Khoi Duong

Prof. Yang

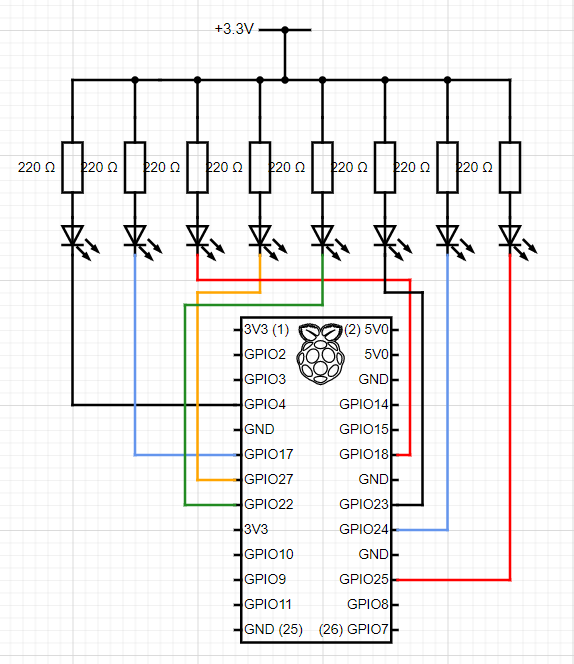
CE450L

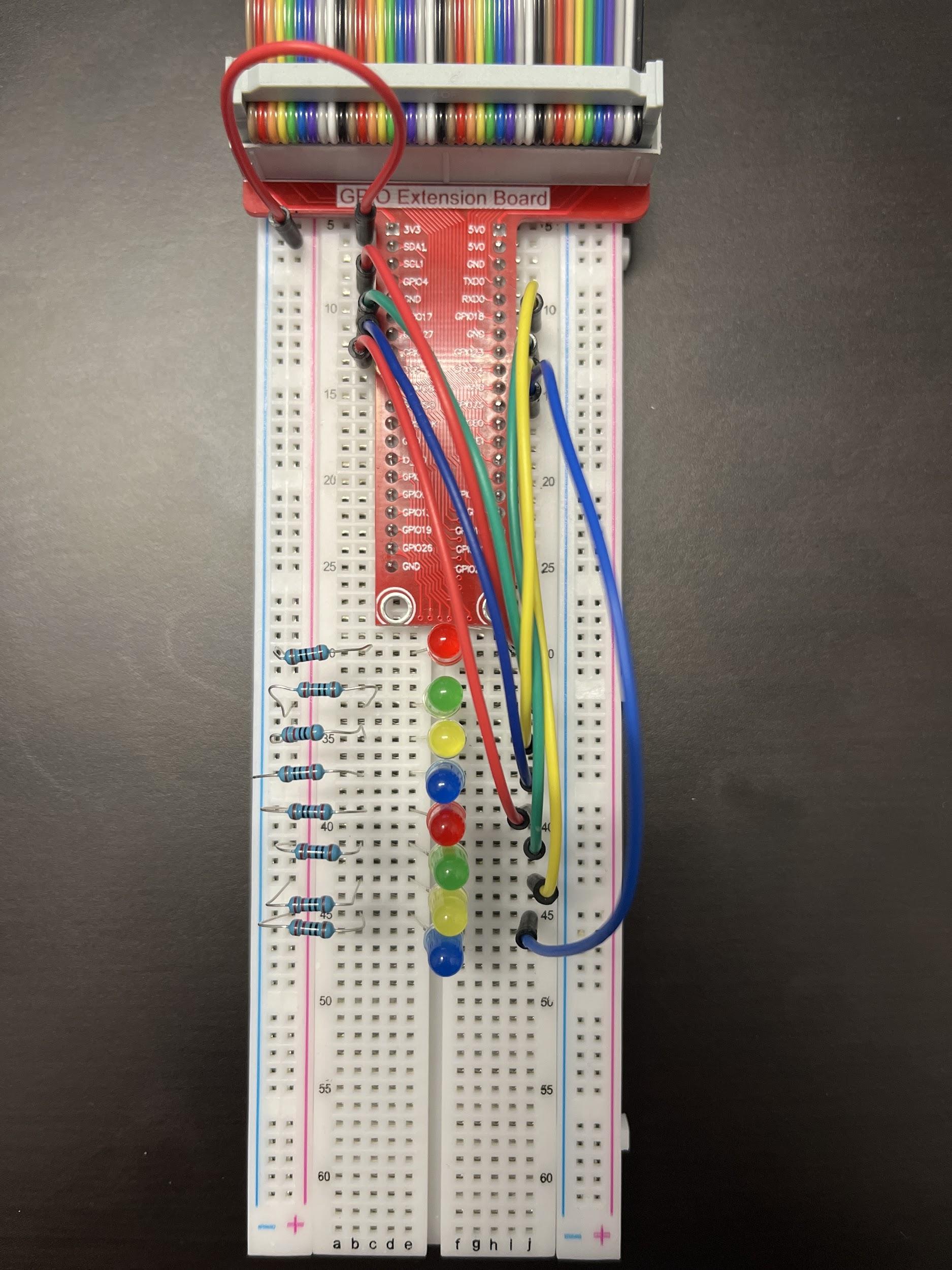
10/12/2022

LAB#2

GitHub link: <https://github.com/MynameisKoi/CE450L/tree/main/Lab%232>

Breadboard setup for all 3 exercises







Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%232/1.py>

#!/usr/bin/env python3

import RPi.GPIO as GPIO

import time

from sys import version\_info

if version\_info.major == 3:

raw\_input = input

# Set 8 Pins for 8 LEDs.

LedPins = [4,17,18,27,22,23,24,25]

# Define a function to print message at the beginning

def print\_message():

print ("========================================")

print ("| 8 LEDs |")

print ("| ------------------------------ |")

print ("| LED0 connect to GPIO4 |")

print ("| LED1 connect to GPIO17 |")

print ("| LED2 connect to GPIO18 |")

print ("| LED3 connect to GPIO27 |")

print ("| LED4 connect to GPIO22 |")

print ("| LED5 connect to GPIO23 |")

print ("| LED6 connect to GPIO24 |")

print ("| LED7 connect to GPIO25 |")

print ("| |")

print ("| Flow LED effect |")

print ("| |")

print ("| SunFounder|")

print ("========================================\n")

print ("Program is running...")

print ("Please press Ctrl+C to end the program...")

raw\_input ("Press Enter to begin\n")

# Define a setup function for some setup

def setup():

# Set the GPIO modes to BCM Numbering

GPIO.setmode(GPIO.BCM)

# Set all LedPin's mode to output,

# and initial level to High(3.3v)

GPIO.setup(LedPins, GPIO.OUT, *initial*=GPIO.HIGH)

# Define a main function for main process

def main():

# Print messages

print\_message()

leds = ['-', '-', '-', '-', '-', '-', '-', '-']

while True:

# Turn LED on from left to right

print ("From left to right.")

for pin in LedPins:

#print pin

GPIO.output(pin, GPIO.LOW)

leds[LedPins.index(pin)] = 0 # Show which led is on

print (leds)

time.sleep(0.1)

GPIO.output(pin, GPIO.HIGH)

leds[LedPins.index(pin)] = '-' # Show the led is off

# Turn LED off from right to left

print ("From right to left.")

for pin in reversed(LedPins):

#print pin

GPIO.output(pin, GPIO.LOW)

leds[LedPins.index(pin)] = 0 # Show which led is on

print (leds)

time.sleep(0.1)

GPIO.output(pin, GPIO.HIGH)

leds[LedPins.index(pin)] = '-' # Show the led is off

# Define a destroy function for clean up everything after

# the script finished

def destroy():

# Turn off all LEDs

GPIO.output(LedPins, GPIO.HIGH)

# Release resource

GPIO.cleanup()

# If run this script directly, do:

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

setup()

try:

main()

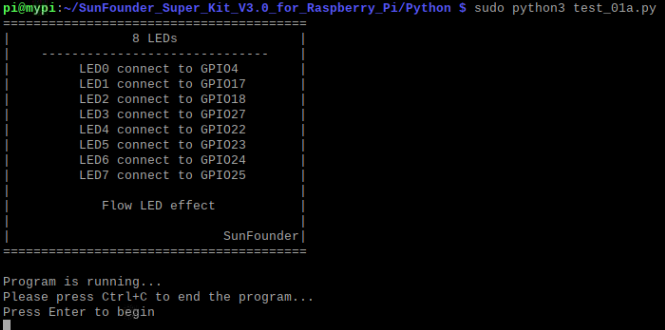
# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()

Run code and demonstration:



Video link:

<https://github.com/MynameisKoi/CE450L/blob/main/Lab%232/Demonstrations/1.mp4>



Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%232/2.py>

#!/usr/bin/env python3

import RPi.GPIO as GPIO

import time

from sys import version\_info

if version\_info.major == 3:

raw\_input = input

# Set 8 Pins for 8 LEDs.

LedPins = [4,17,18,27,22,23,24,25]

# Define a function to print message at the beginning

def print\_message():

print ("========================================")

print ("| 8 LEDs |")

print ("| ------------------------------ |")

print ("| LED0 connect to GPIO4 |")

print ("| LED1 connect to GPIO17 |")

print ("| LED2 connect to GPIO18 |")

print ("| LED3 connect to GPIO27 |")

print ("| LED4 connect to GPIO22 |")

print ("| LED5 connect to GPIO23 |")

print ("| LED6 connect to GPIO24 |")

print ("| LED7 connect to GPIO25 |")

print ("| |")

print ("| Flow LED effect |")

print ("| |")

print ("| SunFounder|")

print ("========================================\n")

print ("Program is running...")

print ("Please press Ctrl+C to end the program...")

raw\_input ("Press Enter to begin\n")

# Define a setup function for some setup

def setup():

# Set the GPIO modes to BCM Numbering

GPIO.setmode(GPIO.BCM)

# Set all LedPin's mode to output,

# and initial level to High(3.3v)

GPIO.setup(LedPins, GPIO.OUT, *initial*=GPIO.HIGH)

# Define a main function for main process

def main():

# Print messages

print\_message()

leds = ['-', '-', '-', '-', '-', '-', '-', '-']

while True:

# Turn LED on from left to right

print ("From left to right.")

for a in range (0,7):

pin1 = LedPins[a]

pin2 = LedPins[len(LedPins)-1-a]

#print pin

GPIO.output(pin1, GPIO.LOW)

GPIO.output(pin2, GPIO.LOW)

leds[LedPins.index(pin1)] = 0 # Show which led is on

leds[LedPins.index(pin2)] = 0

print (leds)

time.sleep(0.1)

GPIO.output(pin1, GPIO.HIGH)

GPIO.output(pin2, GPIO.HIGH)

leds[LedPins.index(pin1)] = '-' # Show the led is off

leds[LedPins.index(pin2)] = '-'

# Turn LED off from right to left

print ("From right to left.")

for a in range (0,7):

pin1 = LedPins[len(LedPins)-1-a]

pin2 = LedPins[a]

#print pin

GPIO.output(pin1, GPIO.LOW)

GPIO.output(pin2, GPIO.LOW)

leds[LedPins.index(pin1)] = 0 # Show which led is on

leds[LedPins.index(pin2)] = 0

print (leds)

time.sleep(0.1)

GPIO.output(pin1, GPIO.HIGH)

GPIO.output(pin2, GPIO.HIGH)

leds[LedPins.index(pin1)] = '-' # Show the led is off

leds[LedPins.index(pin2)] = '-'

# Define a destroy function for clean up everything after

# the script finished

def destroy():

# Turn off all LEDs

GPIO.output(LedPins, GPIO.HIGH)

# Release resource

GPIO.cleanup()

# If run this script directly, do:

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

setup()

try:

main()

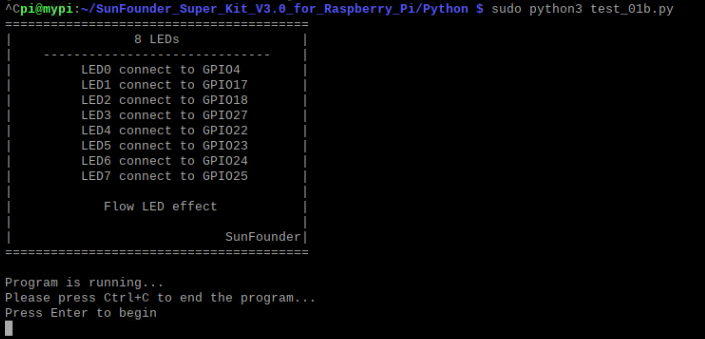
# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()

Run code and demonstration:



Video link:

<https://github.com/MynameisKoi/CE450L/blob/main/Lab%232/Demonstrations/2.mp4>



Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%232/3.py>

#!/usr/bin/env python3

import RPi.GPIO as GPIO

import time

from sys import version\_info

if version\_info.major == 3:

raw\_input = input

# Set 8 Pins for 8 LEDs.

LedPins = [4,17,18,27,22,23,24,25]

# Define a function to print message at the beginning

def print\_message():

print ("========================================")

print ("| 8 LEDs |")

print ("| ------------------------------ |")

print ("| LED0 connect to GPIO4 |")

print ("| LED1 connect to GPIO17 |")

print ("| LED2 connect to GPIO18 |")

print ("| LED3 connect to GPIO27 |")

print ("| LED4 connect to GPIO22 |")

print ("| LED5 connect to GPIO23 |")

print ("| LED6 connect to GPIO24 |")

print ("| LED7 connect to GPIO25 |")

print ("| |")

print ("| Flow LED effect |")

print ("| |")

print ("| SunFounder|")

print ("========================================\n")

print ("Program is running...")

print ("Please press Ctrl+C to end the program...")

raw\_input ("Press Enter to begin\n")

# Define a setup function for some setup

def setup():

# Set the GPIO modes to BCM Numbering

GPIO.setmode(GPIO.BCM)

# Set all LedPin's mode to output,

# and initial level to High(3.3v)

GPIO.setup(LedPins, GPIO.OUT, *initial*=GPIO.HIGH)

# Define a main function for main process

def main():

# Print messages

print\_message()

leds = ['-', '-', '-', '-', '-', '-', '-', '-']

while True:

# Turn LED on from left to right

print ("From side to middle.")

for a in range (0,4):

pin1 = LedPins[a]

pin2 = LedPins[len(LedPins)-1-a]

#print pin

GPIO.output(pin1, GPIO.LOW)

GPIO.output(pin2, GPIO.LOW)

leds[LedPins.index(pin1)] = 0 # Show which led is on

leds[LedPins.index(pin2)] = 0

print (leds)

time.sleep(0.1)

GPIO.output(pin1, GPIO.HIGH)

GPIO.output(pin2, GPIO.HIGH)

leds[LedPins.index(pin1)] = '-' # Show the led is off

leds[LedPins.index(pin2)] = '-'

# Turn LED off from right to left

print ("From middle to side.")

for a in range (3,-1,-1):

pin1 = LedPins[len(LedPins)-1-a]

pin2 = LedPins[a]

#print pin

GPIO.output(pin1, GPIO.LOW)

GPIO.output(pin2, GPIO.LOW)

leds[LedPins.index(pin1)] = 0 # Show which led is on

leds[LedPins.index(pin2)] = 0

print (leds)

time.sleep(0.1)

GPIO.output(pin1, GPIO.HIGH)

GPIO.output(pin2, GPIO.HIGH)

leds[LedPins.index(pin1)] = '-' # Show the led is off

leds[LedPins.index(pin2)] = '-'

# Define a destroy function for clean up everything after

# the script finished

def destroy():

# Turn off all LEDs

GPIO.output(LedPins, GPIO.HIGH)

# Release resource

GPIO.cleanup()

# If run this script directly, do:

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

setup()

try:

main()

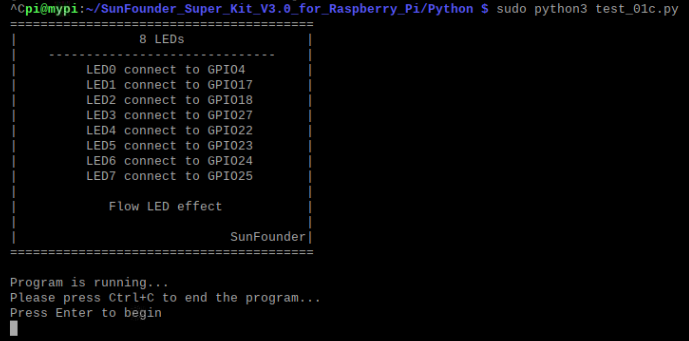
# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()

Run code and demonstration:



Video link:

<https://github.com/MynameisKoi/CE450L/blob/main/Lab%232/Demonstrations/3.mp4>