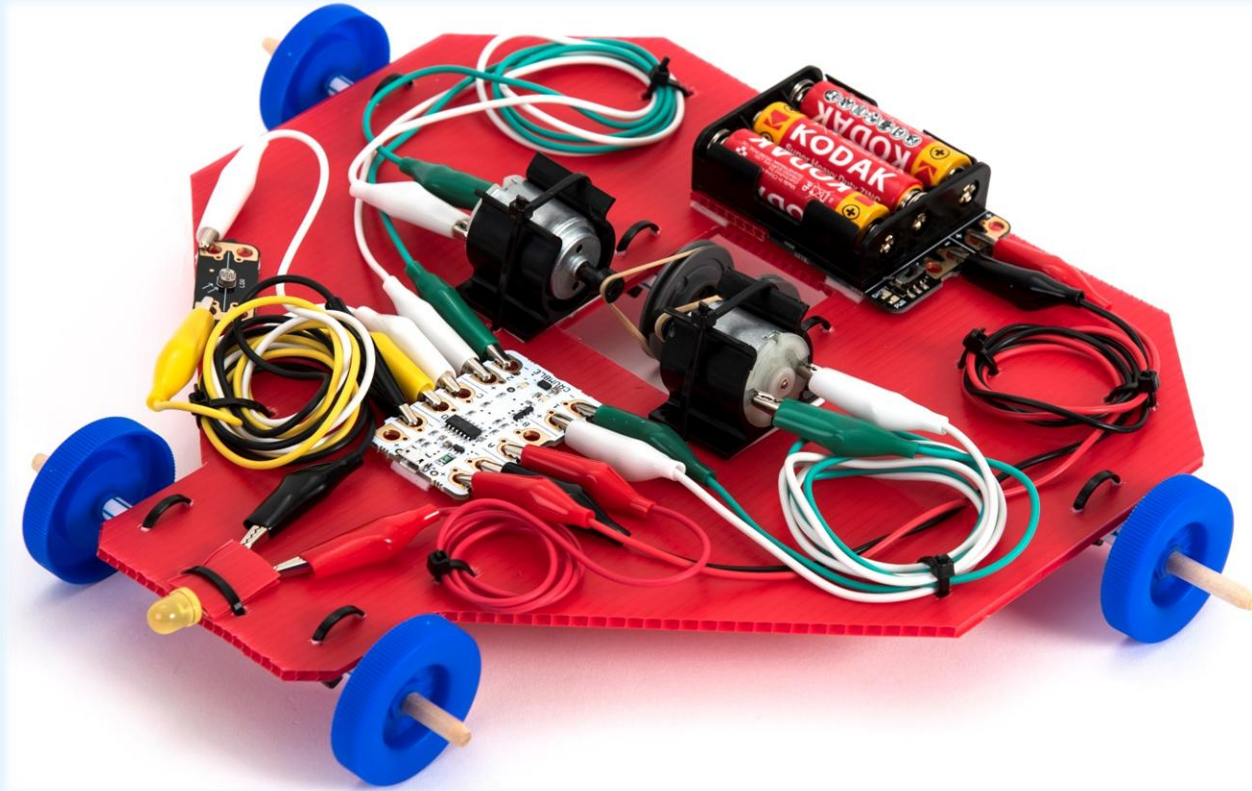


Crumble Robotic Vehicle



Extension activities



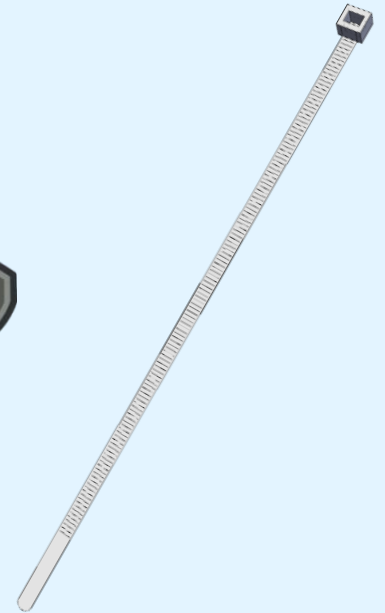
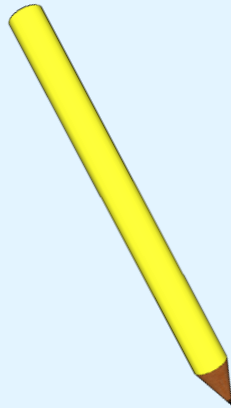
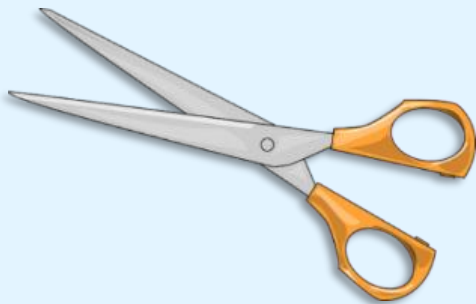
Learning Objectives

- Computing – use sequence, selection, repetition, variables, inputs and outputs.
- Electricity – compare and give reasons for variations in how components function.
- Forces – identify the effects of friction that act between moving surfaces.
- D&T – design, make, evaluate and improve products.



Work Safely

Look at the tools and equipment. Can you spot any potential hazards?

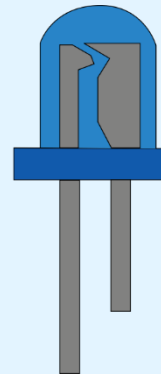
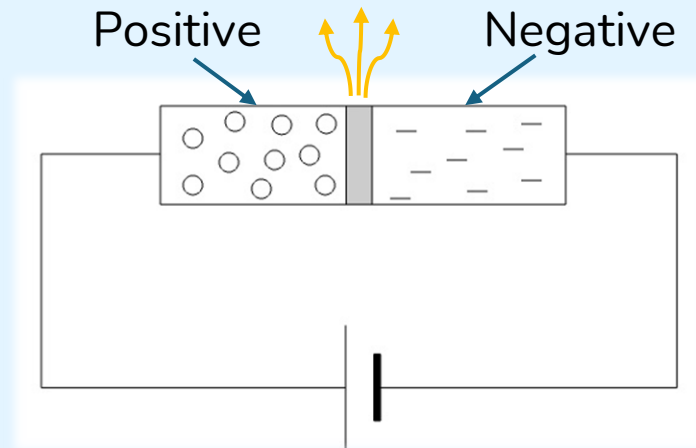


Can you think of ways to reduce the risks?



How a light emitting diode (LED) works

- An LED contains a special material called a semiconductor.
- You can think of it as having two sides, one of which contains positive particles, and the other negative particles.
- When you connect it up the right way, the particles move towards each other.
- When they meet, they join together and give out light.
- A traditional bulb makes light by heating a wire until it glows.
- The LED wastes much less power as heat.



LED

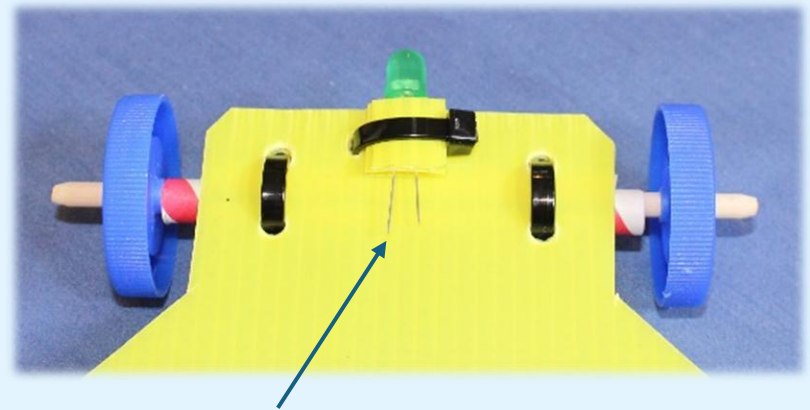
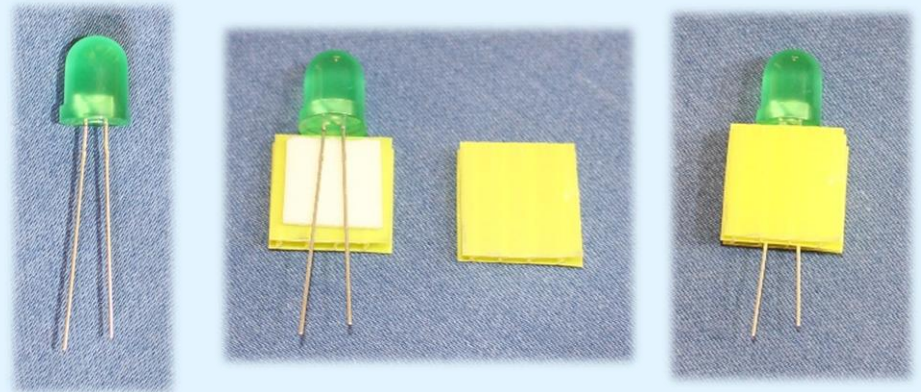


Traditional bulb



Fit an LED headlight

- Collect an LED.
- Gently separate the legs.
- Cut two 1.5cm squares of corrugated plastic.
- Use foam tape to sandwich the LED legs between the squares.
- Foam tape then cable tie the LED sandwich to the front of the vehicle.

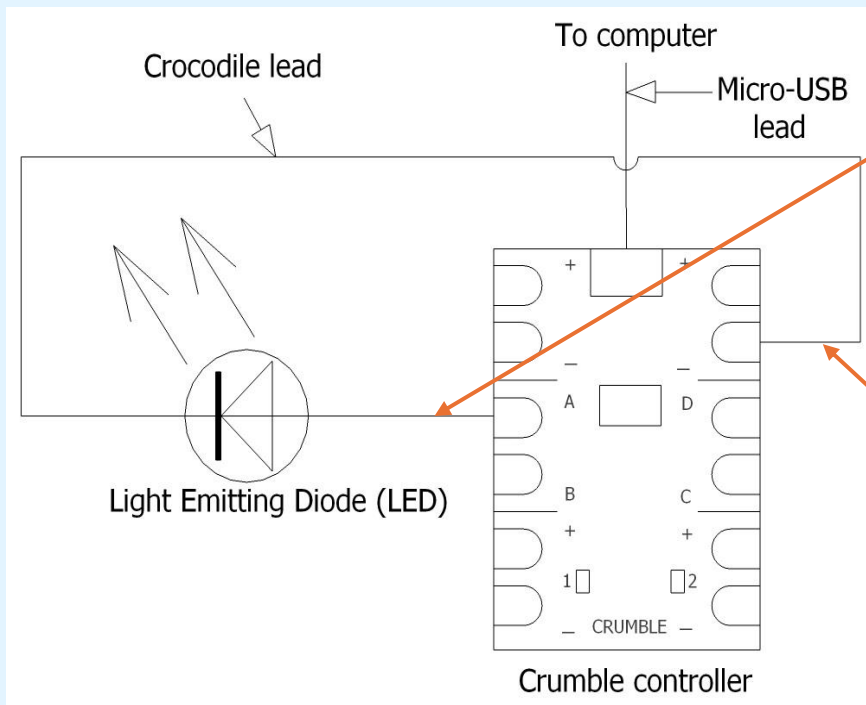


The long leg should be on the left.



How to connect the headlight

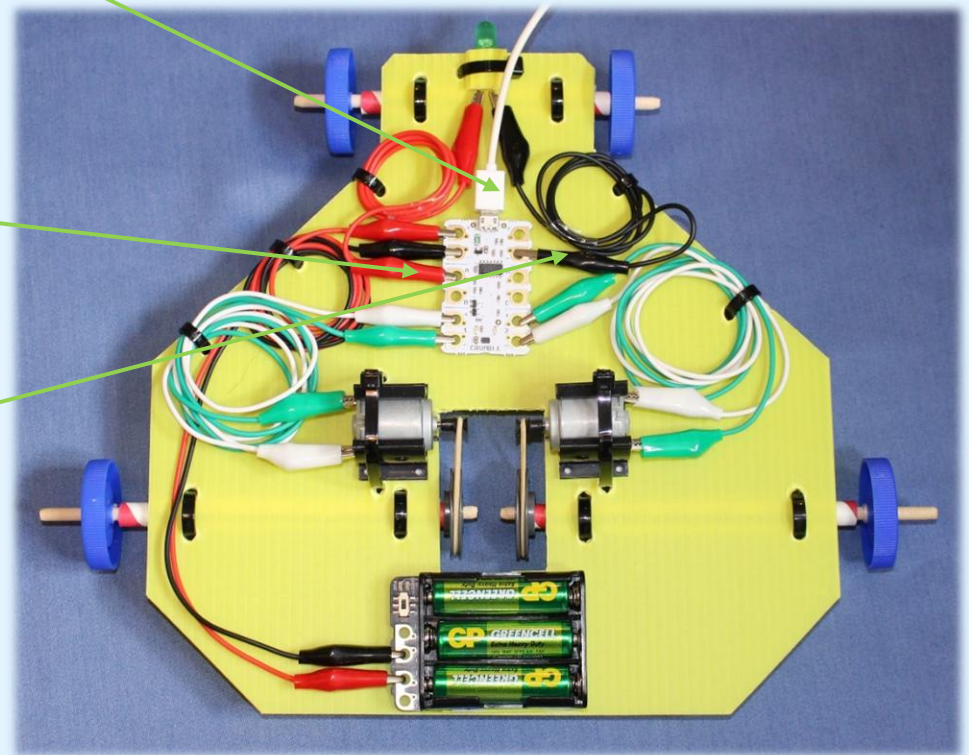
- This circuit diagram shows how the LED will be connected to the Crumble controller.



- The positive (long) leg of the LED will connect to output A.
- The negative (short) leg will connect to the negative 'Power out' terminal.

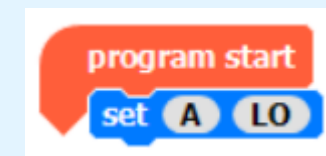
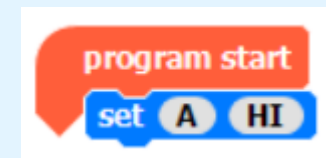
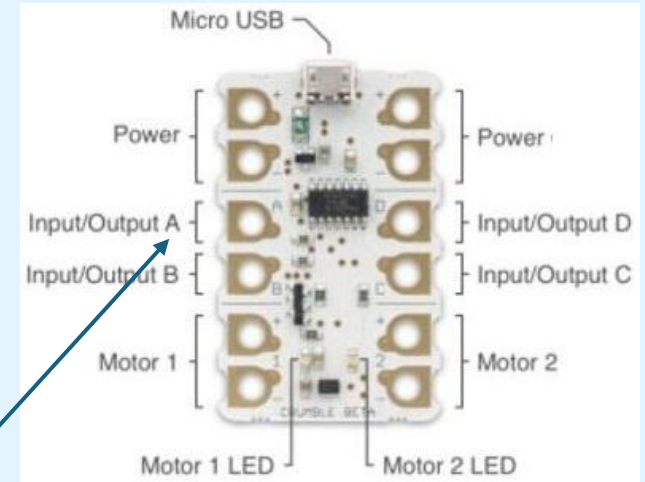
Connect up the headlight

- Plug in the micro-USB lead.
- Connect a crocodile lead from the long leg of the LED to terminal A on the Crumble controller.
- Connect a crocodile lead from the short leg to the negative (-) 'Power out' terminal.
- Coil the wires neatly and cable-tie them to the base.



Switch on the headlight

- Terminals A to D on the Crumble can be used as inputs or outputs.
- An output is information that a program produces and sends out. An input is information that a program receives.
- The headlight is connected to terminal A, which will be used as an output.
- Create this program to set output A to high. Run the program and check that the headlight comes on.
- Create and run this program. Check that the headlight goes off.



Algorithms

- An algorithm is a list of instructions to follow to complete a task or program.
- Here is an example of an algorithm to flash the headlight twice.

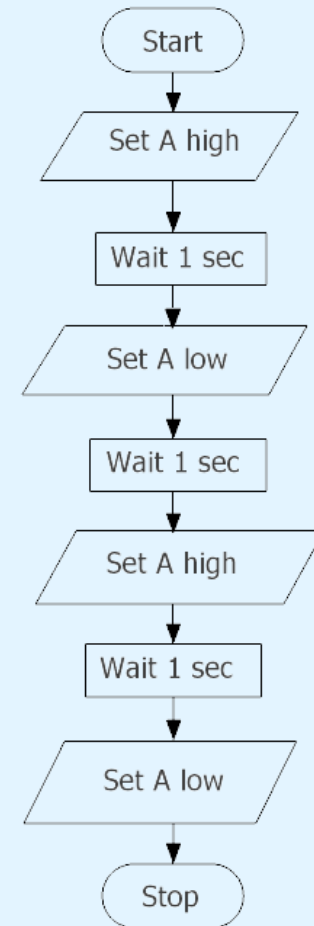
Algorithm

- Switch on the headlight.
- Wait 1 second.
- Switch off the headlight.
- Wait 1 second.
- Switch on the headlight.
- Wait 1 second.
- Switch off the headlight.



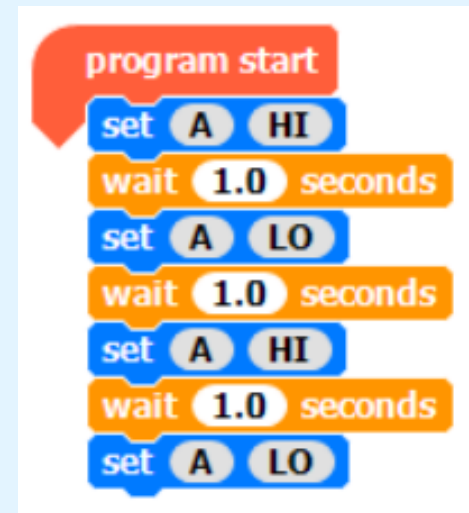
Flowcharts

- A flowchart is a simple diagram to show the steps in a process.
- A flowchart has the advantage of being a visual representation.
- This may be easier to follow than an algorithm, particularly if there are decisions or loops in the logic.
- Here is an example of a flowchart to flash the headlight twice.



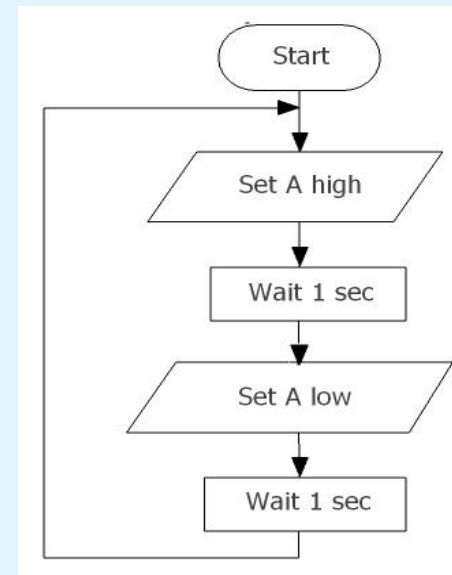
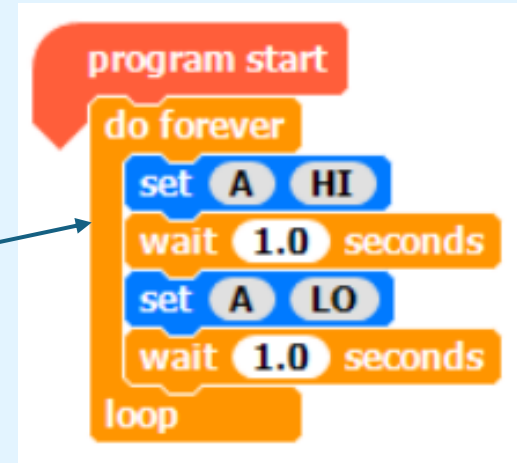
Flash the headlight on and off

- Sequencing is the ordered, step-by-step execution of instructions within a program.
- Here is an example of sequencing. What would you expect the headlight to do when you run this program?
- Create and run the program. Check that the headlight behaves as you expect.



Flash the headlight continuously

- Repetition is the repeated execution of a set of instructions until certain conditions are met.
- This 'do forever' loop is an example of repetition.
- What would you expect the headlight to do when you run this program?
- Create and run the program. Check that the headlight behaves as you expect.

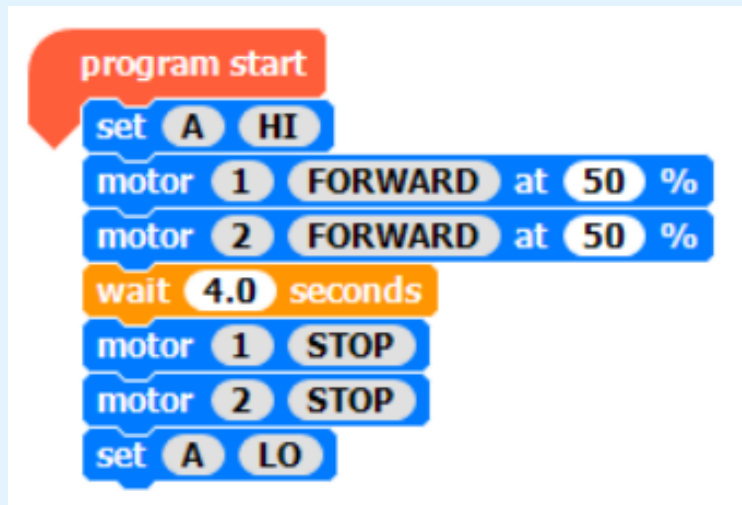


Run the vehicle with the headlight on

- Create this program to switch the headlight on, run the vehicle forward for four seconds, stop and switch the headlight off.
- Remember to run the motors at 50% power or less when starting. This is to leave enough power for the Crumble controller to boot up.
- After downloading the program, remember to unplug the micro-USB lead and switch on the battery box to run the vehicle.

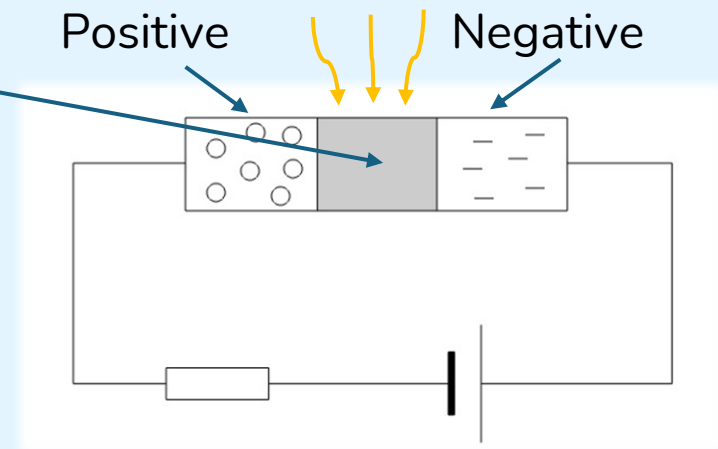
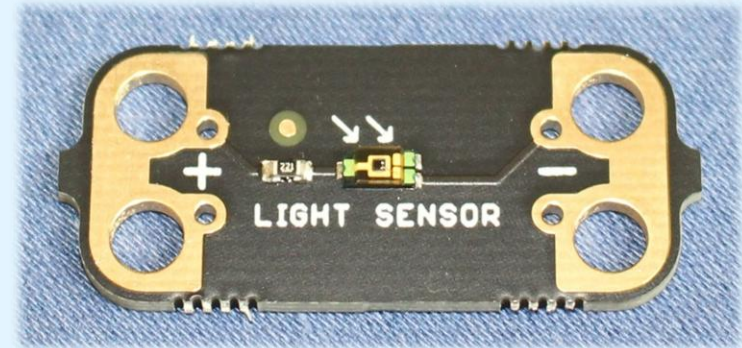
Algorithm

- Switch on the headlight.
- Run motor 1 forward at 50% power.
- Run motor 2 forward at 50% power.
- Wait 4 seconds.
- Switch off motor 1
- Switch off motor 2
- Switch off the headlight.



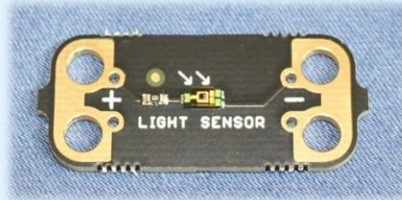
How a phototransistor works

- You can think of phototransistor as a light-controlled switch.
- Like the LED, it contains a semiconductor material with two sides, one of which contains positive particles, and the other negative particles.
- In between is the 'depletion zone'. When this zone is in darkness, the particles are not free to flow round.
- When light falls on it, this frees up particles, allowing current to flow round the circuit.
- So, when it is dark the phototransistor 'switch' is off. When it is light, the 'switch' is on.

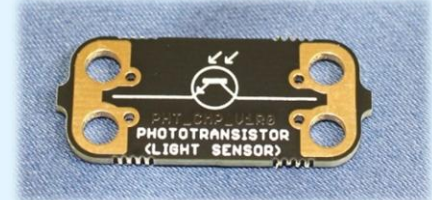


Fit a light sensor (phototransistor)

- Collect a light sensor. You can use it to check whether it is dark, then program the headlight to come on automatically.
- Attach a double layer of foam tape to the **bottom** of the light sensor.
- Stick the light sensor somewhere accessible on your vehicle.

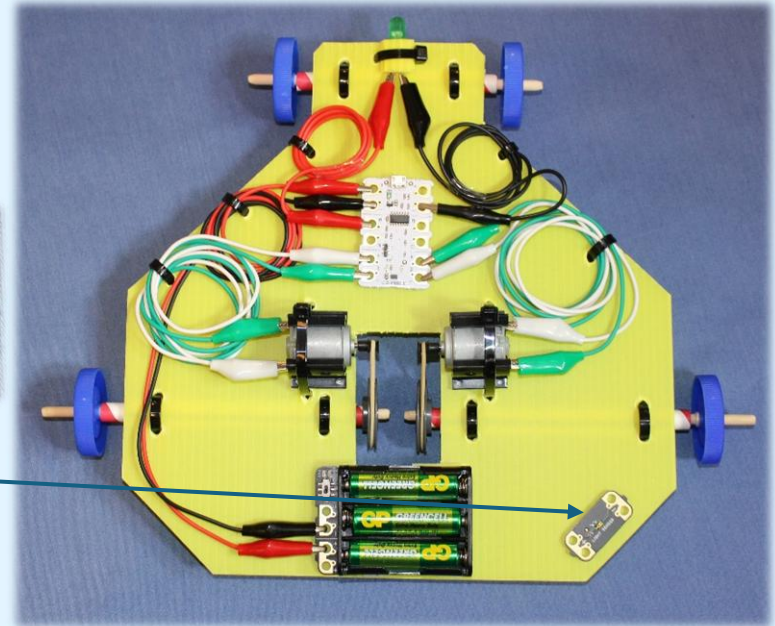
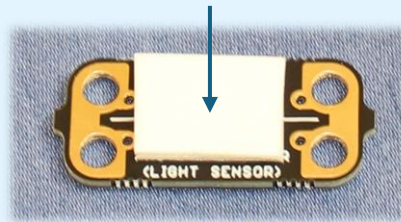


Light sensor (top)



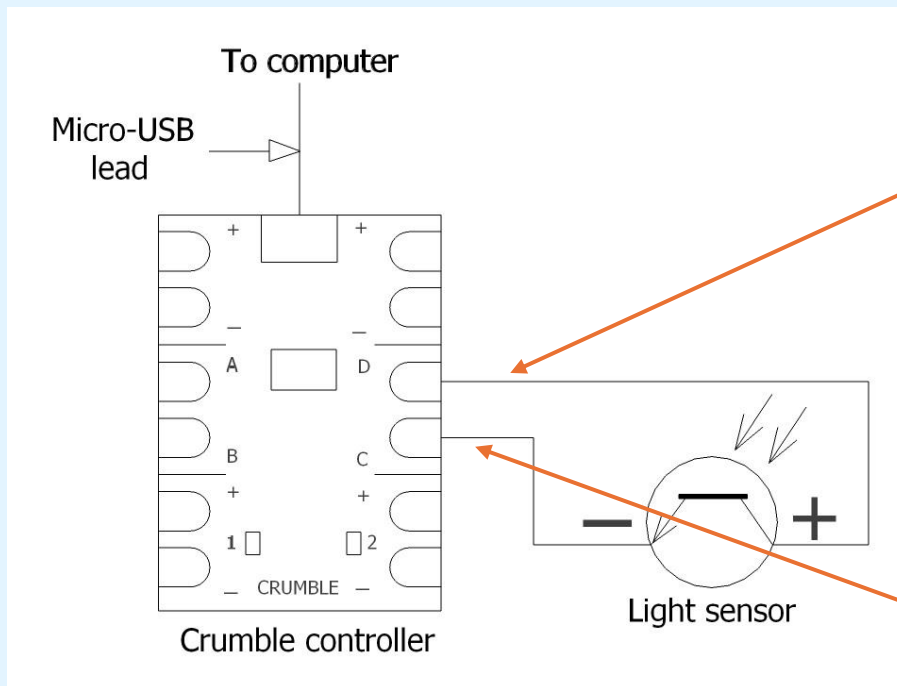
Light sensor (bottom)

Attach the foam tape this side



How to connect the light sensor

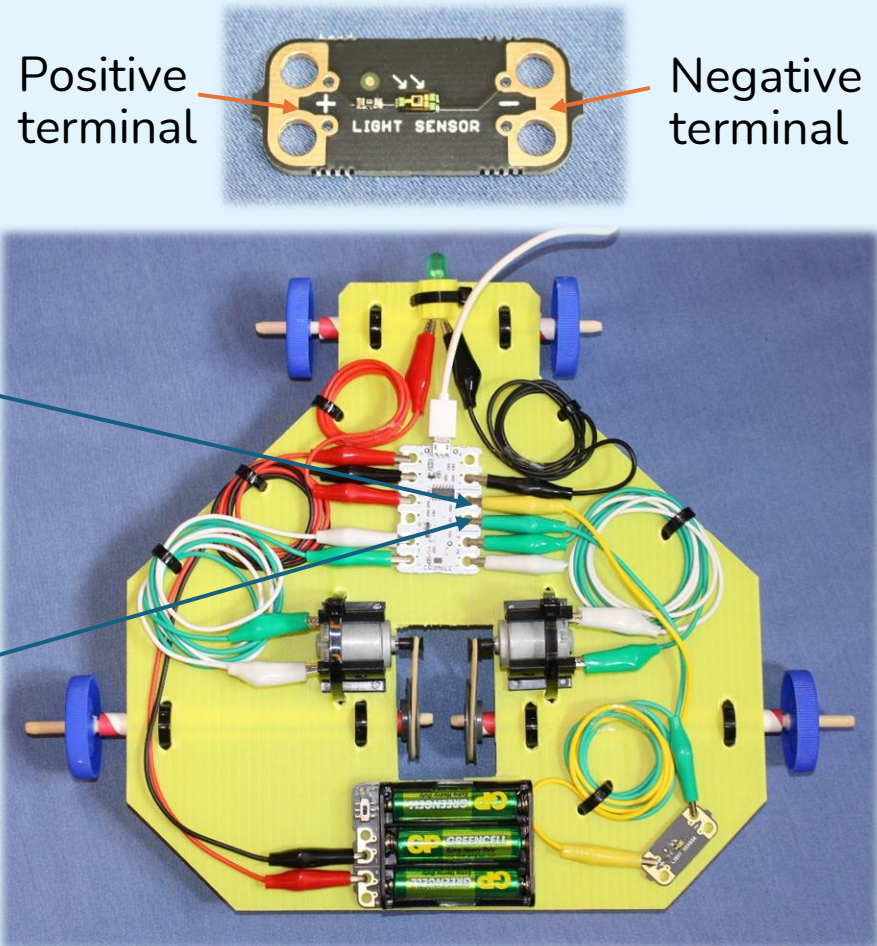
- This circuit diagram shows how the light sensor will be connected to the Crumble controller.



- The positive (+) terminal of the light sensor will connect to terminal D.
- The negative (-) terminal of the light sensor will connect to terminal C.

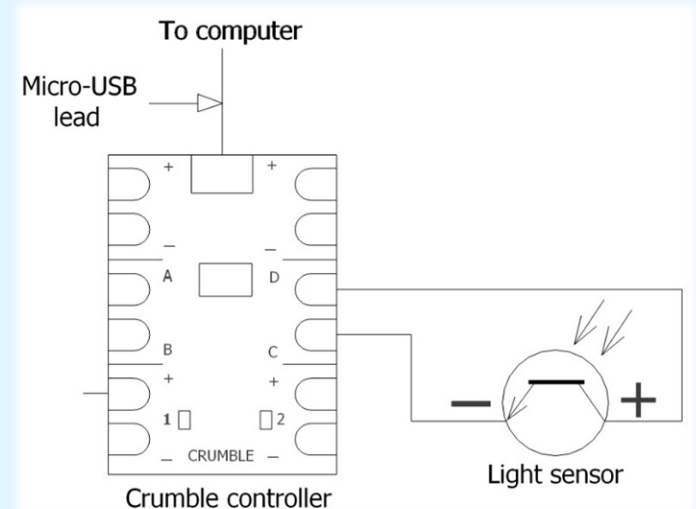
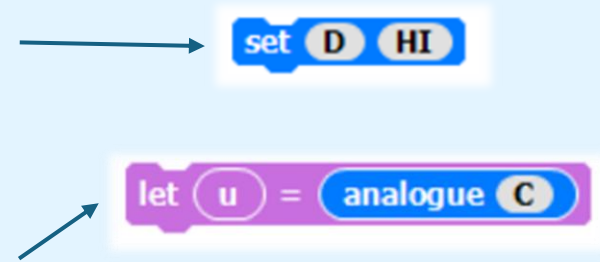
Connect up the light sensor

- Plug in the micro-USB lead.
- Use a crocodile lead to connect the positive (+) terminal on the light sensor to terminal D on the Crumble.
- Use a second crocodile lead to connect the negative (-) terminal on the light sensor to terminal C.
- Coil the wires neatly and cable-tie to the base.



How to use the light sensor

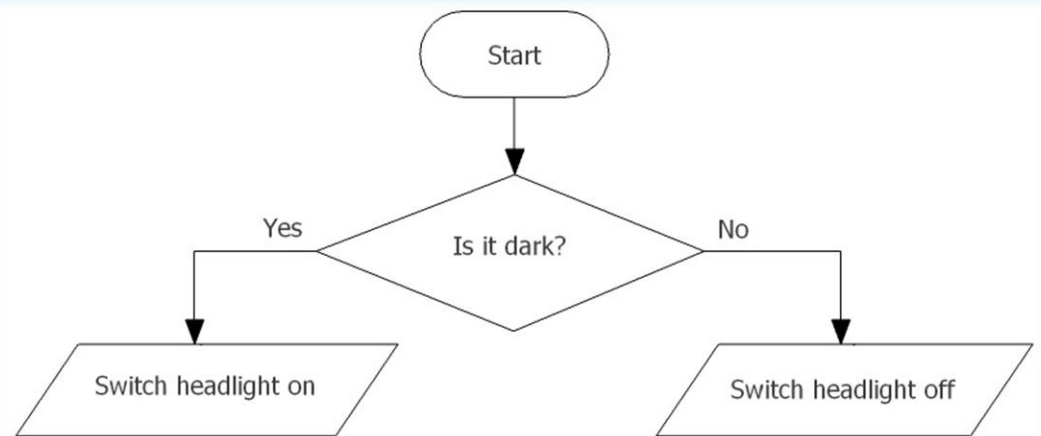
- You will program the Crumble to send a 'high' output to the light sensor via terminal D.
- You will then read the value input from the light sensor at terminal C.
- When light falls on the phototransistor, it will conduct electricity easily so you will receive a high value at C.
- When light does not fall on the sensor, it will not conduct easily so you will receive a low value at C.



Using variables and selection

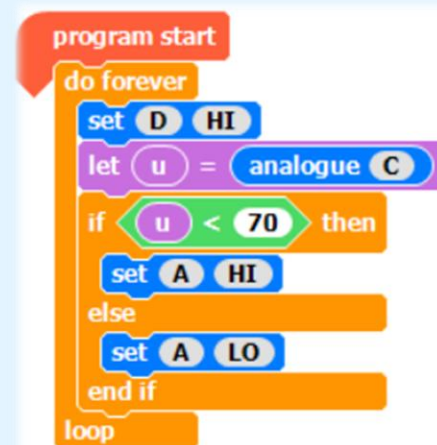
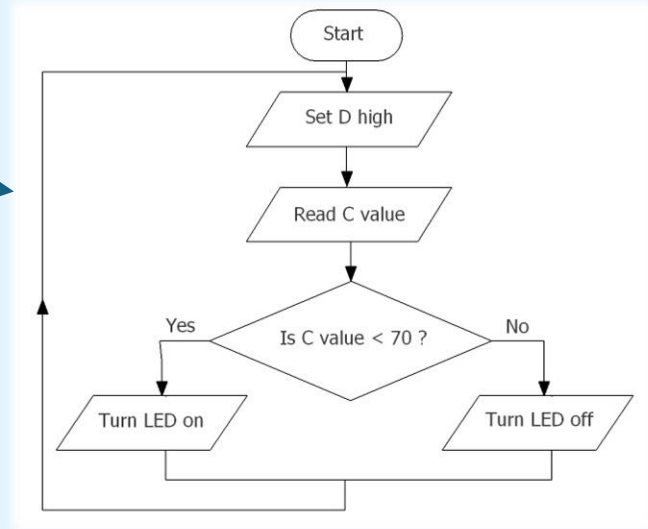
- A variable is a storage location used to hold data. In this case, the storage location is called 'u' and the data it holds is the value received from terminal C.
- Selection is making a decision about which instructions to follow, based on certain conditions. Here is an example of selection – if it is dark, switch the headlight on. If it is light, switch the headlight off.

```
let u = analogue C
```



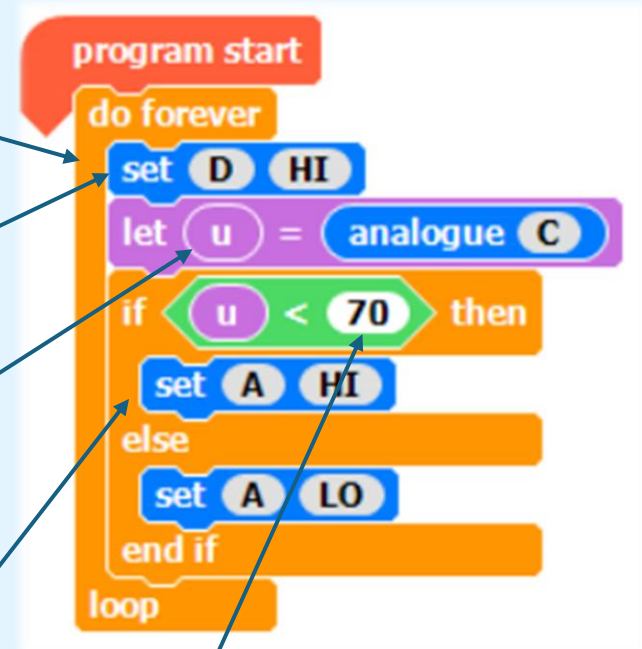
Make the headlight come on in the dark

- Here is a flowchart explaining how to check whether it is dark, and if so, to turn on the headlight.
- Here is a program to do this. The reason for the loop is so that the program keeps on checking whether it is still dark, instead of just checking once.
- Create and run the program.
- Try covering the light sensor with your hand – the headlight should come on.
- Remove your hand – the headlight should go off.



How the program works

- The first programming construct used here is repetition. It keeps going round in a loop to check whether it is still dark.
- The output D is set to high to activate the light sensor.
- A variable (u) is used to store the value of the input from the light sensor. This will be used to determine whether it is light or dark.
- Selection is used to check the value of the variable u, to determine whether to set the output A to high or low, switching the LED on or off.



How dark is dark? You can experiment with different values here, getting the headlight to come on at different light levels.



Run the vehicle with the automatic headlight

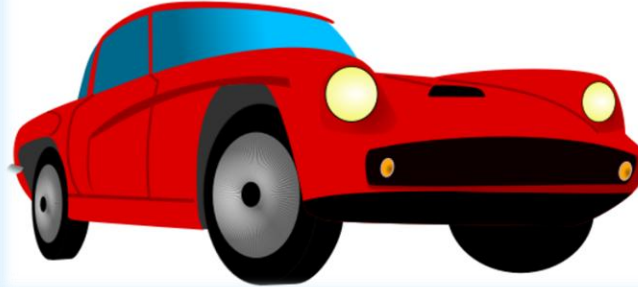
- Here is an example of a program to run a vehicle with an automatic headlight.
- It checks for light/darkness once at the beginning.
- You can create your own program to run the vehicle, incorporating a check for light/darkness.
- Test out your program. Check the headlight comes on if you cover the light sensor with your hand when starting the vehicle.



```
program start
set D HI
wait 1.0 seconds
let u = analogue C
if u < 70 then
set A HI
else
set A LO
end if
motor 1 FORWARD at 50 %
motor 2 FORWARD at 50 %
wait 3.0 seconds
motor 1 STOP
motor 2 STOP
wait 2.0 seconds
motor 1 FORWARD at 50 %
motor 2 REVERSE at 50 %
wait 2.0 seconds
motor 1 STOP
motor 2 STOP
set A LO
```

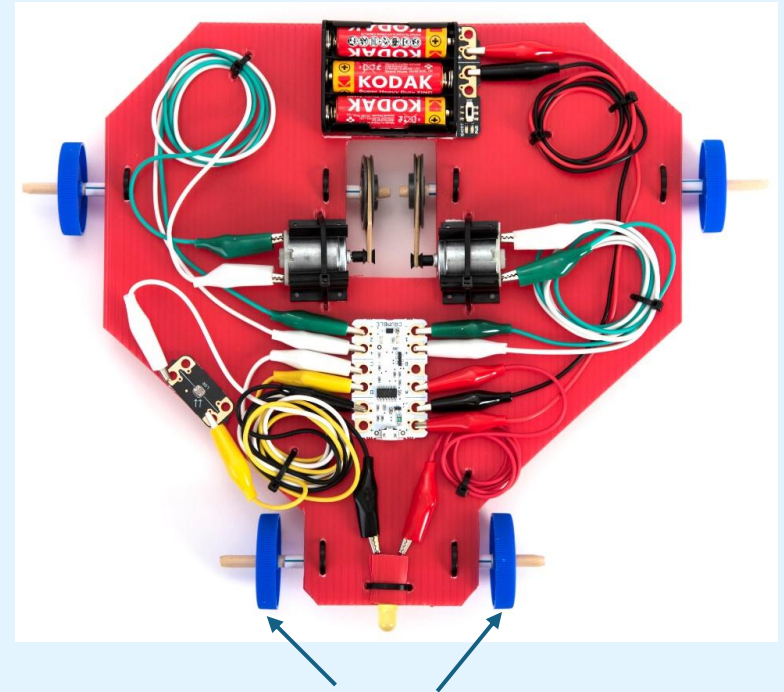
Other uses for light sensors

Suggest reasons why a light sensor could be useful in these real-world applications.



How the vehicle is steered

- The vehicle is steered by driving the two rear axles independently.
- In order to turn, you can:
 - Drive one axle faster than the other.
 - Leave one axle stationary and drive the other.
 - Drive one axle forwards and one backwards.



The front wheels are not steerable. In order for the vehicle to steer round a corner, the front wheels need to skid across the floor. How does this differ from a normal car?

Where do we want high or low friction?

Some materials provide more friction than others. In which of these real-world applications do we want high friction, and which low friction?



Training shoes



Slalom skiing



Bicycle tyres and brakes

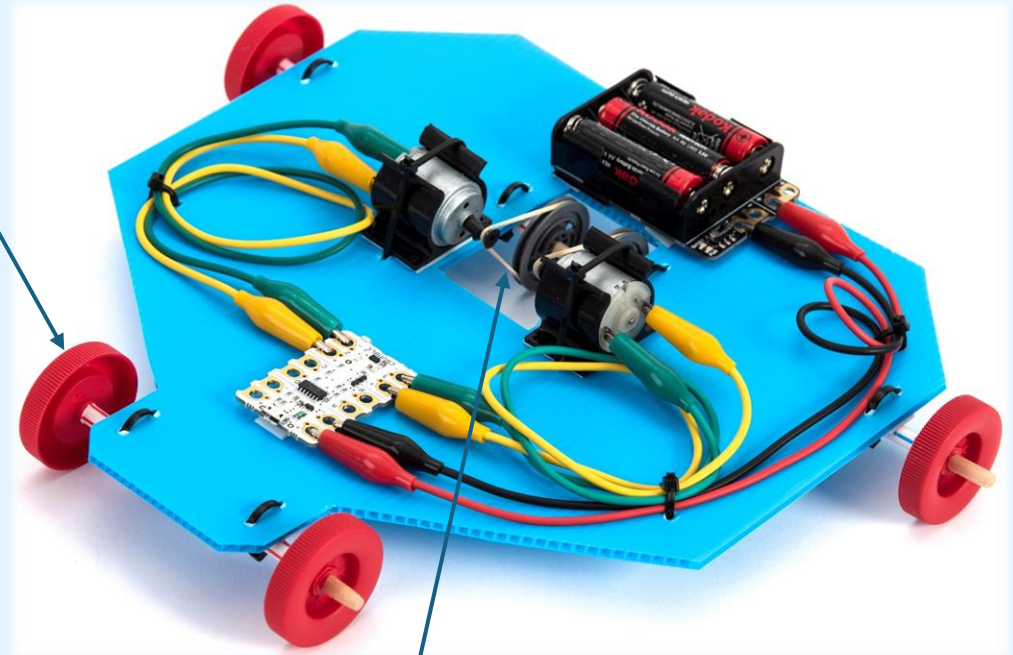


Water slide



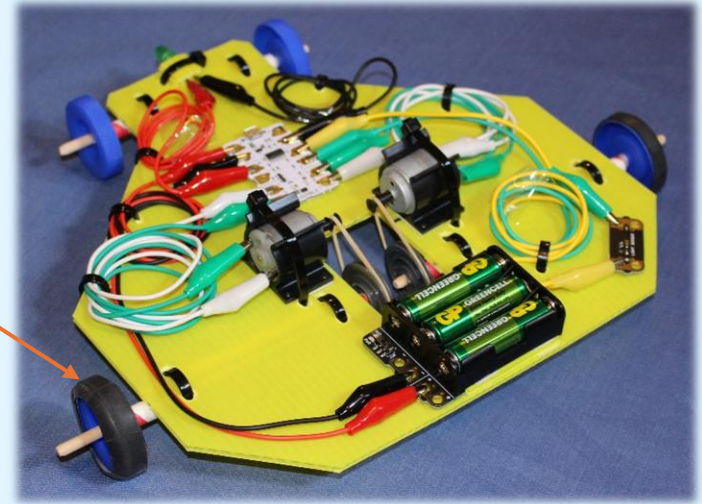
High and low friction materials

- The plastic wheels of the vehicle are relatively low friction. This is useful for the front wheels, as it enables them to skid when steering the vehicle round a corner.
- However, the back wheels need to grip the floor well to make the vehicle go forwards.
- Rubber is a high friction material, so we can fit rubber tyres on the back wheels.
- The drive belts which transmit the power from the motors to the axles also need high friction to grip the pulleys well. They are made from rubber bands.



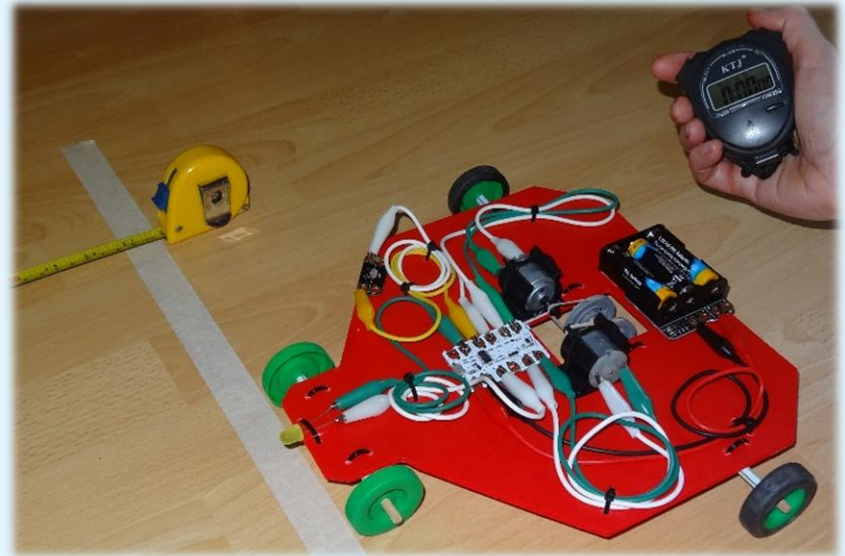
Improve traction on the driving wheels

- Cut 1.5 cm slices of inner tube or balloon (they are both made from rubber) to make tyres for your rear wheels and help them grip the floor better.
- Try the vehicle on different surfaces, e.g. the school hall, carpet or the playground.
- Can the vehicle turn easily on these?
- Is there a way to reduce the friction on the front wheels to help the vehicle turn?



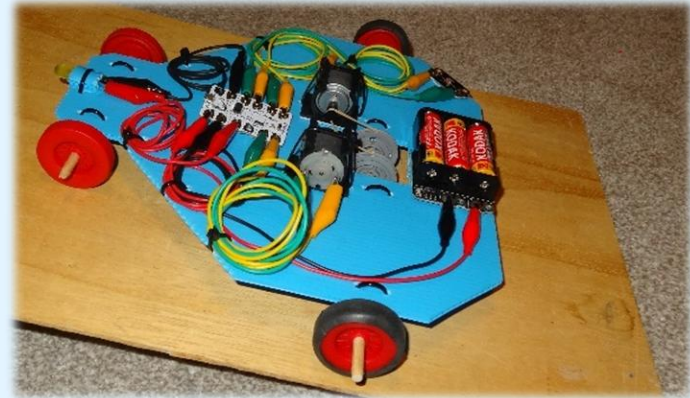
Measure your vehicle's performance

- You can have a drag race. Set out a start and finish line at a known distance apart.
- Time the vehicle over the distance. You could run the vehicles at 50% power throughout. However, you could try a higher power after the initial boot-up, particularly if you don't power the headlight or the light sensor.
- Remember to program the vehicle to stop.
- Calculate and record the average speed.
- Which vehicle is the fastest? You can create a leader board. You could also race the vehicles head-to-head and have an elimination.



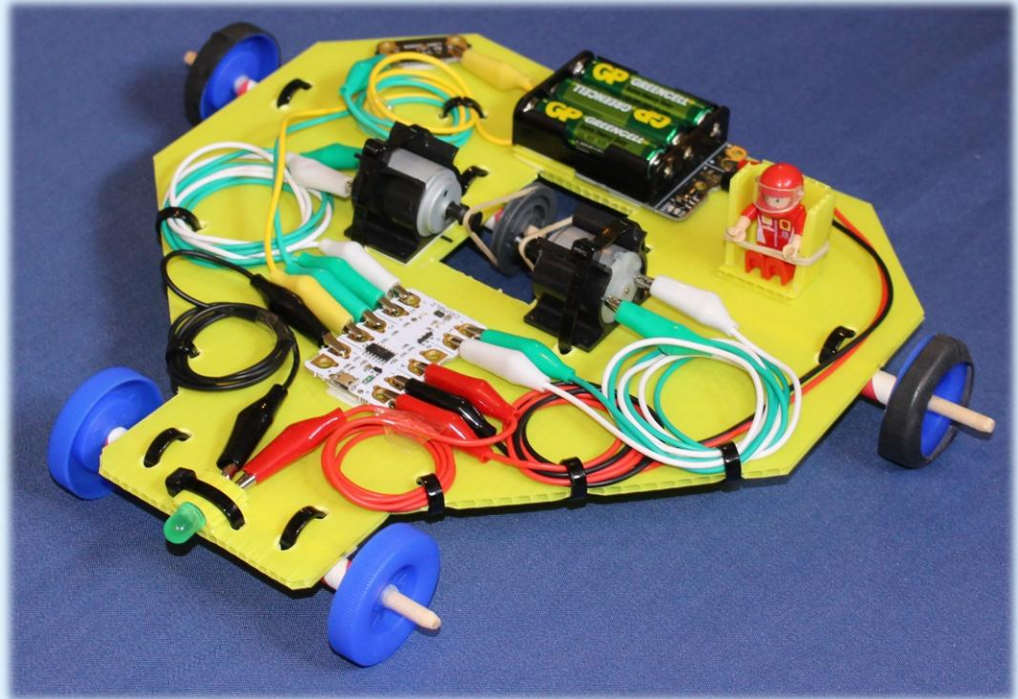
Try out different courses

- Try out the vehicle up a ramp and record the steepest angle the vehicle can travel up.
- Program your vehicle to follow the course laid out – who can complete it in the shortest time?
- You can try constructing other courses, for example a slalom or a figure of eight.
- You can try racing the vehicles on different surfaces and over different course lengths, for example in the playground (you could mark the start and finish lines with chalk).
- You can try tying two vehicles back-to-back and having a tug of war, or having two vehicles face to face in a sumo competition.



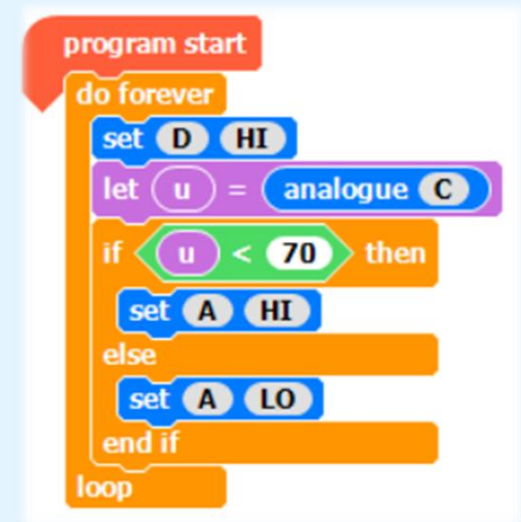
Improve your vehicle

- You can modify your vehicle to include a passenger, such as a mini-figure.
- Where will it fit on your vehicle?
- How will you stop the passenger from falling out?
- Do you want to decorate your vehicle, or make any other improvements?



Plenary

- Discuss what you have learnt.
- What did you enjoy the most?
- Which is more efficient, an LED or a traditional bulb? Why?
- What is a light sensor used for?
- Where did you want high or low friction on your vehicle?
- Why do bicycles have rubber tyres?
- Which part of this demonstrates repetition?
- Which part uses selection?
- Which is the variable?



```
program start
do forever
  set D HI
  let u = analogue C
  if u < 70 then
    set A HI
  else
    set A LO
  end if
loop
```

The image shows a Scratch code block with a red 'program start' tab. It contains a 'do forever' loop block. Inside the loop, there are four blocks: a 'set D HI' block, a 'let u = analogue C' block, an 'if u < 70 then' block, and an 'end if' block. The 'if' block has two branches: 'then' with a 'set A HI' block, and 'else' with a 'set A LO' block. A blue arrow points from the question 'Which part uses selection?' to the 'if' block.



More fun TTS STEM class kits

Balloon buggies



Renewable energy



Periscopes



Make your own light



Build a house



Crumble kit

