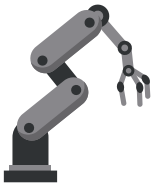
CUBOT Moodle Year Course

LESSON PLAN

TITLE:	**Introduction to Line Following**
LESSON:	**8 / 32**
DURATION:	1h
TOPICS INTRODUCED:	Line following with the Line Following Module.



INTRODUCTION:

This lesson introduces line following with the Line Following Module (LFM). The students will learn how line following works, then print a line for the robot to follow and do some line following themselves.

Note: This lesson requires an active internet connection for Arduino firmware updates.

IN THIS LESSON:

- We will look at how to attach the line following module to the robot.
- The students will learn how line following works.
- We will look at optimisation and why it matters.
- We will reinforce the importance of frequent testing.

Below, there is a *theory overview* that covers the theoretical questions asked in this lesson's presentation. The trainer is required to familiarise themselves with the presentation content before teaching the class.

RESOURCES REQUIRED:

1. CUBOT and Arduino UNO Kit
2. Laptop and USB B – A Cable
3. Printer
4. A mat with a line on it (print-out line included)*
5. Optional: Black electrical tape

LESSON STRUCTURE:

1. Presentation | **5 mins**
2. Activity: Following the Line | **45 mins**
3. Quiz | **10 mins**

EXTRA RESOURCES:

Watch: A line following competition at the University of Manchester

LESSON 8
THEORY OVERVIEW:

- **Line following robots are robots that follow a line autonomously (without any help).**
  In this lesson we're going to learn how to turn the CUBOT into a line following robot.

- **Our CUBOT comes with a line following sensor.**
  This sensor allows us to program the robot to follow a line.

- **What's inside?**
  Inside is an LDR (light dependant resistor), and a blue LED. This allows us to measure reflected brightness, which will be helpful to tell if the robot is over a black line or not.

- **How does it work?**
  We could say that the robot should drive straight when the line following module detects black, and turns left when it detects white. This would work if the line is curving to the right, but not if the line curves to the left.

  The solution to this is: Instead of following the line, we can follow the edge of the line. The pseudocode would be as such:

  "When the LFM detects black, turn left"
  "When the LFM detects white, turn right"

- **Remember, you can follow the *left* or the *right* side of the line.**
  Sometimes one side of the line would be easier to follow.
  To follow the other side of the line, you will need to flip your logic:

  "When the LFM "sees" black, turn **right**" (instead of left)
  "When the LFM "sees" white, turn **left**" (instead of right)

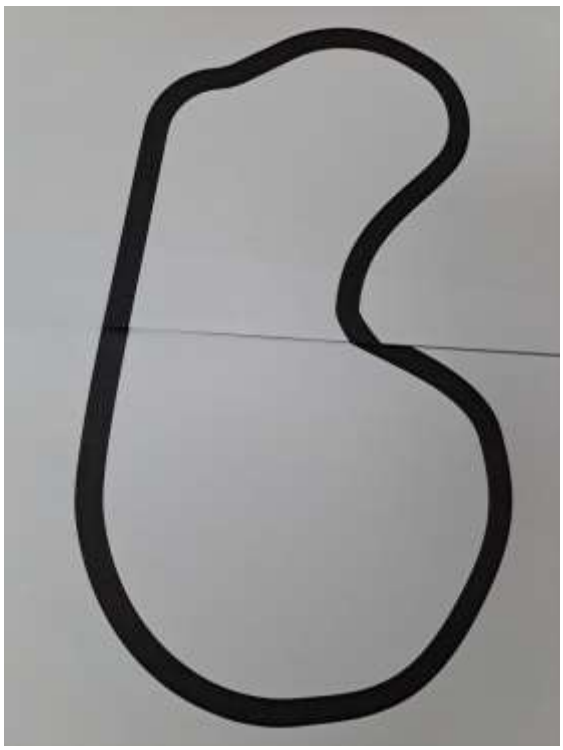
- **What is optimisation?**
  Optimisation is the process of making improvements to a system to get a better or more efficient result.

  When you first program your robot to follow a line, it won't go as fast as it could.
  You will need to optimise your code to get it to drive better.
  While doing this, you need to test your robot as often as you can.

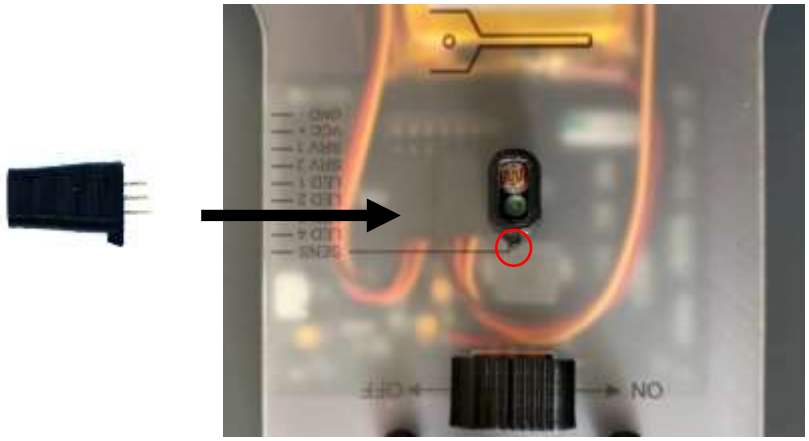
- **Activity Guide – Program the Robot to Follow a Line**

In this activity, you're going to program your robot to follow a line.

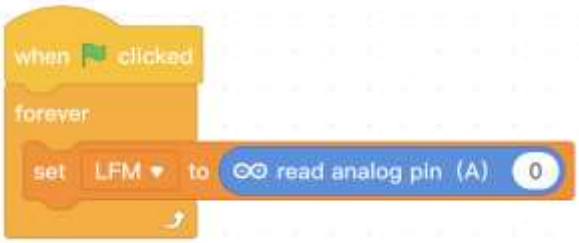
1. Print out the line for your robot to follow on 2x A4 sheets of paper. You will need to tape these to a table. Optionally, you may create your own line using black electrical tape and a light surface/tabletop



2. **Connect your LFM to your robot** with the arrow facing the **front** of the robot.

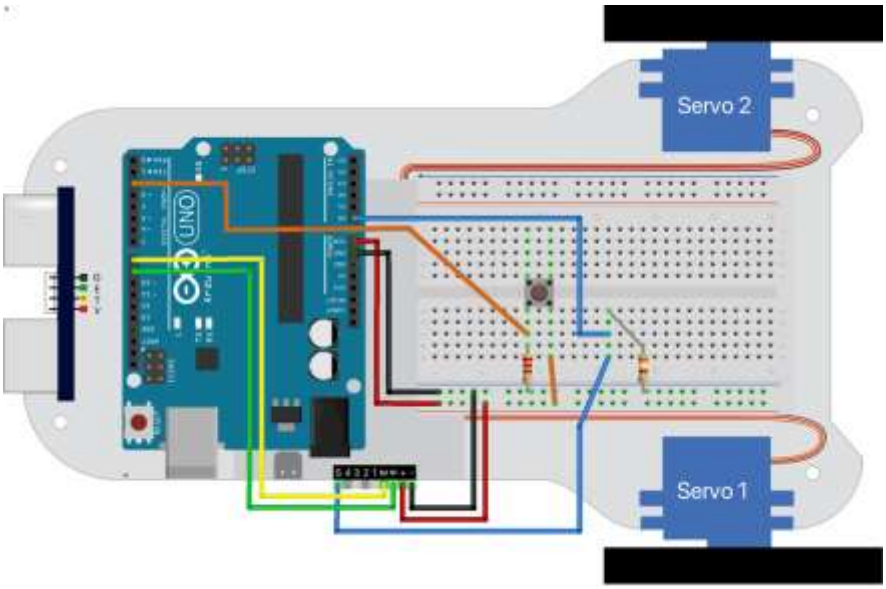


3. **Switch to Live mode** and upload the following code to your robot. Take note of the largest number value and the smallest number value. These values will change depending on the lighting of a room.
   Note: for Arduino, the students are to open the serial monitor to get these values.



4. **We will now use these two numbers to determine the middle value:**
   Add the numbers together and divide by 2.

5. **Build the following circuit:**



6. **Switch back to upload mode and upload the following code to your robot.**
   Place the value obtained in step 3 where the arrow is pointing below.

When you press the pushbutton, the program will start. To stop the robot, press the **red button** on the Arduino UNO.

What's happening?

When the value is greater than your set value, the robot knows the LFM is situated over white, so it will turn in one direction. When the value is smaller than your set value, the robot knows that the LFM is situated over black, so it will turn in the other direction.

7. **Tweak the servo values to optimise your code.**
   The key to get your robot to drive better is to test as often as you can.

   Hint: If both wheels are turning forward, the robot will move forward faster.

- Troubleshooting
  - Sometimes you will get an upload error. You may need to:
  - Disconnect and reconnect the Arduino.
  - Select a different port from the programming environment.
  - Try a different USB port