**Πείραμα με Βομβητή**

1. **Περιγραφή:**

Ο βομβητής είναι μια ολοκληρωμένη δομή ηλεκτρονικού ηχητικού σήματος που οδηγείται από τροφοδοτικό συνεχούς ρεύματος. Χρησιμοποιείται ευρέως σε υπολογιστές, εκτυπωτές, φωτοτυπικά μηχανήματα, συναγερμούς, ηλεκτρονικά παιχνίδια, ηλεκτρονικό εξοπλισμό αυτοκινήτων, τηλέφωνα, χρονοδιακόπτες και άλλα ηλεκτρονικά προϊόντα. Ο ενεργός βομβητής μπορεί να λειτουργεί συνεχώς όταν είναι απευθείας συνδεδεμένος με την ονομαστική παροχή ρεύματος 5V.

1. **Υλικά:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **1\* Raspberry pi** | **1\* GPIO Extension board** | **1\* 40 pin colorful jumper wires** |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **1\* Breadboard** | **1\* Active Buzzer Module** | **Jumper Wires** |

1. **Γνώση υλικού:**

Ορισμός ακίδων

S: Είσοδος σήματος

V: Τροφοδοσία ισχύος 3,3V/5V

G: γείωση

A close-up of a blue circuit board

Description automatically generated

**Πειραματική συσκευή:**

■ Πίνακας ελέγχου Raspberry Pi

■ Μονάδα βομβητή

■ Γραμμές σύνδεσης

**Πειραματική σύνδεση:**

|  |  |
| --- | --- |
| The Buzzer module | Raspberry Pi |
| V | 5V |
| G | GND |
| S | 13 |

Ανεβάστε τον κώδικα στην αναπτυξιακή πλακέτα και εκτελέστε τον

1. **Πειραματικό συμπέρασμα:**

Μεταφορτώστε το πρόγραμμα στον ελεγκτή και ακούστε τον συναγερμό του βομβητή με περίοδο 1s.

**Βεβαιωθείτε ότι έχετε εγκαταστήσει τη βιβλιοθήκη RPi.GPIO στο Raspberry Pi σας, αν δεν είναι ήδη εγκατεστημένη. Μπορείτε να το κάνετε αυτό με την ακόλουθη εντολή στη γραμμή εντολών(command line):**

**pip install RPi.GPIO**

1. **Κώδικας Python:**

import RPi.GPIO as GPIO

import time

buzzPin = 13  # Define the buzzer pin

def setup():

    GPIO.setmode(GPIO.BCM)

    GPIO.setup(buzzPin, GPIO.OUT)  # Set the buzzer pin as an output

    print("Buzzer pin setup complete")

def loop():

    try:

        while True:

            GPIO.output(buzzPin, GPIO.HIGH)  # Output high level signal

            time.sleep(1)  # Delay for 1 second

            GPIO.output(buzzPin, GPIO.LOW)  # Output low level signal

            time.sleep(1)  # Delay for 1 second

    except KeyboardInterrupt:

        GPIO.cleanup()  # Clean up GPIO on Ctrl+C exit

if \_\_name\_\_ == '\_\_main\_\_':

    setup()

    loop()

**Παραλλαγή Πειράματος:**

import RPi.GPIO as GPIO

import time

# Set the GPIO mode

GPIO.setmode(GPIO.BCM)

# Define the GPIO pins for the buzzer and initialize them

buzzer\_pin = 13

GPIO.setup(buzzer\_pin, GPIO.OUT)

try:

    # Activate the buzzer for 1 second

    GPIO.output(buzzer\_pin, GPIO.HIGH)

    time.sleep(1)

    GPIO.output(buzzer\_pin, GPIO.LOW)

except KeyboardInterrupt:

    GPIO.cleanup()  # Clean up GPIO on Ctrl+C

finally:

    GPIO.cleanup()  # Clean up GPIO on script exit

**Παραλλαγή Πειράματος Με Μελωδία:**

import RPi.GPIO as GPIO

import time

# Set the GPIO mode

GPIO.setmode(GPIO.BCM)

# Define the GPIO pin for the buzzer and initialize it

buzzer\_pin = 13

GPIO.setup(buzzer\_pin, GPIO.OUT)

# Define a simple melody

melody = [261.63, 293.66, 329.63, 349.23, 392.00, 440.00, 493.88, 523.25]

# Function to play a note

def play\_note(frequency, duration):

    if frequency == 0:

        time.sleep(duration)

    else:

        period = 1.0 / frequency

        half\_period = period / 2

        cycles = int(duration \* frequency)

        for \_ in range(cycles):

            GPIO.output(buzzer\_pin, GPIO.HIGH)

            time.sleep(half\_period)

            GPIO.output(buzzer\_pin, GPIO.LOW)

            time.sleep(half\_period)

try:

    for note in melody:

        play\_note(note, 0.5)  # Play each note for 0.5 seconds

except KeyboardInterrupt:

    GPIO.cleanup()  # Clean up GPIO on Ctrl+C

finally:

    GPIO.cleanup()  # Clean up GPIO on script exit