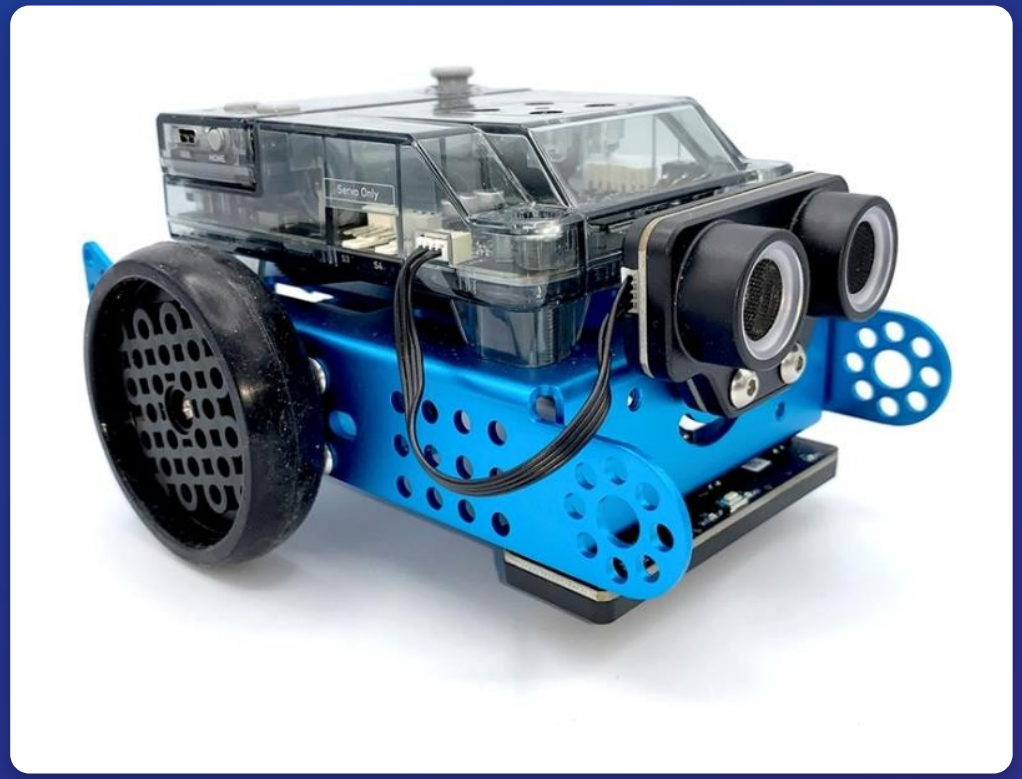
Lesson 1

Let's Move!

STEAM · 45 minutes · Beginner

Today you become the robot's driver.

You'll learn the blue driving blocks, then draw your own maze on paper and program mBot2 to drive all the way through it — without touching it!



What You Will Be Able To Do Today

1

Find the blue 'Chassis' blocks in mBlock and know what each one does

2

Drive mBot2 forward, backward, left and right — with precision

3

Turn by an exact number of degrees and travel an exact distance

4

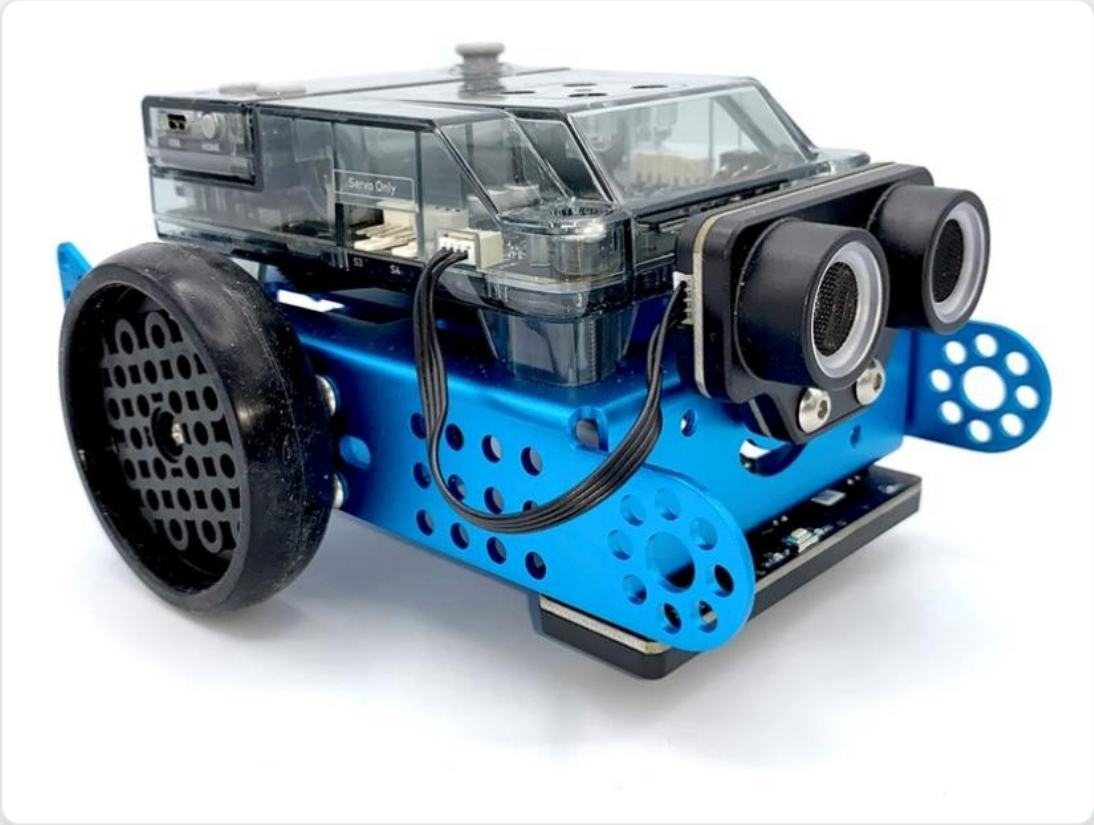
Program your robot through a maze you design yourself

Robots Are All Around You

- A robot is a machine that senses its surroundings and runs a program to decide what to do
- At home: vacuum robots, lawn-mowing robots
- At work: robot arms building cars in factories
- In space: rovers driving on Mars
- Robots do jobs that are repetitive, need great accuracy, or are dangerous

Your turn: name 3 robots you meet in everyday life

Meet mBot2 — It Has Special Motors



- mBot2 is a programmable robot that can sense, act and communicate
- You program it in mBlock by dragging and clicking blocks together
- Its motors are called encoder motors — they have a sensor built inside
- That sensor counts how far the wheel has turned
- So the robot always knows its own speed and distance travelled

Encoder = the motor counts its own turns, so you can be exact

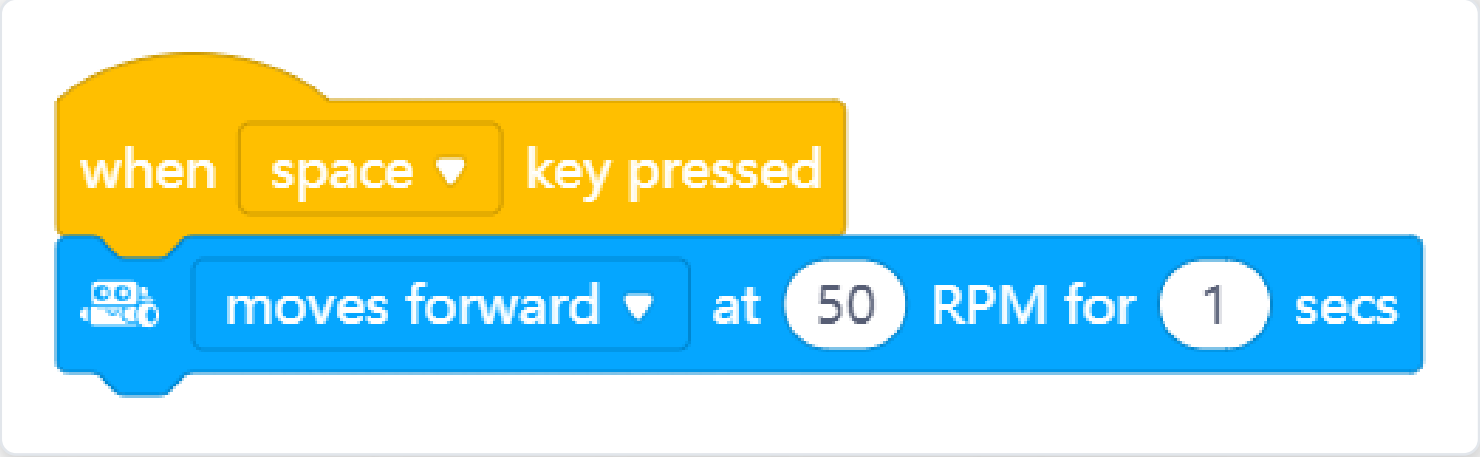
Hands-on

The Blue Driving Blocks

All the blocks that make mBot2 move live in the 'Chassis' category — and they are all blue. · 10 minutes

We work in LIVE mode today

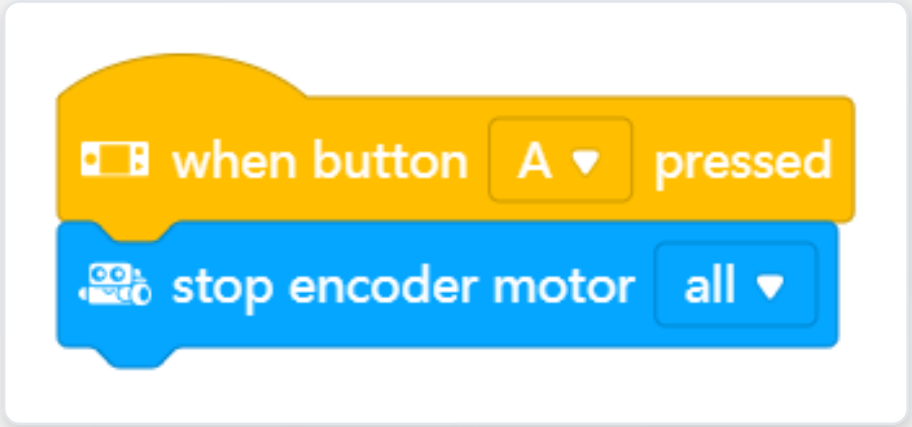
Block 1 — Drive For A Number Of Seconds



- Choose the direction: forward, backward, left or right
- Set the wheel speed in RPM — rotations per minute
- Set how many seconds it should keep going
- In this program, pressing the space key drives mBot2 forward at 50 RPM for 1 second
- Useful when mBot2 has to push a load forward

Try changing 50 to 20, then to 100 — what feels different?

Block 2 — The Emergency Stop



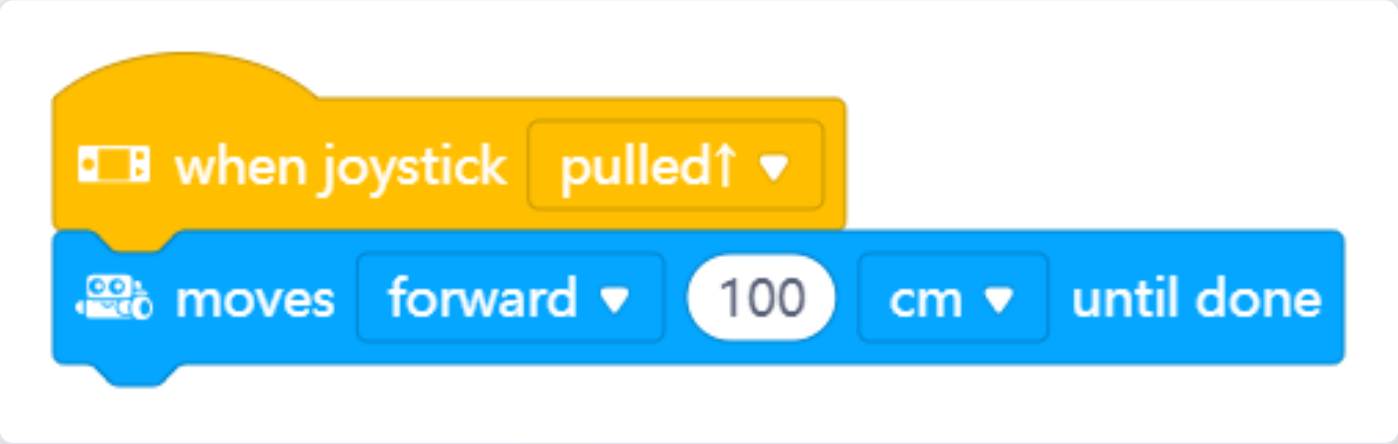
- stop encoder motor all makes the robot stop moving straight away
- This is your safety block during testing
- If a program misbehaves, this stops the robot immediately
- In the example above, pressing button A on the CyberPi stops everything

Block 3 — Turn By Exact Degrees

- Pick left or right, then type the number of degrees
- 90° is a quarter turn — a perfect corner
- 180° turns the robot right around to face backwards
- until done means the robot finishes the turn before doing anything else
- In these two programs the arrow keys steer the robot

90° = quarter turn · 180° = about face · 360° = full spin

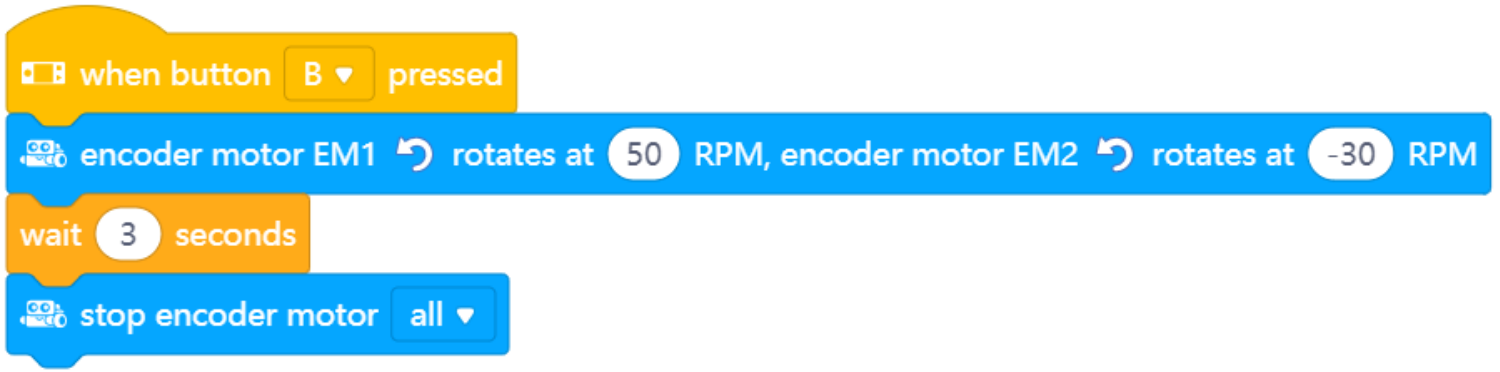
Block 4 — Travel An Exact Distance



- Here you give a distance in centimetres, not a time
- The encoder motors count the wheel turns to measure it
- This program drives forward 100 cm when you push the joystick up
- This is the block that makes a maze possible — you can plan every leg

Seconds guess the distance. Centimetres measure it.

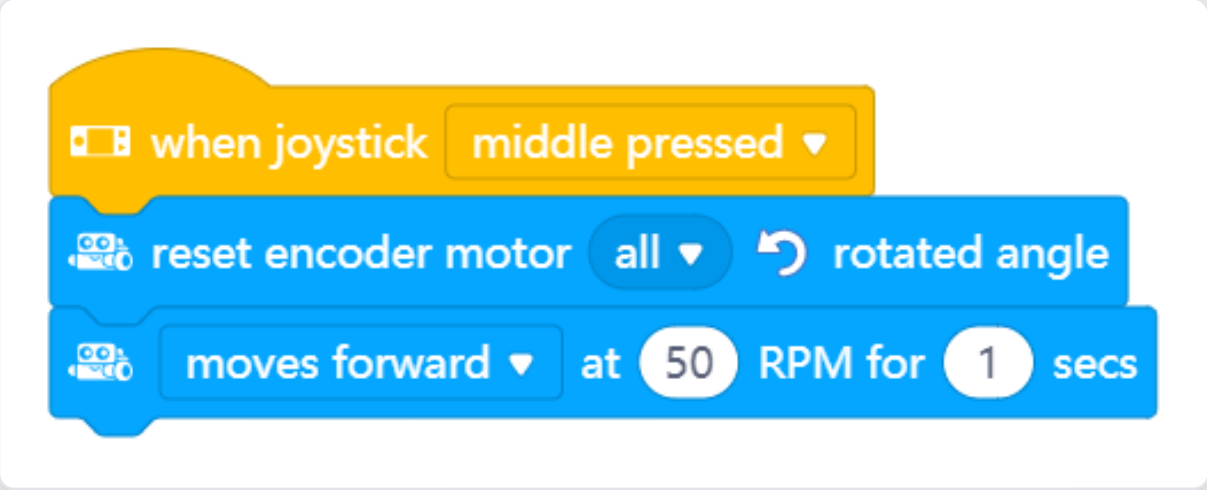
Block 5 — Control Each Wheel Separately



- EM1 and EM2 are the two motors — you can set each one's speed
- Different speeds make the robot drive in a curve
- One number is negative on purpose
- The motors face opposite ways, so one must spin backwards to go forward
- Think about it: what happens if both are +40 RPM?

Answer: the robot spins on the spot instead of driving!

Block 6 — The Robot Reports Back



- Because the motors count, you can read the numbers yourself
- Tick the checkbox next to the block and the value appears on the stage
- Speed is measured in RPM — 1 RPM is 6 degrees per second
- This program resets the count to zero, then drives forward 1 second
- Question: what number do you see on the stage afterwards?

Reset to zero first — otherwise you're reading an old number

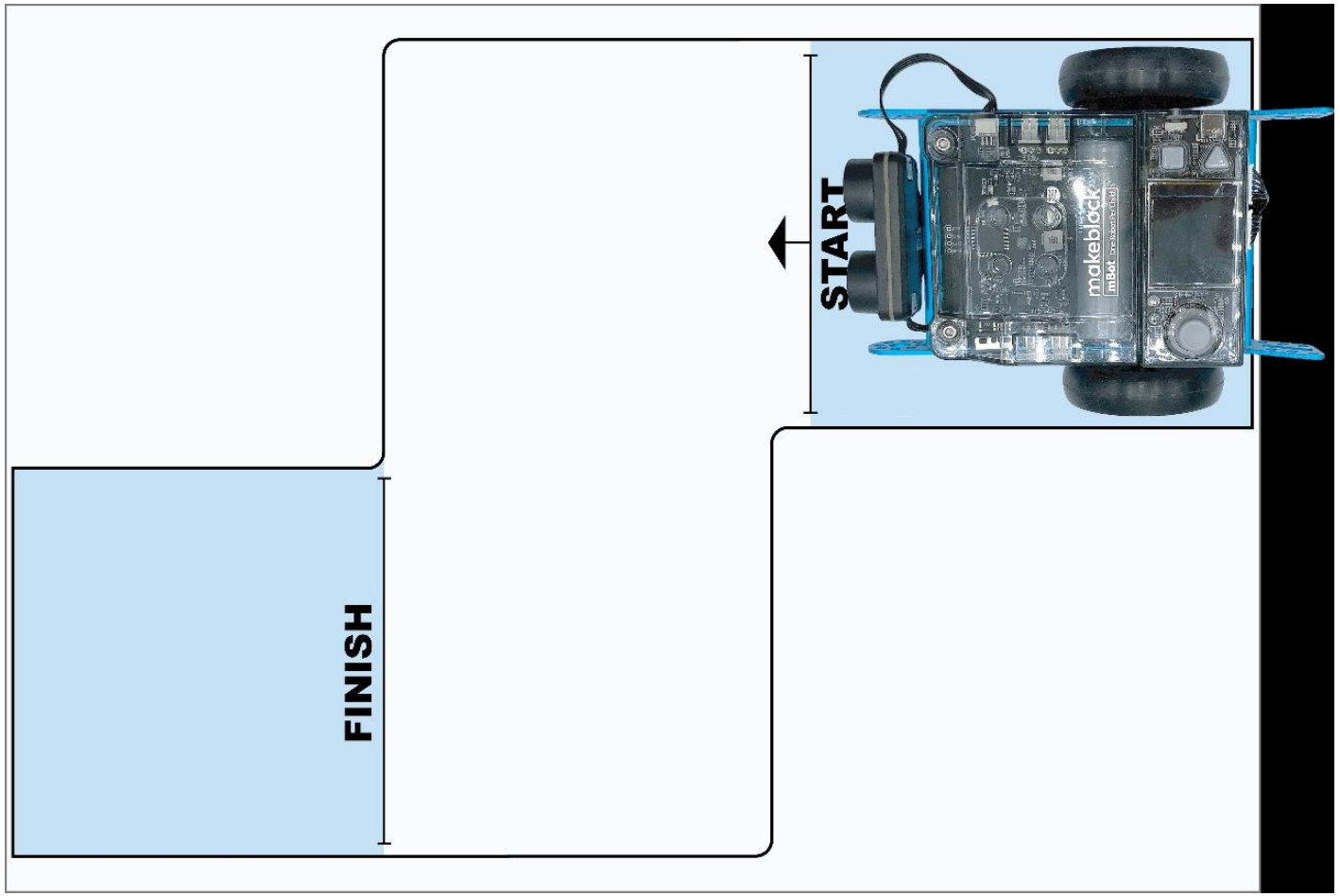
Your Mission

Drive The Maze!

- Take a sheet of A3 paper and draw your own maze
- Keep it simple, and make the corridors wider than the robot
- mBot2 does NOT find the route by itself — it drives the route you program
- Get it from START to FINISH with no mistakes

25 minutes

Need Inspiration? Here's An Example



- The robot starts at START facing along the corridor
- Break the route into legs: drive, turn, drive, turn
- For each leg write down the distance in cm
- For each corner write down left or right, and how many degrees
- Then turn your list into blocks, one line at a time

Drive · Turn · Drive · Turn — that's all a maze is

Plan Before You Code — 4 Steps

1

What do you want? What route should mBot2 drive? What parts does the maze have?

2

What do you need? How long is each distance? Which way and how many degrees for each turn?

3

Which blocks? How will you make it drive? Sketch your program in plain words or a flowchart.

4

Test and improve Write down what to fix, then fix it until the robot drives the maze perfectly.

- Do steps 1 and 2 on paper BEFORE you touch the keyboard
- Stuck? Ask a classmate, ask me, or use the built-in help on every mBlock block
- Finished early? Build a harder maze — or swap mazes with a classmate

If Your Robot Misbehaves

Robot drives off and won't stop

Build the stop encoder motor all block on button A first. It's your brake.

Turns are not quite 90°

Wheels slip on smooth floors. Try a slower speed, or adjust the number of degrees.

Distance is slightly off

That's normal! Real machines have small errors. Measure, then tune your number.

Nothing happens at all

Check Live mode is on, the cable is connected, and you pressed the right key.

Wrap-up

Showtime & Reflection

- Film a short, fun movie of your robot driving the maze
- With permission, share it with the hashtag #mBot2moves
- What are you most proud of? What would you improve?
- Which parts were easy, and which were tricky?
- What would you like more explanation about, and who could help?

Today you learned:

· What robots are and where we meet them
· That encoder motors count their own turns
· Six blue Chassis blocks
· How to drive with real precision

Next lesson: the robot's senses — sensors and data