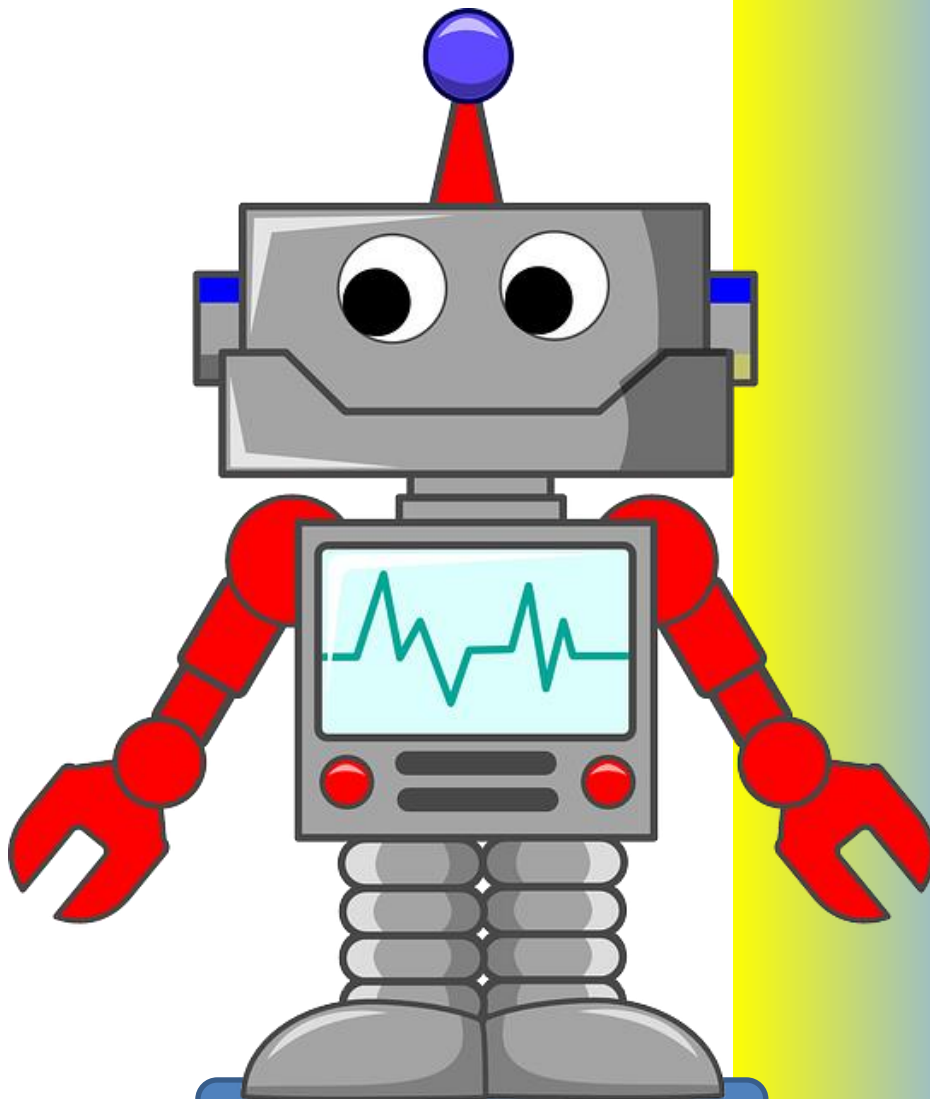


3

# ROBOTER

Arbeiten mit den InOBots



LEHRERBAND



Hermann Milchram

29.11.2021







**Bildnachweis:**

sofern nicht anders angegeben befinden sich die Grafiken im Eigentum von NÖ Media oder sind eigene Aufnahmen und Zeichnungen oder Screenshots von Programmen.

Roboter auf der Titelseite robot-312566\_640 Quelle Pixabay CC0

LEDs led-306562\_640 Quelle Pixabay CC0

zielscheibe-ziel-bogenschießen-2304567 Quelle Pixabay CC0

smiley-1914523\_640 Quelle Pixabay CC0


robot-1470108\_640 Quelle Pixabay CC0

Herzlichen Dank für das **Korrekturlesen** an **Ilse Doppler** und **Angela Kampichler**!





# INHALTSVERZEICHNIS

INHALTSVERZEICHNIS.....	1
1. INO-BOT.....	2
Arbeiten mit mehreren InO-Bots im Klassenverband .....	3
Programmierung unter Verwendung der InO-Bot APP  .....	4
Installation und Ausführen der InO-Bot Blockly APP unter Windows .....	5
Erste Programme mit der InO-Bot APP .....	6
Kontrollstrukturen: Schleife (ITERATION): .....	7
Ausgabe von Tönen und Geräuschen (Sounds):.....	9
Steuerung der Front-LEDs.....	10
RGB LEDs – Little light show .....	11
SENSOREN: Lichtsensor .....	12
SENSOREN: Distanzsensor .....	13
Programmierung mit Scratch.....	14



# 1. InO-Bot

Alter: **8+** (Bezugsquellen: <http://www.tts-group.co.uk> ca. 115€)

Der "Input-Output-Roboter" (kurz **InO-Bot**) wurde speziell für die Verwendung mit der bekannten Programmiersprache „Scratch“ entwickelt.



InO-Bot: H. Milchram 2017

Der **InO-Bot** kann über Bluetooth mit dem Computer verbunden werden. Die Software kann direkt mit dem InO-Bot kommunizieren, wodurch die Kinder die programmierten Befehle in Echtzeit beobachten können. Die transparente Verkleidung ermöglicht einen Einblick in die technischen Komponenten des InO-Bots. Der InO-Bot wird mit einem wieder aufladbaren Akku betrieben. Das **Aufladen** erfolgt mit einem Standard **USB-A → USB-B** Kabel.

## Technische Daten:

- ✓ Wieder **aufladbare Akkus**
- ✓ **Bluetooth**
- ✓ Transparentes Gehäuse



## Sensoren:

- ✓ Ultraschall **Distanzsensor** (Frontseite)
- ✓ 4 optische Näherungssensoren (in den Ecken)
- ✓ **2 optische Sensoren** zur Linienverfolgung
- ✓ **Mikrofon**
- ✓ **Helligkeitssensor**

## Aktoren:

- ✓ **8 RGB LEDs**
- ✓ **2 weiße LEDs** Frontlichter
- ✓ **Lautsprecher**
- ✓ **2 Motoren** mit Wegmessung
- ✓ Elektromagnetischer **Stifthalter** mit Vorrichtung zum Heben und Senken des Stiftes



**Stifte zum Einsetzen in den InO-Bot**  
Die Stifte müssen einen **Durchmesser von 10mm** besitzen

## Programmierung

- ✓ mit **Scratch** auf Geräten mit **Windows** - Betriebssystem oder über APP für **Andoid** und **iOS** (Tablet, Handy) programmierbar




## Arbeiten mit mehreren InO-Bots im Klassenverband

Werden mehrere **InO-Bots** im Klassenverband verwendet, muss jeder Roboter mit einem eigenen Namen versehen werden. Die Umbenennung des InO-Bots erfolgt über die **InO-Bot APP**.



1. **InO-Bot** über das **Bluetooth-Symbol** verbinden



2. Stift-Symbol antippen  und den gewünschten Namen eintragen!

InO-BotMIL

3. Den Anweisungen auf dem Bildschirm folgen!

2

Rename



InO-BotMIL

Cancel

OK

3

Hold Up A Minute!



You will need to turn the InO-Bot off for 10 seconds and then on again. When you have done this press OK.

OK



Bitte unbedingt genau die Reihenfolge der Anweisungen befolgen! Die APP unbedingt im **Querformat** verwenden, da sonst einige Elemente der Oberfläche unsichtbar sind.

**Achtung:** Bei allen Geräten (Notebooks, Handys, Tablets), die nicht für die Programmierung mit einem Ino-Bot verwendet werden, die Bluetooth-Verbindung deaktivieren!

## Programmierung unter Verwendung der InO-Bot APP



Die InO-Bot APP kann aus den jeweiligen Stores heruntergeladen werden:

**Achtung:** Die Sprache der Beschriftung der einzelnen Befehle ist bunt durcheinander gemischt auf Deutsch, Englisch und Französisch

- ✓ Handys und Tablets mit **Android** Betriebssystem:  Google play-Store
- ✓ iPhone und iPad (ab **iOS10**): 
- ✓ Um die InO-Bot APP unter Windows ausführen zu können, ist zusätzlich ein ANDROID-Emulator wie **BlueStacks** oder **NOX** erforderlich.

### APP Programmieroberfläche

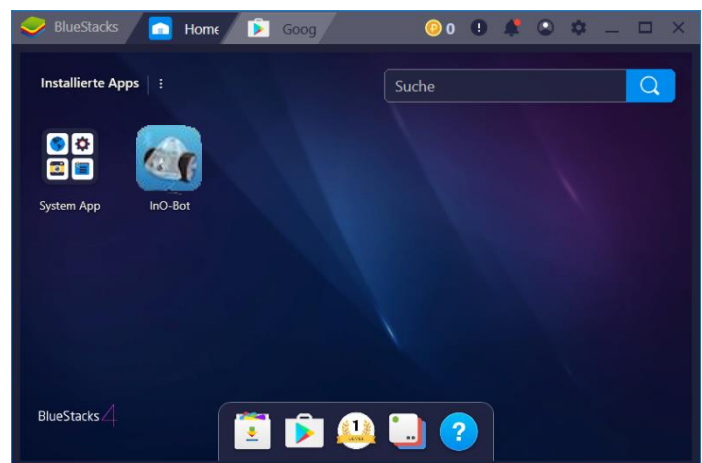
The screenshot shows the InO-Bot APP interface with the following callouts:

- Programm speichern**: Points to the save icon in the top bar.
- Programm öffnen**: Points to the folder icon in the top bar.
- Schritt zurück**: Points to the UNDO button.
- Schritt vor**: Points to the REDO button.
- Workspace löschen**: Points to the CLEAR button.
- Quellcode anzeigen**: Points to the VIEW CODE button.
- Bluetooth-Verbindung mit InO-Bot herstellen!**: Points to the Bluetooth icon in the top bar.
- Register mit den Blöcken für die Programmierung**: Points to the 'set LED' and 'set all LEDs' blocks.
- Zum Löschen einzelner Blöcke werden nach links hinaus gezogen**: Points to the left edge of a block.
- Workspace (Arbeitsfläche)**: Points to the main workspace area.
- Vergrößern bzw. Verkleinern der Blöcke auf der Arbeitsfläche**: Points to the zoom in (+) and zoom out (-) icons.
- Zentrieren der Blöcke auf der Arbeitsfläche**: Points to the center icon (=).
- Programm an InO-Bot senden und ausführen!**: Points to the GO button.
- Aktoren**: Points to the status bar at the bottom.
- Batterie Ladezustand**: Points to the BATTERY 3.9V indicator.
- Anzeige der Sensorenwerte für Helligkeit, Lautstärke, Entfernung, Infrarot**: Points to the LIGHT 17, SOUND 3, DISTANCE 0cm, and IR 253 indicators.



## Installation und Ausführen der InO-Bot Blockly APP unter Windows

- ✓ Download von einem **Android Software Emulator** für **Windows**  
Dieses Emulationsprogramm ermöglicht das Ausführen von Android-APPs auf einem Windows-PC oder Notebook  
**BlueStacks** <https://bit.ly/2CqBLGW> und  
**NOX** <https://noxofficial.com/nox-for-pc/>  
sind Android-Emulatoren für Windows PCs zum Ausführen der **InO-Bot APP**
- ✓ Software Emulator am Windows PC installieren
- ✓ Installierten Emulator ausführen und über den Google-Play Store die **InO-Bot APP** suchen und installieren!

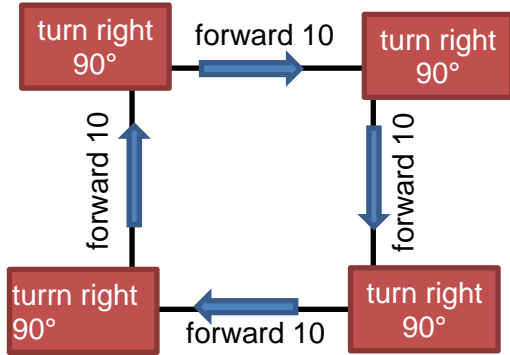




# Erste Programme mit der InO-Bot APP



Bringe deinen Ino-Bot dazu, ein **Quadrat mit einer Seitenlänge von 10 cm** zu zeichnen!



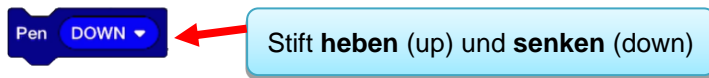
Trage hier die Befehle in der richtigen Reihenfolge ein:

1. Wenn angeklickt wird
2. Pen down
3. Forward 10
4. Turn right 90°
5. Forward 10
6. Turn right 90°
7. Forward 10
8. Turn right 90°
9. Forward 10
10. Turn right 90°
11. Pen up

Was fällt dir auf, wenn du den Programmablauf betrachtest?



Der **InO-Bot** besitzt eine integrierte **Stifthalterung**. Ein eingesetzter Stift kann über folgenden Block gesteuert werden:



**TIPP:** Denke daran, dass du den **Stift**, vor der ersten Bewegung des Roboters **senken** (pen down) und am Schluss wieder **heben** (pen up) musst!


Folgende Befehle aus dem Register „InO-Bot Facile“  werden auch noch benötigt:

## Moves (Bewegungen) & Turns (Drehungen):

**Richtung der Bewegung**  
forward (vorwärts),  
reverse (rückwärts)

**Geschwindigkeit:** lent (langsam), moyen (mittel), vite (schnell)

Zusätzlich ist ein **Ereignis** erforderlich, durch welches das Programm gestartet wird. Wir verwenden dazu das Ereignis

Wenn  angeklickt wird

**Strecke in cm**

forward **lent** for 10 cm


**Richtung der Drehung**  
left (links), right (rechts)

**Winkel in Grad**

spin left **vite** by 90 degrees

**Geschwindigkeit:** lent (langsam), moyen (mittel), vite (schnell)

```

Wenn  angeklickt wird
Pen DOWN
forward lent for 10 cm
spin left vite by 90 degrees
forward lent for 10 cm
spin left vite by 90 degrees
forward lent for 10 cm
spin left vite by 90 degrees
forward lent for 10 cm
spin left vite by 90 degrees
Pen UP
    
```

Was fällt dir auf, wenn du den Programmablauf betrachtest?  
**Die Befehle „Forward“ und „Turn right“ werden jeweils 4x wiederholt!**

## Kontrollstrukturen: Schleife (ITERATION):

Eine Schleife (**Iteration**) ist bei der Programmierung eine Kontrollstruktur, bei der eine Anweisung oder ein Anweisungsblock solange wiederholt wird, bis eine Abbruchbedingung erfüllt wird. Die **Schleife** erspart das mehrfache Notieren von Befehlen.



Man unterscheidet:

- **Kopfgesteuerte Schleifen**

Die Überprüfung der Abbruchbedingung erfolgt vor dem Durchlauf der Schleife, das kann dazu führen, dass die Anweisungen innerhalb der Schleife überhaupt nicht ausgeführt werden!

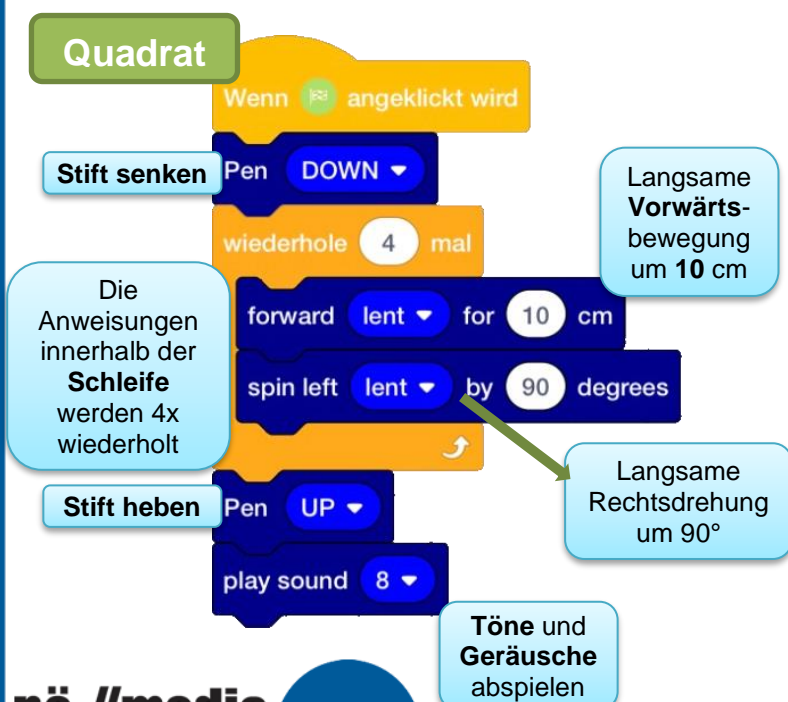
- **Fußgesteuerte Schleifen**

Die Abfrage der Abbruchbedingung erfolgt am Ende der Schleife, diese Schleife wird dadurch zumindest einmal durchlaufen!

Die Blöcke zum Programmieren einer Schleife findest du im Register „**Steuerung**“



## InO-Bot AB2a Lösung



### Quelltext

```
//When play button pressed
pen("DOWN");
var xye = 4;
for(var upp = 0; upp < xye; upp++)
{
  forwardForDistance("02", "10");
  spinLeftByDegrees("02", "90");
}
pen("UP");
playSound("8");
```

InO-Bot AB2b Lösung

Gleichseitiges Dreieck

```

Wenn angeklickt wird
  Pen DOWN
  wiederhole 3 mal
    forward lent for 10 cm
    spin left lent by 120 degrees
  Pen UP
    
```

Regelmäßiges Sechseck

```

Wenn angeklickt wird
  Pen DOWN
  wiederhole 6 mal
    forward lent for 10 cm
    spin left lent by 60 degrees
  Pen UP
    
```

Rechteck

```

Wenn angeklickt wird
  Pen DOWN
  wiederhole 2 mal
    forward moyen for 7 cm
    spin right vite by 90 degrees
    forward moyen for 4 cm
    spin right vite by 90 degrees
  Pen UP
    
```

## Ausgabe von Tönen und Geräuschen (Sounds):

Der Befehl „play“ für die Ausgabe von Tönen und Geräuschen befindet sich im Register

„InO-Bot Facile“ 

Es gibt insgesamt **29** verschiedene voreingestellte Möglichkeiten, die über den folgenden Block zugewiesen werden!

### 29 verschiedene Sounds

- ✓ 1 – 9: verschiedene **Klangeffekte**
- ✓ 10 – 16: **Klavier** (Tonleiter)
- ✓ 17 – 29: **Xylophon** (Tonleiter)

Zahl zwischen 0 und 29



## InO-Bot AB3 Lösung

Schleife mit 10 Wiederholungen



Ausgabe der **Sounds**: Die Funktion „play sound“ bekommt durch den Operator der Reihe nach 10 zufällig ausgewählte Zahlen zwischen 1 und 29 zugewiesen!

Nach jedem Sound wird 1 Sekunde gewartet

Quellcode

```
var ton, test;  
  
//When play button pressed  
var yid = 10;  
for(var itz = 0; itz < yid; itz++)  
{  
  playSound("randomNumber(1, 29)");  
}  
waitForSecond(1);
```



## Steuerung der Front-LEDs



Der InO-Bot besitzt **2 Front-LEDs** (weiß), die unabhängig voneinander gesteuert werden können!

- ✓ Helligkeitswerte für die Front-LEDs 0-10

Helligkeitswerte 0-10

```
white LED Droite to 0
```

<input checked="" type="checkbox"/> <b>Droite</b>	rechts
<input type="checkbox"/> <b>Gauche</b>	links
<input type="checkbox"/> <b>Les deux</b>	beide

Endlos-Schleife

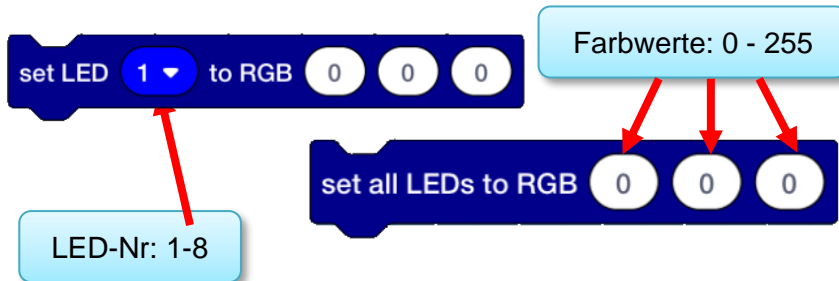
```

Wenn angeklickt wird
wiederhole fortlaufend
  white LED Droite to 5
  warte 1 Sekunden
  white LED Droite to 0
  warte 1 Sekunden
  white LED Gauche to 5
  warte 1 Sekunden
  white LED Gauche to 0
  warte 1 Sekunden
  white LED Les deux to 10
  warte 3 Sekunden
  white LED Les deux to 0
  warte 1 Sekunden
  play sound 1
  
```

## RGB LEDs – Little light show

Zusätzlich zu den beiden weißen Front-LEDs besitzt der InO-Bot insgesamt 8 einzeln steuerbare RGB-LEDs. Die Farben der LEDs können sowohl Farbbezeichnungen (nur einige Grundfarben → Rot, Gelb, Grün, Blau, Weiß) und über **Farbwerte** 0 – 255 gesteuert werden.

Diese LEDs können sowohl einzeln (1-8) als auch alle zusammen angesteuert werden!



set all LEDs to Rouge ▾

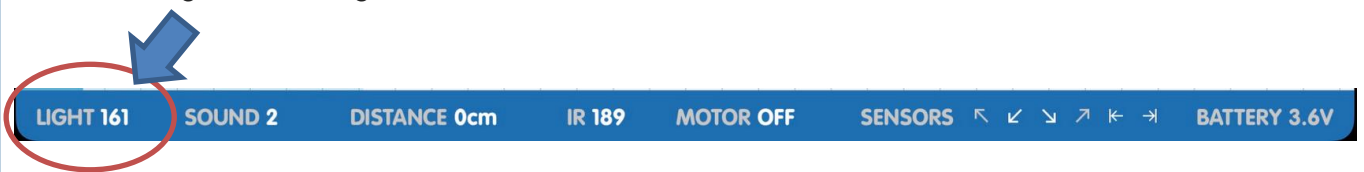
- ✓ Rouge
- Jaune
- Vert
- Bleu
- Blanc
- Off

InO-Bot AB4b Lösung

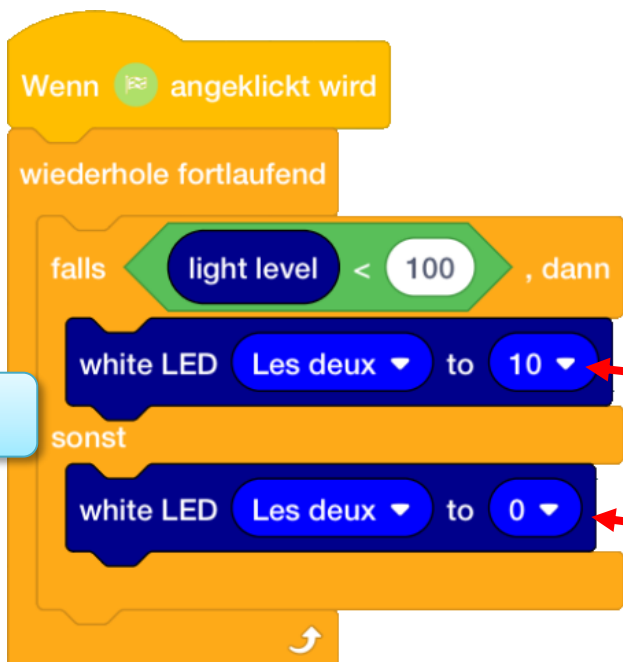
```
Wenn angeklickt wird
  set all LEDs to Rouge
  warte 2 Sekunden
  set all LEDs to Jaune
  warte 2 Sekunden
  set all LEDs to Vert
  warte 2 Sekunden
  set all LEDs to Bleu
  warte 2 Sekunden
  set all LEDs to Blanc
  warte 2 Sekunden
  set all LEDs to RGB 0 255 255
  warte 2 Sekunden
  set all LEDs to RGB 255 0 255
  warte 2 Sekunden
  set all LEDs to RGB 0 0 0
```

## SENSOREN: Lichtsensor

Der InO-Bot besitzt einen **Lichtsensor**, mit dem die LEDs abhängig von der Helligkeit ein- oder ausgeschaltet werden können! In der Statusleiste der Programmieroberfläche kann jederzeit der aktuelle Helligkeitswert abgelesen werden.



### InO-Bot AB5 Lösung

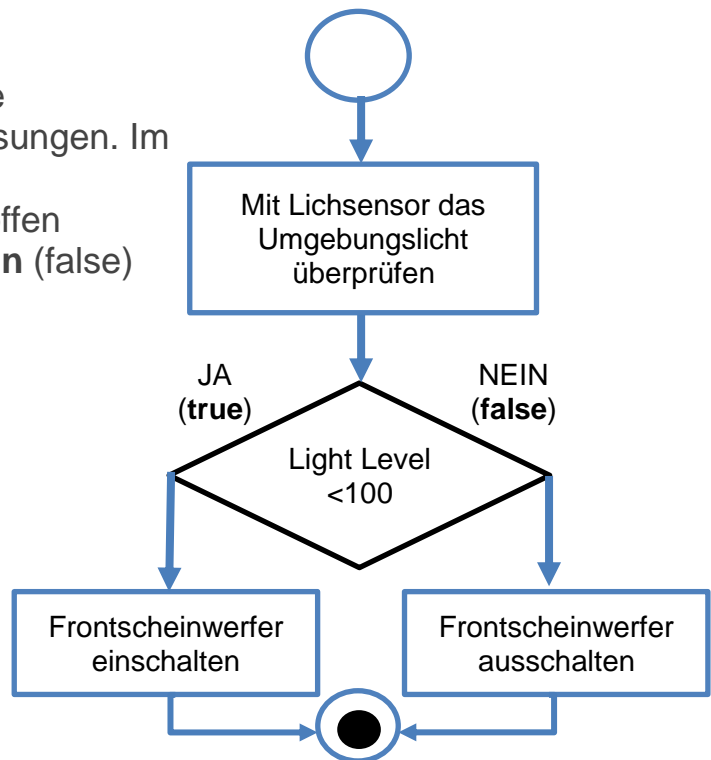


„IF then Else“ - Verzweigung

Wird ausgeführt, wenn die Bedingung erfüllt wird → Wert ist „true“

Wird ausgeführt, wenn die Bedingung nicht erfüllt wird → Wert ist „false“

Bislang waren unsere Programme eine Hintereinander-Ausführung von Anweisungen. Im obigen Programm muss aber je nach Lichtintensität eine Entscheidung getroffen werden, die man mit **Ja** (true) oder **Nein** (false) beantwortet werden kann.



## SENSOREN: Distanzsensor



Mit Hilfe seiner beiden Ultraschall **Distanzsensoren** kann dein InO-Bot erkennen, ob sich ein Hindernis vor ihm befindet und damit eine Kollision droht. Die Entfernung wird in cm (0 – 255 cm) angegeben.

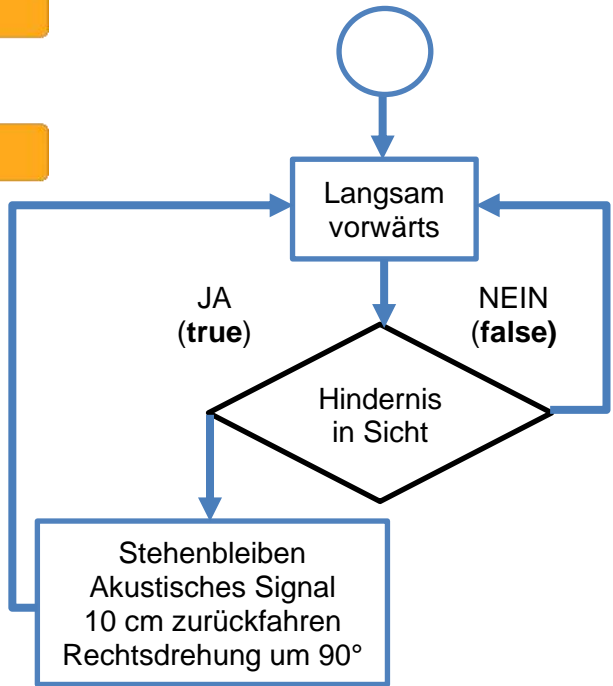
InO-Bot AB6 Lösung

```

    Wenn  angeklickt wird
    wiederhole fortlaufend
    falls  < 30 , dann
    stop motors
    play sound 4
    reverse  for 10 cm
    spin right  by 90 degrees
    sonst
    forward 
    
```

Endlos-Schleife

In einer **Endlosschleife** ist die zugehörige Bedingung immer erfüllt, es gibt kein Abbruchkriterium. Bekannt sind Endlosschleifen in erster Linie als unerwünschte Programmierfehler. Jedoch gibt es auch Szenarien (siehe Programm), in denen man durchaus ganz bewusst keinen Abbruch möchte.



Weitere interessante Anwendungsbeispiele ergeben sich aus dem Einsatz der zwei Sensoren auf der Unterseite des InO-Bot. Der Roboter kann damit so programmiert werden, dass er einer schwarzen Linie folgt


Quellcode



## Programmierung mit Scratch

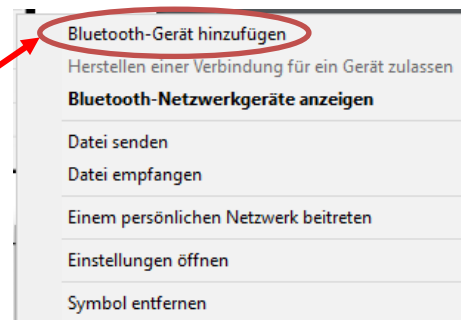
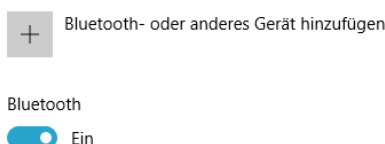
Folgende Dinge werden benötigt:

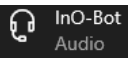
- ✓ PC oder Tablet mit Windows7, 8.1 oder **Windows10**
- ✓ **Bluetoothverbindung**
- ✓ Offline Version von **Scratch** → <https://scratch.mit.edu/download>  
**Achtung:** Unbedingt **Scratch 2.0** verwenden, Version 3.x und neuer verursacht Probleme!!!

- ✓  **Scratch Launcher** von  → <https://bit.ly/2P70wuf>

### Herstellung der Bluetoothverbindung mit Windows10

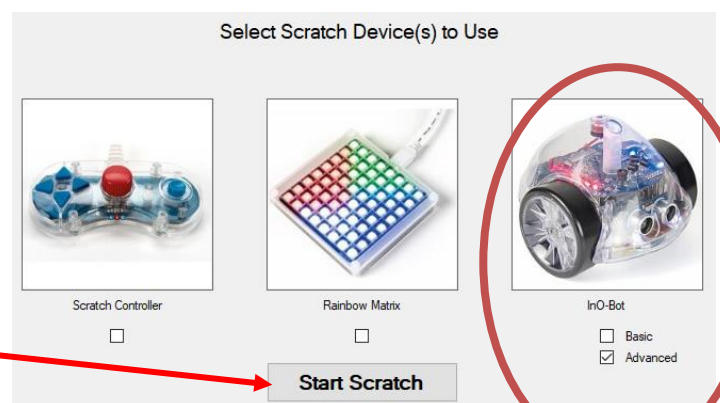
- ✓ InO-Bot einschalten
- ✓ Bluetooth am PC aktivieren und anschließend im System Tray auf das Bluetooth Icon klicken
- ✓ „**Bluetooth-Gerät hinzufügen**“ auswählen
- ✓ Bluetooth- und andere Geräte



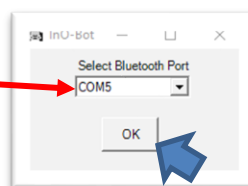
- ✓ Sobald der InO-Bot gefunden wurde, Headphone-Symbol  anklicken, um den Roboter mit dem PC zu verbinden

### Programmierung starten

- ✓ **InO-Bot** einschalten
- ✓ **Scratch Launcher** starten
- ✓ **InO-Bot** auswählen
- ✓ Gewünschten **Modus (Basic oder Advanced)** auswählen
- ✓ **Start Scratch**-Schaltfläche anklicken



- ✓ **Bluetooth-Port** auswählen (Voreinstellung beibehalten!)
- ✓ **Scratch Programmierumgebung** wird aufgerufen



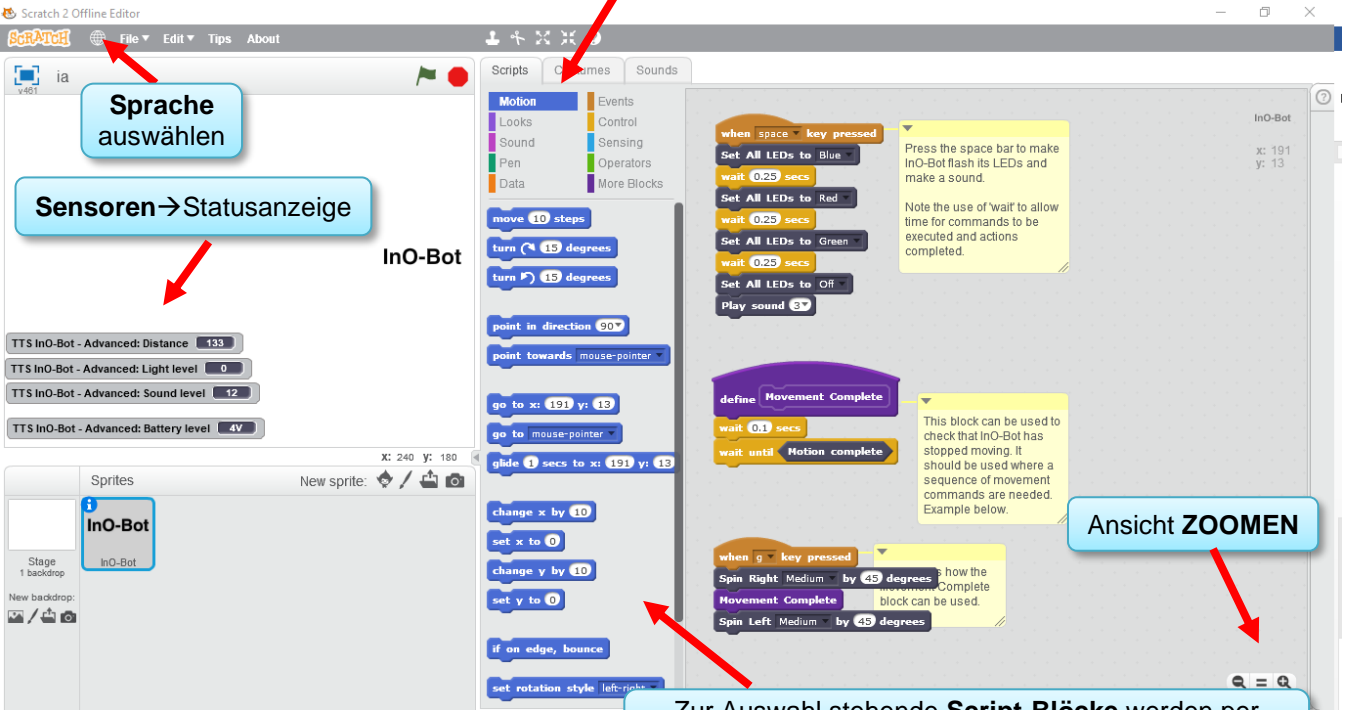
# Scratch-Oberfläche

Zur Auswahl stehende **Script-Bereiche: Sound, Pen, Control ...**

Sprache auswählen

Sensoren → Statusanzeige

InO-Bot



Zur Auswahl stehende **Script-Blöcke** werden per **DRAG&DROP** in den Programmierbereich gezogen!

Ansicht **ZOOMEN**

## Programmbeispiele

```
when space key pressed
  Set All LEDs to Blue
  wait 1 secs
  Set All LEDs to Red
  wait 1 secs
  Set All LEDs to Green
  wait 1 secs
  Set All LEDs to Off
  Play sound 3
```

InO-Bot AB7 Lösung

Weitere Aufgaben findest du in den

# ACTIONCards for InO-BOT

Einführung  
in die Turtle-Grafik mit  
Scratch



Weitere Informationen und Materialien:

<http://hemi.bplaced.net/Robotik/Roboter.htm>