Khoi Duong

Prof. Yang

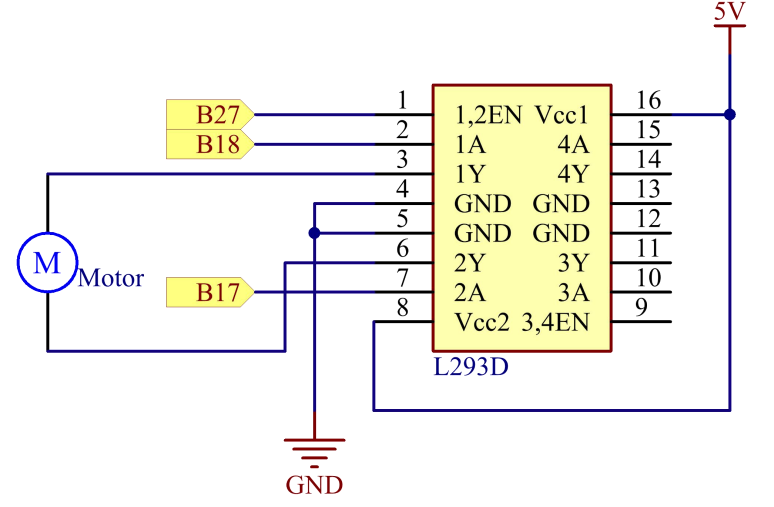
CE450L

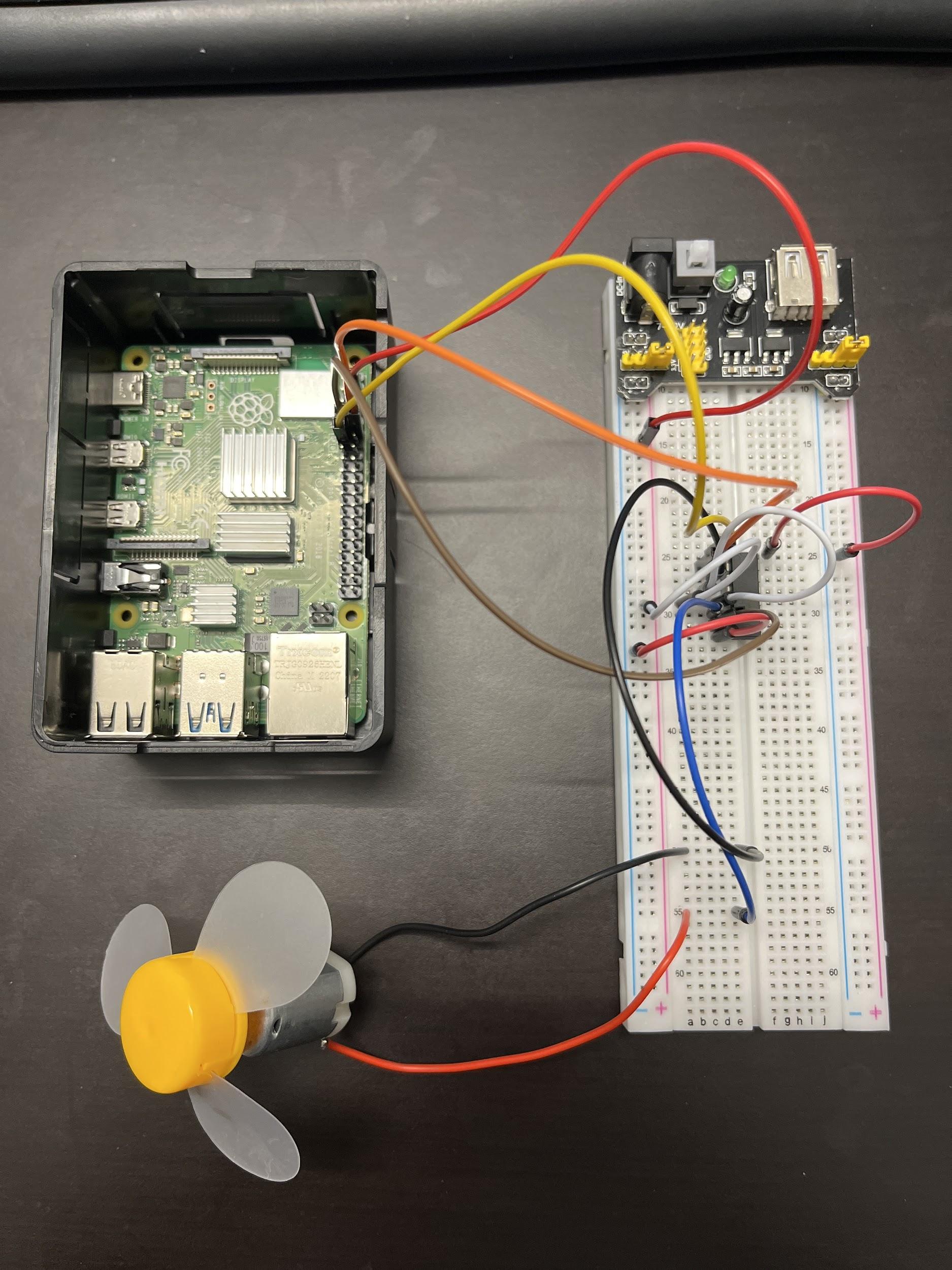
11/5/2022

LAB#7

GitHub link: <https://github.com/MynameisKoi/CE450L/tree/main/Lab%237>

Breadboard setup:





Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%237/DC_motor.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 up pins

MotorPin1 = 17

MotorPin2 = 18

MotorEnable = 27

def print\_message():

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

print ("| Motor |")

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

print ("| Motor pin 1 connect to GPIO17 |")

print ("| Motor pin 2 connect to GPIO18 |")

print ("| Motor enable connect to GPIO27 |")

print ("| |")

print ("| Controlling a motor |")

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")

def setup():

# Set the GPIO modes to BCM Numbering

GPIO.setmode(GPIO.BCM)

# Set pins to output

GPIO.setup(MotorPin1, GPIO.OUT)

GPIO.setup(MotorPin2, GPIO.OUT)

GPIO.setup(MotorEnable, GPIO.OUT, *initial*=GPIO.LOW)

# Define a motor function to spin the motor

# direction should be

# 1(clockwise), 0(stop), -1(counterclockwise)

def motor(*direction*):

# Clockwise

if *direction* == 1:

# Set direction

GPIO.output(MotorPin1, GPIO.HIGH)

GPIO.output(MotorPin2, GPIO.LOW)

# Enable the motor

GPIO.output(MotorEnable, GPIO.HIGH)

print ("Clockwise")

# Counterclockwise

if *direction* == -1:

# Set direction

GPIO.output(MotorPin1, GPIO.LOW)

GPIO.output(MotorPin2, GPIO.HIGH)

# Enable the motor

GPIO.output(MotorEnable, GPIO.HIGH)

print ("Counterclockwise")

# Stop

if *direction* == 0:

# Disable the motor

GPIO.output(MotorEnable, GPIO.LOW)

print ("Stop")

def main():

print\_message()

# Define a dictionary to make the script more readable

# CW as clockwise, CCW as counterclockwise, STOP as stop

directions = {'CW': 1, 'CCW': -1, 'STOP': 0}

while True:

# Clockwise

motor(directions['CW'])

time.sleep(5)

# Stop

motor(directions['STOP'])

time.sleep(5)

# Anticlockwise

motor(directions['CCW'])

time.sleep(5)

# Stop

motor(directions['STOP'])

time.sleep(5)

def destroy():

# Stop the motor

GPIO.output(MotorEnable, GPIO.LOW)

# Release resource

GPIO.cleanup()

# If run this script directly, do:

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

setup()

try:

main()

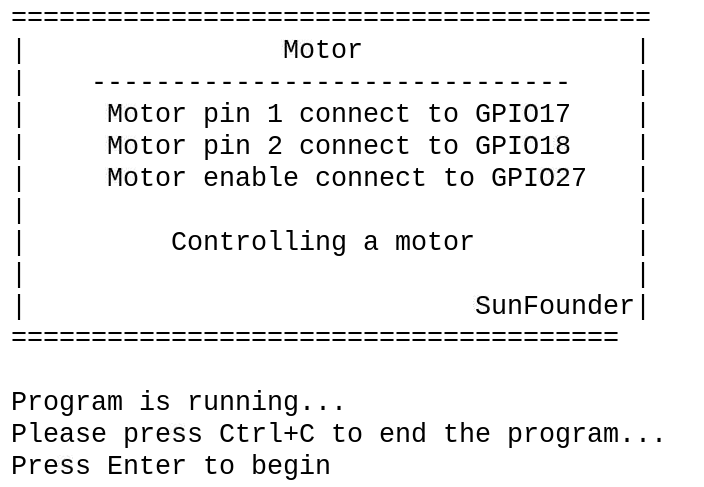
# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()

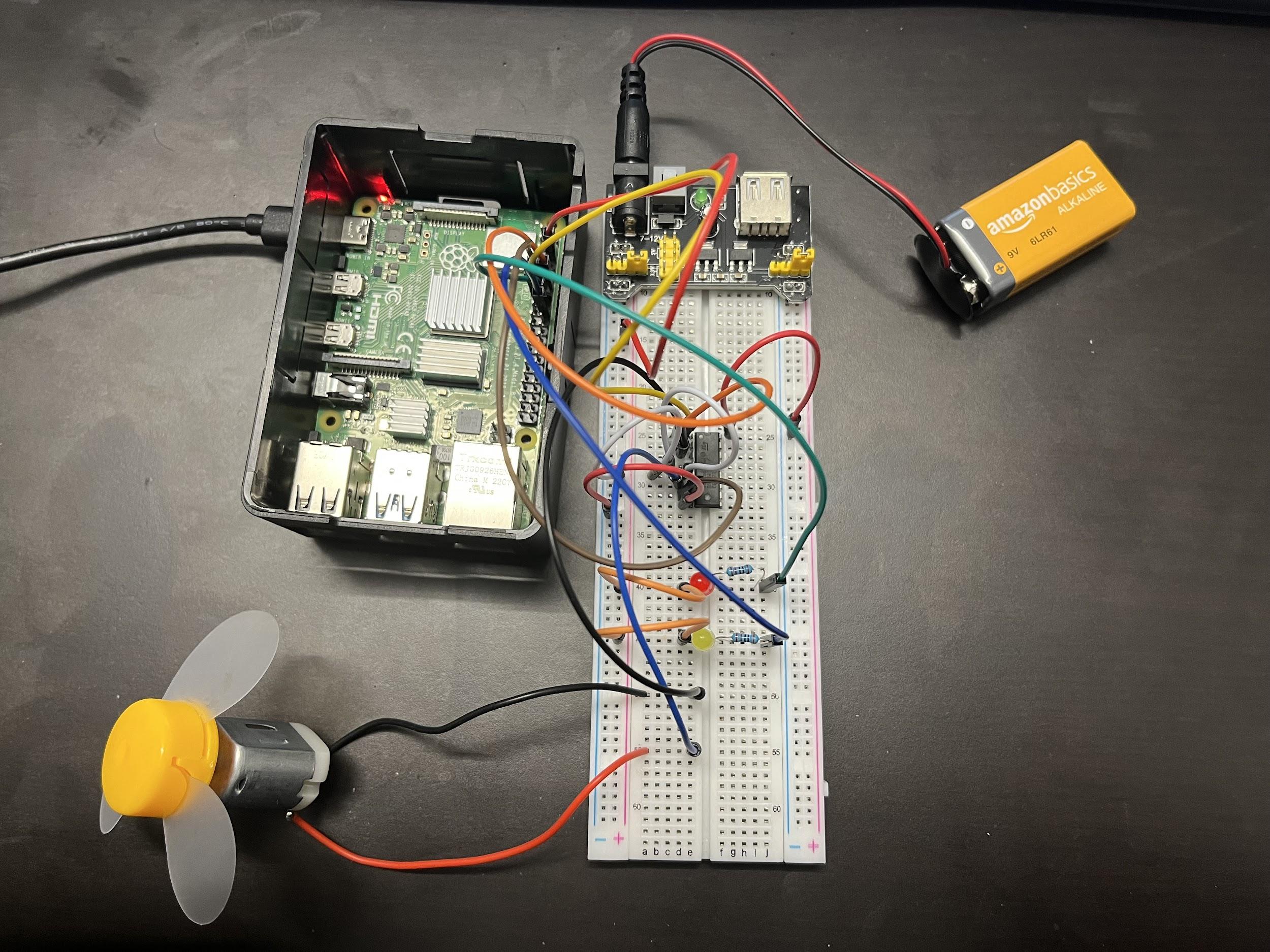
Run program & demonstration:



Video link: <https://youtu.be/o5y4QLYKuIs>



Breadboard setup:



For the 2 LEDs (Yellow and Red), first we connect GPIO 23 and 24 respectively to two resistors, then connect to the LEDs and yield them to the same GND with the DC motor.

Source code:

<https://github.com/MynameisKoi/CE450L/blob/main/Lab%237/Motor_with_LED.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 up pins

MotorPin1 = 17

MotorPin2 = 18

MotorEnable = 27

YellowLED = 24

RedLED = 23

def print\_message():

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

print ("| Motor |")

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

print ("| Motor pin 1 connect to GPIO17 |")

print ("| Motor pin 2 connect to GPIO18 |")

print ("| Motor enable connect to GPIO27 |")

print ("| YellowLED connect to GPIO24 |")

print ("| RedLED connect to GPIO23 |")

print ("| |")

print ("| Controlling a motor |")

print ("| |")

print ("| Khoi Duong|")

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

print ("Program is running...")

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

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

def setup():

# Set the GPIO modes to BCM Numbering

GPIO.setmode(GPIO.BCM)

# Set pins to output

GPIO.setup(MotorPin1, GPIO.OUT)

GPIO.setup(MotorPin2, GPIO.OUT)

GPIO.setup(MotorEnable, GPIO.OUT, *initial*=GPIO.LOW)

GPIO.setup(YellowLED, GPIO.OUT, *initial*=GPIO.LOW)

GPIO.setup(RedLED, GPIO.OUT, *initial*=GPIO.LOW)

# Define a motor function to spin the motor

# direction should be

# 1(clockwise), 0(stop), -1(counterclockwise)

def motor(*direction*):

# Clockwise

if *direction* == 1:

# Set direction

GPIO.output(MotorPin1, GPIO.HIGH)

GPIO.output(MotorPin2, GPIO.LOW)

# Enable the motor

GPIO.output(MotorEnable, GPIO.HIGH)

print ("Clockwise - Yellow")

# Counterclockwise

if *direction* == -1:

# Set direction

GPIO.output(MotorPin1, GPIO.LOW)

GPIO.output(MotorPin2, GPIO.HIGH)

# Enable the motor

GPIO.output(MotorEnable, GPIO.HIGH)

print ("Counterclockwise - Red")

# Stop

if *direction* == 0:

# Disable the motor

GPIO.output(MotorEnable, GPIO.LOW)

print ("Stop")

def main():

print\_message()

# Define a dictionary to make the script more readable

# CW as clockwise, CCW as counterclockwise, STOP as stop

directions = {'CW': 1, 'CCW': -1, 'STOP': 0}

while True:

# Clockwise

motor(directions['CW'])

GPIO.output(YellowLED, GPIO.HIGH)

GPIO.output(RedLED, GPIO.LOW)

time.sleep(5)

# Stop

motor(directions['STOP'])

GPIO.output(YellowLED, GPIO.LOW)

GPIO.output(RedLED, GPIO.LOW)

time.sleep(2)

# Anticlockwise

motor(directions['CCW'])

GPIO.output(RedLED, GPIO.HIGH)

GPIO.output(YellowLED, GPIO.LOW)

time.sleep(5)

# Stop

motor(directions['STOP'])

GPIO.output(YellowLED, GPIO.LOW)

GPIO.output(RedLED, GPIO.LOW)

time.sleep(2)

def destroy():

# Stop the motor

GPIO.output(MotorEnable, GPIO.LOW)

GPIO.output(RedLED, GPIO.LOW)

GPIO.output(YellowLED, GPIO.LOW)

# Release resource

GPIO.cleanup()

# If run this script directly, do:

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

setup()

try:

main()

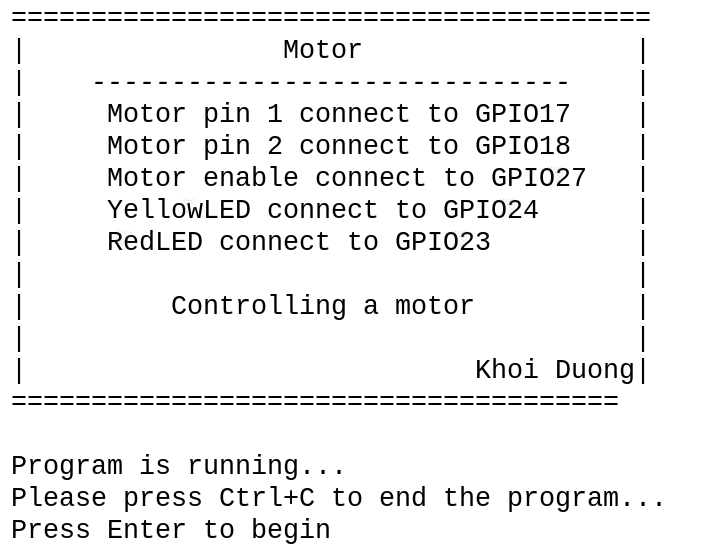
# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()

Run program & demonstration:



Video link: <https://youtu.be/PwwBYkZvklg>