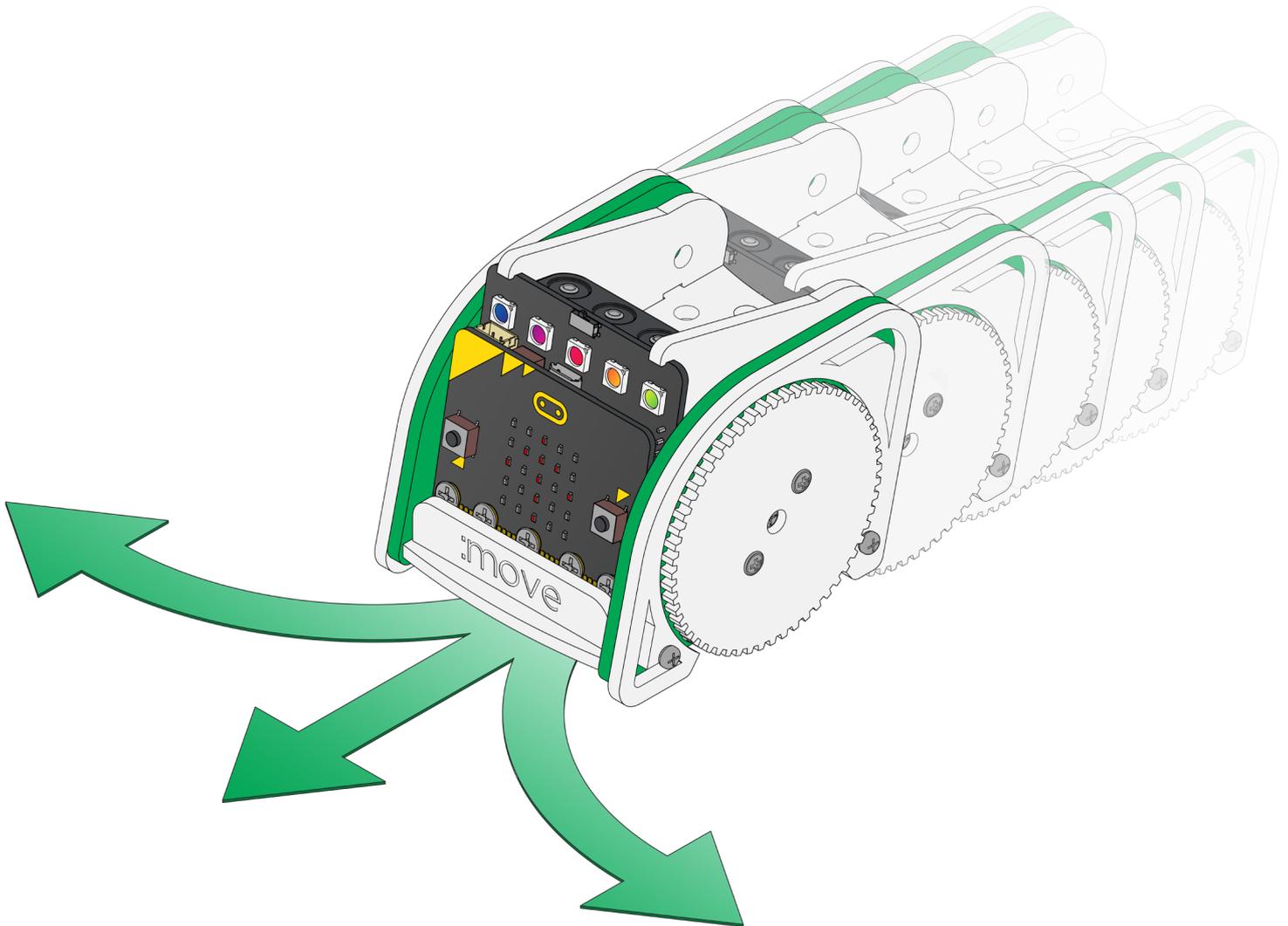


THE TEACHERS LESSON GUIDE TO THE :MOVE MINI



LESSON 2: DRAW SHAPES



This lesson includes curriculum mapping, practical challenges and a linked PowerPoint presentation.

www.kitronik.co.uk

TEACH YOUR STUDENTS HOW TO CONTROL A ROBOT WITH CODE!

DRAW SHAPES



This is the second lesson on robots for students in Key Stage 3 to robotics. The lesson involves a discussion about automation. Students will write their own algorithms on how to draw certain shapes using a robot. They will then code a shape in pairs using their algorithms as a guide. The teacher will then introduce functions as a more efficient way of drawing multiple shapes. Recommended ratio of students to robot is 6:1

Classroom setup

Students will be writing algorithms individually and coding in pairs. They will need:

- Pen & Paper (Each)
- A computer/laptop with a USB port and Internet access (shared between 2)

Position the assembled robots in the centre of each group of 6 students. Make sure there is no code on the micro:bits. A sample piece of code you could have:



The teacher will be writing on the board as well as demonstrating code on a projected board (if available)

Timings

The lesson is expected to take 1 hour. It can be shortened and lengthened if needed.

Suggested shortening:

Remove Challenge 2: Functions.

Suggested lengthening:

Challenge the students with more shapes, e.g. a star, a hexagon or allow them to come up with their own shapes.

Suggestions:

- A Christmas tree!
- The first letter of their name

Differentiation

For younger or less able students there is differentiation in the lesson plan.

The robot can be controlled using servo commands like "Servo write P1 to 120" OR it can be controlled using simple blocks like "Turn left 90 degrees". See www.kitronik.co.uk/blog/kitronik-custom-makecode-editor-servo-blocks for more details. All students will understand how robots move through the algorithm but only the most able/older students will code the servos directly.

KEY



Teacher asks this question to the class and discusses the answers with them.



Where this content maps onto the curriculum.



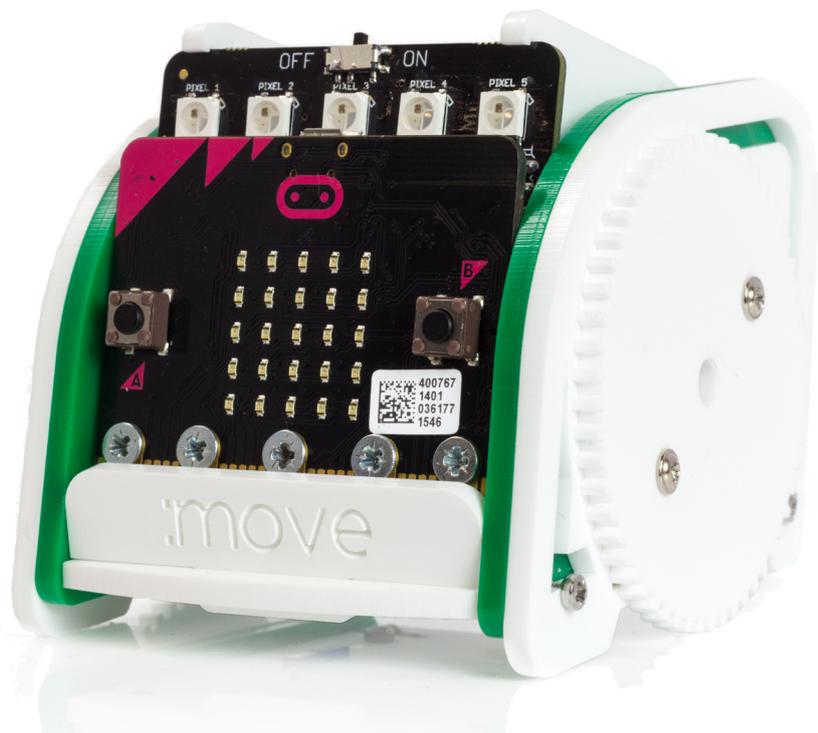
Teacher writes the text in this box on the board.



Students take notes, either on paper or in their electronic workbooks.



This part of the lesson aligns to Slide 1 of the attached PowerPoint.



DRAW SHAPES

INTRODUCTION



Teacher: Explain that you're going to make the robots draw shapes.



Why would we want a robot to follow a set pattern?

Class discussion on automation: What is it? How does it work? Have you seen real world examples? What do you think of it? Advantages and disadvantages of automation. Examples of automation: machines that build cars, automatic tractors that plough fields.



Curriculum mapping

Design, use and evaluate computational abstractions that model the state and behaviour of real world problems and physical systems.

Main Lesson

Challenge 1: Shape Algorithms



Teacher: Show a square. How do we explain to a computer how to draw a square?



What is the algorithm for drawing a square?



Students: On paper, in pairs: list the instructions you would give a computer to draw a square.

Students will come up with lots of different examples using phrases like "go forwards, go north, go straight". Stop them after a few minutes and set rules for the algorithm.

- The robot doesn't understand north (yet) or straight.
- Remember the algorithms from last week (check your notes) how does it go forwards?
- Be precise

Example algorithm from a student:

- Go forwards
- Turn right 90 degrees



DRAW SHAPES



Question: Do you notice a pattern? How can we make this algorithm more efficient?

Answer: The algorithm is basically:

Repeat 4 times:

Go forwards

Turn right



Ask students to write their algorithms again using the loop. Compare the number of lines from algorithm 1 to algorithm 2 and point out how much writing/coding we've saved ourselves by spotting a pattern and using a loop.

Write the best algorithm on the board.



Square

Repeat four times:

- Go forwards
- Turn right 90 degrees



Teacher: Next shape! Triangle.



What is the algorithm for drawing a triangle?



Students: On paper, list the instructions you would give a computer to draw a triangle.

Example algorithm:

Repeat 3 times:

- Go forwards
- Turn right 120 degrees

Teacher: Next shape! Rectangle.



What is the algorithm for drawing a rectangle?

DRAW SHAPES



Students: On paper, list the instructions you would give a computer to draw a rectangle.

Example algorithm:

Repeat 2 times:

- Go forwards for 100 distance
- Turn right 90 degrees
- Go forwards for 50 distance
- Turn right 90 degrees

Teacher: We've spent a long time writing algorithms. It's better to get the algorithm correct first than to continually write code, download it, test it, get it wrong and start again. There will be some testing needed but having a good algorithm will save time in the long run.

Let's get coding!



1. Open <http://makecode.com>
2. Select the micro:bit



Curriculum mapping

Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems.

Give each pair a shape and a button, e.g.

Pair 1: Button A: Square

Pair 2: Button B: Triangle

Pair 3: Button A&B: Rectangle

Show the students where all the blocks they need are and ask them to get coding. They should use the algorithms in their notes to guide them in creating the code.

DRAW SHAPES

Differentiation – higher ability

Servos work as follows:

The left motor is on Pin 2 – P2. The right motor is on Pin 1 – P1

- For Right, P1 to make it go forward you set it to any angle less than 90. The lesser, the faster.
- For Right, P1 to make it go backwards you set it to any angle greater than 90. The greater, the faster.
- For Left, P2 to make it go forward you set it to any angle greater than 90. The greater, the faster.
- For Left, P2 to make it go backwards you set it to any angle less than 90. The lesser, the faster.

Answers

Square:

```

on button A pressed
  repeat 4 times
    do
      drive forwards 50 distance
      turn right 90 degrees
  
```

Triangle:

```

on button B pressed
  repeat 3 times
    do
      drive forwards 50 distance
      turn right 120 degrees
  
```

Rectangle:

```

on button A+B pressed
  repeat 2 times
    do
      drive forwards 100 distance
      turn right 90 degrees
      drive forwards 50 distance
      turn right 90 degrees
  
```

AUTOMATION

Differentiated Answers



From the last lesson students could code the servos directly to move the robot. This is for the high ability/ older students to try. The timings will need changing on each individual robot, e.g. on the code below turning right for half a second at top speed is about 90 degrees. It might take multiple attempts at coding, downloading and testing to get a decent shape.

```

on button A pressed
  repeat 4 times
    do
      servo write pin P2 to 180
      servo write pin P1 to 0
      pause (ms) 3000
      servo write pin P1 to 90
      servo write pin P2 to 90
      pause (ms) 1000
      servo write pin P2 to 0
      servo write pin P1 to 90
      pause (ms) 500
      servo write pin P1 to 90
  
```

Ask all the students coding squares to download their code. Place the robots on the floor and ask:



1. What's it going to do?
2. Press A.
3. Did it do what you expected? Whose was the best square? What was their code?
4. Can we improve this code?



Students: On paper, list the instructions you would give a computer to draw a rectangle.

Repeat for the groups who created the triangle and the rectangle.

AUTOMATION

Challenge 2: Functions



Question: I want to draw 2 squares and a triangle all at once. How can I do this?

Student Answers:

- Press A, A then B
- Change the code for Button A to draw 2 squares and a triangle

Your Answer:

- A more efficient way is to create a function called drawSquare, drawTriangle and call them.

Describe what a function is and why programmers use them.



Definition: A function is a named section of a program that performs a specific task.

Why use it: When you do the same tasks over and over again it is more efficient to call a function that stores that task than repeat the code. Functions make our code shorter, neater and more efficient.

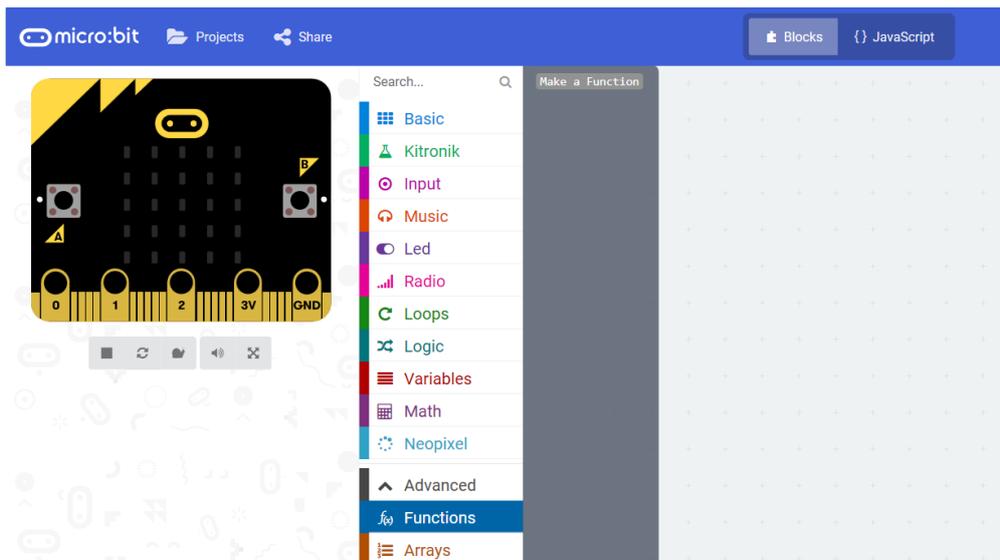


Curriculum mapping

Design and develop modular programs that use procedures or functions.

Demonstrate how to create a function in MakeCode:

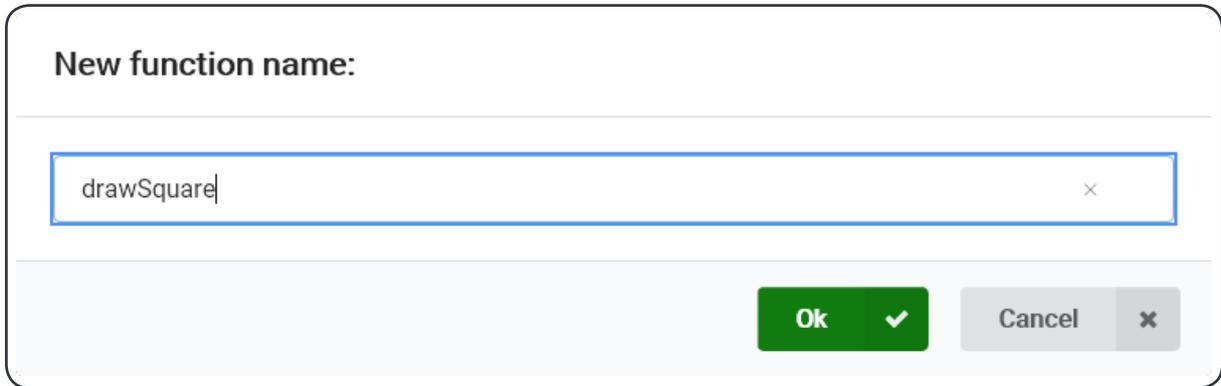
1. Select Advanced
2. Select Functions
3. Select Make a Function



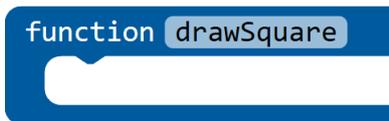
Continued on next page >

DRAW SHAPES

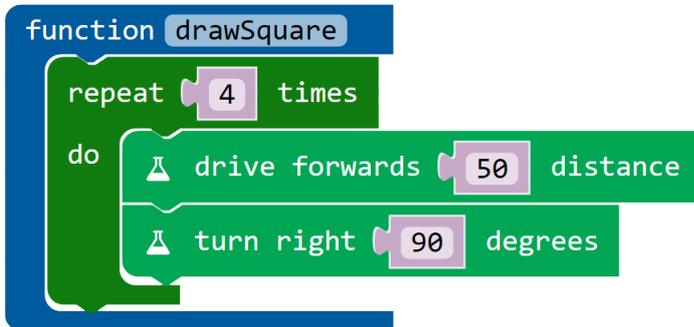
4. Name the Function "drawSquare"
5. Select OK



6. On the screen the new function will appear:



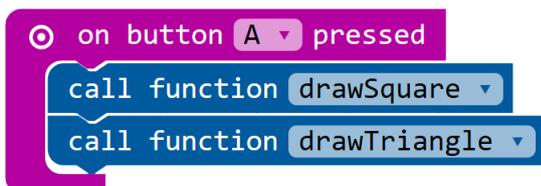
7. Drag the blocks from A button press to go under this function, e.g.



8. Repeat the same steps for drawTriangle, dragging the blocks from B button press.

Now when they press button A you just call the functions:

9. Select Advanced
10. Select Functions
11. Drag the block "call function drawSquare" and place it under on button A pressed
12. Repeat for "call function drawTriangle" until you have the following:



Continued on next page >

DRAW SHAPES

Ask the students to repeat your steps to create functions of their own.

This code will not work as some shapes will overlap. Encourage the students to think about spacing between shapes. If there's time, download and test the code.



Evidence

In this lesson the students will each have written down algorithms in their notes.

With one robot students will be waiting between each task. As each group finishes a task ask them to take a screenshot of their code and paste it into a document. Ask them to comment the code with labels.

Summary



Students modelled automation in a robot. They created automatic robots that follow a set pattern, wrote the algorithms for these set patterns and coded the robots to draw these shapes. They learnt what a function is and why would you use functions and how to create and call a function in code.



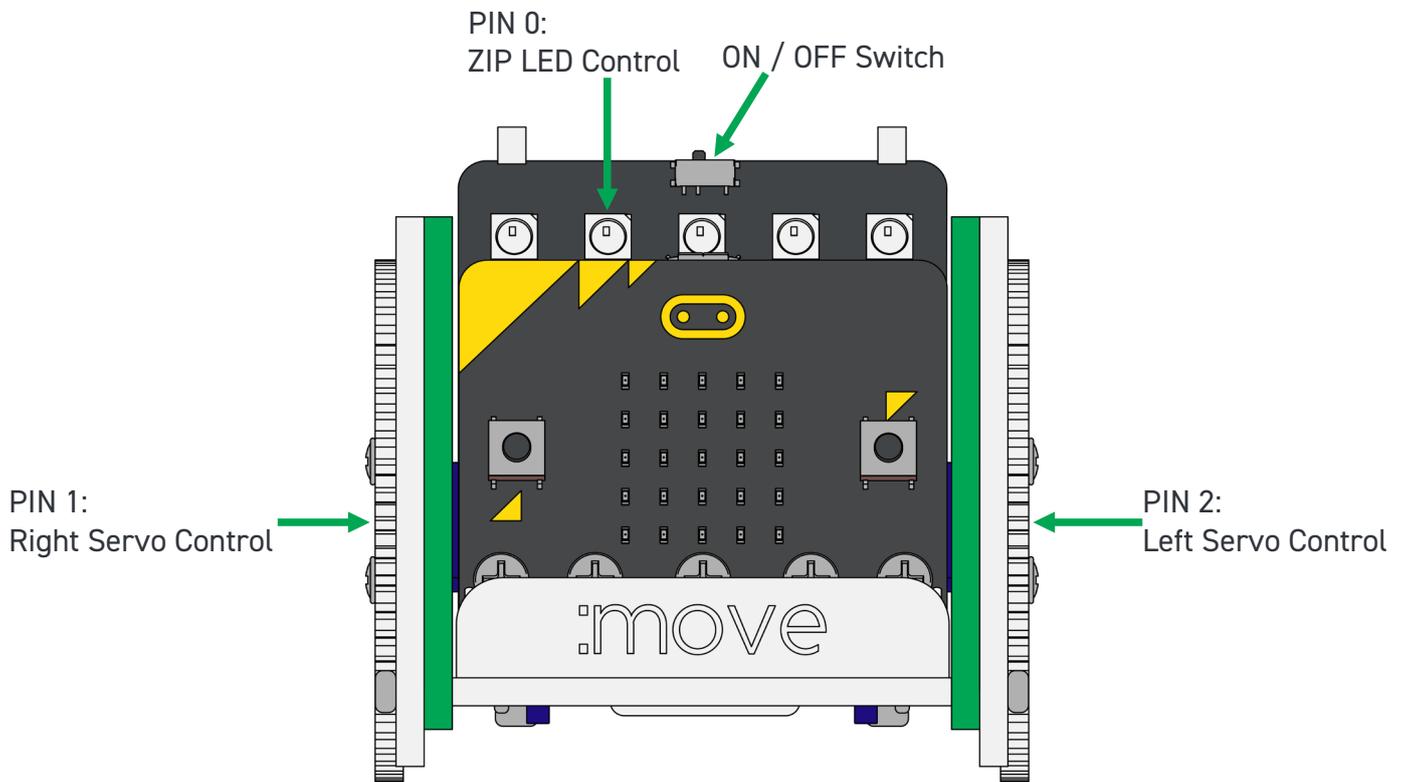
This is the second lesson on robots for students in **Key Stage 3** for robotics with the **Kitronik :MOVE mini** for **BBC micro:bit**. The lesson involves a discussion about automation. Students will write their own algorithms on how to draw certain shapes using a robot. They will then code a shape in pairs using their algorithms as a guide. The teacher will then introduce functions as a more efficient way of drawing multiple shapes. The recommended ratio of students to robot is 6:1. Please find additional lesson plans, accompanying **PowerPoint** presentations and more at www.kitronik.co.uk/5624.

The Kitronik :MOVE mini buggy kit for the BBC micro:bit provides a fun introduction to the world of robotics. The :MOVE mini is a 2 wheeled robot that is suitable for autonomous operation, remote control projects and more. Once built it can be coded for a variety of activities. A range of add on boards can expand the capabilities to include more advanced functionality.



LESSON REQUIRES:

- Pen & Paper
- A computer/laptop with a USB port and Internet access
- A :MOVE mini
- A BBC micro:bit
- 3 x AAA batteries (included with :MOVE mini)
- A micro USB cable



WARNING : Contents may inspire creativity

T: 0845 8380781

W: www.kitronik.co.uk

E: support@kitronik.co.uk

🇬🇧 Designed & manufactured
in the UK by



[kitronik.co.uk/twitter](https://twitter.com/kitronik)



[kitronik.co.uk/facebook](https://www.facebook.com/kitronik)



[kitronik.co.uk/youtube](https://www.youtube.com/kitronik)



[kitronik.co.uk/google](https://www.google.com/kitronik)