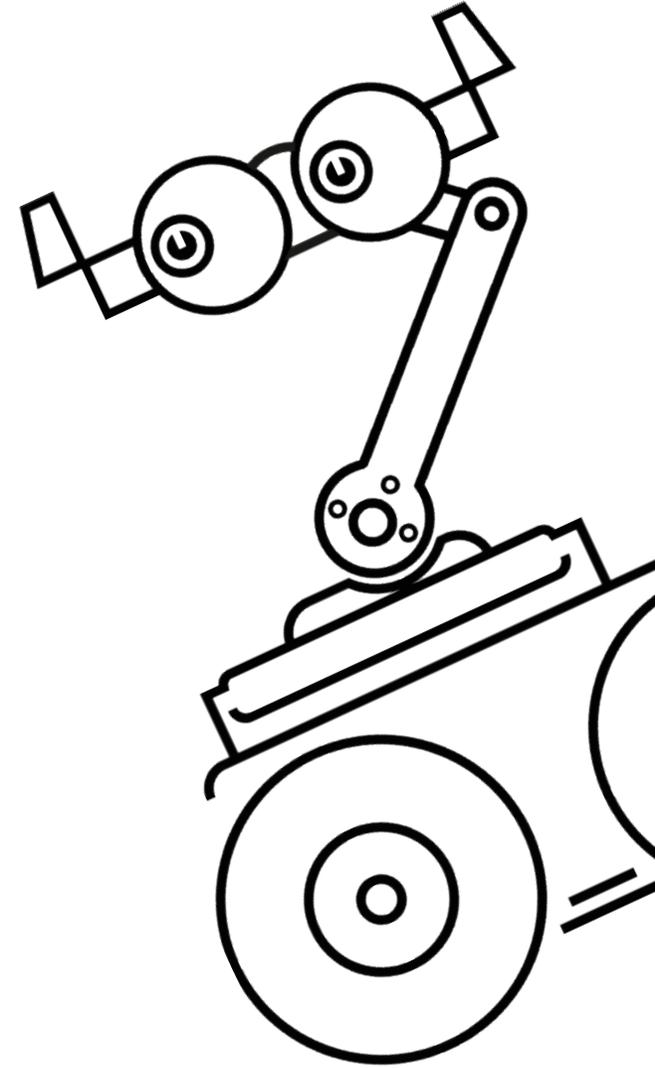

Best Practices

Open Roberta mit Lerngruppen

Calliope mini

- Verwalten von Benutzerkonten und Programmen
 - Überwachung des Programmablaufs und Fehlersuche
 - Textbasierte Programmiersprachen praxisnah kennenlernen
 - Übergang von Offline- zu Online-Coding
 - Einstieg und Methoden
-
- Links



Open Roberta Lab

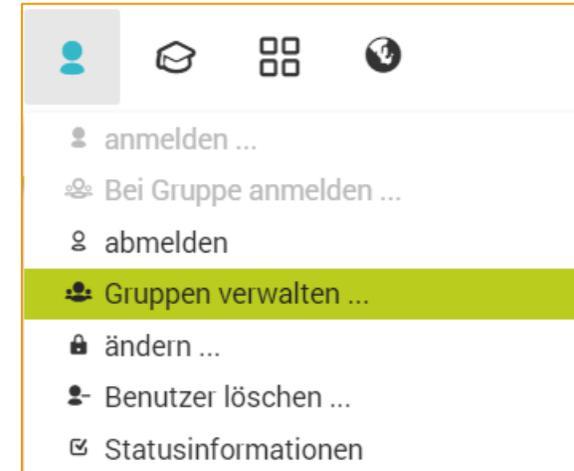
Verwalten von Benutzerkonten und Programmen

Das **Gruppen**-Feature richtet sich primär an Gruppenleiter*innen und sollen diese bei der Verwaltung von Lerngruppen unterstützen.

Vorteile des Gruppen-Features sind:

- Bereitstellung von Gruppen-Benutzerkonten für die Anmeldung, ohne Registrierung.
- Verwaltung von Programmen in der Cloud (Open Roberta Server).
- Teilen von Programmen mit der gesamten Gruppe oder einzelnen Gruppenmitgliedern.

! Die Verwaltung von Gruppen ist ausschließlich verifizierten Benutzer:innen des Open Roberta Labs zugänglich.



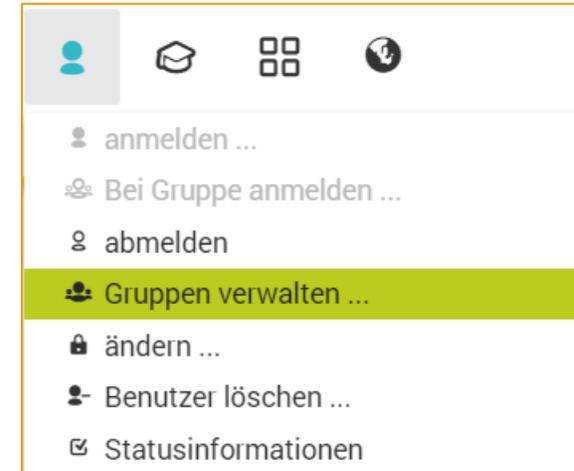
Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Einfacher Austausch von Programmen zwischen Gruppeneigentümer*in und den Gruppenmitgliedern

Gruppeneigentümer*in

- Mitglieder hinzufügen / entfernen
- Generierte Gruppenmitglieder erhalten als Benutzernamen eine Nummer zugewiesen, die vom Gruppeneigentümer*in angepasst werden können.
- Vergessene Passwörter können wieder auf das Initialpasswort zurückgesetzt werden.
- Programme können mit der gesamten Gruppe oder einzelnen Mitgliedern geteilt werden, sodass Gruppenmitglieder automatisch Zugriff darauf erhalten.
- Einzelne, mehrere Gruppemitglieder oder eine vollständige Gruppe – Mitglieder und Programme – können gelöscht werden.
- ! Einmal gelöscht, kann eine Gruppe nicht wiederhergestellt werden!



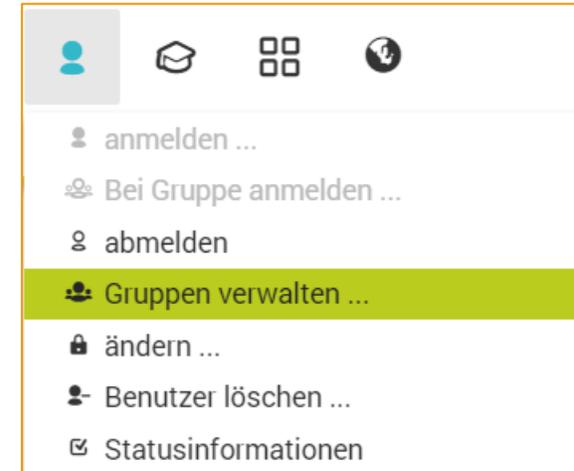
Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Einfacher Austausch von Programmen zwischen Gruppeneigentümer*in und den Gruppenmitgliedern

Gruppenmitglieder

- Programme, die die Mitglieder der Gruppe abspeichern sind automatisch mit dem*der Gruppeneigentümer*in geteilt, sodass diese eingesehen werden können.
- Programme können mit Gruppenmitglieder geteilt werden.
- ! Vergessene Passwörter können wieder auf das Initialpasswort zurückgesetzt werden.

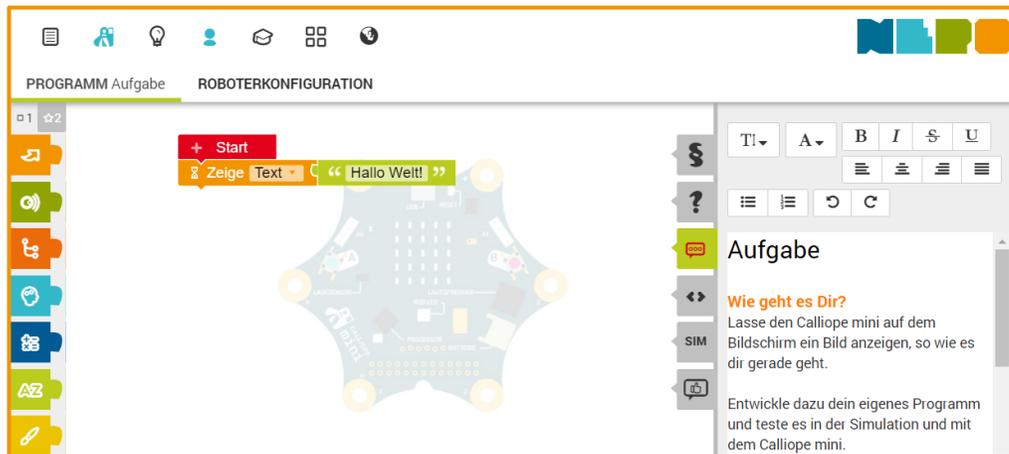


Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Gruppeneigentümer*in

1. Aufgabenstellung vorbereiten



The screenshot displays the Open Roberta Lab interface. At the top, there are navigation icons and the text "PROGRAMM Aufgabe" and "ROBOTERKONFIGURATION". The main workspace shows a Calliope mini robot with a program block: a red "Start" block followed by a yellow "Zeige Text" block containing the text "Hallo Welt!". To the right, there is a task description panel titled "Aufgabe" with the following text: "Wie geht es Dir? Lasse den Calliope mini auf dem Bildschirm ein Bild anzeigen, so wie es dir gerade geht. Entwickle dazu dein eigenes Programm und teste es in der Simulation und mit dem Calliope mini." The interface also includes a left sidebar with various tool icons and a top-right menu.

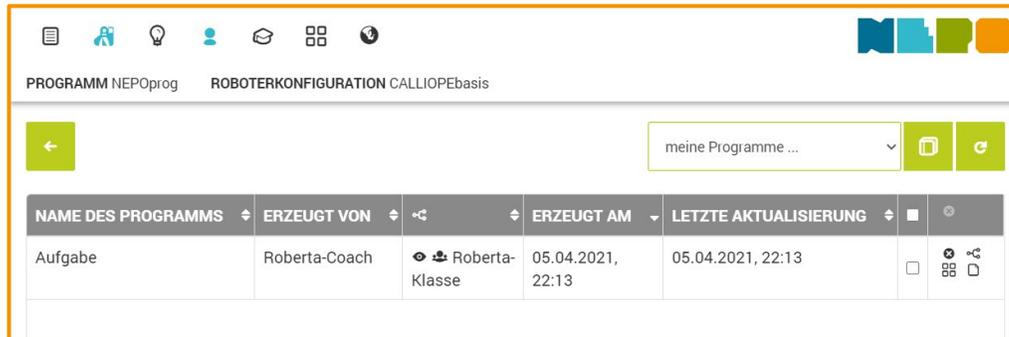
Gruppenmitglied

Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Gruppeneigentümer*in

1. Aufgabenstellung vorbereiten
2. Aufgabe mit **Gruppe** teilen



NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13

Gruppenmitglied

1. Anmeldung



Bei Gruppe anmelden

Benutzername des Gruppeneig: Roberta-Coach

Name der Gruppe: Roberta-Klasse

Benutzername: Roberta

Passwort: []

OK

Passwort zurücksetzen ...

ändere Passwort ...

Das Passwort deines Benutzerkontos ist noch nicht sicher. Setze dein eigenes Passwort, damit nur du dich bei deinem Konto anmelden kannst.

Neues Passwort: []

Passwort wiederholen: []

OK Abbrechen

Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Gruppeneigentümer*in

1. Aufgabenstellung vorbereiten
2. Aufgabe mit **Gruppe** teilen

NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13

BENUTZER	PASSWORT
Calli	Roberta-Klasse:Calli
Nao	Roberta-Klasse:Nao
Roberta	*****

Geänderte Passwörter von Mitgliedern sind nicht einsehbar!

Gruppenmitglied

1. Anmeldung
2. Programm mit Aufgabenstellung öffnen

NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13

Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Gruppeneigentümer*in

1. Aufgabenstellung vorbereiten
2. Aufgabe mit **Gruppe** teilen

The screenshot shows two parts of the Open Roberta Lab interface. The top part displays a list of programs under the heading 'PROGRAMM NEPOprog' and 'ROBOTERKONFIGURATION CALLIOPEbasis'. The bottom part shows the group settings for 'Roberta-Klasse'.

NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13

BENUTZER	PASSWORT
Calli	Roberta-Klasse:Calli
Nao	Roberta-Klasse:Nao
Roberta	*****

Geänderte Passwörter von Mitgliedern sind nicht einsehbar!

Gruppenmitglied

1. Anmeldung
2. Programm mit Aufgabenstellung öffnen

The screenshot shows the Open Roberta Lab interface with a program titled 'PROGRAMM HalloWelt' by 'Roberta-Coach' under 'ROBOTERKONFIGURATION'. The main workspace displays a Calliope mini robot and a block-based program with a 'Start' block and a 'Zeige | Text' block containing 'Hallo Welt!'. The right sidebar shows the task description.

Aufgabe

Wie geht es Dir?
Lasse den Calliope mini auf dem Bildschirm ein Bild anzeigen, so wie es dir gerade geht.

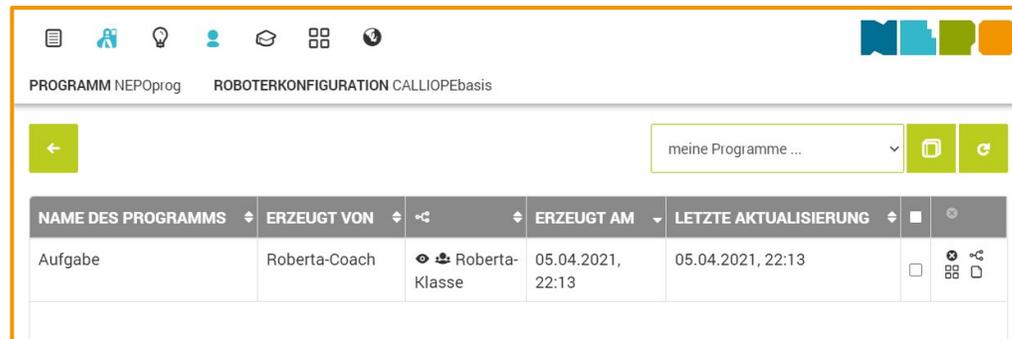
Entwickle dazu dein eigenes Programm und teste es in der Simulation und mit dem Calliope mini.

Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Gruppeneigentümer*in

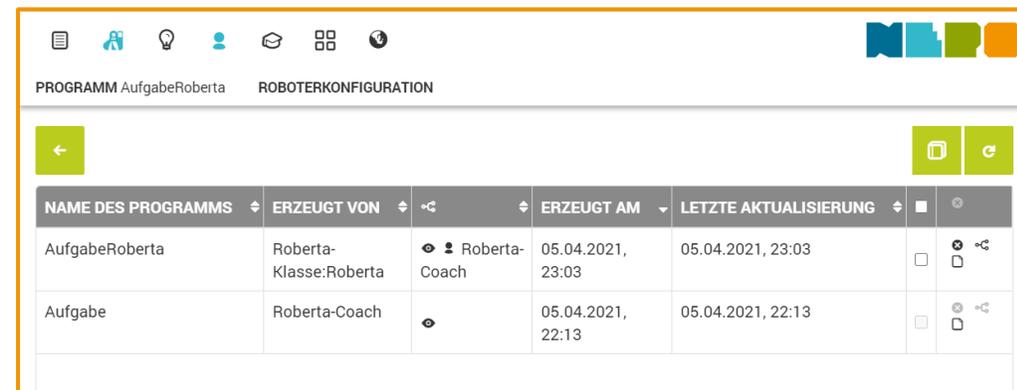
1. Aufgabenstellung vorbereiten
2. Aufgabe mit **Gruppe** teilen



NAME DES PROGRAMMS	ERZEUGT VON	GRUPPE	ERZEUGT AM	LETZTE AKTUALISIERUNG		
Aufgabe	Roberta-Coach	Roberta-Klasse	05.04.2021, 22:13	05.04.2021, 22:13	<input type="checkbox"/>	

Gruppenmitglied

1. Anmeldung
2. Programm mit Aufgabenstellung öffnen
3. Programm entwickeln/ergänzen
4. Beschreiben
5. »speichern unter ...« unter *Programm + „Mitglied“*



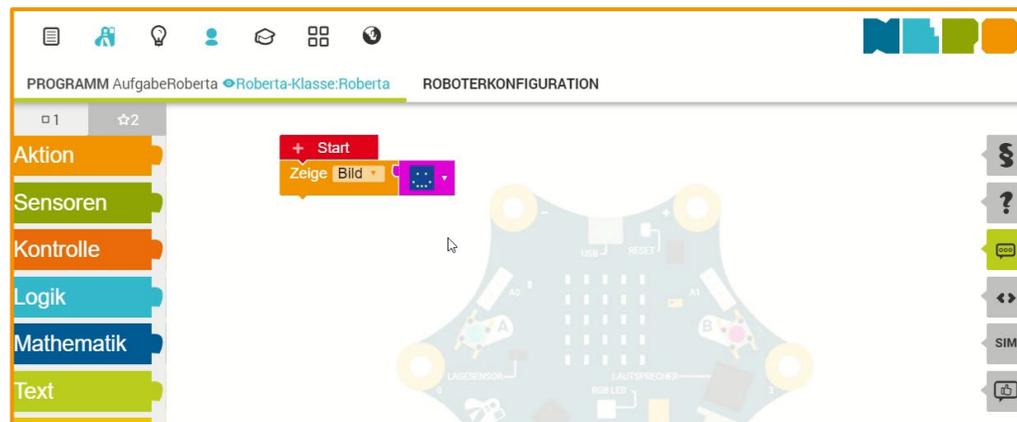
NAME DES PROGRAMMS	ERZEUGT VON	GRUPPE	ERZEUGT AM	LETZTE AKTUALISIERUNG		
AufgabeRoberta	Roberta-Klasse:Roberta	Roberta-Coach	05.04.2021, 23:03	05.04.2021, 23:03	<input type="checkbox"/>	
Aufgabe	Roberta-Coach	Roberta-Klasse	05.04.2021, 22:13	05.04.2021, 22:13	<input type="checkbox"/>	

Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Gruppeneigentümer*in

1. Aufgabenstellung vorbereiten
2. Aufgabe mit **Gruppe** teilen
3. Programm einsehen und kommentieren



Gruppenmitglied

1. Anmeldung
2. Programm mit Aufgabenstellung öffnen
3. Programm entwickeln/ergänzen
4. Beschreiben
5. »speichern unter ...« unter *Programm* + „Mitglied“

NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG		
AufgabeRoberta	Roberta-Klasse:Roberta	05.04.2021, 23:03	05.04.2021, 23:03	<input type="checkbox"/>	
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13	<input type="checkbox"/>	

Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Gruppeneigentümer*in

1. Aufgabenstellung vorbereiten
2. Aufgabe mit **Gruppe** teilen
3. Programm einsehen und kommentieren
4. »speichern unter ...« unter *Programm* + „Mitglied“ + „zurück“

Das Programm wird nun nicht mehr mit »Roberta-Klasse:Roberta« geteilt!

PROGRAMM NEPOprog ROBOTERKONFIGURATION

meine Programme ...

NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG
AufgabeRobertazurück	Roberta-Coach	05.04.2021, 23:19	06.04.2021, 23:15
AufgabeRoberta	Roberta-Klasse:Roberta	05.04.2021, 23:03	05.04.2021, 23:03
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13

Gruppenmitglied

1. Anmeldung
2. Programm mit Aufgabenstellung öffnen
3. Programm entwickeln/ergänzen
4. Beschreiben
5. »speichern unter ...« unter *Programm* + „Mitglied“

PROGRAMM AufgabeRoberta ROBOTERKONFIGURATION

meine Programme ...

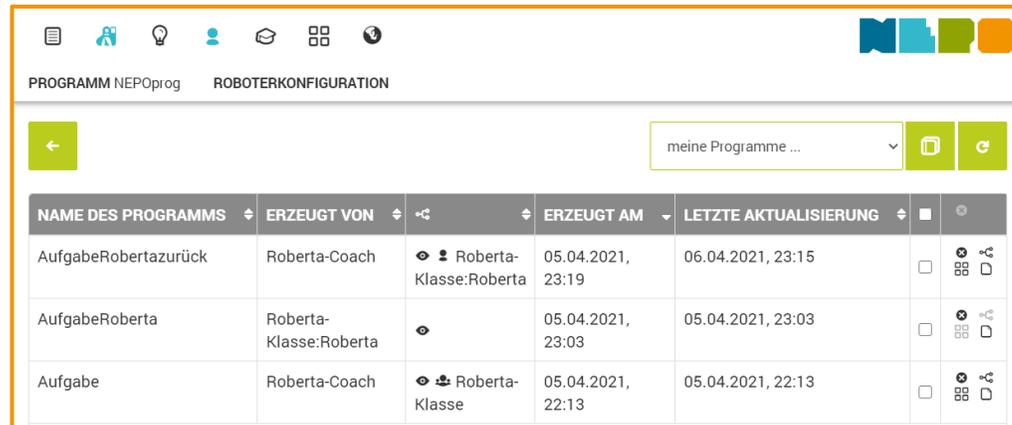
NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG
AufgabeRoberta	Roberta-Klasse:Roberta	05.04.2021, 23:03	05.04.2021, 23:03
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13

Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Gruppeneigentümer*in

1. Aufgabenstellung vorbereiten
2. Aufgabe mit **Gruppe** teilen
3. Programm einsehen und kommentieren
4. »speichern unter ...« unter *Programm* + „Mitglied“ + „zurück“
5. Teilen mit **Mitglied**

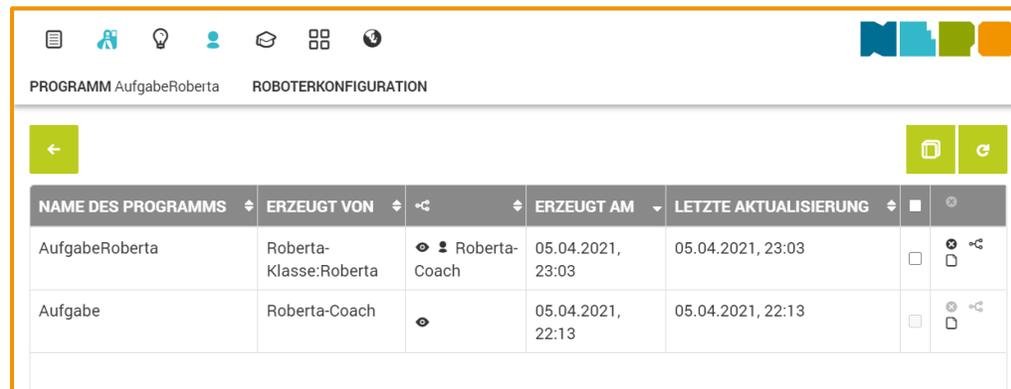


The screenshot shows the 'ROBOTERKONFIGURATION' page for the program 'NEPOprog'. A search bar contains 'meine Programme ...'. Below is a table with the following data:

NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG
AufgabeRobertazurück	Roberta-Coach	05.04.2021, 23:19	06.04.2021, 23:15
AufgabeRoberta	Roberta-Klasse:Roberta	05.04.2021, 23:03	05.04.2021, 23:03
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13

Gruppenmitglied

1. Anmeldung
2. Programm mit Aufgabenstellung öffnen
3. Programm entwickeln/ergänzen
4. Beschreiben
5. »speichern unter ...« unter *Programm* + „Mitglied“



The screenshot shows the 'ROBOTERKONFIGURATION' page for the program 'AufgabeRoberta'. Below is a table with the following data:

NAME DES PROGRAMMS	ERZEUGT VON	ERZEUGT AM	LETZTE AKTUALISIERUNG
AufgabeRoberta	Roberta-Klasse:Roberta	05.04.2021, 23:03	05.04.2021, 23:03
Aufgabe	Roberta-Coach	05.04.2021, 22:13	05.04.2021, 22:13

Open Roberta Lab

Verwalten von Benutzerkonten und Programmen

Hinweis

Keine Angabe von personenbezogenen Daten für Gruppenmitglieder (wie Vor- oder Nachname), wenn dafür keine schriftliche Einverständniserklärung des Gruppenmitglieds bzw. einer erziehungsberechtigten Person vorliegt.

Empfehlung

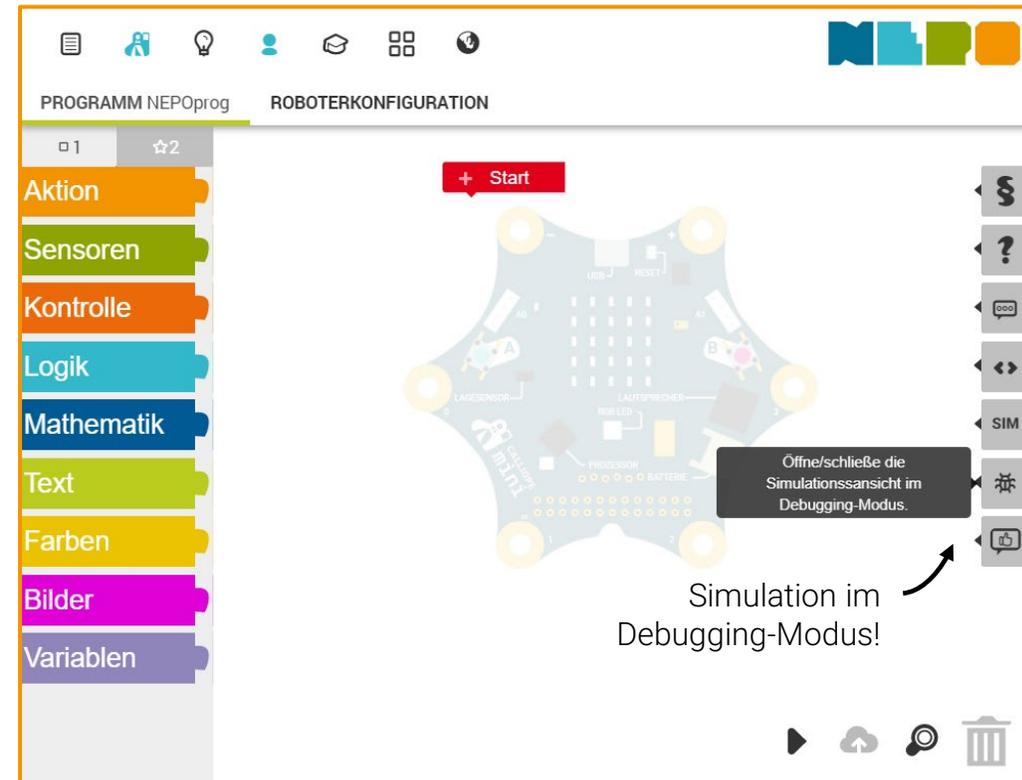
Verwendung von Pseudonymen aus denen sich die Personen dahinter nicht ohne Weiteres ableiten lassen.

Open Roberta Lab

Überwachung des Programmablaufs und Fehlersuche

Debugging dient der Fehlersuche und -behebung in NEPO-Programmen und kann dabei unterstützen, einen visuellen Programmablauf darzustellen.

- Schrittweise bzw. blockweise Ausführung des Programms.
- Farbliche Hervorhebung des NEPO-Blocks, der aktuell ausgeführt wird.
- Setzen von Haltepunkten, bei denen das Programm automatisch im Ablauf anhält bis manuell der Ablauf erneut gestartet wird.



Open Roberta Lab

Überwachung des Programmablaufs und Fehlersuche

 Blockweise Ausführung der NEPO-Blöcke bis Programmende oder der nächste Haltepunkt erreicht wird. Weitere Ausführung durch erneutes Anstarten.

 Ausführung eines NEPO-Blocks
! Funktionen werden sofort und als ein Block ausgeführt.

 Ausführung eines NEPO-Blocks
! Funktionen werden nicht sofort ausgeführt, sondern in diese wird hineingesprungen und kann blockweise ausgeführt werden.

```

+ Start
- Variable Startwert : Zahl ← 5
- Variable Endwert : Zahl ← 9
zaehle
  von Startwert
  bis Endwert
Schreibe Startwert 1
Schreibe Endwert 6
zaehle
  von Startwert
  bis Endwert
    
```

```

+ Start
- Variable Startwert : Zahl ← 5
- Variable Endwert : Zahl ← 9
zaehle
  von Startwert
  bis Endwert
Schreibe Startwert 1
Schreibe Endwert 6
zaehle
  von Startwert
  bis Endwert
    
```

↖ Haltepunkt

```

+ zaehlen mit:
- Variable von : Zahl
- Variable bis : Zahl
Wiederhole bis von = bis
mache
  Zeige Zeichen von
  Schreibe von von + 1
    
```

```

+ zaehlen mit:
- Variable von : Zahl
- Variable bis : Zahl
Wiederhole bis von = bis
mache
  Zeige Zeichen von
  Schreibe von von + 1
    
```

Open Roberta Lab

Überwachung des Programmablaufs und Fehlersuche



Öffnen/Schließen der Ansicht der aktuellen Werte der Sensoren und Variablen, falls im Programm verwendet.

```

+ Start
- Variable Startwert : Zahl ← 5
- Variable Endwert : Zahl ← 9

zaehle
von Startwert
bis Endwert
Schreibe Startwert 1
Schreibe Endwert 6

zaehle
von Startwert
bis Endwert
    
```

aktuelle Werte von globalen und lokalen Variablen

+ System-Werte

+ Zeitgeber-Werte

- Sensor-Werte

Lage: aufrecht
 Kompasssensor: 0°
 Lichtsensor: 0%
 Temperatursensor: 20°
 Geräuschsensor: 2%

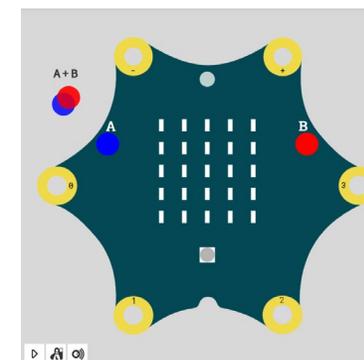
- Variablen-Werte

Startwert: 5
 Endwert: 9
 bis: 9
 von: 5

```

+ zaehlen mit:
- Variable von : Zahl
- Variable bis : Zahl

Wiederhole bis
mache
  Zeige Zeichen von
  Schreibe von von + 1
    
```

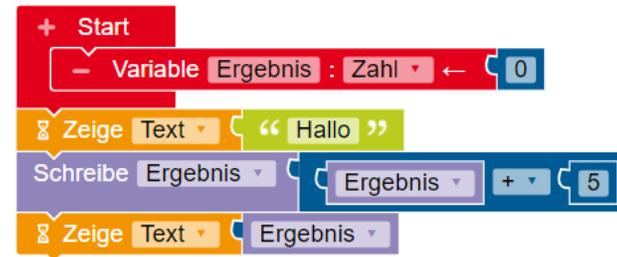


Open Roberta Lab

Textbasierte Programmiersprachen praxisnah kennenlernen

Die **Quellcode**-Einsicht dient als Brücke zu textbasierten Programmiersprachen und ermöglicht es, praxisnah bis zur Nutzung des **Quellcodeeditors** Erfahrungen zu sammeln.

- Quellcode einsehen und selbständig ändern
- Erste textbasierte Programme erstellen, ohne ganz von vorne anfangen zu müssen:
 - grobe Struktur des Programms mit NEPO erstellen,
 - z.B. mathematische Operationen im Quellcode hinzuzufügen



```

1. #define _GNU_SOURCE
2.
3. #include "MicroBit.h"
4. #include "NEPODefs.h"
5. #include <list>
6. #include <array>
7. #include <stdlib.h>
8. MicroBit_uBit;
9.
10.
11. double __Ergebnis;
12.
13. int main()
  
```

```

1. #define _GNU_SOURCE
2.
3. #include "MicroBit.h"
4. #include "NEPODefs.h"
5. #include <list>
6. #include <array>
7. #include <stdlib.h>
8. MicroBit_uBit;
9.
10.
11. double __Ergebnis;
12.
  
```

Textbasierte Programmiersprachen praxisnah kennenlernen

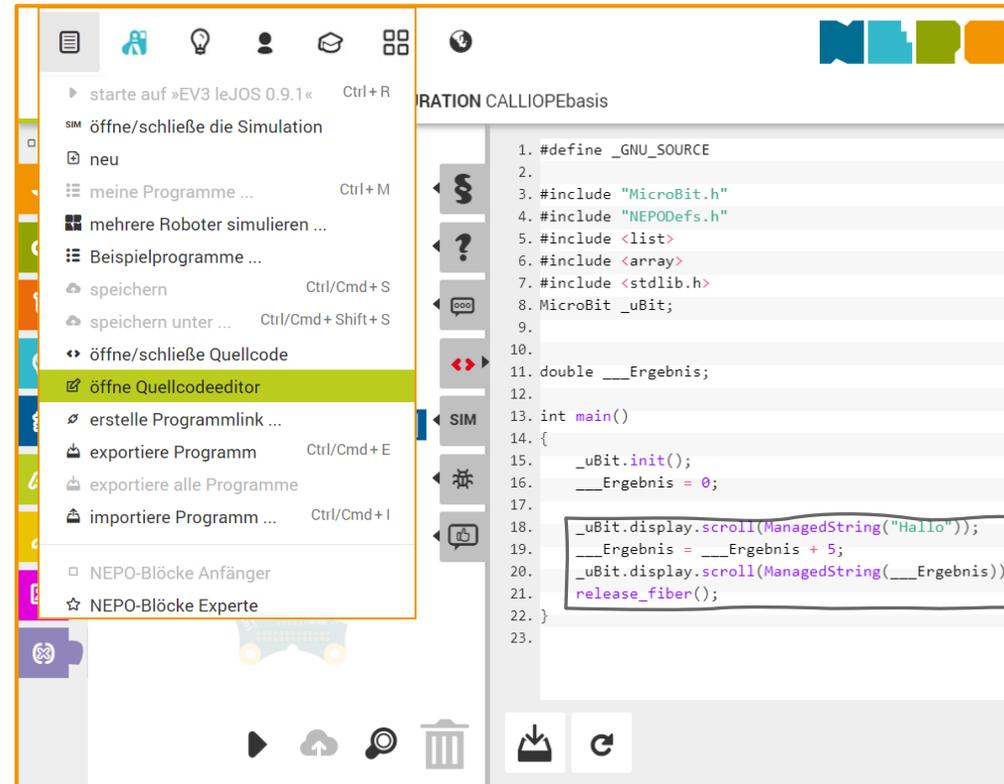
- Quellcode einsehen
 - Quellcodeansicht
- ! Der Quellcode wird entweder während des Öffnens der Ansicht aktualisiert oder manuell über die ↻-Schaltfläche.

The screenshot displays the Open Roberta Lab interface. On the left, a block-based program is visible with the following steps: Start, Variable 'Ergebnis' Zahl ← 0, Zeige Text « Hallo », Schreibe Ergebnis Ergebnis + 5, and Zeige Text Ergebnis. On the right, the corresponding C++ source code is shown:

```
1. #define __GNU_SOURCE
2.
3. #include "MicroBit.h"
4. #include "NEPODefs.h"
5. #include <list>
6. #include <array>
7. #include <stdlib.h>
8. MicroBit _uBit;
9.
10.
11. double __Ergebnis;
12.
13. int main()
14. {
15.     _uBit.init();
16.     __Ergebnis = 0;
17.
18.     _uBit.display.scroll(ManagedString("Hallo"));
19.     __Ergebnis = __Ergebnis + 5;
20.     _uBit.display.scroll(ManagedString(__Ergebnis));
21.     release_fiber();
22. }
23.
```

Textbasierte Programmiersprachen praxisnah kennenlernen

- Quellcode einsehen und selbständig ändern
 - Quellcodeeditor

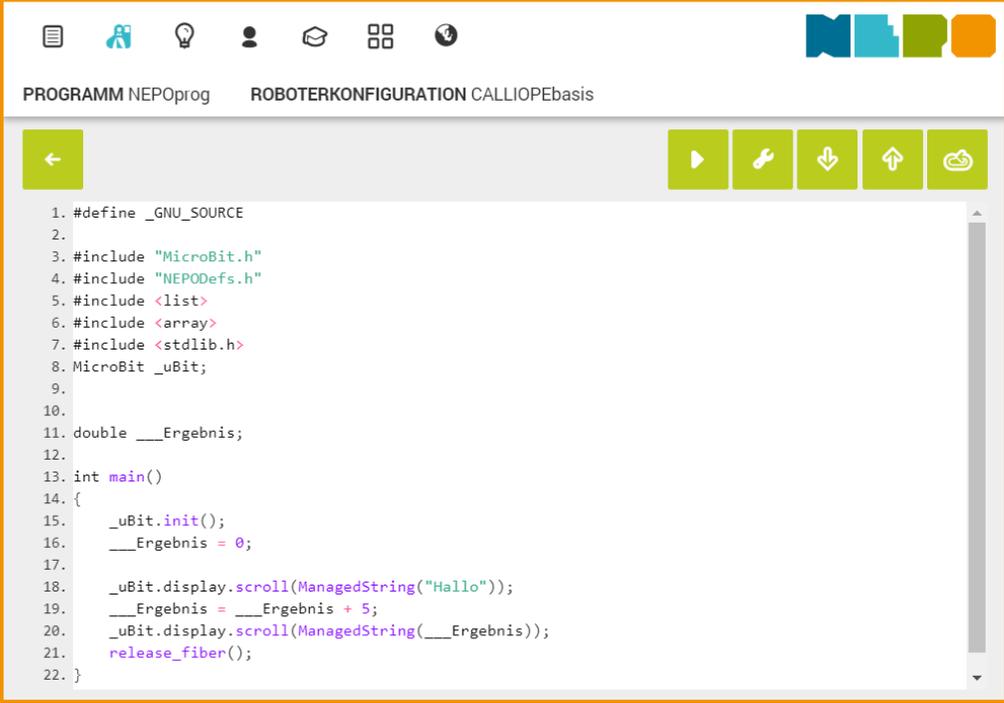


Textbasierte Programmiersprachen praxisnah kennenlernen

■ Quellcode einsehen und selbständig ändern

■ Quellcodeeditor

-  Ausführbares Programm, z.B. auf das System herunterladen und starten
-  Quellcode in Maschinensprache übersetzen und prüfen
-  Quellcode lokal herunterladen
-  Quellcode hochladen
-  Aktuelles NEPO-Programm importieren
 - aktualisieren des Quellcodes
 - zurücksetzen von Änderungen im Quellcode

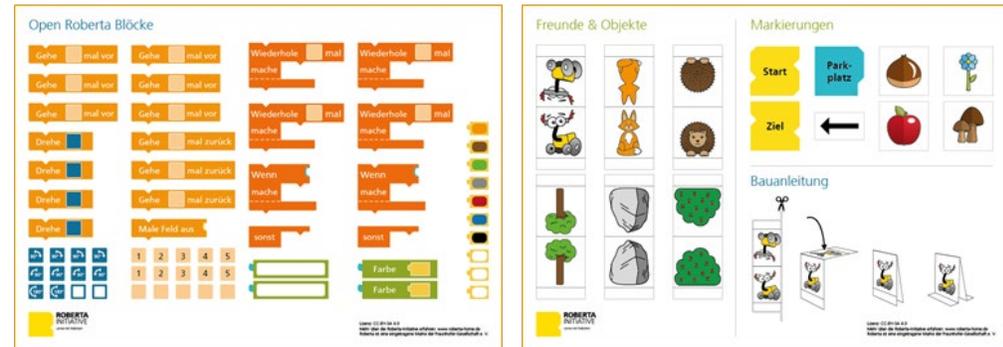


```
1. #define _GNU_SOURCE
2.
3. #include "MicroBit.h"
4. #include "NEPODefs.h"
5. #include <list>
6. #include <array>
7. #include <stdlib.h>
8. MicroBit _uBit;
9.
10.
11. double __Ergebnis;
12.
13. int main()
14. {
15.     _uBit.init();
16.     __Ergebnis = 0;
17.
18.     _uBit.display.scroll(ManagedString("Hallo"));
19.     __Ergebnis = __Ergebnis + 5;
20.     _uBit.display.scroll(ManagedString(__Ergebnis));
21.     release_fiber();
22. }
```

Übergang von Offline- zu Online-Coding Programmierpapier

Programmieren ganz analog - **ohne technische Hürden** - Grundprinzipien der Programmierung vermitteln

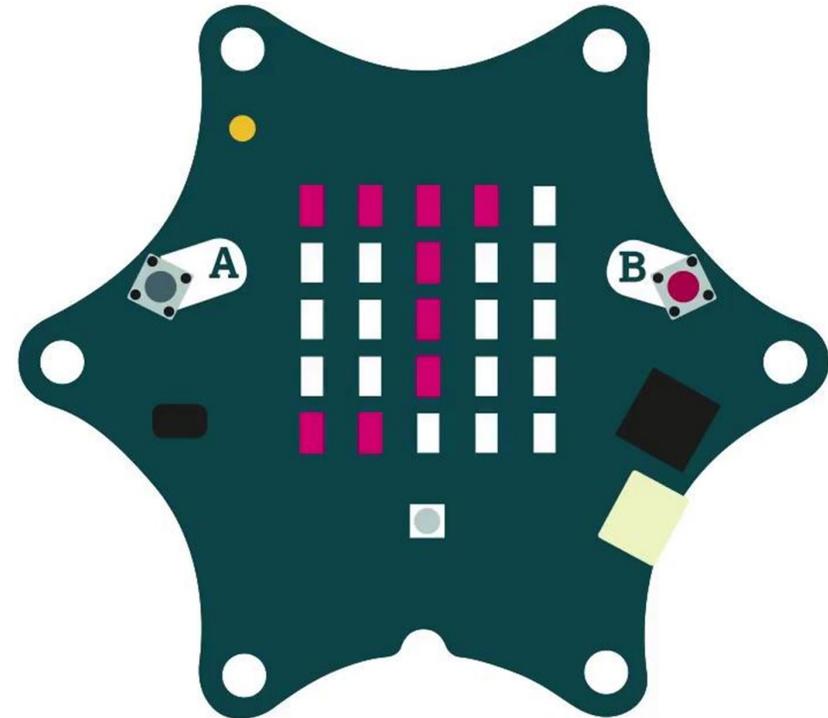
- Bestandteile einer Anweisung
- Ablauf
- Entscheidungen



Einstieg und Methoden

Beispiel

- **Aufgabe**
Orakel mit zwei Antwortmöglichkeiten
- **Eingabe**
 - Schütteln des Calliope mini
- **Ausgabe**
 - Antworttext anzeigen - **JA** oder **NEIN**



Einstieg und Methoden

Beispiel

■ Aufgabe

Orakel mit zwei Antwortmöglichkeiten

■ Eingabe

- Schütteln des Calliope mini

■ Ausgabe

- Antworttext anzeigen - **JA** oder **NEIN**

■ Zerlegung

- Anzeige einer Antwortmöglichkeit
- Schütteln des Calliope mini erkennen
- Zufällig Zahl zwischen 1 und 2 ermitteln mit der Bedingung Zufallszahl = 1 = JA
- Nicht-Erfüllung der Bedingung Zufallszahl \neq 1 = NEIN



Hinweis:

Die Ermittlung der Zufallszahl und deren Auswertung sind in einem Zerlegungsschritt zusammengefasst. Um dies aufzulösen, ist die Empfehlung einen Zwischenschritt mit der Ausgabe der Zufallszahl einzubauen.

[Empfehlung: Die Erstellung einer Zufallszahl in einer eigenen Aufgabe (Programm) zu erarbeiten.]

Einstieg und Methoden

Beispiel

- **Aufgabe**
Orakel mit zwei Antwortmöglichkeiten
- **Eingabe**
- **Verarbeitung**
 - 1. Schritt: Sequenziell ausführen
- **Ausgabe**
 - 1. Schritt: Antworttext **JA** anzeigen

- **Programm**

- 1. Schritt:



Einstieg und Methoden

Beispiel

- **Aufgabe**
Orakel mit zwei Antwortmöglichkeiten
- **Eingabe**
 - 2. Schritt: **Lagesensor** auslesen
- **Verarbeitung**
 - 1. Schritt: Sequenziell ausführen
 - 2. Schritt: Warte auf **Schütteln**
- **Ausgabe**
 - 1. Schritt: Antworttext **JA** anzeigen

■ Programm

■ 1. Schritt:



■ 2. Schritt



Einstieg und Methoden

Beispiel

- **Aufgabe**
Orakel mit zwei Antwortmöglichkeiten
- **Eingabe**
 - 2. Schritt: **Lagesensor** auslesen
- **Verarbeitung**
 - 1. Schritt: Sequenziell ausführen
 - 2. Schritt: Warte auf **Schütteln**
 - 3. Schritt: Entscheide **zufällig** für einen Fall
- **Ausgabe**
 - 1. Schritt: Antworttext **JA** anzeigen

■ Programm

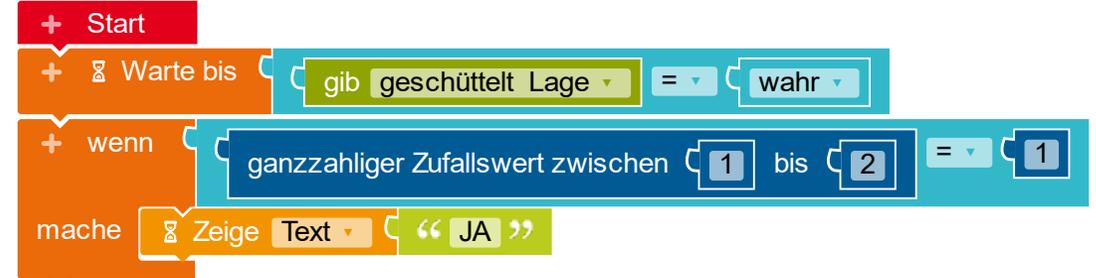
■ 1. Schritt:



■ 2. Schritt



■ 3. Schritt



Einstieg und Methoden

Beispiel

■ Aufgabe

Orakel mit zwei Antwortmöglichkeiten

■ Eingabe

■ 2. Schritt: **Lagesensor** auslesen

■ Verarbeitung

■ 1. Schritt: Sequenziell ausführen

■ 2. Schritt: Warte auf **Schütteln**

■ 3. Schritt: Entscheide **zufällig falls ...**

■ 4. Schritt: Entscheide **zufällig falls ... sonst ...**

■ Ausgabe

■ 1. Schritt: Antworttext **JA** anzeigen

■ 4. Schritt: Antworttext **JA** oder **NEIN** anzeigen

■ Programm

■ 4. Schritt:



■ Optionaler Schritt



Einstieg und Methoden

Programmvorgabe

Programmablauf lesen, verstehen und verändern

- Programmablauf ausprobieren, beobachten und beschreiben

Nutzung von bekannten Formulierungsmöglichkeiten, um Schritt für Schritt zu beschreiben, z.B.:

- Alltagssprache
- Programmablaufplan



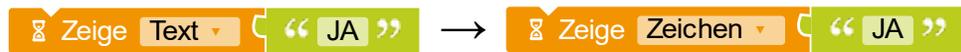
Einstieg und Methoden

Programmvorgabe

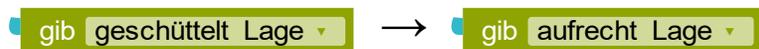
Programmablauf lesen, verstehen und verändern

- Programmablauf ausprobieren, beobachten und beschreiben
- Parameter verändern

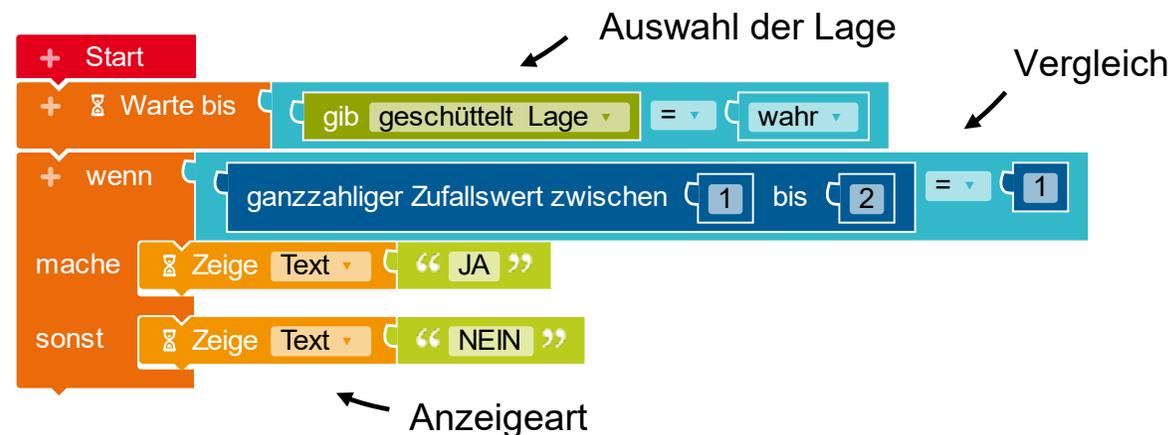
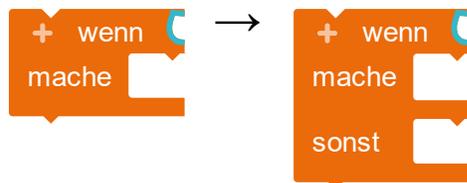
Ausgabe



Eingabe



Verarbeitung

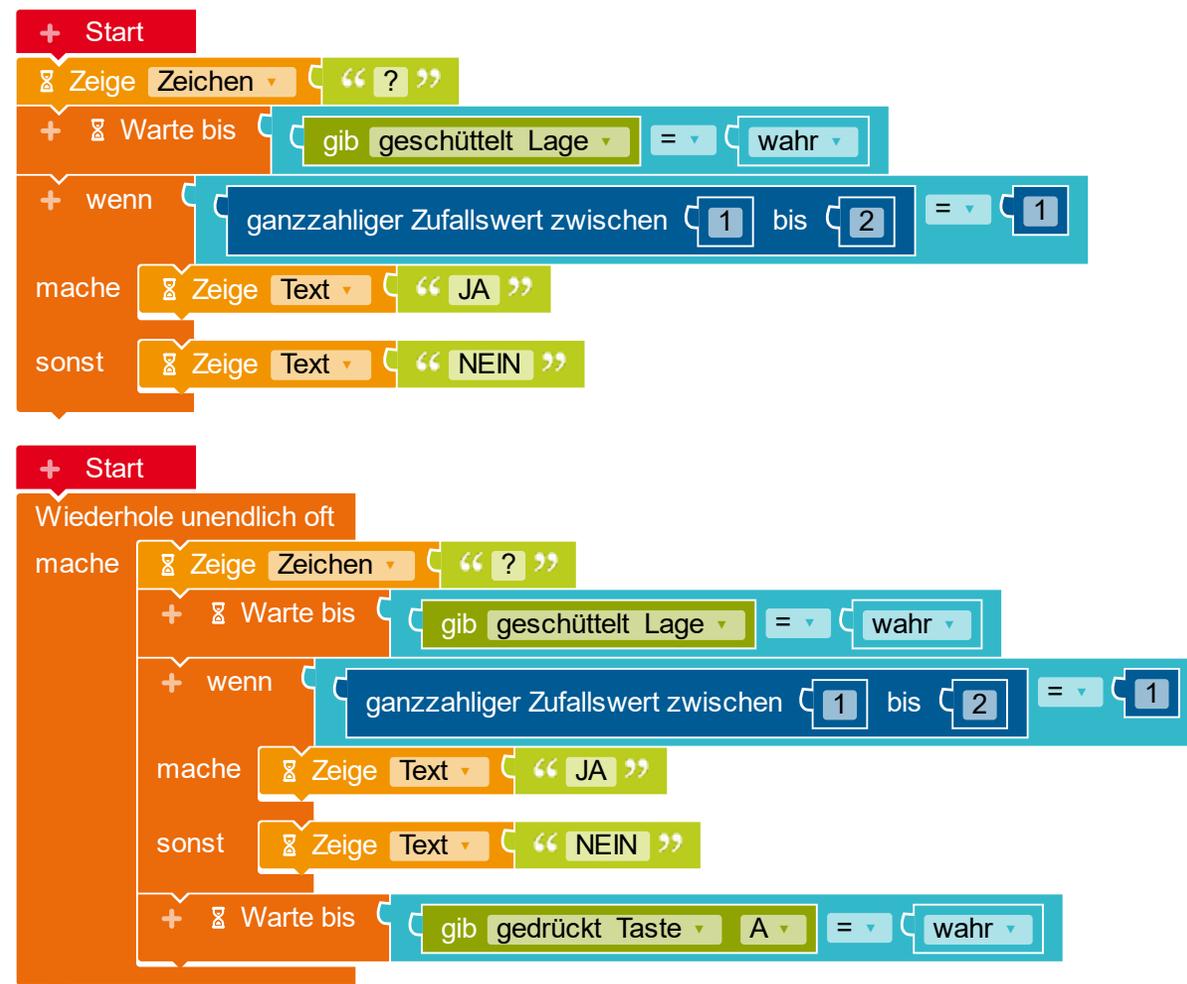


Einstieg und Methoden

Programmvorgabe

Programmablauf lesen, verstehen und verändern

- Programmablauf ausprobieren und mit eigenen Worten Verhalten beschreiben
- Parameter verändern
- Programm individualisieren sowie mit eigenen Ideen ergänzen
 - Ausgabe des Fragezeichensymbol, als Aufforderung die Frage ans Orakel zustellen
 - Programmablauf auf Tastendruck wiederholen



Einstieg und Methoden

Puzzle aus NEPO-Blöcken

Programm aus vorausgewählten NEPO-Blöcken zusammensetzen

- Benötigte Blöcke abbilden

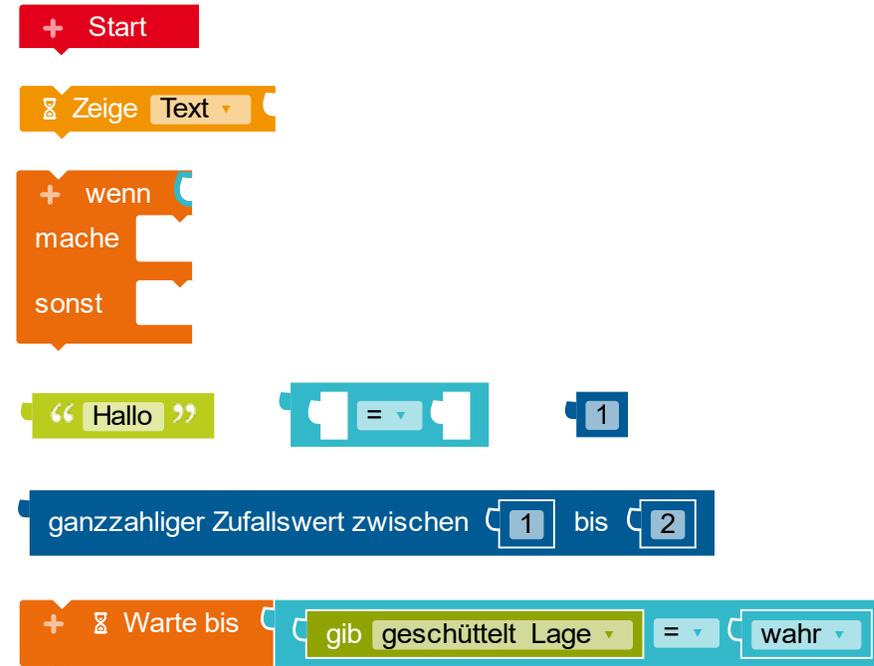


Einstieg und Methoden

Puzzle aus NEPO-Blöcken

Programm aus vorausgewählten NEPO-Blöcken zusammensetzen

- Benötigte Blöcke abbilden
- Differenzierung über die Kombination der bereitgestellten NEPO-Blöcke

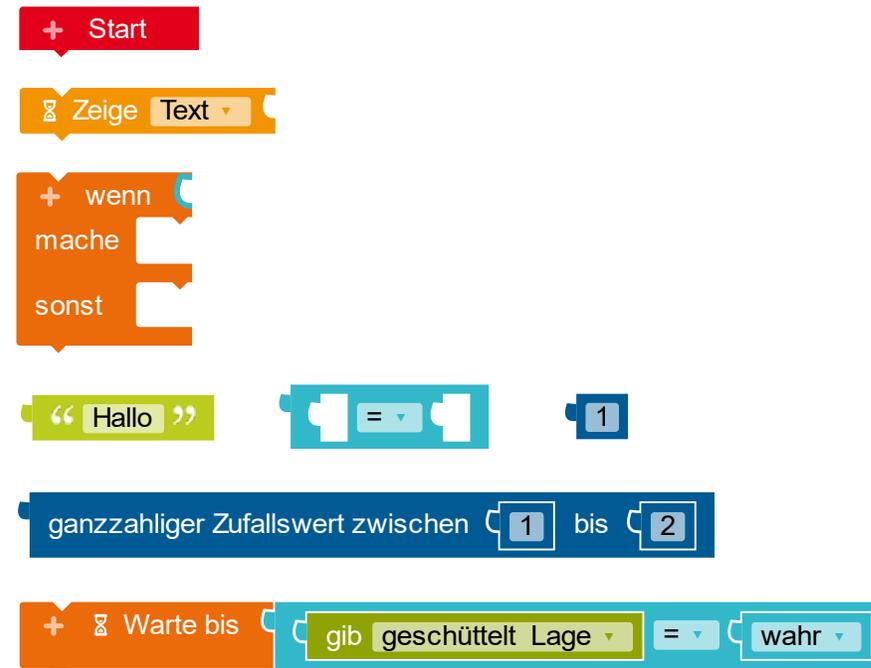


Einstieg und Methoden

Puzzle aus NEPO-Blöcken

Programm aus vorausgewählten NEPO-Blöcken zusammensetzen

- Benötigte Blöcke abbilden
- Differenzierung über die Kombination der bereitgestellten NEPO-Blöcke
- Erweiterung durch eigene Blöcke, z.B.
 - aus der Kategorie **Aktion**
 - usw.



Einstieg und Methoden

Pseudocode

Nach Anweisungen ein eigenes Programm entwickeln

- Ansicht der Programmdokumentation
- Formulierte Anweisungen geben den Programmablauf vor
- Differenzierung über die Ausführung der Anweisungen
- Zusätzliche Hinweise als Hilfestellung

The screenshot shows a software development environment. At the top, there is a red button labeled '+ Start'. Below it, a green tooltip contains the text 'Bitte schau in die Programmdokumentation.' The main area is a code editor with a toolbar at the top containing various icons for text formatting (bold, italic, underline, etc.) and a list icon. The code editor displays the following text:

```

Orakel
Erstelle ein Programm, dass der Calliope mini als Orakel dir eine Frage mit JA oder NEIN beantwortet.

1. Der Text JA soll auf dem Bildschirm angezeigt werden.
2. JA wird angezeigt, falls der Calliope mini geschüttelt wird.
3. Wenn eine zufällig ermittelte Zahl zwischen 1 und 2 = 1 ist, dann wird JA angezeigt.
4. Ist die zufällig ermittelte Zahl nicht = 1, dann wird NEIN angezeigt.
  
```

Einstieg und Methoden

Selbstlerneinheiten

Lernkarten

Kreative Schritt-für-Schritt Anleitungen zum Einstieg und ersten Programmierideen

E1 Calliope mini EINFÜHRUNG
Hier findest du eine Übersicht aller Sensoren und Eingabemöglichkeiten des Calliope mini.

E2 Calliope mini EINFÜHRUNG
Open Roberta Lab
Robotersystem im »Open Roberta Lab«
ke der Programmiersprache »NEPO«.

E3 Calliope mini EINFÜHRUNG
Open Roberta Lab
In der Roboterkonfiguration findest du anfänglich Sensoren und Aktoren mit ihren vorgegebenen Namen, die auf dem Calliope mini verbaut sind.

E4 Calliope mini EINFÜHRUNG
Open Roberta Lab
Dein Programm ist fertig und du willst es zum Ausprobieren auf den Calliope mini übertragen?

Schritt 1 Klicke auf die ► Schaltfläche.

Schritt 2 Folge den Hinweisen des neu geöffneten Fensters und klicke mit der rechten Maustaste auf den blauen unterstrichenen Link.

Schritt 3 Wähle »Link speichern unter...« aus und sichere dein Programm auf dem Laufwerk »MINI«.

Schritt 4 Beobachte die Statusleuchte.

Einstieg und Methoden

Selbstlerneinheiten

Tutorials

Interaktive Schritt-für-Schritt-Anleitungen mit unterstützenden Quizzes





Hallo Calliope

Tutorial 1: Lasse dich von Calliope mini begrüßen und schreibe dein erstes Programm.

Lernziel: Lerne erste Programmieranweisungen kennen

Vorkenntnisse: Du solltest wissen, wie man die Simulation öffnet und ein Programm schreibt.

	30 min
	ab 8 Jahre
SIM	ja
	☆☆☆



Musik mit Schleifen

Tutorial 2: Mache Musik mit Calliope und lerne dabei die Schleifen kennen! Schreibe ein Programm, das ein Lied spielt.

Lernziel: Lerne Schleifen kennen und probiere sie aus!

Vorkenntnisse: Du solltest das erste Tutorial »Hallo Calliope« bereits erfolgreich durchgeführt haben.

	30 min
	ab 8 Jahre
SIM	nein
	☆☆☆

1
2
3
4
5
6
7
8

Musik mit Schleifen ✕

+ Start



Schritt 1: Der erste Ton




Musik mit Schleifen

Tutorial 2: Mache Musik mit Calliope und lerne dabei die Schleifen kennen! Schleifen sind sog. Kontrollstrukturen und helfen dir dabei dein Programm übersichtlich und einfach zu gestalten.

Kennst du das Lied »Bruder Jakob«? Falls nicht, kein Problem, am Ende dieses Tutorials wird Calliope mini dieses Lied für dich spielen!

Lernziel: Lerne Schleifen kennen und probiere sie aus!

Vorkenntnisse: Du solltest das erste Tutorial »Hallo Calliope« bereits erfolgreich durchgeführt haben. Wenn das schon etwas länger her ist, kannst du es auch einfach noch einmal machen.

	30 min
	ab 8 Jahre
SIM	nein
	☆☆☆
	Calliope mini



Einstieg und Methoden

Selbstlerneinheiten

Galerie

Interaktive Schritt-für-Schritt-Anleitungen mit unterstützenden Quizzes

The screenshot shows a gallery interface for the Roberta Initiative. At the top, there is a navigation bar with icons for home, search, ideas, user, messages, a grid view (highlighted), and a refresh icon. Below this is a search bar containing 'Calliope mini' and 'roberta_zentrale', and a sorting dropdown set to 'Neueste zuerst'. The main area displays a grid of project cards, each featuring a white icon of a person with a camera, the project title, the creator 'von: roberta_zentrale', the creation date 'erstellt:', view count (eye icon), and like count (heart icon). Some cards also show tags like 'BosonKit' and 'Boson'.

Project Title	Created	Views	Likes	Tags
Advent_StilleNacht	09.12.2021, 17:06	516	6	
Advent_Schneekugel	01.12.2021, 11:36	333	0	
Advent_Farbenspiel	30.11.2021, 12:50	235	0	
back2school_Pause	01.09.2021, 16:26	86	2	
BosonKit_Feuchtigke	12.12.2020, 10:59	172	1	BosonKit, Boson
BosonKit_Tür	12.12.2020, 10:56	112	2	BosonKit, Boson
BosonKit_Servomoto	
BosonKit_Ventilator	
BosonKit_Alarm	
Grove_Ultraschallsen	
Grove_Feuchtigkeits	
Raketenstart	

Links

Roberta-Initiative

■ Roberta-Initiative



■ Roberta-Schulungen



■ Roberta-Material



■ Roberta-Newsletter



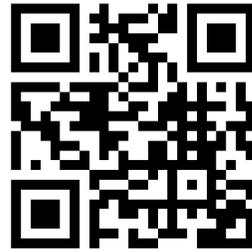
■ Transparenz



Links

Open Roberta

■ Open Roberta Projekt



■ Open Roberta FAQ



■ Open Roberta Lab



■ Open Roberta Group



■ Open Roberta Wiki



■ Open Roberta Github



»Roberta[®] – Lernen mit Robotern«

die Bildungsinitiative des Fraunhofer-Instituts für
Intelligente Analyse- und Informationssysteme IAIS