The MagPi ESSENTIALS

[CHAPTER EIGHT]

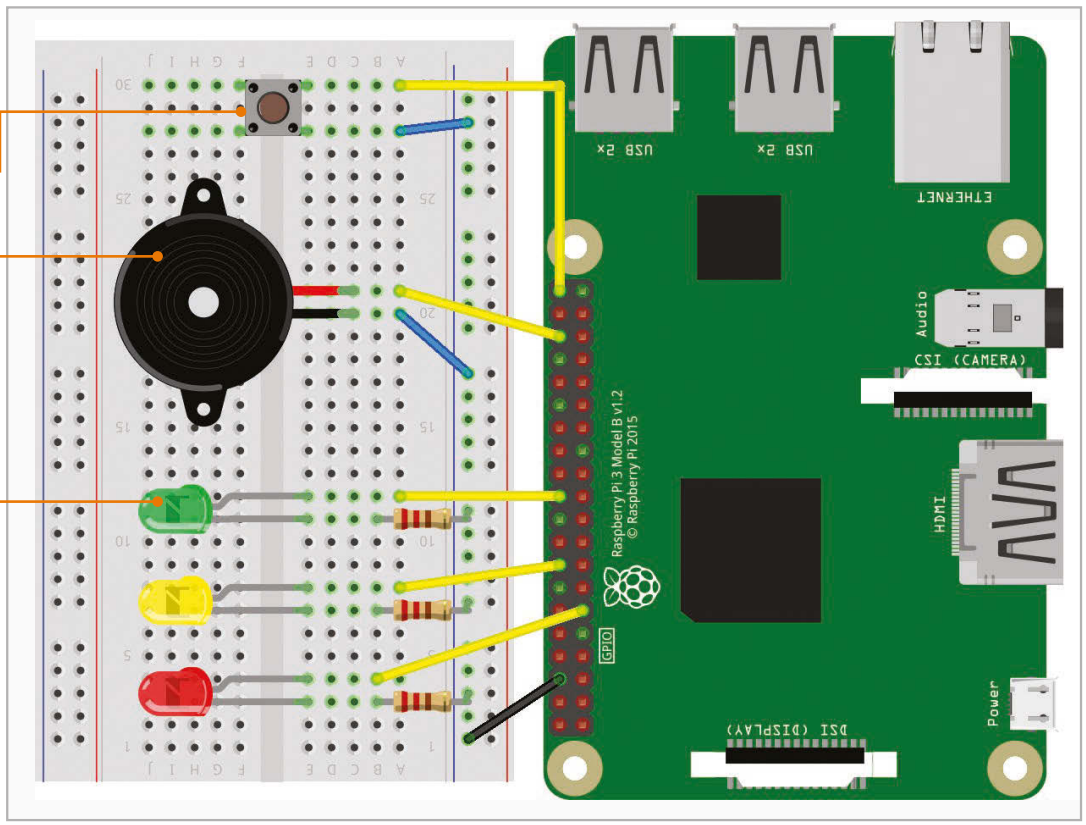
LED TRAFFIC LIGHTS

Following on from the previous chapter, we'll use three LEDs and a push button to make a pedestrian crossing

When the button is pressed, the circuit is broken and Scratch senses a zero value from GPIO pin 21

A piezo buzzer is wired up to the ground rail and GPIO pin 16, for our pedestrian crossing beeps

Each LED is connected to a different GPIO pin, so it can be triggered during the traffic light sequence



You'll Need

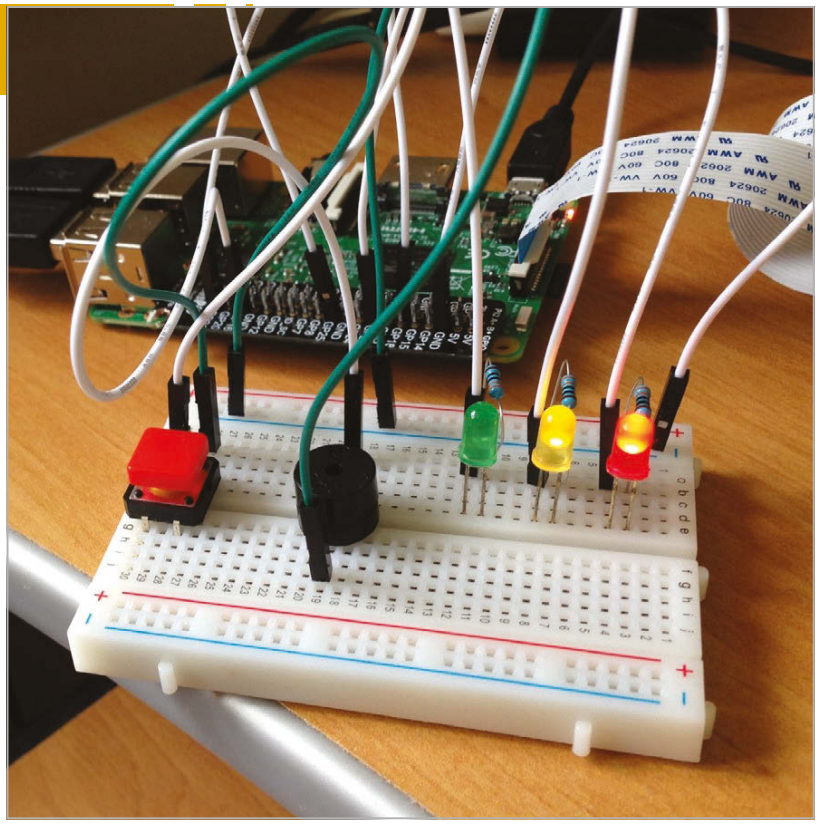
- Solderless breadboard
- 3× LEDs: red, yellow, and green
- 3× 333Ω resistors
- Push button
- Piezo buzzer
- 5× male-to-female jumper wires
- 2× male-to-male jumper wires

In the latest version of Raspbian Jessie, Scratch features a built-in GPIO server to make it easier to control electronic components or add-on boards. In this second GPIO tutorial, we'll create some traffic lights with a pedestrian crossing using LEDs, a push button, and a buzzer. Again, all the components required are in the CamJam EduKit #1 (**magpi.cc/1OcXtim**).

>STEP-01

Connect the LEDs

It's best to turn the Pi off when building your circuit. The breadboard features numbered columns, each comprising five connected holes. Add the LEDs to it, as shown in the diagram. If you've just finished chapter 7, you can leave those components, including the red LED, in place. As before, the shorter (negative) leg of each LED should be connected via a resistor to the '–' row (common ground rail), which is wired to a GND pin on the Pi. Each LED's longer (positive) leg should be connected to the respective GPIO pin via a male-to-female jumper cable.



Above: While there's quite a jumble of wires, it's relatively easy to connect all the components

>STEP-02

Configure Scratch GPIO

First, we need to turn on Scratch's GPIO server. Under a **when green flag clicked** block, add a **broadcast** Control block, click its arrow, select new/edit, and enter **gpioserveron**. We also need to configure our LEDs' GPIO pins as outputs, so add three more **broadcast** blocks and change them to **config17out**, **config23out**, and **config25out** respectively. While we're at it, we'll configure the pins for the buzzer (**config16out**) and button (**config21in**) we'll use later – your code should look like **Listing 1**.

>STEP-03

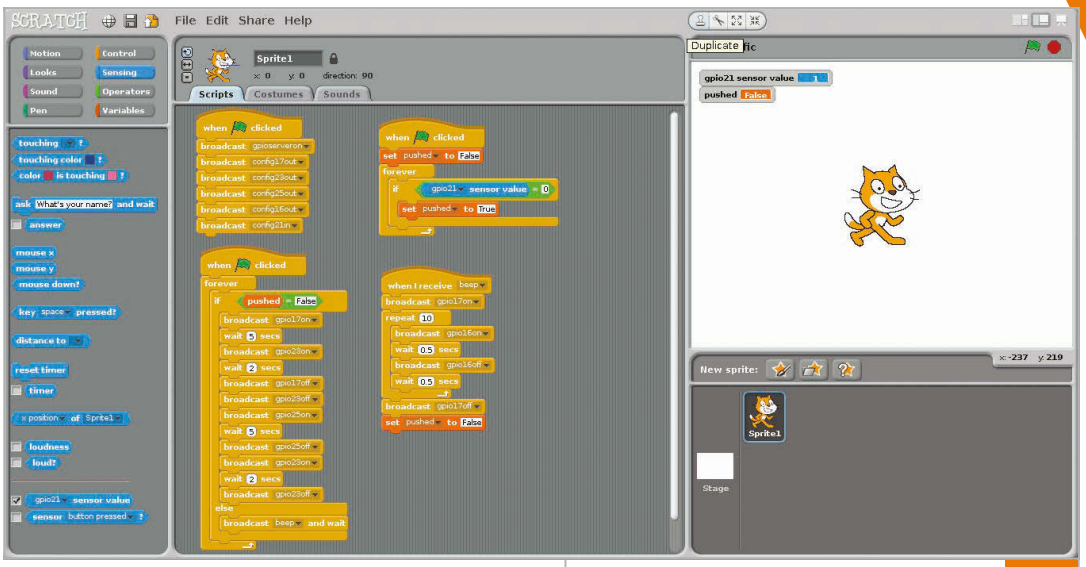
Traffic light sequence

We'll now test our circuit by creating a traffic light sequence: red, red/amber, green, amber. Add the code from **Listing 2**. Here, within a **forever** block, are blocks to turn the LEDs on and off in the correct sequence, waiting a few seconds between each change. Try running it to check that all the LEDs are connected correctly and working.

>STEP-04

Connect the button

For our pedestrian crossing, we'll need a push button. Again, you can use the one already placed in chapter 7, which straddles the central groove of the breadboard and is connected to the ground rail and GPIO pin 21. We've already configured it as an output in step 2; run and stop that code.



Left: Four pieces of code are used for GPIO configuration, light sequence, button press detection, and buzzer beeping

Now, click Sensing in the top-left pane. Find the **sensor value** block and change it to **gpio21**. Click its tickbox to show its value on the stage: when the button is pressed, it'll change from 1 to 0.

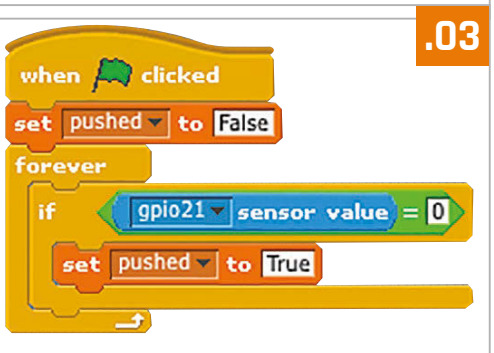
>STEP-05

Stop the lights

We need to get a button press to cause the traffic lights to stay on red for a few seconds. Select Variables from the top-left, then click 'Make a variable' and enter 'pushed' in the text field. Add the code from **Listing 3**, keeping it separate from the rest. Using an **if** block, this sets **pushed** to **True** when the value sensed from GPIO pin 21 is zero, i.e. when the button is pressed. Next, we need to add an **if…else** block to our traffic light sequence code, to stop it when **pushed** is **True**. After moving the light sequence

.02

.03



.04

.05

blocks out of the **forever** block (keeping them in the Scripts area), add in an **if…else** block and put the light sequence blocks back under **if**. In the **if** field, use an **=** Operator block with **pushed** in the left field and 'False' in the right. Under **else**, add a **broadcast and wait** block set to 'beep' – we'll be using this for our buzzer in the next step. Your light sequence code should now resemble **Listing 4**.

>STEP-06

Add a buzzer

Finally, we'll add a piezo buzzer, connected to the ground rail (short leg) and GPIO pin 16 (long leg), to make a beeping noise when it's safe to cross the road. Add the code from **Listing 5** as a separate script. This runs whenever **beep** is broadcast, after the button is pressed and the light sequence ends. It shows a red light and uses a **repeat** loop to turn the buzzer on and off for a beeping sound. Finally, it turns off the red LED and resets the **pushed** variable to **False**. Test out your pedestrian crossing by pressing the button!