techcamp.org.uk

D - Change the Speed

What if we want to change the speed our robot is following the line, without stopping it and reprogramming? Let's use variables and switches to do this.

This document was generated on 2021-12-25 06:48:57 AM (MST).

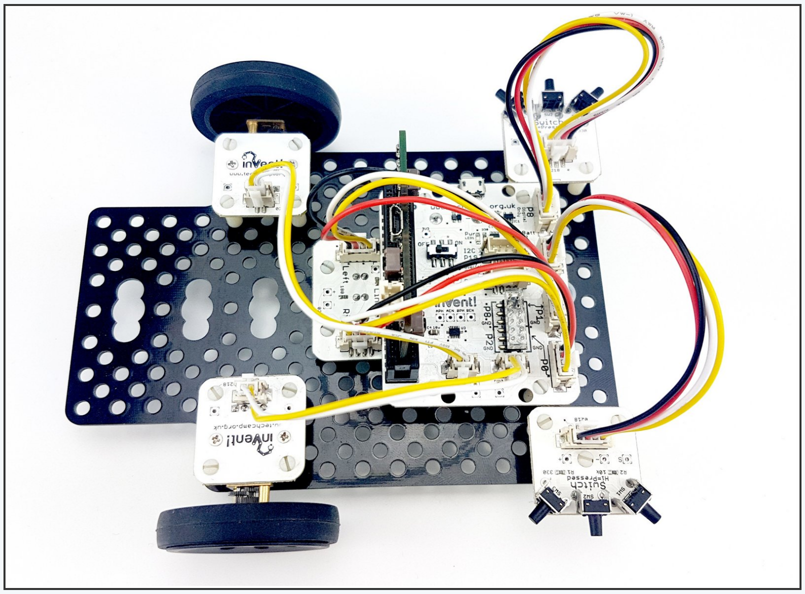
INTRODUCTION

What if we want to change the speed our robot is following the line, without stopping it and reprogramming? Let's use variables and switches to do this.

Step 1

Add the Switches

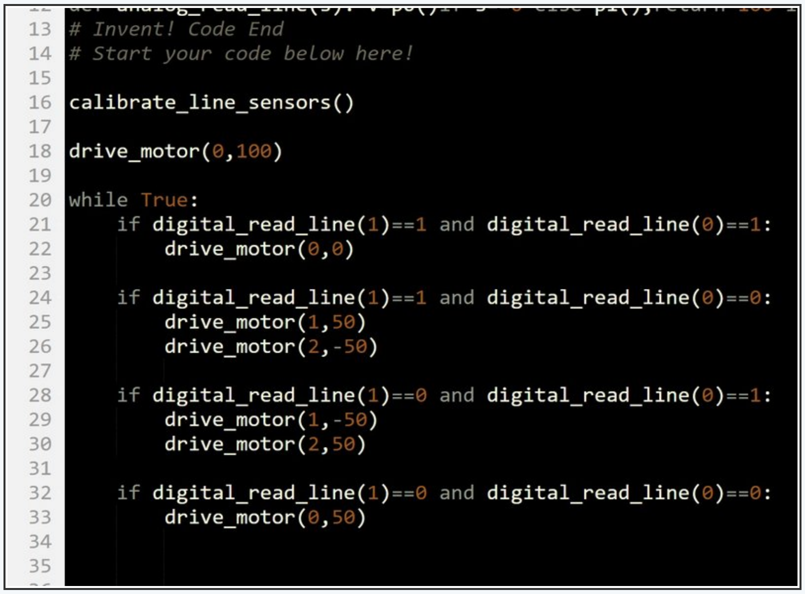
- We're going to need **two** switches - one to **increase** the speed, and one to **decrease** the speed.
- Add the **two switch modules** to your robot, and plug them into **P2** and **P8.**
- You can remove the Sparkles for now - **setup the robot just like the picture**



Step 2

Two Sensor Follower Program

- **Load** your two sensor line follower program from the last lesson.
- **Remove** all the Sparkle code for now - we don't have the sparkles anymore.
- Your program should look like the **picture** - your speeds might be **different**, depending on what works best for you.



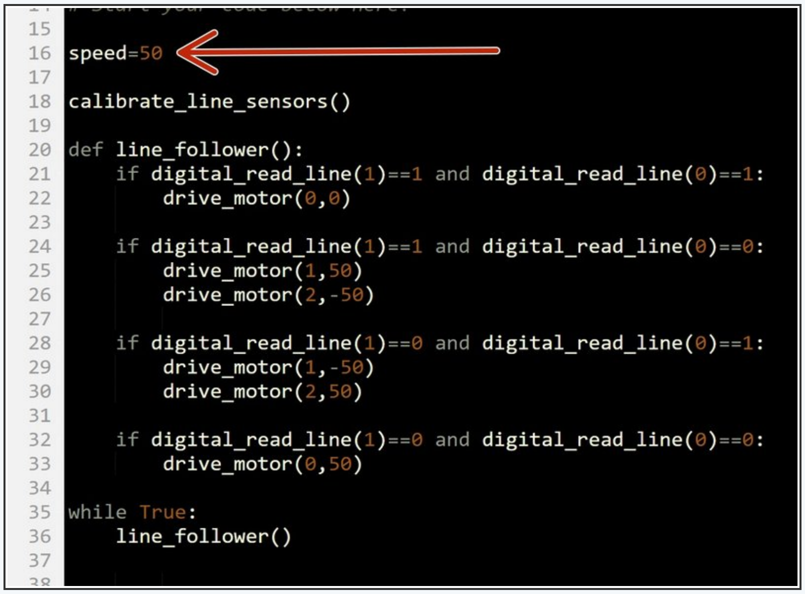
```
# Invent! Code End
# Start your code below here!

calibrate_line_sensors()

drive_motor(0,100)

while True:
    if digital_read_line(1)==1 and digital_read_line(0)==1:
        drive_motor(0,0)

    if digital_read_line(1)==1 and digital_read_line(0)==0:
        drive_motor(1,50)
        drive_motor(2,-50)

    if digital_read_line(1)==0 and digital_read_line(0)==1:
        drive_motor(1,-50)
        drive_motor(2,50)

    if digital_read_line(1)==0 and digital_read_line(0)==0:
        drive_motor(0,50)
```

This document was generated on 2021-12-25 06:48:57 AM (MST).

Step 3

Add a Variable

- Remember variables? Here's a quick **reminder** of what we can do with them:
  - **Call** them anything we like (variable **name**)
  - **Store** any number we like inside them (variable **contents**)
  - **Change** the contents at any time (add, subtract, multiply, divide and so on)
  - **Access** the contents at any time, so long as we know the **name** of the variable.
  - **Add** a new variable called **speed** at the start of the program, which is equal to 50

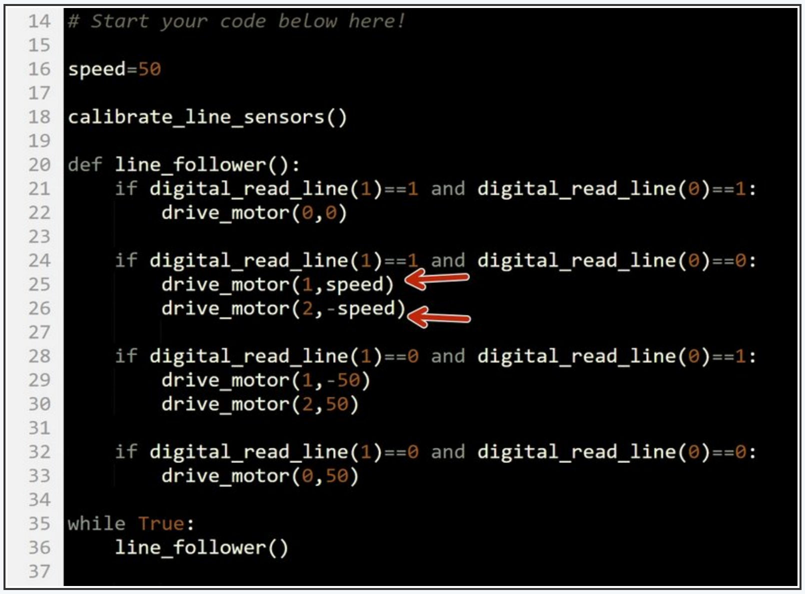


```
# Start your code below here!

speed=50

calibrate_line_sensors()

def line_follower():
    if digital_read_line(1)==1 and digital_read_line(0)==1:
        drive_motor(0,0)

    if digital_read_line(1)==1 and digital_read_line(0)==0:
        drive_motor(1,50)
        drive_motor(2,-50)

    if digital_read_line(1)==0 and digital_read_line(0)==1:
        drive_motor(1,-50)
        drive_motor(2,50)

    if digital_read_line(1)==0 and digital_read_line(0)==0:
        drive_motor(0,50)

while True:
    line_follower()
```

Step 4

Use the Variable

- Let's use the variable we have just created to **set the motor speeds!**
- **Replace** all the motor speeds in the motor blocks with the **speed variable.** We've done the first one for you!
- **Program** your robot and **test** to make sure it still works correctly.



```
# Start your code below here!

speed=50

calibrate_line_sensors()

def line_follower():
    if digital_read_line(1)==1 and digital_read_line(0)==1:
        drive_motor(0,0)

    if digital_read_line(1)==1 and digital_read_line(0)==0:
        drive_motor(1,speed)
        drive_motor(2,-speed)

    if digital_read_line(1)==0 and digital_read_line(0)==1:
        drive_motor(1,-50)
        drive_motor(2,50)

    if digital_read_line(1)==0 and digital_read_line(0)==0:
        drive_motor(0,50)

while True:
    line_follower()
```

Step 5

Increase the speed

- Let's use the **first switch** connected to **P2** to **increase** the speed.

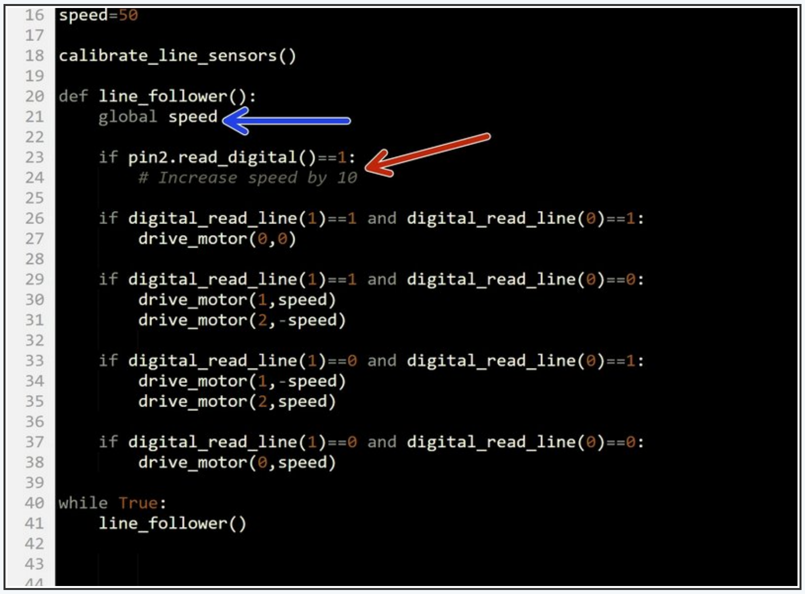
- Add an **IF statement** right at the top of the loop to **check** the switch.

- We also need to add the line **global speed** at the top of the function. Speed is a variable that is created **outside** of the function - we can access it without this line, but to **change** it we need to tell the robot it is **global**, which just means it has already been created outside of the function.

- If the switch is pressed, **increase speed by 10.**

- There is a **hint** if you need it!

- **Test** out the program - can you work out **what is wrong?**



```
speed=50

calibrate_line_sensors()

def line_follower():
    global speed

    if pin2.read_digital()==1:
        # Increase speed by 10

    if digital_read_line(1)==1 and digital_read_line(0)==1:
        drive_motor(0,0)

    if digital_read_line(1)==1 and digital_read_line(0)==0:
        drive_motor(1,speed)
        drive_motor(2,-speed)

    if digital_read_line(1)==0 and digital_read_line(0)==1:
        drive_motor(1,-speed)
        drive_motor(2,speed)

    if digital_read_line(1)==0 and digital_read_line(0)==0:
        drive_motor(0,speed)

while True:
    line_follower()
```

Step 6

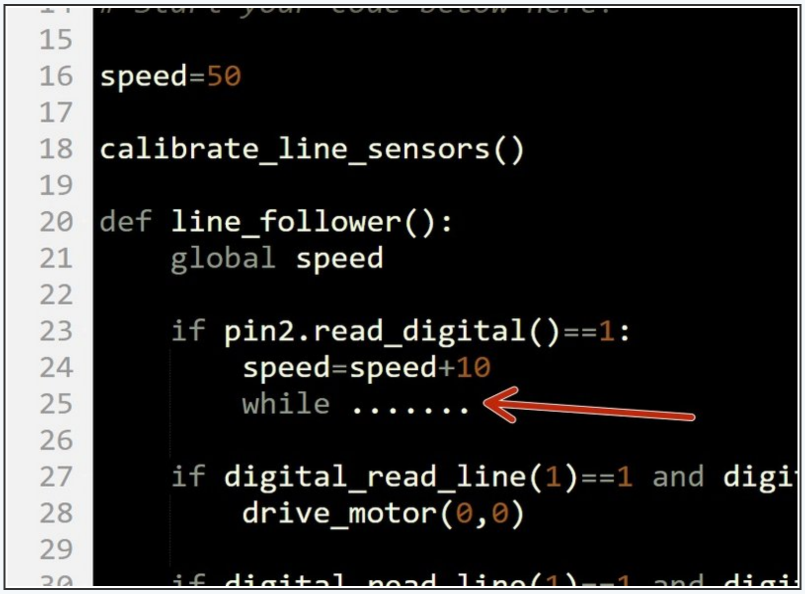
Wait for Release

- Remember last time we used a switch to change something? We had to add something else so it didn't change **too fast!**

- We want to use a **while loop** to wait until the switch is not pressed anymore, so we only increase the speed **once** each time the switch is pressed.

- Add a while loop **after** you increase the speed by 10 to fix this.

- **Test it out** - make sure it works properly now!

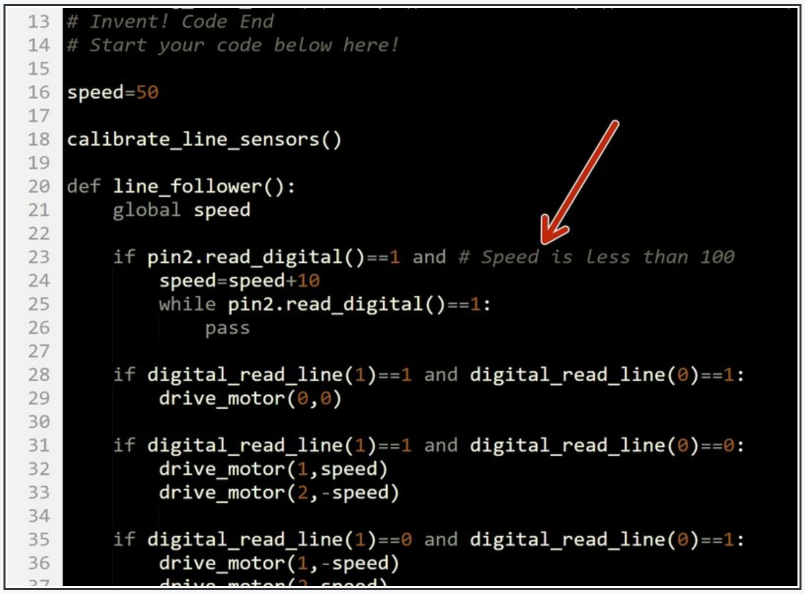


```
# Start your code below here!

speed=50

calibrate_line_sensors()

def line_follower():
    global speed

    if pin2.read_digital()==1:
        speed=speed+10
        while .......

    if digital_read_line(1)==1 and digi
        drive_motor(0,0)

    if digital_read_line(1)==1 and digi
```

Step 7

Limit the Speed

- The motor blocks **cannot have a speed of over 100**, so we need to make sure **speed is never more than 100!**
- To do this, let's edit the IF statement that checks the switch.
- Change the condition so that it checks if P2 is 1, **AND** speed is less than **(<)** 100.
- Test it out!



```
# Invent! Code End
# Start your code below here!

speed=50

calibrate_line_sensors()

def line_follower():
    global speed

    if pin2.read_digital()==1 and # Speed is less than 100
        speed=speed+10
        while pin2.read_digital()==1:
            pass

    if digital_read_line(1)==1 and digital_read_line(0)==1:
        drive_motor(0,0)

    if digital_read_line(1)==1 and digital_read_line(0)==0:
        drive_motor(1,speed)
        drive_motor(2,-speed)

    if digital_read_line(1)==0 and digital_read_line(0)==1:
        drive_motor(1,-speed)
```

Step 8

Decrease the Speed

- Now its **up to you!**
- Add some more code to check the **other switch**, and **decrease** the speed by 10 each time it is pressed.
- This time, you will need to make sure that speed is only **decreased** if it is **more than 0.**

Step 9

One switch only!

- This is a **hard** extension challenge, so don't worry if you find it difficult!
- Can you change the code so only **one switch** is needed?
- The speed should **increase** with a **short** press, and **decrease** with a **long** press.

Step 10

Speed change with sparkles

- If you're feeling really clever, **add the Sparkle code back in** once you've got rid of one switch!
- For super advanced coders only - can you change the **brightness** of the Sparkles depending on the **speed of the robot?** For example, at **maximum speed (100)** they should be as **bright as possible**, and at **0** speed they should be **off**.