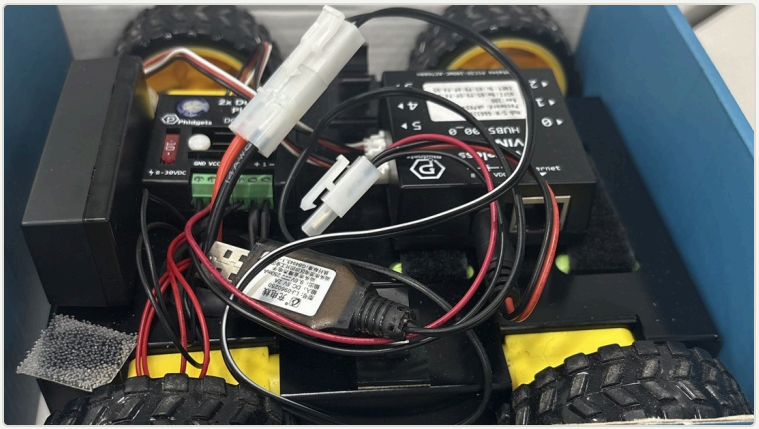
PERSONAL PROJECT · MECHANICAL, ELECTRONICS AND CONTROLS

Autonomous Obstacle-Course Rover

A four-wheel rover I designed in CAD, 3D printed, wired and programmed, then ran through a six-stage obstacle course using timed moves, a sonar sensor and a light sensor.

ROLE	BUILT WITH	RESULT
Solo: design, build, code	**CAD, 3D printing, Phidgets, Java**	**All 6 stages cleared**



The assembled rover: Phidgets motor controller (left), wireless Phidgets hub (right), battery pack, four driven wheels.



The course from above: textbooks stand on edge to form the walls the rover has to get through.

THE BRIEF

The assignment, *Maze Navigator & Environment Logger*, asked for a rover that navigates a maze on its own, uses sonar to avoid obstacles and a light sensor to read its environment, and records sensor readings at each step so the run can be analysed afterwards.

THE SIX-STAGE COURSE

STAGE	CHALLENGE	WHAT THE ROVER DOES
1	Hard-coded turns	Follows a timed sequence of drive and turn moves
2	Sonar obstacle detection	Stops when something is closer than 200 mm and turns
3	Speed recognition	Drives at a different speed for each part of the run
4	Light sensing	Creeps toward the lamp and stops when the light level is high enough
5	Precision turning	Turns calibrated by trial and error to land on the right heading
6	Textbook push	Pushes a textbook clear of the finish line

Mechanical design

Everything the rover rides on was drawn in CAD, then made and finished by hand.

WHAT I DESIGNED

- **Main frame:** the structure everything else attaches to, printed flat and folded along designed bend lines, with bays for the motors underneath.
- **Battery and motor frame:** holds the batteries and motors in place.
- **Suspension:** keeps the wheels in contact with the ground on uneven floor.
- **Sensor mounts:** position the sonar and light sensor for a clear field of view and clearance.
- **Wireless-module mount:** carries the piece that connects the rover to my computer's network.

DESIGNING FOR BENDING

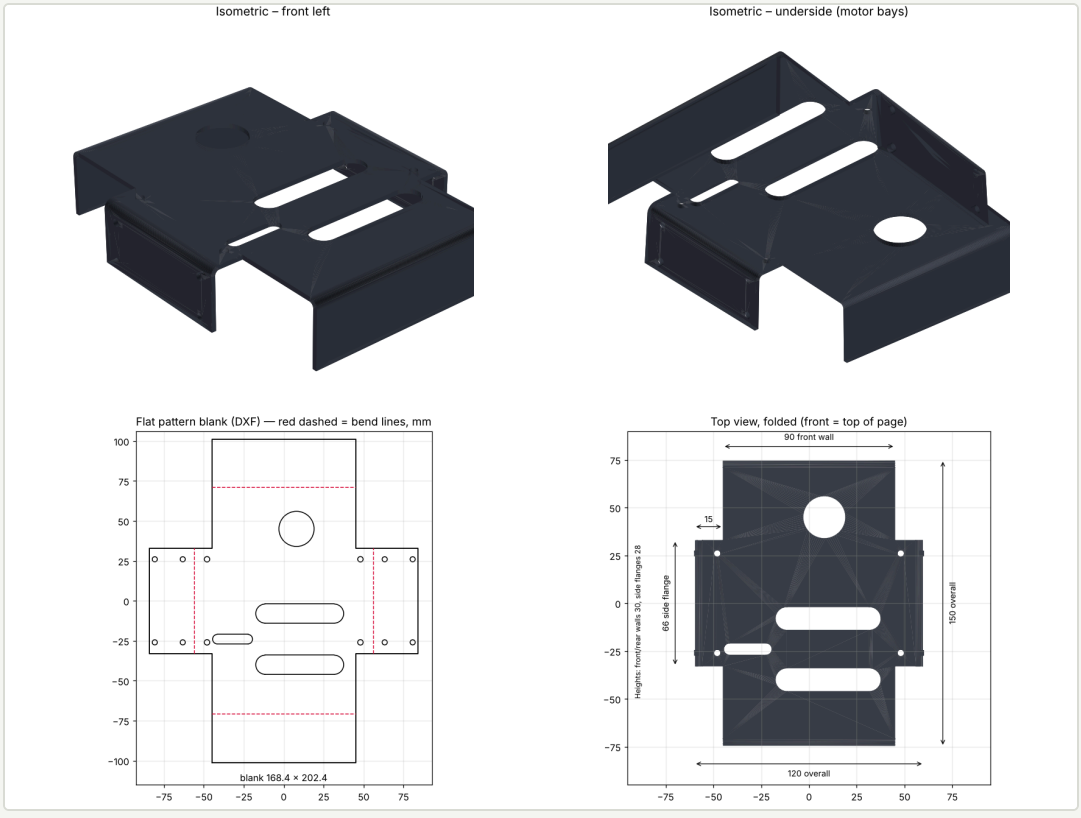
The frame is printed flat and folds along bend lines I added in CAD. I predicted where the weight of the batteries, motors and wheels would bend it and put the bend lines there. Once the rover was loaded, the parts bent at those lines.

FROM CAD TO HARDWARE

First printed in PLA; that frame bent too easily under load (next page), so the final build uses PETG. None of the 15+ printed parts was usable straight off the printer: I cut, shaped and sanded each one before assembly.

MAIN FRAME IN CAD

MATERIAL	PRINTED FLAT	FOLDED SIZE	MASS
PETG (first in PLA)	**168.4 × 202.4 mm**	**150 × 120 mm**	**62 g**



Top: the frame from above and from underneath (motor bays). Bottom left: the flat printed blank with bend lines in red (mm). Bottom right: the folded top view.

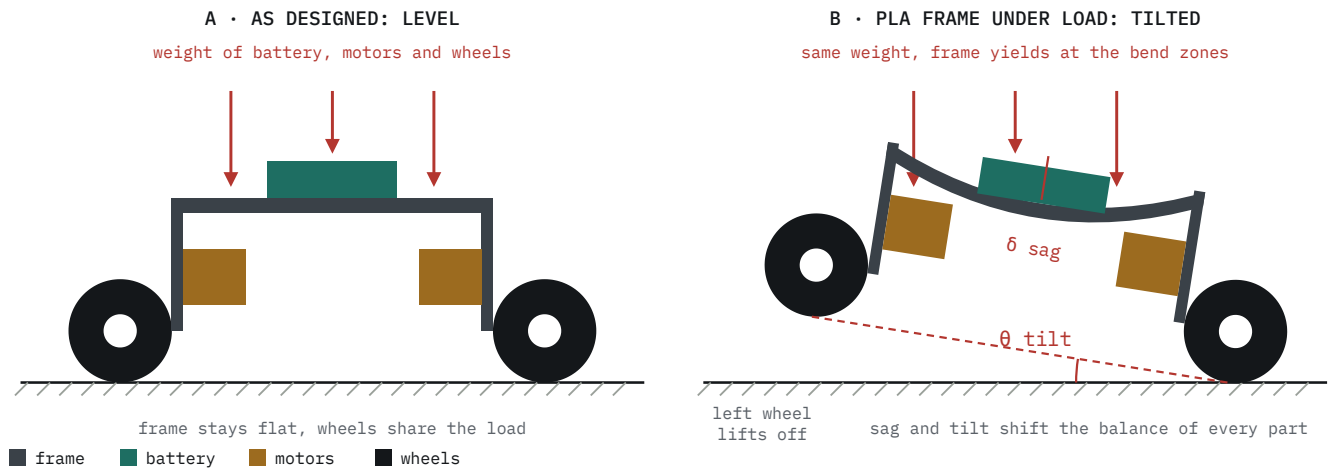
Design failure: the PLA frame bent under load

The first frame was printed in PLA. With the batteries, motors and wheels fitted, it bent so easily that the whole rover tilted off balance.

WHAT HAPPENED

- **Symptom:** under the full load, the PLA frame bent and tilted the rover, shifting the balance of every component mounted on it.
- **Why it matters:** a tilted frame changes how the wheels meet the ground and where anything mounted on it points, which undermines timed moves and sensor readings.
- **Evidence:** I didn't photograph the failure, so Figure 1 is a schematic of it, exaggerated for clarity and not to scale.

FIGURE 1 · LEVEL FRAME VERSUS THE PLA FRAME UNDER LOAD (SCHEMATIC, FRONT VIEW)



DIAGNOSIS IN ENGINEERING TERMS

STRESS

Stress (σ) is load per unit area. In bending, σ = M·c / I. The weight of the batteries, motors and wheels created a bending moment that concentrated stress in the thin bend zones.

STRAIN

Strain (ε) is deformation per unit length. In the elastic range σ = E·ε and the part springs back. Past the elastic limit the strain no longer recovers.

YOUNG'S MODULUS

Young's modulus (E) is stiffness in the elastic range. Beam stiffness scales with E·I, and I = b·t³ / 12, so thin sections are extremely sensitive to thickness.

YIELD AND DUCTILITY

Yield strength marks where permanent (plastic) deformation starts. Ductility is the strain a material tolerates. Printed PLA yields at only about 3% strain.

CREEP

Slow permanent deformation under constant load. PLA softens at about 55 to 60 °C (its glass transition), so warm batteries and motors can speed it up.

ROOT CAUSE

A constant bending load on thin, low-ductility bend zones caused local yielding and creep. The permanent set left the frame sagging, and the rover tilted.

The fix: switching to PETG

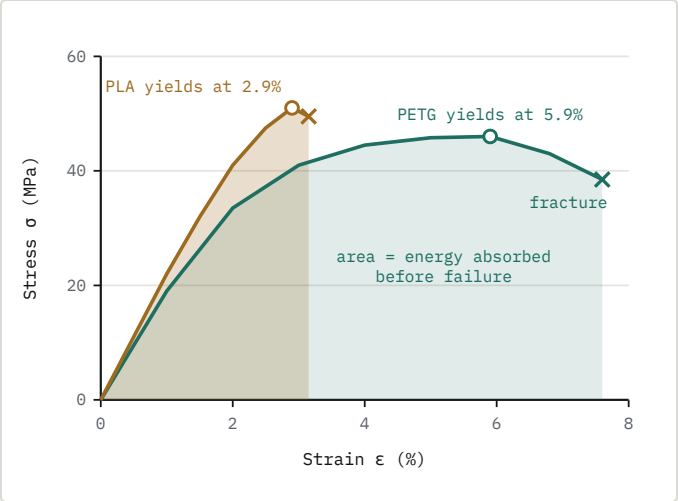
My teacher recommended PETG. I reprinted the frame in it, and it bent less under the same load.

WHY IT WORKED

Printed PETG is not stiffer than PLA: its Young's modulus is slightly lower. So the improvement did not come from elastic stiffness. It came from how each material behaves once the bend zones are loaded.

- **Strain at yield:** PETG tolerates about twice the strain before it deforms permanently.
- **Toughness:** the area under its stress–strain curve is much larger, so it absorbs more energy before failing.
- **Glass transition:** about 77 °C against 55 to 60 °C, so it resists creep near warm parts.

FIGURE 2 · STRESS–STRAIN (SCHEMATIC)



The initial slope of each curve is Young's modulus, E. Curves are sketched from the datasheet values in the table below.

PRINTED PLA VERSUS PRINTED PETG

PROPERTY	PLA	PETG	WHAT IT MEANS FOR THE FRAME
Young's modulus, E	2.2 GPa	1.9 GPa	Elastic stiffness. PLA is slightly stiffer
Tensile yield strength, σy	51 MPa	46 MPa	Similar strength, so not the deciding factor
Strain at yield	2.9%	5.9%	How far it bends before it deforms permanently
Glass transition, Tg	about 55–60 °C	about 77 °C	Creep resistance near warm batteries and motors

Typical datasheet values for printed specimens in the flat (XY) direction: Prusament PLA (ISO 527) and Ultimaker PETG (ASTM D3039). Test methods and printers differ, so compare the trends, not the decimals.

RESULT

- Same geometry and bend lines, reprinted in PETG.
- Under the same load of batteries, motors and wheels, the PETG frame bent noticeably less and the rover's balance held.
- Final frame mass: **62 g**.

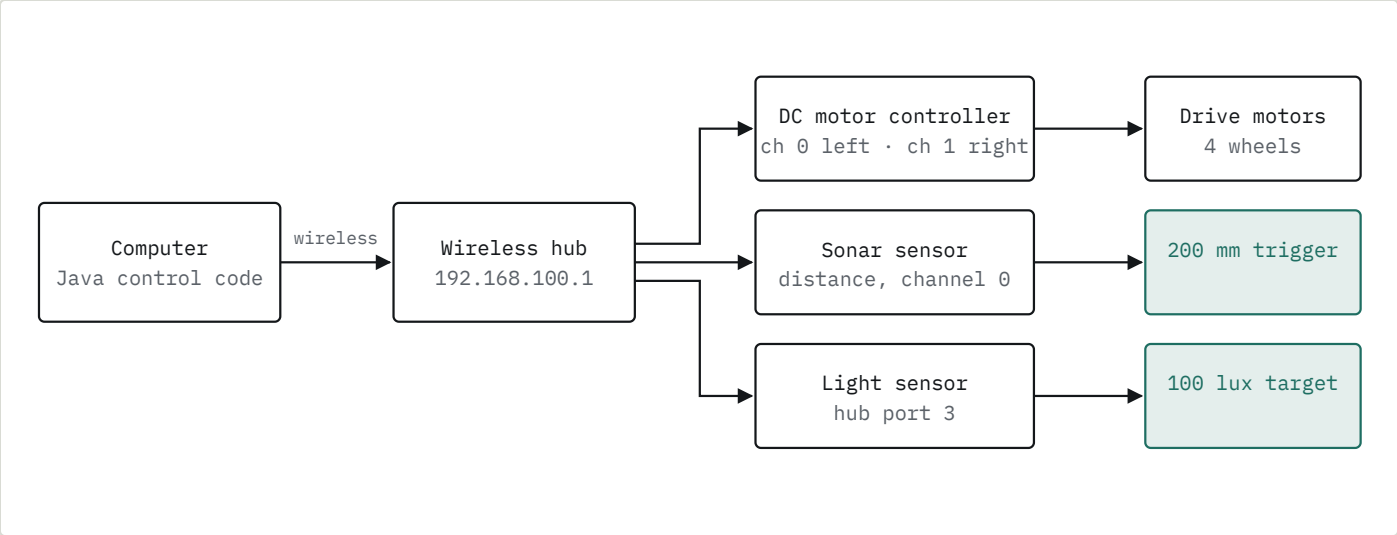
IF IT NEEDED MORE

- **Thicker bend zones:** stiffness scales with t^3, so 50% more thickness gives about 3.4× the bending stiffness.
- **Print orientation:** PETG's layer-to-layer strength is about half its in-plane strength (19 against 38.5 MPa at break), so keep bends from peeling layers apart.
- **Check first:** estimate deflection with $\delta = F \cdot L^3 / (3 \cdot E \cdot I)$ before printing.

Electronics and sensors

A wireless hub links the rover to my computer. The code runs on the computer and sends every command to the rover over the network.

SYSTEM LAYOUT



COMPONENTS

PART	ROLE	NOTES
Phidgets DC motor controller	Drives the motors	Left side on channel 0, right side on channel 1
Four drive motors and wheels	Locomotion	Wired as a left pair and a right pair, so the rover turns by driving the two sides in opposite directions
Sonar distance sensor	Sees obstacles	Reads distance in millimetres
Light sensor	Finds the lamp	Reads light level in lux, on hub port 3
Wireless Phidgets hub	Network link	Connects every device to my computer's network
Batteries	Power	Mounted in the printed battery frame

INTEGRATION WORK

- **Soldered wiring** between the sensors, hub and motor controller.
- **Sensor placement** chosen so the sonar sees ahead and the light sensor is not blocked.
- **Maximum acceleration** set on both motor channels for the course runs.
- **Sensor-driven decisions** use two thresholds: 200 mm for sonar and 100 lux for light.

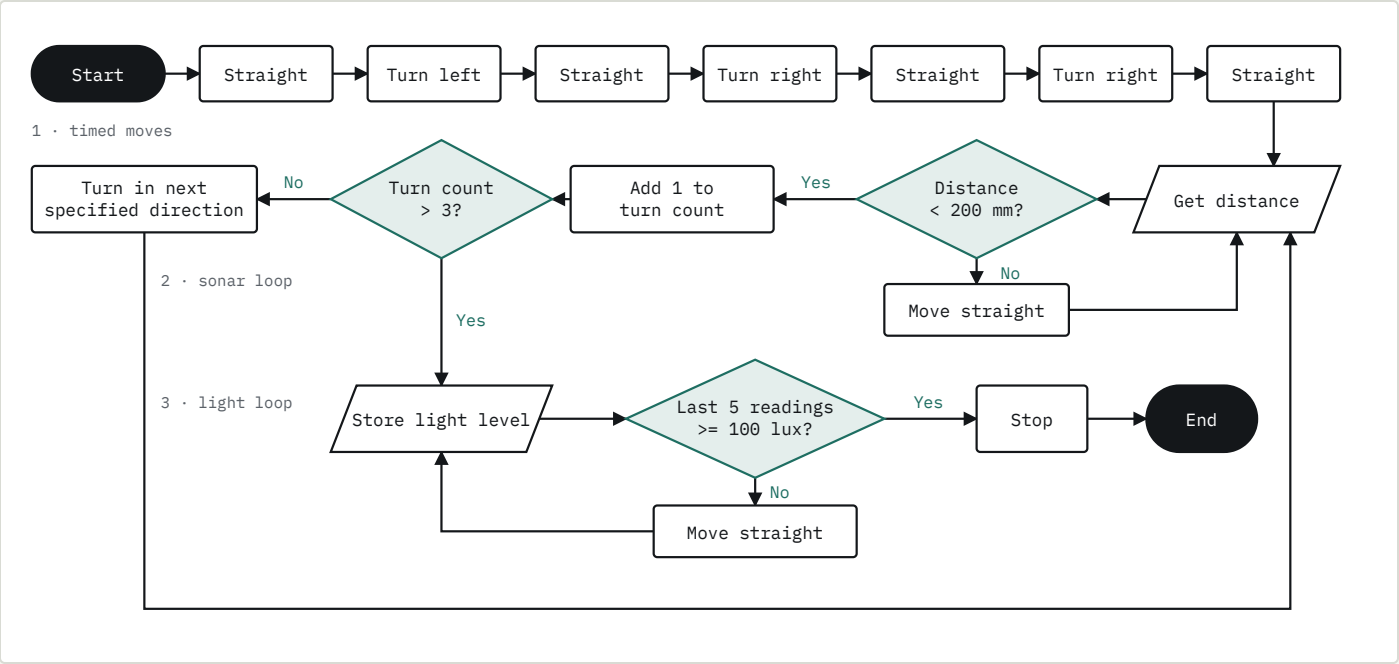
Software: four versions, each building on the last

The control code is written in Java with the Phidget22 library. I started with a square driven by pure timing and added sensing one step at a time.

HOW THE CODE EVOLVED

VERSION	WHAT IT ADDED	WHAT THE CODE DOES
1 · Square test	First working movement	Forward 3000 ms at 0.5, then a turn at full speed lasting 582, 582, 580 and 578 ms, repeated for four sides
2 · Hard-coded course	Gentler turns, max acceleration	Turns at half speed (545, 550 and 560 ms), with forward runs of 1000, 1100 and 2200 ms between them
3 · Sonar	Distance triggers each turn	Drives at 0.25. Under 200 mm it stops for 1 s, turns for 550 ms in the next preset direction, then moves on. Reads every 250 ms
4 · Light	Smoothed light reading	Drives at 0.2 and stops once the average of the last 5 readings passes 100 lux, just short of the textbook

CONTROL FLOW



The rover runs the timed moves first. It then loops on the sonar reading until it has made three sonar-triggered turns, and finally loops on the light reading until it reaches the lamp.

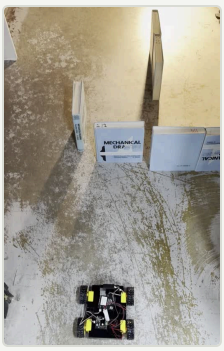
HONEST LIMITS

> **Timing, not feedback.** Turns are calibrated durations, not measured angles, so battery level and floor friction can change them. The sonar decides *when* to turn, while the turn directions are set in advance. Next step: add a gyro or wheel encoders so turns use real heading feedback.

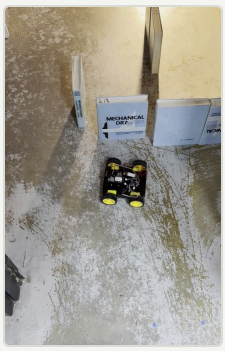
Results and reflection

The rover cleared all six stages on the demo run, finishing by pushing the textbook clear of the line.

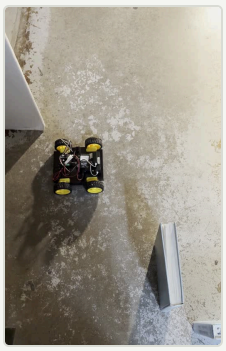
THE RUN, FRAME BY FRAME



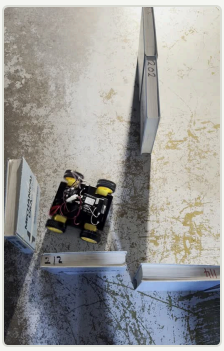
0:00
Start



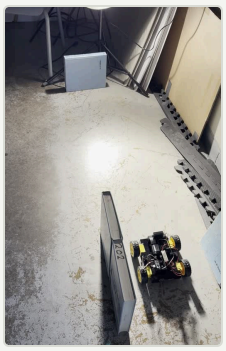
0:10
Into the course



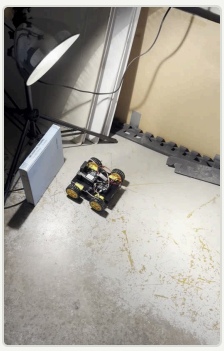
0:20
Between walls



0:30
Turning through



0:40
Lamp in view



0:47
Finish

WHAT WORKED

- Building in layers: every version was a working rover before the next sensor went on.
- Calibrating turns by hand until the rover hit each corner.
- Smoothing the light signal so the stop point was consistent.
- Designing the mounts around the sensors, not the other way round.
- Putting bend lines where I predicted the frame would flex under the weight of the batteries, motors and wheels. It bent at those lines.
- Treating the tilting PLA frame as a materials problem and fixing it by moving to PETG.

WHAT I'D CHANGE

- Replace timed turns with gyro or encoder feedback.
- Let the sonar choose the turn direction, not just the moment.
- Log every sensor reading and analyse the run afterwards, as the brief asks. The current code only prints readings to the screen.

WHAT I TOOK AWAY

> Hardware and software limit each other, and material properties decide whether a design works, not just its shape. Every mechanical choice, from how the frame flexes under load to where the sensors sit, changed what the code could rely on, and every sensor reading was noisier than I expected until I filtered it. Going from CAD to a rover that finishes a course taught me to test in small steps, change one thing at a time, and keep notes on what each change did.