

Lego Mindstorms Colour Sensor Workshop

Teacher Pack

Produced by Thomas Dack (2018) Updated by Daniel Westwood (2024)

[thomas.dack@stfc.ac.uk](mailto:thomas.dack@stfc.ac.uk) [daniel.westwood@stfc.ac.uk](mailto:daniel.westwood@stfc.ac.uk)

This workshop is designed to introduce the concepts of robotics and computer programming through the use of Lego Mindstorms robots. The activity consists of an introductory presentation and three programming tasks.

Each task worksheet has student instructions as well as the pseudocode for the logic needed to write the model solutions. Model solutions for each task can be found within this pack.

Provisional timings for a typical 50 minute session are:

**Introductory Talk & Presentation:** 15 minutes

**Task 1: Stop at Line:** 5 minutes

**Task 2: Line Counting:** 15 minutes

**Task 3: Line Following:** 15 minutes

**Debrief & Wrap Up:** 5 minutes

The curiculum links for this activity are as follows:

KS2:

* design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
* use sequence, selection, and repetition in programs; work with variables and various forms of input and output
* use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

KS3:

* design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
* use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
* understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems

**Task 1: Stop at Line**

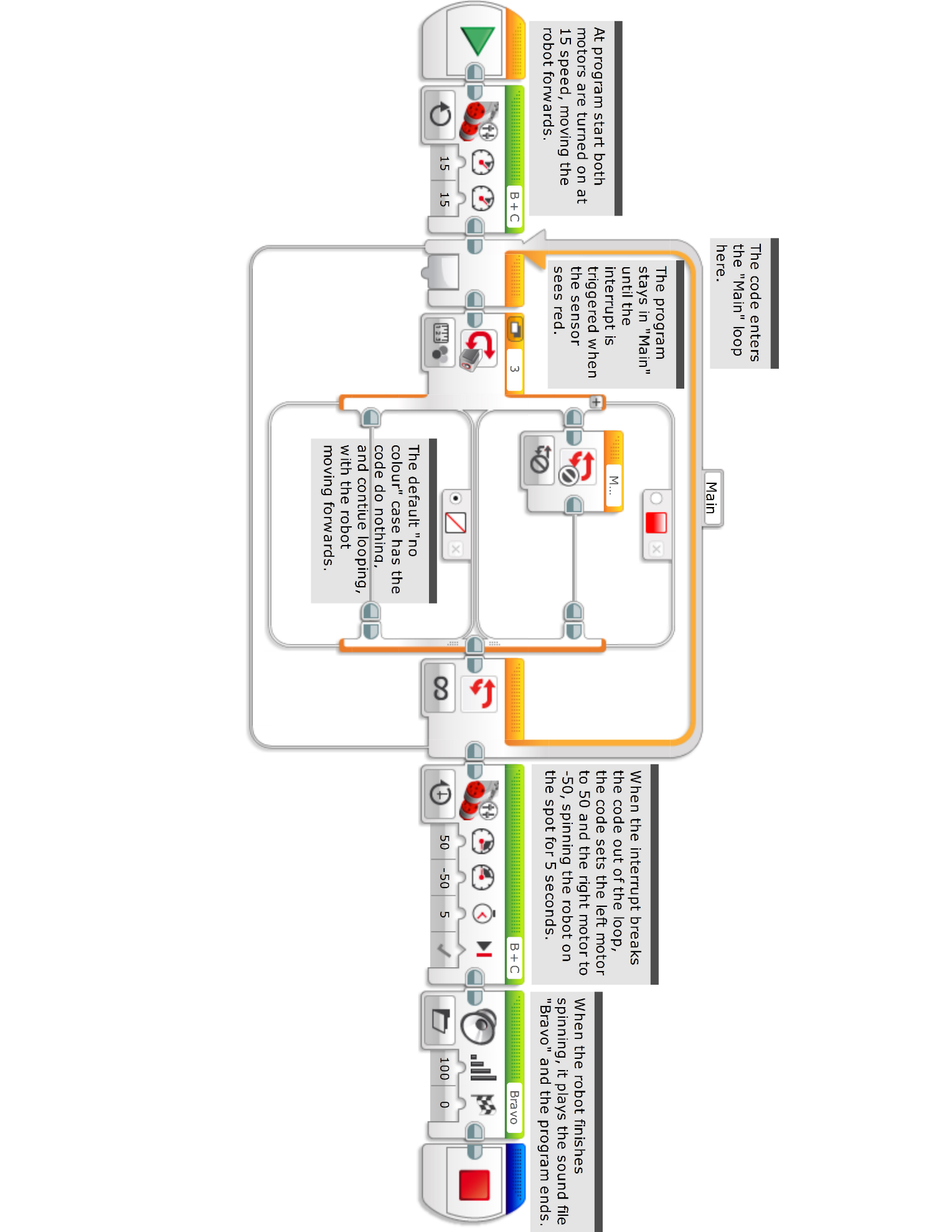
This task introduces students to the Mindstorms colour sensor by having them drive forward until the robot sees a red line.

The worksheet for this task is:

A white rectangular object with red and green lines

Description automatically generated

Hint text: To achieve this, you want the robot to have its motors on while it cannot see the red line, and then turn them off when it does.

A model solution for Task 1 would be:

**Task 2: Line Counting**

This task introduces the concept of variables, as the students must store the number of lines the robot passes over.

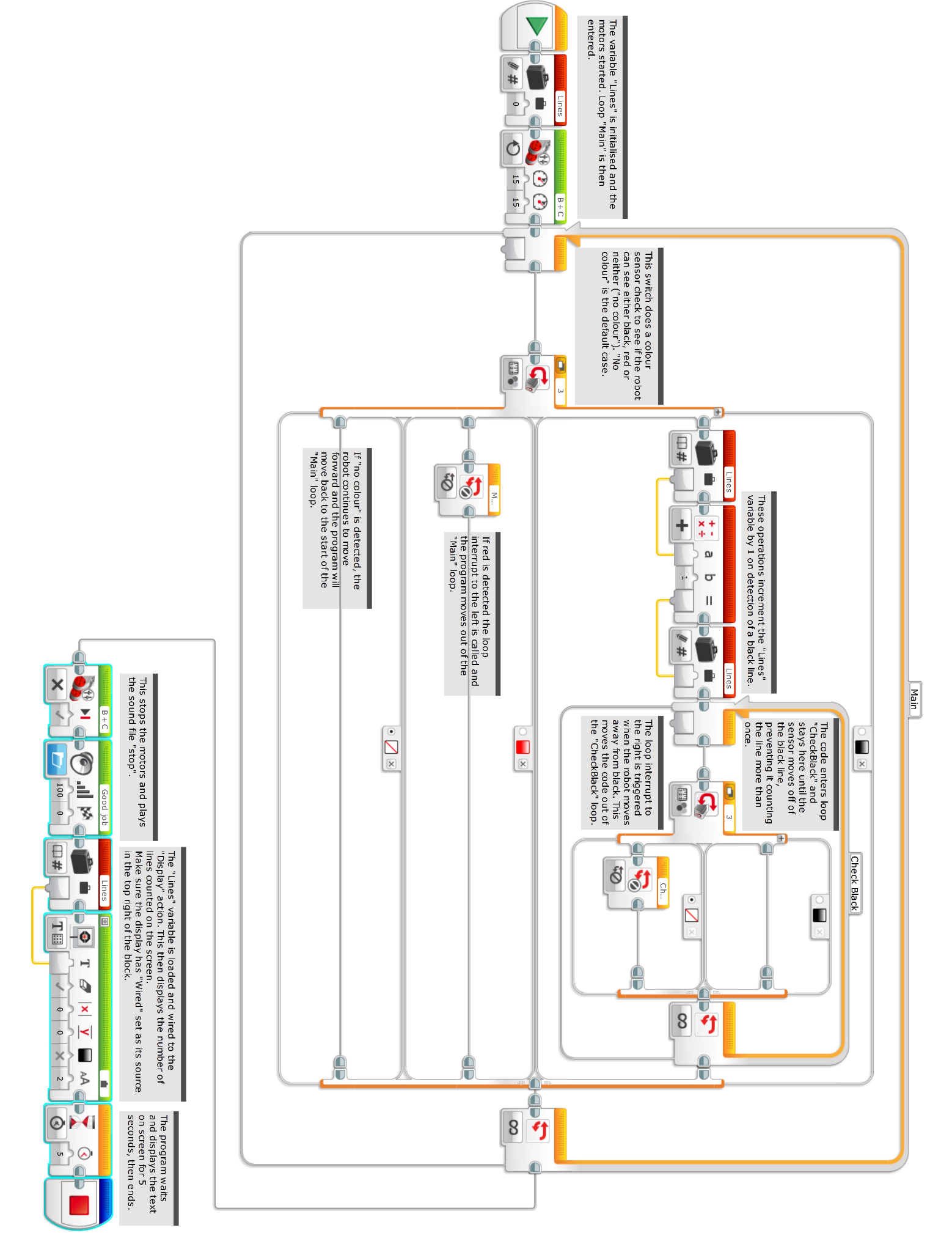
With this exercise one potential error is that the robot counts each line more than once. This is due to the program progressing through the loop body multiple times before the sensor moves away from the black line. One “brute-force” solution is to have the program sleep after it detects black, so that it has time to move away from the line. A better solution is depicted in the model solution for this task, where the code waits until it detects white again.

The worksheet for this task is:

A screen shot of a computer screen

Description automatically generated

Hint text: In order to count lines, your robot will need to store a number in its memory - much like you may remember a secret number!

A model solution for this would be:

**Task 3: Line Following**

This task has students program a robot to follow a line. The coding solution for this task will likely be easier than Task 2, however the logic required for the solution will require more thought. A line following robot will follow the edge of a line, and so the model solution has the robot turn left if it sees black (towards the left edge) and turn right if it does not (returning it towards the left edge).

The worksheet for this task is:

A black and green line on a white background

Description automatically generated

Hint text: It may be easier to think of the robot following the left edge of the line: if it sees black, how will it find the edge? What if it sees white?

A model solution for this would be:

