stemoutreach / **PicoBot** Public

Code Issues Pull requests Actions Projects Security Insights

PicoBot / 3-Code / **Motors_LED_Buzzer_Button_Worksheet_With_Functions.md**

stemoutreach Add files via upload 21702d3 · 14 hours ago History

Preview Code Blame 124 lines (94 loc) · 2.64 KB Raw

□ Raspberry Pi Worksheet: Motors + LED, Buzzer, and Button (Full Version)

Objective:
Combine everything you've learned so far—control a robot's motors, light an LED, and sound a buzzer with a button press. This version includes additional motor movement functions.

□ Wiring Summary

Component	Description	GPIO Pin
Motor A - IN1	OUT1 on L298N	GP6
Motor A - IN2	OUT2 on L298N	GP7

Component	Description	GPIO Pin
Enable A	ENA	GP8
Motor B - IN3	OUT3 on L298N	GP4
Motor B - IN4	OUT4 on L298N	GP3
Enable B	ENB	GP2
LED	Long leg to GPIO via resistor	GP28
Buzzer	Positive to GPIO	GP10
Button	One leg to GPIO, one to GND	GP14

□ Python Code with All Movement Functions

```
from machine import Pin
import time

# Motor setup
In1 = Pin(6, Pin.OUT)
In2 = Pin(7, Pin.OUT)
EN_A = Pin(8, Pin.OUT)

In3 = Pin(4, Pin.OUT)
In4 = Pin(3, Pin.OUT)
EN_B = Pin(2, Pin.OUT)

EN_A.high()
EN_B.high()
```

```
# LED and buzzer setup
led = Pin(28, Pin.OUT)
buzzer = Pin(10, Pin.OUT)

# Button input
button = Pin(14, Pin.IN, Pin.PULL_UP)  # Using internal pull-up, button connected to GND

# Movement functions
def move_forward():
    In1.high()
    In2.low()
    In3.high()
    In4.low()

def move_backward():
    In1.low()
    In2.high()
    In3.low()
    In4.high()

def turn_left():
    In1.low()
    In2.high()
    In3.high()
    In4.low()

def turn_right():
    In1.high()
    In2.low()
    In3.low()
    In4.high()

def stop_motors():
    In1.low()
    In2.low()
    In3.low()
    In4.low()

# Alert functions
def alert_on():
    led.high()
    buzzer.high()
    print("Alert ON")

def alert_off():
    led.low()
    buzzer.low()
    print("Alert OFF")

# Main loop
print("Press the button to move forward with lights and sound!")

while True:
    if button.value() == 0:  # Button pressed (LOW)
        move_forward()
        alert_on()
        time.sleep(1)
        stop_motors()
        alert_off()
        time.sleep(0.5)
```

✍️ Additional Challenge Ideas

1. Use the new `move_backward()` function to drive in reverse when the button is held for 2 seconds.
2. Use `turn_left()` and `turn_right()` when using two buttons.
3. Flash the LED while the robot is moving.
4. Use timed sequences to navigate a mini maze using forward and turn movements.

✅ Reflection Questions

- What happens when one motor spins forward and the other spins backward?
- How can you slow down one motor or both motors?
- Can you combine `turn_left()` and `move_forward()` to curve?

Have fun building and programming your robot! □□□