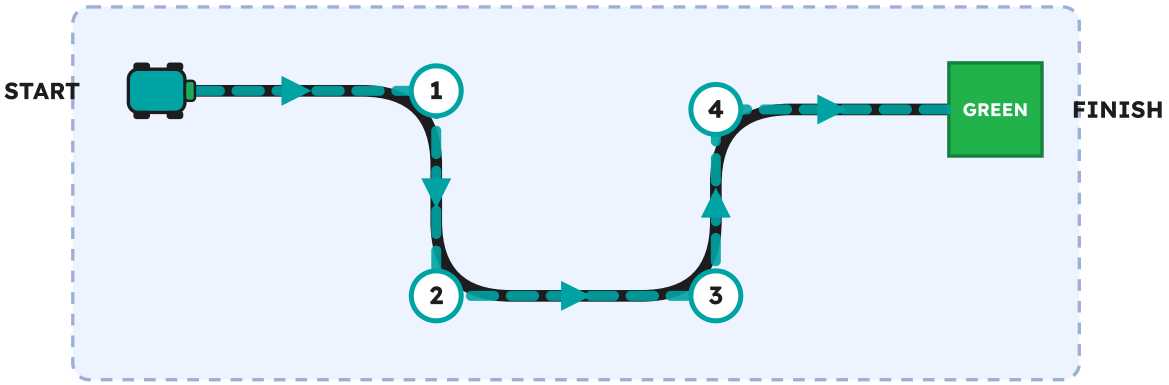
MISSION 6 · SOLUTION

Follow the Line

There is more than one right answer. Yours may look different — that is fine if it works.

THE ROUTE



What the car senses

1. Straight sections: the reading stays near the target, so the steering stays near 0.
2. Bends 1–4: the reading drifts away from the target, so the steering grows and pulls the car back onto the edge of the line.
3. One rule handles every bend — nothing is counted.
4. Green tile under the car → stop and play the sound.

FIRST

Make these My Blocks

celebrate
set hub light to green → play beep 80 for 0.5 s → play beep 100 for 0.5 s · Sound + light on the tile.

THE PROGRAM

Word Blocks

when program starts

set movement motors to A + E
use the ports from YOUR car

set movement speed to 25 %
slower = smoother

start moving forward

repeat until color sensor F sees green

    motor D go to position (color sensor F reflected light − 50) × 0.6
    target 50 = halfway between your white and black readings

stop moving

motor D go to position 0

celebrate
My Block: sound + light

– Same thing in Python

```
from hub import port, sound, light_matrix
import runloop, motor, motor_pair, color_sensor, color

DRIVE      = motor_pair.PAIR_1
STEER      = port.D            # large motor, steers the back
BELLY      = port.F            # color sensor pointing down
TARGET_REFLECT = 50          # halfway between white and black readings
K_LINE     = 0.6               # line-following strength

def on_color(c):
    return color_sensor.color(BELLY) == c

async def celebrate():
    light_matrix.show_image(light_matrix.IMAGE_HAPPY)
    await sound.beep(880, 300)
    await sound.beep(1175, 500)

async def follow_line_until_green():
    motor_pair.move(DRIVE, 0, velocity=250)
    while not on_color(color.GREEN):
        error = color_sensor.reflection(BELLY) - TARGET_REFLECT
        angle = max(-60, min(60, int(error * K_LINE)))
        await motor.run_to_absolute_position(STEER, angle, 1000)
    motor_pair.stop(DRIVE)
    await motor.run_to_absolute_position(STEER, 0, 800)

async def main():
    motor_pair.pair(DRIVE, port.A, port.E)      # your drive motor ports
    await follow_line_until_green()
    await celebrate()

runloop.run(main())
```

Starting values only. Steering angle, roll distances, the OPEN limit and the line target depend on *your* car and maze. Measure, then tune one at a time. I have not run these on a car — expect to adjust.

LEVEL UP

Ready for more?

- Add a 10 cm **gap** in the line: the car must drive straight across it.
- Add a second line that branches: turn only where a **green square** tells you to.
- Beat your best time with a higher speed.
