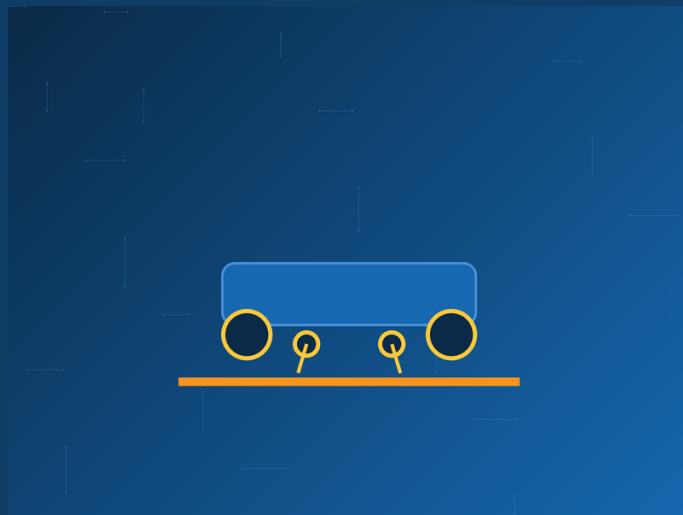


STAGE 4

Line-Following Robot

Teach a robot to follow a line on the floor using two IR sensors and a Raspberry Pi Pico.



Skill level: Intermediate

Recommended age: 12+

Hands-on builds: 4

What you'll build

This manual walks through assembling and programming the **Stage 4: Line-Following Robot** step by step, alongside a full materials list and the original circuit diagram. It's a companion to — not a replacement for — the kit's original build guide, which is linked from the product page for reference.

#	Project	What it teaches
1	IR sensor test	Wire both IR sensors to the Pico's digital inputs and read their on/off state as they pass over light and dark surfaces.
2	Motor driver wiring	Wire all 4 BO motors to the L298N driver so the Pico can drive each side independently.
3	Line-following logic	Write the control loop that steers the robot based on which sensor currently sees the line.
4	Calibrated line follower	Tune the sensor thresholds and motor speed until the robot reliably tracks a line at a steady pace.

Note: The code above is shown exactly as provided, including the point where the original source document is cut off. The complete script (turning logic and the main loop) is included in the original build guide PDF linked from the product page.

What's in the box

11 line items. Check every part against this list before you start building.

Item	Qty	What it's for
Raspberry Pi Pico	1	The microcontroller — runs the MicroPython program that reads the sensors and drives the motors.
L298N motor driver	1	Lets the Pico safely switch the higher-current drive motors.
IR sensor	2	Detects the contrast between a dark line and a light floor by reflecting infrared light.
300 RPM BO motor-straight	4	Geared DC motors that drive the wheels.
65mm robot wheel	4	Larger wheels for smoother, more stable tracking.
Robot chassis	1	The frame every component mounts to.
Jumper wires	40	Wire the sensors and driver board together.
18650 Li-ion 2500mAh battery pack	1	Power source for the motors and electronics.
Breadboard	1	Prototyping surface for the sensor wiring.
Data cable (type B)	1	Connects the Pico to a computer to upload the MicroPython code.
Double-sided foam tape	1	Secures the breadboard and sensors to the chassis.

Step-by-step build instructions

1 Assemble the chassis

Mount all 4 BO motors and 65mm wheels onto the robot chassis.

2 Wire the motor driver

Mount the L298N to the chassis and wire the left-side and right-side motor pairs to its output terminals, as shown in the circuit diagram on page 4.

3 Wire the driver to the Pico

Connect the L298N's enable and direction pins to the Pico's GPIO pins as shown in the diagram.

4 Mount the IR sensors

Fit the two IR sensors at the front of the chassis, close to the ground and angled toward the floor, with a small gap between them so the line can pass between or under them.

5 Wire the sensors

Connect the left and right IR sensors' output pins to two of the Pico's digital input pins (GPIO 2 and 3 in the sample code).

6 Connect power

Wire the battery pack in and confirm the Pico, L298N, and both sensors share a common ground.

7 Upload the code

Flash the MicroPython code on page 6 onto the Pico as main.py.

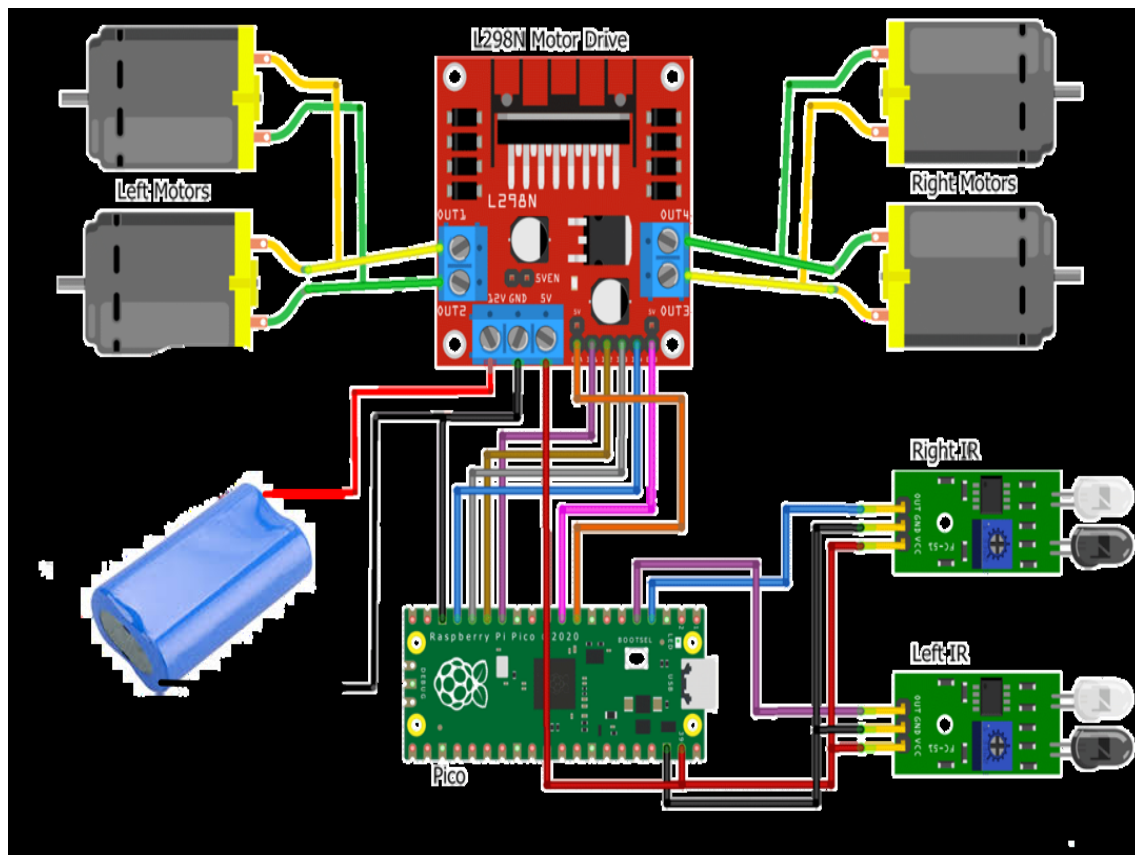
8 First test

Place the robot on a light surface with a dark line and power it on. Watch for both sensors correctly changing state as they cross the line.

9 Calibrate

Adjust the sensor height/angle and the motor speed values in the code until the robot follows curves smoothly without losing the line.

Circuit diagram



MicroPython code

Copy this onto the Raspberry Pi Pico as main.py using Thonny (or your preferred MicroPython IDE), then run it.

```
# importing PIN and PWM
from machine import Pin, PWM
import time
import utime

# Defining motor pins
motor1 = Pin(10, Pin.OUT)
motor2 = Pin(11, Pin.OUT)
motor3 = Pin(12, Pin.OUT)
motor4 = Pin(13, Pin.OUT)

# Defining enable pins and PWM object
enable1 = PWM(Pin(6))
enable2 = PWM(Pin(7))

# Defining right and left IR digital pins as input
right_ir = Pin(2, Pin.IN)
left_ir = Pin(3, Pin.IN)

# Defining frequency for enable pins
enable1.freq(1000)
enable2.freq(1000)

# Setting maximum duty cycle for maximum speed
enable1.duty_u16(65025)
enable2.duty_u16(65025)

# Forward
def move_forward():
    motor1.low()
    motor2.high()
    motor3.high()
    motor4.low()

# Backward
def move_backward():
    motor1.high()
    motor2.low()
    motor3.low()
    motor4.high()

# --- The remainder of the line-following control loop (reading
# right_ir/left_ir and calling move_forward/move_backward plus turning
# functions) is included in full in the original build guide PDF –
# see the "Original spec sheet" link on the product page.
```

Troubleshooting

Symptom	Fix
A motor spins the wrong way	Swap its two wire connections at the driver board or switch.
Nothing powers on	Check battery orientation first, then confirm every switch or connector is fully seated.
Bluetooth won't pair (Stage 3)	Confirm the HC-05's LED is blinking (pairing mode) and try the default PIN, 1234 or 0000.
Pico isn't recognised by your computer	Hold the BOOTSEL button on the Pico while plugging in the USB cable, then release it once connected.
Line follower loses the line on curves (Stage 4)	Lower the sensor height slightly, slow the motor speed, or widen the gap between the two sensors.
Ultrasonic sensor gives erratic readings (Stage 5)	Make sure nothing else is close enough to cause echo interference, and double-check the Trig/Echo pins aren't swapped.

Need more help? Email our STEM instructors or visit the FAQ at streamedutech.com. The original source build guide for this kit (with the manufacturer's own circuit reference) is linked from this kit's product page alongside this manual.