The Maqueen robot with the micro:bit: follow the line, avoid obstacles

The little DFRobot Maqueen Lite robot is programmed from MakeCode with the official extension: a first drive, line following and the ultrasonic sensor, in three short programs.

Online guide, always up to date

By **Ada Intelisano** · informaticait.it technical team

IN SHORT

- The Maqueen Lite comes assembled: you need a micro:bit, three AAA batteries and a browser.
- In MakeCode you add DFRobot's “maqueen” extension: the blocks for the motors, the LEDs and the sensors.
- The line sensors say 1 on white and 0 on black: the robot turns toward the sensor that sees the line.
- The ultrasonic sensor measures the distance in centimeters: below 30 the robot changes course.

Time: one hour | Level: starting from scratch | No soldering

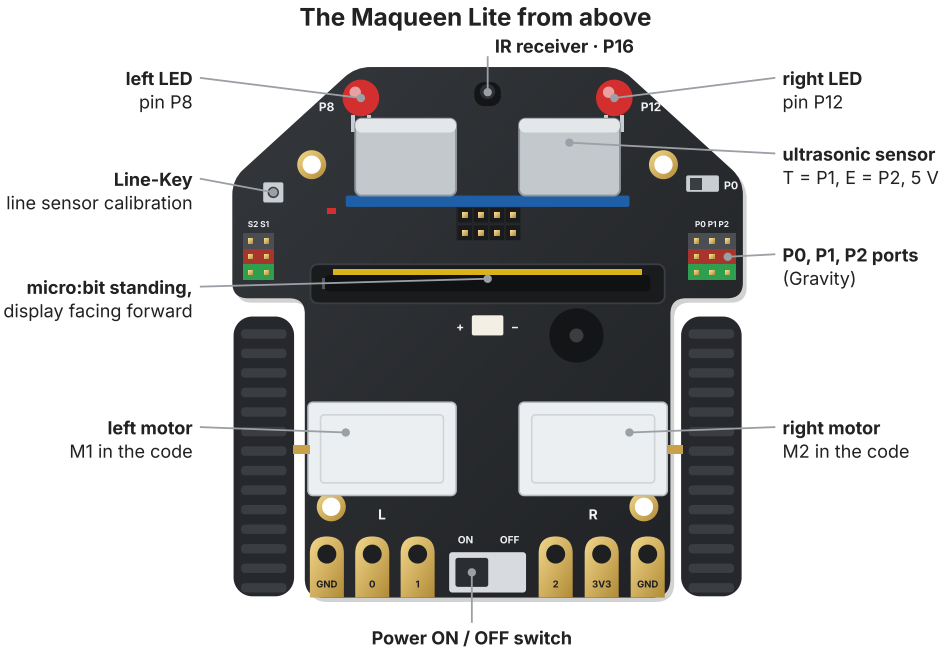
A little robot that starts when you press a button, follows a black line and steers clear of obstacles. Its brain is the **micro:bit**, the board designed for schools; its body is the DFRobot **Maqueen Lite**, which comes already assembled, with motors, wheels and sensors. You program it from the browser with **MakeCode**, using blocks or typing, and there is nothing to install.

1 What you need

- the DFRobot **Maqueen Lite** robot, with the blue or white skin. The box contains the robot with motors, wheels and battery holder already fitted, the ultrasonic sensor and the map with the track;
- a **micro:bit**, V2 or V1: it is not in the box. In our shop you can find it in the SparkFun Inventor's Kit for micro:bit v2;
- **three 1.5 V AAA alkaline batteries** (not included) and a **micro-B USB cable**;
- a computer with a browser: with Chrome or Edge the program goes straight into the board.

New word: micro:bit
A 51.6 × 42 mm board designed for schools: 25 LEDs in a 5 × 5 grid, two buttons, A and B, and on the V2 a speaker as well. You program it from makecode.microbit.org.

2 Meet the robot



under the board: 3 AAA batteries, line sensors (P13 left, P14 right), 4 RGB LEDs (P15)
drawn without the plastic skin; motors driven over I²C

informaticait.it

- The micro:bit reads the sensors and controls everything from the pins of its edge connector. It does not drive the motors directly, though: it sends the orders to a chip on the robot's board, over the I²C bus (guide no. 22).
- At the front there are the two **red LEDs**, the **infrared receiver** and the socket for the **ultrasonic sensor**. Underneath are the two **line sensors** and four colored LEDs.
- The **Line-Key** button is for calibrating the line sensors, but DFRobot already calibrates them at the factory: leave it alone for now.

3 Get the robot ready

- Set the **Power** switch to **OFF**, then put the three batteries in the battery holder under the robot, with + and − the way they are marked inside: DFRobot recommends this.
- Plug the **ultrasonic sensor** into the socket at the front, with its two cylinders facing forward.
- Slide the **micro:bit** into the slot, with the display and the buttons facing the front, as in the DFRobot photos. Hold it by the edges and do not rest metal objects on the circuits (micro:bit safety advice).

4 MakeCode and the extension

- Open **makecode.microbit.org** and create a new project.
- From the gear icon at the top right choose **Extensions** and search for **maqueen**, or paste `https://github.com/DFRobot/pxt-maqueen`. A green group of blocks appears,

“Maqueen v4”.

- The programs in this guide are written as text. At the top choose **JavaScript**, delete what is there and paste; when you go back to **Blocks**, MakeCode turns them into blocks.
- To upload them press **Download**: with Chrome or Edge you connect the micro:bit once and the program goes across directly; with other browsers you download a .hex file and drag it onto the **MICROBIT** drive. While it uploads, the status LED of the micro:bit flashes.

> **New word: extension**
> An extra package of blocks for MakeCode. The DFRobot one knows how to talk to the robot's motors and sensors: you write “left motor forward at speed 100” and it sends the right commands.

5 A first drive

```
// Press A: headlights on, one second forward, then stop
input.onButtonPressed(Button.A, function () {
    basic.showArrow(ArrowNames.North)
    maqueen.writeLED(maqueen.LED.LEDLeft, maqueen.LEDswitch.turnOn)
    maqueen.writeLED(maqueen.LED.LEDRight, maqueen.LEDswitch.turnOn)
    maqueen.motorRun(maqueen.Motors.All, maqueen.Dir.CW, 100)
    basic.pause(1000)
    maqueen.motorStop(maqueen.Motors.All)
    maqueen.writeLED(maqueen.LED.LEDLeft, maqueen.LEDswitch.turnOff)
    maqueen.writeLED(maqueen.LED.LEDRight, maqueen.LEDswitch.turnOff)
    basic.showIcon(IconNames.Happy)
})
// Press B: spin on the spot to the right for half a second
input.onButtonPressed(Button.B, function () {
    basic.showArrow(ArrowNames.East)
    maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 80)
    maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CCW, 80)
    basic.pause(500)
    maqueen.motorStop(maqueen.Motors.All)
    basic.showIcon(IconNames.Happy)
})
basic.showIcon(IconNames.Happy)
```

- `motorRun(motor, direction, speed)`: **M1** is the left motor, **M2** the right one, **All** both of them; **CW** is forward (“Forward” in the block), **CCW** backward; the speed goes from 0 to 255.
- To spin on the spot, one motor goes forward and the other backward.

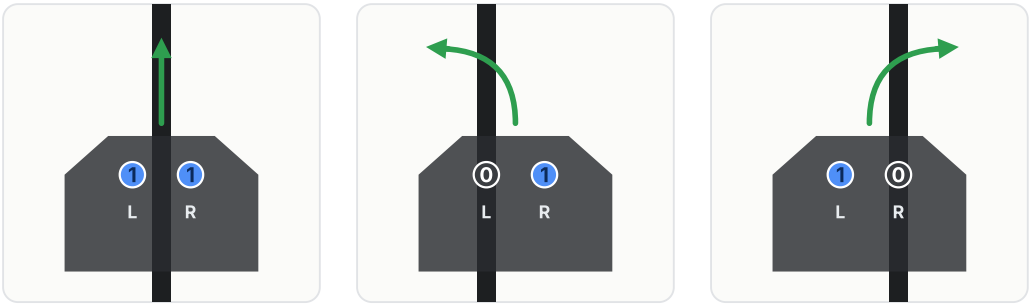
> **✓ What you should see**
> The display shows a smiley face. Press A: the arrow appears, the headlights come on, the robot goes forward for one second and stops. Press B: it turns to the right.

If it doesn't work

- Do your tests on the floor, not on the table: a robot that sets off suddenly can easily fall.
- It does not move but the micro:bit responds: is Power set to ON? If the power LED is on and the motors stay still, according to DFRobot the batteries may be flat.
- Red blocks or errors: the maqueen extension is not loaded in the project.

6 Follow the line

How it follows the line: two sensors, three cases



straight
L 1 · R 1
M1 40 · M2 40

turn left
L 0 · R 1
M1 back 10 · M2 40

turn right
L 1 · R 0
M1 40 · M2 back 10

1 = the sensor sees white, 0 = it sees black (DFRobot wiki); diagram, not to scale

informaticait.it

Each line sensor says **1** when it sees white underneath and **0** when it sees black; under the robot two blue indicator LEDs light up on white and go out on black. The official DFRobot program keeps the line between the two sensors and goes slowly, because the map is small.

```
// Follow the black line: 1 = the sensor sees white, 0 = it sees black
basic.forever(function () {
    let left = maqueen.readPatrol(maqueen.Patrol.PatrolLeft)
    let right = maqueen.readPatrol(maqueen.Patrol.PatrolRight)
    if (left == 1 && right == 1) {
        // the line runs between the sensors: straight on, slowly
        maqueen.motorRun(maqueen.Motors.All, maqueen.Dir.CW, 40)
    } else if (left == 0 && right == 1) {
        // the left sensor is on the black: turn left
        maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CCW, 10)
        maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 40)
    } else if (left == 1 && right == 0) {
        // the right sensor is on the black: turn right
        maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 40)
        maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CCW, 10)
    }
})
```

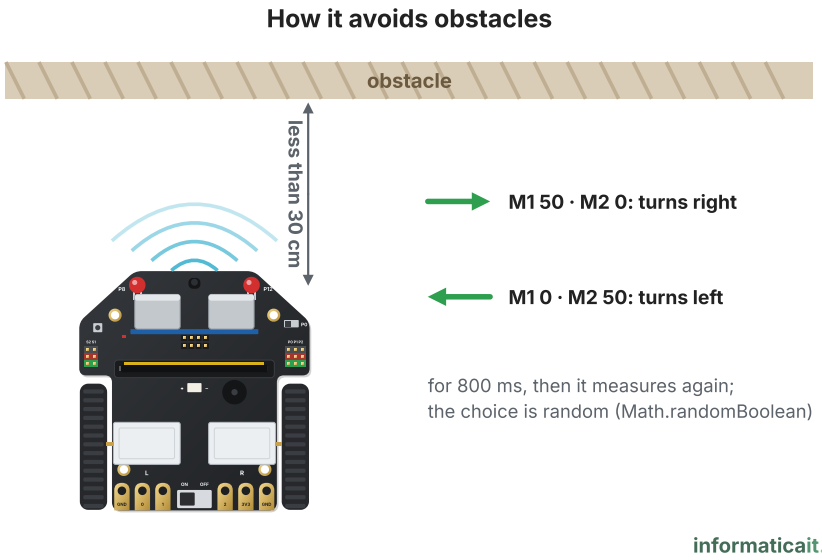
✓ What you should see

Put the robot on the map with the black line between the two sensors: it follows the track, correcting itself in small jerks.

If it doesn't work

- It runs off the track on the bends: lower the speed, from 40 to 30.
- First of all, look at the blue indicator LEDs: if they do not change when they go from white to black, the problem is the sensors, not the program.

7 Steer clear of obstacles




```
// Avoid obstacles: below 30 cm it turns, randomly to the right or to the left
basic.forever(function () {
    let distance = maqueen.Ultrasonic()
    if (distance < 30) {
        if (Math.randomBoolean()) {
            maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 50)
            maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 0)
        } else {
            maqueen.motorRun(maqueen.Motors.M1, maqueen.Dir.CW, 0)
            maqueen.motorRun(maqueen.Motors.M2, maqueen.Dir.CW, 50)
        }
        basic.pause(800)
    } else {
        maqueen.motorRun(maqueen.Motors.All, maqueen.Dir.CW, 50)
    }
})
```

`maqueen.Ultrasonic()` returns the distance in centimeters; when no echo comes back, the extension returns 500, which means “nothing ahead”. The logic is that of the official DFRobot example, written for an older version of the extension and adapted here to the current one.

✓ What you should see

The robot goes straight; when it gets closer than 30 cm to a wall or a box it turns, sometimes right and sometimes left, at random, and sets off again.

If it doesn't work

- It bumps into things: is the sensor pushed all the way in, with the cylinders facing forward?
- It turns even with no obstacles: something close is in front of the sensor, for example a cable.

Try it yourself

- Light up the four colored LEDs: add the **neopixel** extension and create the strip on pin P15 with 4 LEDs, as in the DFRobot examples.
- Make it play a note when it sees an obstacle.
- Fit a Mechanic kit, for example the loader, and make it move something.

✓ What you learned

- the micro:bit controls the robot from the pins of its edge connector and, for the motors, over I²C;
- a line sensor says 1 on white and 0 on black;
- to turn, you just run the two motors at different speeds.

Sources: DFRobot, micro:Maqueen Lite (ROB0148-EN, -LB, -LW): product pages, wiki (Getting Started, Read Line-tracking Sensor, Auto Line-tracking, Ultrasonic Obstacle-avoiding) and official MakeCode examples; dimension drawing · DFRobot, pxt-maqueen 1.7.17 MakeCode extension (GitHub) · Micro:bit Educational Foundation, microbit.org and tech.microbit.org: features, transferring code, safety advice; Microsoft MakeCode for micro:bit, blocks reference · informaticait guide no. 22