2023 Undergraduate Research Conference

Computer & Information Sciences Department

Raspberry Pi Memory Game with LED Feedback and Decimal to Binary Number Converter

Jisun Kim

Advisor: Dr. Yonghong Tong

Topic

The topic of this project is the development of two Raspberry Pi projects using Python programming language. The first project is a memory game that tests the player's ability to remember a random pattern of LED lights. The second project is a decimal-binary converter that allows the player to input decimal numbers and displays their binary equivalent on a 10-segment bar LED.

Goals

The main goal of this project is to showcase the use of Raspberry Pi and Python programming language in developing simple but fun and educational projects. The memory game helps improve memory skills, while the decimal-binary converter project helps players understand binary numbers and how to convert between decimal and binary. By completing these projects, players can develop their programming skills and learn more about electronics.

Introduction

Raspberry Pi is a credit-card-sized single-board computer that has gained popularity for its versatility and ease of use. The projects presented here showcase the use of Python programming language and Raspberry Pi to develop a memory game and a decimal-binary converter.

The memory game is designed to test the player's ability to remember a pattern of 10 LED lights that light up in a random sequence. The player has to input the sequence using two buttons, a green LED lights up if the player is correct, and a red LED lights up if the player is incorrect.

The decimal-binary converter allows the player to input decimal numbers between 0 and 1023 using a keyboard, and the 10-segment bar LED displays the binary equivalent of the input number.

Design and Development

The memory game project uses Raspberry Pi GPIO pins to control the LED lights and push-buttons. The LED lights are arranged in a 10-segment bar and are controlled using the GPIO.output() function. The push-buttons are connected to GPIO pins and monitored using the GPIO.input() function. The program logic is implemented in Python using the RPi.GPIO module. The game starts by randomly lighting up some of the LED lights for 5 seconds, after which the player has to input the pattern using the push-buttons. If the player inputs the pattern correctly, a green LED lights up; otherwise, a red LED lights up.

The decimal-binary converter project also uses Raspberry Pi GPIO pins to control the 10-segment bar LED. A keyboard allows the player to input numbers between 0 and 1023. The input numbers are converted to binary using the Python bin() function, and the binary digits are displayed on the 10-segment bar LED using the GPIO.output() function. The program logic is implemented in Python using the RPi.GPIO module.

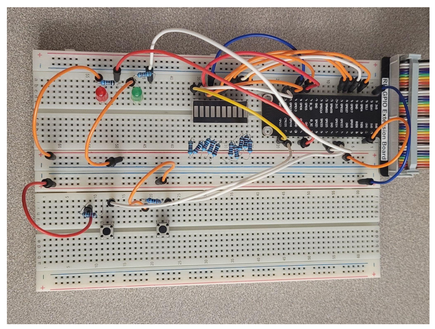
Future Work

The projects can be further extended and improved in various ways.

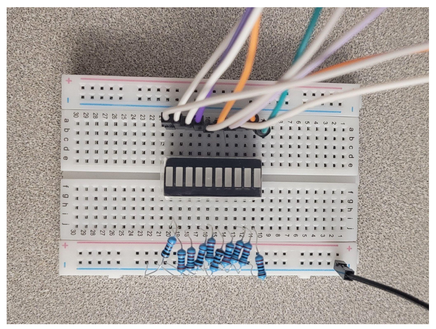
For example, the memory game can be made more challenging by increasing the number of LED lights in the pattern or adding sound effects.

The decimal-binary converter can be enhanced to allow the player to input numbers in other number systems, such as hexadecimal and octal. The projects can also be integrated with a graphical user interface (GUI) to make them more user-friendly and attractive. Overall, the projects provide a foundation for further exploration and creativity in Raspberry Pi programming.

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

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

led_pins = [21,26,19,13,6,5,22,27,17,4]
green_led = 25
red_led = 12
on_button = 20
off_button = 24

GPIO.setup(led_pins, GPIO.OUT)
GPIO.setup(green_led, GPIO.OUT)
GPIO.setup(red_led, GPIO.OUT)
GPIO.setup(on_button, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
GPIO.setup(off_button, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

pattern = [random.choice([GPIO.HIGH, GPIO.LOW]) for _ in range(10)]
GPIO.output(led_pins, pattern)

time.sleep(5)

GPIO.output(led_pins, GPIO.LOW)
player_pattern = []
```



```
for i in range(10):
    while True:
        if GPIO.input(on_button) == GPIO.HIGH:
            player_pattern.append(GPIO.HIGH)
            print("on")
            time.sleep(1)
            break
        if GPIO.input(off_button) == GPIO.HIGH:
            player_pattern.append(GPIO.LOW)
            print("off")
            time.sleep(1)
            break
        time.sleep(0.1)

if player_pattern == pattern:
    GPIO.output(green_led, GPIO.HIGH)
    GPIO.output(red_led, GPIO.LOW)
else:
    GPIO.output(green_led, GPIO.LOW)
    GPIO.output(red_led, GPIO.HIGH)
time.sleep(2)

GPIO.output(led_pins, GPIO.LOW)
GPIO.output(green_led, GPIO.LOW)
GPIO.output(red_led, GPIO.LOW)
```



```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
led_pins = [2, 1, 7, 8, 25, 24, 23, 18, 15, 14]
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pins, GPIO.OUT)
try:
    while True:
        input_num = input("Enter a number between 0 and 1023: ")
        if input_num is None:
            print("Input is empty.")
        else:
            input_num = int(input_num)
            print("Input number:", input_num)
            binary = bin(input_num)[2:].zfill(10)
            print("Binary number:", binary)
            for i in range(10):
                if binary[i] == '1':
                    GPIO.output(led_pins[i], GPIO.HIGH)
                else:
                    GPIO.output(led_pins[i], GPIO.LOW)
        time.sleep(0.1)
finally:
    GPIO.cleanup()
```