

Robotik Workshop



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Was machen wir heute?

Einführung in Robotik? Was sind Roboter?

Wie programmiert man Roboter?

Kleine Wettbewerbs-Aufgaben lösen.



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Was macht einen Roboter aus?

Welche Roboter kennt ihr?

Welche habt ihr vielleicht sogar zu Hause ?



CCA – COMPETENCE CENTRE
HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

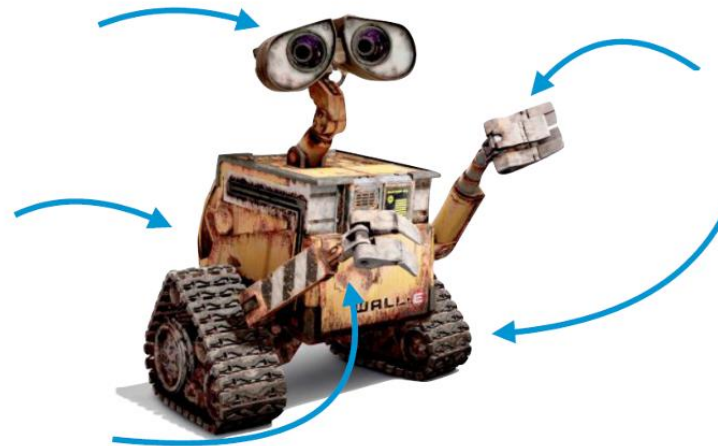


Roboter Teile

Diskutiert was an eurem Roboter alles verwendet wird

Tipp: es gibt folgende Teile:

- Motoren
- Akku
- Sensoren
- Controller
- Lego, Metallteile



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Roboter Komponenten: Lego- und Metallteile

Lego Metallteile

- Bietet dem Roboter Halt, wie ein Skelett
- An den Gelenken am Roboter sind normalerweise Aktoren (Motoren), vergleichbar mit deinen Muskeln, angebracht
- Hält die Sensoren in Position



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Roboter Komponenten: Motoren

Motoren werden:

- benutzt um den Zustand des Roboters selbst zu verändern
- benutzt um den Zustand der Welt herum zu verändern

Beispiele:

- Geradeaus Fahren
- Drehen



CCA – COMPETENCE CENTRE

HTL Anichstraße

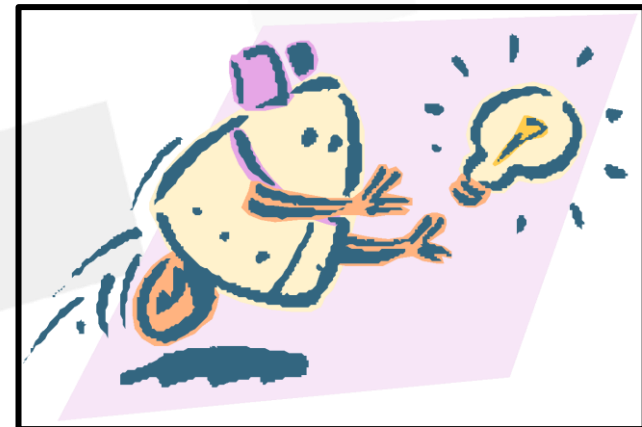
ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Roboter Komponenten : Sensoren

Berichte über den aktuellen Zustand der Umgebung, in der sich der Roboter befindet. Vergleichbar mit den Menschlichen Sinnesorganen
Zum Beispiel : Lichtsensoren, Entfernungssensoren, Berührungssensoren, usw.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

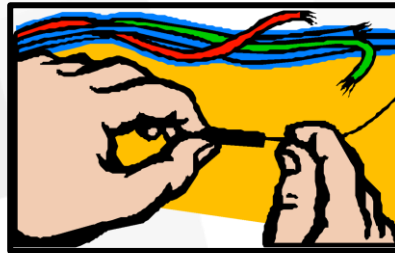
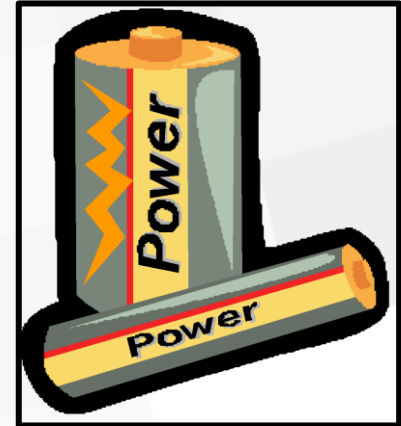
Roboter Komponenten: Akku

Stromquelle

- Akku, Batterie, Solarzellen
- Federn, Hydraulik, Pneumatik
- Windkraftwerk

Stromverteilung

- Kabeln



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

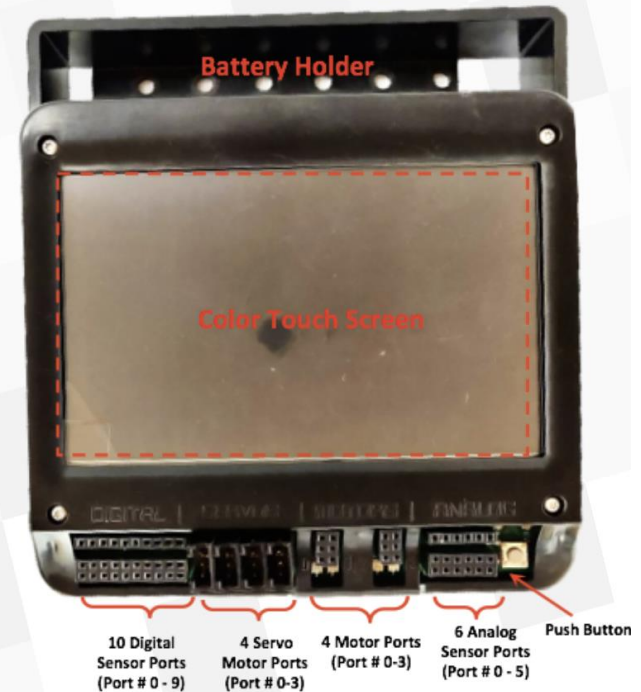
©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Roboter Komponenten: Controller

Der Controller:

- ist das “Gehirn” des Roboters
- verarbeitet die Informationen der Sensoren
- hat das Programm gespeichert
- steuert die Motoren und Servomotoren



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Match it Up!

Material: Gedrucktes "Match it Up"-Aktivitätsblatt für jede(n) Schüler:innen

1. Nachdem ihr die Definitionen der Roboter-Teile habt, gibt es nun die "Match it Up" Aufgabe.
2. Kannst du sie zuordnen und begründen, warum?



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Verbinde! – Match It Up!

Menschen

- Knochen
- Muskeln
- Sinne
- Gehirn
- Nahrung
- Wissen

Roboter

- Computer
- Strom/Energie
- Computerprogramm
- Sensoren
- Motoren und Servomotoren
- Mechanische Strukturen



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Programmiersprache



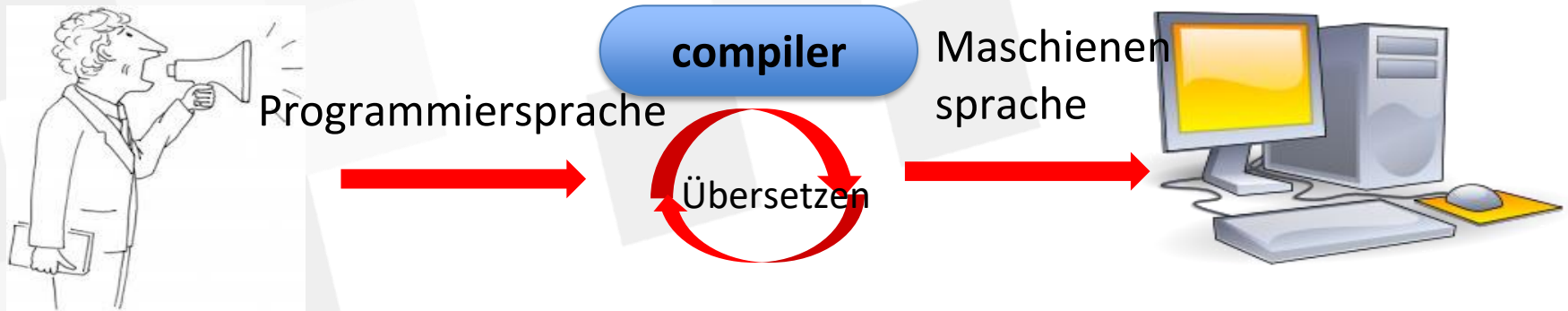
Computer verstehen nur Maschinensprache (Einsen und Nullen), die sie dann lesen und ausführen können.

Der Mensch hingegen kommt mit Maschinensprache nicht gut zurecht.

Wieso also nicht Interpreter verwenden?

Die Menschen haben Sprachen mit lustigen Namen wie C, C++, JAVA, Python entwickelt, die es ihnen ermöglichen, "Quellcode" zu schreiben, den sie verstehen und bearbeiten können.

Dieser Quellcode wird dann in Maschinensprache kompiliert (übersetzt), die der Computer verstehen und ausführen kann.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Begriffe beim Programmieren

- o **Maschinensprache** -- Versteht der Computer Bytes
- o **Ausführen** -- im Falle vom Computer Anweisungen ausführen
- o **Source Code** -- Bezeichnung für in einer Programmiersprache geschriebenen Code
- o **Kompiliert** -- aus einer Programmiersprache in eine Maschinensprache übersetzt
- o **Programmiersprache** -- Sprache, die Menschen verstehen und die in Maschinensprache umgewandelt werden kann
- o **C, C++, Java, Python** -- Name von Programmiersprachen
- o **IDE** -- Integrated Development Environment - Umgebung in der man programmiert



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

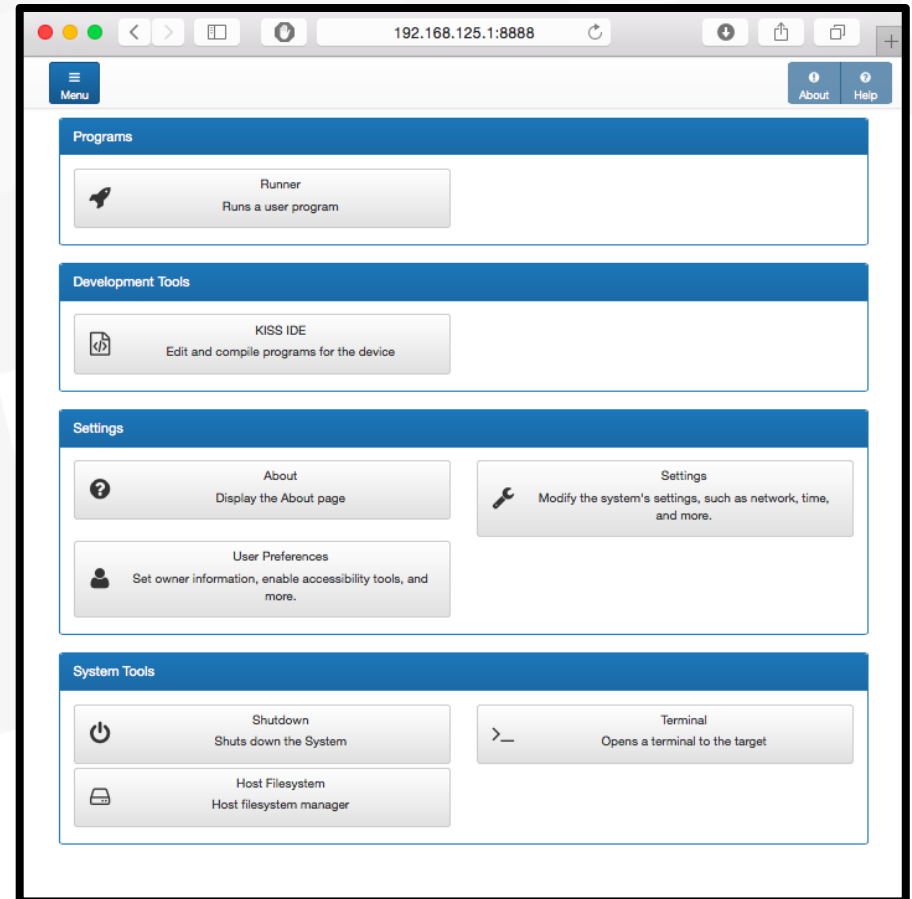
©KISS Institute for Practical Robotics, 1997- 20



**KISS
INSTITUTE FOR
PRACTICAL
ROBOTICS**

KIPR Software Suite

Wir werden die KIPR Software Suite verwenden, um Quellcode in der Programmiersprache C zu entwickeln. Die KIPR Software Suite ist eine Softwareanwendung, die es uns leicht macht, den Quellcode zu schreiben, zu bearbeiten, und ihn zu debuggen (nach Fehlern zu suchen und zu korrigieren) und zu kompilieren (zu übersetzen).



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Ein Programm im Überblick:

`int main`

- Das `int main` ist das Hauptprogramm, `int` steht für Integer(Ganzzahl)
- Die geschwungenen Klammern klammern die Programmieranweisungen ein
- Programme müssen immer etwas zurückgeben, in diesem Fall eine `0`

```
int main()  
{  
    return 0;  
}
```



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

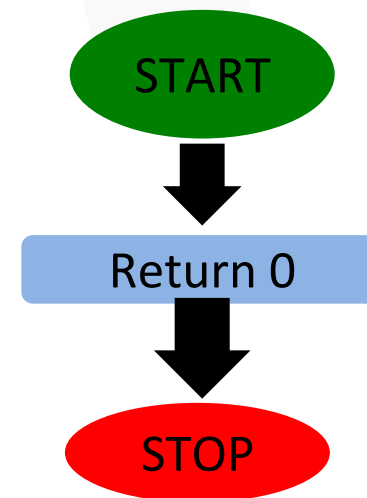
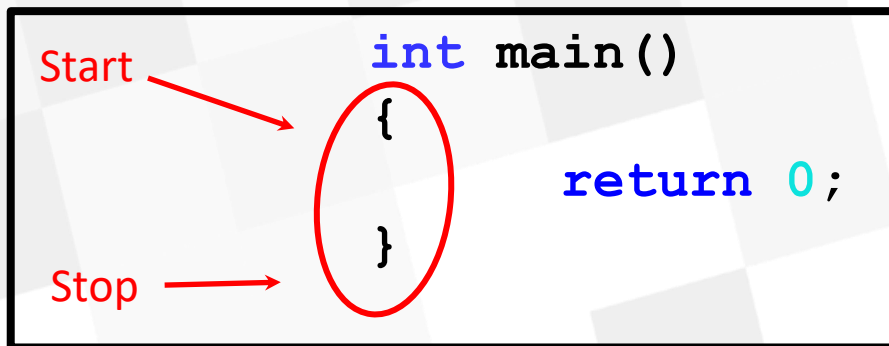
©KISS Institute for Practical Robotics, 1997- 2023



Ein Programm im Überblick:

Geschwungene Klammern

Die geschwungenen Klammern organisieren alles, was das Programm tun (ausführen) soll. Wenn der Computer die letzte geschwungene Klammer erreicht, beendet er das Hauptprogramm. Die Aktionen, die der Roboter ausführen soll, stehen zwischen den geschweiften Klammern.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

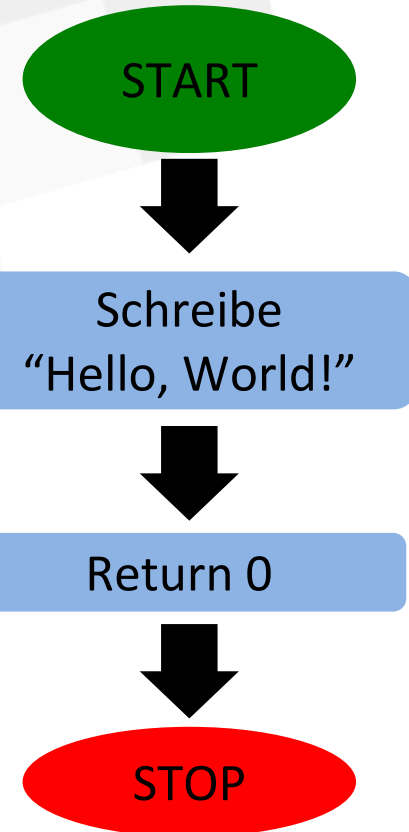


Ein Programm im Überblick:

Code

Das ist der Code, der angibt, welche Dinge (Funktionen) das Programm ausführen soll.

```
int main()  
{  
    printf("Hello, World!\n");  
    return 0;  
}
```

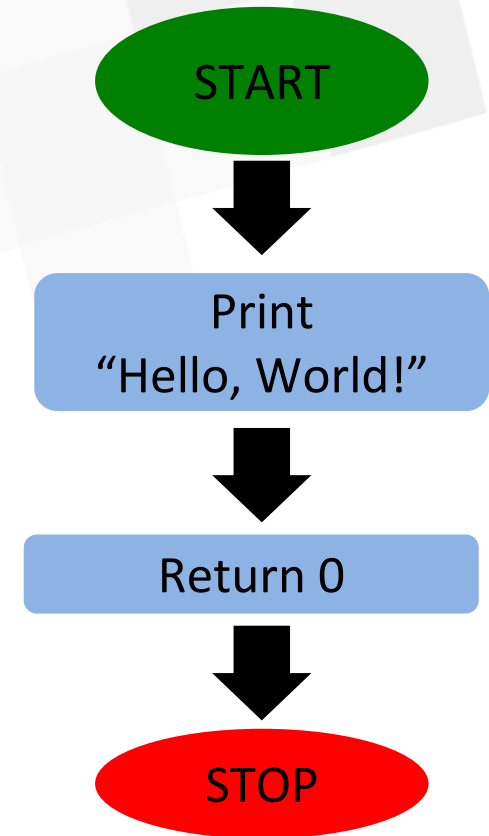


Ein Programm im Überblick:

return

Beachte, dass das Programm einen Wert zurückgibt, obwohl er 0 ist.

```
int main()  
{  
    printf("Hello, World!\n");  
    return 0;  
}
```



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



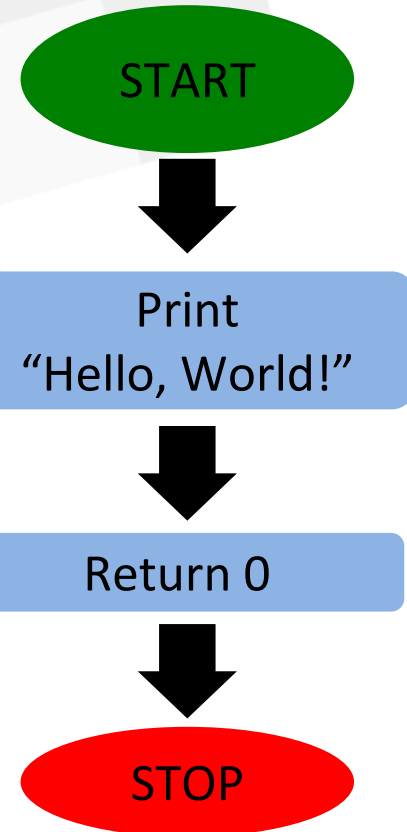
Ein Programm im Überblick:

Semikolons

Wenn das Programm ausgeführt wird, beendet das Semikolon den Code und sagt: "Gehe zur nächsten Zeile". Ohne das Semikolon wird der Code nicht kompiliert (übersetzt, so dass der Computer ihn verstehen kann).

```
int main()  
{  
    printf("Hello, World!\n");  
    return 0;  
}
```

Semikolons beenden den Code



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Ein Programm im Überblick:

Farben

Die KISS IDE hebt Teile eines Programms hervor, um es leichter lesbar zu machen

- Standardmäßig färbt die KISS IDE Ihren Code und fügt Zeilennummern hinzu
- Kommentare sind grün
- Schlüsselworte sind dick blau
- lesbarer Text ist rot
- Zahlen sind türkis

KISS's Instructional Software System - No Target

File Edit Source Target Developer Help

File Project Open Copy Cut Paste Compile Download

Start Page *Untitled

```
1 // Created on Thu January 10 2013
2
3 int main()
4 {
5     printf("Hello, World!\n");
6     return 0;
7 }
```

```
int main()
{
    printf("Hello, World!\n");
    return 0;
}
```



Identifiziere die Teile eines Programms

1. Bespreche die Folien "Ein Programm im Überblick".
2. Beschrifte und erkläre, was die folgenden Teile des Beispielprogramms auf der nächsten Folie bewirken.
 - Geschwungene Klammern
 - Semikolons
 - Main Funktion
 - Rückgabe
 - Quellcode (Programmieranweisungen)
 - Farbe des Textes im Code



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Beispiel Programm

```
#include <kipr/wombat.h>
int main()
{
    printf("Hello, World!\n");

    motor(3,100);
    motor(0,100);
    msleep(3800);           //vorwärts

    motor(3,100);
    motor(0,25);
    msleep(1000);          //rechts drehen

    return 0;
}
```



CCA - COMPETENCE CENTRE

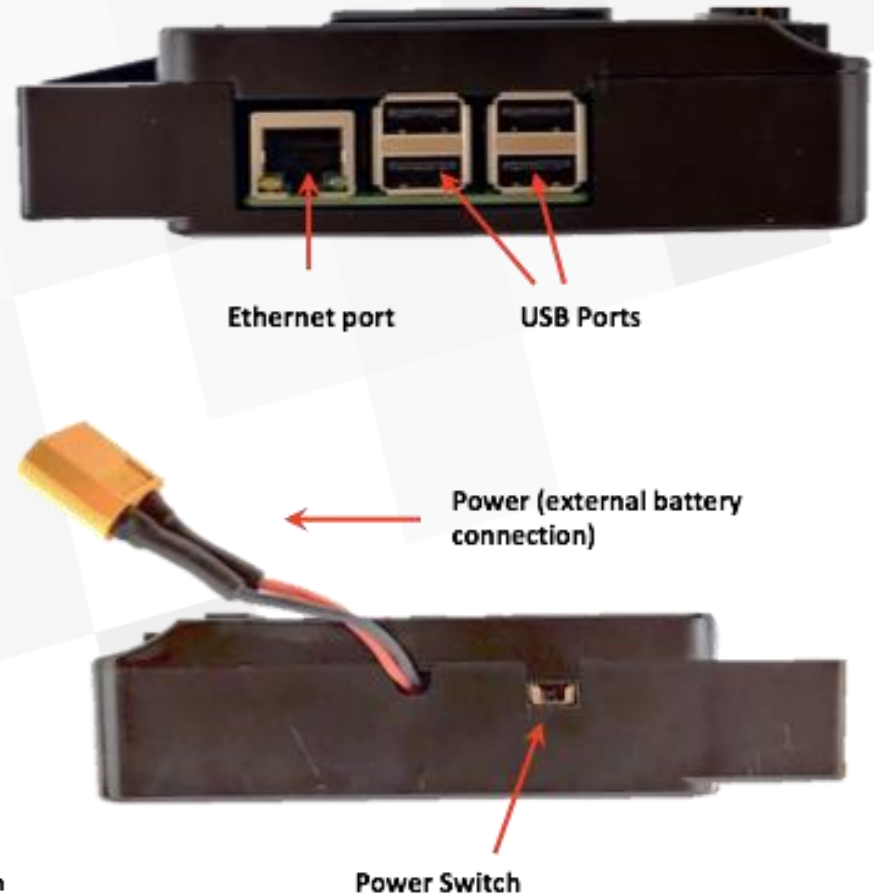
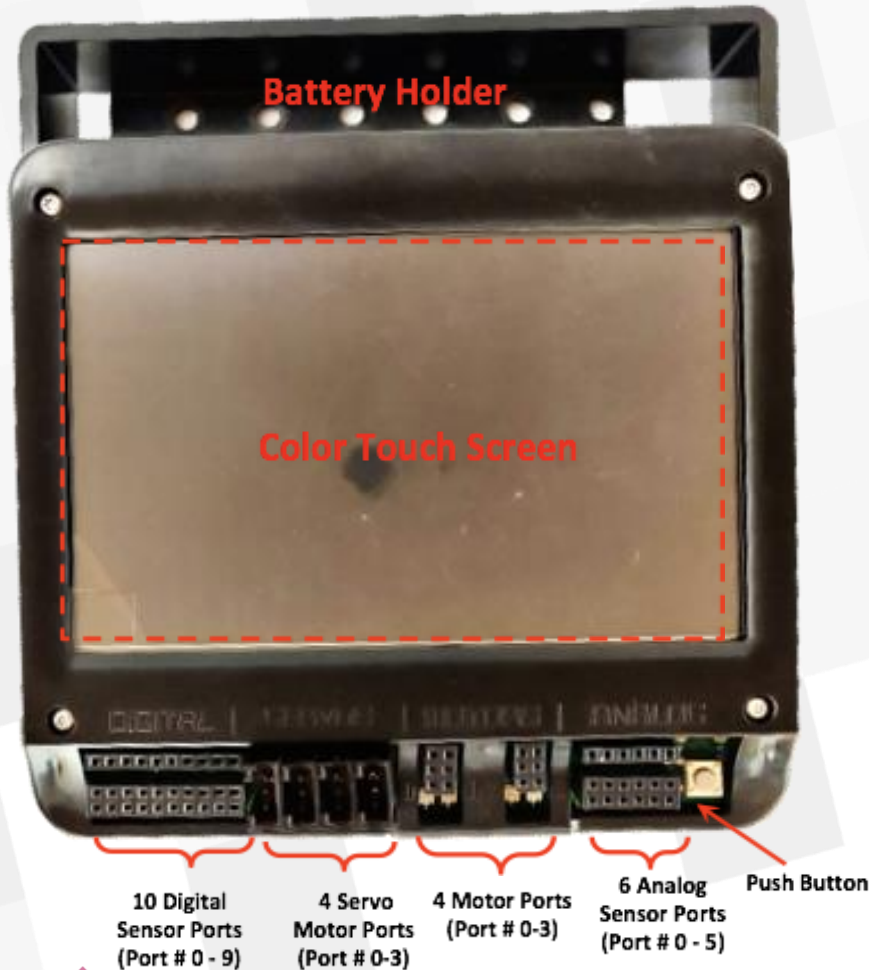
HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Wombat Controller Guide

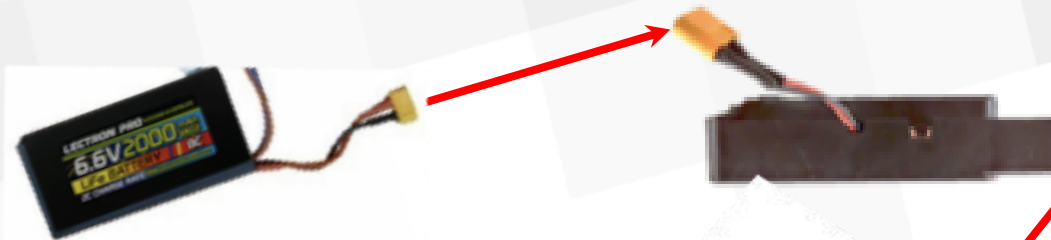


Batterie richtig anstecken

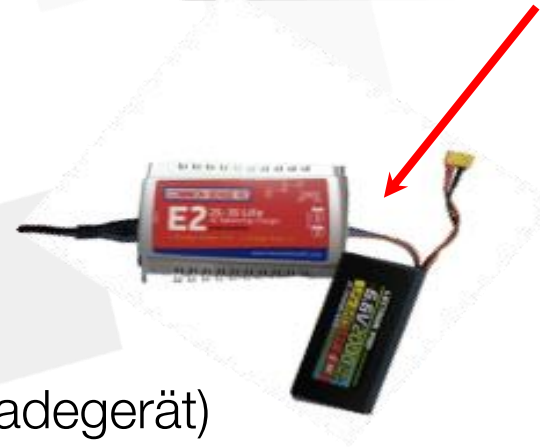
Material

Die Verbindungen sind wie folgt

- **gelb zu gelb**(Batterie zum Controller)



- **Weiß klein zu weiß kleinen Stecker**



- **Schwarz an Schwarz** (Ladekabel an Ladegerät)



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

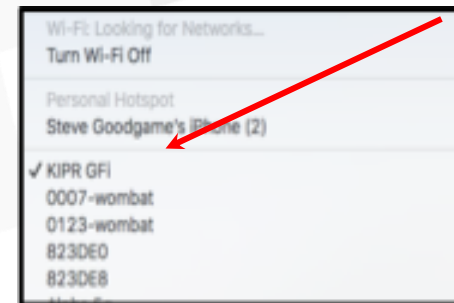
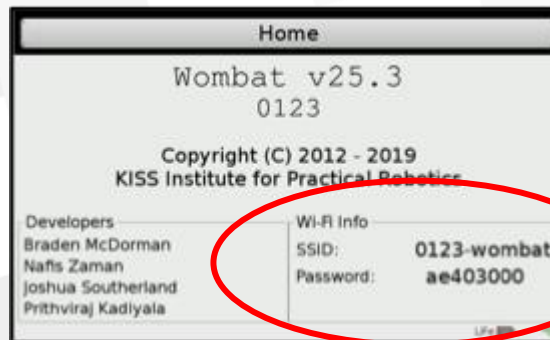
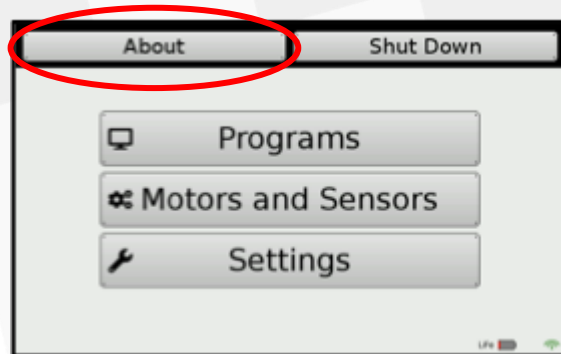
©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Verbindung zum Wombat herstellen mittels WLAN

Verbinde den Wombat über Wi-Fi mit deinem Browser

1. Schalte den Wombat mit dem schwarzen Schalter an der Seite ein (nach dem Einschalten, warte, bis du deinen Wombat als verfügbares mit WLAN Gerät findest. Dies sollte etwa eine Minute dauern)



2. Verwende die Informationen (Wombat SSID# und Passwort) von der Info-Seite, um sich über WLAN zu verbinden.



CCA – COMPETENCE CENTRE

HTL Anichstraße

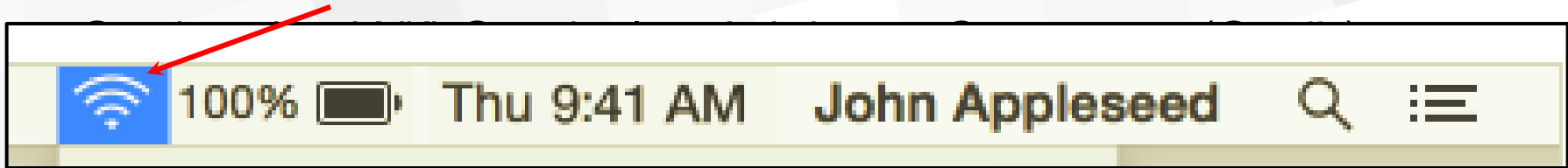
ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Verbindung zum Wombat herstellen mittels WLAN

Wie finde ich meine Wifi-Einstellungen auf meinem Computer?



2. Dieses kann sich oben oder unten auf deinem Computer (Gerät) befinden.
3. Klicke auf das Symbol.
4. Fahre mit der nächsten Folie fort.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Verbindung zum Wombat herstellen mittels Wlan



Gehe zum WiFi-Symbol auf deinem Computer und klicke auf deine Wombat-Nummer und füge das Passwort hinzu (dies kann einige Minuten dauern).

Du siehst eine grüne Wlan Anzeige auf deine Wombat Controller, wenn Wlan eingeschaltet ist.



Wenn du mit dem Wombat verbunden sind, kann dein Gerät (Computer) verschiedene Fehlermeldungen anzeigen: "Keine Internetverbindung" oder "Verbindung ist eingeschränkt".

Dies ist normal. Fahre fort mit dem Öffnen eines Browsers und der Verbindung zur KISS IDE.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

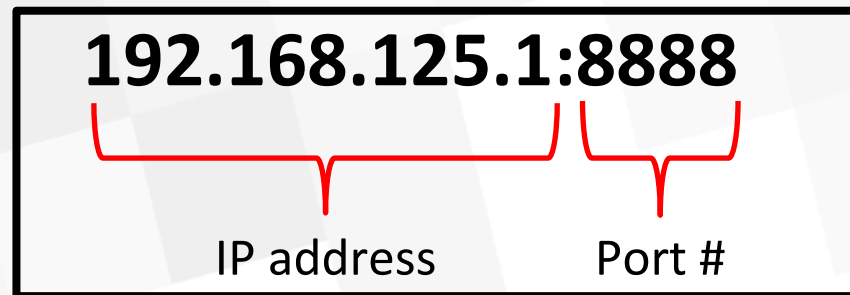
Fahre mit der nächsten Folie fort.

©KISS Institute for Practical Robotics, 1997- 2023



Verbindung zum Wombat herstellen mittels Wlan

1. Starte nach der Verbindung einen Webbrowser (z. B. Safari, Chrome oder Firefox).
2. Kopiere diese IP-Adresse in die Adressleiste deines Browsers, gefolgt von ":" und der Portnummer 8888



3. Die KISS Software Suite wird nun in deinen Browser angezeigt.
4. Du kannst einen Computer, ein Tablet, ein iPad oder auch ein Smartphone verwenden.
5. Wenn du ein Ethernet-Kabel verwenden musst, um eine Verbindung herzustellen, lies bitte unter "Wombat Update und optionale Funktionen" nach.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

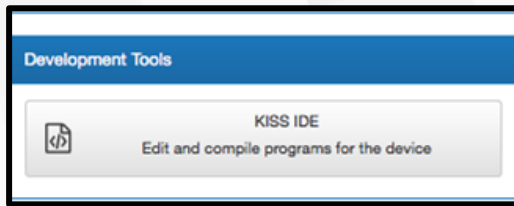
©KISS Institute for Practical Robotics, 1997- 20



**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

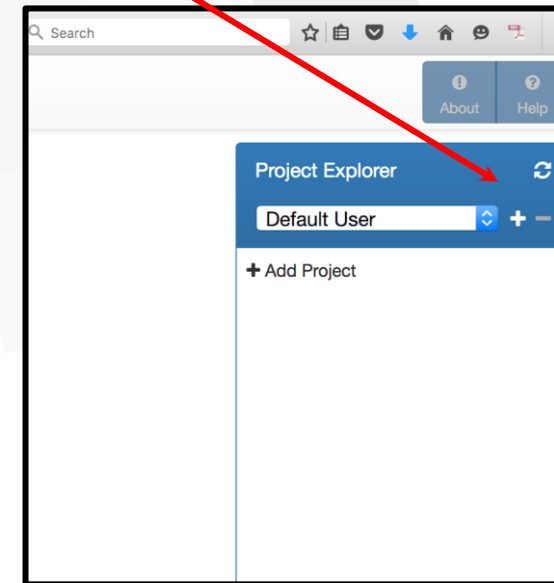
Team Ordner erstellen

1. Klicke auf **KISS IDE**

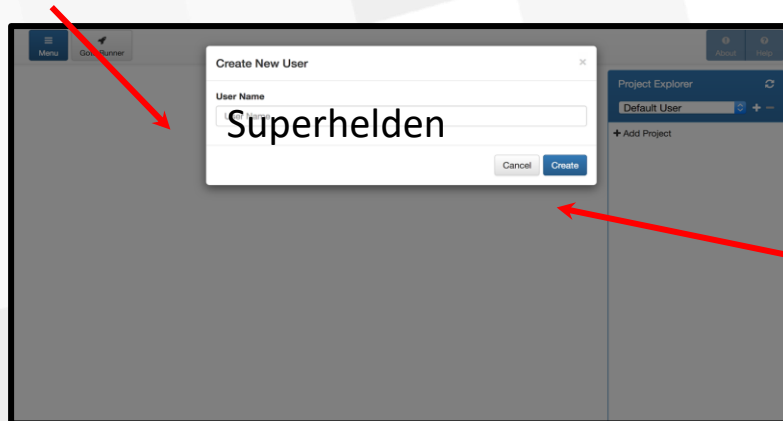


2. Unter **Project-Explorer** hast du die Möglichkeit, einen Benutzer hinzuzufügen, indem du auf das +-Zeichen klickst.

3. Nennt euren neuen Benutzer nach eurem **Teamname**. Dies ist der Ordner, in dem ihr alle eure verschiedenen Projekte ablegen werdet.



Keine
Umlaute!



4. Klicke "Create" also erstellen



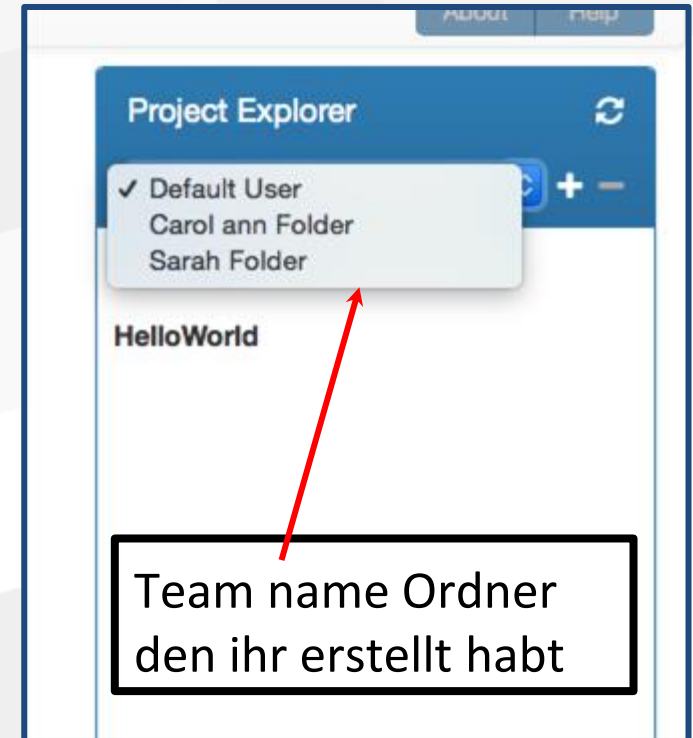
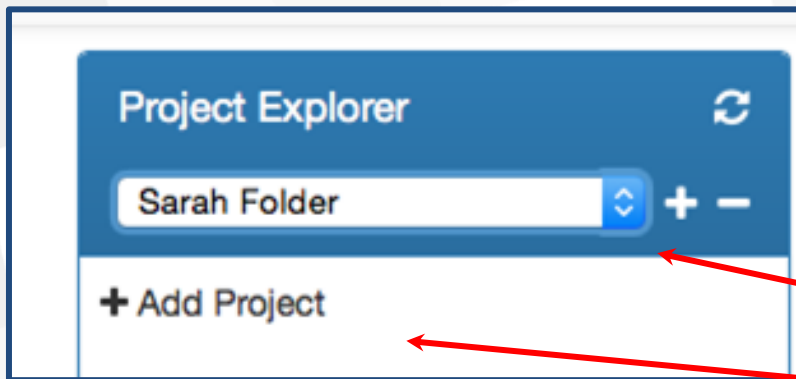
Projekt hinzufügen

Geht zurück zu "Project Explorer" und wählt den von euch erstellten Teamnamen aus der Dropdown-Liste. Ihr solltet den von euch erstellten Ordner sehen.

Klickt auf "+Add Project".

Fügt nun ein Projekt zu euren Ordner hinzu. Gebt keine Umlaute, Sonderzeichen oder Punkte usw. ein (schreibt nicht: Jösefs-Ordner).

Fahrt mit der nächsten Folie fort, um weitere Informationen zu erhalten.



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Benennt euer Projekt

- Gebt dem Projekt einen Namen
- Fügt keine Sonderzeichen, Punkte usw. hinzu.



Create New Project

Project name

Hello World

Programming Language

C

Source file name

main.c

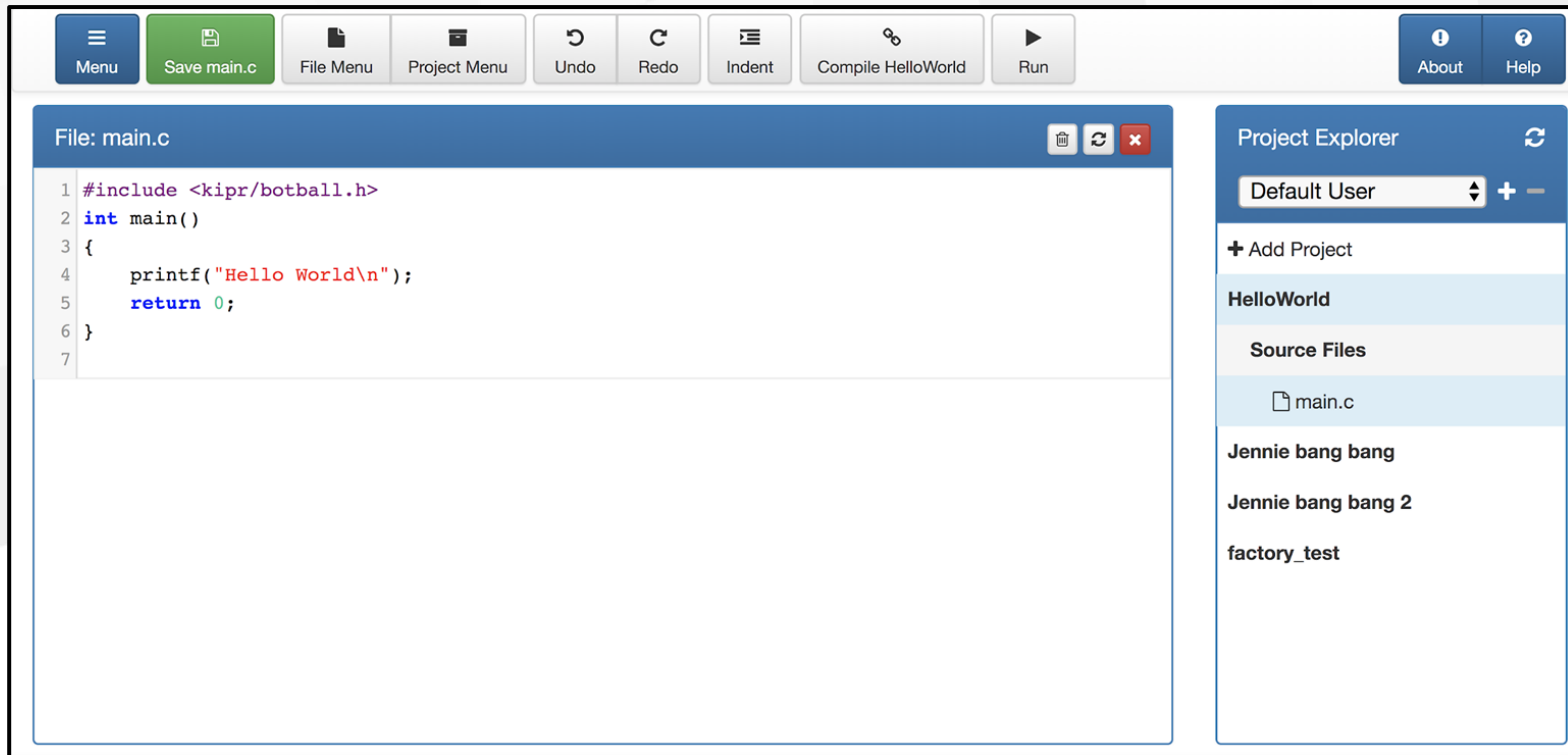
Cancel Create

Klicke Create



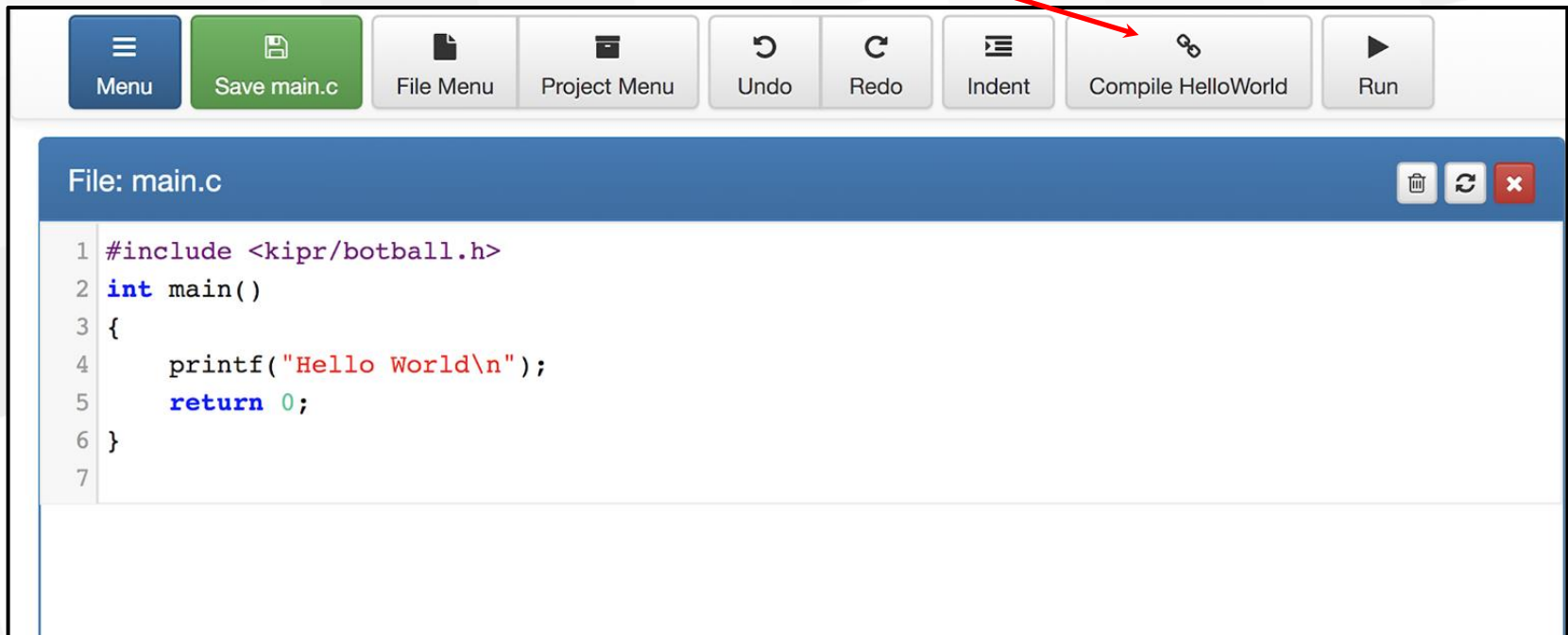
Vorlage

Information: So wird jedes neu erstellte Projekt ausschauen



Kompiliert das Programm

1. Klickt "Compile" damit wird der Code in **Maschinensprache** übersetzt
2. Der Compile Button sendet das Programm auch an den Controller



CCA - COMPETENCE CENTRE

HTL Anichstraße

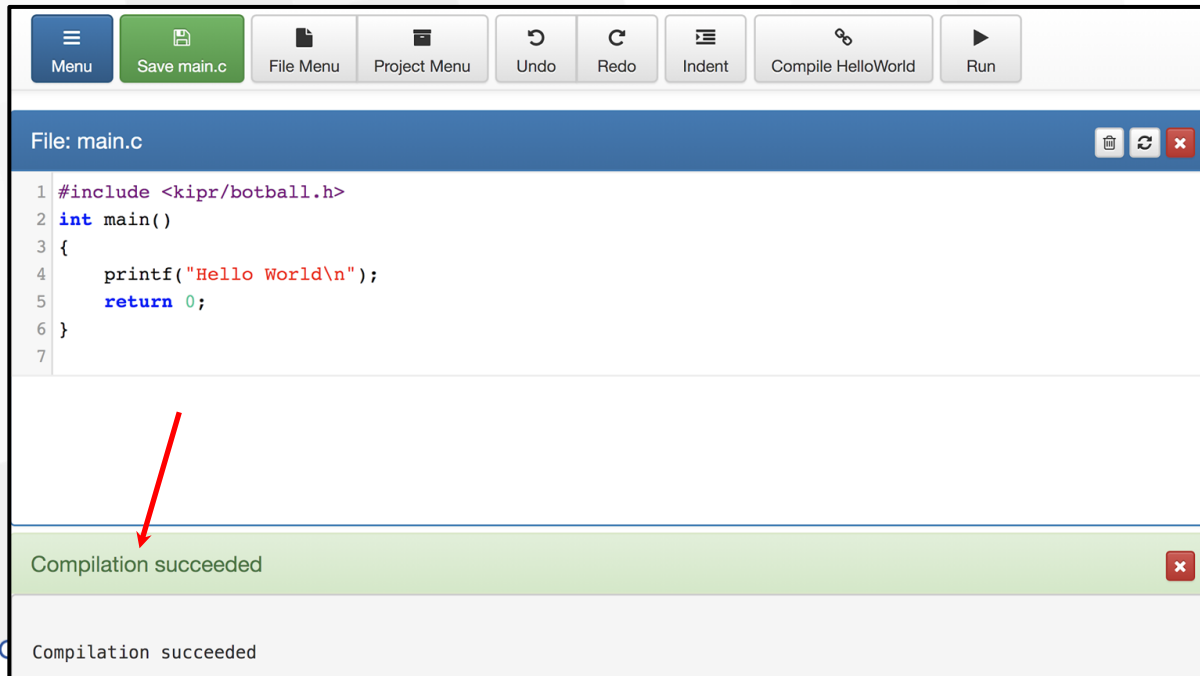
ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Compiling Your First Project

1. **“Compilation succeeded”** bedeutet das alles einwandfrei funktioniert hat
2. Falls **“compilation failed”** dort steht bedeutet es, dass entweder etwas bei der Verbindung schief gegangen ist oder sich ein Fehler im Code eingeschlichen hat



The screenshot shows a code editor window with a toolbar at the top containing buttons for Menu, Save main.c, File Menu, Project Menu, Undo, Redo, Indent, Compile HelloWorld, and Run. The main text area displays the following C code:

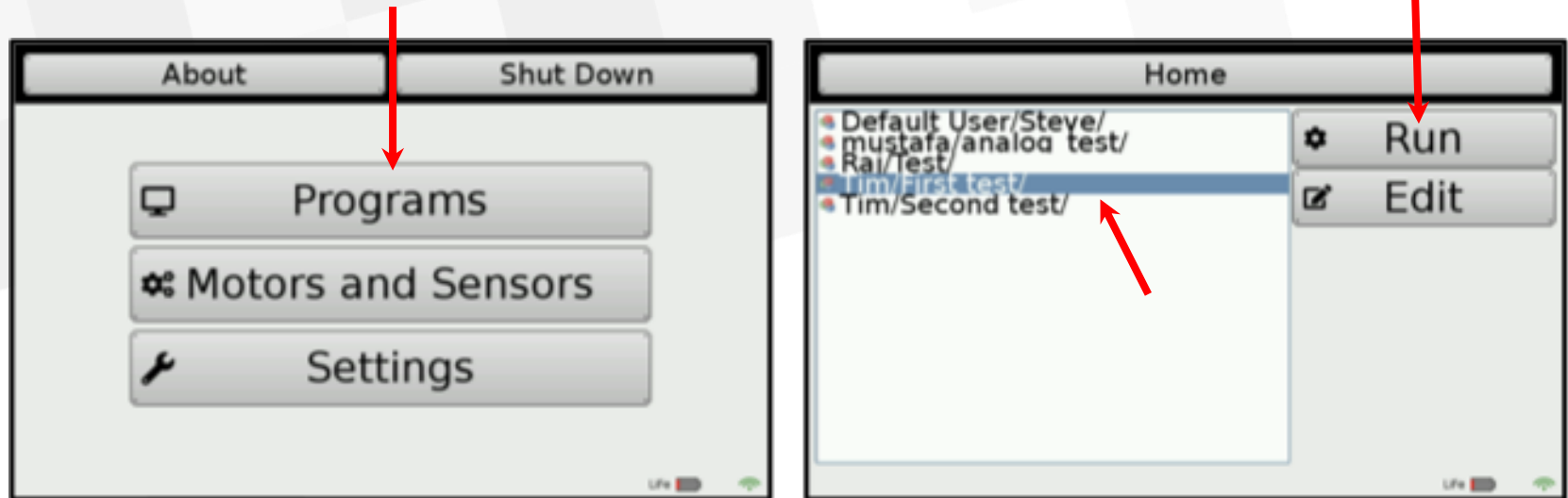
```
1 #include <kipr/botball.h>
2 int main()
3 {
4     printf("Hello World\n");
5     return 0;
6 }
7
```

Below the code editor, a green status bar displays the message "Compilation succeeded". A red arrow points from the text "Compilation succeeded" in the list above to this status bar. At the bottom of the editor window, another status bar also displays "Compilation succeeded".



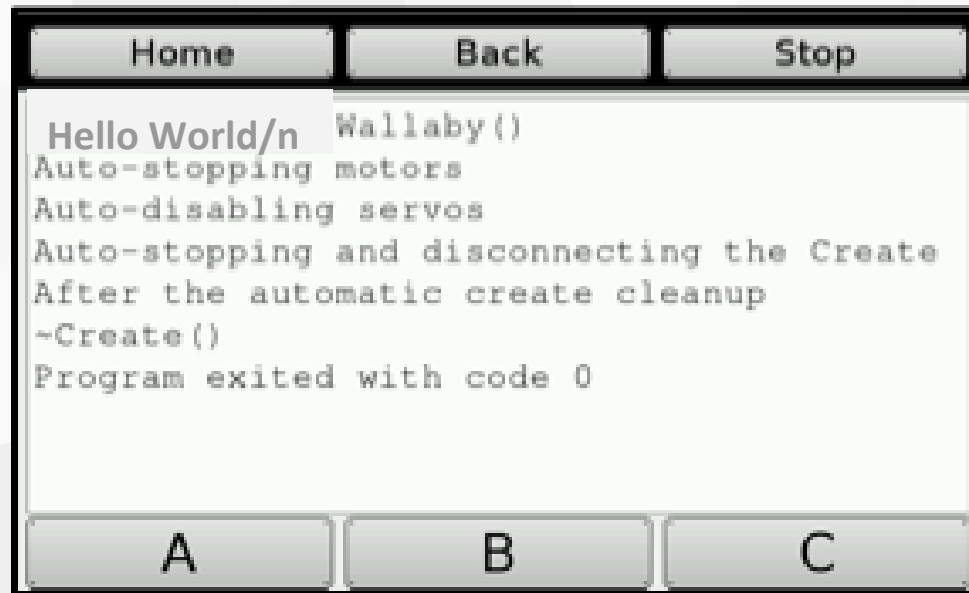
Programm am Roboter starten

1. Klicke auf Programs
2. Klicke auf dein eben erstelltes Programm
3. Klicke auf Run



Programm am Roboter

Nun sollte **“Hello, World!”** am Bildschirm eures Wombat erscheinen



```
Home Back Stop

Hello World/n Wallaby()
Auto-stopping motors
Auto-disabling servos
Auto-stopping and disconnecting the Create
After the automatic create cleanup
~Create()
Program exited with code 0

A B C
```



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023




Kommentarte hinzufügen

1. Kommentare können mit `//` hinzugefügt werden Sie dienen dazu den Code verständlicher zu machen
Fügt ein Kommentar in euer Programm hinzu. Das Kommentar kann prinzipiell überall stehen
1. Compiliere das Programm erneut starte es am Controller
Du wirst sehen das sich nichts an der Ausgabe geändert hat.

On same line as the function

```
#include <kipr/wombat.h>
int main()
{
    printf("Hello, World!\n"); //Print Hello World to screen
    return 0;
}
```



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Schummelzettel

Bis du mit den Funktionen, die du beim Programmieren verwenden wirst, vertraut bist, benutze den "Schummelzettel" als einfache Referenz. Kopieren und Einfügen ist ebenfalls sehr hilfreich.

```
printf("text\n"); //Gib Text am Bildschirm aus
motor(port#, % power); //Schaltet den Motor mit der eingestellten
Leistung in % ein
msleep(# milliseconds); //Das Programm wartet für die angegebene
Anzahl an Millisekunden
ao(); //All off, schaltet alle Motoren aus
enable_servos(); //Schaltet die Servo-Ports ein
set_servo_position (port#, position); //Bewegt den Servo am angeben
Servo-Port zu angegeben Position
disable_servos(); //Schaltet die Servo-Ports aus
digital(port #); //Liest den Wert des angegeben digitalen Ports # aus
analog (port #); //Liest den Wert des angegeben analogen Ports # aus
```

*Viele weitere Funktionen sind in der KIPR-Library vorhanden



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Die `msleep()` Funktion

Informationen: Denkt daran, dass der Controller den Code liest und sehr schnell in die nächste Zeile wechselt.

Bei 800 MHz führt der Controller ~800 Millionen Codezeilen/Sekunde aus!

Um das Programm zu verlangsamen, wird die eingebaute Funktion `msleep()` verwendet.

Mit dem Befehl `msleep()` kann das Programm eine Pause einlegen (schlafen), bevor es den nächsten Befehl ausführt (läuft).



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023




Einführung `msleep()` Funktion

Aktivität:

1. Schreibe ein Programm, mit dem der KIPR Wombat "Hello World!" auf dem Bildschirm anzeigt, zwei Sekunden lang pausiert und dann "Hello <Dein Name>" auf dem Bildschirm anzeigt.
2. Füge eine **`msleep(#milliseconds)`** ;Anweisung zwischen die beiden printf Anweisungen.

`msleep(1000)` ; bewirkt, dass das Programm 1 Sekunde lang "pausiert" (das m steht für Millisekunden oder 1/1000 einer Sekunde), bevor es zur nächsten Zeile übergeht. Dieses Programm weist den Roboter an, 1 Sekunde zu warten, bevor er zum nächsten Befehl übergeht.

Diese Funktion hat nur ein Argument, das Argument der Zeit in Millisekunden.

`msleep(1000)` ; 

Frage:

Wie viele Sekunden sind 2000ms? 3000ms? Wie viele Millisekunden braucht man, um den Roboter 4 Sekunden lang laufen zu lassen?



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



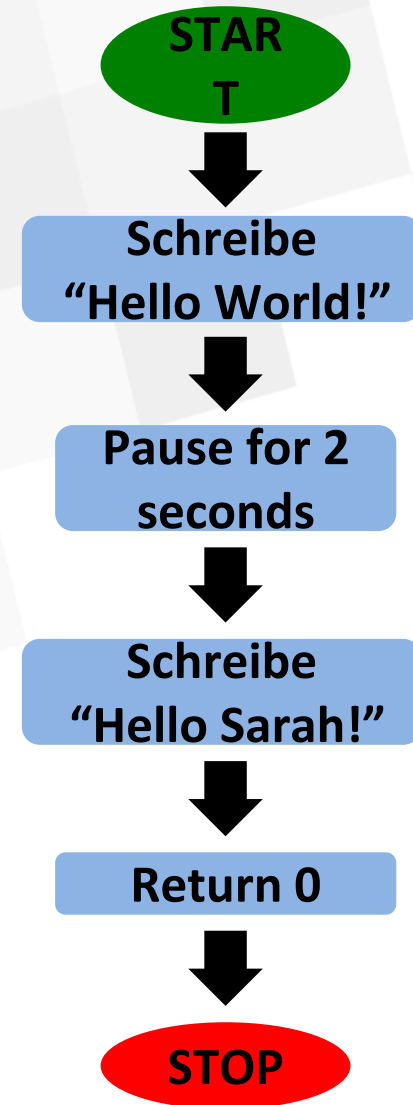
Einführung msleep() Funktion

1. Erkläre deinem Sitznachbar, was dieses Programm tun wird.

Pseudocode (Task Analysis)

```
// 1. Zeige "Hello World!" auf dem Bildschirm an.  
// 2. 2 Sekunden lang pausieren.  
// 3. Anzeige Ihres Namens auf dem Bildschirm.
```

```
printf("Hello World!\n"); //Hello World  
msleep(2000); //Pause  
printf("Hello Sarah!\n"); //Hello Sarah
```



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Einführung msleep() Funktion

Pseudocode als Kommentar

1. Zeige "Hello, World!"
2. Warte für 2 Sekunden.
3. Zeige deinen Namen.



Quellcode (Code in der KISS IDE)

```
#include <kipr/wombat.h>
int main()
{
    printf("Hello, World!\n"); // Zeige "Hello, World!"
    msleep(2000); // Warte für 2 Sekunden

    printf("Hello Sarah\n"); // Zeige "Sarah" am Bildschirm.

    return 0;
}
```



CCA - COMPETENCE CENTRE

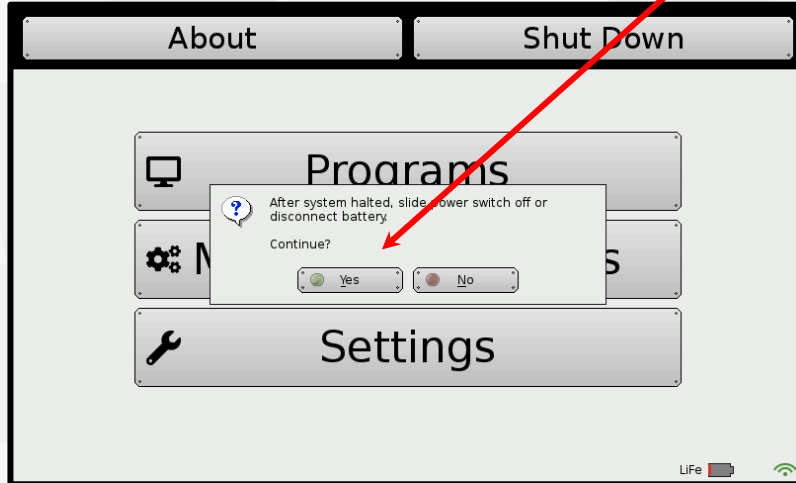
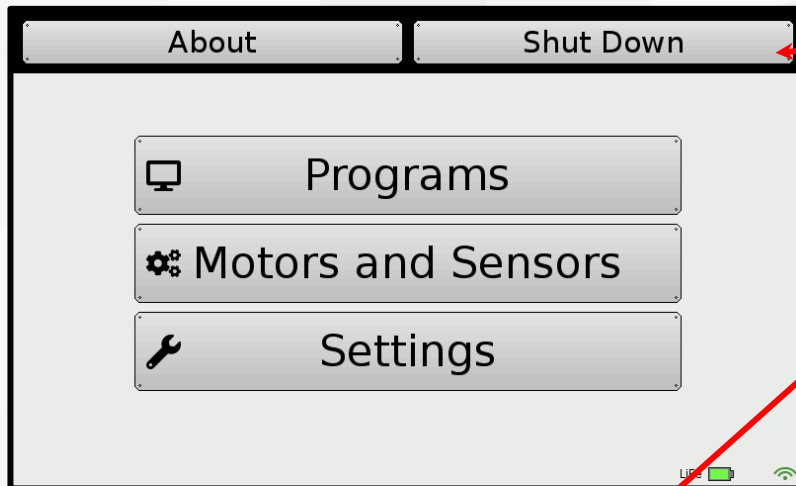
HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

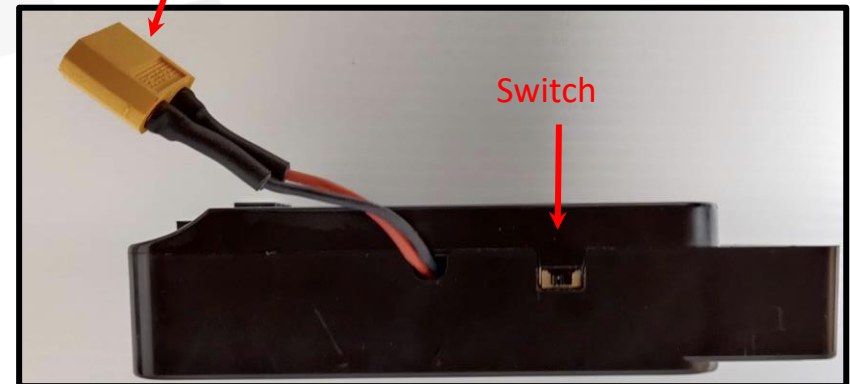
©KISS Institute for Practical Robotics, 1997- 2023



Richtiges Herunterfahren des Wombat-Controllers



1. Wähle Shutdown
2. Es erscheint die Frage, ob du sicher bist, dass du weitermachen willst; wähle Yes.
3. Jetzt ist es sicher, den Netzschalter auszuschalten.
4. Trenne den Akku vom Wombat (gelber Stecker)



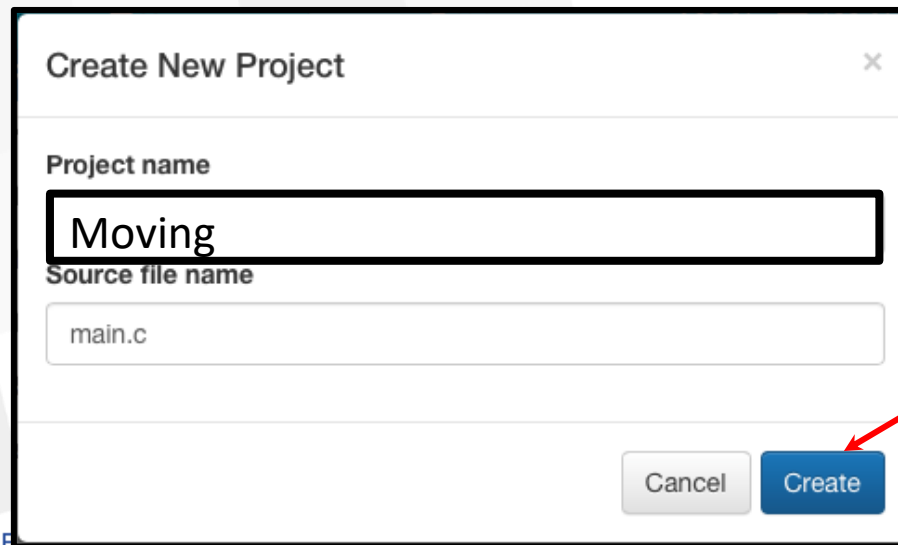
Roboter Bewegen Neues Projekt

Resource

1. Klicken auf "+Projekt hinzufügen".
2. Gebt eurem Projekt einen Namen

Gib keine Sonderzeichen, Punkte usw. ein.

Beispiele für schlechte Projektnamen: m.j.c., my amazing project!, Mrs Davis' project.



Create New Project

Project name

Moving

Source file name

main.c

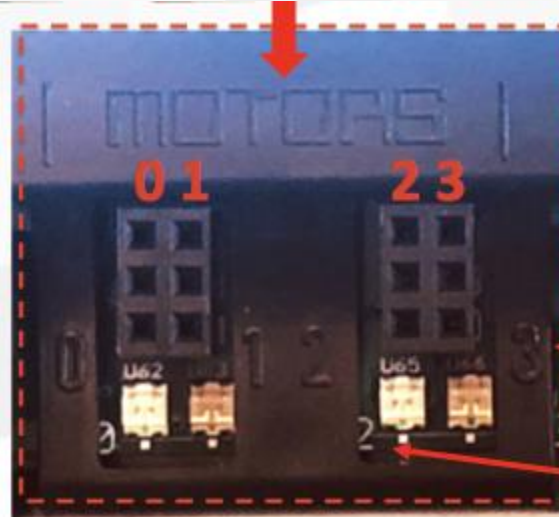
Cancel Create

Click Create



Erkundung der Motoren und des Motor-Widgets

- Die Ansteckplätze heißen Ports diese sind von 0-3 durchnummeriert
- Die Motoren sollten dort schon angesteckt sein



Motor Port Labels are on the Case

Motor Port Labels 0, 2 are also on the board



CCA - COMPETENCE CENTRE

HTL Anichstraße

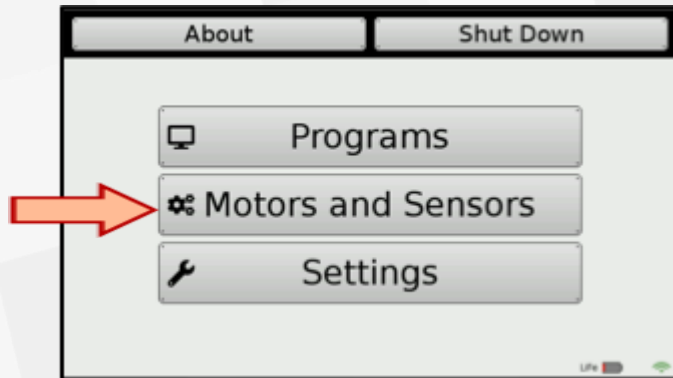
ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

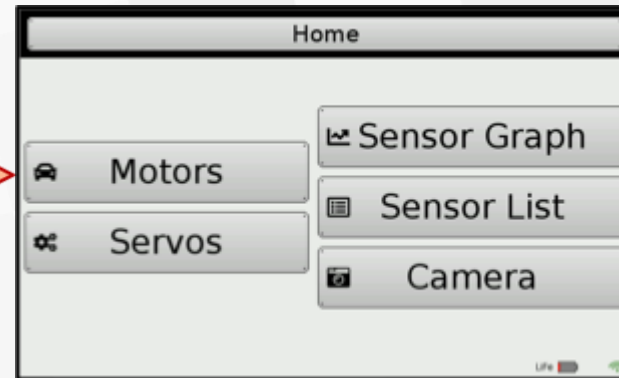
**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Erkundung der Motoren und des Motor-Widgets

1. Klicke auf *Motors and Sensors*.

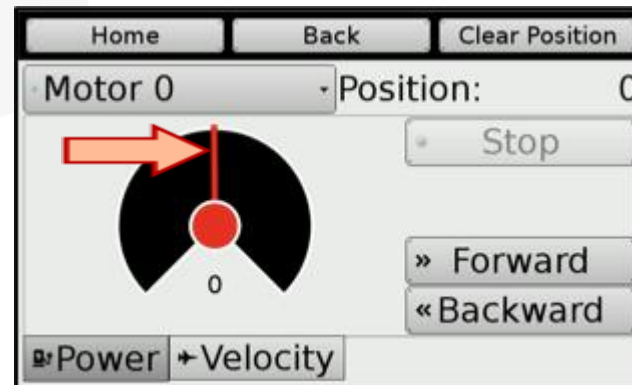


2. Klicke auf *Motors*.



3. Zieht die Rote Linie nach Links oder Rechts

4. Um den anderen Motor anzusteuern drücke auf Motor 0 und wähle dann Motor 1 aus



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Erkundung der Motoren und des Motor-Widgets

- Nun drückt mal auf forward (auf deutsch vorwärts)
- Falls die LED (dort wo der Motor am Controller ist **grün** leuchtet **habt ihr Glück gehabt**
- Falls sie Rot leuchtet steck den Motor ab und dreht den Stecker umgekehrt wieder ein.



↑
forward



↑
backward



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Kennenlernen der Funktion motor()

Informationen: Dies ist eine neue Funktion.

Es gibt mehrere Funktionen zur Steuerung von Motoren, wir werden mit motor() beginnen.

```
motor(0,60); //Schaltet den Motor von Port 0 mit 60%  
             power an  
motor(1,60); //Schaltet den Motor von Port 1 mit 60%  
             power an  
msleep(4000); // Lässt das Programm für 4 Sekunden  
              pausieren ohne den würde der Motor nur an und  
              gleich wieder ausgeschalten werden  
ao(); //Schaltet alle Motoren aus
```

Zeit in Millisekunden

msleep(3000)

Port Power

motor(0,50);



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

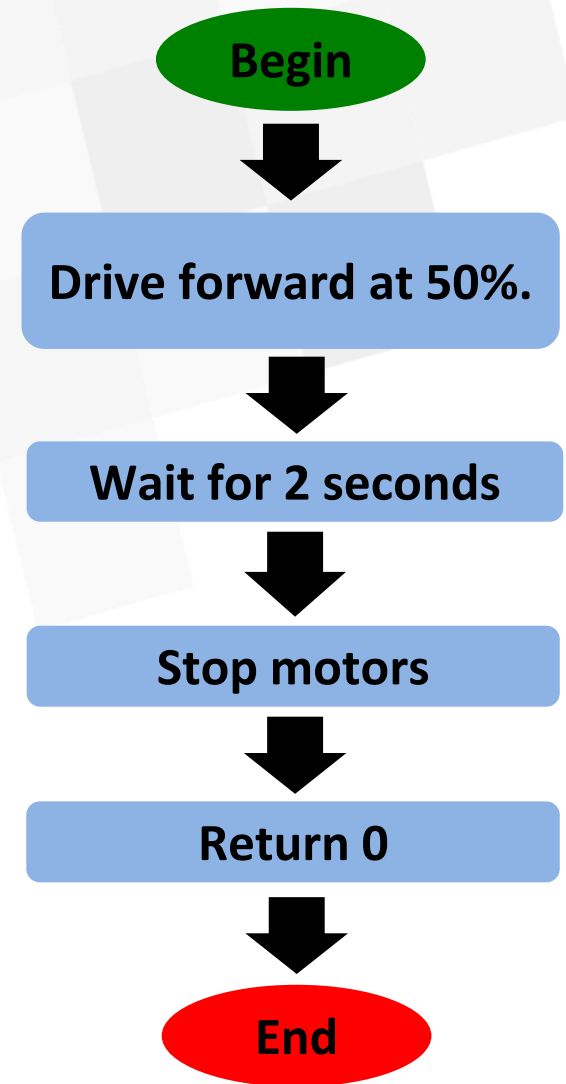
Moving the Robot

Aktivität: Schreibe ein Programm für den Wombat, das den Roboter zwei Sekunden lang mit 50% Leistung vorwärts fährt und dann anhält.

Leitfrage: Was soll das Programm tun?

Pseudocode:

1. Fahre mit 50% vorwärts.
2. Warte 2 Sekunden lang.
3. Stoppe die Motoren.
4. Beende das Programm.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Moving the Robot

1. Gib den Code ein, indem du die Funktionen motor(port, power); msleep(Zeit); und ao(); verwendest.
2. Kompilier dein Programm. War es erfolgreich?
3. Wenn nicht, suche den Fehler.
4. Kompiliere erneut, bis die Meldung "compilation succeeded" erscheint.
5. Führen das Programm auf dem Wombat aus. Lief es so, wie es erwartet wurde?



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Lösung

```
#include <kipr/wombat.h>
int main()
{
    printf("Move\n");

    motor(0, 50);
    motor(3, 50);
    msleep(2000);

    ao();
    return 0;
}
```

Geleitete Fragen:

1. Welchen Wert würdest du ändern, damit der Roboter eine längere Strecke zurücklegt?



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

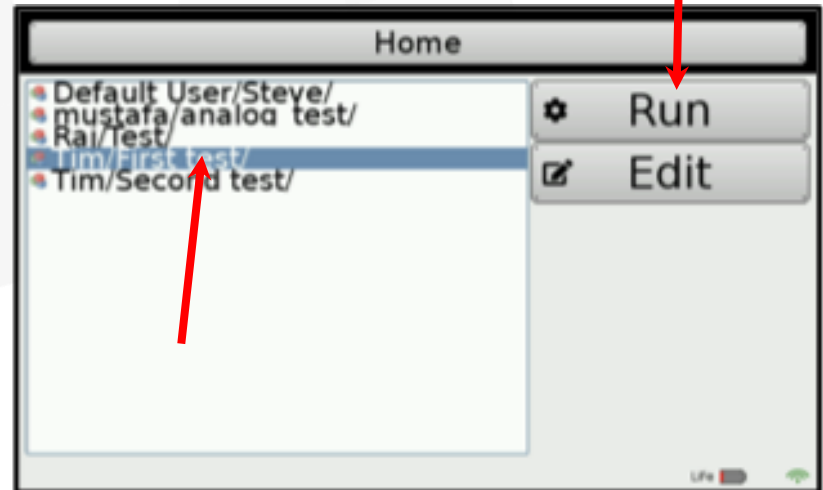
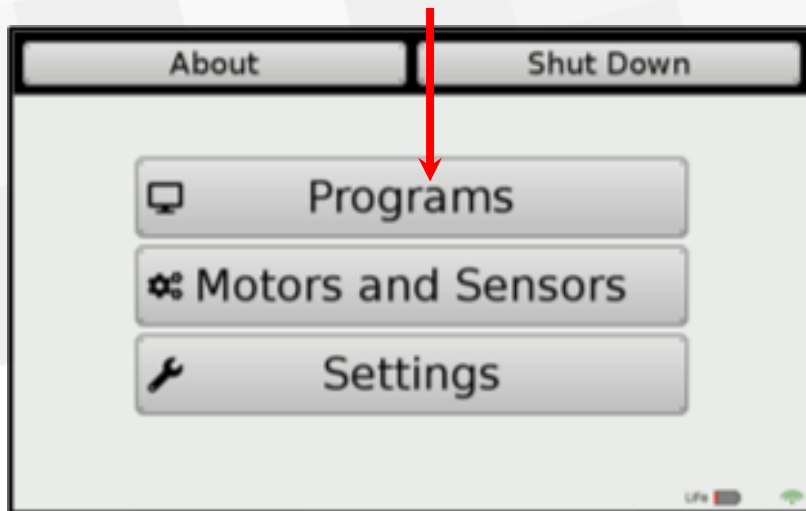
©KISS Institute for Practical Robotics, 1997- 2023



Programm am Roboter ausführen

Resource

1. Drücke auf "Programme" auf dem KIPR Wombat.
2. Markiere das Programm, das ausgeführt werden soll.
3. Stelle den Roboter auf den Boden und wähle dann "Run"..



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 20

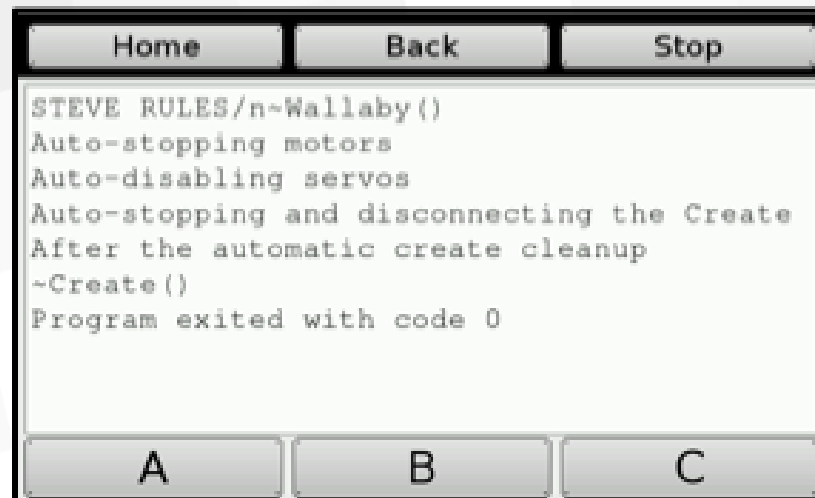


**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Programm am Roboter ausführen

Resource

- Nachdem das Programm auf dem Wombat läuft und beendet wird, sieht man diesen Bildschirm auf dem Wombat



The screenshot shows a small LCD screen with a black border. At the top, there are three buttons labeled 'Home', 'Back', and 'Stop'. The main area of the screen displays the following text in a monospaced font:

```
STEVE RULES/n~Wallaby()  
Auto-stopping motors  
Auto-disabling servos  
Auto-stopping and disconnecting the Create  
After the automatic create cleanup  
~Create()  
Program exited with code 0
```

At the bottom of the screen, there are three buttons labeled 'A', 'B', and 'C'.



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 20

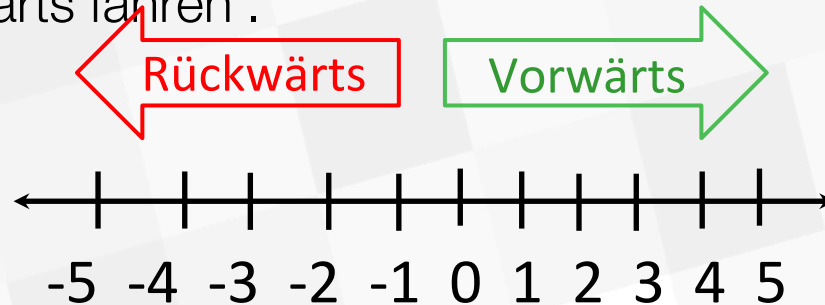


**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Resource

Robot Driving Hints

Denke daran, positive Zahlen lassen den Roboter vorwärts und negative Zahlen rückwärts fahren.



Geradeaus fahren - ist nicht einfach

Warum fährt der Roboter nicht geradeaus?

- Die Motoren sind nicht genau gleich
- Die Reifen sind vielleicht nicht gut ausgerichtet
- Ein Reifen hat mehr Widerstand, usw.

Du kannst dies durch Verlangsamen und Beschleunigen der Motoren korrigieren.

Kurven fahren

Lasse ein Rad schneller oder langsamer fahren als das andere.

Ein Rad soll sich bewegen, während das andere stillsteht (die Reibung ist geringer, wenn sich beide Räder bewegen).



CCA - COMPETENCE CENTRE

HTL Anichstraße

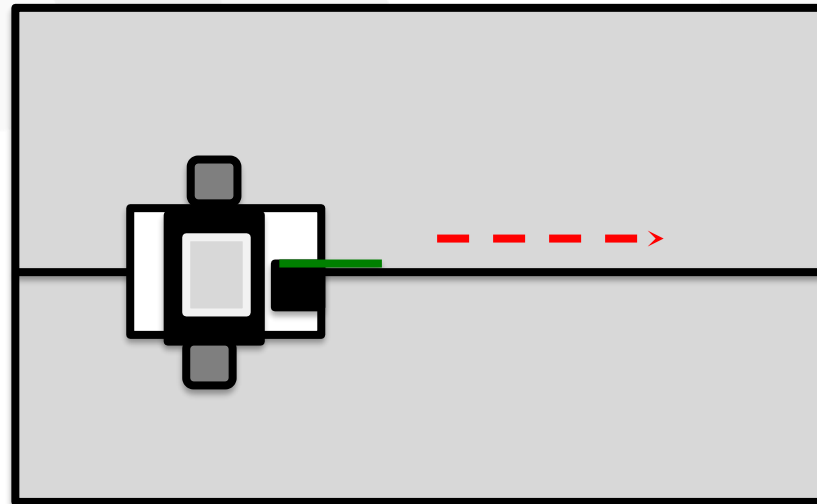
ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Gerade aus fahren

1. Benutze Matte B und fahre die Mittellinie entlang.
2. Der Roboter muss die gestrichelte Linie überqueren.
3. Passe die Kräfte deiner Räder an, bis sie gerade sind.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Rückwärts fahren

Informationen: Positive Power bewegt den Roboter vorwärts.

```
motor(0,100);
```

```
motor(1,100);
```

Geleitete Frage:

Wodurch wird der Roboter rückwärts fahren können?



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

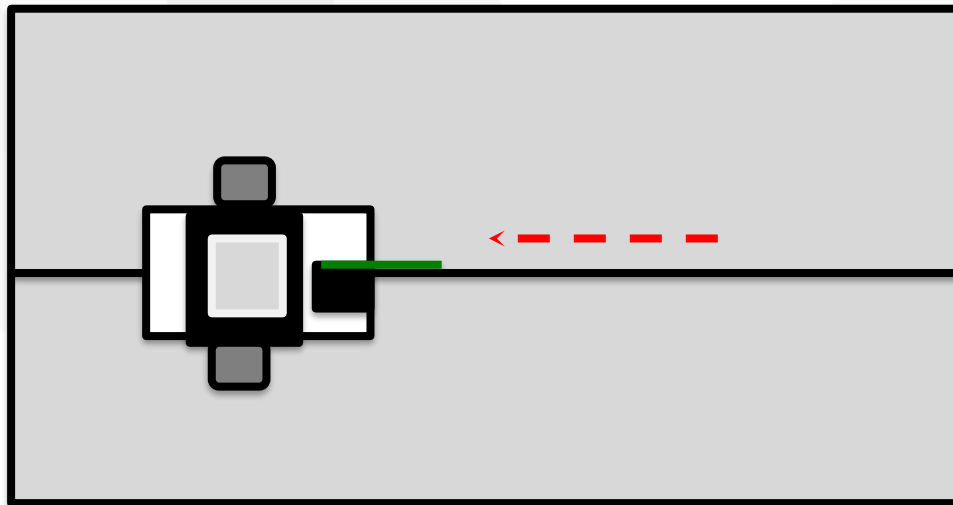


Rückwärts fahren

```
motor(0, -100) ;  
motor(1, -100) ;  
msleep(2000) ;
```

Negative Zahlen lassen die Räder rückwärts drehen

1. Erstelle ein neues Projekt mit dem Namen "Rueckwaerts".
2. Schreibe ein Programm, das rückwärts auf der Mittellinie fährt.

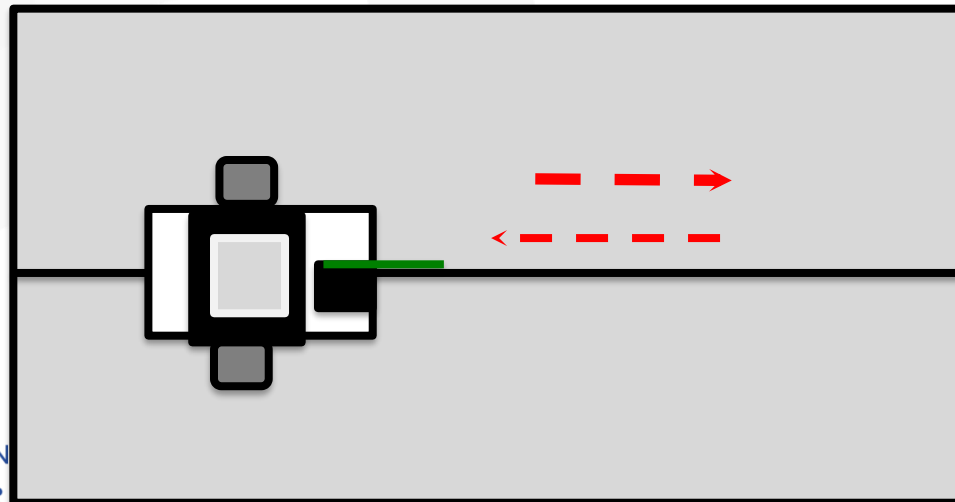


Vorwärts und Rückwärts fahren

Materials: Matte B

Activity:

- Fahre vorwärts die Mittellinie entlang.
- Der Roboter muss die Mittellinie überfahren .
- Dann fahre rückwärts zurück an die Anfangsposition



CCA - COMPETENZ

HTL Anichstrasse

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

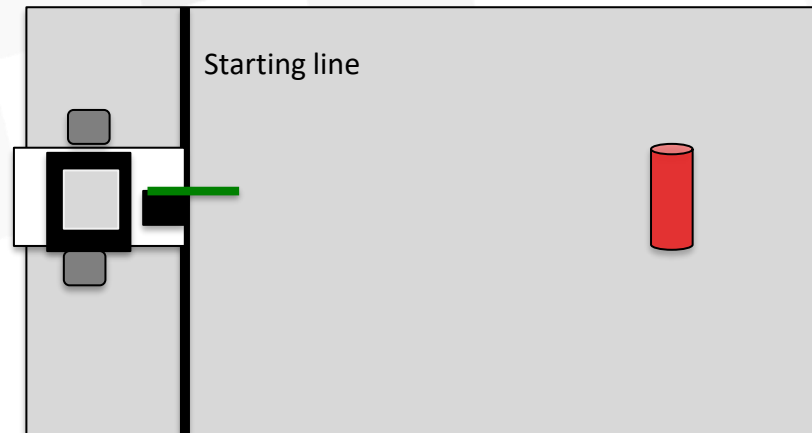
**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Berühre die Dose

- Öffne ein neues Projekt und nenne es "DoseBeruehren".

Der Roboter soll:

- Fahre zur Dose im Kreis 6 und berühre diese.
- Fahre rückwärts zur Startlinie zurück.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Mögliche Lösung Berühre die Dose

```
#include <kipr/wombat.h>

int main ()
{
    printf("Sarah Rueckwaerts");

    motor(0,100); // vorwärts
    motor(1,100);
    msleep(3000);

    motor(0,-100); // rückwärts
    motor(1,-100);
    msleep(3000);
    return 0;
}
```



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Kurven fahren Geschwindigkeit anpassen

1. Erstelle ein neues Projekt mit dem Namen "Kurven".
2. Ersetze in der Vorlage "Hello World" durch den Namen des Projekts "Kurven".
3. Verringere die Leistung am Motoranschluss 1 auf 70 %, kompiliere und starte das Projekt. Mache Beobachtungen.

```
Motor (0,100);  
Motor (1,70);  
msleep (3000);
```

**Verändere die
Kraft auf dem
Motor**

4. Verringere die Leistung am Motoranschluss 1 auf 10 %, kompiliere und starte das Programm. Mache Beobachtungen. Halte msleep konstant (3000).
5. Wiederhole die obigen Schritte mit der folgenden Leistung am Motoranschluss 1; 0%, -50%,



Kurven fahren - Verändere msleep()

1. Verringere die Leistung am Motor-Port 3 auf 50 %, dann kompiliere und starte das Programm.

```
motor (0,100);  
motor (3,50);  
msleep (3000);
```

Verändere die Zeit bei msleep

2. Erhöhe die msleep-Zeit auf 4500. Beobachte was passiert.
3. Verringere die msleep-Zeit auf 1500. Beobachte was passiert.
4. Erhöhe die msleep-Zeit auf 9000. Beobachte was passiert.
5. Verringere die msleep-Zeit auf 500. Beobachte was passiert.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

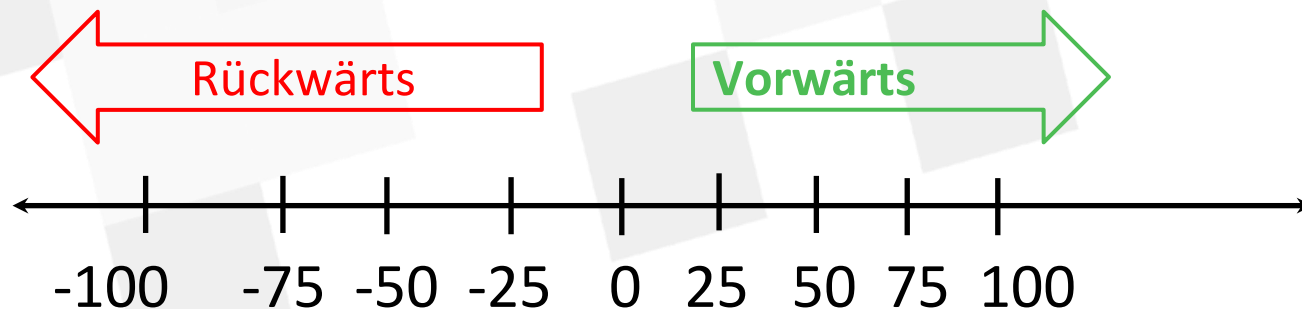


Kurven fahren

1. Ändere weiterhin die `msleep()` oder die Motorleistung und beobachte was passiert.

Fragen:

1. Was passiert, wenn die Leistung verringert wird?
2. Was passiert, wenn man die `msleep()`-Zeit ändert?
3. Wie entsteht ein kleiner Kreis?
4. Wie entsteht ein großer Kreis?
5. Welche Entdeckungen hast du gemacht?



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Rechter Winkel

1. **Öffene deinen Ordner!** Erstelle in deinem Ordner ein neues Projekt mit dem Namen **Rechter Winkel**.
2. Finde heraus, wie du ein Programm schreibst, das eine rechtwinklige Kurve macht, kompilierst und auf deinem Roboter ausführst.
3. Als nächstes lass den Roboter ein:
 - kleines Rechteck fahren
 - großes Rechteck fahren
 - Quadrate verschiedener Größen fahren

Tipp: Teste dein Programm einen Schritt nach dem anderen, bevor du mit dem nächsten weitermachst.

Das ist keine leichte Aufgabe!



CCA - COMPETENCE CENTRE

HTL Anichstraße

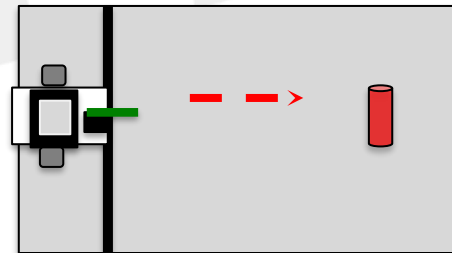
ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Nähester an der Dose

1. **Öffne deinen Ordner!** Erstelle in deinem Ordner ein neues Projekt mit dem Namen **Naehester an der Dose**.
2. Folge den nachstehenden Anweisungen, kompiliere dein Programm und führe es aus.
 - Nähester an der Dose: Bewege die Dose zu verschiedenen Abständen
 - Die Roboter müssen hinter der Startmarkierung beginnen und sich auf das Objekt zubewegen, mit dem Ziel, so nah wie möglich an der Dose anzuhalten, ohne sie zu berühren.
 - Wenn sie die Dose berühren, müssen sie an der Startlinie neu beginnen.
 - Verwende Lineale, um den Abstand zu messen, der von der Dose gestoppt wurde - erstelle eine Datentabelle
 - Ihr könnt die Anzahl der Versuche begrenzen und den besten Lauf nehmen, den Durchschnitt aus mehreren Läufen ermitteln oder die Entfernungen für eine Gesamtsumme zusammenzählen.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

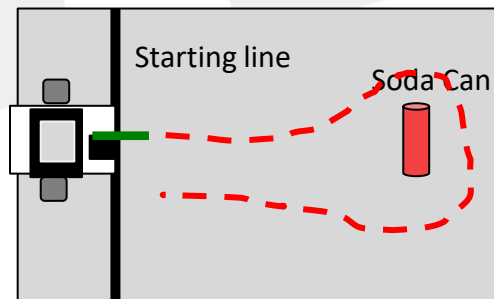


Umfahre die Dose

1. Platziere eine Dose auf Kreis #6. Befolge die unten stehenden Anweisungen, kompiliere dein Programm und lass es auf deinem Roboter ausführen.
2. Der Roboter muss um die Dose herumfahren, ohne sie umzukippen oder sie vollständig aus dem Kreis zu bewegen. Sobald die beiden Antriebsräder die Ziellinie überquert haben, haben sie die Aktivität abgeschlossen (Denk daran, deinen Code zu kommentieren).

Variationen:

- Bewege die Dose in verschiedene Kreise (4,2,9).
- Lass deinen Roboter im Uhrzeigersinn und dann gegen den Uhrzeigersinn um die Dose fahren
- Mathematik - Messe die Entfernung zum Objekt und die Zeit des Roboters, die du zur Berechnung der Geschwindigkeit verwenden kannst (Geschwindigkeit = Entfernung/Zeit).



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023

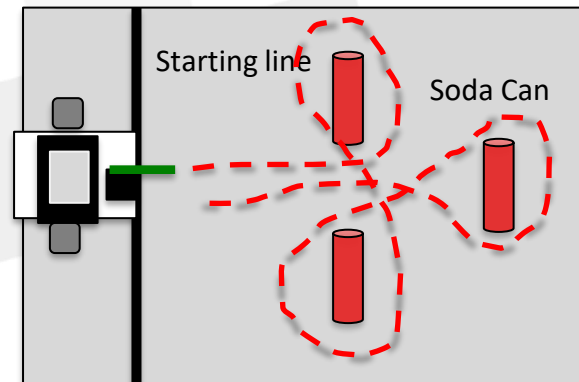


Fass Rennen

Variation von Umfahre die Dose:

Aktivität:

1. Der Roboter muss um die Dosen herumfahren, ohne sie umzukippen oder sie ganz aus dem Kreis zu bewegen. Sobald die beiden Antriebsräder wieder auf der Startlinie stehen, hast du die Aktivität abgeschlossen. (Denk daran, deinen Code zu kommentieren).
2. Fass Rennen (lasse den Roboter um drei Dosen fahren)



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Parken in der Garage

Erstelle ein neues Projekt, nenne es, *“Parken in der Garage”*.

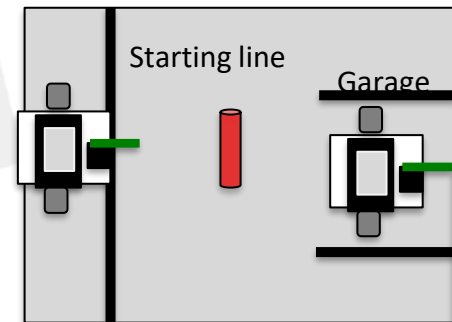
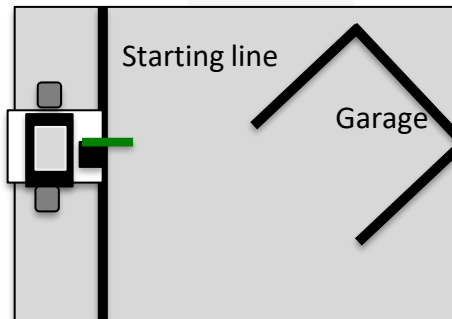
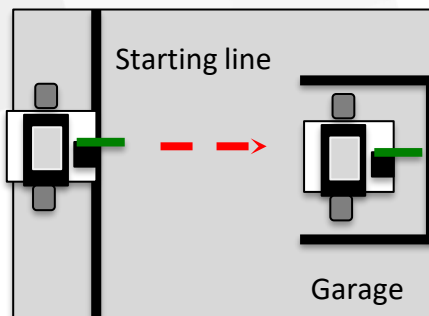
Ziel: Die Roboter müssen hinter der Startlinie starten und in der blauen Garage parken. (Denk daran, deinen Code zu kommentieren).

Die Aufgabe ist abgeschlossen, wenn sich die Antriebsräder in der Garage befinden und keine der Garagenwände berühren.

Regeln: Der Roboter darf die durchgezogenen Linien der Garage, in der er parkt, nicht überqueren oder berühren.

Variationen:

- In der Orangen oder Grünen Garage parken
- Nicht gegen das Fahrrad stoßen: Platziere eine Cola-Dose zwischen der Startlinie und der Garage. Der Roboter darf die Dose nicht berühren.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2023



Entscheidungsfindung und Sensoren

Information

Robotersensoren werden verwendet, um den Zustand und die Umgebung eines Roboters einzuschätzen. Diese Signale werden an den Controller weitergeleitet, um ein angemessenes Verhalten zu ermöglichen. Die Sensoren in Robotern basieren auf den Funktionen der menschlichen Sinnesorgane. Roboter benötigen umfangreiche Informationen über ihre Umgebung, um effektiv arbeiten zu können

Warum Sensoren verwenden?

- Sensoren tragen zur Entscheidungsfindung bei.
- Sensoren machen unsere Roboter intelligenter.



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Smarter Roboter

Resource

Informationen:

Wenn du dich an deinem Computer anmeldest, musst du ein Passwort eingeben. Das Programm vergleicht dieses mit einem gespeicherten Wert, und wenn er übereinstimmt, wird das Programm geöffnet. Wenn das Passwort nicht übereinstimmt, führt das Programm einen anderen Code aus, der dich auffordert, es noch einmal zu versuchen, oder dich sogar aussperrt!

Was macht einen Roboter intelligenter?

Um einen intelligenten Roboter zu bauen, müssen wir Sensorwerte überprüfen und vergleichen

Sensorwerte sind entweder:

Digital - Liefert einen Wert von 0 oder 1 (wahr oder falsch), (berührt nicht berührt)

Zu den Sensoren gehören: kleine Berührung, große Berührung und Hebel

Analog - liefert ganzzahlige Werte zwischen 0-4095 (12 Bit analog =

2¹² oder 4096 - denke daran, dass wir bei 0 anfangen zu zählen)

Sensoren: Licht, kleiner Zylinder und ET (Entfernungsmesser)

*Informationen zu den Sensoren finden Sie im Sensor- und Motor Hilfe, das in der Hilfedatei des Wombat zu finden ist (obere rechte Ecke des Compiler-Bildschirms).



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

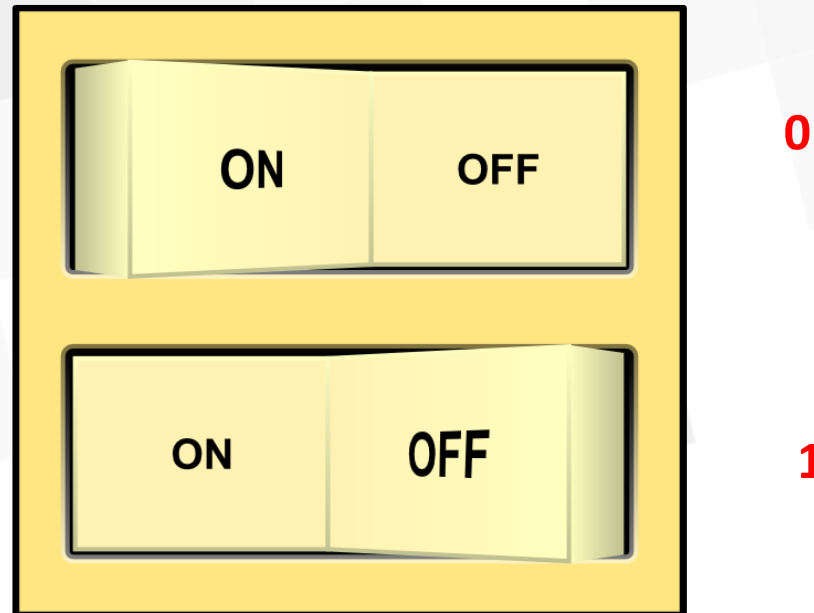
©KISS Institute for Practical Robotics, 1997- 2022



Digitale Sensoren

Information

- Man kann sich diese digitalen Sensoren ähnlich wie einen Lichtschalter vorstellen: Sie haben nur zwei Möglichkeiten, aus nicht berührt (0) und an, berührt (1).



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Es ist dunkel, ich brauche meine Sensoren!

Informationen

Eine gute Analogie für Sensoren ist das Szenario, dass du im Dunkeln zu einer Taschenlampe läufst, wenn der Strom ausfällt.

Wenn du nachts aufwachst und das Licht aus ist (es ist dunkel), nimmst du deine digitalen Sensoren (deine Hände) hoch und suchst nach Wänden und Gegenständen, die dir im Weg stehen.

Prüfst du nur einmal? Nein!

Du prüfst (wiederholst) und prüfst (wiederholst) immer wieder. In der Computersprache nennt man das **Schleifenbildung!**

Du wiederholst dies so lange, bis deine Sensoren die Wand wahrnehmen, und dann tust du etwas anderes



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022

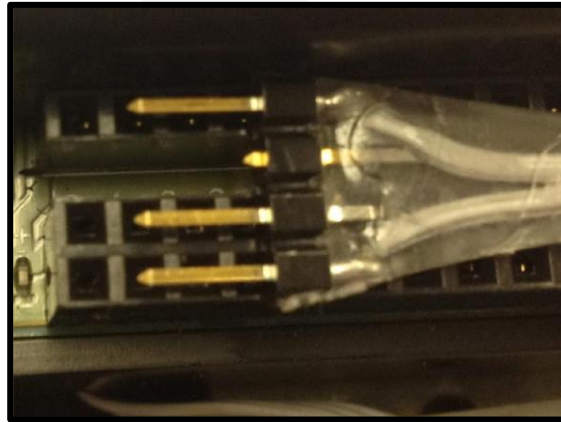


Sensor anstecken

Resource Digital Ports



Der Sensor sollte bei
Digital Port 0
angeschlossen werden.
Also ganz Links!



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

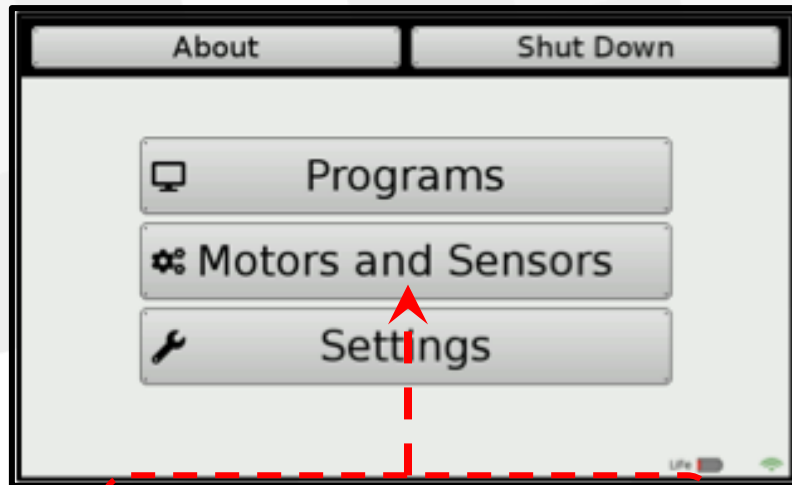
©KISS Institute for Practical Robotics, 1997- 2022

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

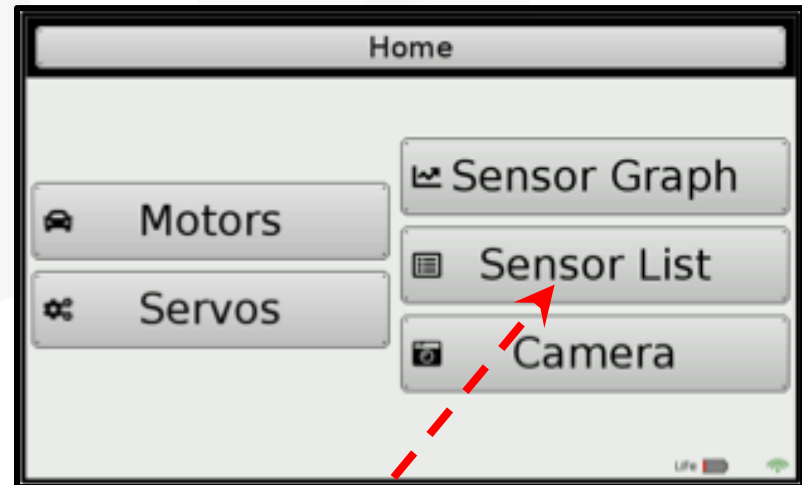
Lichtsensor: Ablesen von Sensorwerten

Informationen

- Die Sensorwerte können über die Sensorliste des Wombats abgerufen werden.
- Dies ist sehr hilfreich, um Messwerte von allen Sensoren zu erhalten, die verwendet werden, die Werte kann man dann im Code verwenden.



Klicke Sensoren und
Motoren



Klicke Sensor List



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022



Lichtsensor: Ablesen von Sensorwerten

Informationen

| Home | | Back | |
|---------------|----------|------|--|
| Analog | Sensor 0 | 1297 | |
| Analog | Sensor 1 | 1066 | |
| Analog | Sensor 2 | 1122 | |
| Analog | Sensor 3 | 1139 | |
| Analog | Sensor 4 | 1234 | |
| Analog | Sensor 5 | 1195 | |
| Digital | Sensor 0 | 0 | |
| Digital | Sensor 1 | 0 | |
| Digital | Sensor 2 | 0 | |
| Digital | Sensor 3 | 0 | |
| Digital | Sensor 4 | 0 | |
| Digital | Sensor 5 | 0 | |
| Digital | Sensor 6 | 0 | |
| Digital | Sensor 7 | 0 | |
| Digital | Sensor 8 | 0 | |
| Digital | Sensor 9 | 0 | |
| Accelerometer | X | 3 | |



Licht Sensor

Sensor Port Sensor Wert

Lese die Werte ab, wenn du mit der Handy-Taschenlampe auf den Lichtsensor leuchtest und, wenn nicht



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Verstehen von **while** Schleifen

Information

Wir verwenden eine **while** Schleife.

while Schleifen halten den Codeblock am Laufen (Wiederholung/Schleife), damit die Sensorwerte ständig überprüft werden können und eine Entscheidung getroffen werden kann.

Die **while** Schleife prüft, ob etwas wahr oder falsch ist (boolesche Operatoren).

```
while ( Bedingung )  
{  
    Code welcher  
    ausgeführt werden  
    soll solange die  
    Bedingung wahr ist  
}
```



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022



Beispiel

solange



while es regnet

{

der Regenschirm ist offen und über meinen Kopf

}

es regnet nicht mehr ich räume den Regenschirm weg



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022



While Schleife

Resource

```
while (digital (port#)==0)
```

Kein Strichpunkt

Sensortyp

Port Nummer

```
{  
  motor(0,100);  
  motor(3,100);  
}
```

Code der ausgeführt wird solange die **Bedingung** wahr ist - also der Digitale Sensor am Port 0 als Wert 0 hat.



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

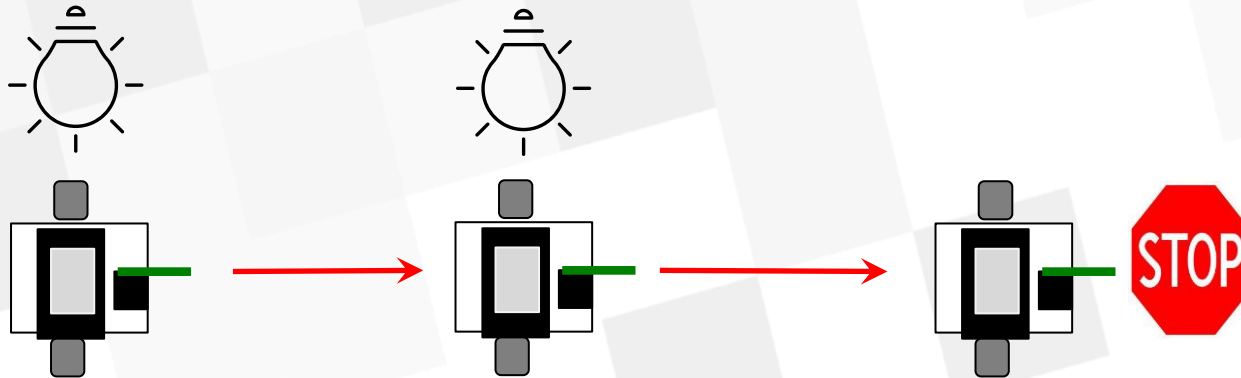
©KISS Institute for Practical Robotics, 1997- 2022



Fahre solange die Handy-Lampe leuchtet

Der Roboter wird vorwärts fahren, solange die Handy-Lampe auf den Lichtsensor leuchtet

1. Öffne ein neues Projekt und nenne es "Vorwaerts bis zur Wand".
2. Gebe ein Programm ein, kompilieren es und führen es aus, damit der Roboter vorwärts fährt, solange der Lichtsensor Licht hat.



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022

**KISS
INSTITUTE
FOR
PRACTICAL
ROBOTICS**

Fahre solange Licht

```
#include <kipr/wombat.h>

int main ()
{
    printf("Drive_while_light\n");

    while (digital (0)==1)
    {
        motor(0,100);
        motor(1,100);
    }

    ao();
    return 0;
}
```



CCA - COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022



Sensor Wert ändern

Activity 14.4

1. Ändere den erwarteten Sensorwert von 1 auf 0
2. Treffe Voraussagen über das Verhalten des Roboters
3. Führe das Programm aus
4. Diskutiere die Ergebnisse
5. Leuchte mit der Handy-Lampe zum Lichtsensor.



CCA – COMPETENCE CENTRE

HTL Anichstraße

ELEKTRONIK & TECHNISCHE INFORMATIK

©KISS Institute for Practical Robotics, 1997- 2022

