

B - Rock Field Navigation 2

Using some switch sensors and clever coding, let's make your robot safely navigate the Martian rock fields!



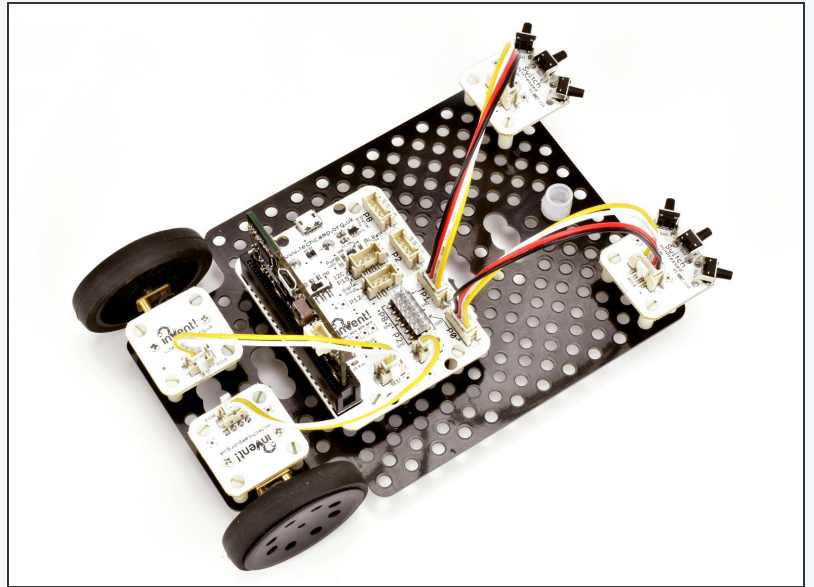
INTRODUCTION

Using some switch sensors and clever coding, let's make your robot safely navigate the Martian rock fields!

Step 1

Two Switches

- Hopefully your robot now makes it across the rock field fairly well - but only if the obstacle is **directly in front**.
- You might find it gets stuck with obstacles on the left and right - but we have **two** switch sensors, so we can **upgrade** the robot to fix that!
- Put your **other** switch module onto your robot, and **move the first one** so it looks like the picture.
- You should have 1 switch plugged into **P0**, and the other into **P1**.



Step 2

Left Switch

- Let's test the left sensor.
 - When the left switch is pressed, the robot should **reverse** and **turn right** to avoid the obstacle.
 - Try building the program in the picture and **test your program!**
 - Don't forget, you might need to **adjust** how far the robot **turns** and **reverses** for it to work well.
- i** Make sure you understand this program - we have **taken out** some of the `drive_motor` lines that aren't needed, as sometimes one of the motors is **already going in the direction we want**.


```
1 # Start your code below here!
2
3 drive_motor(0,100)
4
5 while True:
6     if pin1.read_digital()==1:
7         drive_motor(0,-100) # Reverse
8         sleep(100)
9         drive_motor(1,100) # Turn right
10        sleep(200)
11        drive_motor(0,100) # Forwards
```

Step 3

Comments

- You might have noticed something new in the last program - the text explaining what the code was doing was an example of **comments**.
- Comments are very important - they **remind you** what your code is doing, but also **help others** read and understand your code.
- You can add a comment using a hash symbol (#). You can then type **anything you like** afterwards, and it will be ignored by the robot.
- Try **adding some more comments** to your code now!

```
1 # Invent! Code End
2 # Start your code below here!
3
4 drive_motor(0,100)
5
6 while True:
7     if pin1.read_digital()==1:
8         drive_motor(0,-100) # Reverse
9         sleep(100)
10        drive_motor(1,100) # Turn right
11        sleep(200)
12        drive_motor(0,100) # Forwards
```



Step 4

Right Switch

- Now let's test the **right switch**.
- **Change the program** so it checks the **right sensor**, and then turns **left** to avoid the obstacle!

```
13 # Invent! Code End
14 # Start your code below here!
15
16 drive_motor(0,100)
17
18 while True:
19     if pin0.read_digital()==1:
20         drive_motor(0,-100) # Reverse
21         sleep(100)
22         drive_motor(2,100) # Turn Left
23         sleep(200)
24         drive_motor(0,100) # Forwards
25
26
27
28
```


Two Sensor Evasion

Challenge!



```
12 def analog_read_line(s): v=p0() if s==0
13 # Invent! Code End
14 # Start your code below here!
15
16 drive_motor(0,100)
17
18 while True:
19     if pin0.read_digital()==1:
20         # Add some code here
21
22     if pin1.read_digital()==1:
23         # Add some code here
24
25
26
27
```

- **Change** your one switch code from the last challenge so **both** switches are being checked.
- Make sure that when the **left switch** is hit, the robot reverses and then turns to the **right**, (away from the obstacle) and when the **right switch** is hit it reverses and then turns to the **left**.
- **Test** your program properly in the rock field, and try to adjust the delay times so it works as well as possible!
- There's some example code in the picture if you need a hint - but it is **missing a lot of lines** you will need to fill in!

Step 6

Feedback and Safety

Buzzer

- Let's add some more feedback to the robot so we know what its doing.
- **Add the LED and buzzer modules** to your robot, and plug them into the last two pins.
- Extend your program so that:
 - When driving forwards, the **green LED** is on.
 - When an obstacle is hit and the robot is reversing and turning, the **red LED** is on.
 - When the robot is reversing, the **buzzer** should **beep** to warn people to get out of the way!
 - If you can, try to make the buzzer **beep several times** as the robot is reversing instead of being on all the time.

