Ri Engineering Masterclass 2021

Robotics – Moving Intelligently

WMG
THE UNIVERSITY OF WARWICK

CATAPULT
High Value Manufacturing

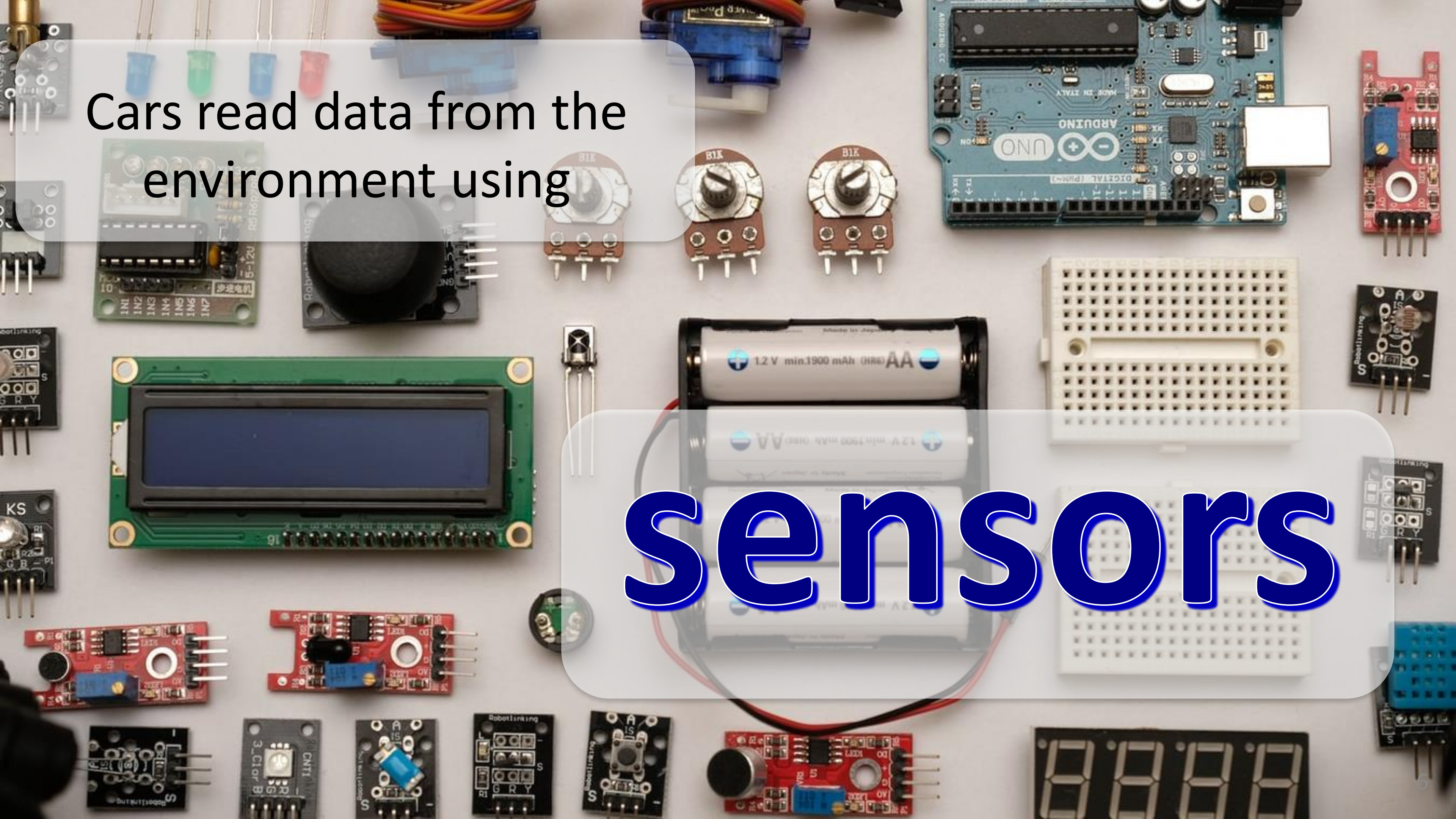
Ri The Royal Institution
Science Lives Here
Masterclass network

Contents

What will cars in the future be like?


Cars don’t just rely on
the driver anymore..
modern cars use
data
100m
60
km/h
100

How do vehicles use data?



Cars read data from the environment using

sensors

Parking sensors

Anti-lock Braking Systems

Cruise control

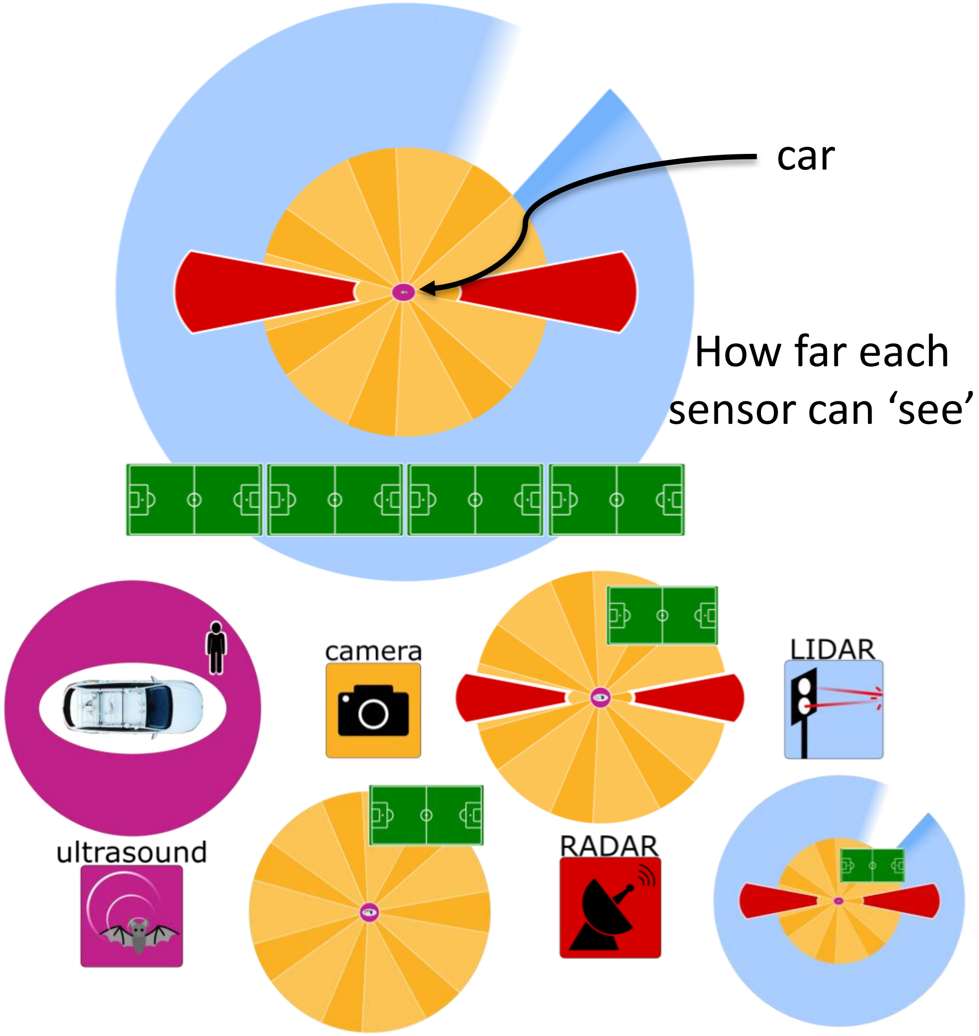
Self-driving cars

Collision Avoidance System

Connected

Eco-friendly

Sustainability



Robotics Challenge

Your challenge is to build a car that a computer can control

Fair Test

Doing a good experiment means being fair.

Each time you test something try and keep things as close to the same as possible.

Variables

Only change one thing at a time. The things you change are called **variables**.

A variable you change is called an **independent variable**. This could be the weight, the wheels, grip, *etc*.

The variable that you measure – how far the car went – is the **dependent variable**.

Make, Test, Improve

You'll never make the perfect thing on your first try.

There are always ways to improve. Engineers go through a cycle of making, testing, and improving until they get the best results.

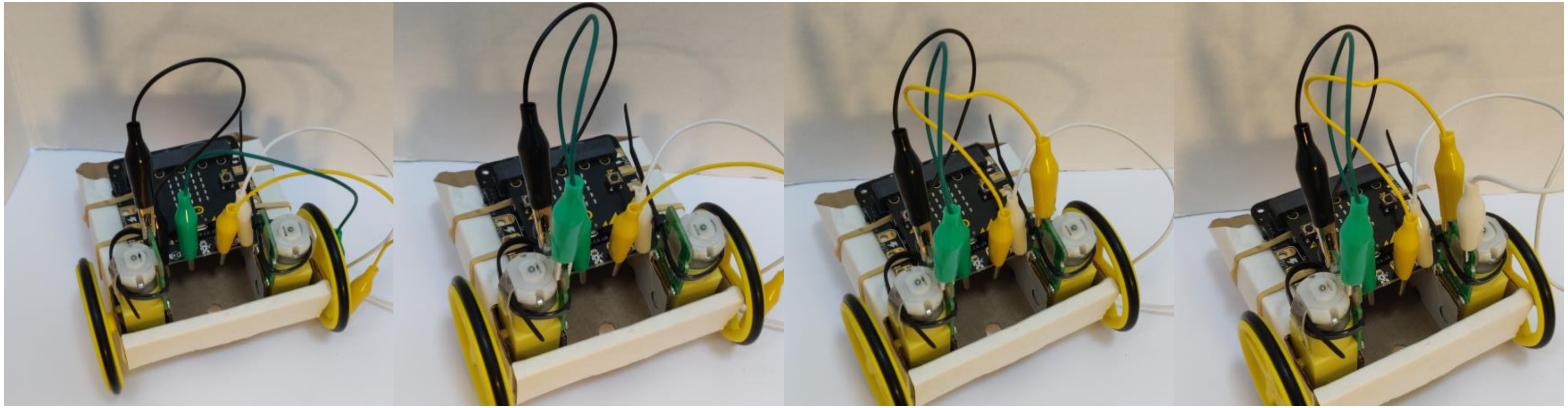
Follow the instructions on pages 6-17 of the Simple Robotics Kit instruction book to build your car. This is the task for the first lesson.

BUILDING INSTRUCTIONS

WIRING INSTRUCTIONS

If the wiring directions on p 17 of the Simple Robotics Kit handbook are confusing – these pictures might help.

1. Looking from the back of the vehicle (where the wheels are), the left most cable (black in the below pictures) connects to the front terminal of the left motor.
2. The next cable (green below) connects to the back terminal of the left motor.
3. The next cable (yellow below) connects to the front terminal of the right motor.
4. The next cable (white blelow) connects to the back terminal of the right motor.



TESTING YOUR CAR

Once the car is put together, flick the switch on the side to the ‘ON’ position.

You should see the word ‘start’ written out in red LEDs on the front of the micro:bit. This should then turn into a small arrow pointing towards the ping pong ball at the front of the vehicle.

Tilting the vehicle towards one wheel or the other while in Drive Forwards (Motors OFF) – which is the default mode when the micro:bit powers on – should change the bias setting. A number will appear on screen and increase the longer you hold the vehicle tilted.

Pressing **A** should turn the motors on.

Pressing **B** should change the mode the vehicle is in. It will cycle from Drive Forwards, Drive Backwards, Drive in a Circle (Left), Drive in a Circle (Right) and Follow a Line.

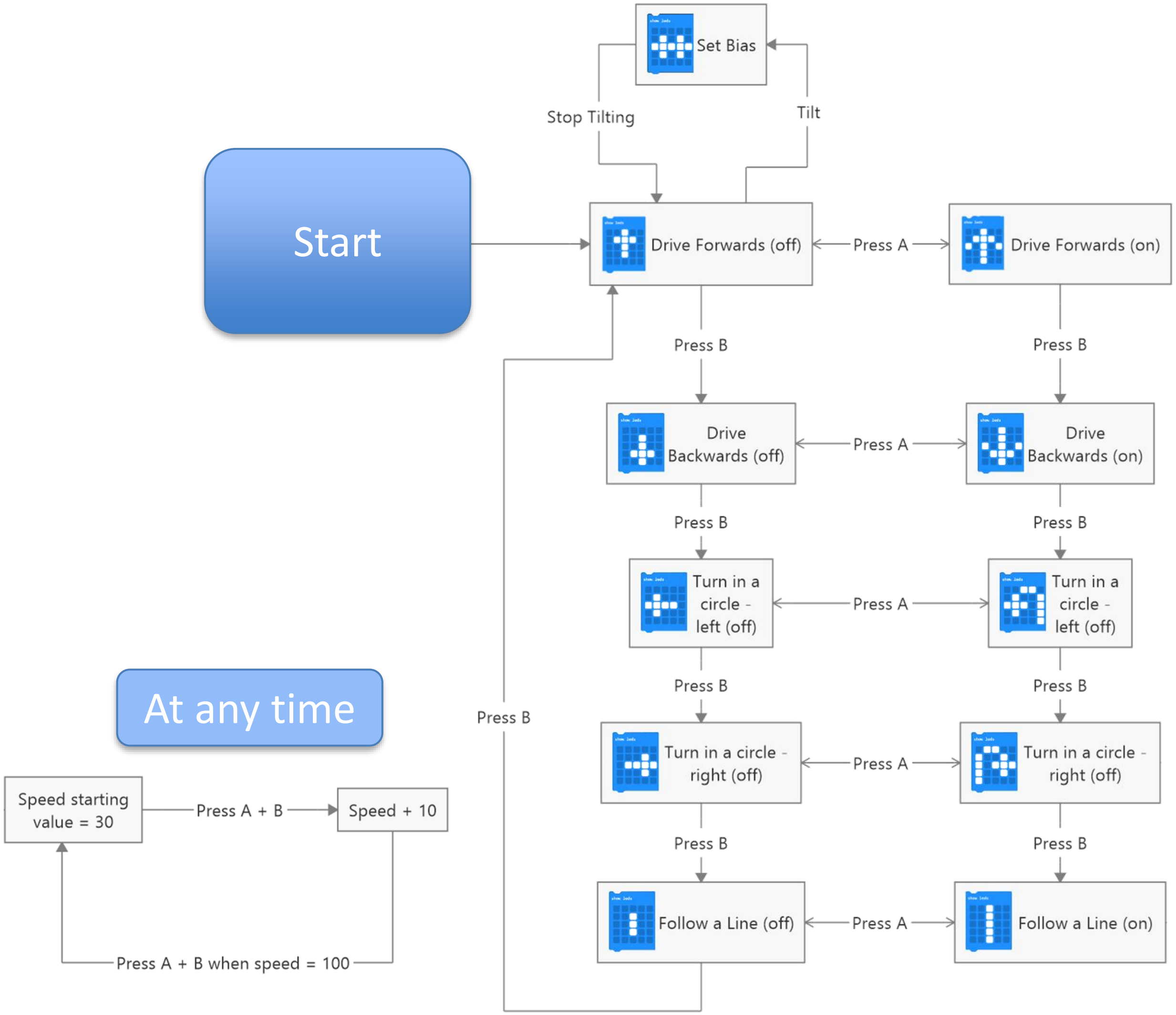
Pressing **A+B** together should change the speed of the vehicle by 10. The new speed number will be shown on screen.

Test that you can navigate through all these modes.

STATE DIAGRAM

A flowchart that helps you find your way around the modes and code on this Micro:Bit.

The TESTING CHECKLIST on the next page can be used to record whether your vehicle can get into each of these states.



TESTING CHECKLIST

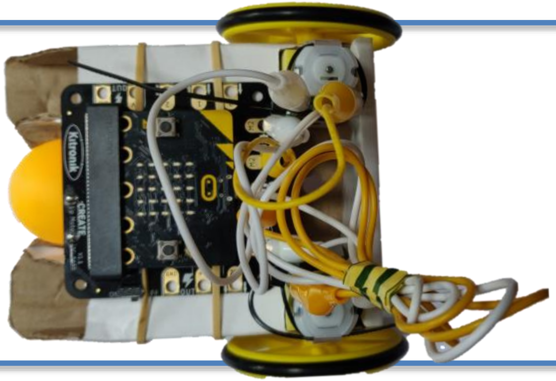
Complete this checklist to test your vehicle out. Once all these steps have been tested you are ready to take on the challenges.

- ‘Start’ displayed when powered ON
- Drive Forwards arrow shows
- Tilting the car while in Drive Forwards (Motors OFF) changes the Bias
- Pressing **A** turns on the motors (for 10 seconds)
- Turning on the motors makes the vehicle drive in the direction of the arrow
- Pressing **B** changes the mode
- Pressing **A+B** changes the speed

The first task is to get your vehicle to drive in a totally straight line.

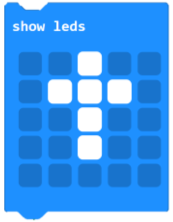
CHALLENGE #1

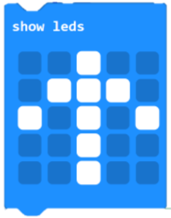
Going Wheely Far



Mark out two straight, parallel lines on the floor. They should be far enough apart that each wheel of the car touches both lines.

Drive Forwards



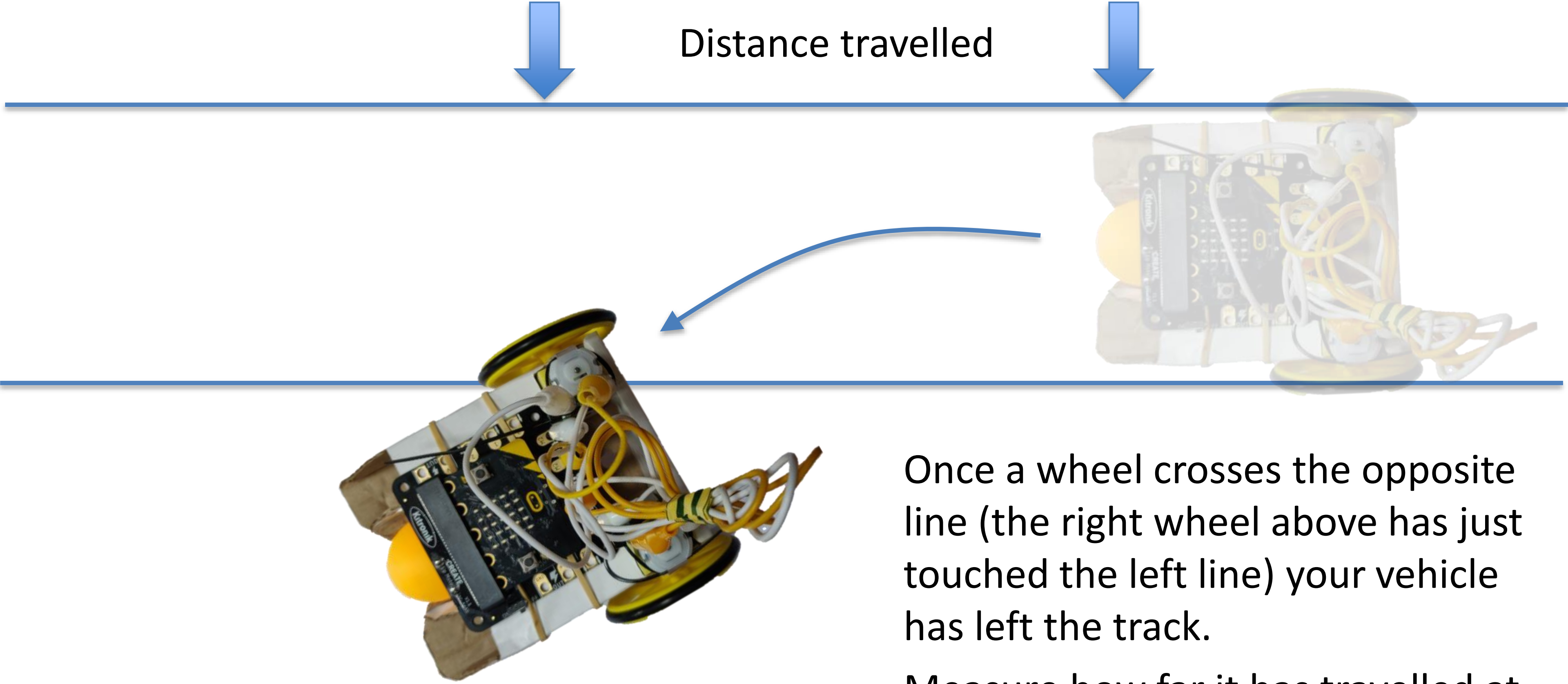


Off On

Put your vehicle into 'Drive Forwards', place it on the lines and then hit 'A' to start it.

Can it go the full 10 seconds without leaving the track?

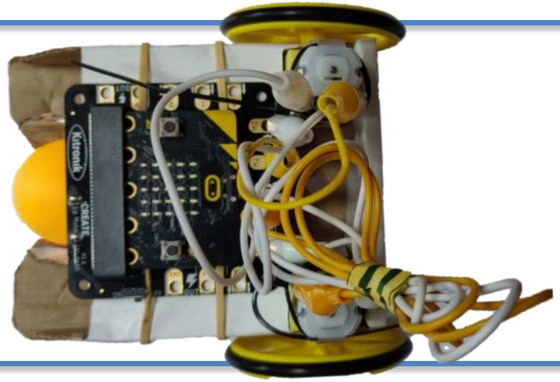
Write your results on p. 21!



Once a wheel crosses the opposite line (the right wheel above has just touched the left line) your vehicle has left the track.

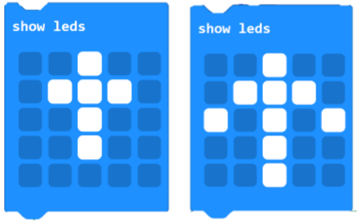
Measure how far it has travelled at this point.

Going Wheely Far



Troubleshooting

Drive Forwards



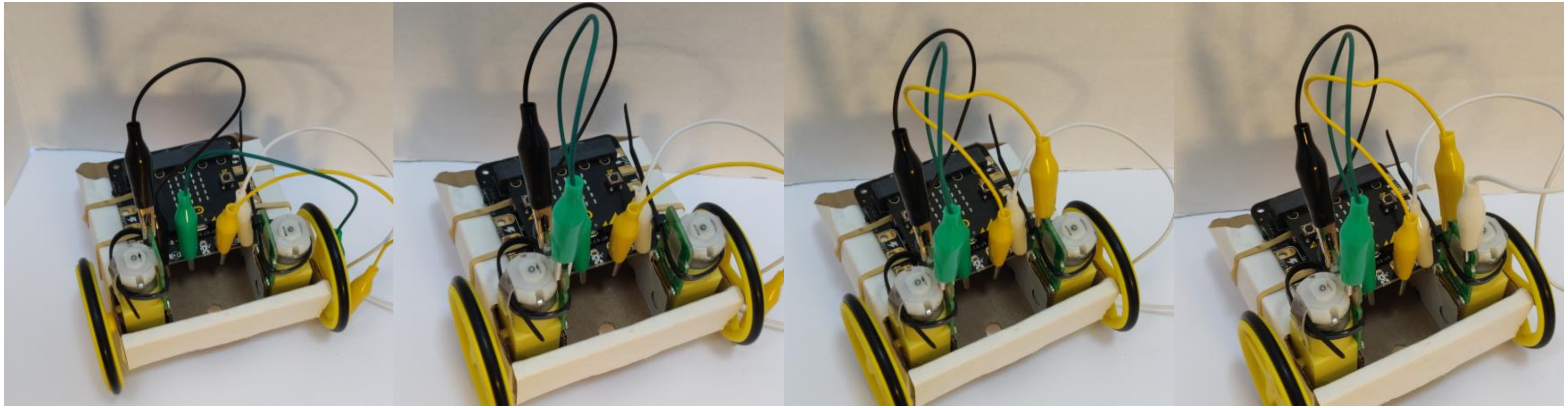
Off On

- Is your vehicle driving in curves? One wheel is spinning faster than the other.
  - The **Bias** mode can be used to fix this. **Bias** will set one motor to spin faster than the other (p. 20)
  - You could also experiment with changing the circumference of the wheel by making a thicker tire on one wheel.

WIRING PROBLEMS

If your vehicle spins in circles instead of going straight, one of your motors is wired up wrong. Swap the two cables on that motor over.

If your vehicle drives backwards instead of forwards, check which mode the vehicle is in (the arrow on the screen points which way the car will go!). If it’s going in the opposite direction to the arrow, both motors are wired wrong. Swap the cables over on both motors.



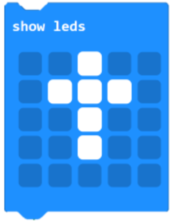
HOW TO SET BIAS

When to use: If one wheel is turning faster than the other side causing the vehicle to drive in curves instead of straight lines.

How to use:

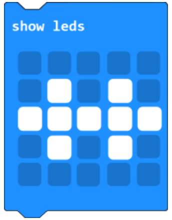
- Put the vehicle into drive forwards mode
- Make sure the motors are off. You should see this:
- Tilt the whole vehicle towards the wheel that is spinning **slower** until you see the **bias** symbol:
- A number will appear on screen that increases the longer you hold the vehicle tilted. A higher number means a higher **bias**.
- Experiment with different bias values until your vehicle drives straight.

Drive Forwards



Off

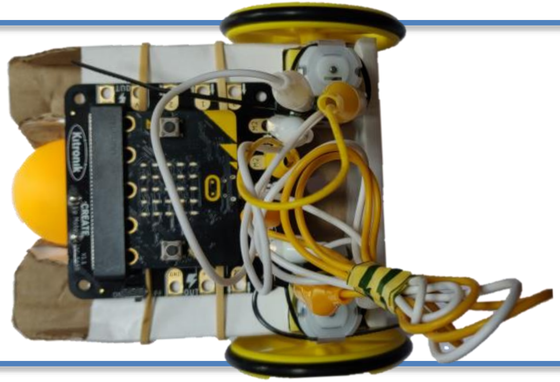
Change Bias



Try out your vehicle (Forwards)

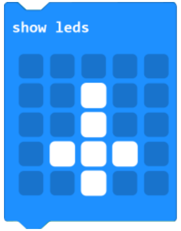
Test #	What have you changed?	Distance travelled before leaving the straight track / cm	Did it travel the full 10 seconds before leaving the track?
1	n/a		
2			
3			
4			
5			

Going Wheely Far

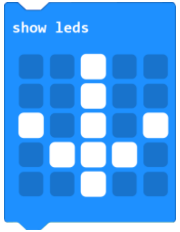


Does your vehicle work the same in reverse? Can it reverse for the full 10 seconds without leaving the track?

Drive Backwards



Off



On

Put your vehicle into ‘Drive Backwards’, place it on the lines and then hit ‘A’ to start it.

Can it go the full 10 seconds without leaving the track?

Try out your vehicle (Backwards)

Test #	What have you changed?	Distance travelled before leaving the straight track / cm	Did it travel the full 10 seconds before leaving the track?
1	n/a		
2			
3			
4			
5			

YOUR BEST RESULTS

Forwards

cm
before leaving the track

OR

completed 10 seconds inside the track

bias setting

Backwards

cm
before leaving the track

OR

completed 10 seconds inside the track

bias setting

Why is this useful?

First – vehicles should drive in straight lines.

Having well-aligned wheels actually saves a lot of fuel usage in the car!

Why is this useful?

Second – you now know the speed of the vehicle!

Instead of **Speed = 30** in the code you now know a distance the vehicle travels in 10 seconds. You can work out a speed in cm/s (centimetres per second)

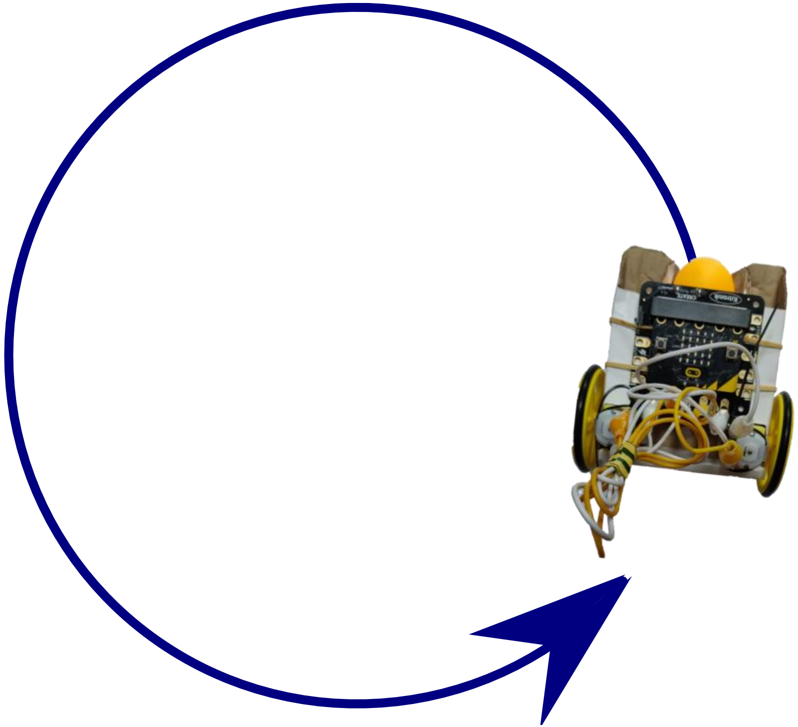

Understanding speed
will be crucial…
..if cars are ever going
to drive themselves.
100m
60 km/h

The second task is to get your vehicle to drive in one perfect circle.

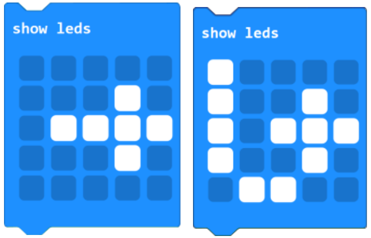
CHALLENGE #2

Round and Round

Your car should start and end on the same spot, having turned a full circle.



Turn in a Circle (Left)



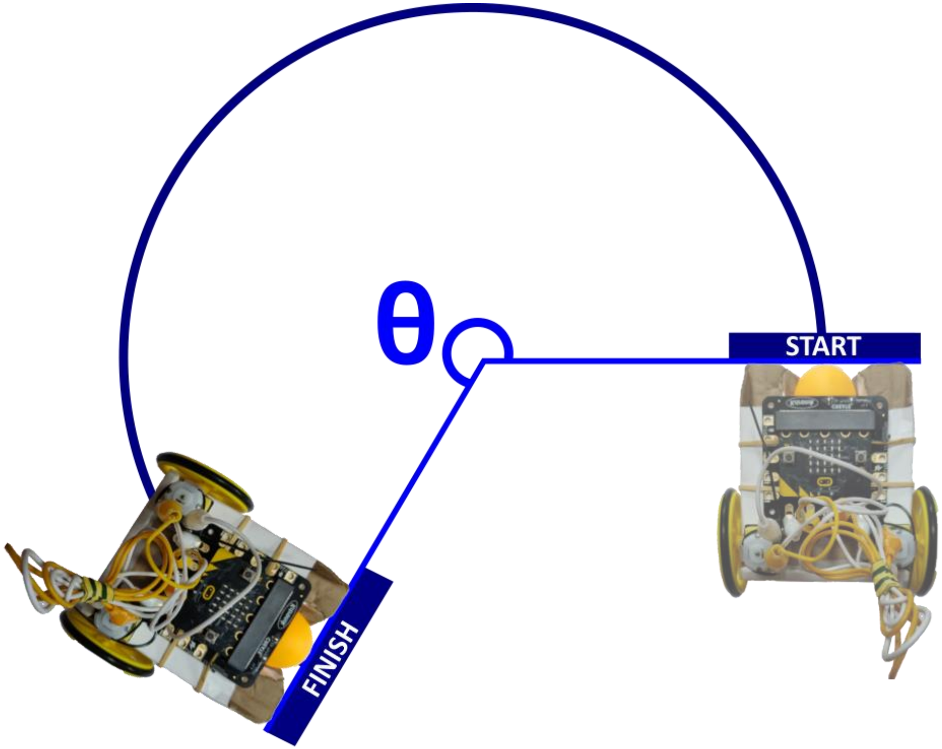
Off On

Put your vehicle into 'Turn in a Circle (Left), place it on a mark on the floor and turn the motors on with **A**.

How far around the full circle does the car drive?

Write your results on p. 32

Mark a line across the front of your vehicle as your START line.

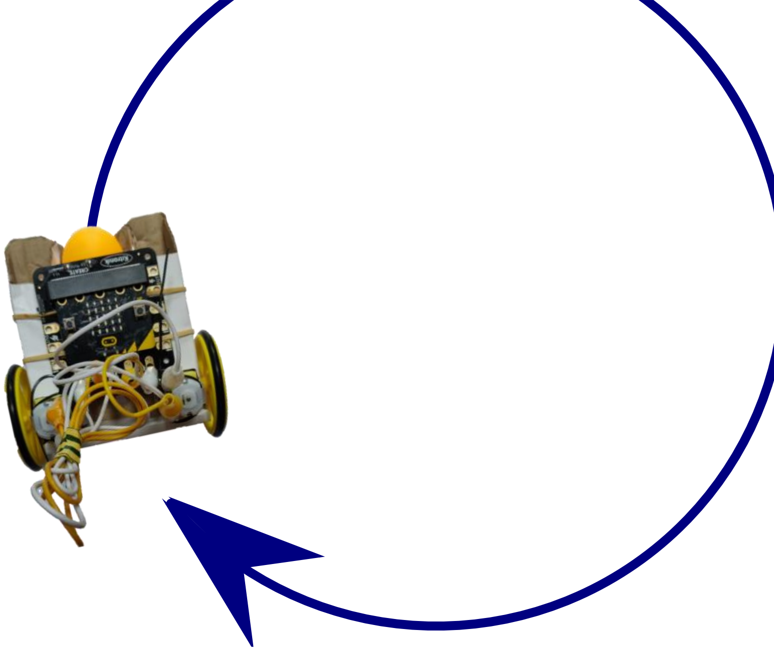


Mark a line across the front of your vehicle as the FINISH line.

The START and FINISH lines meet in the centre of the circle your vehicle is driving.

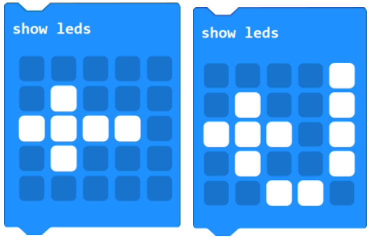
The angle between the two lines is how far your car has travelled.

Round and Round



Your car should start and end on the same spot, having turned a full circle.

Turn in a Circle (Right)



Off On

Put your vehicle into ‘Turn in a Circle (Right), place it on a mark on the floor and turn the motors on with **A**.

Does it turn through the same angle as the Left Circle?

Write your results on p. 32

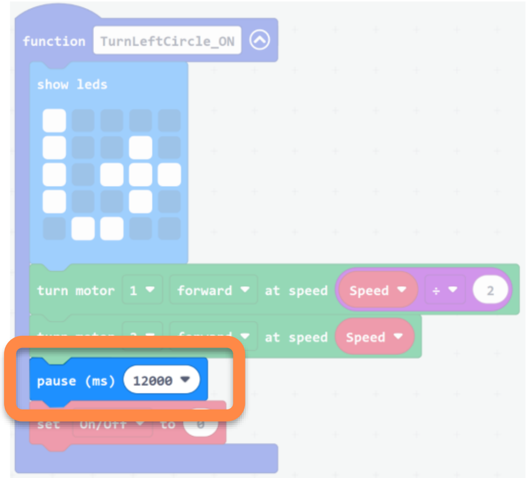
Try out your vehicle

Test #	Angle travelled around the circle / degree
Left 1	
Left 2	
Left 3	
Left (average)	

Test #	Angle travelled around the circle / degree
Right 1	
Right 2	
Right 3	
Right (average)	

MAKING THE PERFECT CIRCLE

If 12 seconds (12000 ms) isn't getting your car the whole way around the circle, we need to work out how long your vehicle needs to turn for!



Left average / 360 = % of a full circle

pause (ms) 12000 is getting us this % of the way around the circle. How can you get from here to the length of pause that would turn a full 360 degree circle?

Your working

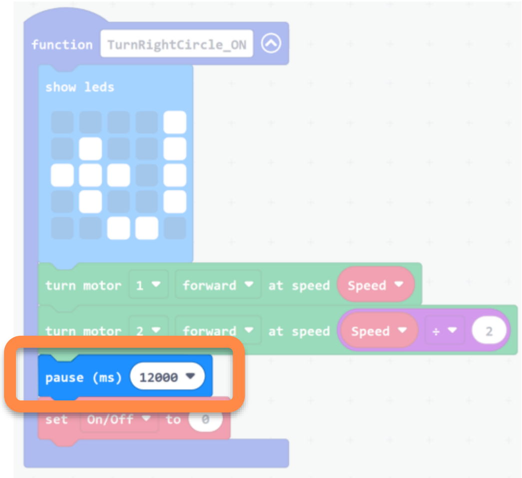
= Time needed

MAKING THE PERFECT CIRCLE

Now do the same for the right circle too.

Right average

[ ] / 360 = [ ] % of a full circle



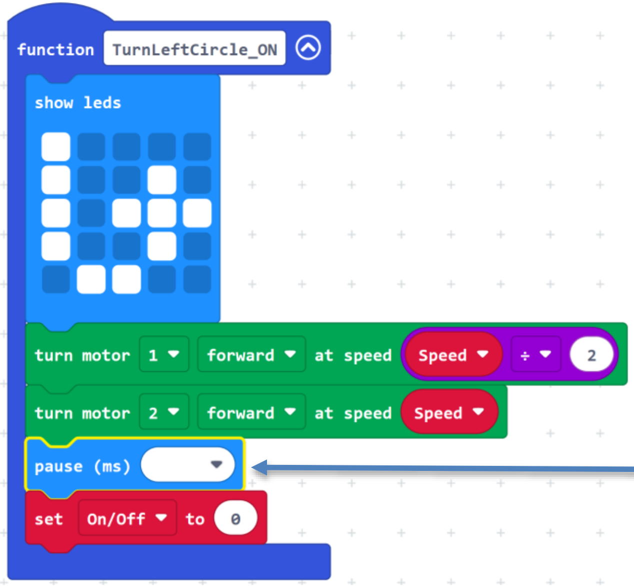
pause (ms) 12000 is getting us this % of the way around the circle.

How can you get from here (12000) to the length of pause that would turn a full 360 degree circle?

Your working

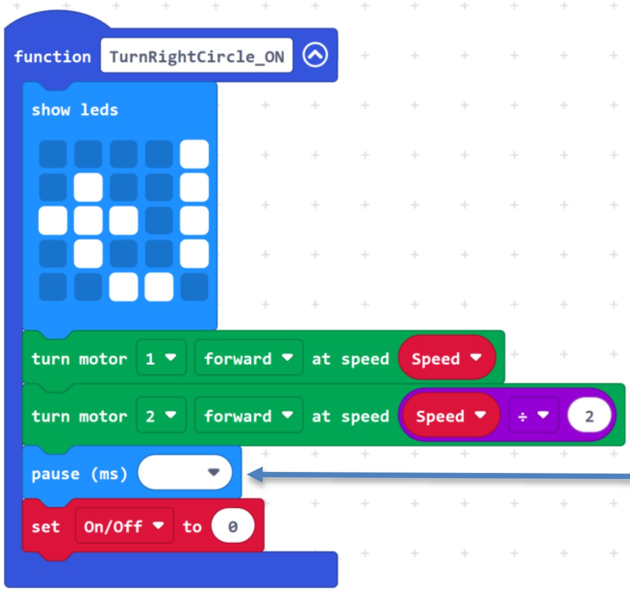
= [ ] Time needed

YOUR PERFECT CIRCLES



Time needed for a left circle

ms



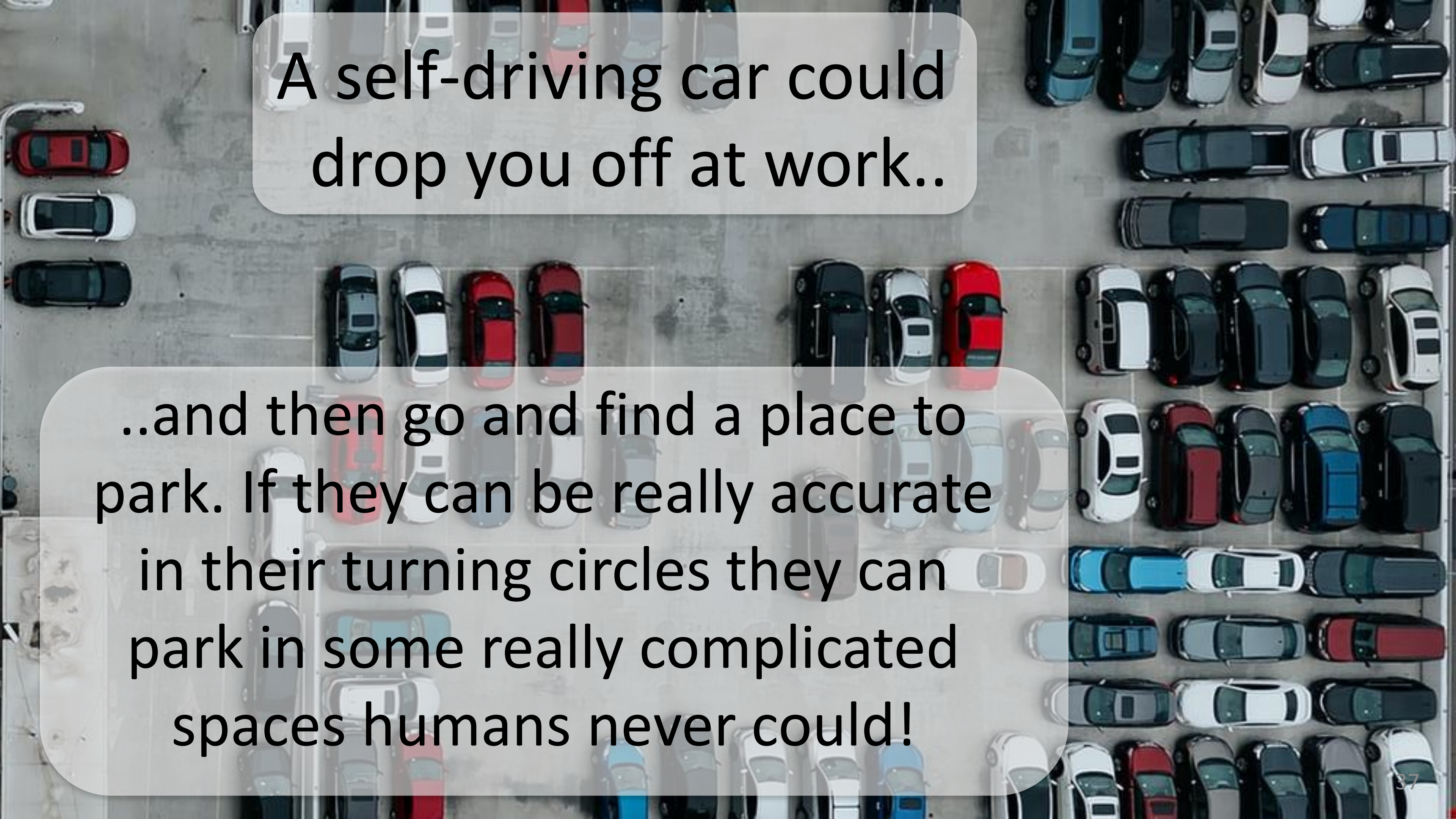
Time needed for a right circle

ms

Why is this useful?

You now know how many milliseconds it takes to turn each angle of a circle.

You could program this vehicle to navigate complicated courses and take corners!



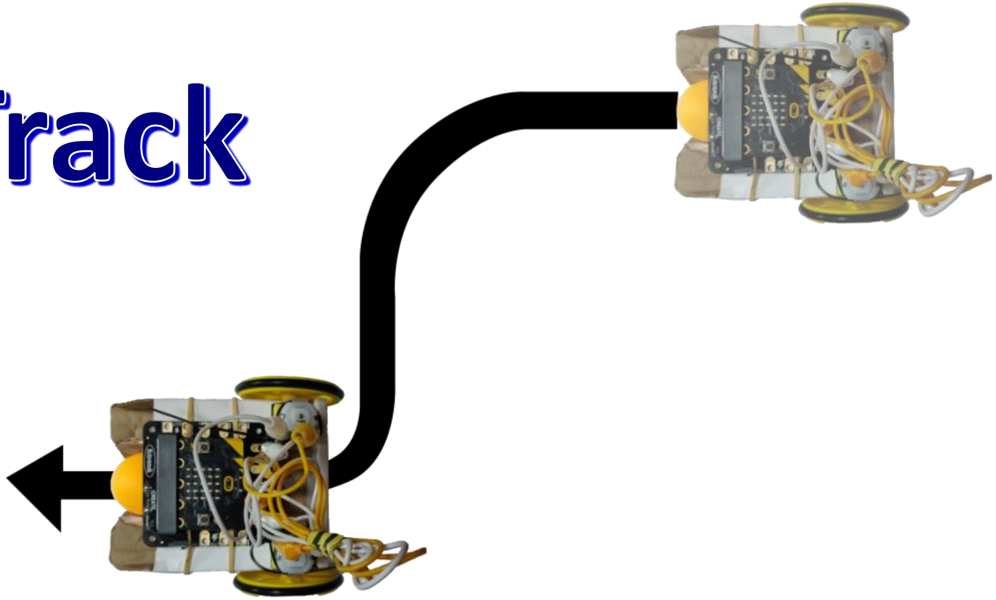
A self-driving car could drop you off at work..

..and then go and find a place to park. If they can be really accurate in their turning circles they can park in some really complicated spaces humans never could!

The third task is to get your vehicle to follow a black line drawn to form a track.

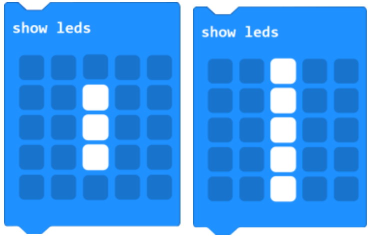
CHALLENGE #3

On the Right Track



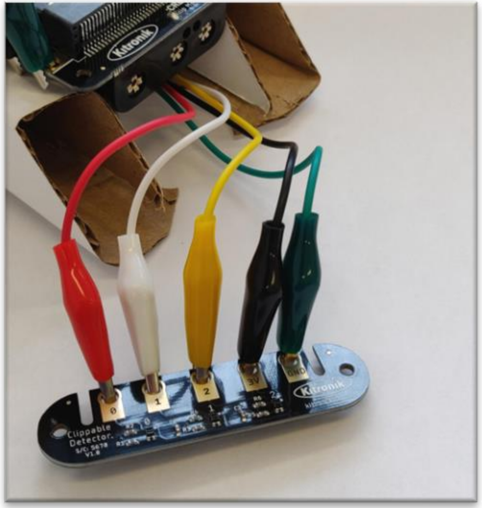
We're going to add an extension on to the vehicle to get it to follow a line drawn to form a track.

Line Following



Off On

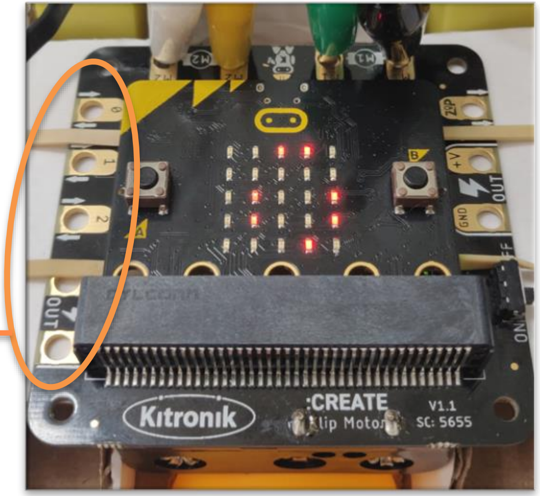
WIRING INSTRUCTIONS



Take the line following detector out of your box.

Clip 5 crocodile clips on to it – one for each gold coated opening.

These clips then connect onto the Kitronik Clip Motor. Make sure to match the labels on the detector to the labels on the Clip Motor.

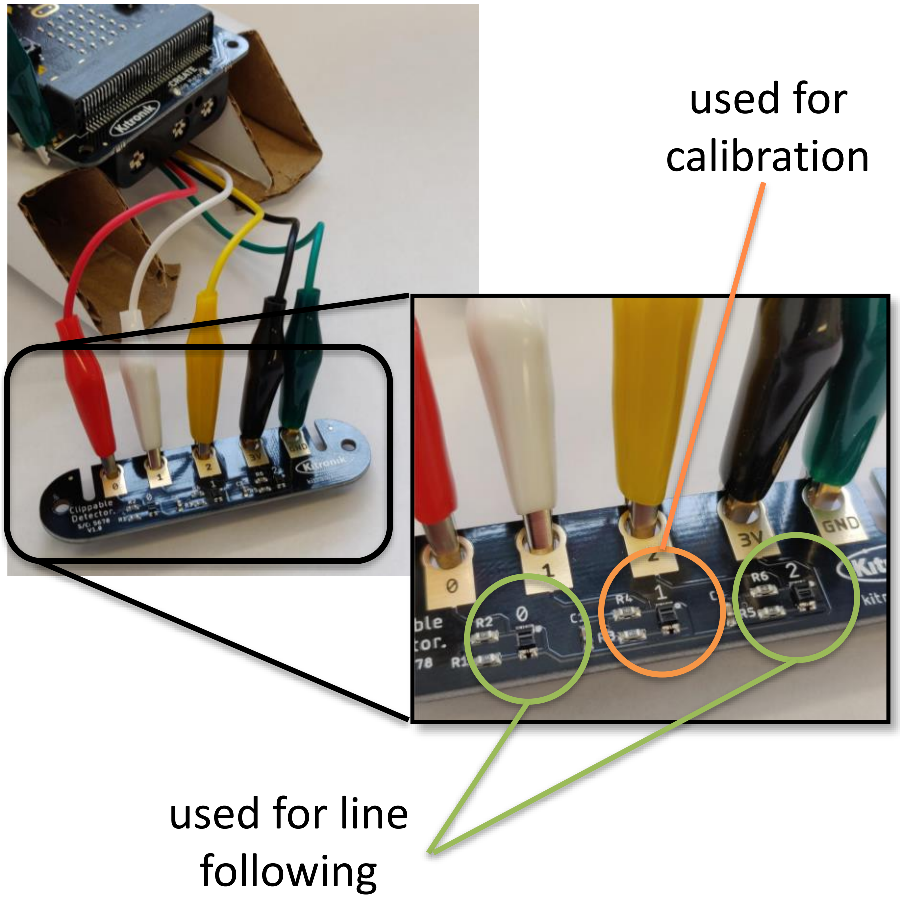


Getting Our Bearings

To detect the difference between the paper and the black line the robot uses sensors.

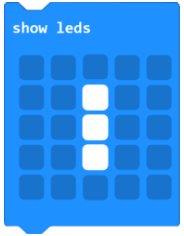
We're going to use the middle sensor to calibrate how these sensors read the line.

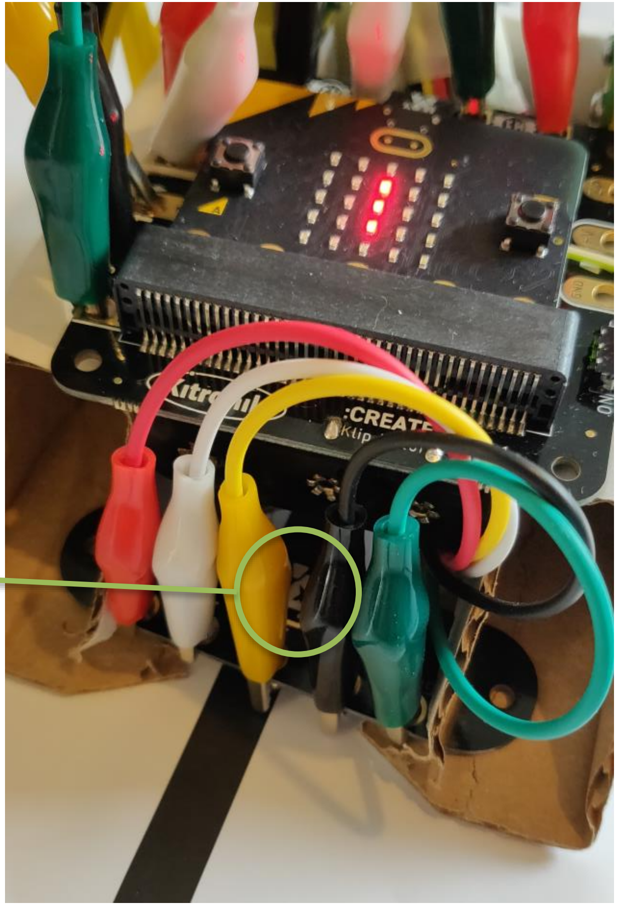
When you flick the power switch on the vehicle **ON**, the sensor takes a reading. This should always be done over **white** paper.



Getting Our Bearings

Slot the sensor board into the front of the chassis (where there ping pong ball used to be). Make sure the sensors are pointing down. You should see the Kitronik logo pointing up.





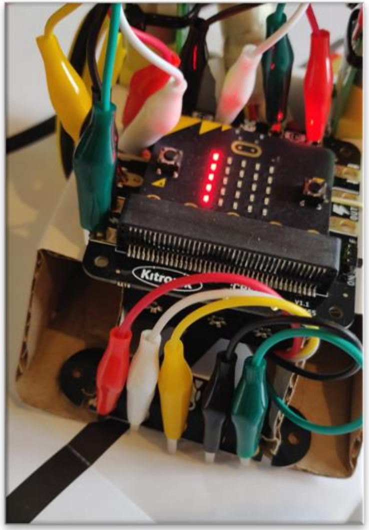
Place the vehicle over white paper and turn the power on with the switch. Select the line following mode but, before you turn the motors on, place the vehicle carefully over your black line so that the middle sensor is pointing directly down on to the black line.

When you press **A** in Line Following Mode to turn the motors on, this middle sensor then takes a reading of the black line.

The robot then knows the difference between the white paper and the black line.

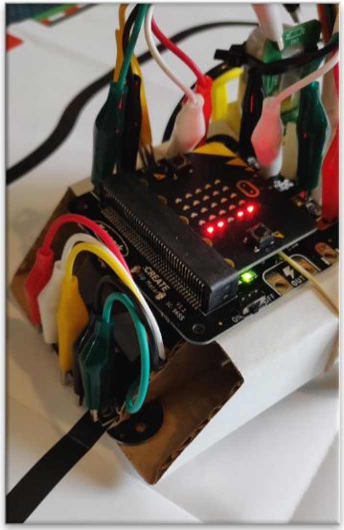
Getting Our Bearings

When the sensor on the right detects the black line instead of white paper, the vehicle has strayed to the left. The left motors keep spinning but the right motors stop.



When the sensor on the left detects the black line the opposite happens.

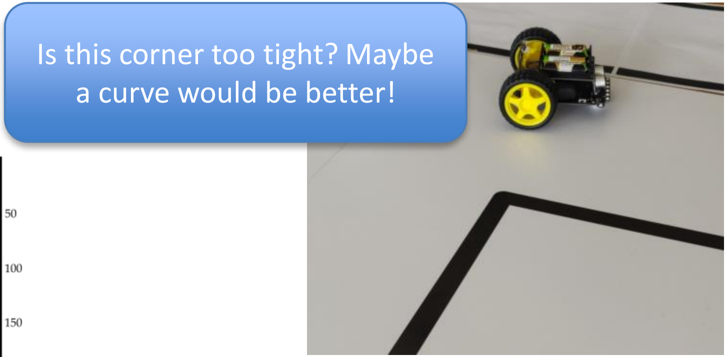
The vehicle should keep wiggling along the black line following it.

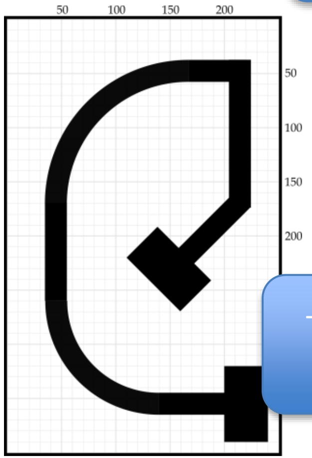


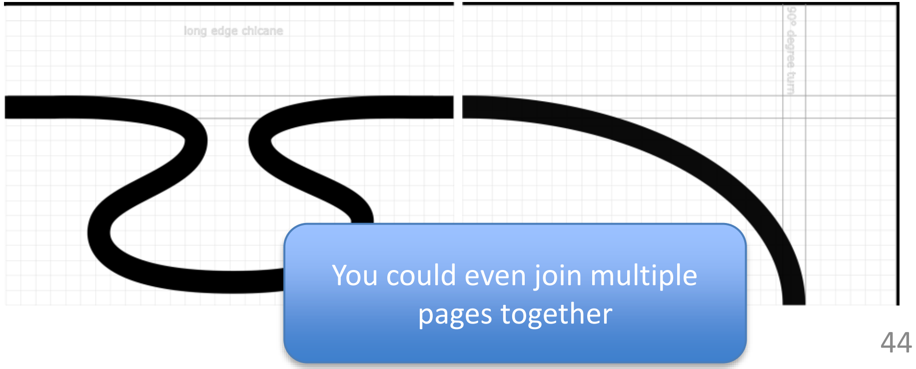
The vehicle might not be able to make every corner you can throw at it!

Carefully test out with black lines drawn on paper what kind of corners, curves, angles and line thicknesses work best for your vehicle.

Remember to calibrate it again if anything changes by switching the vehicle off and starting again.

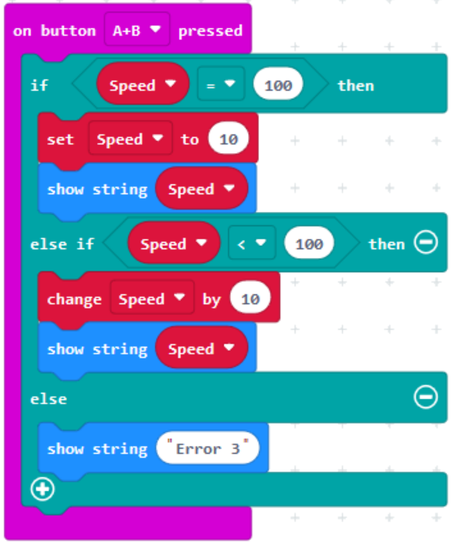


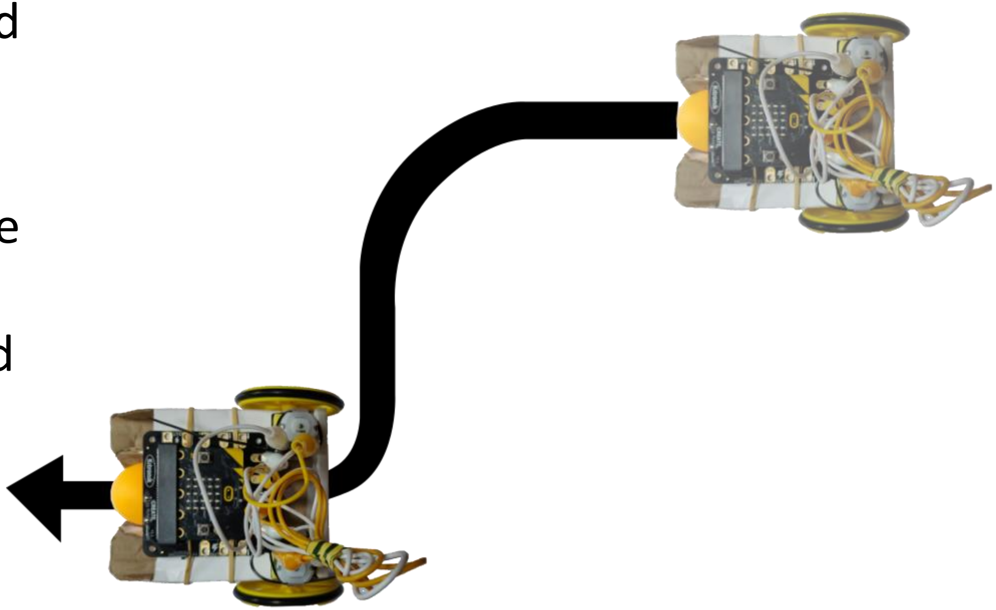




If your vehicle keeps losing the line and driving off, you might need to change the speed of the vehicle with **A+B**.

Remember that the speed will increase in steps of 10 until it hits 100 before pressing **A+B** again will drop the speed to 10.





Is there a lot of bright light shining on the map? That could interfere with the sensors!

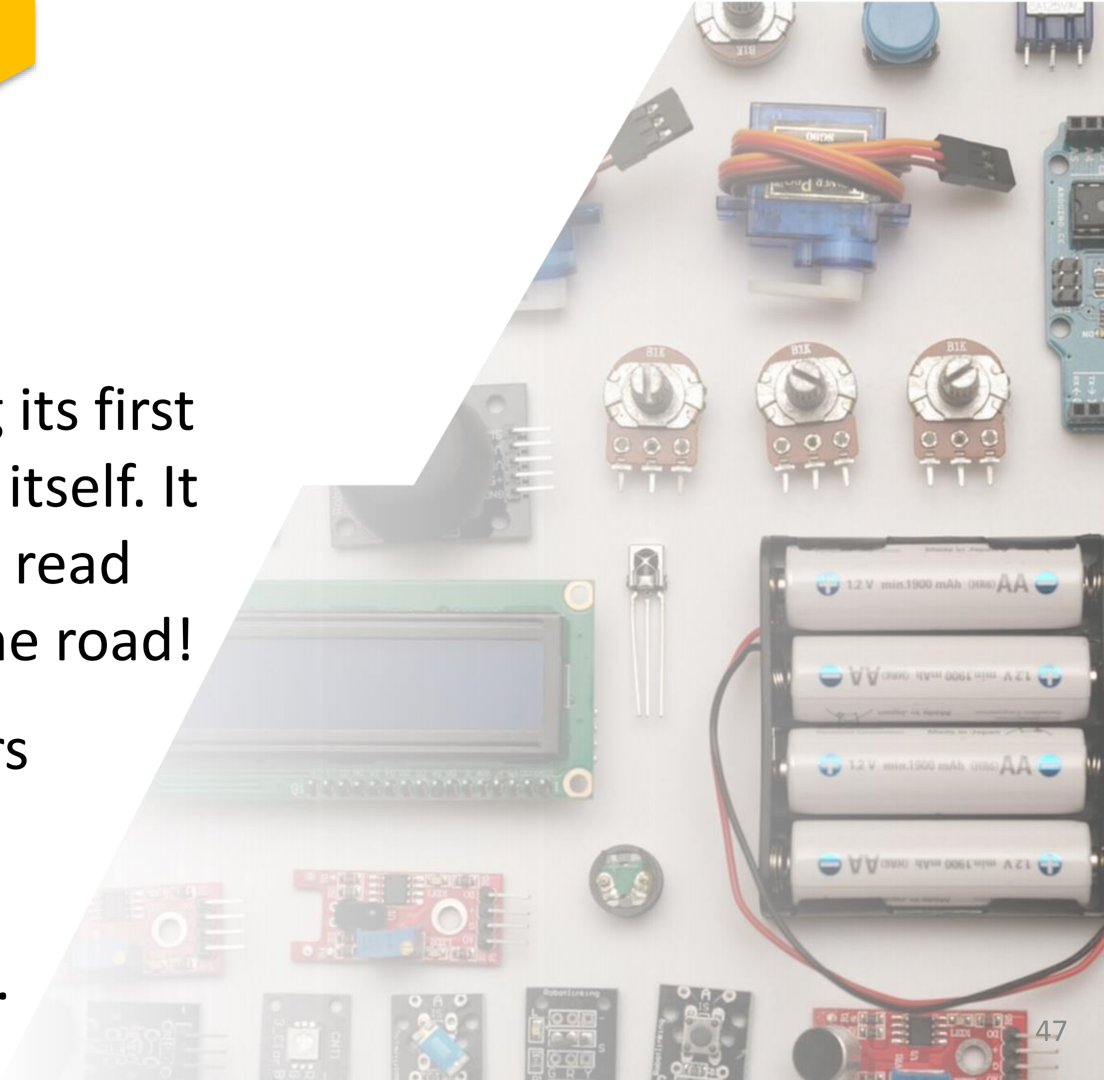
Try out your vehicle (Follow a Line)

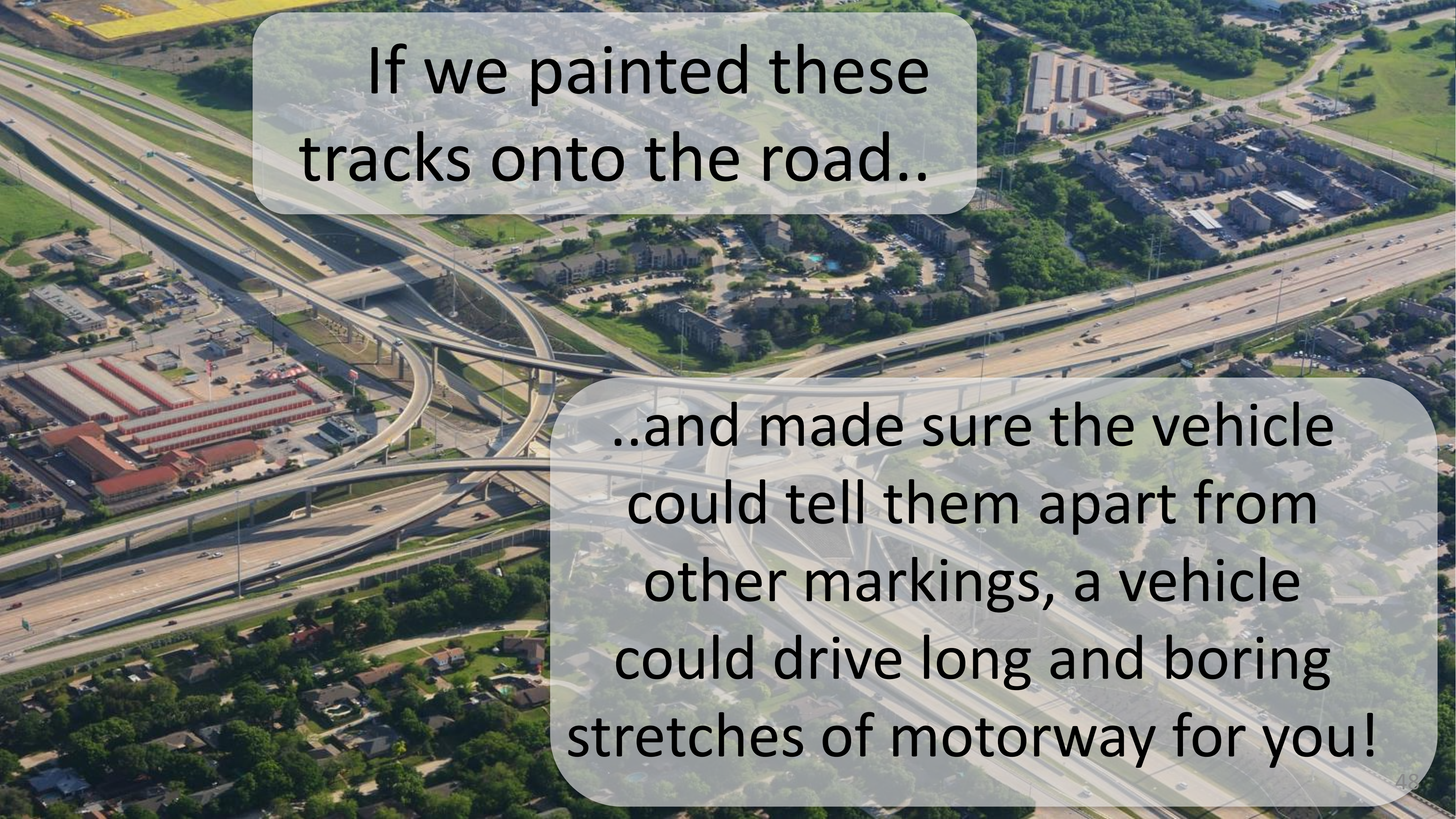
Test #	What was the track like? Pen line, town map, line thickness, corners, curves, lighting in the room *etc*.	Vehicle speed (default is 30)	How well did the vehicle follow the line?
1			
2			
3			
4			
5			

Why is this useful?

Your vehicle is taking its first steps into driving for itself. It is using sensors to read information about the road!

Add more complicated sensors and code, and your car could respond to the environment and other drivers on the road.

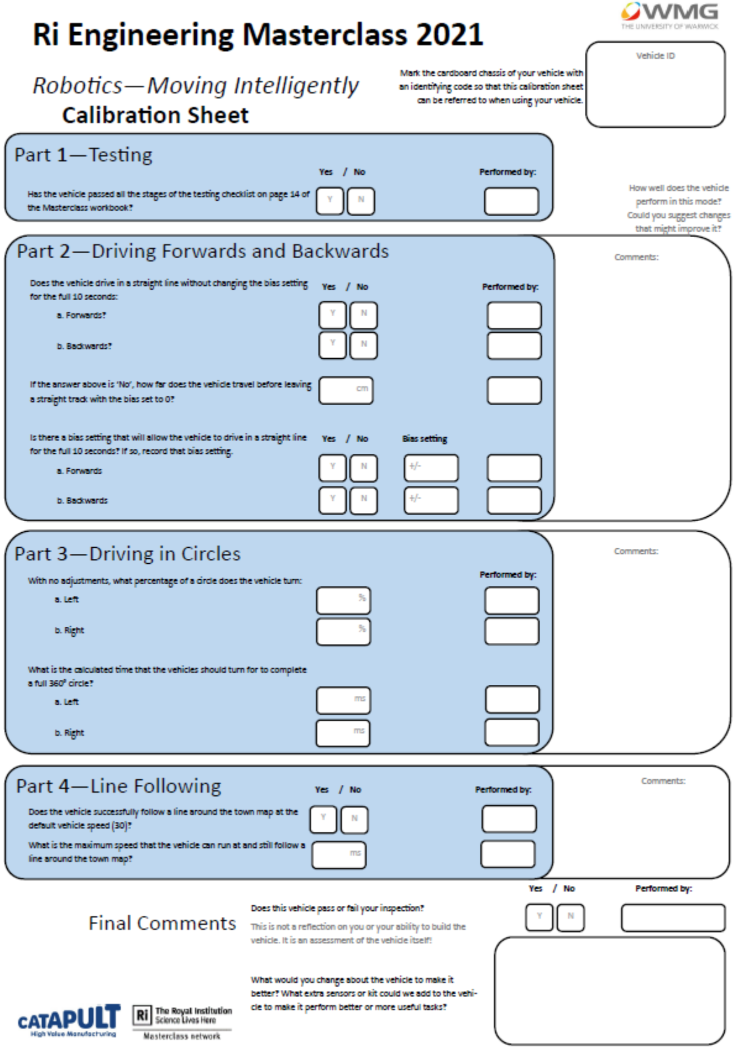



If we painted these tracks onto the road..
..and made sure the vehicle could tell them apart from other markings, a vehicle could drive long and boring stretches of motorway for you!

The final task is to fill in a calibration sheet. This way any future work using this vehicle kit can be set up straight away to perform functions accurately.

Using calibration sheet is a standard practice across all STEM fields.

CALIBRATION SHEET



Ri Engineering Masterclass 2021

Robotics—Moving Intelligently

Calibration Sheet

WMG THE UNIVERSITY OF WARWICK

Mark the cardboard chassis of your vehicle with an identifying code so that this calibration sheet can be referred to when using your vehicle.

Vehicle ID

Part 1—Testing

Has the vehicle passed all the stages of the testing checklist on page 14 of the Masterclass workbook? Yes / No — Y N — Performed by:

How well does the vehicle perform in this mode? Could you suggest changes that might improve it?

Part 2—Driving Forwards and Backwards

Does the vehicle drive in a straight line without changing the bias setting for the full 10 seconds: Yes / No — Performed by:

a. Forwards? Y N

b. Backwards? Y N

If the answer above is 'No', how far does the vehicle travel before leaving a straight track with the bias set to 0? cm

Is there a bias setting that will allow the vehicle to drive in a straight line for the full 10 seconds? If so, record that bias setting. Yes / No — Bias setting

a. Forwards Y N +/-

b. Backwards Y N +/-

Comments:

Part 3—Driving in Circles

With no adjustments, what percentage of a circle does the vehicle turn: Performed by:

a. Left %

b. Right %

What is the calculated time that the vehicles should turn for to complete a full 360° circle?

a. Left ms

b. Right ms

Comments:

Part 4—Line Following

Does the vehicle successfully follow a line around the town map at the default vehicle speed (30)? Yes / No — Y N — Performed by:

What is the maximum speed that the vehicle can run at and still follow a line around the town map? ms

Comments:

Final Comments

Does this vehicle pass or fail your inspection? Yes / No — Y N — Performed by:

This is not a reflection on you or your ability to build the vehicle. It is an assessment of the vehicle itself!

What would you change about the vehicle to make it better? What extra sensors or kit could we add to the vehicle to make it perform better or more useful tasks?

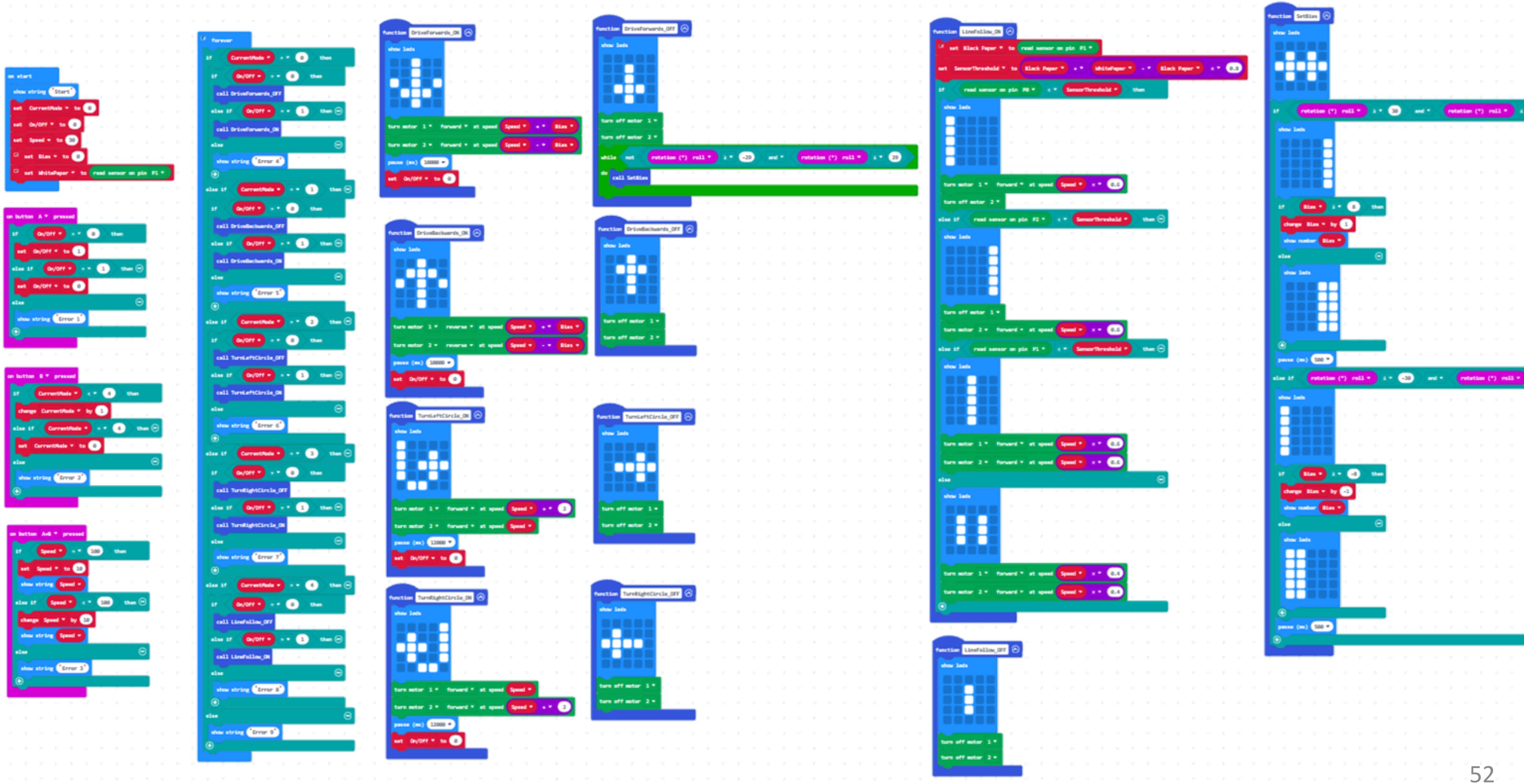
CATAPULT High Value Manufacturing — Ri The Royal Institution Science Lives Here — Masterclass network

You should have a paper version of this calibration form with your kit.

This section will look under the bonnet to see how this vehicle works.

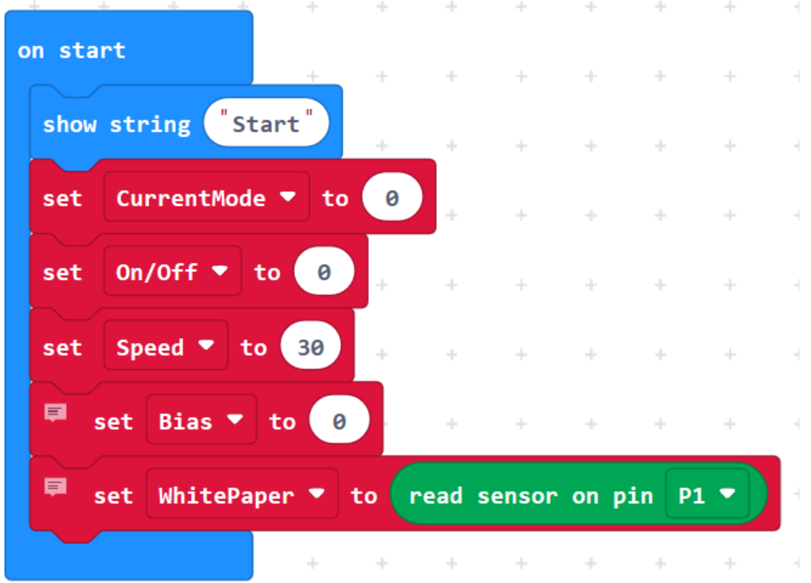
Each section of the code is broken down because it can look a bit overwhelming all at once!

WHAT IS THE CODE DOING?



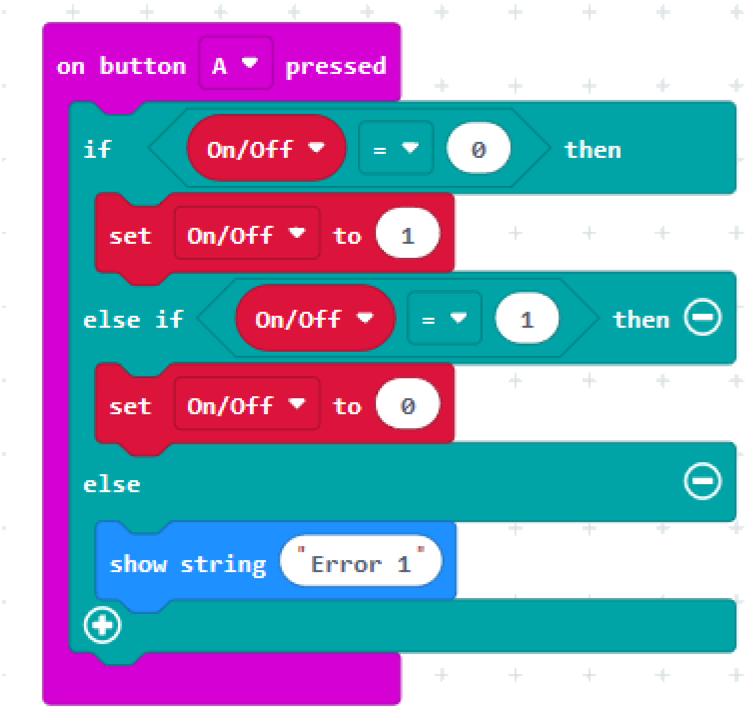
INITIAL STATE

- **When the Micro:Bit first turns on this bit of code runs. It puts it into a helpful state for you to start off in.**
  - The first mode is '**DriveForwards**' – which is represented by **0**.
  - The motors start on the **Off** state – represented by **0**.
  - The first time you turn the motors on they will be going slowly at a **speed** of **30** out of a maximum **100**.
  - We assume that the motors on each side are balanced so the bias is set to **0**.



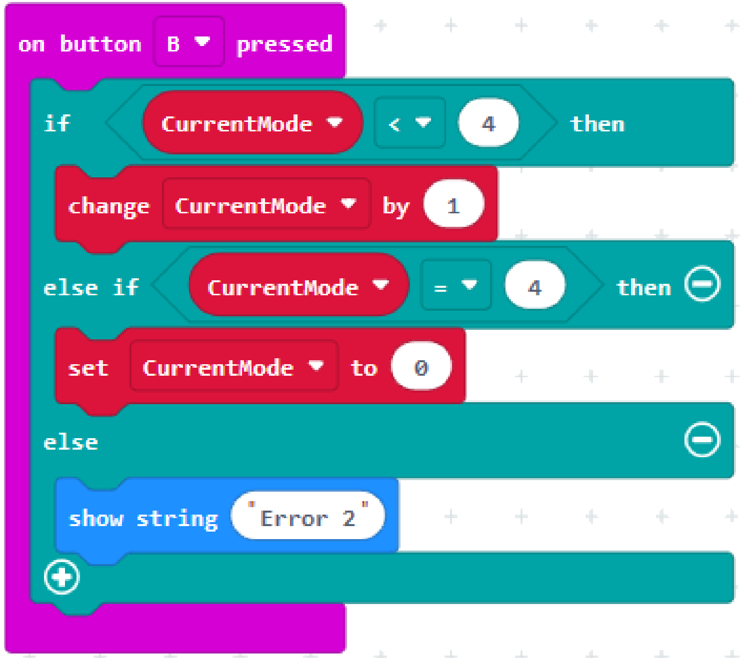
MOTORS – ON/OFF

- **When you press the 'A' button on the microbit:**
  - You toggle whether the motors are on or off.
  - Motors **ON** is represented by **1**
  - Motors **OFF** is represented by **0**



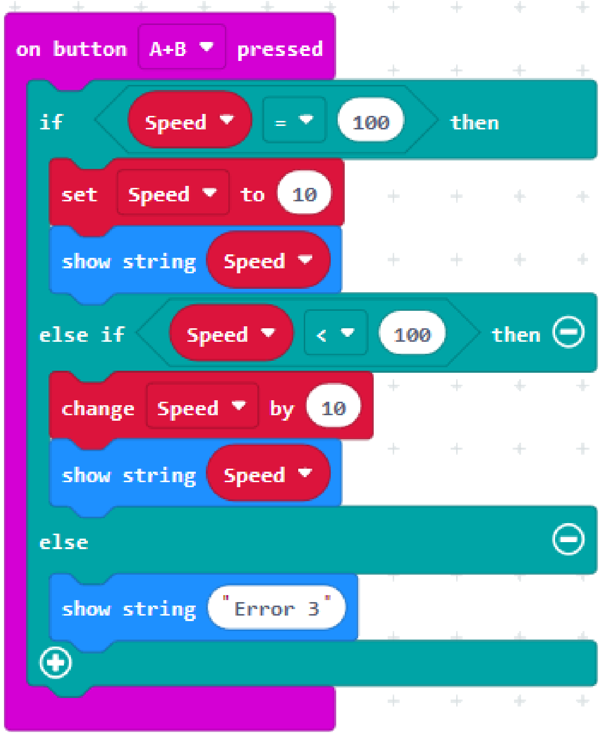
CHANGING MODE

- **When you press the 'B' button on the microbit:**
  - You change the mode the vehicle is driving in.
  - Pressing B cycles through modes 0, 1, 2, 3 and 4.
  - If the mode is 0,1,2, or 3 the number is increased by 1.
  - If the mode is 4 the number is reset to 0 (there is no mode 5!)



CHANGING SPEED

- **When you press the ‘A’ and ‘B’ buttons at the same time**
  - You change the speed the motors will run at.
  - The motors start at a slow **30**.
  - Pressing **A+B** will increase the speed by 10 and display the new speed on the screen.
  - When the speed is **100** it is reset to the minimum speed **10** (since **0** would be **OFF**)



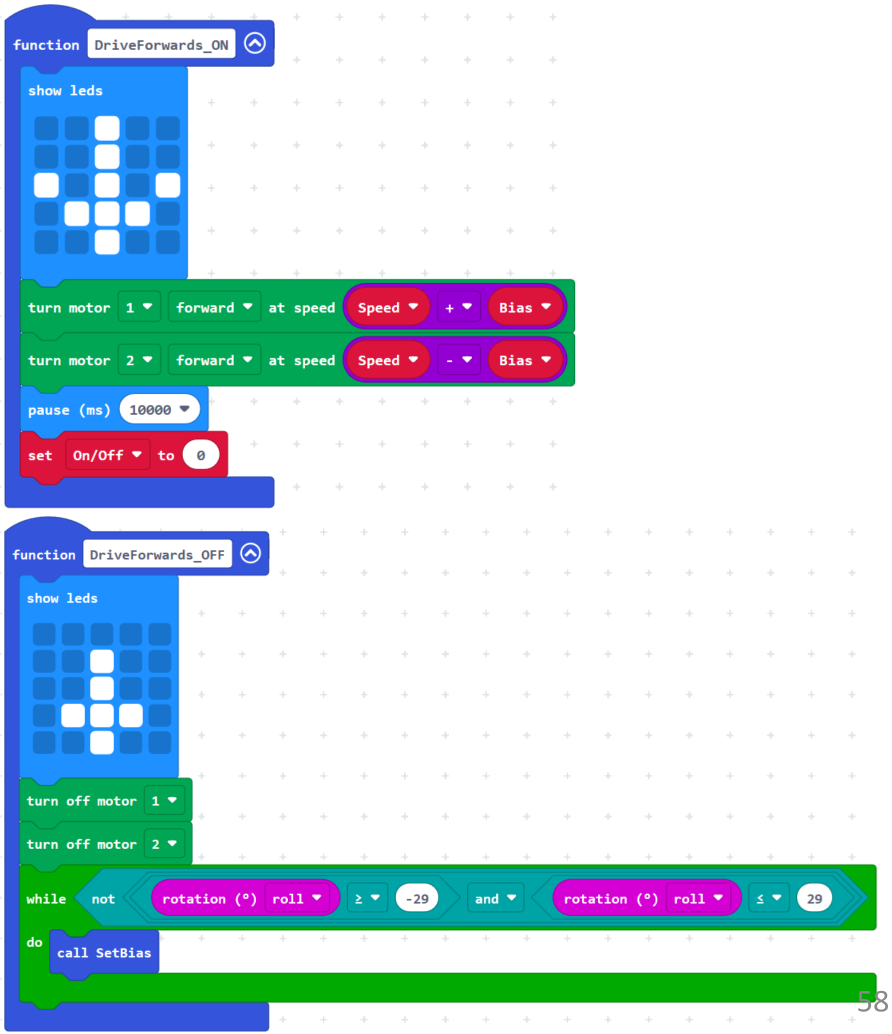
MAIN CONTROL LOOP

The main process that controls the robot is called the 'control loop'. It determines the actions the robot will take and these decisions are based on inputs from sensors and buttons. If you want to change an aspect of the robots behaviour then start by looking at the logic in the control loop.

- Pressing **A** turns the motors **ON** or **OFF**
- Pressing **B** changes **CurrentMode**
- With those 2 buttons you can navigate through the whole of this code!

DRIVE FORWARDS

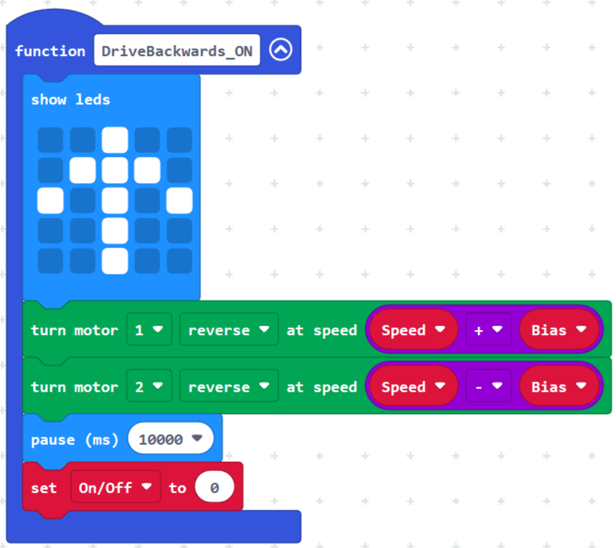
- **Drives forwards, surprisingly.**
  - Both motors are set to '**forwards**'
  - **Speed** is changed with **A+B**
  - **Bias** can be used to make the motors spin at the same speed as each other (see p. 17)
  - The car will drive forwards for 10 seconds before the motors turn off.

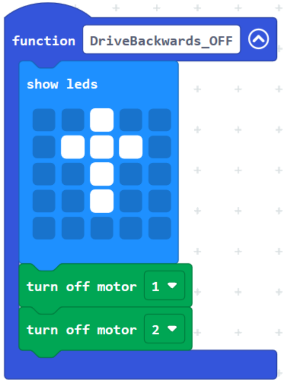


Note: the arrows point in the opposite direction you might expect here because the bottom of the micro:bit points in the direction of travel when clipped into the car. All the arrows need to be flipped in the program to account for this.

DRIVE BACKWARDS

- **Drives backwards.**
  - Behaves in the same way as **DRIVE FORWARDS**.
  - The motors are set to **Reverse**.
  - **Speed** can be changed as usual with **A+B**.
  - **Bias** can be used to make the vehicle drive in straight lines. See p. 17.
  - The vehicle will reverse for 10 seconds before turning the motors off.

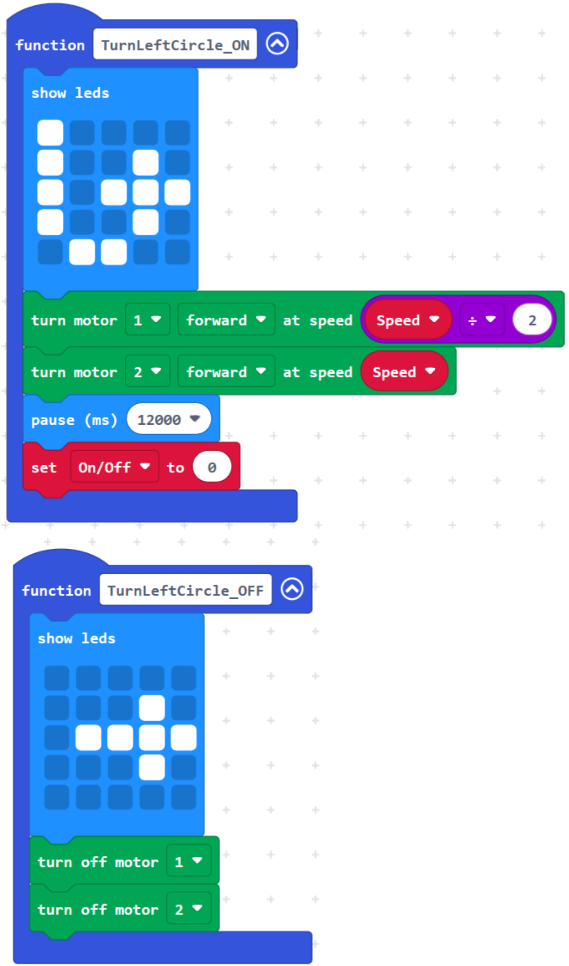




Note: the arrows point in the opposite direction you might expect here because the bottom of the micro:bit points in the direction of travel when clipped into the car. All the arrows need to be flipped in the program to account for this.

TURN IN A CIRCLE (LEFT)

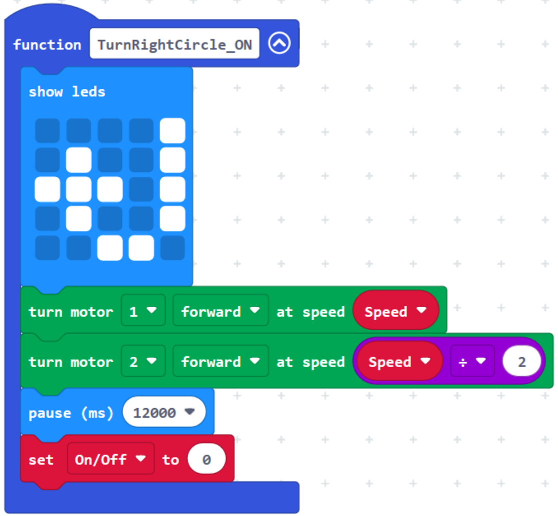
- **Drives the right motor faster than the left.**
  - This spins the vehicle in a circle to the left.
  - The motors will be **ON** for **8 seconds** and then will turn **OFF**.
  - This should get your vehicle close to 1 full circle.

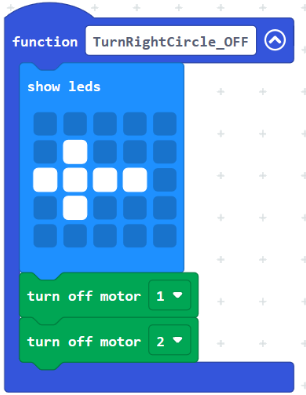


Note: the arrows point in the opposite direction you might expect here because the bottom of the micro:bit points in the direction of travel when clipped into the car. All the arrows need to be flipped in the program to account for this.

TURN IN A CIRCLE (RIGHT)

- **Drives the left motor faster than the right.**
  - This spins the vehicle in a circle to the right.
  - The motors will be **ON** for **12 seconds** and then will turn **OFF**.
  - This should get your vehicle close to 1 full circle.

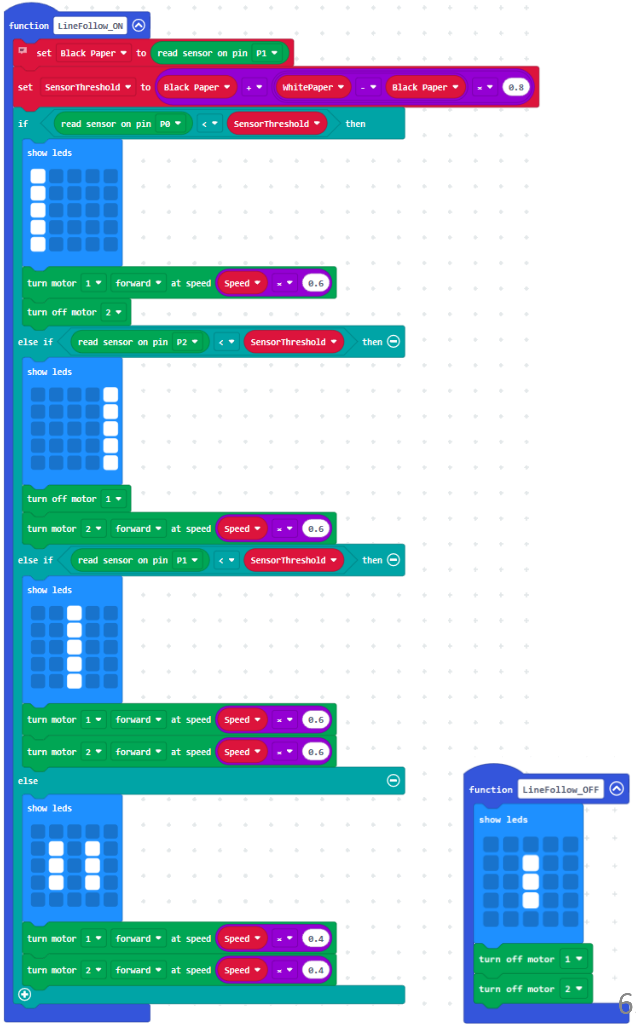




Note: the arrows point in the opposite direction you might expect here because the bottom of the micro:bit points in the direction of travel when clipped into the car. All the arrows need to be flipped in the program to account for this.

FOLLOW A LINE

- There are 3 sensors on the line following sensor module that can detect light.
  - If the sensor is over a black line it will detect less light – so the measurement from that sensor will be a smaller number.
  - If the sensor is over white paper the measurement will be a larger number.
- The LEDs on the micro:bit show whether the line is currently detected by either the left, middle or right sensor.



What impact could intelligent vehicles have?

Amar Gohil, WMG

By adding sensors to future vehicles, it allows us to phase out the need for a human driver with the added benefit of the sensors being incredibly robust and being able to detect objects with much more detail and further away compared to humans.

We at WMG are bridging this gap between what we have on our roads now and where we want to be with connected autonomous vehicles in order to create much safer and accessible transport systems for everyone.

What have you learned?

Tinkering

Hardware and Software

Intelligent Vehicles

Coding

Doing a fair experiment

Future of Transport

Sensors

Make, Test, Evaluate, Improve

Variables

EXTENSIONS

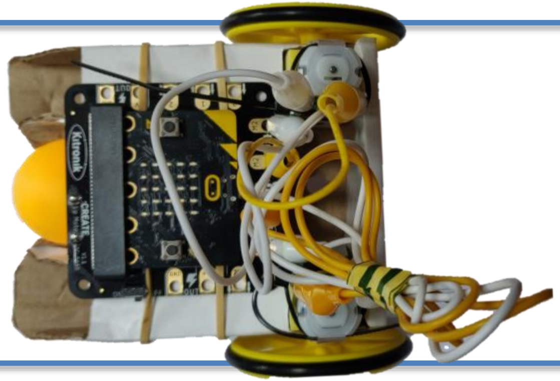
If you’ve flown through all those challenges and your car works well, here are some extensions!

If you have access to a computer, the Simple Robotics Kit handbook pages 4-5 and 18-23 can help you make new code for the micro:bit controller.

The code that is already on your microbit has been sent by email to your teachers. If you want to edit the code you’ve been using so far, ask them for the **hex file**.

Other extensions will work without access to a computer.

Going Wheely Far



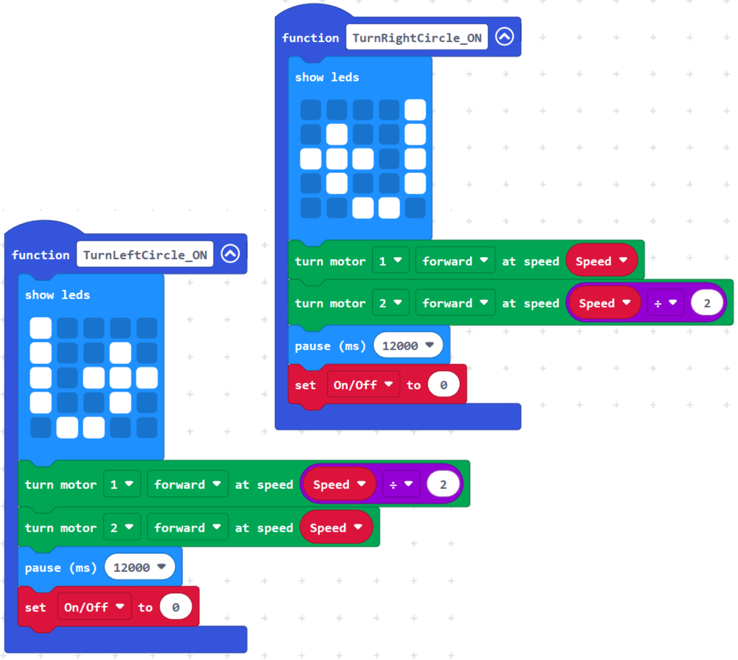
Extension Question

Measure how far the vehicle travels in 10 seconds in Drive Forwards and Drive Backwards mode to get a ‘real world’ speed for the vehicle.

Can you work out how long the ‘pause’ command would have to be in the code for your vehicle to drive exactly 1 metre forwards or 1 metre backwards?

If you have access to a computer – can you edit the code to add your new pause length?

Round and Round
Extension Question



Given the functions for turning left and turning right, and your calculations for a complete circle on p. 30-31, can you think of what a function for Driving in a Figure 8 would look like?

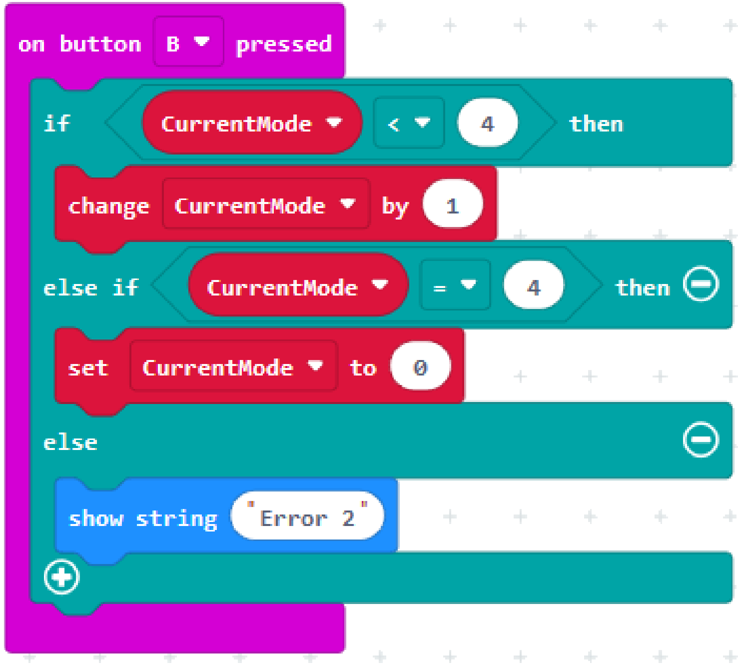
Which blocks would you need, and in what order? Can you think of a way to use the LED grid to make a symbol for a Figure 8?

If you have access to a computer – can you make this new function?

What should the figure 8 symbol look like?

ADDING A NEW MODE

Extension Question



If you had an idea for a new mode for the vehicle – how could you change the mode selection code to allow you to select the new mode?

Which parts of the code to the left would you need to change?

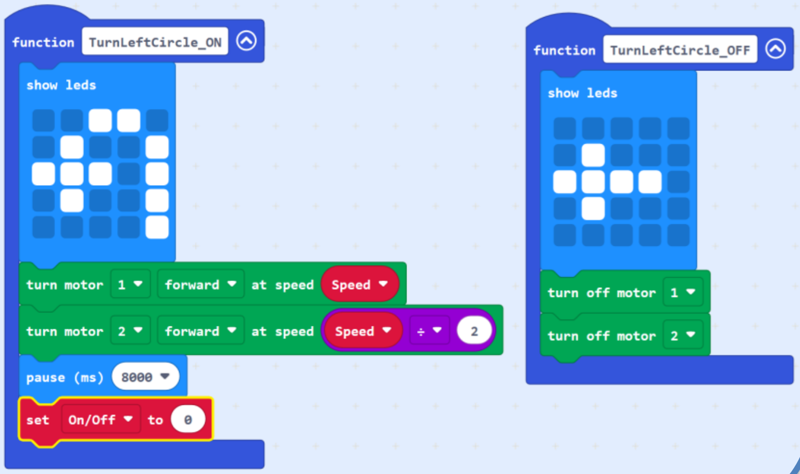
If you have access to a computer can you make the Figure 8 function form p. 57 and add a new mode to the selection menu?

TURN IN A CIRCLE
Extension Question

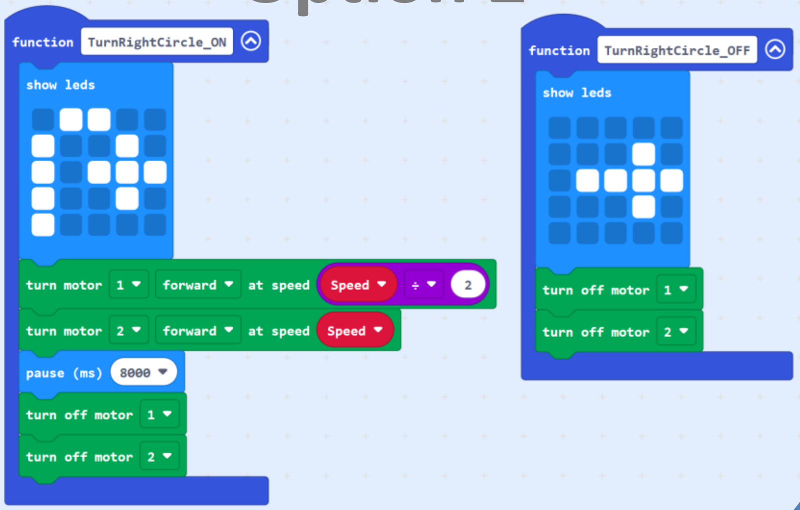
Why is Option 1 used instead of Option 2?

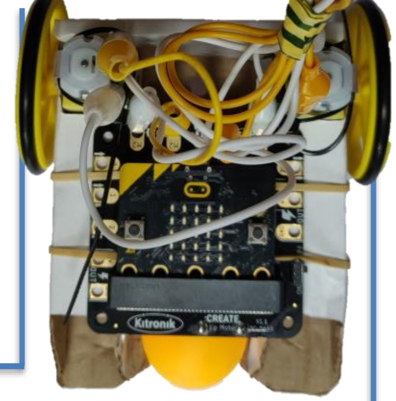
What is the difference between the two?

Option 1



Option 2





OBSTACLE COURSE

Extension Question

Using the real world speed you have calculated for your car you should be able to program it to drive specific distances and then move on to the next line of code. Using the amount of time needed to turn a full circle, you should be able to program the vehicle to turn a specific distance and then move on to the next line of code.

Could you make an obstacle course and then program your vehicle to navigate through it?