

## A - Rock Field Navigation

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

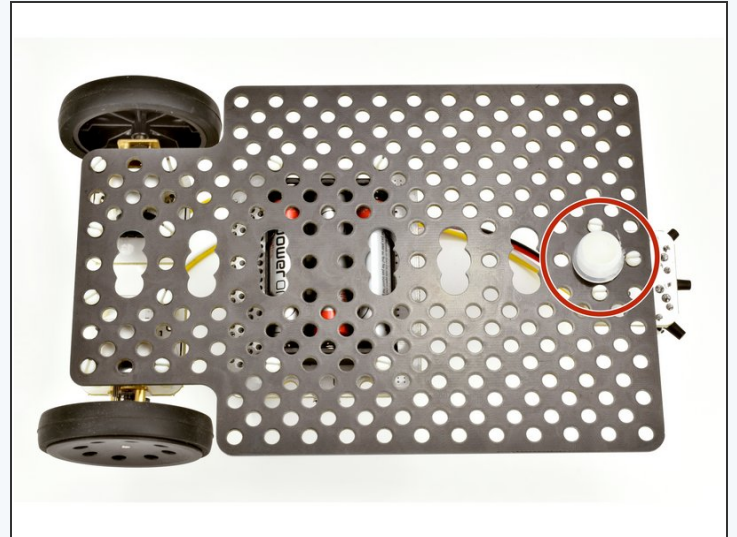
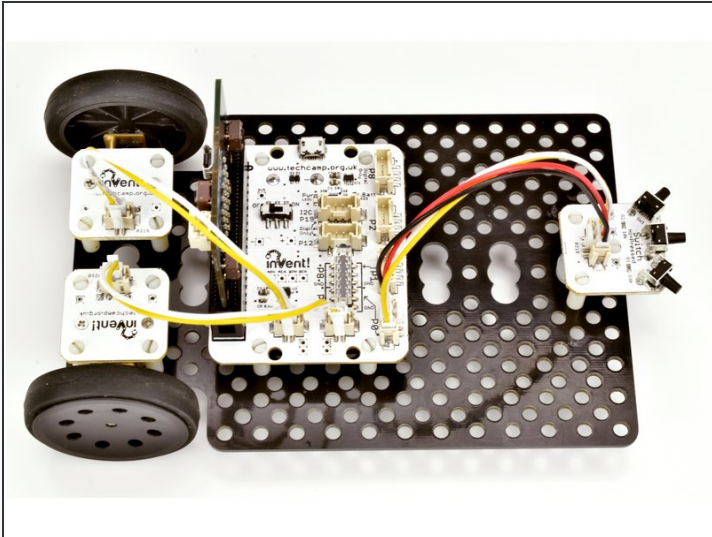


# INTRODUCTION

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

## Step 1

### Setup Your Robot




- Setup your robot like in the picture!
- Make sure the switch module is at the front, and in the **middle** of the baseboard.
- The left motor should be plugged into **M1**, the right motor into **M2** and the switch into **P0**
- The trackball should be **directly under the switch!**

## Step 2

### Detect an Obstacle

- The Martian surface is very rocky - there are lots of things the robot would get stuck on if it just drove forward. We are going to use switches to **detect** obstacles, and try to drive around them.
- For now, let's make a program to drive forward, and **stop** if we hit an obstacle.
- Have a look at the sample program in the picture - **add some lines inside the if statement, to make the robot stop** when the switch hits an obstacle.
- **Test your program** using **your hand** as an obstacle!

```
10 while(p0()>p[7]and p1()>p[6]):d(0 if
11 def digital_read_line(s): return 1 if (s
12 def analog_read_line(s): v=p0()if s==0 e
13 # Invent! Code End
14 # Start your code below here!
15
16 while True:
17     drive_motor(0,100)
18     if pin0.read_digital()==1:
19         # Stop the motors!
20
21
22
23
24
25
```





## Step 3

### Simple Evasion

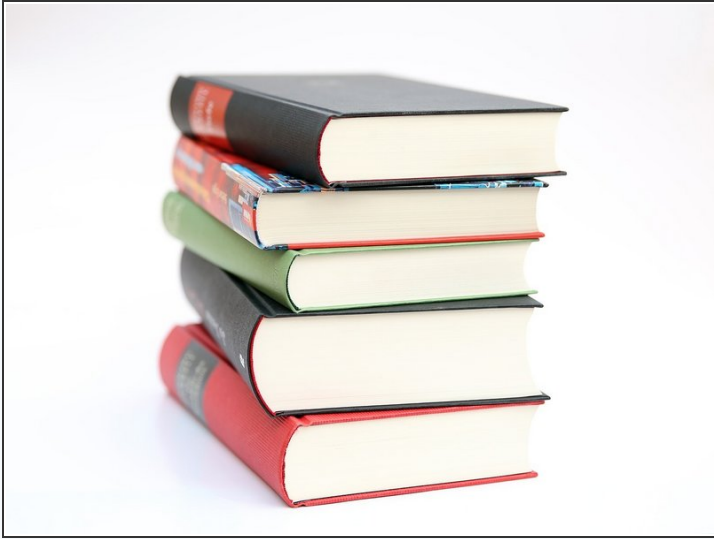
- Now we can detect obstacles, let's make our robot **turn out of the way** and keep trying to go **forward**:
- If the switch is **pressed** (1), we must have hit an obstacle
- When we hit an obstacle, **turn**, then drive **forward** again
- When driving forwards, **always check** to see if we have hit another obstacle! (We are doing this already, so long as the if statement is inside a **while True: loop**).
- Edit your program and **test it out with your hand**!

```
12 def analog_read_line(s): v=p0()if s==0 e
13 # Invent! Code End
14 # Start your code below here!
15
16 while True:
17     drive_motor(0,100)
18     if pin0.read_digital()==1:
19         drive_motor(1,100)
20         drive_motor(2,-100)
21         sleep(200)
22         drive_motor(0,100)
23
24
25
26
27
```





## Setup the Rock field



- Now let's **make a rock field** on your Mars surface - you can use anything, so long as it is solid and tall enough to touch the switches when they are on your robot. Check out the pictures for some ideas!
- Make sure there is **enough room between your obstacles** so your robot can still get through.
- You might find your robot pushes light obstacles out of the way - you can always **tape them down** if you want.
- **Test** your evasion program in the rock field - what do you think will happen? Will it make it to the other side?



## Step 5

### Better Evasion

- You might find that your robot **gets stuck** when it hits a real obstacle, as it doesn't **reverse** a bit first when it hits something.
- For this challenge, change your program so when it hits an obstacle, your robot **reverses slightly, turns a small amount**, then goes **forward** again.
- **Test it out** in your rock field - you will need to **adjust** how far it **reverses** and **how much it turns** to get it working well!



## Step 6

### Evasion Race

- For an extra challenge, let's add some **feedback** to our robot so it can tell us what its doing.
- **Add the LED module** to your robot and plug it into two pins.
- Add code to 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.

