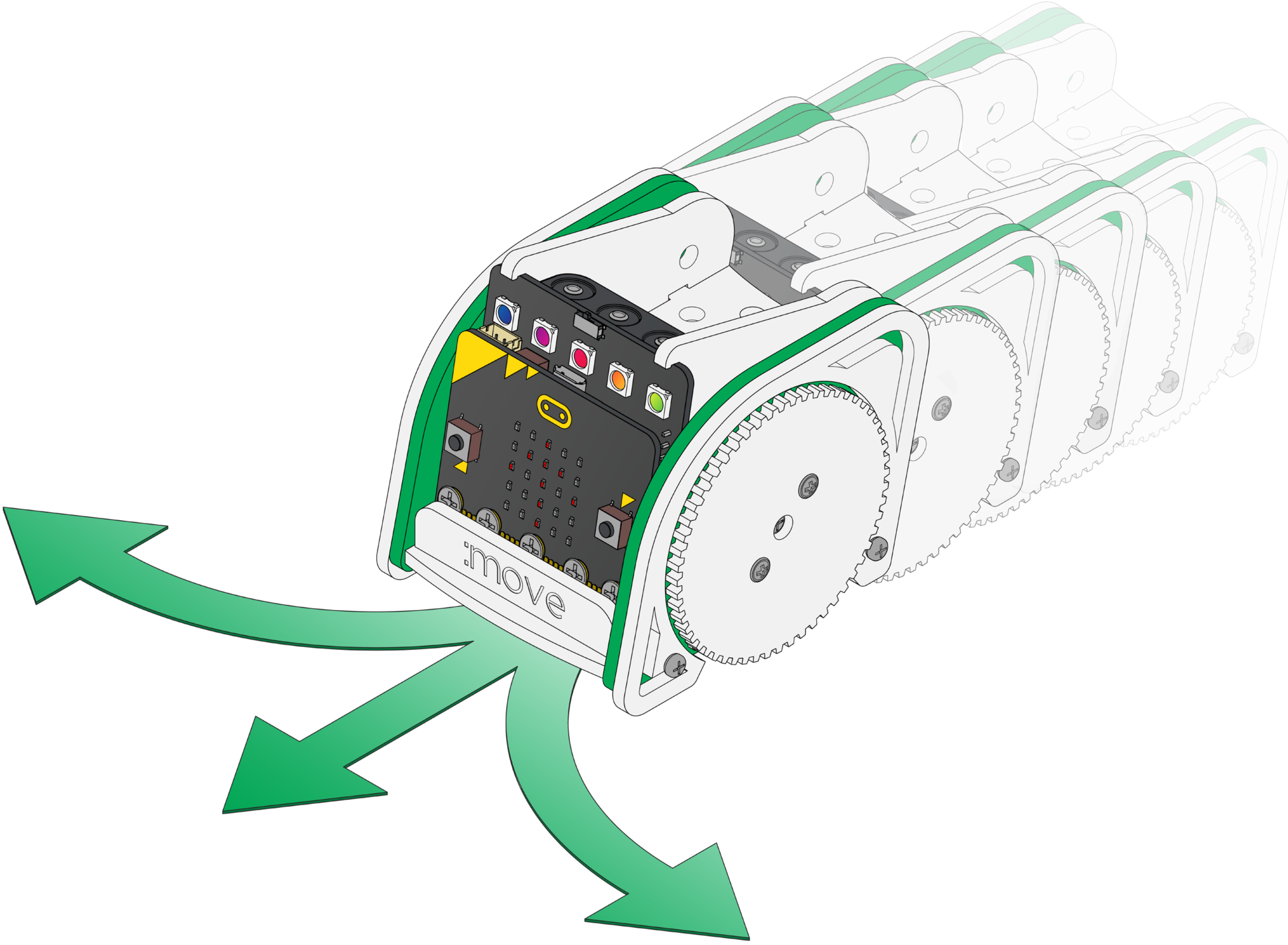
THE TEACHERS LESSON GUIDE TO THE :MOVE MINI



LESSON 3:
SIMPLE AUTONOMY

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!

LESSON 3

SIMPLE AUTONOMY

This is the third lesson on robots for students in Key Stage 3. The lesson involves a discussion about autonomous robots. Students will write algorithms to make a robot move towards a compass direction then how to control the robot using a line follower. They will then code the robots to move around a map on the ground. Recommended ratio of students to robot is 6:1

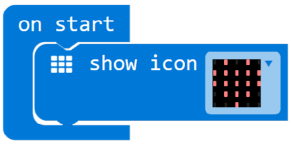
Classroom setup

Students will work be working in pairs. They will need

- Pen
- Paper
- A computer/laptop with a USB port and Internet access
- You will need a few arenas for the students to drive around.

Position the assembled robots in the centre of each group of 6 students. Make sure there is no code on the micro:bits. Each robot will need a line follower attached.

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 1: Compass.

Suggested lengthening:

Challenge the students to try out the advanced code included in the answers.

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.

LESSON 3

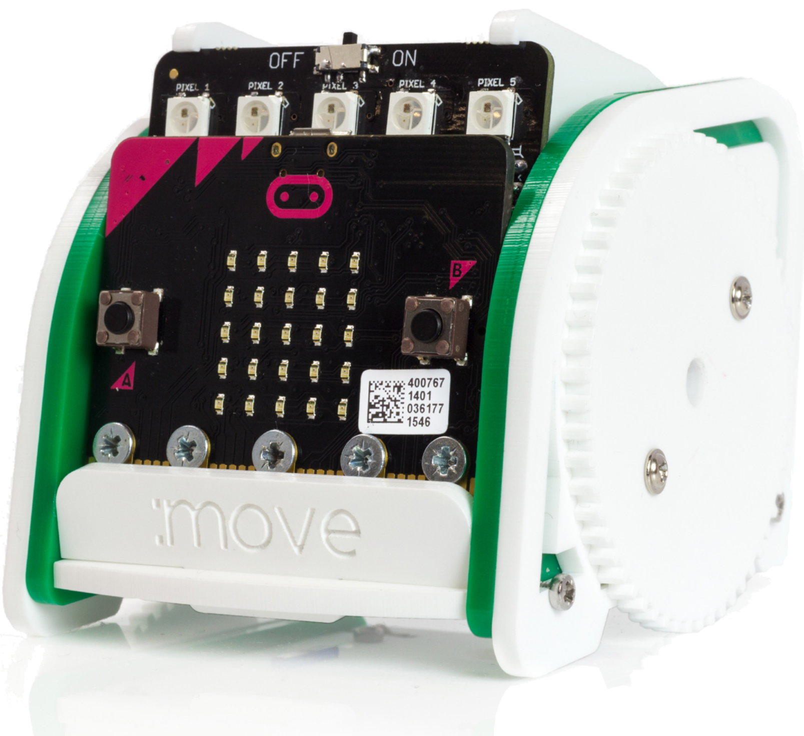
KEY

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.

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



LESSON 3

SIMPLE AUTONOMY

INTRODUCTION

Teacher: Last week we drew shapes with the robots by programming set shapes. The real world is not a square! Or a triangle. Let's think about a tractor ploughing a (misshapen) field. How else could we program the robot?

Question: How do we code a robot when the shape is not known?
Possible answers: Obstacle avoiders, line followers, using compass bearings.

Class discussion on autonomous cars. What are they? How do they work? Have you seen real world examples? What do you think of them?

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: Compass bearing
Teacher: Draw a circle on the board with the key points – 360/0, 90, 180, 270

Question: What is North?
Answer: 360 or 0.
Question: That's too precise! What is North?
Answer: Between two numbers, e.g. 340 and 20.

Write the algorithm for determining North with the help of students.

North
If compass is greater than 340 or less than 20.

Question: Why isn't the algorithm -
If compass is greater than 340 AND less than 20?
Answer: There is no number greater than 340 and less than 20.

Opportunity to discuss the difference between AND and OR in Computing.
So we want the robot to head North. We'll need to scan for North.

Kitronik

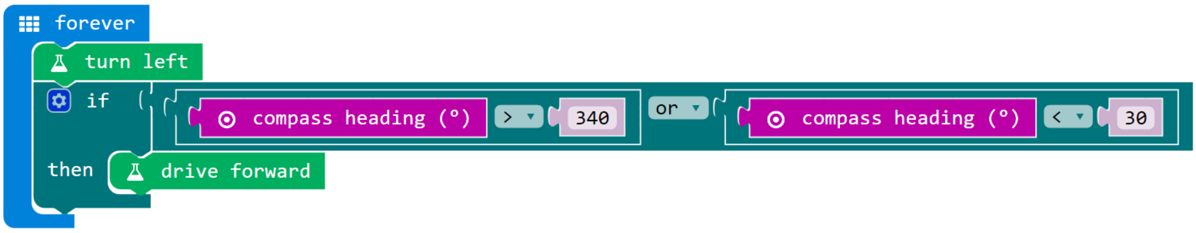
LESSON 3

SIMPLE AUTONOMY

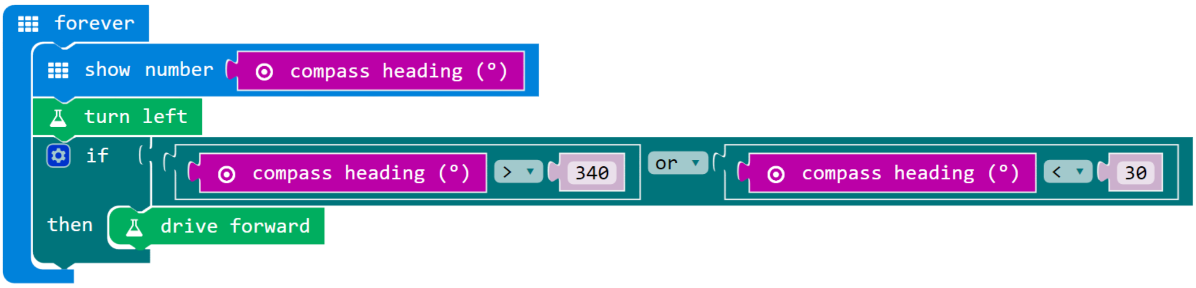
Loop Forever:
Turn Left
If compass heading greater than 340 OR compass heading less than 20
Then
Go Forwards

Doing challenge 1 and challenge 2 in one lesson may be a tight squeeze. Perhaps demo just one robot following the compass direction?

Suggested answer:



It might be a good idea to show the compass heading so the student knows how far they're off!



Challenge 2: Line Follower

The hedge problem is perfect for a line follower. Explain how the line follower works:

It has two sensors underneath: one left and one right. These sensors are detecting the reflectance of the surface beneath them. Black absorbs light so it's not very reflective. Therefore these sensors can tell the difference between a black line and a white surface. Using that idea we can get the robots to follow a line.

We want the robot to travel with the line underneath it. If the left sensor senses the line, we want to move the robot right. If the right sensor detects a line, want the robot to move left. If neither sensor are detecting the line (because it's directly underneath) – go forwards.

Question: When the robot is going forward or backwards, when does it stop?
Answer: Never!

LESSON 3

SIMPLE AUTONOMY

Students: In pairs, how would we use these two sensors to get the robot to follow a line?

Line Following
As a class come up with the best algorithm for using the line following sensors.

Sample algorithm
Forever Loop:
If left sensor is true AND the right sensor is false
Turn left
If right sensor is true AND the left sensor is false
Turn right
If left sensor is true AND the right sensor is true
Go Forwards

Let's get coding

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

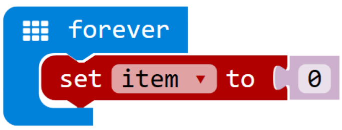
Curriculum mapping
Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems.

The students need to setup the code to use the line follower. The line follower sensors are attached to Pins 15 and 16 on the micro:bit

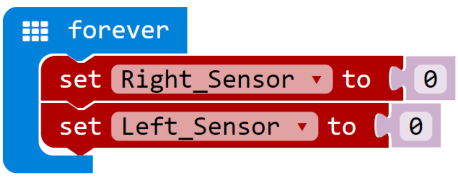
1. Create a variable called Left_Sensor
   a. Variables > Make a Variable > Type in Left_Sensor, Press OK
2. Create a variable called Right_Sensor
   a. Variables > Make a Variable > Type in Right_Sensor, Press OK

We want to read the values of both sensors constantly

3. Select Variables > set item to 0 and place it under a forever loop



4. Change item to Right_Sensor
5. Do the same for Left_Sensor

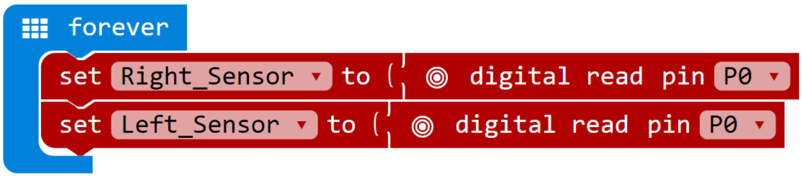


Continued on next page ›

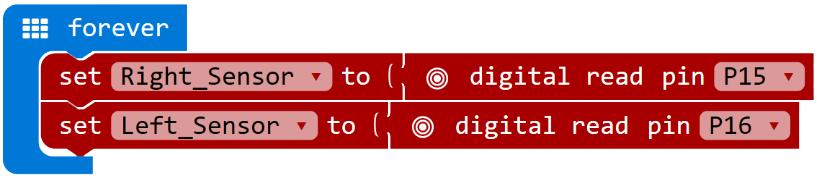
LESSON 3

SIMPLE AUTONOMY

6. Select Advanced > Pins and drag out “digital read pin P0” and place it over the 0
7. Do the same for the second 0



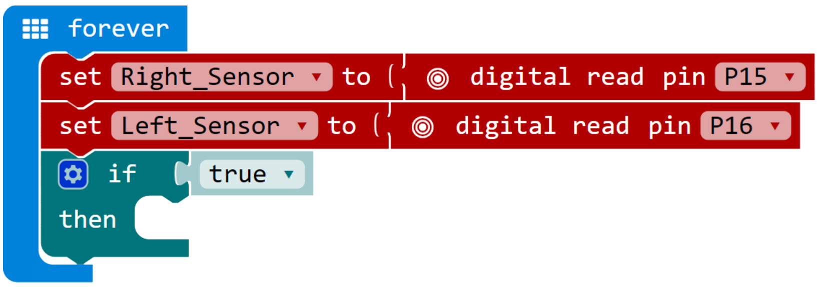
8. The right sensor is connected to Pin 15. Change P0 to P15 on the Right_Sensor
9. The left sensor is connected to Pin 15. Change P0 to P16 on the Left_Sensor

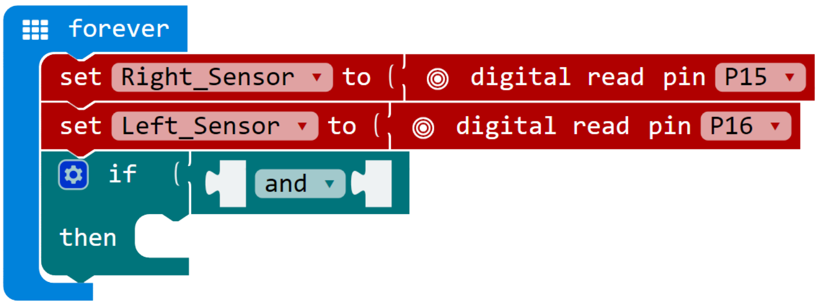


Students can now start trying to create their code using the algorithm on the board
For the sensors they return 0 when they see a line and 1 when they see paper. This changes the algorithm to:
Forever Loop:
If left sensor = 0 AND the right sensor = 1
Turn left
If right sensor = 0 AND the left sensor = 1
Turn right
If left sensor = 0 AND the right sensor = 0
Go Forwards

Suggested answers:

The students need to build up the logic statements using blocks. From the Logic menu:

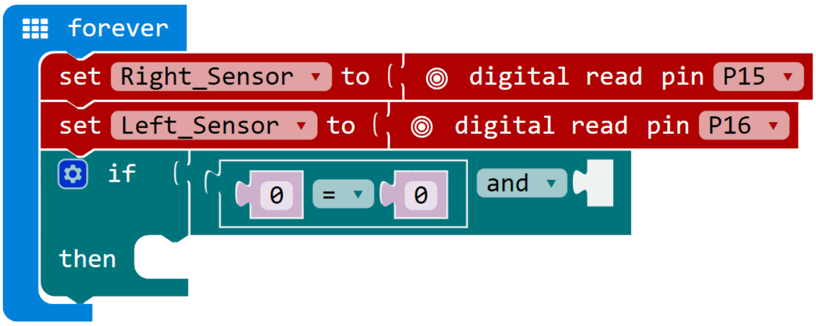


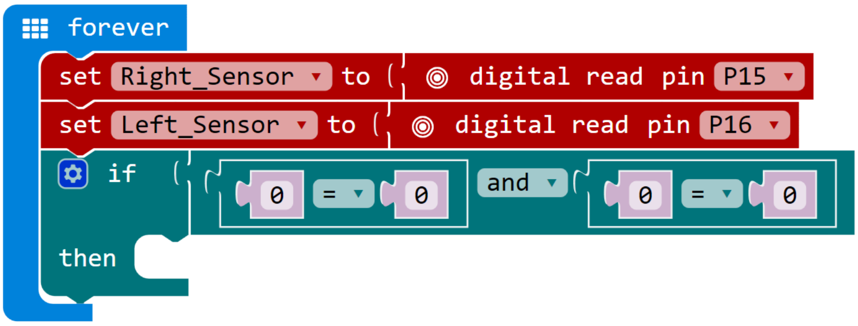


Continued on next page ›

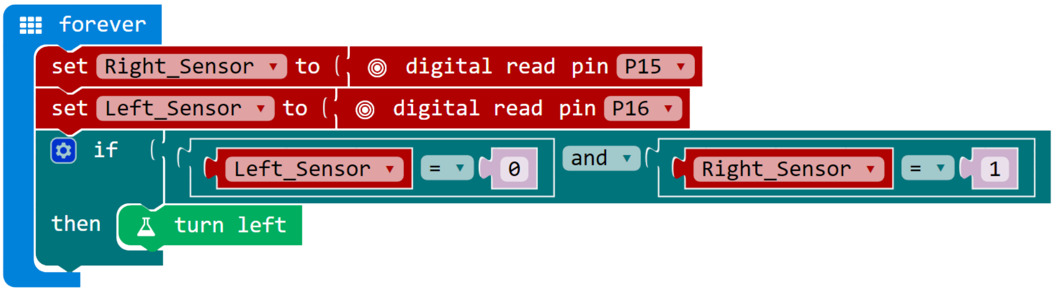
LESSON 3

SIMPLE AUTONOMY





From variables drag out Left_Sensor and Right_Sensor.



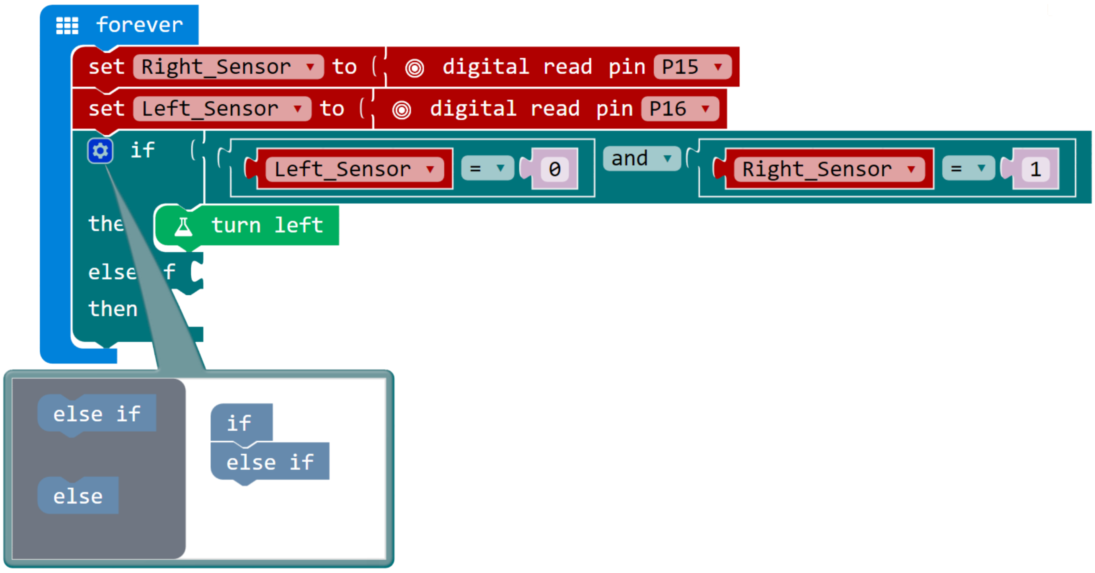
Continued on next page ›

LESSON 3

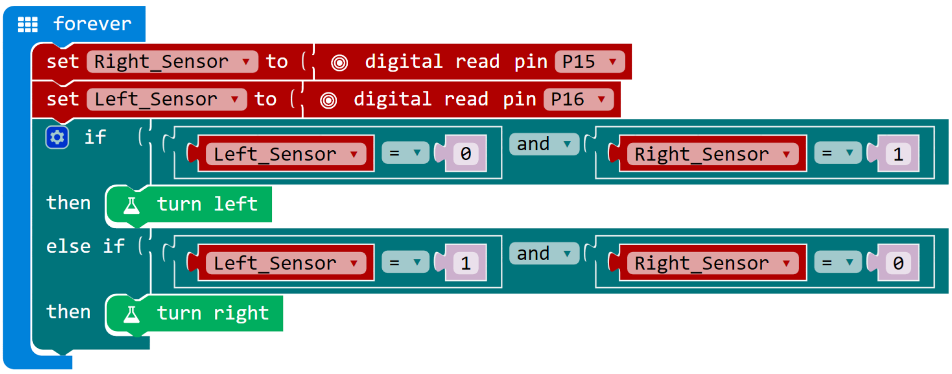
SIMPLE AUTONOMY

Advanced

In the example above, say the robot does turn left. We now don't need to check if we need to turn right, because we've just turned left. We should finish the loop and go back to the start.
The next statement should be else if. Select blue cog next to if and drag "else if" into the statement to create:



Then create the next statement:



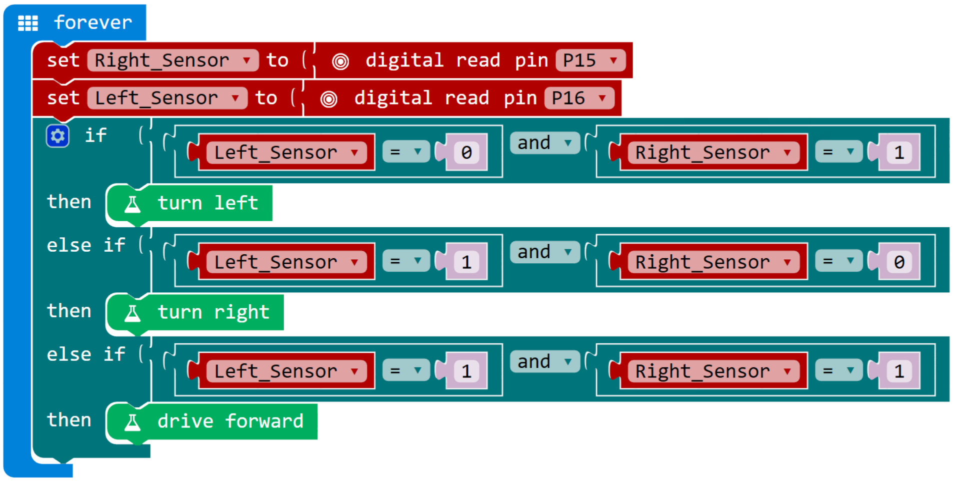
Continued on next page ›

Kitronik

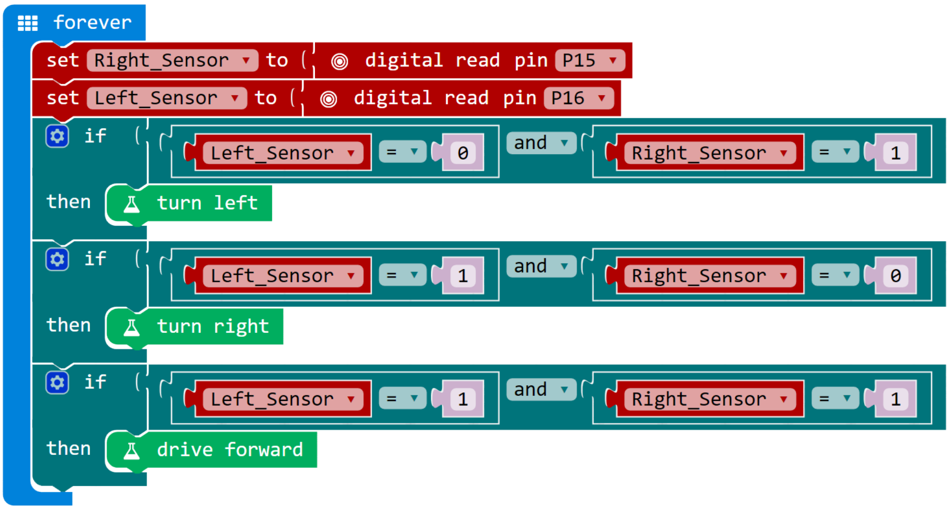
LESSON 3

SIMPLE AUTONOMY

The final advanced answer:

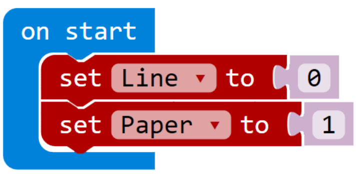


A simpler answer (which is valid)



Advanced 2

1s and 0s can confuse students!
On start you could create a variable called line and set it to 0 and a variable called paper and set it to 1 like so:



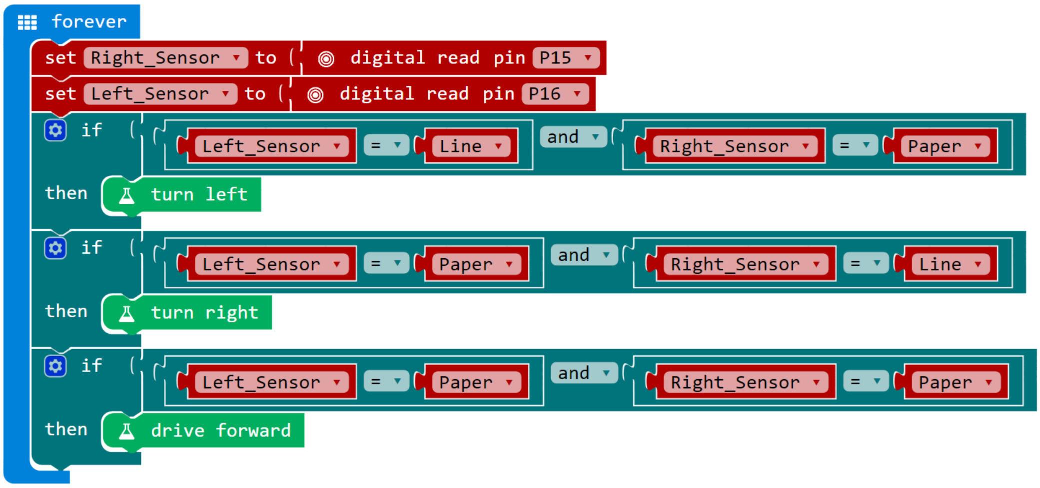
Continued on next page ›

LESSON 3

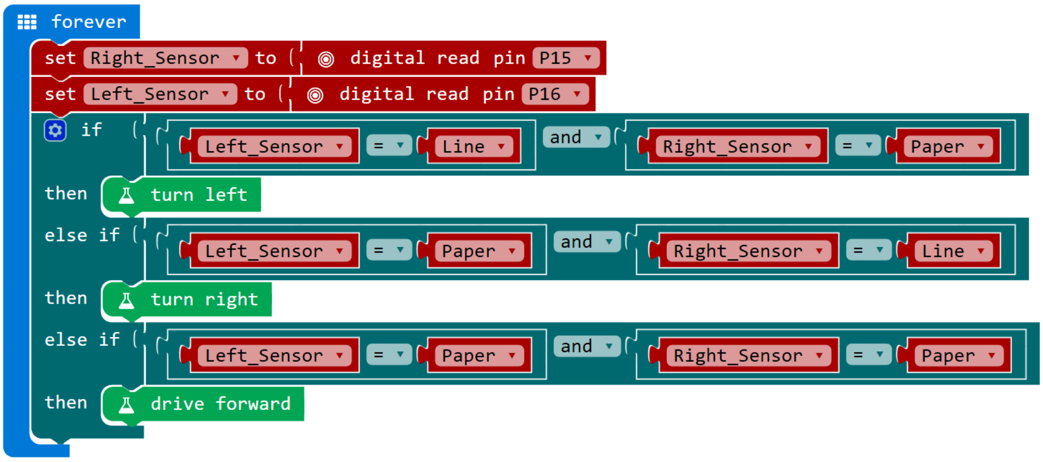
SIMPLE AUTONOMY

Then replace them in the main code:

Basic



Advanced



Ask the students to practice their line following robots on some lines!

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

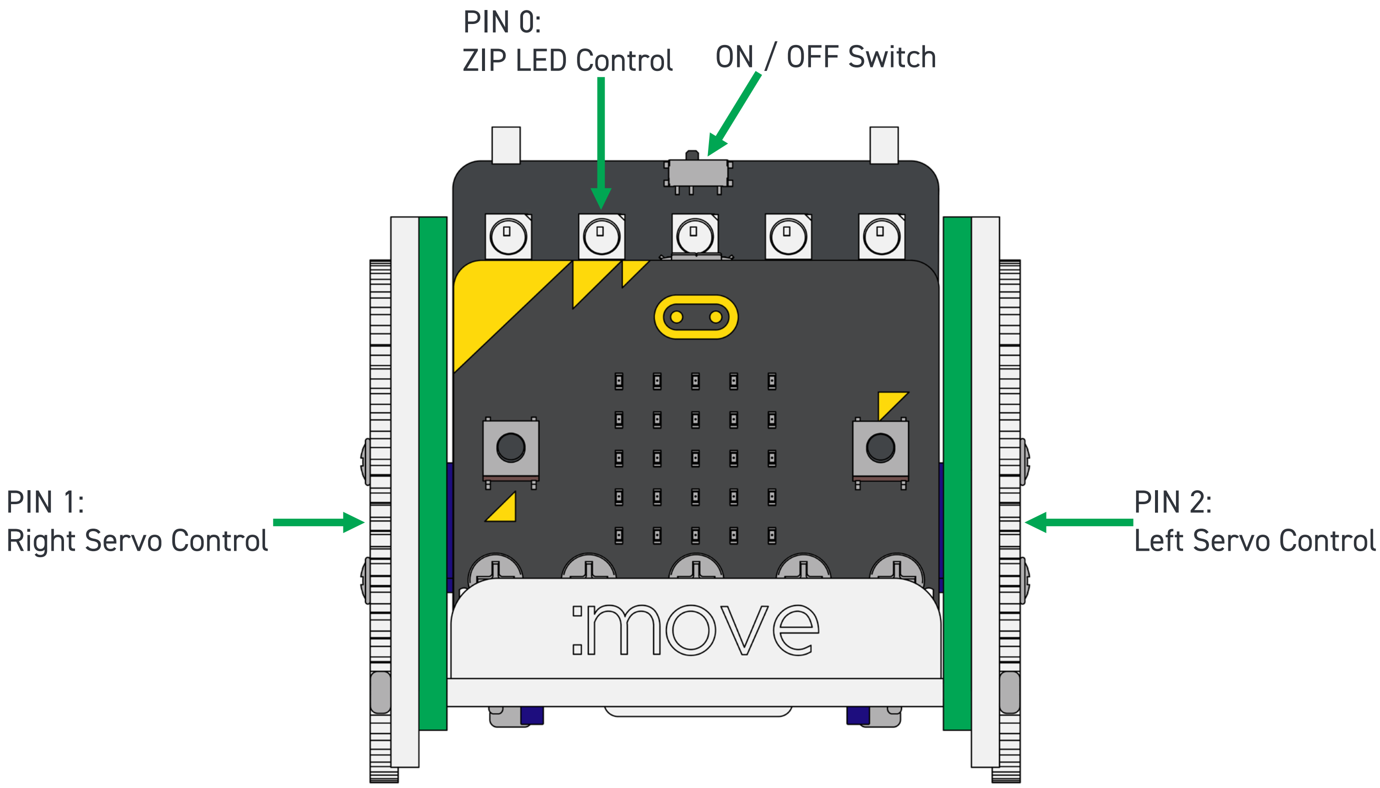
Students have learnt more ways that robots can be controlled. They have written algorithms about how a compass can be used to drive a robot and have learnt how line followers work. They wrote an algorithm using the idea of line followers and coded and tested their algorithms.

This is the third 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 autonomous robots. Students will write algorithms to make a robot move towards a compass direction then how to control the robot using a line follower. They will then code the robots to move around a map on the ground. 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



T: 0845 8380781
W: www.kitronik.co.uk
E: support@kitronik.co.uk

Designed & manufactured in the UK by Kitronik

kitronik.co.uk/twitter
kitronik.co.uk/facebook
kitronik.co.uk/youtube
kitronik.co.uk/google