

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.



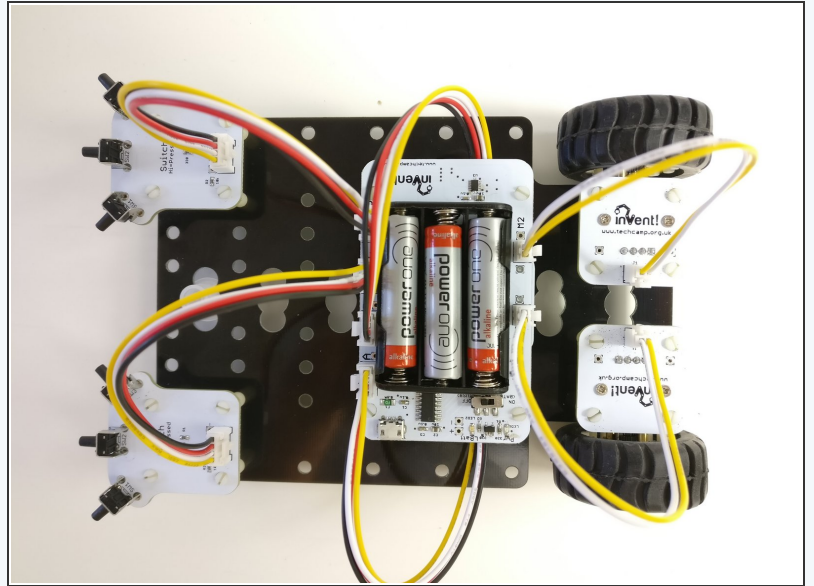
INTRODUCTION

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

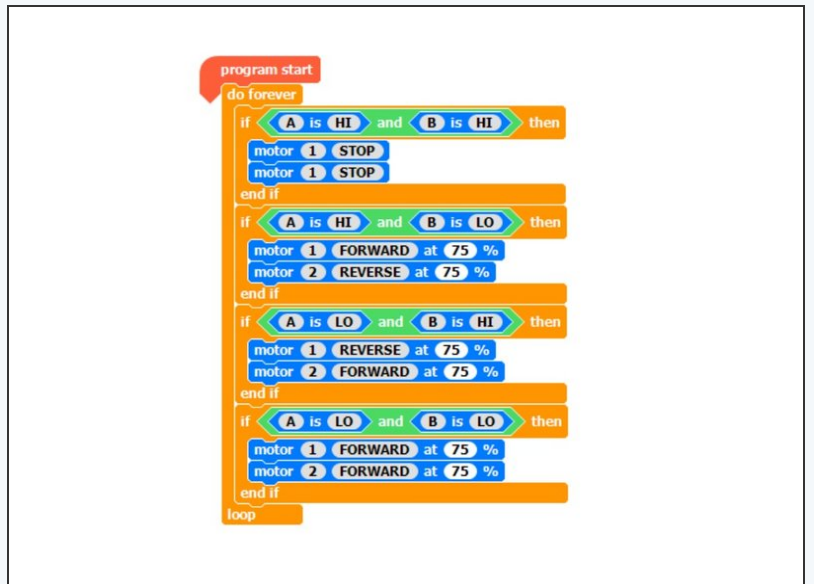
- We're going to need **two** switches - one to **increase** the speed, and one to **decrease** the speed.
- Add the **two switch modules** to your robot, and plug them into **C** and **D**.



Step 2

Two Sensor Follower Program

- **Load** your two sensor line follower program from the last lesson.
- **Remove** all the Sparkle blocks - we **don't have space** to plug the Sparkles in anymore!
- Your program should look like the **picture** - your speeds and waits might be **different**, depending on what works best for you.



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**
- Right at the **start** of your program, **let speed = 50**

The screenshot shows the Crumble programming environment. On the left, the 'Variables' palette is open, and a new variable named 'speed' has been created and is circled in red. On the right, the program logic is shown. A blue arrow points to the 'let speed = 50' block at the beginning of the program. The program starts with a 'program start' block, followed by 'let speed = 50'. A 'do forever' loop contains four conditional blocks: 1) 'if A is HI and B is HI then' with 'motor 1 STOP' and 'motor 2 STOP'; 2) 'if A is HI and B is LO then' with 'motor 1 FORWARD at 75 %' and 'motor 2 REVERSE at 75 %'; 3) 'if A is LO and B is HI then' with 'motor 1 REVERSE at 75 %' and 'motor 2 FORWARD at 75 %'; 4) 'if A is LO and B is LO then' with 'motor 1 FORWARD at 75 %' and 'motor 2 FORWARD at 75 %'.

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. **What speed** will the motors be going at?

The screenshot shows the same program logic as in Step 3, but with modifications. A red arrow points to the 'motor 1 FORWARD at speed %' block in the second conditional statement, indicating that the value '75' has been replaced with the variable 'speed'. The rest of the program logic remains the same.

Step 5

Increase the speed

- Let's use the **first switch** connected to **C** to **increase** the speed.
- Add an **IF block** right at the top of the program to **check** the switch.
- If the switch is pressed, **increase speed by 10**.
- There are some **hint blocks** if you need them!
- **Test** out the program - can you work out **what is wrong?**

The screenshot shows the Scratch programming environment. On the left, the 'Script' area contains a 'let speed = 0' block, followed by 'increase by 1' and 'decrease by 1' blocks. A 'hint block' is shown in a blue box, containing 'C is HI' and 'increase speed by 10'. On the right, the 'do forever' loop contains several 'if' blocks. A red arrow points to the first 'if' block, which is currently empty. The subsequent 'if' blocks contain conditions like 'A is HI and B is HI', 'A is HI and B is LO', 'A is LO and B is HI', and 'A is LO and B is LO', each followed by motor control blocks and a 'wait 100 milliseconds' block.

Step 6

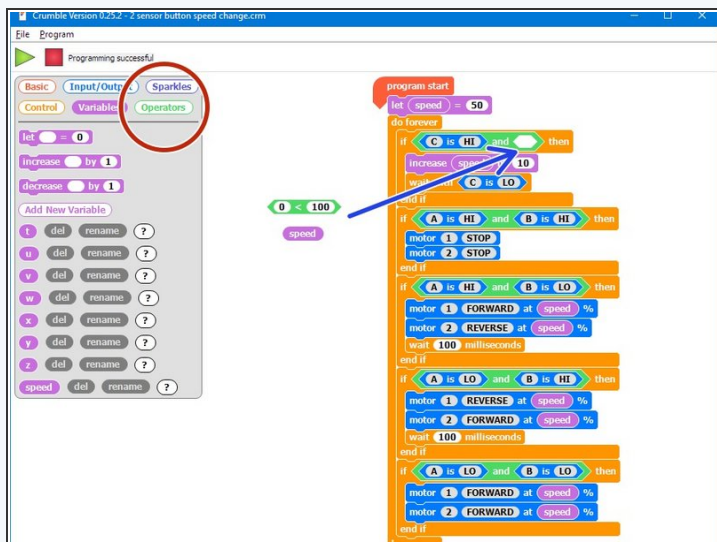
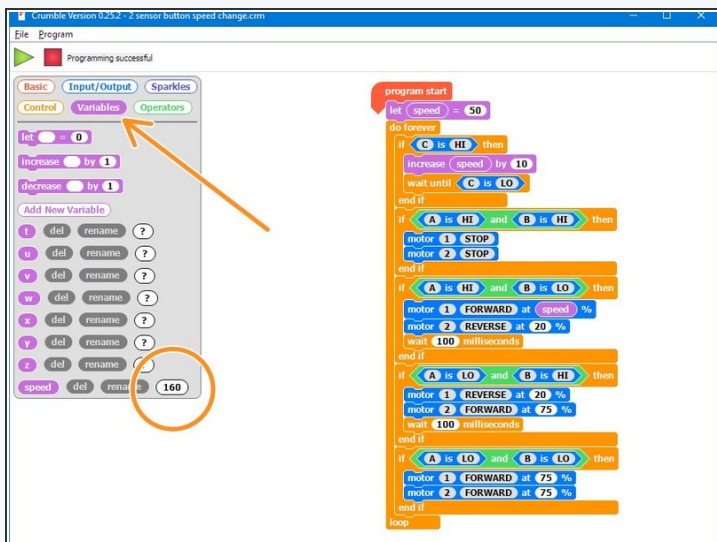
Wait Until

- 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 **wait until** the switch is not pressed anymore, so we only increase the speed **once** each time the switch is pressed.
- Add a **wait until** block **after** you increase speed by 10 to fix this.
- **Test it out** - make sure it works properly now!

The screenshot shows the same Scratch programming environment as in Step 5. The 'do forever' loop now includes a 'wait until C is LO' block (highlighted with a red arrow) immediately following the 'increase speed by 10' block within the 'if C is HI' condition. This modification ensures that the speed is only increased once per switch press.

Step 7

Limit the Speed



- You may have noticed that if you press the switch **lots** of times, **strange things** start to happen when the speed goes **over 100**.
- Program your robot and **keep it plugged in** - watch the value of speed on the **variables screen** and see what happens when it goes **over 100**.
- 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 block that checks the switch.
- Change the condition so that it checks if C is HI, **AND** speed is less than (<) 100.
- You can find the **less than** operator, < , in the **operators** menu. We've started it for you - **add** the rest of the blocks and **test it out!**

Step 8

Decrease the Speed

Challenge!



```
program start
let speed = 50
go forever
if C is HI and speed < 100 then
  increase speed by 10
wait until C is LO
end if
if A is HI and B is HI then
  decrease by 10
  speed
wait until
end if
if A is HI and B is LO then
  motor 1 FORWARD at speed %
  motor 2 REVERSE at speed %
wait 100 milliseconds
end if
if A is LO and B is HI then
  motor 1 REVERSE at speed %
  motor 2 FORWARD at speed %
wait 100 milliseconds
end if
if A is LO and B is LO then
  motor 1 FORWARD at speed %
  motor 2 FORWARD at speed %
wait 100 milliseconds
end if
```

- Now its **up to you!**
- Add some more blocks 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**.
- We've given you all the blocks you need - just **put them in the right order!**

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.

Extension Challenge!



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**.

