Moonhack 2023 - Rocket Blastoff - Python

Here's a challenge for you. We're going to create a rocket countdown using LEDs, a buzzer, a breadboard, and a Raspberry Pi Pico programmed with Python.

The goal is as follows:

- The LEDs will turn on one by one in a countdown sequence.
- The buzzer will make a beep with each countdown step.
- After the countdown, "Blast off!" will be printed on the console, and the buzzer will make a series of quick beeps.

What you'll need

Grab the following components in order to do this task:

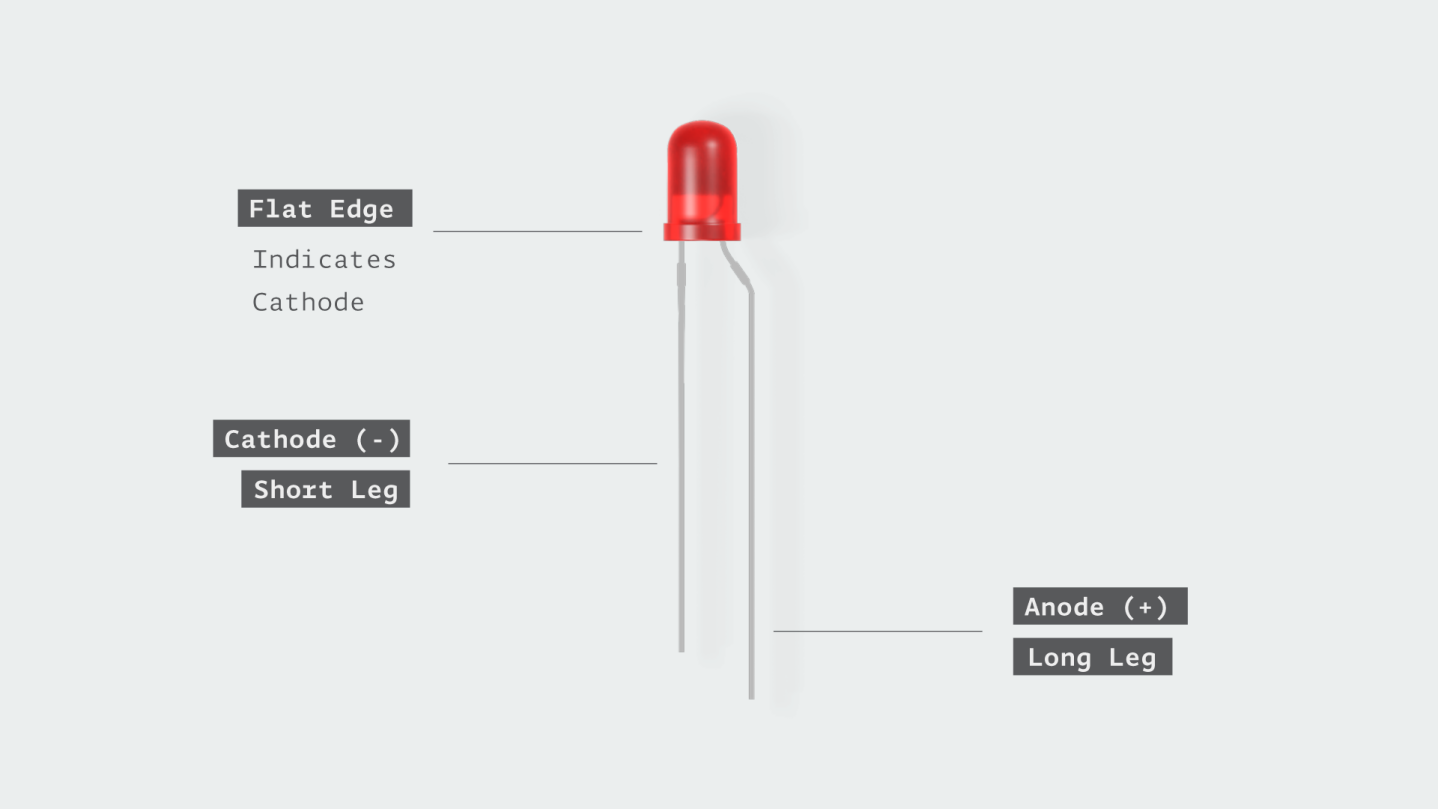
- Raspberry Pi Pico
- Breadboard
- 5 LEDs (Red, Yellow, Green, Blue, White for example but the colours really don't matter)
- 5 x 100 - 220-ohm resistors
- 1 x Buzzer
- Jumper wires

A quick note on circuits

Whenever we make a circuit we need to make sure it's "closed". If a circuit isn't closed then the power doesn't make it all the way round the circuit and the electronic components on it don't work.

Imagine a circuit is like a little race track for electricity. On this track, we have a battery, a tiny light bulb called an LED, and some wires. The battery is like the starting point, and it gives the "electricity racers" the energy they need to go around the track. When you connect the wires from the battery to the LED, you create a complete path or loop for the electricity to flow. As the electricity flows through the LED, it gives it the energy to light up! So, when you connect everything together, you'll see the LED turn on like magic! Remember, for the electricity to go round and round, the circuit has to be closed, meaning there are no gaps in the race track. That's a simple circuit!

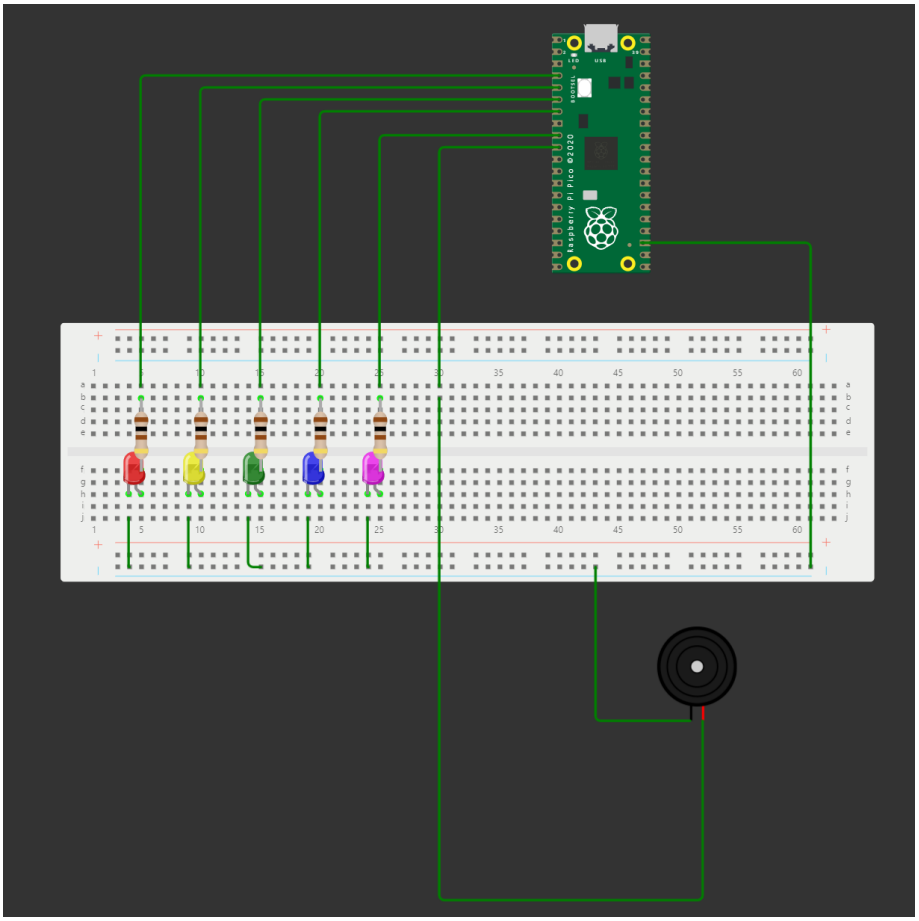
A quick note about LEDs



LEDs, which stand for Light Emitting Diodes, are tiny light bulbs that are used in all sorts of gadgets to give off light. They have two legs, and these legs are different! One leg is called the "anode," and it's the longer one. The anode is like the "entrance door" where electricity comes in. The other leg is shorter and is called the "cathode," which acts like the "exit door" where electricity goes out. It's important to know the difference because LEDs only let electricity flow in one direction, from the anode to the cathode. If you mix them up, the LED won't light up. So remember, connect the long leg (anode) to the positive side of your power source and the short leg (cathode) to the negative side, and your LED will shine bright!

Our project wiring

First we need to wire up our Pico, LEDs and buzzer. To do this we're going to use a breadboard and make the following circuit:



- LEDs: Connect LEDs to GPIO pins 2, 3, 4, 5, and 6 through resistors.
- Buzzer: Connect the positive leg of the buzzer to GPIO pin 7 and the negative leg to the ground (GND).
- Connect Ground: Make sure that all the ground pins of LEDs and the buzzer are connected to one of the GND pins on the Raspberry Pi Pico - this makes each of the six circuits “closed circuits”.

Now for the code!

1. Launch Thonny IDE and select the Raspberry Pi Pico as the device.
2. Enter the following Python code:

```
from machine import Pin
import time

# Initialize LEDs
led_pins = [2, 3, 4, 5, 6]
leds = [Pin(pin, Pin.OUT) for pin in led_pins]

# Initialize Buzzer
buzzer = Pin(7, Pin.OUT)

def turn_off_all_lights():
    for led in leds:
        led.off()

def beep(duration):
    buzzer.on()
    time.sleep(duration)
    buzzer.off()

def countdown():
    for i in reversed(range(5)):
        turn_off_all_lights()
        leds[i].on()
        beep(0.5)
        print(f"{i + 1}...")
        time.sleep(1)

    turn_off_all_lights()
    print("Blast off!")
    for _ in range(3):
        beep(0.2)
        time.sleep(0.2)

while True:
    countdown()
    time.sleep(2)
```

3. Save and Upload: Save the code onto your Raspberry Pi Pico.
4. Run the Code: Run the script in Thonny or press the Pico's reset button.