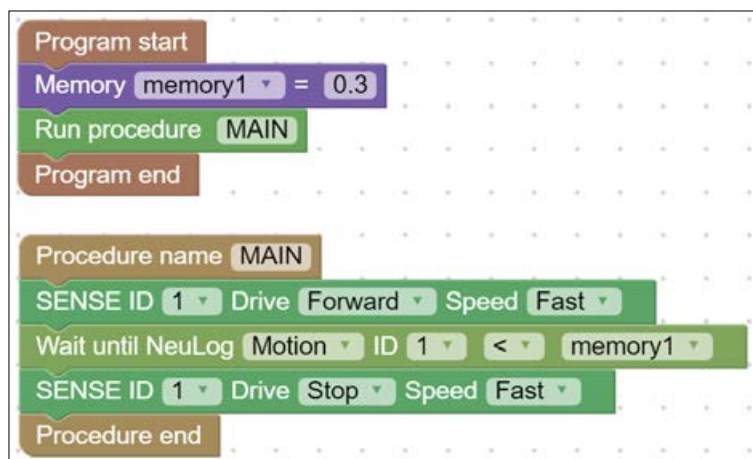
Ablauf:

1. Stecken Sie den Bewegungssensor in einen der Steckplätze des SENSE, mit dem Wandler direkt nach vorne.
2. Verbinden Sie den SENSE mit dem Computer mit Hilfe eines USB-Kabels.
3. Starten Sie die **RobocklySENSE**-Software.
4. Wechseln Sie in den **Block**-Modus .



Erstellen Sie ein Programm, sodass sich der SENSE vorwärts bewegt zu einer Wand und 30 cm vor der Wand stoppt.

5. Bauen Sie das folgende Programm und seinen **MAIN**-Ablauf.



6. Studieren Sie das Programm und vergewissern Sie sich, dass Sie alle Befehle verstanden haben.
7. Legen Sie den Sensor vor eine Wand und starten Sie das Programm. Der SENSE sollte sich vorwärts Richtung Wand bewegen und 30 cm davor anhalten.

2.2.1 Übung – Bewegung in einem Distanzbereich

Vorwärts fahren zu einer Wand, 30 cm vor der Wand stoppen, rückwärts fahren und 50 cm vor der Wand stoppen und zurück zum Anfang.

Aufgabe 1:

Verbessern Sie das Programm so, dass der SENSE...

- ... sich gegen eine Wand bewegt,
- 30 cm vor der Wand stoppt,
- 2 Sekunden wartet,
- sich rückwärts bewegt bis zu einer Distanz von 60 cm,
- für 2 Sekunden anhält,
- zum Ausgangspunkt zurückkehrt.

Experiment 2.3 – Sucher-Einheit

Ziele:

- Die Sucher-Einheit
- Bewegung zu einem IR (Infrarot)-Geber
- Einem IR-Geber folgen

Notwendiges Material:

- Computer
- RobocklySENSE-Software
- SENSE-Roboter
- USB-Verbindungskabel
- Batterie-Modul oder Powerbank mit Adapter
- Sucher-Einheit
- IR-Geber

Diskussion:

2.3.1 IR-Geber

Der Infrarot-Geber kann in eine Steckverbindung auf dem SENSE eingesteckt werden oder in einen Steckplatz an der Reserve-Batterie, gefolgt von der Sucher-Einheit.

Infrarotes Licht ist ein Licht, das von einer Wärmequelle ausgesendet wird. Wir können Infrarot-Licht nicht wahrnehmen. Die Frequenz dieses Lichtes ist ein bisschen tiefer als bei rotem Licht und deshalb nennen wir es infra (unter) rot.

Das Umgebungslicht beeinflusst das Infrarotlicht kaum.

2.3.2 Sucher-Einheit

Die Sucher-Einheit, in einem stabilen Plastikgehäuse, kann in die SENSE-Steckverbindungen eingesteckt werden.

Die Sucher-Einheit hat drei Infrarot-Sensoren, die ihr ermöglichen, dem IR-Geber zu folgen.

Zwei IR-Sensoren liegen auf einer Linie mit einer lichtundurchlässigen Trennung dazwischen.

Wenn IR-Licht auf beide Sensoren trifft, heisst das, dass der SENSE genau vor der IR-Lichtquelle steht.


Wenn der SENSE in einem bestimmten Winkel zur Lichtquelle steht, wird das IR-Licht nur auf einen der beiden Sensoren treffen.

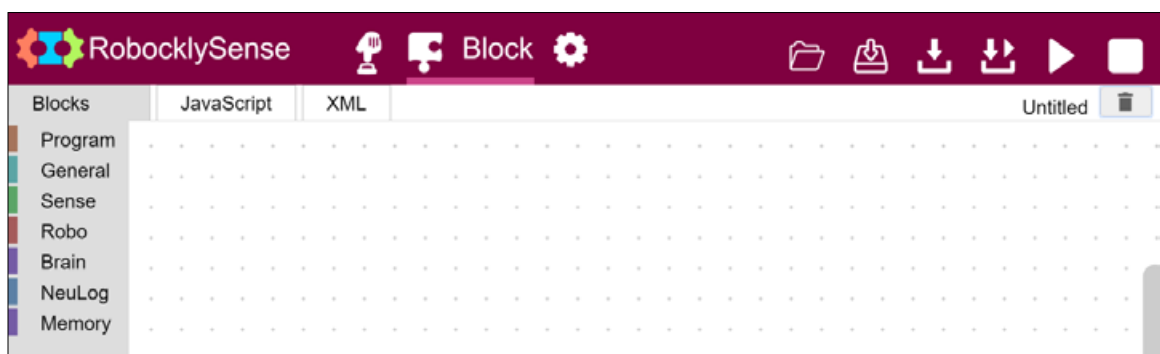
Der dritte IR-Sensor misst den IR-Anteil des Umgebungslichtes. Die Steuerung der Sucher-Einheit benutzt diese Messung, um das Umgebungslicht zu eliminieren.

Die Einheit gibt eine (binäre) Zahl aus, die den Status de IR-Gebers folgendermassen ausdrückt:

- 0 (00) = Kein Licht vom IR-Geber
- 1 (01) = Licht vom IR-Geber von recht
- 2 (10) = Licht vom IR-Geber von links
- 3 (11) = Licht vom IR-Geber von vorne

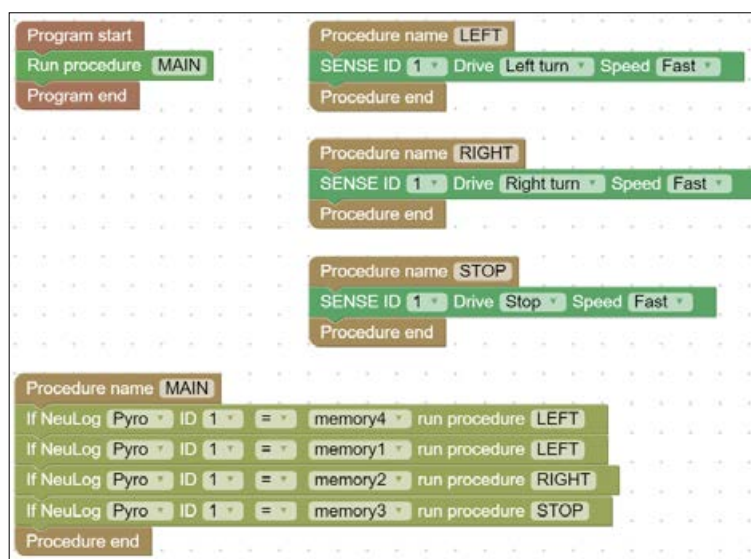
Ablauf:

1. Stecken Sie die Sucher-Einheit in den vorderen Steckplatz des SENSE.
2. Stecken Sie den IR-Geber in eine Reserve-Batterie. Legen Sie die Batterie auf eine 3 cm hohe Oberfläche.
3. Verbinden Sie den SENSE mit dem PC mit Hilfe eines USB-Kabels.
4. Starten Sie die **RobocklySENSE**-Software.
5. Wechseln Sie in den **Block-Modus**  .

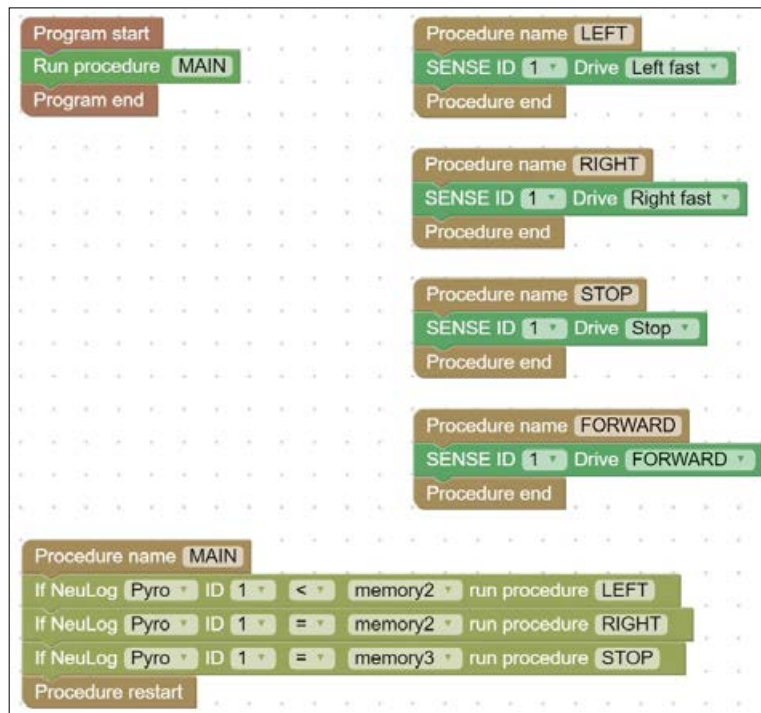


Erstellen Sie ein Programm, das den SENSE nach links dreht, bis er den IR-Geber «sieht» und diesem dann folgt, ohne sich zu bewegen (er dreht sich einfach).

6. Bauen Sie das folgende Programm und seine **MAIN**-Abläufe.



7. Studieren Sie das Programm und vergewissern Sie sich, dass Sie alle Befehle verstanden haben.
8. Laden Sie das Programm herunter.
9. Platzieren den SENSE auf dem Boden und starten Sie das Programm.
Der SENSE sollte nach links drehen, bis er den IR-Strahl «sieht».
10. Bewegen Sie den IR-Geber langsam und überprüfen Sie, dass der SENSE ihm folgt.
11. Ändern Sie das Programm wie folgt:



12. Dieses Programm sollte gleich funktionieren wie das vorherige. Überprüfen Sie das.

2.3.3 Übung – Mit einem Roboter einem IR-Geber folgen

Aufgabe 1:

Erstellen Sie ein Programm mit seinen Abläufen, sodass der SENSE...

- ... sich dreht, bis er ein IR-Licht findet,
- sich zum IR-Geber bewegt,
- vor dem Geber stoppt.

Kapitel 3 – Übungen zu selbstfahrenden Fahrzeugen

3.1. Selbstfahrende Fahrzeuge

Wir leben in der Generation der selbstfahrenden Autos, der lernfähigen Maschinen und der künstlichen Intelligenz. Dies ist die Welt der Maschinen, die Entscheidungen fällen. Diese Entscheidungen erfolgen gemäss der Software und der Programmierung dahinter. Dies ist aber erst der Anfang.

Wir können diese Welt und die laufenden Veränderungen begreifen, wenn wir Programme entwickeln, die denen von selbstfahrenden Autos ähnlich sind.

Der SENSE ist ein Werkzeug für solche Übungen.

Diese Kapitel bietet verschiedene Übungen. Die Idee dahinter ist, dass der Benutzer über Algorithmen nachdenkt und nach Lösungen für diese Herausforderungen sucht.

3.2 Programmiersprachen

RobocklySense ist ein sehr gutes Programm für den Beginn. Es ist einfach, bietet aber sehr viele Möglichkeiten.

Für komplexe Programme, besonders mit teilweise künstlicher Intelligenz, braucht man andere Software.

Wir können dem SENSE Codier-Einheiten hinzufügen für Python, C oder Arduino. Damit kann der SENSE gesteuert werden. Jede Codier-Einheit wird mit einer Gebrauchsanleitung geliefert.

Dieses Kapitel bietet Übungen, die Sie ohne Anleitung lösen sollen.

Es gibt immer mehrere Lösungen. Finden Sie die effizienteste Lösung.

Haben Sie keine Angst davor zu scheitern, versuchen Sie es einfach immer wieder.

Viel Glück!!

3.1 Übung – Entlang schwarzer Linien

Verbessern Sie das Programm aus 1.5. für die Bewegung entlang einer schwarzen Linie so, dass die Bewegung weicher wird.

Benutzen Sie den Deviate-Befehl anstatt des Turn-Befehls, wenn der SENSE nahe am Wert für die schwarze Linie ist.

Benutzen Sie den Turn-Befehl, wenn der SENSE weit weg vom Wert für die schwarze Linie ist.

Überprüfen Sie im direkten Modus den Wert des Bottom-Sensors, wenn der SENSE auf der Mitte der schwarzen Linie ist und wenn er näher an der der Kante der Linie ist.

3.2 Übung – Automatic guided Vehicle (AGV)

Ein AGV ist ein Fahrzeug oder ein Wagen, der sich entlang von Führungslinien bewegt. In der Industrie werden diese Wagen verwendet für den Transport von Rohmaterialien oder Bauteilen von einem Ort zum andern.

Erstellen Sie die folgende Linie:



Schreiben Sie ein Programm, das den SENSE entlang der Linie bewegt und beim grauen Quadrat 5 Sekunden lang anhält.

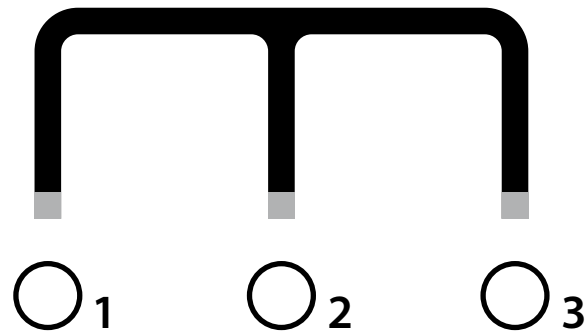
Der SENSE bewegt sich an der äusseren Kante und im Gegenuhrzeigersinn.

Überprüfen Sie im direkten Modus den Wert des Bottom-Sensors, wenn der SENSE auf der schwarzen Linie ist und wenn er auf dem grauen Quadrat ist.

Aufgabe: Ändern Sie das Programm so, dass sich der SENSE entlang der inneren Kante der Linie und im Gegenuhrzeigersinn bewegt.

3.3 Übung – AGV zwischen Positionen

Erstellen Sie die folgende Linie:



Schreiben Sie ein Programm, sodass sich der SENSE sich von einer Station zur nächsten bewegt in der folgenden Reihenfolge 1-2-3-1-2-3...



Der SENSE stoppt an jeder Station und bewegt sich weiter zur nächsten Station, sobald Sie eine kleine Schachtel vor den SENSE hinlegen.

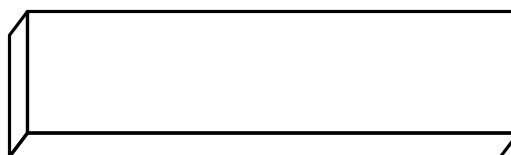
3.4 Übung – Entlang einem Gebäudeblock

Verbessern Sie das Programm aus dem Kapitel 1.5. (Bewegung entlang einer Wand) mit kleinen Veränderungen.

Benutzen Sie die den Deviate-Befehl anstatt des Turn-Befehls, wenn der SENSE kurz vor der geforderten Distanz von der Wand ist.

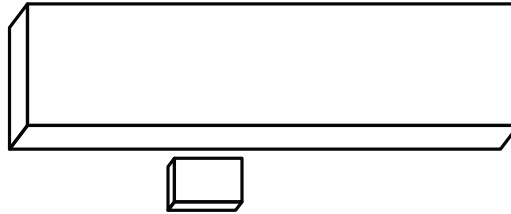
Benutzen Sie die den Turn-Befehl, wenn der SENSE weit weg von der geforderten Distanz zur Wand ist.

Benutzen Sie eine lange Schachtel als Simulation des Gebäudeblocks.



3.5 Übung – Entlang einem Gebäudeblock und um einen Wagen herum

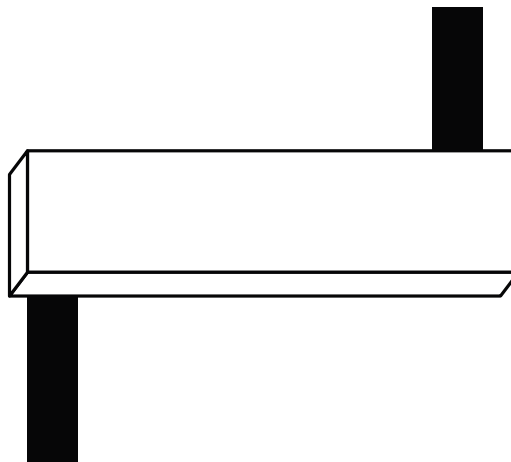
Legen Sie ein Hindernis hin wie im Bild gezeigt.



Schreiben Sie ein Programm, wie in Übung 2.4, sodass der SENSE um den Wagen herum fährt. Der SENSE soll nach dem Passieren des Wagens wieder nach rechts fahren.

3.6 Übung – Entlang einem Gebäudeblock mit einem Stoppzeichen.

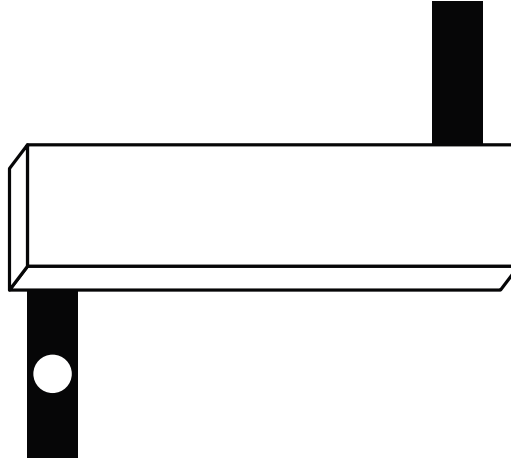
Setzen Sie an die Ecken der Schachtel schwarze Linien wie im folgenden Bild:



Schreiben Sie ein Programm, wie in Aufgabe 2.4, sodass der SENSE an der schwarzen Linie 3 Sekunden lang anhält.

3.7 Übung – Entlang einem Gebäudeblock mit einem Stopp für einen Fussgänger.

Setzen Sie an die Ecken der Schachtel schwarze Linien wie im folgenden Bild:

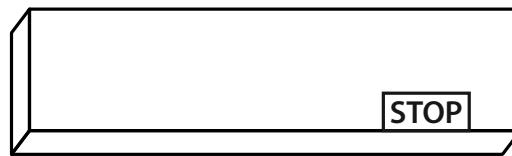


Schreiben Sie ein Programm, wie in Aufgabe 2.4., sodass der SENSE an der schwarzen Linie 3 Sekunden lang anhält.

Der SENSE fährt nicht weiter, wenn ein Hindernis vor ihm liegt.



3.8 Übung – Wächter beim Gebäudeblock

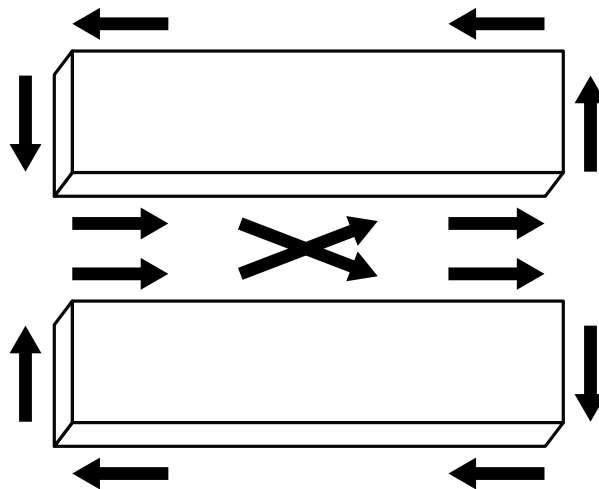


Schreiben Sie ein Programm, wie in Aufgabe 2.4, sodass der SENSE nach jeder Runde 5 Sekunden lang anhält.

Das Programm soll die Runden zählen.



3.9 Übung – Zwei Wächter beim Gebäudeblock

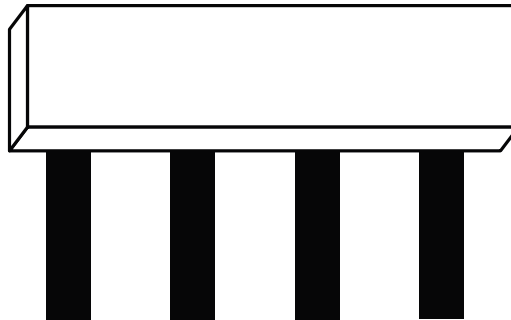


Schreiben Sie ein Programm, sodass der SENSE rund um die beiden Gebäude fährt wie im obigen Bild

Das Programm soll die Eckenrundungen zählen und wechseln zwischen Uhrzeigersinn um das eine Gebäude und Gegenuhrzeigersinn um das andere Gebäude.

3.10 Übung – Taxifahrer

Legen Sie vier schwarze Linien vor das Gebäude wie abgebildet.

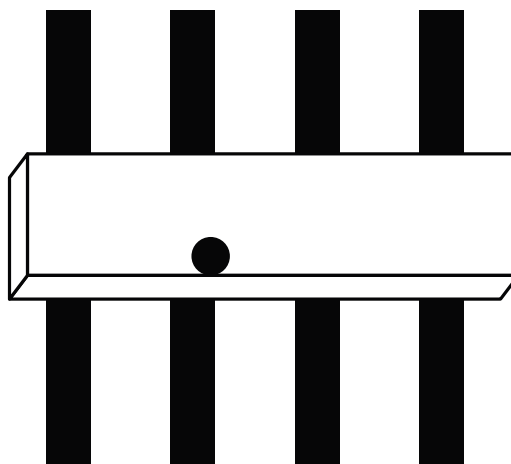


Schreiben Sie ein Programm, sodass der SENSE sich entlang des Gebäudes bewegt und bei der dritten schwarzen Linie stoppt.

Starten Sie den SENSE auf der anderen Seite des Gebäudes.

3.11 Übung – Taxifahrer mit Passagier

Legen Sie schwarze Linien um das Gebäude wie abgebildet.



Schreiben Sie ein Programm, sodass der SENSE sich entlang des Gebäudes bewegt und bei der dritten schwarzen Linie fünf Sekunden lang stoppt. Danach bewegt sich der SENSE weiter auf die andere Seite des Gebäudes und stoppt auf der zweiten schwarzen Linie.



SENSE Robot