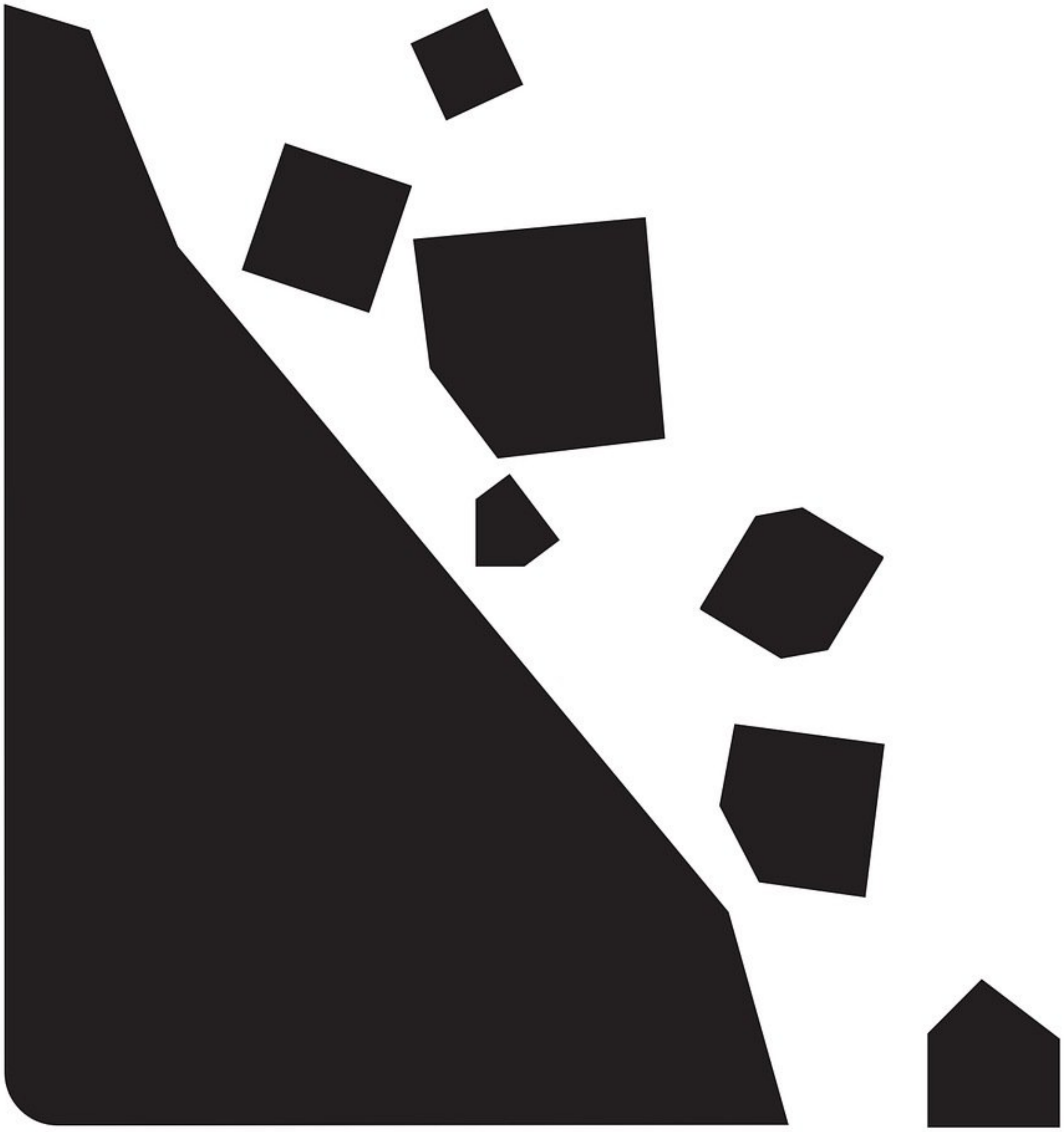
A - Debris on the Track

Rocks have fallen onto the line for the robot to follow, blocking its path. We need to make the program clever enough to not get stuck!



This document was generated on 2022-01-01 09:44:52 PM (MST).

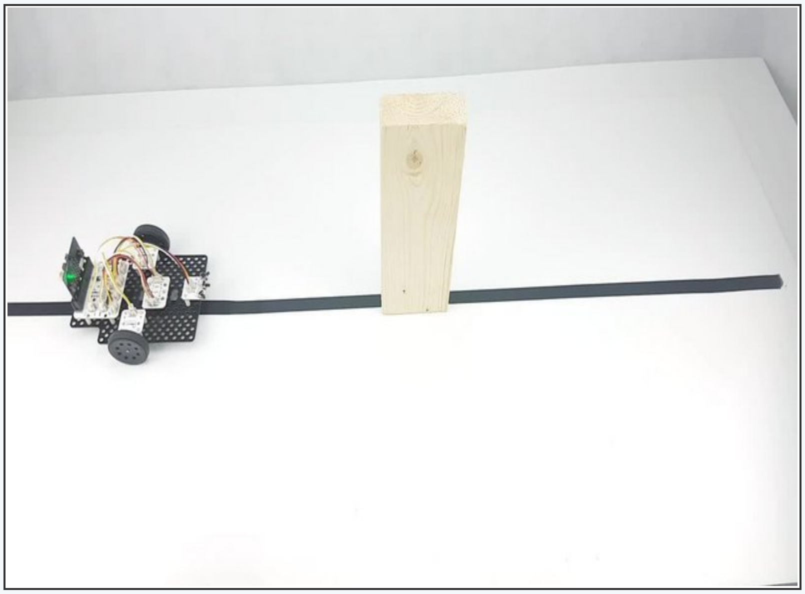
INTRODUCTION

Rocks have fallen onto the line for the robot to follow, blocking its path. We need to make the program clever enough to not get stuck!

Step 1

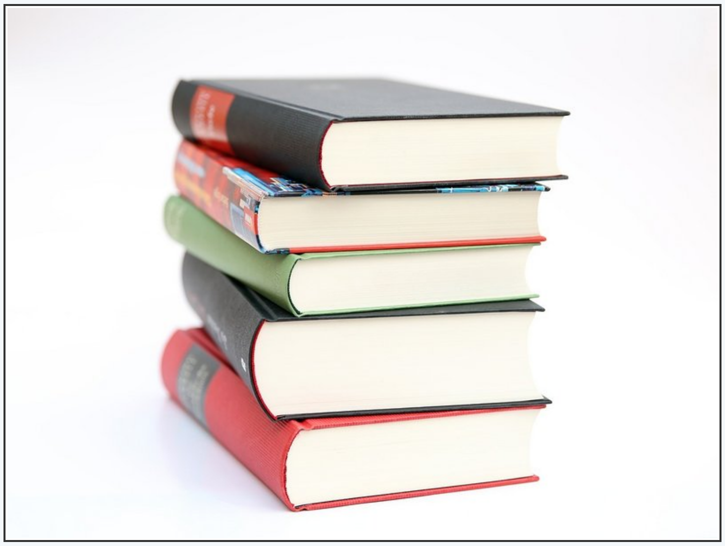
Obstacle Avoider

- In this lesson we're going to combine the **line follower** program with some **obstacle avoiding** code!
- This should allow the robot to easily drive **around** obstacles, and then **find the line again.**
- Hopefully yours should work like our example in the **video!**



Step 2

Make the Obstacle



- Let's put an **obstacle** on the track for the robot to avoid.

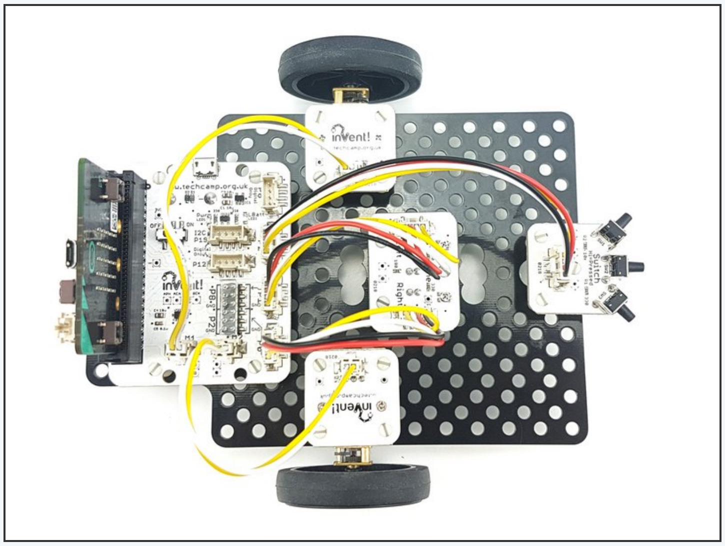
- You can use anything you like, as long as it is at least **4cm tall.**

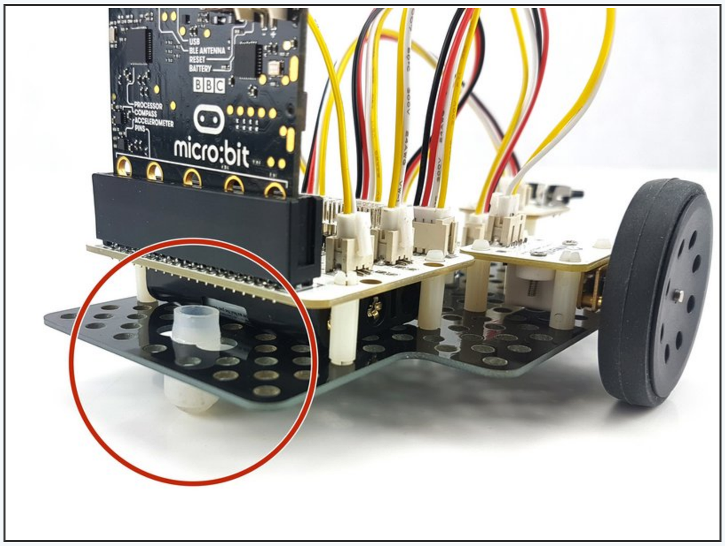
- Put it across the track **wherever you like!** However, having a **straight section** of track both sides of the obstacle will make it **much easier.**

- You can use **books**, **wooden blocks**, or **anything else** you have around. You might need to **tape it down** if its very light so the robot doesn't just push it around.

Step 3

Setup the Robot





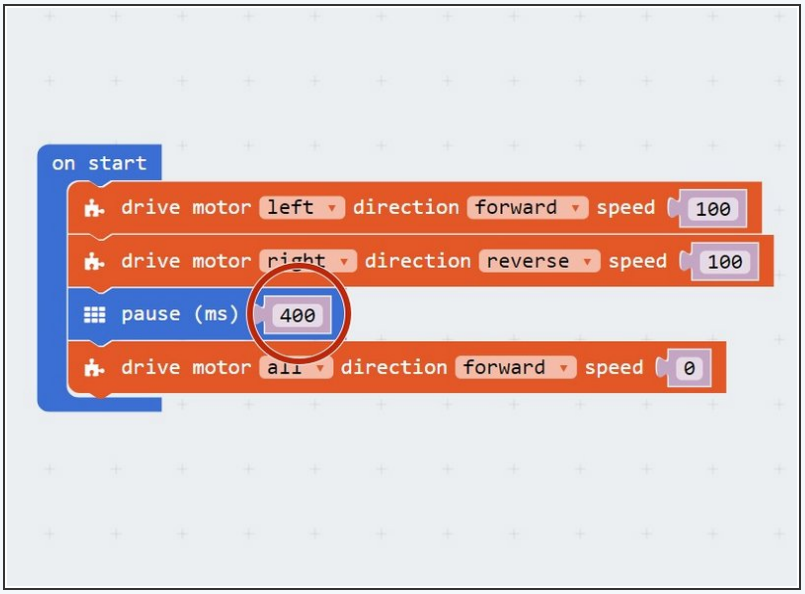
- We're going to use a **switch module** to detect when we hit the obstacle.
- Assemble your robot like the picture
- Make sure the switch is in the **middle!**

⚠ You will need to move the **trackball** to the back of the robot, underneath the main board, to keep it balanced.

Step 4

90 Degree Turn

- We're going to need to turn the robot **90 degrees** again to complete this challenge.
- Write the simple program in the picture and work out **how many milliseconds** you need to wait for your robot to turn **90 degrees.**
- We're not following the line anymore, so let's go at **full speed** to get round the obstacle as **fast as possible**.

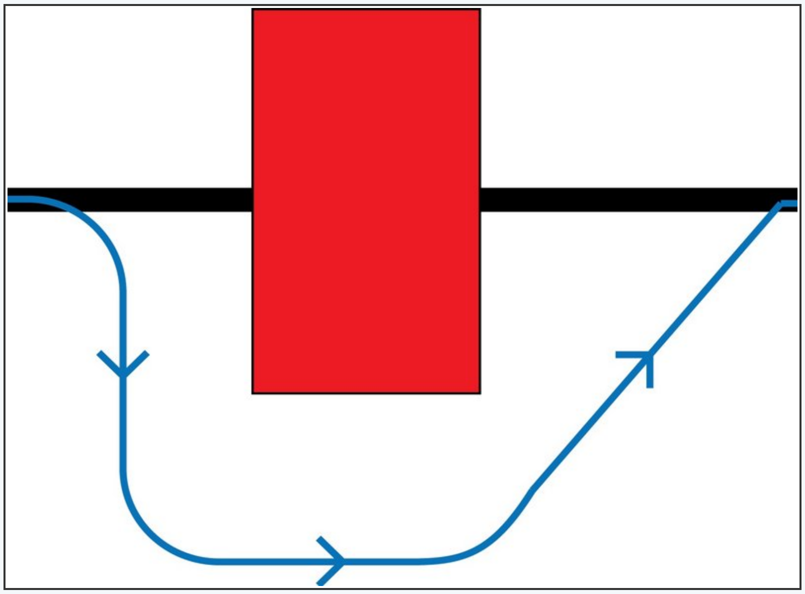


```
on start
    drive motor left direction forward speed 100
    drive motor right direction reverse speed 100
    pause (ms) 400
    drive motor all direction forward speed 0
```

Step 5

Avoiding Path

- We need to decide **how** we want the robot to move so it **avoids the obstacle.**
- You might need to **change** this slightly depending on the **size** and **shape** of your obstacle! When the switch is pressed we need to:
  - **Turn right** 90 degrees
  - Drive **forwards**
  - **Turn left** 90 degrees
  - Drive **forwards**
  - Turn left **45 degrees**
  - Drive forwards until the **right sensor** find the line again (reads as **0**)

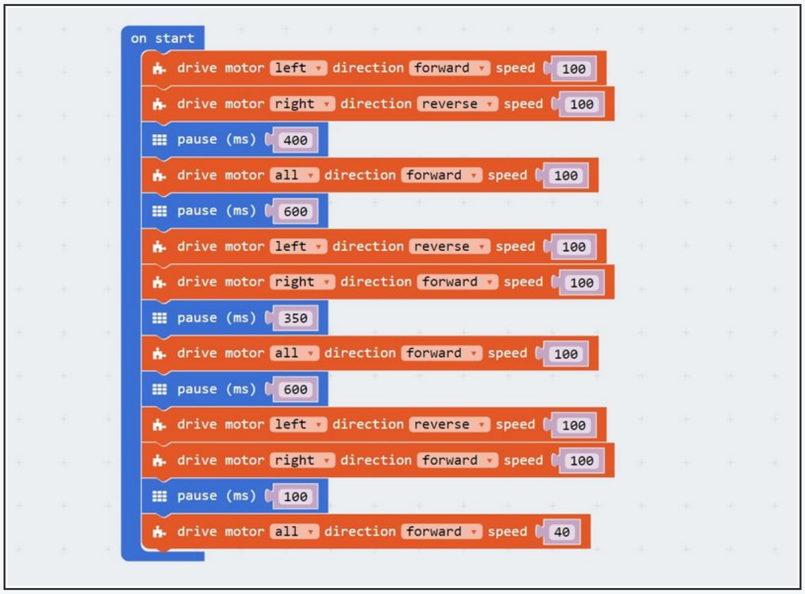


Step 6

Make the Moves

- Using the **wait time** you worked out earlier for turning 90 degrees, write a simple program to make your robot **drive around the obstacle.**
- Our code is just an example - you will need to **change** all of the times depending on the **size** of your obstacle!
- **Test** your code properly until your robot **reliably** drives around the obstacle

⚠ Driving at **45 degrees** to the line after the obstacle is **very important** - if we drive directly at the line, the robot might drive **back towards the obstacle** instead of continuing down the rest of the track.



```
on start
    drive motor left direction forward speed 100
    drive motor right direction reverse speed 100
    pause (ms) 400
    drive motor all direction forward speed 100
    pause (ms) 600
    drive motor left direction reverse speed 100
    drive motor right direction forward speed 100
    pause (ms) 350
    drive motor all direction forward speed 100
    pause (ms) 600
    drive motor left direction reverse speed 100
    drive motor right direction forward speed 100
    pause (ms) 100
    drive motor all direction forward speed 40
```

This document was generated on 2022-01-01 09:44:52 PM (MST).

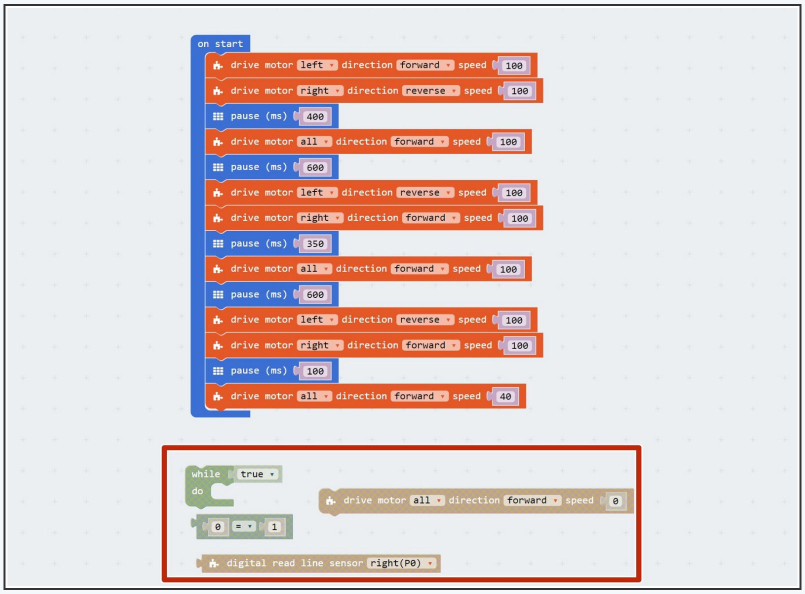
Step 7

Stop on the Line

- After the robot has driven around the obstacle, it should **drive forwards** until **one** of the sensors finds the line again.

- Use a **while loop** to wait until the **right sensor** reads **0** and then **stop the motors**, so the robot **stops** when it finds the line again.

- There are some **hint blocks** if you need them!



```
on start
    drive motor left direction forward speed 100
    drive motor right direction reverse speed 100
    pause (ms) 400
    drive motor all direction forward speed 100
    pause (ms) 600
    drive motor left direction reverse speed 100
    drive motor right direction forward speed 100
    pause (ms) 350
    drive motor all direction forward speed 100
    pause (ms) 600
    drive motor left direction reverse speed 100
    drive motor right direction forward speed 100
    pause (ms) 100
    drive motor all direction forward speed 40

while true
do

drive motor all direction forward speed 0

0 = 1

digital read line sensor right(P0)
```

Step 8

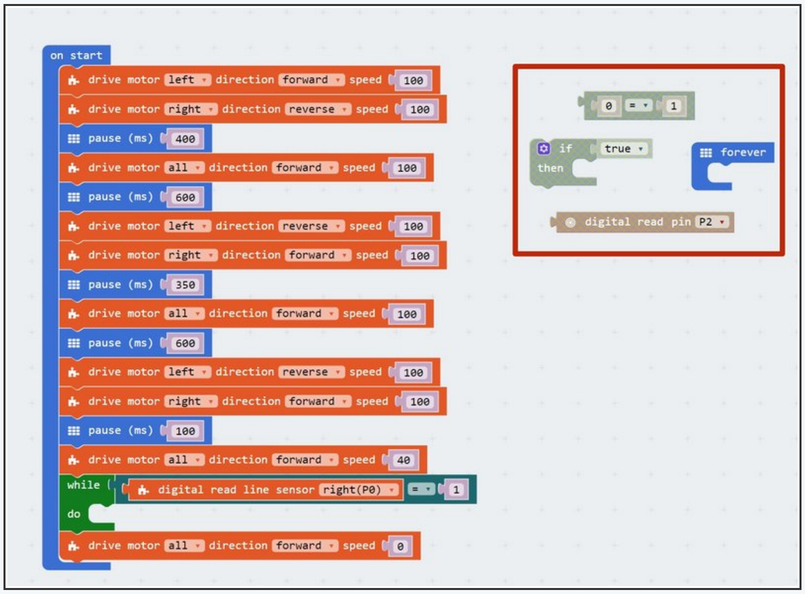
Add the Switch

- **Almost there!** Let's add an IF block so our sequence is only triggered when the switch (P2) is **pressed.**

- There are some **hint blocks** if you need them, but hopefully you are a Pro at using switches by now!

⚠ Don't forget, you'll need to transfer the sequence from the **on start** block to a **forever loop**.

ⓘ If your switch doesn't get pressed when the robot drives into the obstacle, make sure it isn't able to move - **use some tape or blutack to keep it still.**

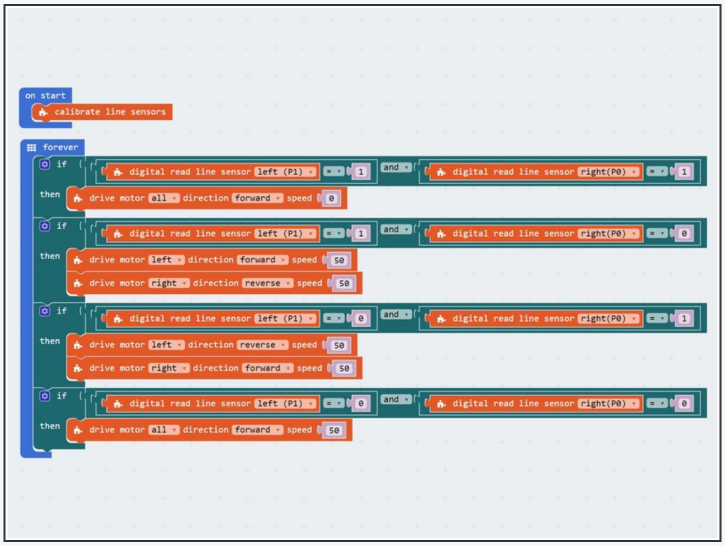


```
on start
    drive motor left direction forward speed 100
    drive motor right direction reverse speed 100
    pause (ms) 400
    drive motor all direction forward speed 100
    pause (ms) 600
    drive motor left direction reverse speed 100
    drive motor right direction forward speed 100
    pause (ms) 350
    drive motor all direction forward speed 100
    pause (ms) 600
    drive motor left direction reverse speed 100
    drive motor right direction forward speed 100
    pause (ms) 100
    drive motor all direction forward speed 40
    while ( digital read line sensor right(P0) = 1
    do
    drive motor all direction forward speed 0

0 = 1

if true
then

forever

digital read pin P2
```

This document was generated on 2022-01-01 09:44:52 PM (MST).

Step 9

Obstacle avoider & line follower



```
on start
    calibrate line sensors

forever
    if digital read line sensor left (P1) = 1 and digital read line sensor right(P0) = 1
    then
        drive motor all direction forward speed 0
    if digital read line sensor left (P1) = 1 and digital read line sensor right(P0) = 0
    then
        drive motor left direction forward speed 50
        drive motor right direction reverse speed 50
    if digital read line sensor left (P1) = 0 and digital read line sensor right(P0) = 1
    then
        drive motor left direction reverse speed 50
        drive motor right direction forward speed 50
    if digital read line sensor left (P1) = 0 and digital read line sensor right(P0) = 0
    then
        drive motor all direction forward speed 50
```

- Time to combine the **obstacle avoiding** code with our **line follower** code.
- Load up your **2 sensor line follower code** - it should look similar to the one in the picture.
- **Add** your obstacle avoiding sequence to the line follower code, and test it on the track - hopefully it drives round the obstacle and **finds the track again!**
- If it doesn't work, try **adjusting** the code and **keep testing** until it works really well.

Step 10 Combined robot with sparkles as well

- If you've finished all that, add your **Sparkle module** back on to the robot, and add the blocks back in to display the **line position** with the red/green Sparkles.
- When the robot is avoiding obstacles, make the Sparkles do something else so we know the robot is **driving round the obstacle.**
- They can do **anything you like:** flashing white, orange or something else entirely - **the more impressive the better!**

This document was generated on 2022-01-01 09:44:52 PM (MST).