Make your own traffic lights

Selby CoderDojo

What you will make!

This should be enough to get you started!

I'm here to help you along the way

You will use Raspberry Pi Pico to create a traffic light

You will need:

- Raspberry Pi Pico
- Breadboard
- 3 x Two prong LEDs
- 1 x Button (Optional)
- Male to Female and Male to Male Jumper Cables
- 3 Resistors
- 1 x Buzzer (Optional)

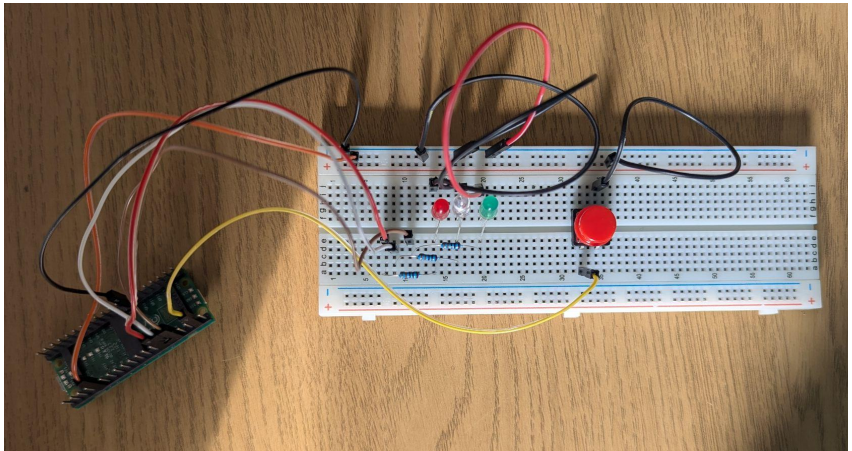
What your project will look like

This is what the finished project will look like.

However yours might look a bit different and that's okay!

The image below shows the "end" product (minus the buzzer)

Yours may not look exactly like mine and that is completely fine!

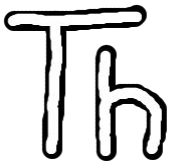


Getting started

If you are using a borrowed laptop then it should already be set up.

To set up your environment you need to have Thonny installed on your computer and have it set up to work with Raspberry Pi Pico and MicroPython

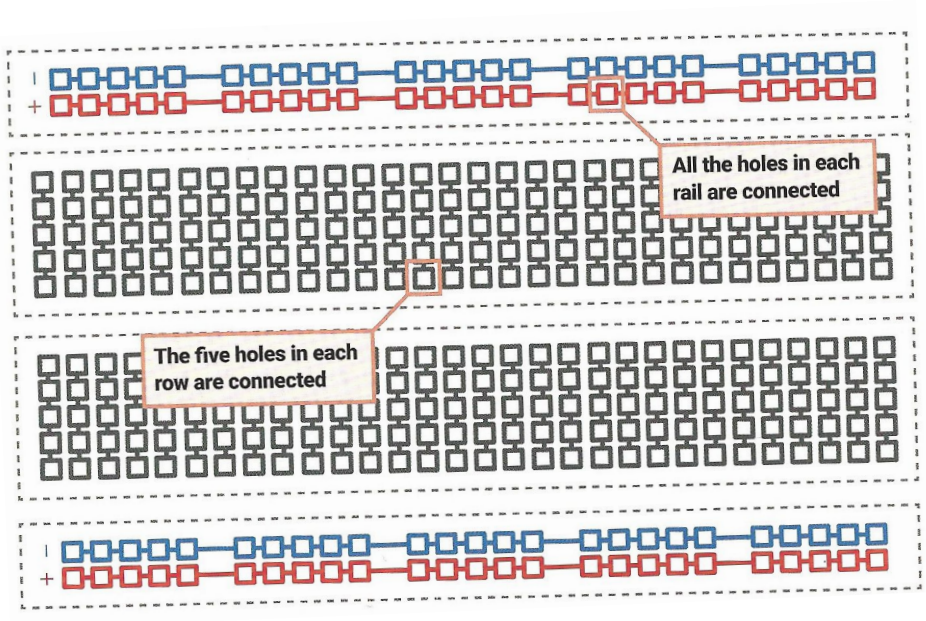
(If you get stuck ask a volunteer or check the Pico Pathway!)



The breadboard

It took me a while however a breadboard is easy. The diagram shows it perfectly

The breadboard can be a tough piece of kit to master however it is a lot easier than you think



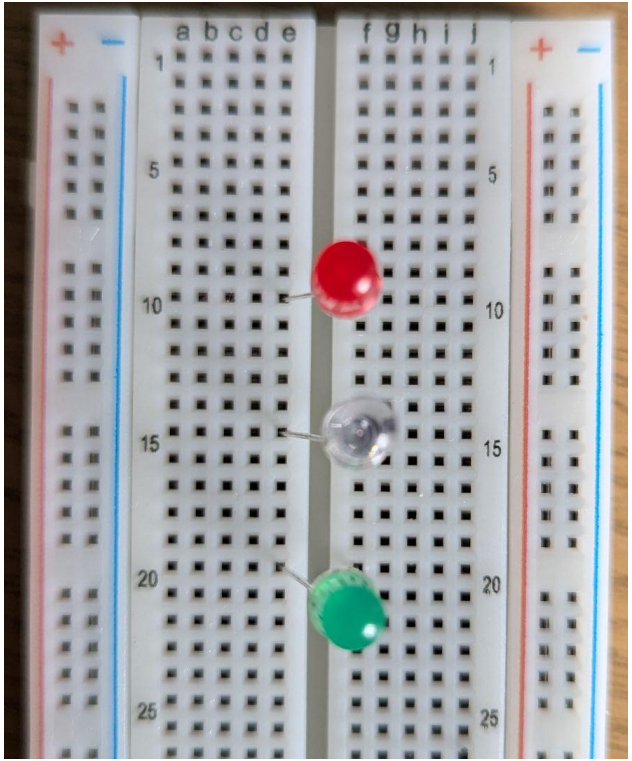
(Halfacree and Everard, 2021, figs 4–2)

We can use the breadboard to set up our hardware

Use traffic light colours! My middle LED is orange even though it looks clear

To get started let's add some LEDs and resistors.

I have placed my red LEDs in rows 10, 15, and 20 respectively. The long leg on the left hand side of the breadboard and short leg on the right



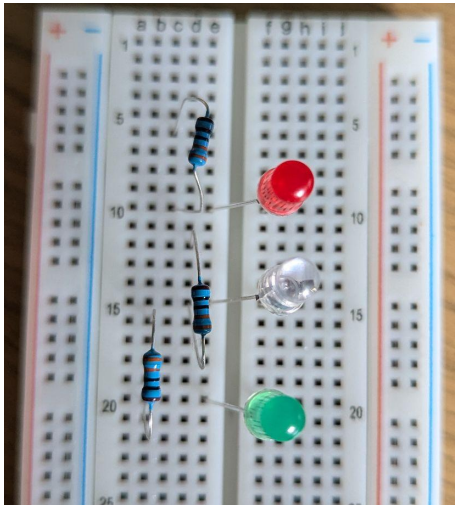
We can use the breadboard to set up our hardware

Resistors stop LEDs from burning out.

A standard 330Ω resistor will work

The resistors are a little trickier and need one leg on the same row as the LED and one leg on its own row

It doesn't matter which way your resistor faces



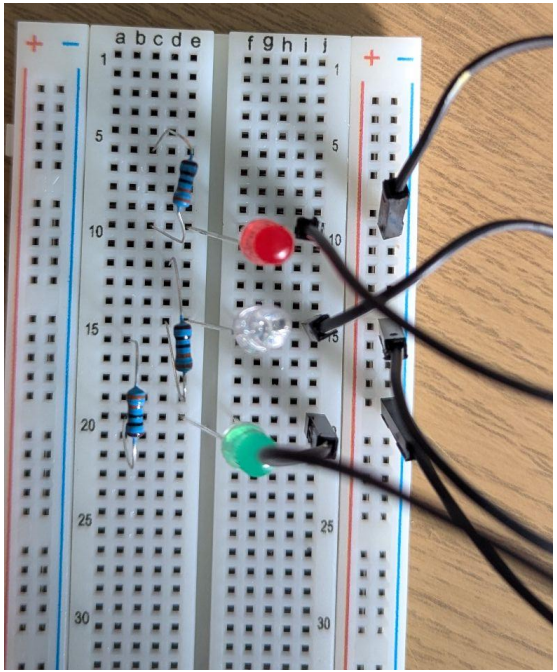
Making a circuit

The ground pin is used to close the circuit

All the LEDs can go to the same Pin so we use the rail for ease

Use a male to male jumper cable to link each LED to the negative rail on the right.

This allows us to link all the LEDs to the same ground pin



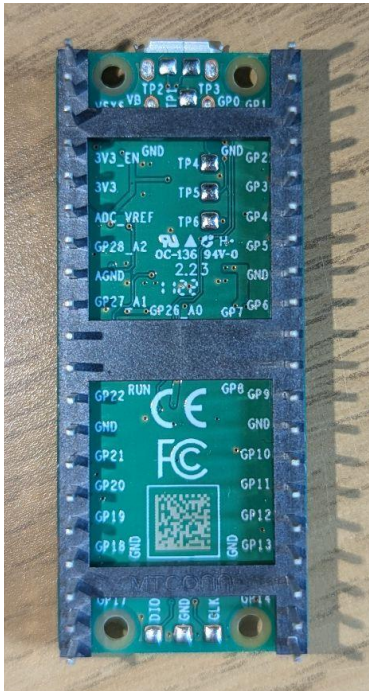
Connecting to Pico

It is tough to see in the picture however there is a great diagram here

On the underside of your Pico you will see that each PIN is labelled with what it is.

A lot of the PINs are labelled GP which means General purpose (Can be used for input and output)

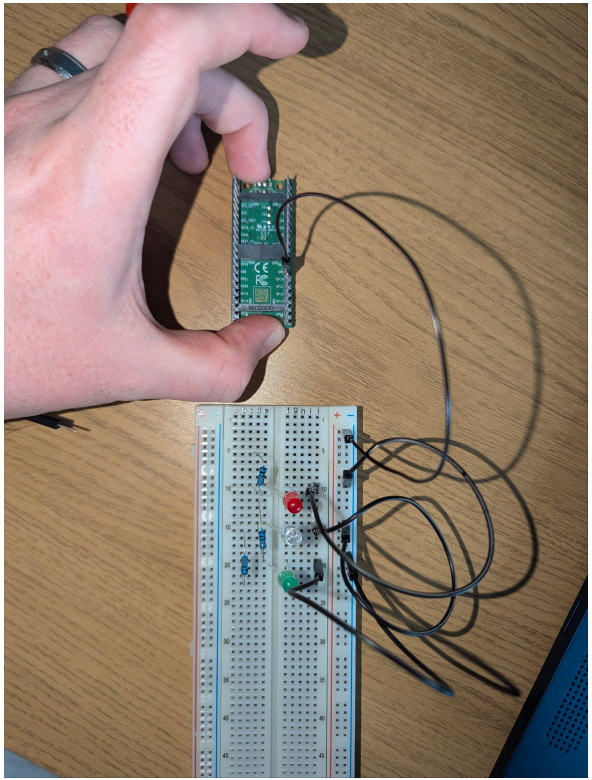
Some PINs are labelled GND this means ground and can be used to complete a circuit



Connecting to Pico

This allows all LEDs to connect to a single ground pin

Use a female to male jumper cable to connect a ground PIN to the negative rail on the right of your breadboard



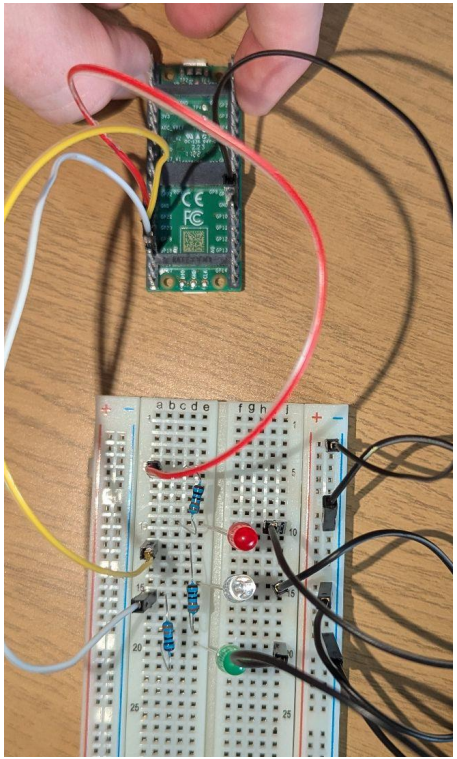
Connecting to Pico

I used GP20-18.

However you can use any GP pin you like

Use a female to male jumper cables to connect some General Purpose Pins to the breadboard.

Make sure to connect to the column that only has a resistor leg in and not the LED. This will make sure power goes through the resistor and to your LED rather than directly



Coding Time!

These libraries give MicroPython additional functionality

In Thonny start a new file and add the following import statements

```
import machine

import time
```

This gets the relevant libraries that we need to interact with our hardware

Let there be light!

For your challenge, set up the orange and green LEDs

We now have to tell the Pico where the LEDs are and how to turn them on.

This is how I set up the red LED

```
red_led = machine.Pin(20, machine.Pin.OUT)
```

This tells the pico that my red LED is on pin 20 and we are using it as an output

Turning the light on

When I did my driving theory I learnt the phrase:

“Red, Red and Amber, Green, Amber, Red”

To turn on the red light for 5 seconds the following code would be used

```
red_led.value(1) # Turns on LED

time.sleep(5) # Wait 5 seconds

red_led.value(0) # Turns LED off
```

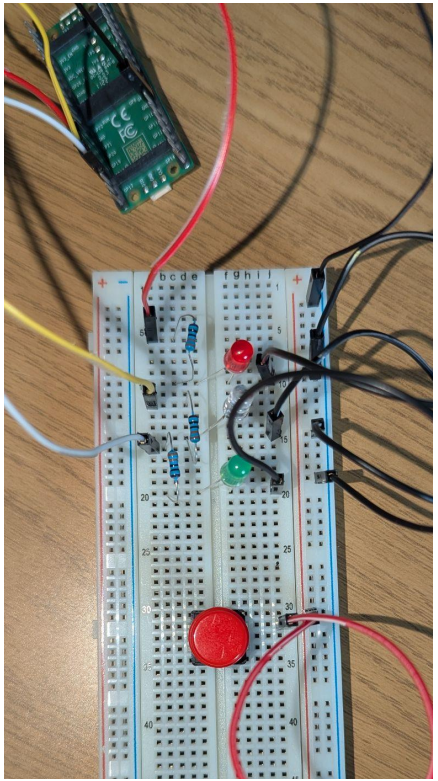
Try making the traffic light sequence.

Adding a button

Buttons don’t need a resistor as they have one built in

Place your button so it goes over the gap in the middle of the breadboard

Using a male to male jumper cable connect one prong on the right hand side on the breadboard to the positive rail on the right hand side of the breadboard.



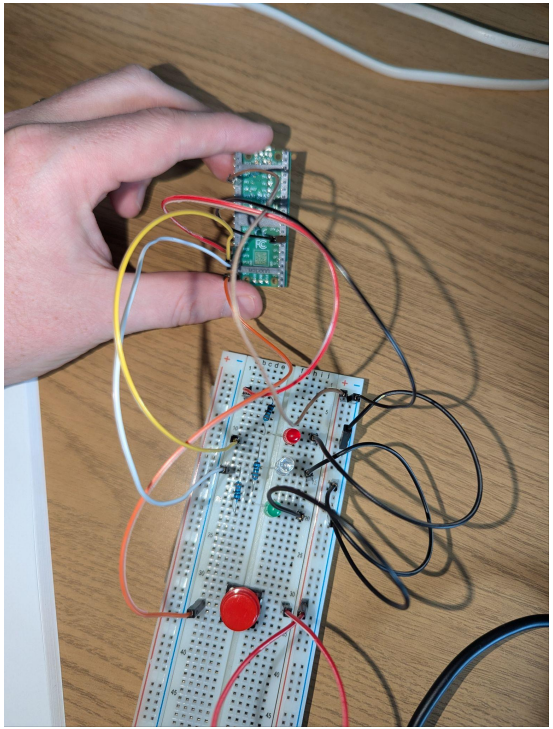
Adding a button

The power pin allows your button to draw power from the Pico

I connected the other side to Pin 17

Use a male to female jumper cable to connect the diagonal opposite prong on the button to a GP Pin on your Pico

Use another male to female jumper cable to connect the positive power rail on the right hand side of the breadboard to the Pin labelled 3v3 on the Pico



Coding the Button

This is similar to the LEDs however this is an input, the pull down allows the voltage to be pulled to zero when the button is not pressed

The following code sets up the button

```
button = machine.Pin(17, machine.Pin.IN, machine.Pin.PULL_DOWN)
```

This tells the Pico:

- The button is on Pin 17
- The button is an input
- Pull the value down to 0 (This is because we are connected to the power pin)

Testing the button

The while True loop allows us to repeat our code forever

Lets try turning the red LED on with the button use the following code:

```
while True:
    if button.value() == 1:
        red_led.value(1)
        time.sleep(5)
        red_led.value(0)
        time.sleep(2)
```

Challenge!

Red, Red and Amber, Green, Amber, Red

Use the internet to help you workout how to use the buzzer

Try coding your Pico so that the LEDs will follow the traffic light sequence and stay on green.

When the button is pressed the lights follow the correct sequence to red and then waiting for 10 seconds

After waiting it follows the sequence back to green.

SUPER CHALLENGE: Try adding a buzzer so that it beeps when the light is on red and stops after the light changes

References

Referencing any figures that are not mine

Halfacree, G. and Everard, B. (2021) *Get Started with Micropython on Raspberry Pi Pico*. Cambridge [United Kingdom]: Raspberry Pi Press.

License

I'm more than happy to share and allow people to remix my work, please just attribute and share alike