

# Klassen - Erklärung

Dateiname	Klassenname	Bezeichner	Funktionen	Erklärung
analog.py	Analog	Helligkeit_V, Helligkeit_H, Distanz	<code>current_value()</code>	Allgemeine Klasse für alle analogen Sensoren
digital.py	Digital	Knopf_VR, Knopf_VL, Knopf_HR, Knopf_HL	<code>is_pressed()</code> , <code>current_value()</code>	Klasse für alle digitalen Sensoren
distance_sensor.py	DistanceSensor	Distanz	<code>current_value()</code>	Klasse für den Distanz Sensor
driveR.py	driveR_two	Fahrzeug	<code>drive_straight()</code> , <code>turn_degrees()</code> , <code>turn_wheel()</code>	Klasse zum Fahren
light_sensor.py	LightSensor	Helligkeit_V, Helligkeit_H	<code>current_value()</code> , <code>get_value_*</code> (), <code>sees_*</code> ()	Klasse für den Licht- und Helligkeitssensor
servo.py	ServoX	Servo_Arm, Servo_Hand	<code>set_pos()</code> , <code>range_to_pos()</code> , <code>range_from_to_pos()</code>	Klasse für die Servo

```
current_value() -> int
```

Gibt den aktuellen Wert zurück (als Ganzzahl)

```
is_pressed() -> bool
```

Sagt, ob der Knopf gedrückt ist ( `True` ) oder nicht ( `False` )

```
drive_straight(millis: int, speed: int = None) -> None
```

Bringt den Roboter dazu, geradeaus zu fahren

```
turn_degrees(direction: str, degree: int) -> None
```

Bringt den Roboter dazu, sich in einem bestimmten Winkel in eine bestimmte Richtung zu drehen

```
turn_wheel(direction: str, millis: int, speed: int = None) -> None
```

Bringt den Roboter dazu, sich in eine Richtung für eine bestimmte Zeit zu drehen, aber nur mit einem Rad

---

```
get_value_black() -> int  
get_value_white() -> int
```

Sagt, ab welchem ganzzahligen Wert der Roboter Schwarz oder Weiß sieht

---

```
sees_black() -> bool  
sees_white() -> bool
```

Sagt, ob der Roboter gerade Schwarz oder Weiß sieht

---

```
set_pos(value: int) -> None
```

Setzt die Position vom Servo zum vorgegebenen Wert

---

```
range_to_pos(value: int, multi: int = 2) -> None
```

Geht langsam vom aktuellen Servo Wert zu dem vorgegebenen Wert (zum flüssigen Auf- oder Zumachen)

---

```
range_from_to_pos(interval: list, multi: int = 2) -> None
```

Geht langsam vom ersten angegebenen Wert zum zweiten angegebenen Wert (zum flüssigen Auf- oder Zumachen)

---