GPIO stands for General Purpose Input/Output and some pins can be either an input or an output, but do you know which is which?

Discuss with a partner or someone sat near your, which of these are inputs or outputs and put them into the table below:

A button, an LED, a buzzer, a motion sensor, speakers, pressure pads, the letter ‘k’ on your keyboard, the cat moving in Scratch, a speech bubble in your animation. Can you think of any more inputs or outputs?

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A PiStop is an output</td>
<td>it has three LEDs which</td>
</tr>
<tr>
<td></td>
<td>you can control using</td>
</tr>
<tr>
<td></td>
<td>Scratch GPIO on a</td>
</tr>
<tr>
<td></td>
<td>Raspberry Pi.</td>
</tr>
</tbody>
</table>

***IMPORTANT Get an adult to check you have the PiStop in the right place***

Once you’ve checked your PiStop is in place you can load up Scratch by clicking on menu -> programming -> Scratch
Now you can tell it that you want to use the GPIO pins – to do this, click on edit and select “Start GPIO server”

Next, you will need to tell the pins you are using whether they need to be inputs or outputs. We are trying to make lights turn on so this means we are creating outputs.

The first pin we will use is pin 17, so let’s start by making pin 17 into an output by placing the following blocks and clicking on them (NB. config is short for configure):

Can you make a single light turn on?

Try this:

What happens?

Can you make the light blink ten times in a row? Does it blink smoothly or flicker? Check your code carefully.

Can you make the other two lights come on? The pins are numbered with a special system called BCM, this diagram should help you figure out which pin numbers you are using – don’t forget to tell them to be outputs!
*** *** Challenge *** ***

Can you make the lights turn on as though they were traffic lights? The order is red, red and amber together, green, amber, red and then repeat.

The red and green lights need to be on for longer as these are the stop & start lights.

Work with a partner to make your traffic lights work in opposition so that while one PiStop is showing red, the other is showing green.

*** *** Challenge 2 *** ***

Can you make a disco dancing set of lights using some of the sound files available in Scratch?

*Hint* try using “play drum 35 for 0.5 beats” with “play drum 39 for 0.5 beats” as a start point
Part Two

In real life, traffic lights wait on red until they sense an approaching car and then run their sequence from red, through to green and back to red again. Can you make your lights wait on red until they sense an approaching vehicle? You will need to fit a PIR (infra-red motion) sensor to your Raspberry Pi.

First, find LXTerminal and type in

```
sudo halt
```

This will shut down your Pi so that you can plug in your sensor as shown in the diagram on the right (remember to check with an adult before you turn your Pi back on).

A PIR sensor is an ‘input’ and so can either be on or off – we use simple BINARY to represent this:

\[
\begin{align*}
1 &= \text{on} \\
0 &= \text{off}
\end{align*}
\]

If the sensor detects movement, it will become on, otherwise it is off.

To get the PIR sensor working, we need to click on the blue sensing blocks in Scratch GPIO.

You’re also going to need to use a green ‘equals’ operator and an ‘if/else’ loop to create a statement which makes your lights change from red to green when a car approaches.

*** *** Challenge *** ***

Can you create a Scratch animation that starts when your traffic lights are activated by the PIR sensor? Add some sound effects to make it really interesting.

*** *** Challenge 2 *** ***

Can you make a game where you need to click the sprite when the red light comes on?