

Startdatei für den SMARS-Roboter

Kopiere den Code und speichere die Datei als `main.py` auf dem `Pico`.

```
from time import sleep, sleep_ms
import uasyncio as asyncio
from robotlibrary.robot import Robot
##### Your class definition
class MyRobot(Robot):
    def __init__(self):
        super().__init__(False) # Call the original constructor.
        print("Start MyRobot")

    # With this method defined here, the robot will not drive as the speed is not set in this
    function.

    # This is to illustrate how overwriting works.
    def set_speed(self,x):
        print(f"Child method set_speed. Value: {x}")

##### End of class definition
# Define functions for your program
async def monitor_dist():
    '''This checks the distance from the ultrasonic sensor continually.
    If the given distance is longer than the measured one, react_to_obstacle() will be called.
    ...

    global distance
    while True:
        if robot.get_dist() < distance:
            react_to_obstacle()
            await asyncio.sleep_ms(100)

def react_to_obstacle():
    '''Do whatever you want to do when an obstacle is detected.
    ...

    global distance
    robot.random_spin(300)
```

```
robot.set_forward(True)
robot.set_speed(80)

async def driving_program():
    robot.set_speed(90)
    while True:
        print("Driving program running.")
        await asyncio.sleep_ms(3000)

async def main():
    asyncio.create_task(monitor_dist())
    await driving_program()

##### Initialize the robot and start the program.
robot = MyRobot()
distance = 20 # Define the distance you want to have.
if __name__ == "__main__":
    # execute only if run as a script
    try:
        asyncio.run(main())
    except KeyboardInterrupt:
        print("The robot was stopped by the user.")
    finally:
        robot.emergency_stop()
```

Revision #6

Created 28 September 2025 08:42:37 by Marcus Jacobs

Updated 12 November 2025 09:43:22 by Marcus Jacobs