Khoi Duong

Prof. Yang

CE450L

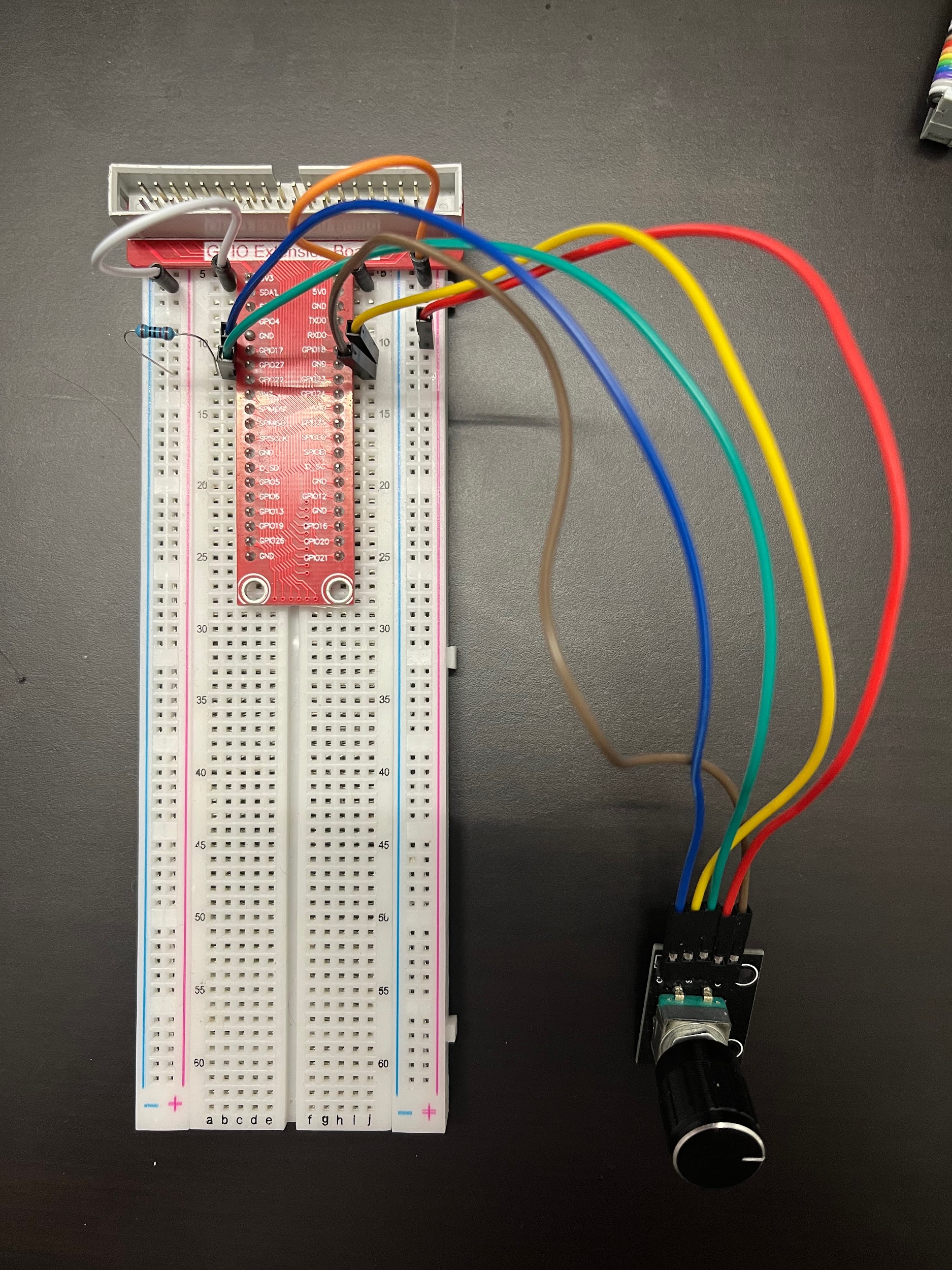
12/15/2022

LAB#8

GitHub link: <https://github.com/MynameisKoi/CE450L/tree/main/Lab%238>



Breadboard setup:



Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%238/RotaryEncoder.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

# Rotary A Pin

RoAPin = 17

# Rotary B Pin

RoBPin = 18

# Rotary Switch Pin

RoSPin = 27

def print\_message():

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

print ("| Rotary Encoder |")

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

print ("| Pin A connect to GPIO17 |")

print ("| Pin B connect to GPIO18 |")

print ("| Button Pin connect to GPIO27 |")

print ("| |")

print ("| Use a Rotary Encoder |")

print ("| Rotary to add/minus counter |")

print ("| Press to set counter to 0 |")

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():

global counter

global Last\_RoB\_Status, Current\_RoB\_Status

GPIO.setmode(GPIO.BCM)

GPIO.setup(RoAPin, GPIO.IN)

GPIO.setup(RoBPin, GPIO.IN)

GPIO.setup(RoSPin,GPIO.IN, *pull\_up\_down*=GPIO.PUD\_UP)

# Set up a falling edge detect to callback clear

GPIO.add\_event\_detect(RoSPin, GPIO.FALLING, *callback*=clear)

# Set up a counter as a global variable

counter = 0

Last\_RoB\_Status = 0

Current\_RoB\_Status = 0

# Define a function to deal with rotary encoder

def rotaryDeal():

global counter

global Last\_RoB\_Status, Current\_RoB\_Status

flag = 0

Last\_RoB\_Status = GPIO.input(RoBPin)

# When RoAPin level changes

while(not GPIO.input(RoAPin)):

Current\_RoB\_Status = GPIO.input(RoBPin)

flag = 1

if flag == 1:

# Reset flag

flag = 0

if (Last\_RoB\_Status == 0) and (Current\_RoB\_Status == 1):

counter = counter - 1

if (Last\_RoB\_Status == 1) and (Current\_RoB\_Status == 0):

counter = counter + 1

print ("counter = %d" % counter)

# Define a callback function on switch, to clean "counter"

def clear(*ev*=None):

global counter

counter = 0

def main():

print\_message()

while True:

rotaryDeal()

def destroy():

# Release resource

GPIO.cleanup()

# If run this script directly, do:

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

setup()

try:

main()

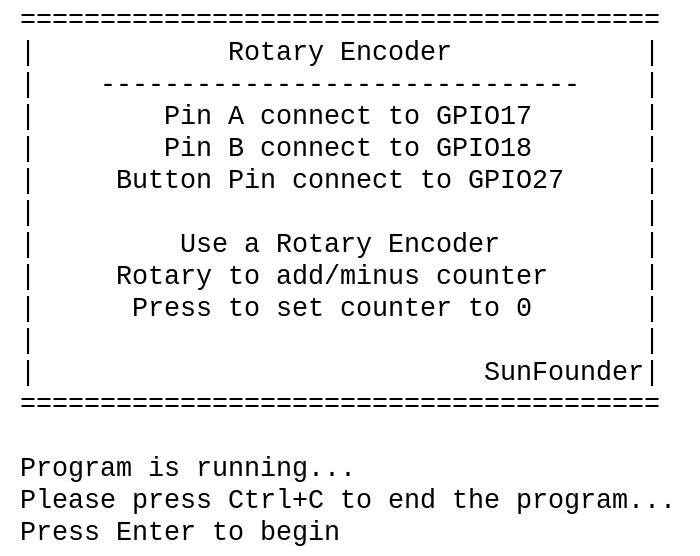
# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()

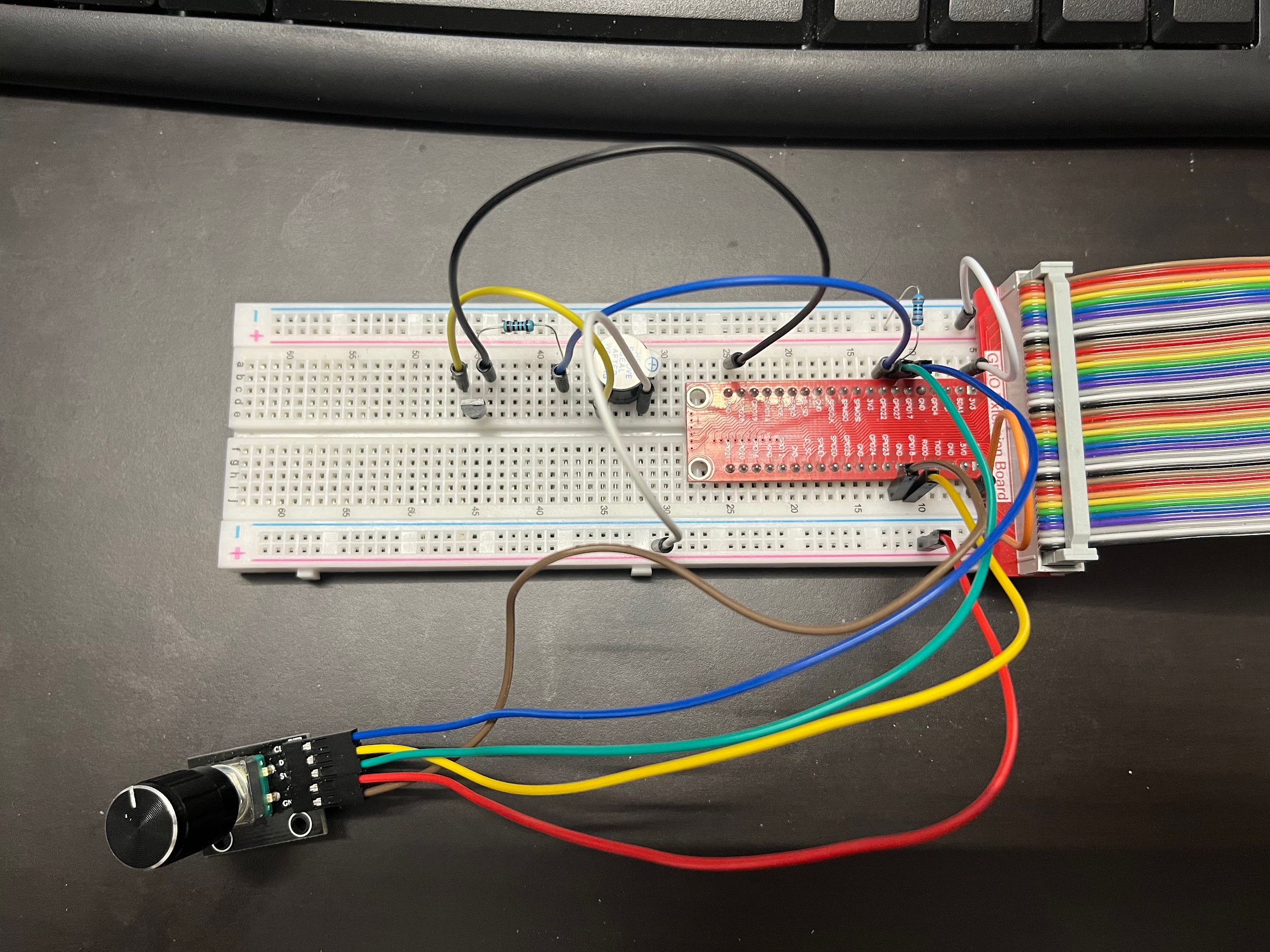
Run program & demonstration:



Video link: [Rotary Encoder - Lab#8 - CE450L](https://youtu.be/RlvQVCfVKtA)



Breadboard setup:



Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%238/Rotary_w_Buzz.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

# Rotary A Pin

RoAPin = 17

# Rotary B Pin

RoBPin = 18

# Rotary Switch Pin

RoSPin = 27

# Buzzer Pin

BeepPin = 22

def print\_message():

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

print ("| Rotary Encoder |")

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

print ("| Pin A connect to GPIO17 |")

print ("| Pin B connect to GPIO18 |")

print ("| Button Pin connect to GPIO27 |")

print ("| |")

print ("| Use a Rotary Encoder |")

print ("| Rotary to add/minus counter |")

print ("| Press to set counter to 0 |")

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():

global counter

global Last\_RoB\_Status, Current\_RoB\_Status

GPIO.setmode(GPIO.BCM)

GPIO.setup(RoAPin, GPIO.IN)

GPIO.setup(RoBPin, GPIO.IN)

GPIO.setup(RoSPin,GPIO.IN, *pull\_up\_down*=GPIO.PUD\_UP)

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

# Set up a falling edge detect to callback clear

GPIO.add\_event\_detect(RoSPin, GPIO.FALLING, *callback*=clear)

# Set up a counter as a global variable

counter = 0

Last\_RoB\_Status = 0

Current\_RoB\_Status = 0

# Define a function to deal with rotary encoder

def rotaryDeal():

global counter

global Last\_RoB\_Status, Current\_RoB\_Status

flag = 0

Last\_RoB\_Status = GPIO.input(RoBPin)

# When RoAPin level changes

while(not GPIO.input(RoAPin)):

Current\_RoB\_Status = GPIO.input(RoBPin)

flag = 1

if flag == 1:

# Reset flag

flag = 0

if (Last\_RoB\_Status == 0) and (Current\_RoB\_Status == 1):

counter = counter - 1

if (Last\_RoB\_Status == 1) and (Current\_RoB\_Status == 0):

counter = counter + 1

print ("counter = %d" % counter)

if counter % 20 == 0:

GPIO.output(BeepPin, GPIO.LOW)

time.sleep(1)

GPIO.output(BeepPin, GPIO.HIGH)

# Define a callback function on switch, to clean "counter"

def clear(*ev*=None):

global counter

counter = 0

def main():

print\_message()

while True:

rotaryDeal()

def destroy():

# Release resource

GPIO.output(BeepPin, GPIO.HIGH)

GPIO.cleanup()

# If run this script directly, do:

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

setup()

try:

main()

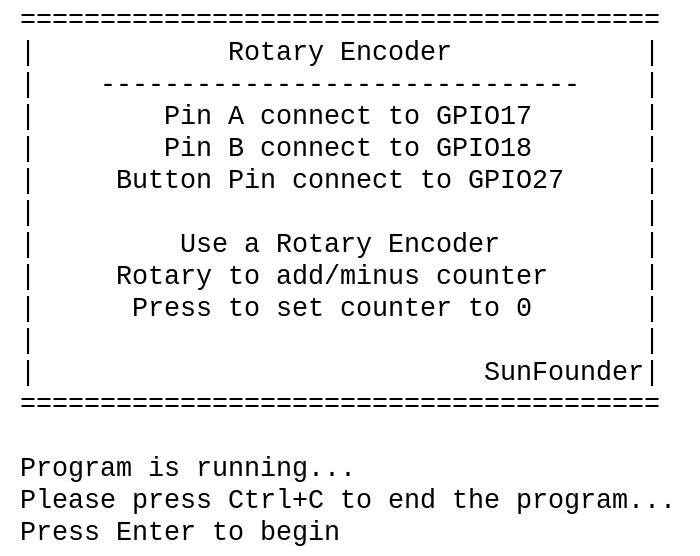
# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()

Run program & demonstration:



Video link: [Rotary Encoder with Buzzer - Lab#8 - CE450L](https://youtu.be/-5KhV0YmePI)