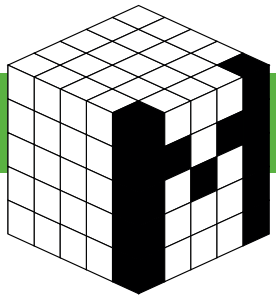


TNT Detonator with Countdown 3

Press a button to place the TNT and begin a five-second countdown timer! Five LEDs show each second of the countdown.

60 mins

1. Component list

In addition to your Raspberry Pi and breadboard, you will need:

8 x male to female jumpers

6 x male to male jumper

5 x momentary push button

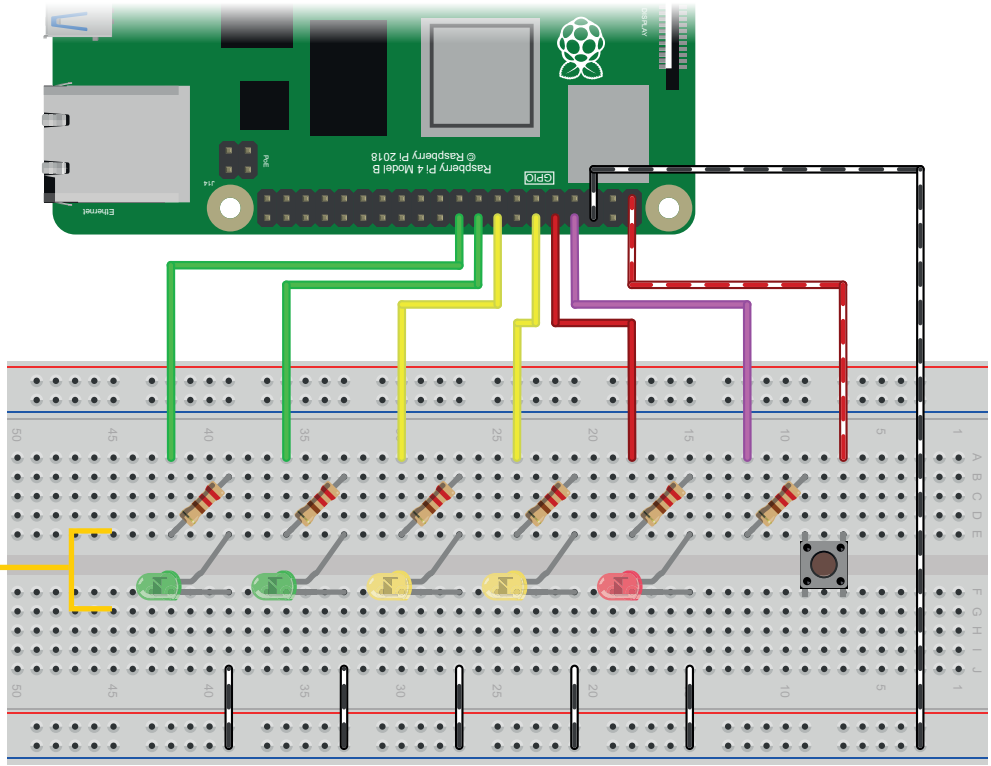
6 x 220 ohm resistor (red, red, brown)

1 x red LED

2 x yellow LEDs

2 x green LEDs

2. Build your circuit



Ensure your LEDs are inserted the correct way around or they will not light-up.

Long leg: anode (positive)

Short leg: cathode (negative)

- GPIO pins 26, 24, 22, 18, and 16 provide the positive current to the LEDs
- The resistors are important to prevent the LEDs from burning-out (permanently destroying them)
- The cathode of each LED is connected to the negative rail of the breadboard
- This allows us to only use one GROUND pin on the Raspberry Pi for all five LEDs

4. Coding

```
1.  # Import Minecraft Library
2.  import mcpi.minecraft as minecraft
3.  import mcpi.block as block
4.
5.  # Import the GPIO Libraries
6.  import RPi.GPIO as GPIO
7.
8.  # Import the time Library
9.  import time
10.
11. # Create an object and link it to Minecraft
12. mc = minecraft.Minecraft.create()
13.
14. # Set which pin numbering to use and turn off warnings
15. GPIO.setmode(GPIO.BOARD)
16. GPIO.setwarnings(False)
17.
18. # Assign the button to GPIO pin 8
19. buttonPin = 8
20. GPIO.setup(buttonPin, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
21.
22. # Store the LED GPIOs in a list
23. ledPins = [26, 24, 22, 18, 16]
24.
25. # Set up and turn off all LEDs
26. for i in ledPins:
27.     GPIO.setup(i, GPIO.OUT)
28.     GPIO.output(i, GPIO.LOW)
29.
30. # Main program
31. while True:
32.     # Check if the button is pressed
33.     if (GPIO.input(buttonPin)):
34.
35.         # Get the player's position
36.         position = mc.player.getTilePos()
37.
38.         # Craft the TNT (block ID 47) at the player's location
39.         # The final '1' arms the TNT.
40.         mc.setBlock(position.x, position.y, position.z, 47, 1)
41.
42.         # Begin the countdown sequence by iterating through the LEDs
43.         for i in ledPins:
44.
45.             # Turn on the LED assigned to postion i in the ledPins list
46.             GPIO.output(i, GPIO.HIGH)
47.
48.             # Wait one second
49.             time.sleep(1)
50.
51.             # Turn off the LED assigned to postion i in the ledPins list
52.             GPIO.output(i, GPIO.LOW)
53.
54.         # FOR loop ends
55.
56.         # Replace the TNT with a block of air (block ID 0)
57.         mc.setBlock(position.x, position.y, position.z, 0)
```

Import libraries which:

- **Give access to Minecraft and its blocks**
- **Give access to the GPIO pins on the Raspberry Pi**
- **Allow us to add a delay**

Define GPIO pin 8 as an INPUT for the button

Our GPIO pins are not numbered sequentially, so it's easier to store them in a list and iterate through them

Iterate through the list, defining each GPIO pin as OUTPUT and setting it to LOW (0 volts to turn it off)

Check if the button is pressed

Store the player's current position

Craft block 47 (TNT) at the player's position

Iterate through the list of LEDs. On each iteration, the GPIO stored in the current list position is set to HIGH (on), there is a one-second pause, then it is set to LOW (off).

When the iteration through the LED list is complete, the TNT block is replace with block 0 (air)

But where's the explosion? Unfortunately, TNT can only be detonated by the player, not directly by the program. By replacing the TNT block with air, we make it vanish. If you want a crater, modify your program to replace the TNT and several surrounding blocks with air.