Khoi Duong

Prof. Yang

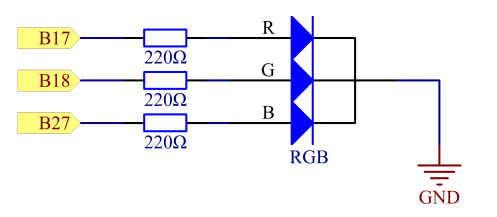
CE450L

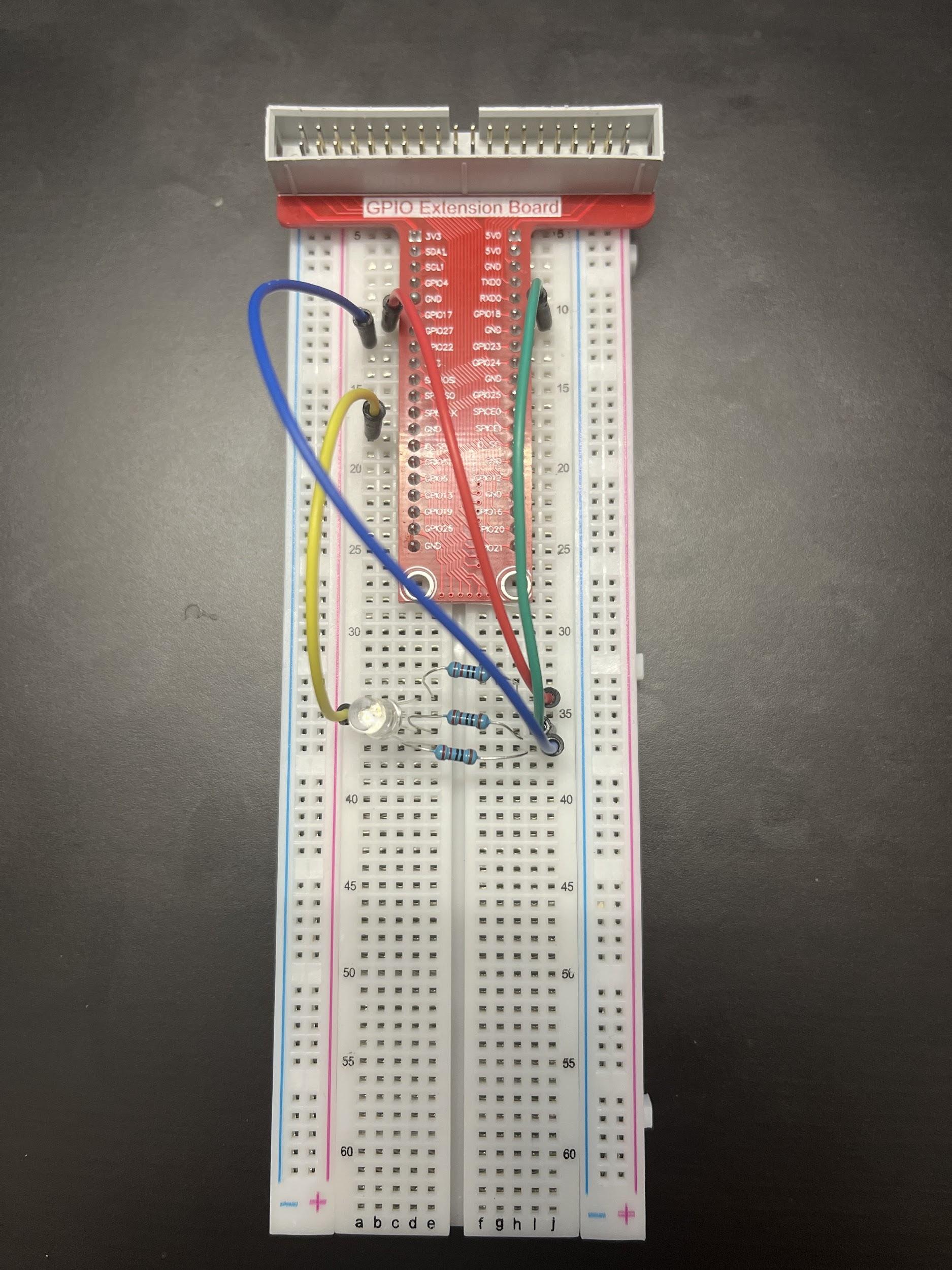
10/14/2022

LAB#3

GitHub link: <https://github.com/MynameisKoi/CE450L/tree/main/Lab%233>

Breadboard setup:





Source code: <https://github.com/MynameisKoi/CE450L/blob/main/Lab%233/MorseCode.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 a color table in Hexadecimal

# Red = 0xFF0000, Green = 0x00FF00, Blue = 0x0000FF

# Since we only use Red and Blue, we have those two colors HEX value

COLOR = [0xFF0000, 0x0000FF]

# Set pins' channels with dictionary

pins = {'Red':17, 'Green':18, 'Blue':27}

message = "- . ... - .---- ..--- ...--"

def print\_message():

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

print ("| Breath LED |")

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

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

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

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

print ("| |")

print ("| Make a RGB LED emits various color |")

print ("| based on Morse Code Message |")

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 p\_R, p\_G, p\_B

# 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)

for i in pins:

GPIO.setup(pins[i], GPIO.OUT, *initial*=GPIO.HIGH)

# Set all led as pwm channel,

# and frequece to 2KHz

p\_R = GPIO.PWM(pins['Red'], 2000)

p\_G = GPIO.PWM(pins['Green'], 2000)

p\_B = GPIO.PWM(pins['Blue'], 2000)

# Set all begin with value 0

p\_R.start(0)

p\_G.start(0)

p\_B.start(0)

# Define a MAP function for mapping values.

# Like from 0~255 to 0~100

def MAP(*x*, *in\_min*, *in\_max*, *out\_min*, *out\_max*):

return (*x* - *in\_min*) \* (*out\_max* - *out\_min*) / (*in\_max* - *in\_min*) + *out\_min*

# Define a function to set up colors

# input color should be Hexadecimal with

# red value, blue value, green value.

def setColor(*color*):

# Divide colors from 'color' variable

R\_val = (*color* & 0xFF0000) >> 16

G\_val = (*color* & 0x00FF00) >> 8

B\_val = (*color* & 0x0000FF) >> 0

# Map color value from 0~255 to 0~100

R\_val = MAP(R\_val, 0, 255, 0, 100)

G\_val = MAP(G\_val, 0, 255, 0, 100)

B\_val = MAP(B\_val, 0, 255, 0, 100)

# Change the colors

p\_R.ChangeDutyCycle(R\_val)

p\_G.ChangeDutyCycle(G\_val)

p\_B.ChangeDutyCycle(B\_val)

def blink():

setColor(0x000000)

time.sleep(0.2)

def main():

print\_message()

# Set color based on the dot, dash, and space

# Red = dot, Blue with double time of dot = dash, Blue with 4 times of dot = space

s = message.split()

for i in range(len(s)):

a = s[i]

for char in a:

if char == ".":

setColor(0xFF0000)

time.sleep(0.25)

blink()

elif char == "-":

setColor(0x0000FF)

time.sleep(0.5)

blink()

if i != (len(s)-1):

setColor(0x0000FF)

time.sleep(1)

blink()

def destroy():

# Stop all pwm channel

p\_R.stop()

p\_G.stop()

p\_B.stop()

# Turn off all LEDs

#GPIO.output(pins, GPIO.HIGH)

# Release resource

GPIO.cleanup()

# If run this script directly, do:

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

setup()

try:

main()

destroy()

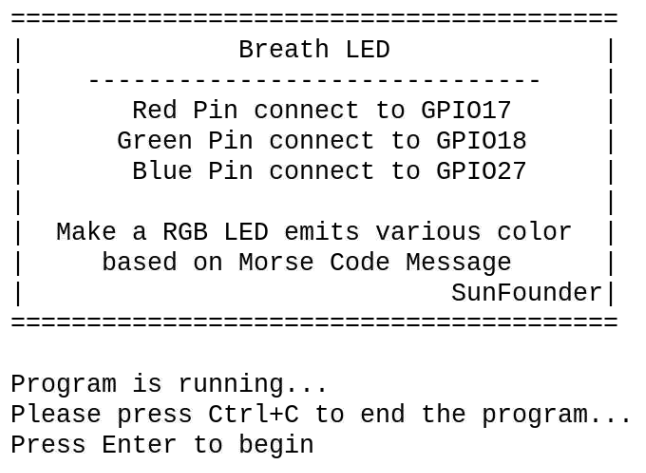
# When 'Ctrl+C' is pressed, the child program

# destroy() will be executed.

except KeyboardInterrupt:

destroy()

Run code and demonstration:



Video link:

<https://github.com/MynameisKoi/CE450L/blob/main/Lab%233/MorseCode.MOV>