Santa Detector Test

What you will make

On Christmas Eve when all through the house,
not a creature was stirring, not even a mouse.
However one creature, a Code Club Creator, was not tucked up in bed
and was down stairs waiting for Santa instead.
They waited and waited and while they did,
They made sure to practice spotting Santa while they hid.

For this project you will create the Santa spotting device that the creator used to practice. It will measure how quick you can spot Santa when he is detected nearby.

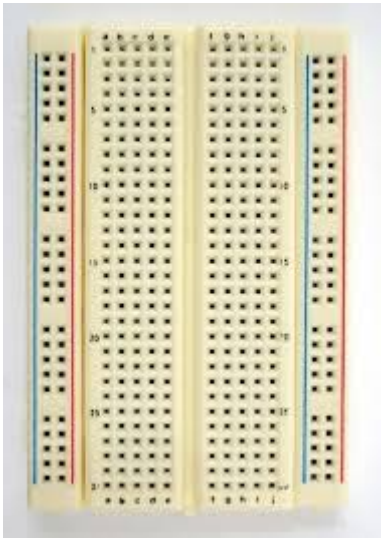
What do I need?

- ☐ 1 x Raspberry Pi Pico
- ☐ 1 x LED (preferably red)
- ☐ 1 x Resistor
- ☐ 1 x Button
- ☐ 2 x Male to Male Cables
- ☐ 3 x Male to Female Cables
- ☐ 1 x Breadboard

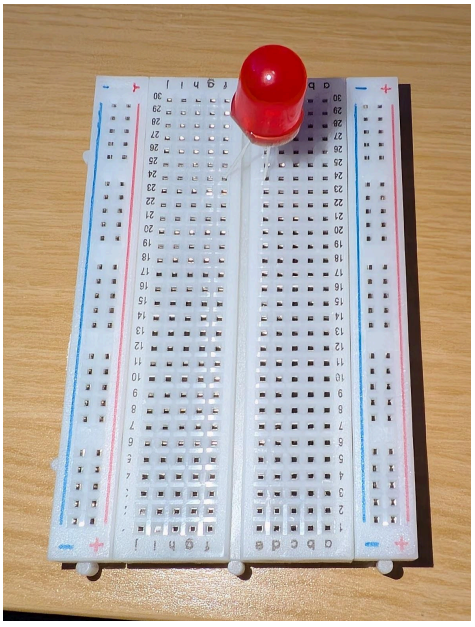
1 - Getting the circuit set up

You are going to set up the circuit on your breadboard. We will get everything attached first and look at programming it second.

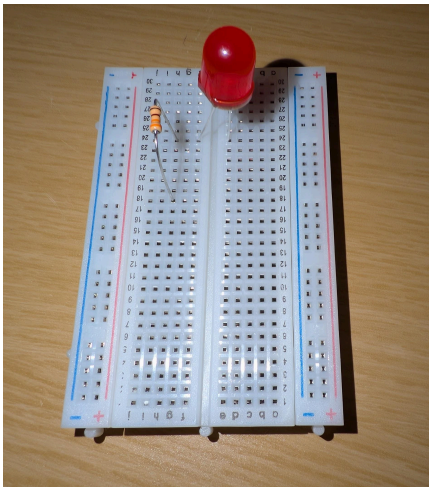
1. Place your breadboard on the table (portrait) it should look similar to the picture below where a deep grove or space runs up the middle



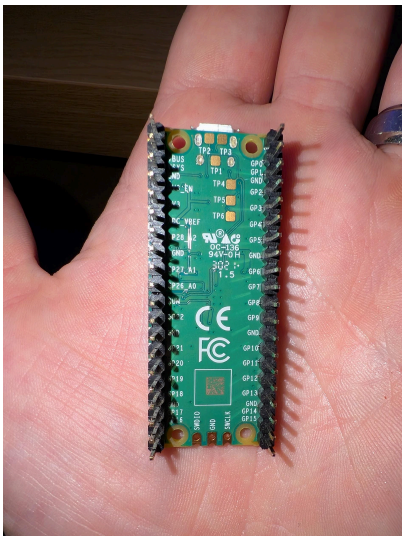
2. Place your LED in the middle of the breadboard so that the legs are straddling the channel. Put the long leg (Anode) on the left hand side of the channel.

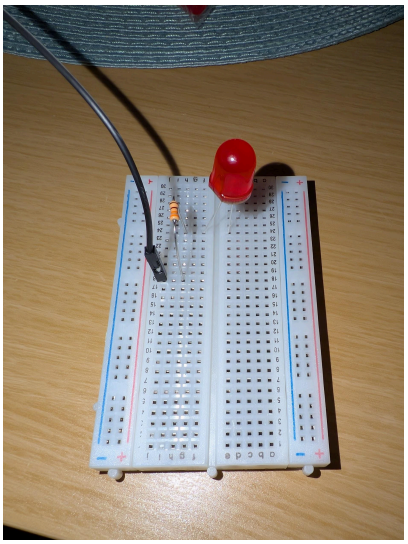


3. We now need to add a resistor so that we can protect our LED from electrical currents. **Place one leg of the resistor in the same row as the long leg of the LED and the other in a different row on the same side of the channel**

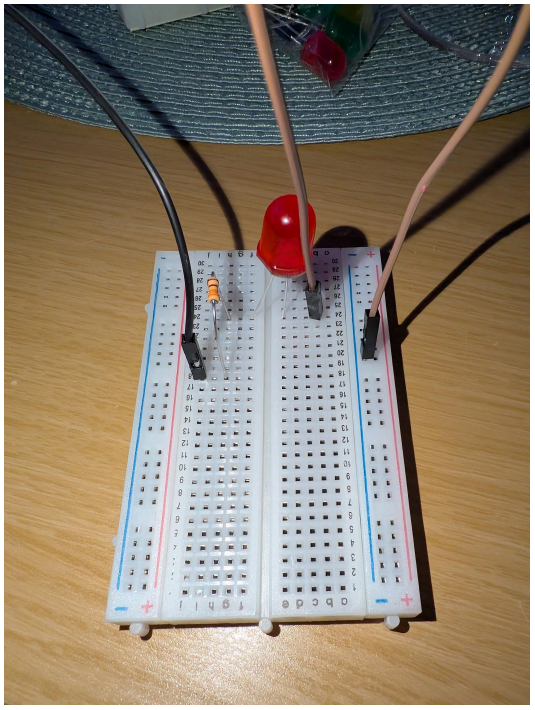


4. Connect a male to female cable to the same row as the resistor. Connect the female end to the pico to the pin labelled GP20

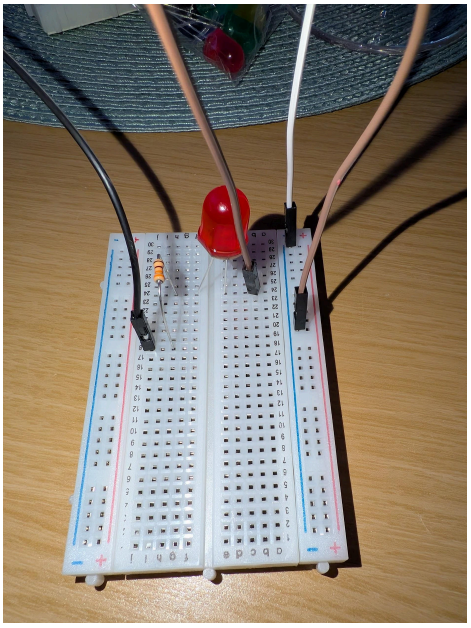




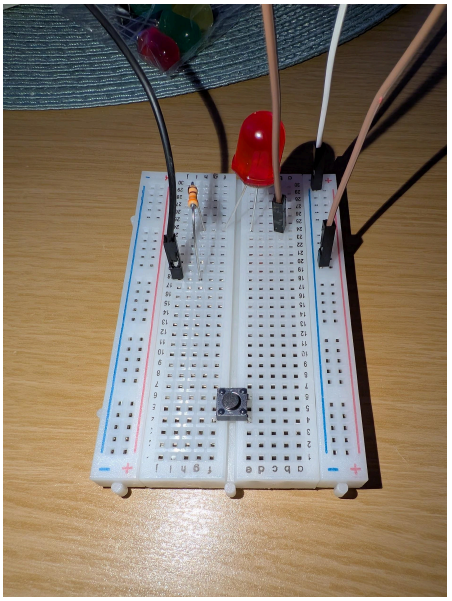
5. Connect a male to male cable from the short leg of the LED to the negative (blue) rail on the side



6. Connect a male to female cable from the top of the blue railing to any pin labelled GND on the pico

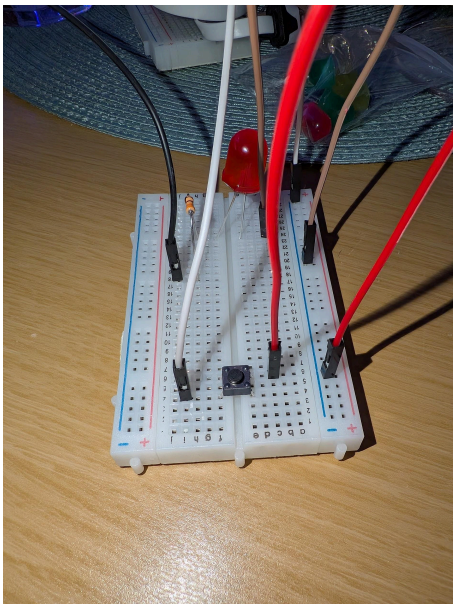


7. Place a button at the bottom of the breadboard straddling the channel (my button has 4 legs however some will have 2 and that is fine)



8. Connect a male to female cable from the button leg on the left hand side of the breadboard to pin GP18 on the pico

9. Connect a male to female cable from the button leg on the right (if your button has 4 legs, then make sure to connect the diagonal opposite leg) hand side of the breadboard to the blue rail



2 - Coding the Game

Before we can start building the game we need to set up our environment. To do that follow these instructions:

1. Open Thonny on your computer
2. Connect your pico to the computer using the USB cable
3. Click in the bottom right corner where it says “Thonny’s Python” and select micro-python (If you get stuck ask a fellow creator or ask a volunteer)

Now we can start coding. Use the following code to help you along the way.

```
1 from machine import Pin
2 from time import sleep, time_ns
3 from random import randint
```

The code above imports all the libraries that we need so that we can start making our game. Once we have this, we can start telling our program where the LED and button are.

```
5 led = Pin(20, Pin.OUT)
6 button = Pin(18, Pin.IN, Pin.PULL_UP)
```

It is now time to make your code turn on the LED after a certain amount of time. The below code shows you how to make the program sleep for a random amount of time between 1 and 3 seconds. Try changing it so that it is a random amount of time between 1 and 10 seconds.

```
sleep(random.randint(1,3))
```

Now we need to turn the LED on, get the time, and wait for the button to be pressed. You can do this by using the following code:

```
10 led.value(1)
11 start = time_ns()
12 finish = 0
13 while True:
14     if button.value() == 0:
15
```

Line 11 gets the current amount of nanoseconds that have paced since Jan 1970. Line 13 repeats the code below it indefinitely. Line 14 checks if the button has been pressed.

```
13 while True:
14     if button.value() == 0:
15         led.value(0)
16         finish = time_ns()
17         break
```

Modify your code so that when the button is pressed it turns the LED off, gets the finish time in nanoseconds, and then breaks out of the loop.

```
19 reaction = finish - start
20 print(reaction, "nanoseconds")
```

Finally, work out the reaction time and print it to the user. **Press the green play button to see if it works!**

You have successfully created the santa detector from the story at the beginning. WELL DONE 🎉

3 - Challenges

It is now time for you to take on a challenge. Try these:

1. Convert the nanoseconds reaction into seconds

2. Make the code run multiple times instead of once

3. Try adding a second button and making it a game between two creators