Finde-Waldo : Robotersteuerung

# Einleitung

## Projekt

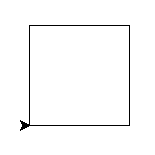
Finde-Waldo ist ein Projekt für Hackanapp.   
Dabei geht es um Teamarbeit und Kommunikation zwischen Mitgliedern einer Gruppe.

Das Ziel des Workshops ist es, eine Person in einem Gruppenfoto zu finden.   
Das sollte wie folgt ablaufen:

1. Eine Person der Gruppe setzt sich vor eine Webcam oder ein Smartphone und ein Foto wird gemacht.
2. Das Foto wird zusammen mit einem Foto der ganzen Gruppe an eine künstliche Intelligenz gesendet, welche die Person auf dem Gruppenfoto findet.
3. Ein Roboter fährt auf einem ausgedruckten Gruppenfoto zu der Position der gesuchten Person und zeigt auf sie.

Dieser Teil der Dokumentation befasst sich mit der Steuerung des Roboters.   
Die Gruppe soll mithilfe von Python den Roboter steuern.

## Beispiel



**Robotersteuerung**

Die Steuerung des Roboters funktioniert bereits mit wenigen Befehlen.  
Hier ist ein Beispiel eines einfachen Skripts, das den Roboter anweist, ein Viereck zu fahren:



Es steht auch ein Simulator für den Roboter zur Verfügung, der Turtle benutzt.   
Darin sieht das Resultat wie folgt aus:

|  |
| --- |
|  |

Der Simulator kann eingeschaltet werden indem man in robot.py Linie 1 kommentiert, und Linie 2 unkommentiert.

# Inhalt

[Einleitung 1](#_Toc13775201)

[Projekt 1](#_Toc13775202)

[Beispiel 1](#_Toc13775203)

[Inhalt 2](#_Toc13775204)

[Motoren 3](#_Toc13775205)

[Räder 3](#_Toc13775206)

[Linker\_Motor(Grad) 3](#_Toc13775207)

[Rechter\_Motor(Grad) 3](#_Toc13775208)

[Beide\_Motoren(Grad) 3](#_Toc13775209)

[Bewegen 3](#_Toc13775210)

[Vorwärts(Cm) 3](#_Toc13775211)

[Rückwärts(Cm) 3](#_Toc13775212)

[Rechts(Grad) 3](#_Toc13775213)

[Links(Grad) 3](#_Toc13775214)

[Zeiger 3](#_Toc13775215)

[Zeigen() 3](#_Toc13775216)

[Nicht\_Zeigen() 3](#_Toc13775217)

[Sprechen 4](#_Toc13775218)

[Text 4](#_Toc13775219)

[Sprechen(Text) 4](#_Toc13775220)

[Beep 4](#_Toc13775221)

[Beep() 4](#_Toc13775222)

[Berechnen 4](#_Toc13775223)

[Hypotenuse(Länge, Höhe) 4](#_Toc13775224)

[Hypotenuse2(Koordinaten) 4](#_Toc13775225)

[Konfigurieren 4](#_Toc13775226)

[ROBOT\_ADDRESS 4](#_Toc13775227)

[WHEEL\_RADIUS 4](#_Toc13775228)

[WHEEL\_DISTANCE 4](#_Toc13775229)

[ROBOT\_ARM\_SIZE 4](#_Toc13775230)

[MOTOR\_BASE\_SPEED 5](#_Toc13775231)

[MOTOR\_BASE\_RAMP\_UP 5](#_Toc13775232)

[MOTOR\_BASE\_RAMP\_DOWN 5](#_Toc13775233)

[ENABLE\_SOUND 5](#_Toc13775234)

[IDLE 5](#_Toc13775235)

# Motoren

## Räder

### Linker\_Motor(Grad)

|  |  |
| --- | --- |
| Parameter: | Grad – Anzahl Grad mit der sich das linke Rad drehen soll. |

Dreht das linke Rad um die angegebene Anzahl Grad.

### Rechter\_Motor(Grad)

|  |  |
| --- | --- |
| Parameter: | Grad – Anzahl Grad mit der sich das rechte Rad drehen soll. |

Dreht das rechte Rad um die angegebene Anzahl Grad.

### Beide\_Motoren(Grad)

|  |  |
| --- | --- |
| Parameter: | Grad – Anzahl Grad mit der sich beide Räder drehen sollen. |

Dreht beide Räder um die angegebene Anzahl Grad.

## Bewegen

### Vorwärts(Cm)

|  |  |
| --- | --- |
| Parameter: | Cm – Anzahl Cm mit der sich der Roboter nach vorne bewegen soll. |

Bewegt den Roboter um die angegebene Anzahl Cm nach vorne.

### Rückwärts(Cm)

|  |  |
| --- | --- |
| Parameter: | Cm – Anzahl Cm mit der sich der Roboter nach hinten bewegen soll. |

Bewegt den Roboter um die angegebene Anzahl Cm nach hinten.

### Rechts(Grad)

|  |  |
| --- | --- |
| Parameter: | Grad – Anzahl Grad um die sich der Roboter nach rechts drehen soll. |

Dreht den Roboter um die angegebene Anzahl Grad um sich selbst nach rechts.

### Links(Grad)

|  |  |
| --- | --- |
| Parameter: | Grad – Anzahl Grad um die sich der Roboter nach links drehen soll. |

Dreht den Roboter um die angegebene Anzahl Grad um sich selbst nach links.

## Zeiger

### Zeigen()

Dreht den Zeiger des Roboters auf 90 Grad so dass dieser zeigt.

### Nicht\_Zeigen()

Dreht den Zeiger des Roboters auf 0 Grad so dass dieser nicht zeigt.

# Sprechen

## Text

### Sprechen(Text)

|  |  |
| --- | --- |
| Parameter: | Text – Text der gesprochen werden soll. |

Der Roboter sagt den angegebenen Text.

## Beep

### Beep()

Der Roboter gibt ein Piepsen von sich.

# Berechnen

### Hypotenuse(Länge, Höhe)

|  |  |
| --- | --- |
| Parameter: | Länge – X-Achse der Strecke . |
| Höhe – Y-Achse der Strecke. |
| Rückgabewert: | Strecke – Strecke bis zum Ziel in cm |
| Winkel – Winkel für die Ausrichtung aufs Ziel |

Berechnet die Hypotenuse und den Winkel um zu der gewünschten Position (Länge, Höhe) zu gelangen. Beispiel:

### Hypotenuse2(Koordinaten)

|  |  |
| --- | --- |
| Parameter: | Koordinaten – Koordinaten einer Person als Box [x\_Start, y\_Start, x\_Ende, y\_Ende]. |
| Rückgabewert: | Strecke – Strecke bis zum Ziel in cm |
| Winkel – Winkel für die Ausrichtung aufs Ziel |

Berechnet die Hypotenuse und den Winkel, um zu der gewünschten Position zu gelangen. Die Koordinaten erhält man von der Finde\_Person() Funktion im image-Modul. Beispiel:



# Konfigurieren

Um zu funktionieren, benutzt das robot-module verschiedene Variablen, welche es aus der Konfigurationsdatei ausliest. Die Datei befindet sich im modules-Ordner und heisst «robot\_conf.py».  
Dort kann man verschiedene Werte anpassen, die das Fahr-verhalten des Roboters beeinflussen.  
Variablen, die hier nicht aufgelistet sind, sollten am besten ignoriert werden. Variablen, welche mit MOTOR anfangen sind für die richtige Benutzung nicht essenziell und sollten wenn möglich auf ihrem Standartwert belassen werden.

### ROBOT\_ADDRESS

Die IP Adresse des Roboters. Sie wird oben links auf seinem Bildschirm angezeigt, wenn dieser mit dem WLAN verbunden ist.

### WHEEL\_RADIUS

Der Radius eines Rads des Roboters in mm.

### WHEEL\_DISTANCE

Der Abstand zwischen den beiden Rädern des Roboters in mm.

### ROBOT\_ARM\_SIZE

Die Länge des Zeigers des Roboters, gemessen von seinem Mittelpunkt aus in mm.

### MOTOR\_BASE\_SPEED

Die maximale Geschwindigkeit des Roboters.

### MOTOR\_BASE\_RAMP\_UP

Die Zeit in welcher der Roboter zur maximalen Geschwindigkeit beschleunigt in ms.

### MOTOR\_BASE\_RAMP\_DOWN

Die Zeit in welcher der Roboter vollständig abbremst in ms.

### ENABLE\_SOUND

Ob der Roboter Töne von sich gibt, wenn Sprechen() und Beep() aufgerufen werden oder nicht als Boolean.

# IDLE

Alle Methoden, welche aufgelistet wurden, können auch direkt in einer Python-Shell wie IDLE verwendet werden können. Zum Beispiel kann in einem Terminal folgendes eingegeben werden:



Um den Roboter 10 cm nach vorne zu bewegen, ohne dafür ein Python Skript zu erstellen.