

**Challenge:** To use the Micro:Bit to control the LED lights.

**Objective:** To write an algorithm that uses the buttons (Inputs) to control the LEDs (Outputs) - on and off.

Bronze	Silver	Gold
<p>With some support can connect and code the LEDs to turn on and off.</p> <p>Can list some of the Input / Outputs associated with the Micro:Bit.</p>	<p>Can connect the LEDs to the Micro:Bit using crocodile clips.</p> <p>Understands it is important to know which Pin they are connected too.</p> <p>Can install the ToodleBit extension and write their own algorithm to turn the LEDs on and off.</p>	<p>Can explain why the LEDs are outputs.</p> <p>Able to complete the extension activities without adult support.</p>

### Notes

An **algorithm** is a sequence of instructions or a set of rules that are followed to complete a task. This task can be anything, so long as you can give clear instructions for it.

An **input** is data that a computer receives. An **output** is data that a computer sends. Computers only work with digital information. Any input that a computer receives must be digitised. Often data has to be converted back to an analogue format when it's **output**, for example the sound from a computer's speakers.

### Equipment

- Micro:Bit, USB Cable, Computer with internet access. A set of LEDs (1 green, 1 amber, 1 red) 4 crocodile clips and access to masking tape and foil.

### Introduction



Ask pupils what an algorithm is? They could discuss in pairs and feed back to the class. Tell them they are going to write an algorithm to make a paper plane. Watch the video 'Paper plane'. Pupils write their own list of steps involved. They pass these steps to somebody else on their table who then get to follow them exactly and make a paper plane.



Follow up: Did everybody follow the algorithm exactly? Were all instructions clear and in the correct sequence? Could you pass these algorithms to a different class and they would be able to follow them and make their own paper planes. Are there any instructions you could add to improve your algorithms? What are the benefits of algorithms?

### Activity



Watch the video 'Installing extensions'. Extensions are extra code blocks. ToodleBit has created some of its own to make coding the traffic lights unit of work easier.



Watch 'Part One' and then complete activities. Show the pupils how to connect one LED and add code to make it light up and turn off.



Watch 'Part Two' and complete activities. Shows the pupils how to add the remaining LEDs and code them to turn on and off.

### Conclusion



Get pupils to show their working lights. Can they identify the Inputs and Outputs? Can they explain what makes the buttons Inputs and the LEDs outputs?

## Extension 1 code blocks

```

on button A pressed
  red light P0 on
  pause (ms) 1000
  red light P0 off

on button B pressed
  amber light P1 on
  pause (ms) 2000
  amber light P1 off

on button A+B pressed
  green light P2 on
  pause (ms) 3000
  green light P2 off
  
```

## Extension 2 code blocks

```

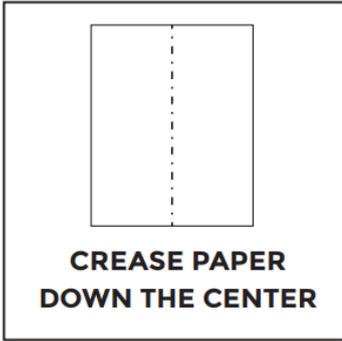
on button A pressed
  green light P2 on
  pause (ms) 1000
  green light P2 off
  amber light P1 on
  pause (ms) 1000
  amber light P1 off
  red light P0 on
  pause (ms) 1000
  red light P0 off

on button B pressed
  red light P0 on
  pause (ms) 1000
  red light P0 off
  amber light P1 on
  pause (ms) 1000
  amber light P1 off
  green light P2 on
  pause (ms) 1000
  green light P2 off
  
```

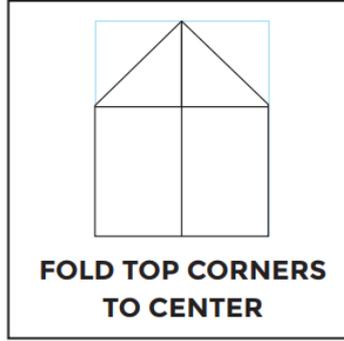
### Extension ideas

- Program your Micro:Bit to;
  1. light the red LED for 1 second when 'Button A' is pressed
  2. light the amber LED for 2 second when 'button B' is pressed
  3. light the green light for 3 second when both 'buttons A+B' are pressed
 All should go off automatically when the time is up.
- Program your Micro:Bit to;
  1. Show green, amber, red for a second each when 'button A' is pressed and in reverse order when 'button B' is pressed.

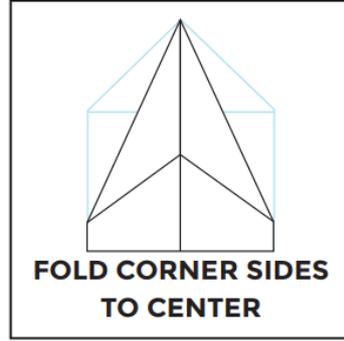




1



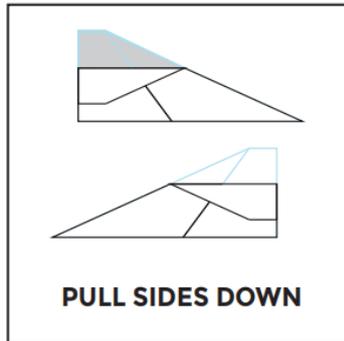
2



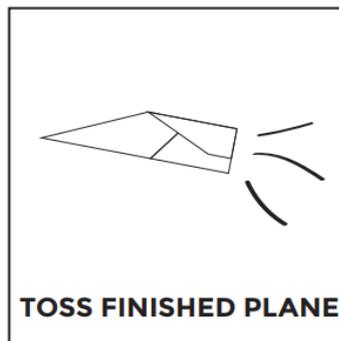
3



4



5



6