Khoi Duong

Prof. Yang

CE450L

10/26/2022

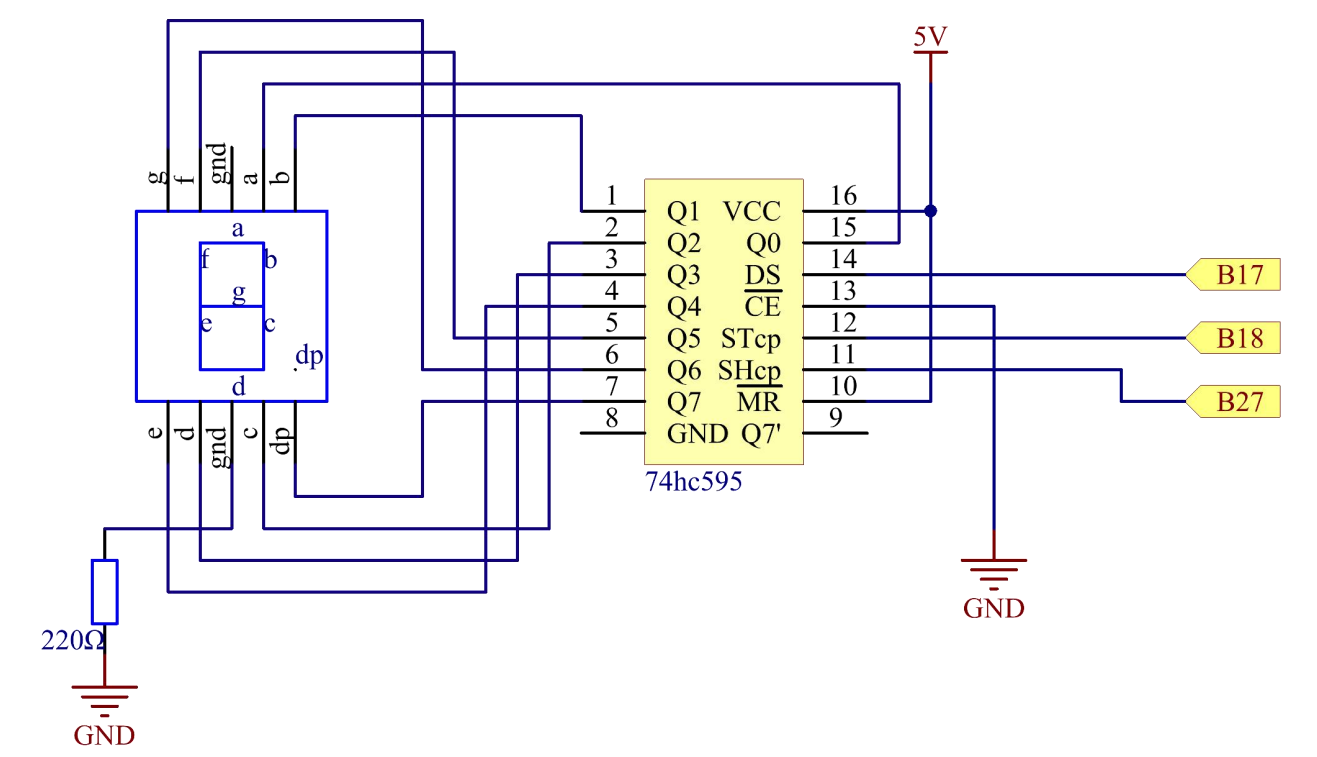
LAB#5

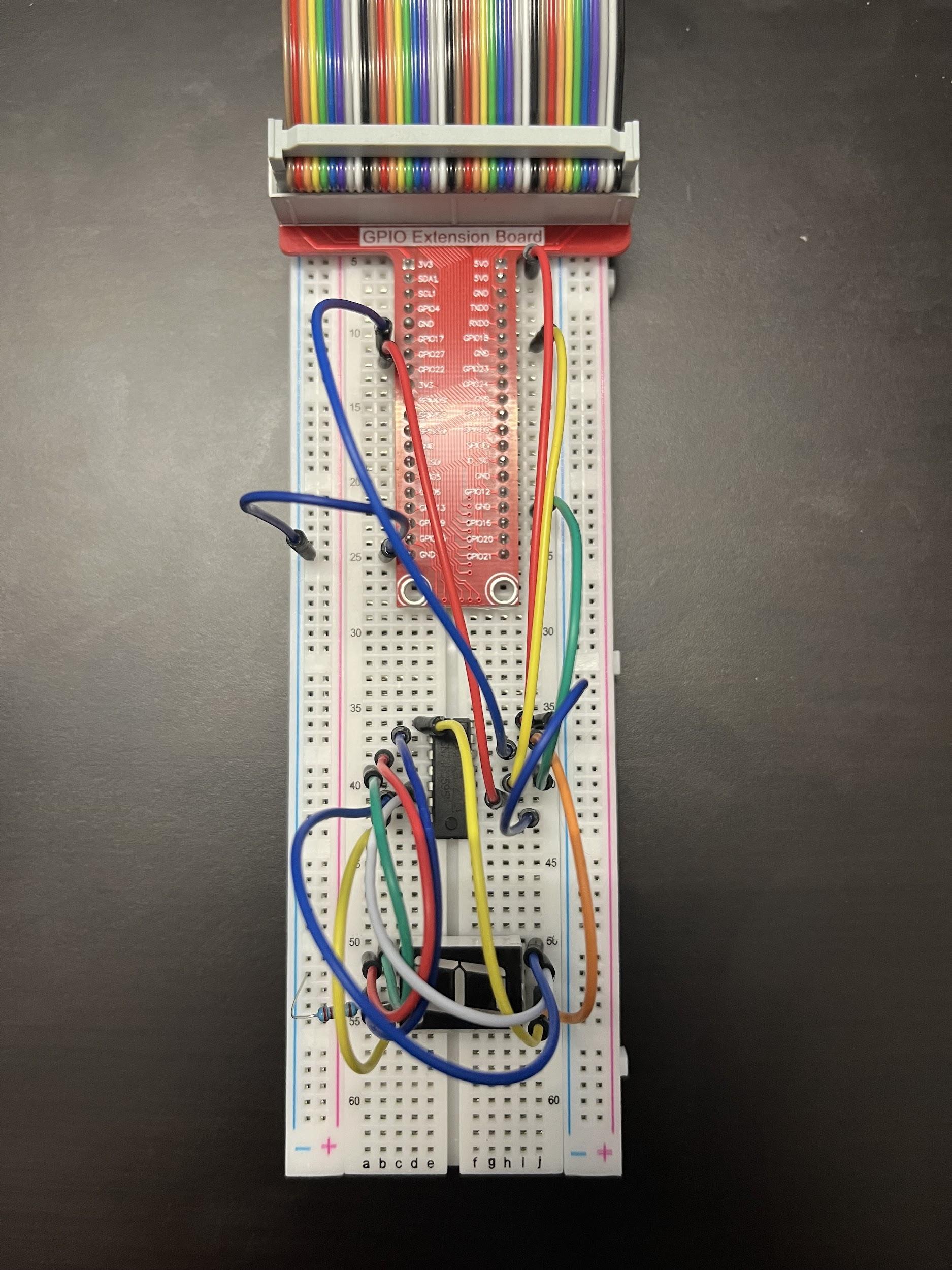
GitHub link: <https://github.com/MynameisKoi/CE450L/tree/main/Lab%235>



***7\_segment.py ⇔ Display on the 7-segment display***

Breadboard setup:





Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%235/7_segment.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

SDI = 17

RCLK = 18

SRCLK = 27

# Define a segment code from 0 to F in Hexadecimal

# Common cathode

segCode = [0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71]

# Common anode

# segCode = [0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e]

def print\_msg():

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

print ("| Segment with 74HC595 |")

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

print ("| SDI connect to GPIO17 |")

print ("| RCLK connect to GPIO18 |")

print ("| SRCLK connect to GPIO27 |")

print ("| |")

print ("| Control segment with 74HC595 |")

print ("| |")

print ("| SunFounder|")

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

print ("Program is running...")

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

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

def setup():

GPIO.setmode(GPIO.BCM)

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

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

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

# Shift the data to 74HC595

def hc595\_shift(*dat*):

for bit in range(0, 8):

GPIO.output(SDI, 0x80 & (*dat* << bit))

GPIO.output(SRCLK, GPIO.HIGH)

time.sleep(0.001)

GPIO.output(SRCLK, GPIO.LOW)

GPIO.output(RCLK, GPIO.HIGH)

time.sleep(0.001)

GPIO.output(RCLK, GPIO.LOW)

def main():

print\_msg()

while True:

# Shift the code one by one from segCode list

for code in segCode:

hc595\_shift(code)

print ("segCode[%s]: 0x%02X"%(segCode.index(code), code)) # double digit to print

time.sleep(0.5)

def destroy():

GPIO.cleanup()

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

setup()

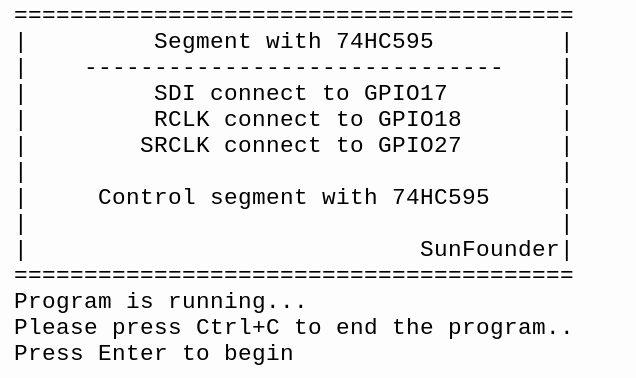
try:

main()

except KeyboardInterrupt:

destroy()

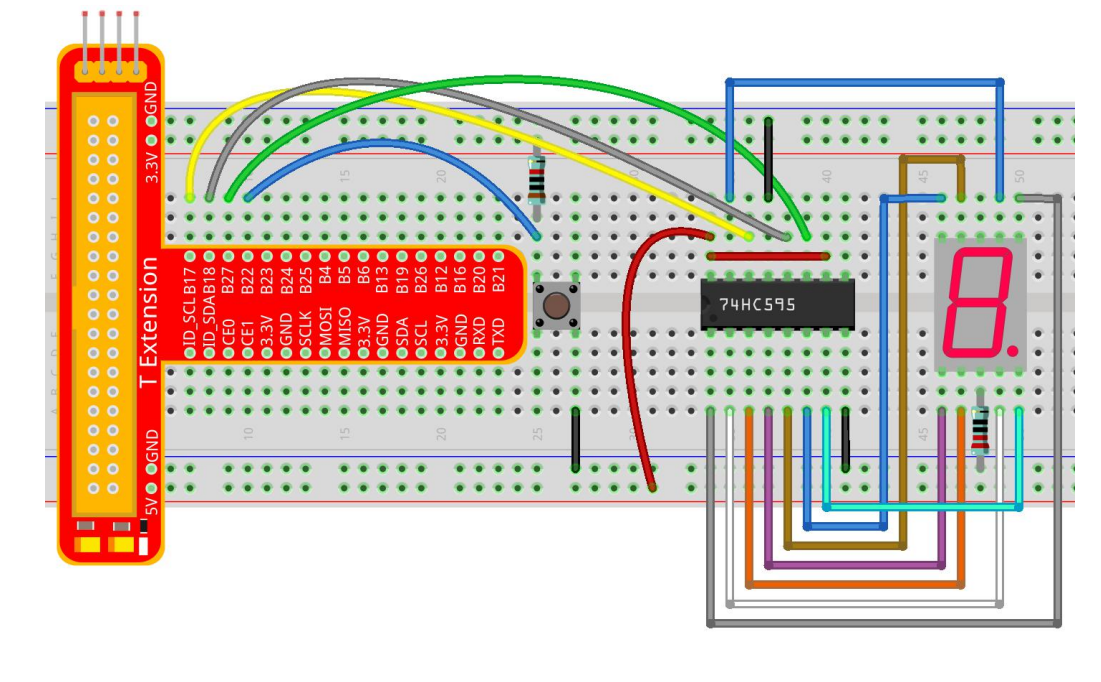
Run code & demonstration:

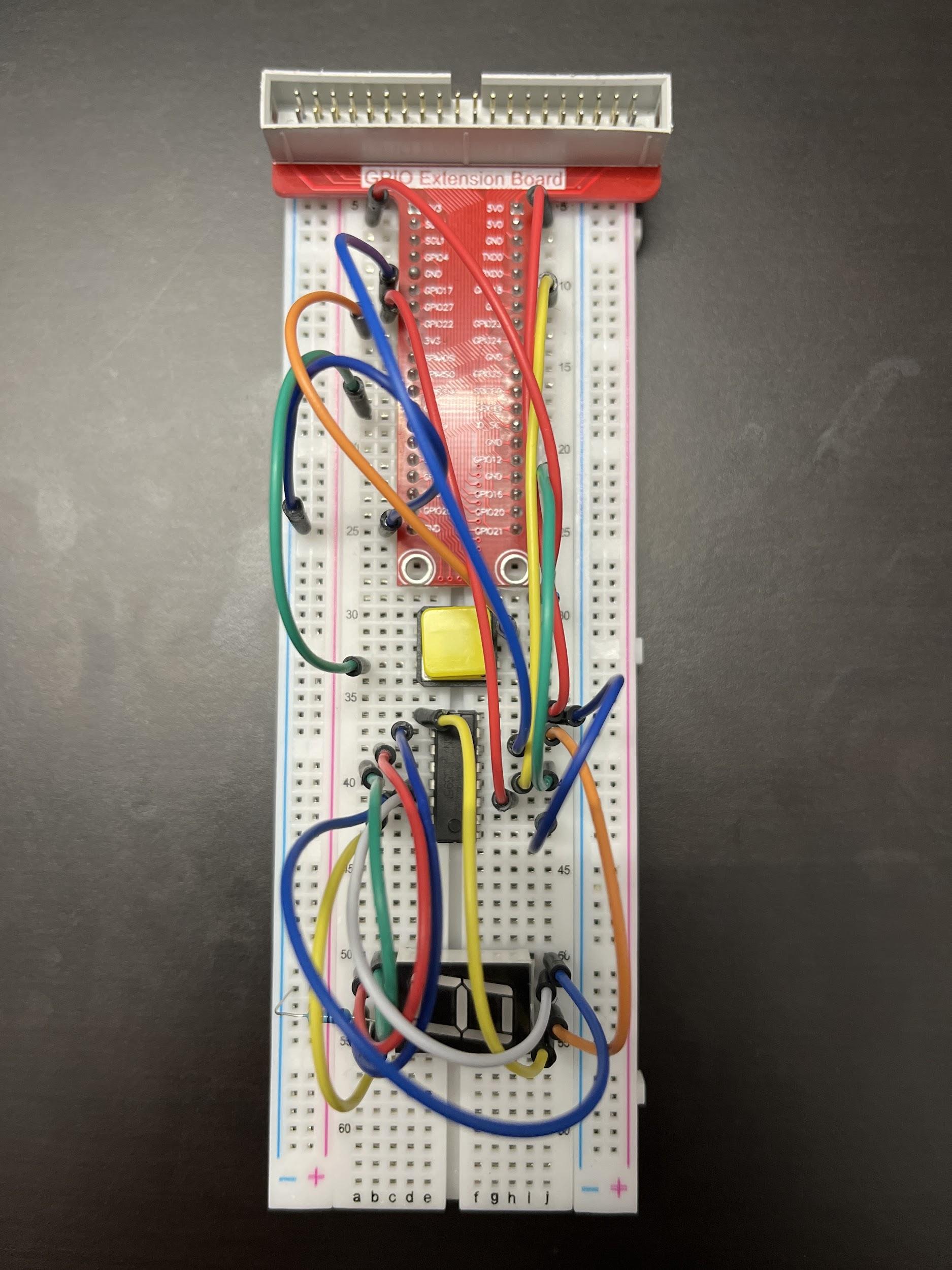


Video link: <https://youtu.be/nybU3K42A50>

***dice.py ⇔ Rolling dice with 7-segment display***

Breadboard setup:





Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%235/dice.py>

#!/usr/bin/env python3

import RPi.GPIO as GPIO

import time

import random

from sys import version\_info

if version\_info.major == 3:

raw\_input = input

# Set up pins

SDI = 17

RCLK = 18

SRCLK = 27

TouchPin = 22

# Define a segment code from 1 to 6 in Hexadecimal

SegCode = [0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d]

# Used to record button press

flag = 0

def print\_msg():

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

print ("| Dice |")

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

print ("| SDI connect to GPIO17 |")

print ("| RCLK connect to GPIO18 |")

print ("| SRCLK connect to GPIO27 |")

print ("| Button Pin connect to GPIO22 |")

print ("| |")

print ("| Control segment with 74HC595 |")

print ("| random number 1~6 |")

print ("| Press to suspend segment 2 seconds |")

print ("| |")

print ("| SunFounder|")

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

print ("Program is running...")

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

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

def setup():

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

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

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

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

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

GPIO.add\_event\_detect(TouchPin, GPIO.RISING, *callback* = randomISR, *bouncetime* = 20)

# Shift the data to 74HC595

def hc595\_shift(*dat*):

for bit in range(0, 8):

GPIO.output(SDI, 0x80 & (*dat* << bit))

GPIO.output(SRCLK, GPIO.HIGH)

time.sleep(0.001)

GPIO.output(SRCLK, GPIO.LOW)

GPIO.output(RCLK, GPIO.HIGH)

time.sleep(0.001)

GPIO.output(RCLK, GPIO.LOW)

def randomISR(*channel*):

global flag

flag = 1

def destroy():

GPIO.cleanup()

def main():

global flag

print\_msg()

while True:

num = random.randint(1,6)

hc595\_shift(SegCode[num-1])

print (num, hex(SegCode[num-1]))

if flag == 1:

print ("Num: ", num)

time.sleep(2)

flag = 0

else:

time.sleep(0.01)

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

setup()

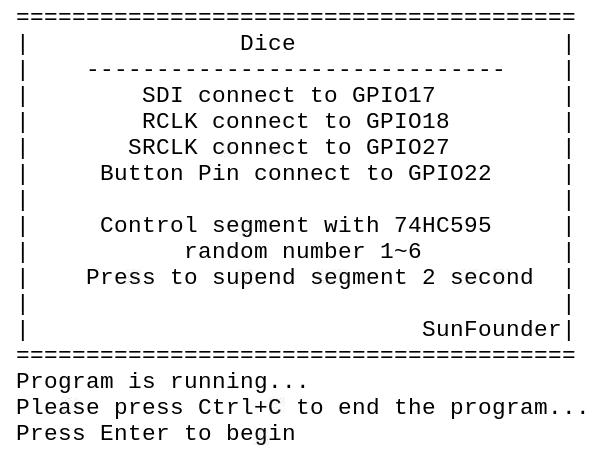
try:

main()

except KeyboardInterrupt:

destroy()

Run program & demonstration:

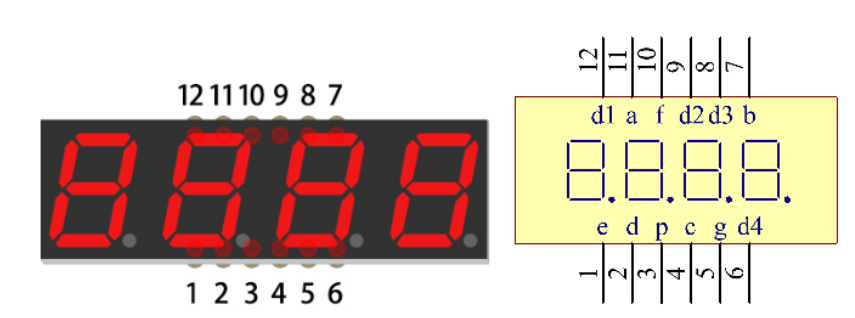


Video link: <https://youtu.be/urOZOrzTO7w>

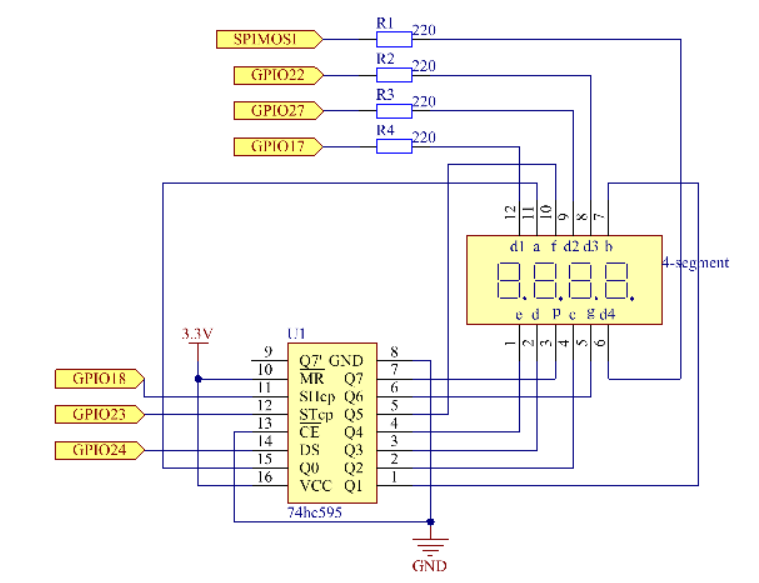


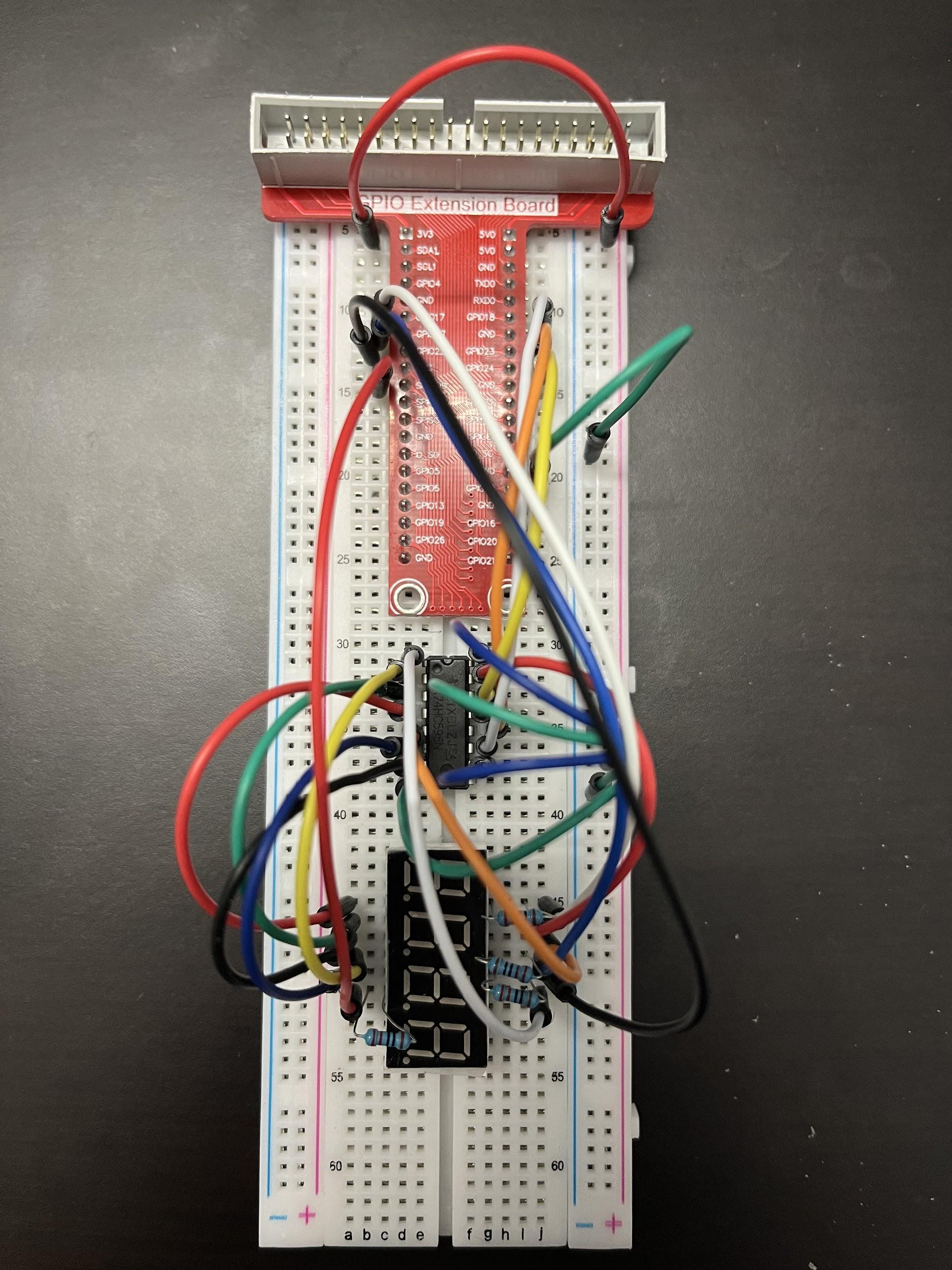
For this question, we have to use the 4-digit 7-segment display.

We have the schema for the 4-digit 7-segment display as below:



Breadboard setup:

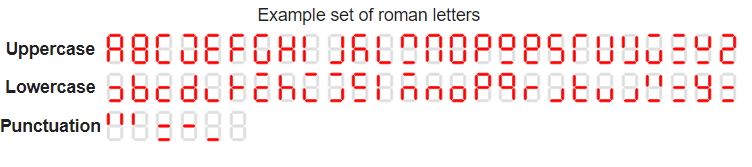




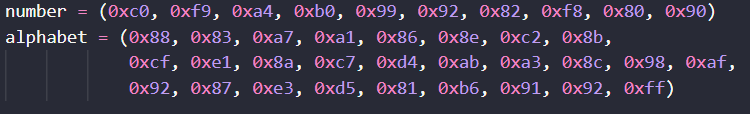
We have the code 1 as OFF and 0 as ON for the 7-segment display

The display follows the following format: (DP)GFEDCBA

We have the following display:



Thus, the hex code for numbers from 0 - 25 and a - z will be as follow:



Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%235/4_7_segments.py>

import RPi.GPIO as GPIO

import time

import threading

from sys import version\_info

if version\_info.major == 3:

raw\_input = input

SDI = 24

RCLK = 23

SRCLK = 18

placePin = (10, 22, 27, 17)

number = (0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90)

alphabet = (0x88, 0x83, 0xa7, 0xa1, 0x86, 0x8e, 0xc2, 0x8b,

0xcf, 0xe1, 0x8a, 0xc7, 0xd4, 0xab, 0xa3, 0x8c, 0x98, 0xaf,

0x92, 0x87, 0xe3, 0xd5, 0x81, 0xb6, 0x91, 0x92, 0xff)

counter = 0

timer1 = 0

def clearDisplay():

for i in range(8):

GPIO.output(SDI, 1)

GPIO.output(SRCLK, GPIO.HIGH)

GPIO.output(SRCLK, GPIO.LOW)

GPIO.output(RCLK, GPIO.HIGH)

GPIO.output(RCLK, GPIO.LOW)

def hc595\_shift(*data*):

for i in range(8):

GPIO.output(SDI, 0x80 & (*data* << i))

GPIO.output(SRCLK, GPIO.HIGH)

GPIO.output(SRCLK, GPIO.LOW)

GPIO.output(RCLK, GPIO.HIGH)

GPIO.output(RCLK, GPIO.LOW)

def print\_msg():

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

print ("| Display 1-25 and a-z |")

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

print ("| SDI connect to GPIO24 |")

print ("| RCLK connect to GPIO23 |")

print ("| SRCLK connect to GPIO18 |")

print ("| |")

print ("| Control 7-seg display with 74HC595 |")

print ("| |")

print ("| SunFounder|")

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

print ("Program is running...")

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

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

def pickDigit(*digit*):

for i in placePin:

GPIO.output(i,GPIO.LOW)

GPIO.output(placePin[*digit*], GPIO.HIGH)

def timer():

global counter

global timer1

timer1 = threading.Timer(0.5, timer)

timer1.start()

print("%d" % counter)

if counter > 51:

counter = 1

else:

counter += 1

def main():

global counter

print\_msg()

global timer1

timer1 = threading.Timer(0.5, timer)

timer1.start()

while True:

if counter < 26:

clearDisplay()

pickDigit(0)

hc595\_shift(number[counter % 10])

clearDisplay()

pickDigit(1)

hc595\_shift(number[counter % 100//10])

else:

clearDisplay()

pickDigit(0)

hc595\_shift(alphabet[counter - 26])

def setup():

GPIO.setmode(GPIO.BCM)

GPIO.setup(SDI, GPIO.OUT)

GPIO.setup(RCLK, GPIO.OUT)

GPIO.setup(SRCLK, GPIO.OUT)

for i in placePin:

GPIO.setup(i, GPIO.OUT)

def destroy(): # When "Ctrl+C" is pressed, the function is executed.

global timer1

GPIO.cleanup()

timer1.cancel() # cancel the timer

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

setup()

try:

main()

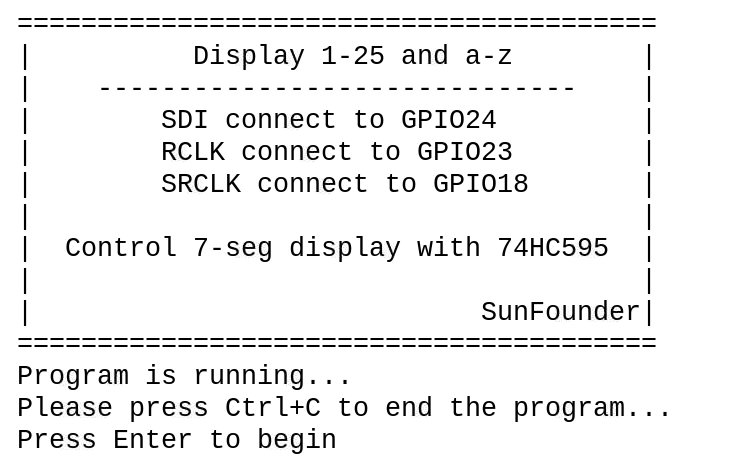
except IndexError:

destroy()

except KeyboardInterrupt:

destroy()

Run code & stimulation:



Video link: <https://youtu.be/sI9M2D-p-i4>