

# Startdatei für Crawly

Um den Bewegungsablauf für Crawly anders als in der Bibliothek zu machen, können auch hier die entsprechenden Klassen überschrieben werden.

```
from time import sleep, sleep_ms
from robotlibrary.crawly import Crawly
from robotlibrary.crawly_leg import Leg
from robotlibrary.crawly_joint import Joint
from robotlibrary import config_crawly as conf
from robotlibrary.servo import Servo
##### Your class definition
class MyCrawly(Crawly):
    def __init__(self):
        self.legs = {
            "front_right" : MyLeg(4, True, True, "front right"),
            "rear_right" : MyLeg(6, True, False, "rear right"),
            "rear_left" : MyLeg(0, False, False, "rear left"),
            "front_left" : MyLeg(2, False, True, "front left")
        }

    def galumph(self):
        for l in self.legs.values():
            l.leg_fully_up()
        sleep_ms(150)
        for l in self.legs.values():
            l.leg_fully_forward()
        sleep_ms(150)
        for l in self.legs.values():
            l.leg_fully_down()
        sleep_ms(150)
        for l in self.legs.values():
            l.leg_fully_backward()
        sleep_ms(150)

class MyLeg(Leg):
    def __init__(self, pin, right, front, name):
        if right and front:
```

```

        self.shoulder = MyJoint(conf.SHOULDER_FRONT, name, False, False, pin)
if right and not front:
        self.shoulder = MyJoint(conf.SHOULDER_REAR, name, False, False, pin)
if not right and front:
        self.shoulder = MyJoint(conf.SHOULDER_FRONT, name, True, False, pin)
if not right and not front:
        self.shoulder = MyJoint(conf.SHOULDER_REAR, name, True, False, pin)
self.knee = MyJoint(conf.KNEE, name, False, True, pin+1)

```

```
class MyJoint(Joint):
```

```
    def __init__(self, j_type, name, left_side, inverted, pin):
```

```
        self.name = name
```

```
        self.j_type = j_type
```

```
        min_duty = conf.SERVO_MIN_DUTY
```

```
        max_duty = conf.SERVO_MAX_DUTY
```

```
        self.left_side = left_side
```

```
        if j_type == conf.SHOULDER_FRONT:
```

```
            self.__min_angle = conf.SHOULDER_FRONT_MIN_ANGLE
```

```
            self.__max_angle = conf.SHOULDER_FRONT_MAX_ANGLE
```

```
        elif j_type == conf.SHOULDER_REAR:
```

```
            self.__min_angle = conf.SHOULDER_REAR_MIN_ANGLE
```

```
            self.__max_angle = conf.SHOULDER_REAR_MAX_ANGLE
```

```
        elif j_type == conf.KNEE:
```

```
            self.__min_angle = conf.KNEE_MIN_ANGLE
```

```
            self.__max_angle = conf.KNEE_MAX_ANGLE
```

```
            # min_duty = conf.SERVO_MIN_DUTY_TYPE2 # Comment out if the duty cycle is
different from the shoulder servo's duty cycle.
```

```
            # max_duty = conf.SERVO_MAX_DUTY_TYPE2 # Comment out if the duty cycle is
different from the shoulder servo's duty cycle.
```

```
        self.servo = Servo(pin, inverted, min_duty, max_duty)
```

```
##### End of class definition
```

```
def move_program():
```

```
    crawly.move_to_start_pos()
```

```
    for i in range(10):
```

```
        crawly.galumph()
```

```
def main():
```

```
move_program()
```

```
##### Initialize the robot and start the program.
```

```
crawly = MyCrawly()
```

```
if __name__ == "__main__":
```

```
    # execute only if run as a script
```

```
    try:
```

```
        main()
```

```
    except KeyboardInterrupt:
```

```
        print("The robot was stopped by the user.")
```

```
    finally:
```

```
        crawly.park()
```

Hier wurden schon alle drei Hauptklassen des Crawly-Roboters überschrieben. Die anderen Methoden in den Klassen können bei Bedarf auch überschrieben werden.

---

Revision #1

Created 28 April 2026 07:11:55 by Marcus Jacobs

Updated 28 April 2026 07:14:52 by Marcus Jacobs