03a LED Reaction Game with Two Buttons

Nicholas Bruzzese

November 22, 2024

Objective

Two players compete to press their button as soon as a lit LED turns on. The first button press will turn the LED off and declare the winner.

Parts Needed

- Raspberry Pi (any model with GPIO pins)
- 1 LED
- 1 220Ω resistor (for the LED)
- 2 push-buttons
- 2 10kΩ resistors (pull-down resistors for buttons)
- Breadboard
- Jumper wires

Circuit Setup

LED Setup

- Connect the **positive leg (long leg)** of the LED to a **GPIO pin (e.g., GPIO 22)** through a **220Ω resistor**.
- Connect the **negative leg (short leg)** of the LED to a **GND pin**.

Button Setup

- **Button 1:**
  - Connect one terminal of the button to **GPIO 17**.

  - Connect the other terminal to **GND** through a **10kΩ pull-down resistor**.

- **Button 2:**
  - Connect one terminal of the button to **GPIO 27**.
  - Connect the other terminal to **GND** through a **10kΩ pull-down resistor**.

Figure 1: Wiring Diagram

Python Code

This code uses the `RPi.GPIO` library to control the LED and detect button presses.

Listing 1: Reaction Game Code

```
import RPi.GPIO as GPIO
import time
import random

# Pin Definitions
LED_PIN = 22
BUTTON1_PIN = 17
BUTTON2_PIN = 27

# GPIO Setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)
GPIO.setup(BUTTON1_PIN, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(BUTTON2_PIN, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

# Function to reset the LED
def reset_led():
    GPIO.output(LED_PIN, GPIO.LOW)

try:
    print("Reaction game starting... Press your button when the LED
        lights up!")

    while True:
        reset_led()
        time.sleep(random.uniform(2, 5)) # Random delay
        GPIO.output(LED_PIN, GPIO.HIGH) # Light up the LED
        print("Go!")

        start_time = time.time()
        winner = None

        # Wait for a button press
        while winner is None:
            if GPIO.input(BUTTON1_PIN) == GPIO.HIGH:
                winner = "Player 1"
            elif GPIO.input(BUTTON2_PIN) == GPIO.HIGH:
                winner = "Player 2"

        # Turn off LED and declare winner
        GPIO.output(LED_PIN, GPIO.LOW)
        reaction_time = time.time() - start_time
        print(f"{winner} wins! Reaction time: {reaction_time:.3f}
```

```
            seconds")

        time.sleep(2) # Pause before next round

except KeyboardInterrupt:
    print("Exiting the game...")
finally:
    reset_led()
    GPIO.cleanup()
```

How It Works

Setup Phase

- The LED is initially off.
- A random delay is introduced to keep the reaction unpredictable.

Start Signal

- The LED lights up after the delay, signaling players to press their buttons.

Button Detection

- The first player to press their button sends a HIGH signal to their respective GPIO pin.
- The program detects the first button press and:
  - Turns off the LED.
  - Displays the winner and their reaction time.

Game Reset

- After a short pause, the game resets for the next round.

Running the Game

1. Save the script as `reaction_game.py`.
2. Run it with:

   ```
   python3 reaction_game.py
   ```

3. Press the buttons when the LED lights up. The program will indicate the winner and their reaction time.

Troubleshooting

Buttons Not Detected

- Check the wiring and ensure the GPIO pins match the code.
- Verify pull-down resistors are in place or configured internally (`PUD_DOWN`).

LED Not Lighting Up

- Ensure the LED is connected correctly with the resistor and the GPIO pin.