Au Reikura
School of Computing and Mathematical Sciences

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

DEPARTMENT OF
COMPUTER SCIENCE
Te Tari Rorohiko

2025

Contributors

J. Turner

V. Moxham-Bettridge

J. Bowen

J. Kasmara

MICRO:BITS & MAQUEEN

Last week we introduced you to programming Micro:bits with MicroPython which was great preparation for this week's session (as we're using the Micro:bits in conjunction with the Maqueen robots to build robot cars!).

What is the Maqueen robot?

Purposely built for STEM education, the Maqueen robot was designed to provide an entry-level platform for people to learn about embedded systems (i.e. how to design, build and program them). As the robot is powered by the BBC Micro:bit, it makes for a good starting place for beginners because of their graphical programming code blocks interface, however as you have already used MicroPython, that will be our tool of choice for this session.

See figures 26 and 27 for features of the Maqueen robot.

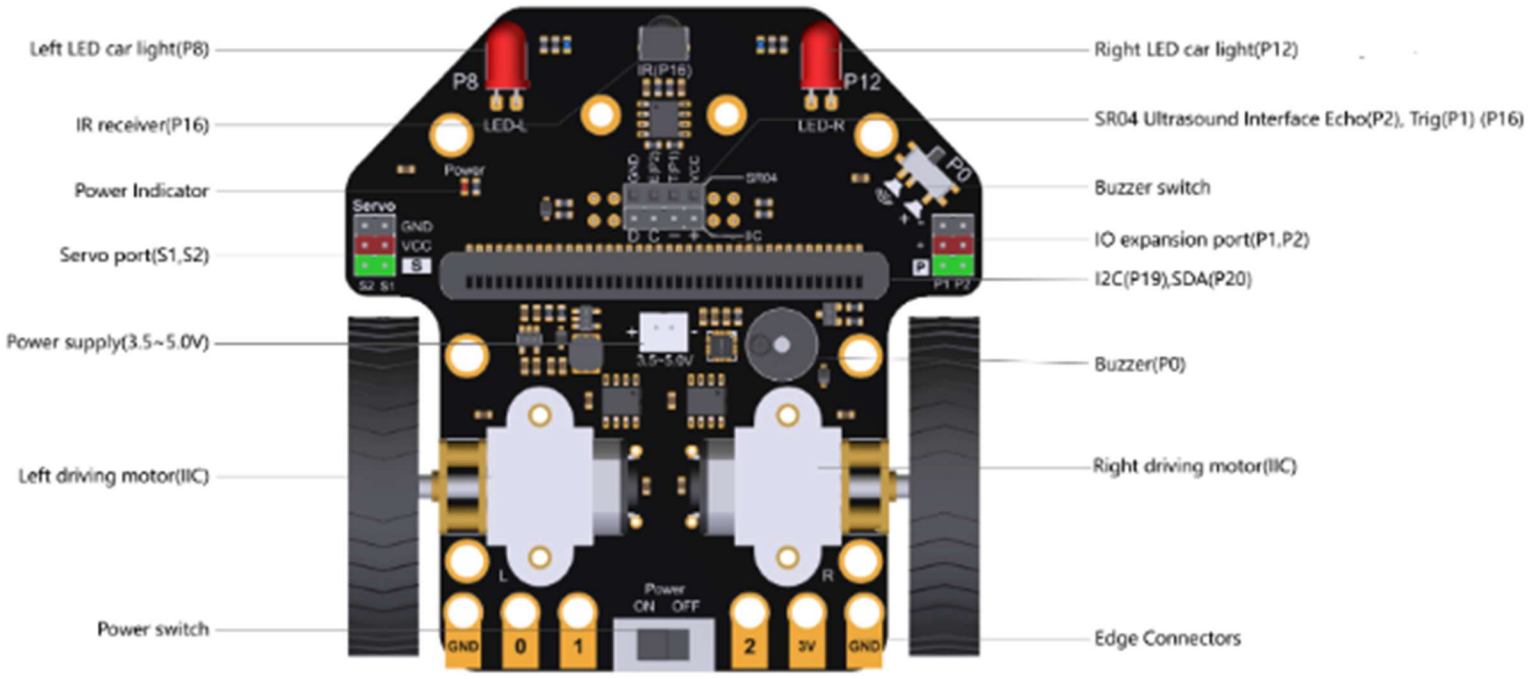

Figure 26: The top-side components of the Maqueen robot

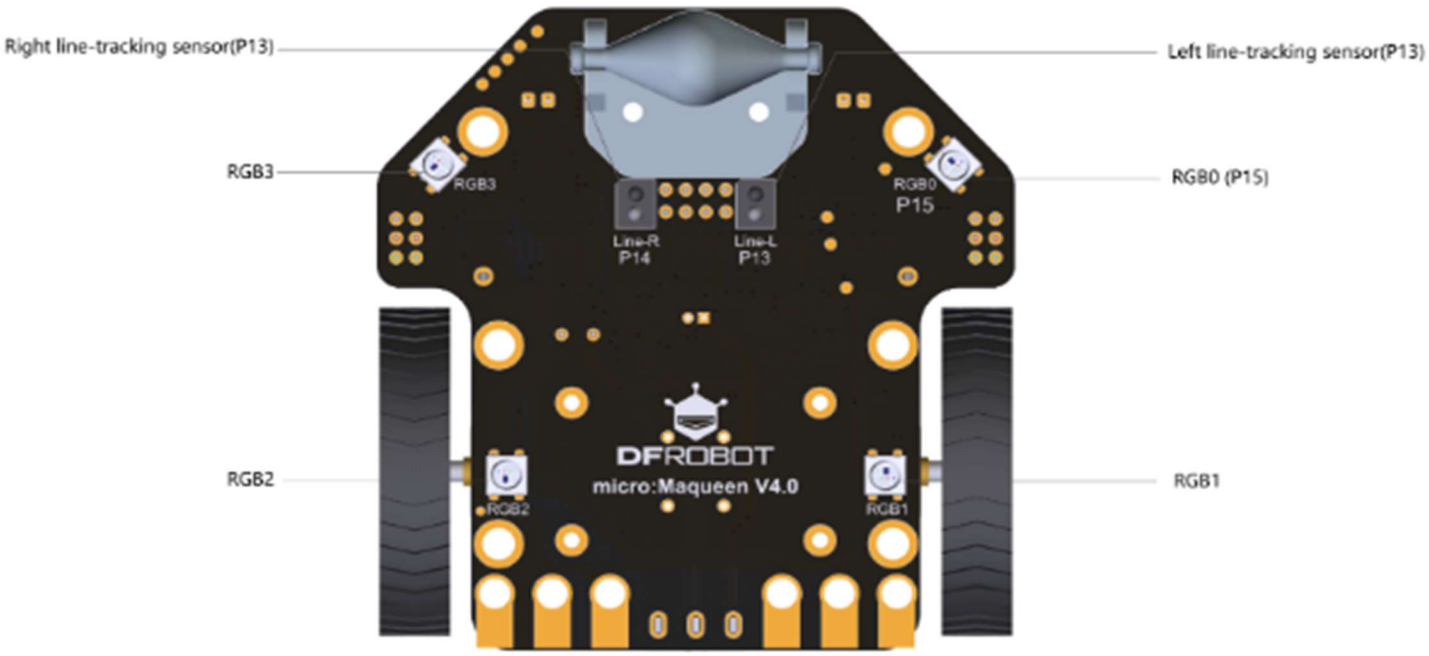

Figure 27: The bottom-side components of the Maqueen robot

Getting Started

Download the project file provided on the Slack channel (has a `.hex` extension) and import it into the MakeCode editor (feel free to seek assistance from staff about this).
To be able to use the Micro:bit to instruct the Maqueen robot, an extension must be enabled but don't worry, we've already done this for you (you should be able to see an additional option in the module panel on the left hand side, like in figure 28 below).

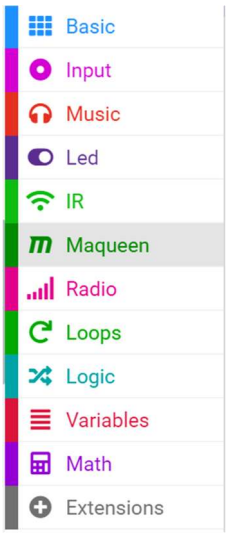

Figure 28: The module panel on the left hand side

Click on the Maqueen module to see the available functions (see figure 29).

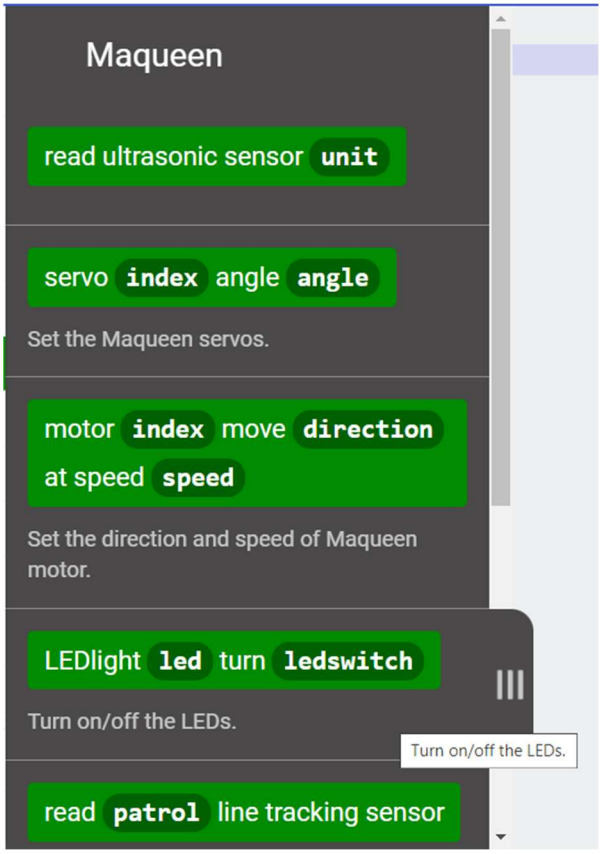

Figure 29: The available functions of the 'Maqueen' module

To start with, let's turn on the LEDs (refer back to figure 26 for their location) by using the `write_led` function. Instead of having to type it out, the MakeCode editor makes it easy by allowing us to drag the function we want to use onto the code panel. Do this now with the `LEDlight` function tile which should give you this line of code:

```
maqueen.write_led(maqueen.LED.LED_LEFT, maqueen.LEDswitch.TURN_ON)
```

Download the program to the Micro:bit and once complete, insert the Micro:bit into the Maqueen (feel free to ask staff for assistance).

Turn the Maqueen robot on, what happens?

Hopefully the left LED (labelled "LED-L") at the front of the robot turned on, if it didn't, make sure the robot is switched on and that you used the correct function.

To make the robot move, we have to instruct the motors to work, so drag the `motor` function tile into your code (see figure 30).

```
1
2   maqueen.write_led(maqueen.LED.LED_LEFT, maqueen.LEDswitch.TURN_ON)
3
4   maqueen.motor_run(maqueen.Motors.M1, maqueen.Dir.CW, 0)
```

Figure 30: What our code currently looks like

Ignoring line 2 for now, line 4 is what makes the robot move so let's break it down. `maqueen.motor_run` is a *function* of the Maqueen *module* (similar to the `on_forever` one you used last week) and `maqueen.Motors.M1`, `maqueen.Dir.CW` and `0` are *parameters* of the function (external values given to a function). What they tell us is, to spin the `M1` motor in a `clockwise direction` at a `speed of zero`. Confusingly, this instructs the left motor to not spin at all, so let's change that by replacing the `0` with `25`. The robot should now slowly spin in a clockwise direction.

If the code on line 4 is for the left motor, how would you get the robot to move forwards in a straight line? Hint: do the same but for the other motor (feel free to ask for help if you get stuck).

You know how to turn the LEDs on/off, make the robot drive in a straight line as well as turn but how do you make it stop? There are 2 ways to stop a robot; by using the `motor_run` function for both motors with speed set to 0 (we did this earlier) or using the `motor_stop` function. As we have already covered the first method, let's try the second one.

Drag the `motor` function tile seen in figure 31 into your code panel.

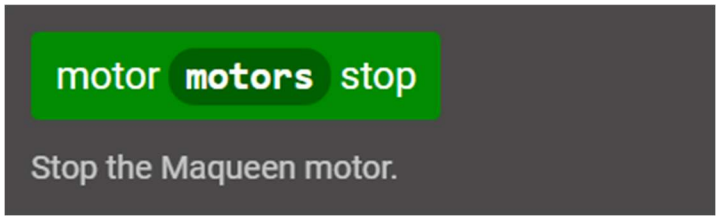

Figure 31: The motor_stop Maqueen function

If your code contains all that we have covered so far, when turned on the robot's left LED should light up but stay stationary (i.e. no movement).

Now you know the basics, you're tasked with making the robot follow and complete the course laid out at the front of the room (the track diagram is presented in figure 32). Have a look at the exercises below for a basic development guide.

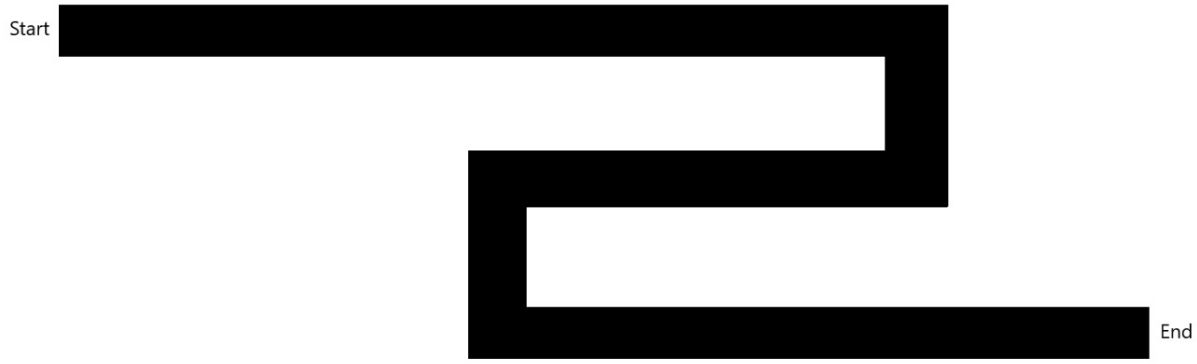

Figure 32: The diagram of the course to complete

Exercises:

1. Get your robot moving from the start line to the end of the first straight.
2. Make your robot turn right (clockwise) around the first corner.
3. Make your robot travel in a straight line down the second straight.
4. Make your robot turn left (anticlockwise) around the second corner.
5. Get your robot to cross the finish line.

Summary

Today we have used Maqueen robots in conjunction with Micro:bits to make small moving vehicles that raced around a track. Hopefully, your recent exposure to MicroPython in the last session made this activity fun and easy to complete. Next week, we'll be exploring human-robot interaction by interacting with our robot, Cruz.

Advanced exercises

Has your robot already crossed the finish line? Want to add some extra features? Modern vehicles (such as cars and trucks etc) have lots of features (like headlights, taillights, horns etc) so why don't you implement some additional actions and features? To start with, you should add indicators to your robot. When turning right (clockwise), the front-right RGB should glow orange, and when turning left, the front-left RGB should glow orange. To do this, you will need

to add another extension to your workspace which you can do by clicking on 'Extensions' in the left side module panel. Then select the 'neopixel' extension (see figure 33).

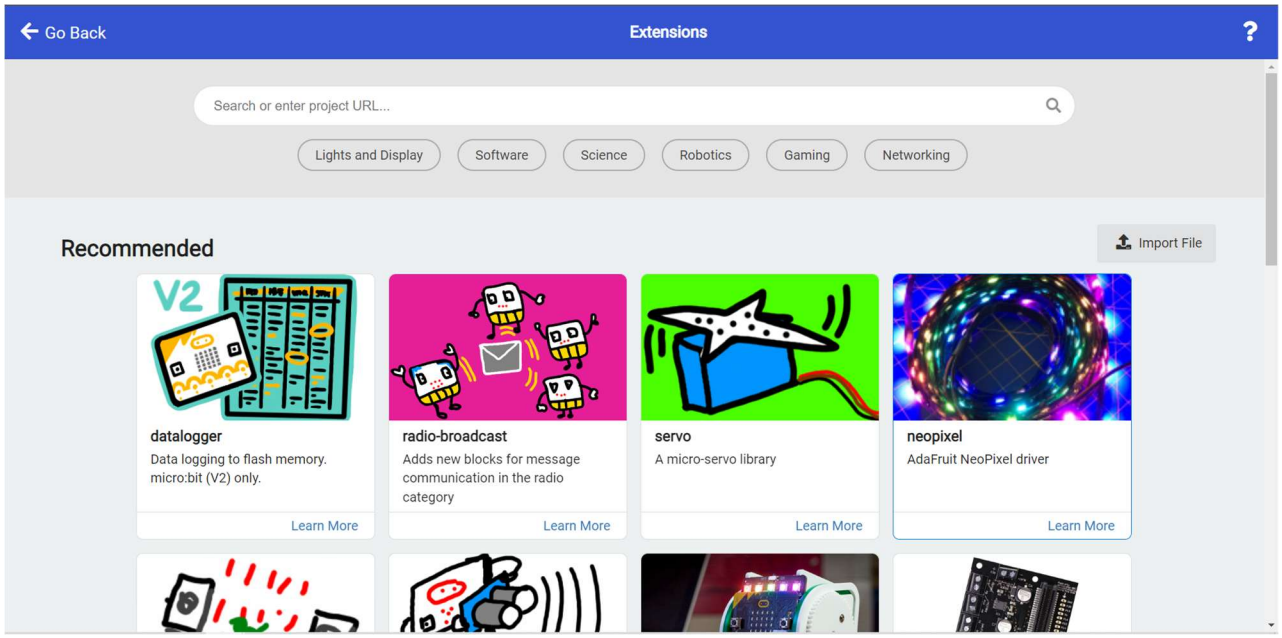

Figure 33: Enabling the 'neopixel' extension

Once you have got the indicators working, consider doing the following exercises:

1. Currently, your indicators just glow for a short period of time, can you make them blink? (i.e. turn on and off repeatedly).
2. If your car were to come to a stop, the rear RGBs should glow red.
3. Using the front distance sensors, can you make the car reverse when it detects an object?
4. When the car is reversing, make the vehicle make a reversing noise.

Useful Resources

- BBC Micro:bits Introduction page: https://microbit.org/get-started/first-steps/introduction/
- MicroPython Home page: https://micropython.org/
- DFRobot Education homepage (the manufacturers of Maqueen): https://edu.dfrobot.com/tag-497.html
- Hackster.io beginner Maqueen project: https://microbit.hackster.io/robocircuits/best-educational-robot-5461f1
- W3 Schools Python tutorials: https://www.w3schools.com/python/default.asp