

18th Annual IEEE
Ottawa Robotics Competition
18^e Compétition annuelle de
robotique d'Ottawa d'IEEE



Micro:bit Basics

About Micro:bit

- Microcontroller developed by the BBC
- Designed to be easy and fun to program
 - Tutorials: <https://microbit.org/>
 - Documentation: <https://makecode.microbit.org/docs>
- Requires 3 AAA batteries as a power source
- Has a Bluetooth connectivity, compass, accelerometer, and uses the front LEDs as light sensors



Sensors on the Micro:bit

Code Reference(s)	Description
light level	“Light sensor” (front LEDs)
temperature	“Temperature sensor” (processor’s temp)
Button A/B	A/B buttons on LED side of the micro:bit

Would recommend against using the compass and accelerometer in developing the challenge since this is for younger kids.



DFRobot Maqueen Kit

- Kit that includes:
 - Wheels and roller (movement)
 - Ultrasonic sensor (distance detection)
 - Infrared sensors (black/white line detection)
 - Various coloured LEDs
- Micro:bit commands the kit's movement and is placed behind the ultrasonic sensor (LED side facing the front)



Ports on DFRobot Maqueen Kit

Code Reference(s)	Description
M1	Left motor
M2	Right motor
PatrolLeft / P13	Left IR sensor
PatrolLeft / P14	Right IR sensor
LEDLeft / P8	Left red LED (front)
LEDRight / P12	Right red LED (front)
Sensor unit	Ultrasonic sensor
P15	Multi-coloured LEDs under the robot
P0	Sound sensor



Notes on Motors/Sensors/LEDs

Part	Notes
Motors	CW – forward, CCW – backwards 0 – nowhere, 255 – full speed
IR sensors	0 – black, 1 – everything else
Multi-coloured LEDs	You must add the Neopixel extension to the project. Custom RGB colours are also possible.
Sound	Mute vs. not mute



Programming with Micro:bit

- Coding is done here:
<https://makecode.microbit.org/>
- Can be programming in blocks or JS
- Students can save the code (as backup) and import the code to other computers
- Code can be downloaded straight to the micro:bit or drag and dropped onto it
- The Micro:bit has **limited** memory!



Programming with Micro:bit

- To work with the Maqueen kit, we do need to add an extension (use the Gear icon on the top right corner when starting a project to add)
- Library link:
<https://github.com/DFRobot/pxt-maqueen>



Coding Example: Line Following

```
forever
  if << Read Patrol PatrolLeft >> = <> 0 and <> Read Patrol PatrolRight >> = <> 0 then
    Motor M1 dir CW speed 0
    Motor M2 dir CW speed 255
  else if << Read Patrol PatrolLeft >> = <> 1 and <> Read Patrol PatrolRight >> = <> 0 then
    Motor M1 dir CW speed 255
    Motor M2 dir CW speed 0
  else if << Read Patrol PatrolLeft >> = <> 0 and <> Read Patrol PatrolRight >> = <> 1 then
    Motor M1 dir CW speed 0
    Motor M2 dir CW speed 255
  else
    Motor M1 dir CW speed 255
    Motor M2 dir CW speed 0
```



Coding Example: Obstacle Detection

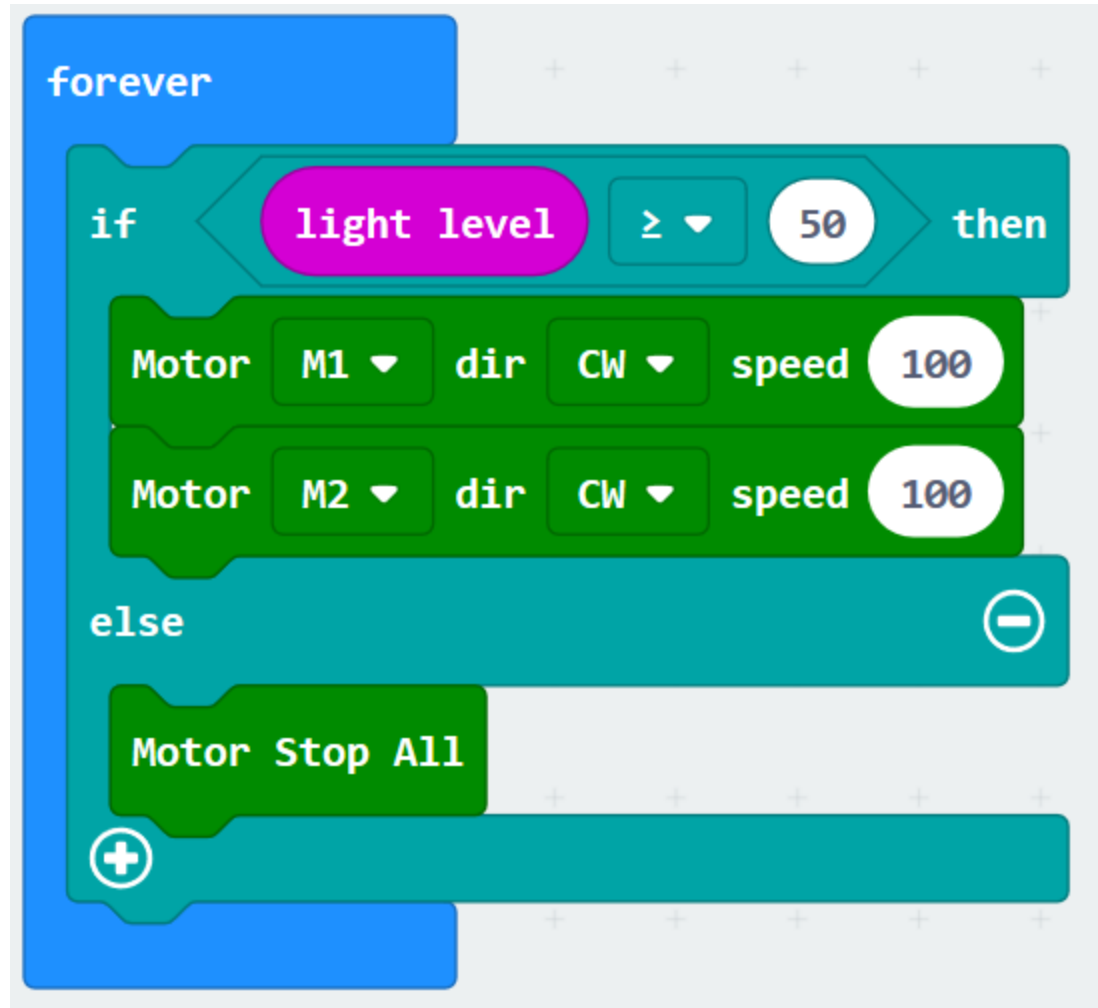
```
forever
  if sensor unit cm < 20 then
    Motor M1 dir CW speed 0
    Motor M2 dir CW speed 0
  else
    Motor M1 dir CW speed 255
    Motor M2 dir CW speed 255
```

The image shows a block-based programming script for obstacle detection. It starts with a 'forever' loop block. Inside the loop, there is an 'if' block. The 'if' block's condition is 'sensor unit cm' followed by a '<' operator and the number '20'. If this condition is true, the 'then' branch contains two 'Motor' blocks: one for 'M1' and one for 'M2', both set to 'dir CW' and 'speed 0'. If the condition is false, the 'else' branch contains two 'Motor' blocks: one for 'M1' and one for 'M2', both set to 'dir CW' and 'speed 255'. The script ends with a '+' sign on the bottom of the 'forever' loop block.

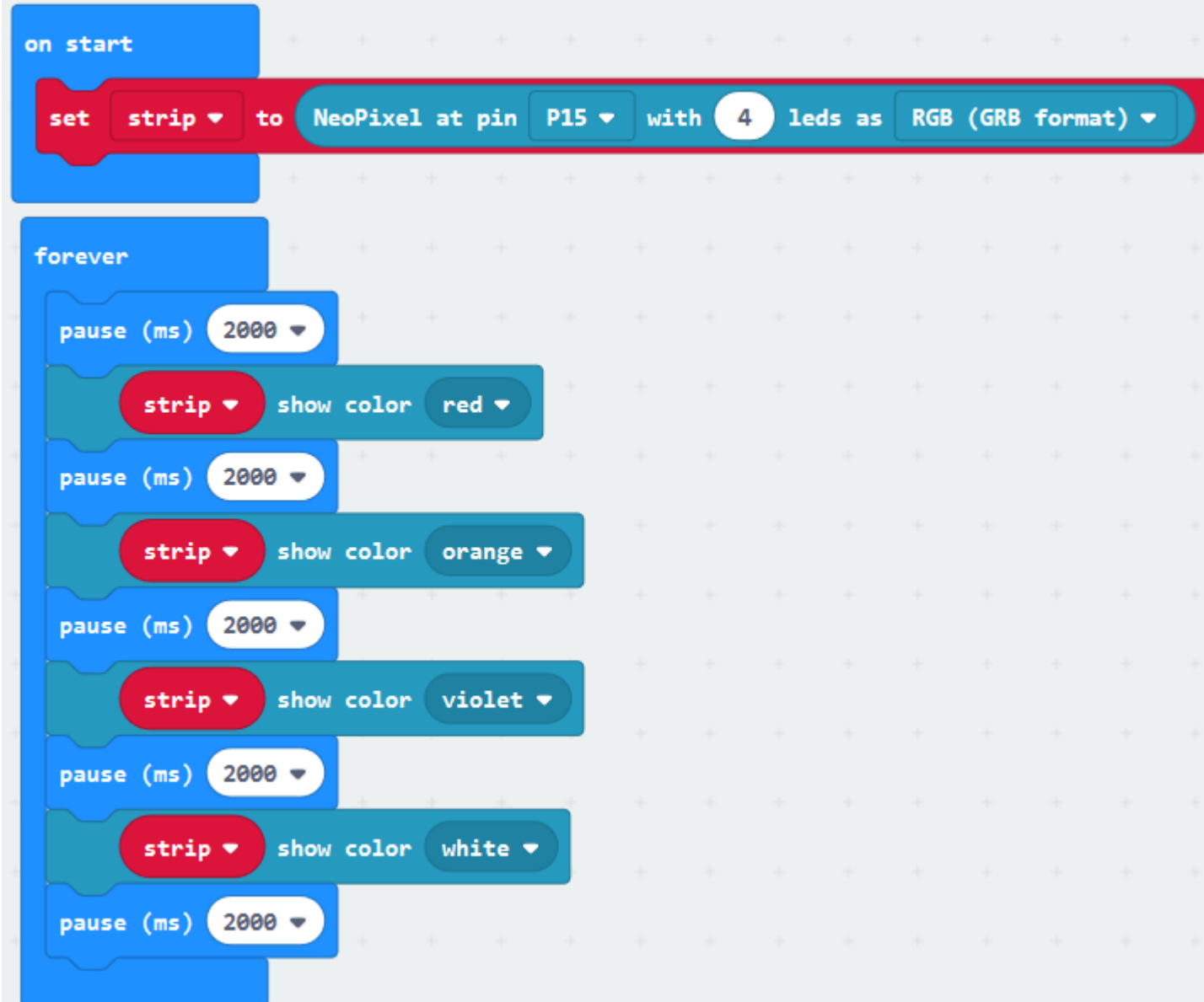


Coding Example: Follow Bright Light

```
forever
  if light level ≥ 50 then
    Motor M1 dir CW speed 100
    Motor M2 dir CW speed 100
  else
    Motor Stop All
```



Coding Example: LEDs



The image shows a Scratch script for controlling an LED strip. It starts with an 'on start' block containing a 'set strip to NeoPixel at pin P15 with 4 leds as RGB (GRB format)' block. This is followed by a 'forever' loop containing a sequence of 'pause (ms) 2000', 'strip show color', and 'pause (ms) 2000' blocks. The colors shown in the loop are red, orange, violet, and white.

```
on start
  set strip to NeoPixel at pin P15 with 4 leds as RGB (GRB format)

forever
  pause (ms) 2000
  strip show color red
  pause (ms) 2000
  strip show color orange
  pause (ms) 2000
  strip show color violet
  pause (ms) 2000
  strip show color white
  pause (ms) 2000
```