



San Francisco Bay University

CE450 Fundamentals of Embedded Engineering Lab 2 LED Driving

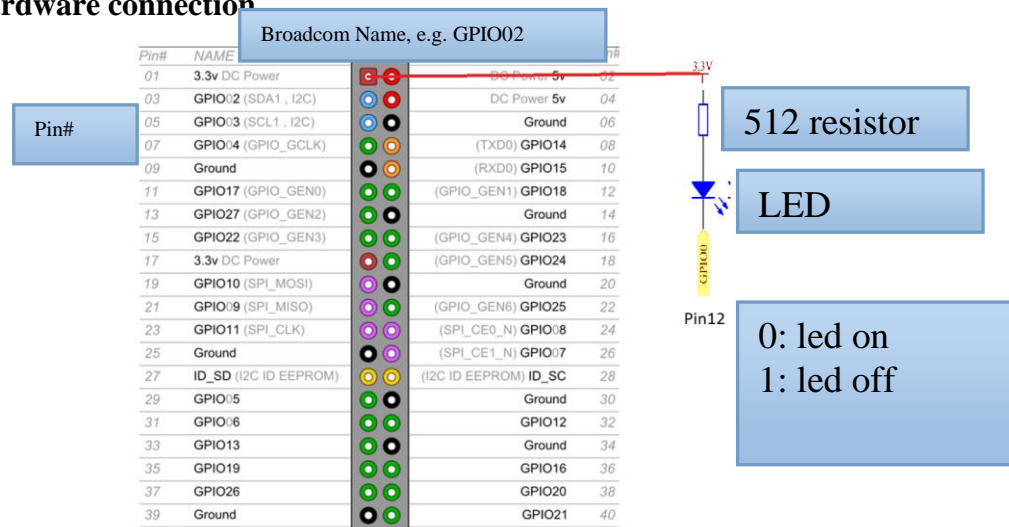
Equipment:

The equipment I am using is as follows:

- Laptop & Raspberry Pi 4B model Board
- FreeMove Starter Kit for Raspberry Pi

The Laboratory Procedure:

1. Hardware connection



2. Control program in Python

```
import RPi.GPIO as GPIO          # RPi.GPIO - func. Lib for gpio port ctrl
import time

LedPin = 12                       # pin#

def setup():                     # port configuration
    global p                      # global var. p
    GPIO.setmode(GPIO.BOARD)     # Numbers GPIOs by physical location
    GPIO.setup(LedPin, GPIO.OUT)  # Set LedPin's mode is output
    GPIO.output(LedPin, GPIO.LOW) # Set LedPin to low(0V) to switch on LED

    p = GPIO.PWM(LedPin, 1000)    # set Frequece to 1KHz
                                   # PWM: pulse width modulation

    p.start(0)                    # Duty Cycle = 0%; DC(Length of Voltage Hi)
```

```

# p.start(20)=20% #0 means half-half
# c.d = 0%, 0% 3.3v at pin 12
# c.d = 50%, 50% 3.3v at pin 12
# c.d = 20%, 20% 3.3v at pin 12

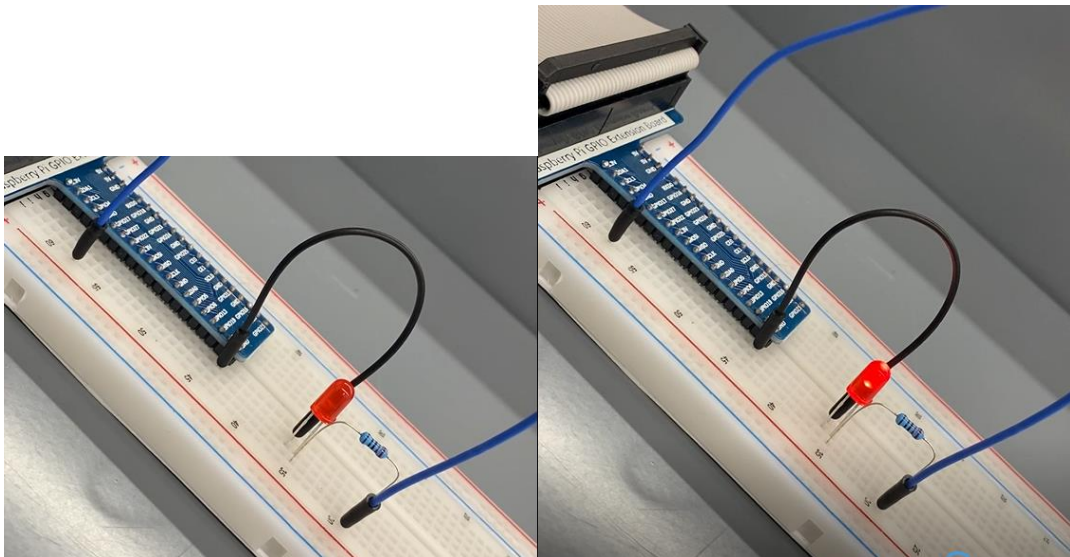
def loop():
    while True: # In GPIO ctrl, must have a forever loop
        for dc in range(0, 101, 4): # Increase duty cycle: 0~100 step=4
            p.ChangeDutyCycle(dc) # Change duty cycle
            time.sleep(0.05) # time.sleep(sec)
            # 1st iteration - dc=0:
            # 2nd iteration - dc=4,
            # 100th iteration - dc =
            # holding 0% 3.3v at pin12 for 0.05 sec, led 100% on
            # 4% 3.3v, led 96% on
            # 100, 100% 3.3v, led 100% off
            time.sleep(1)
        for dc in range(100, -1, -4): # Decrease duty cycle: 100~0
            p.ChangeDutyCycle(dc)
            time.sleep(0.05)
            time.sleep(1)

def destroy():
    p.stop()
    GPIO.output(LedPin, GPIO.HIGH) # turn off all leds
    GPIO.cleanup()

setup()
try:
    loop()
except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the child program destroy() will be executed.
    destroy()

```

Please click [this video link](#) for details.



The Laboratory Assignments:

Extend the above experiment to more complicated situations:

Please find the [video link](#) for the detailed process.

1. Design program to blink one LED and shift back and forth among 6 LEDs.

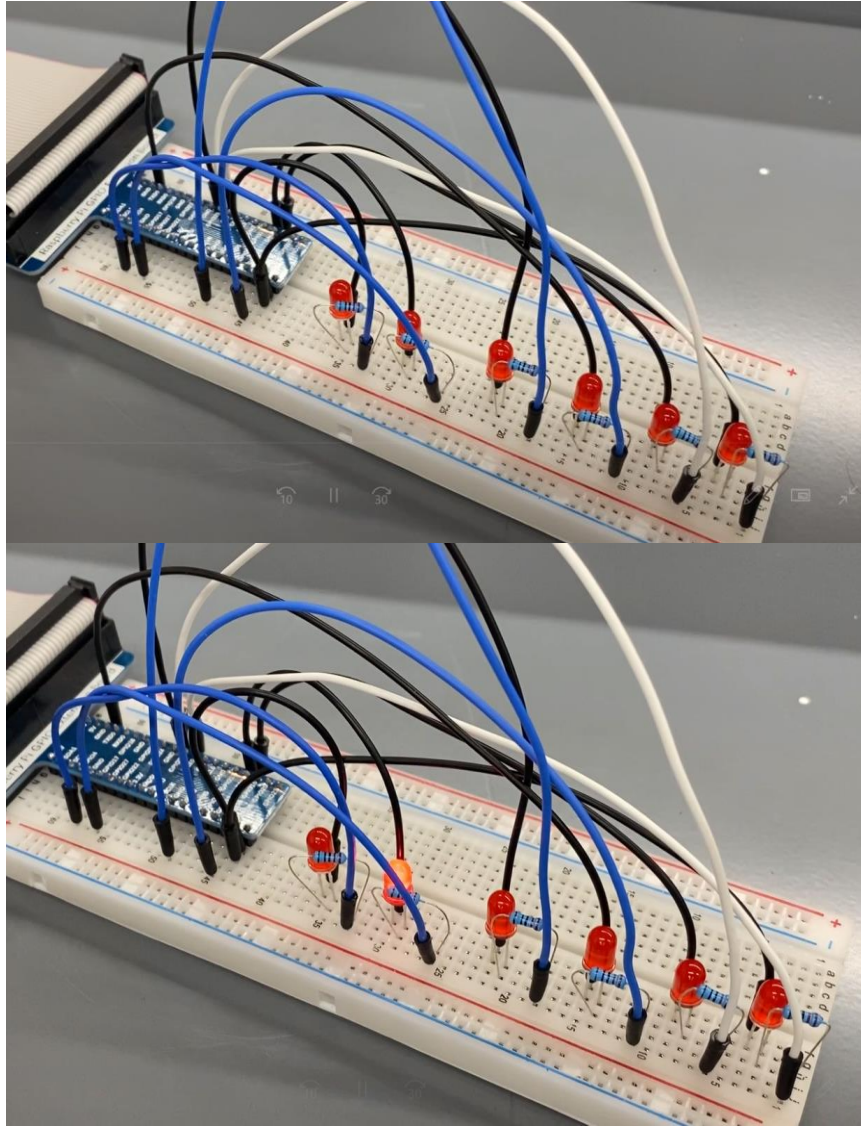
```
import RPi.GPIO as GPIO
import time

ledList = [11, 15, 29, 37, 18, 22]
#ledPin = 11 # define ledPin
def setup():
    GPIO.setmode(GPIO.BOARD) # use PHYSICAL GPIO Numbering
    for ledPin in ledList:
        GPIO.setup(ledPin, GPIO.OUT) # set the ledPin to OUTPUT mode
        GPIO.output(ledPin, GPIO.LOW) # make ledPin output LOW level
        #print ('using pin%d'%ledPin)

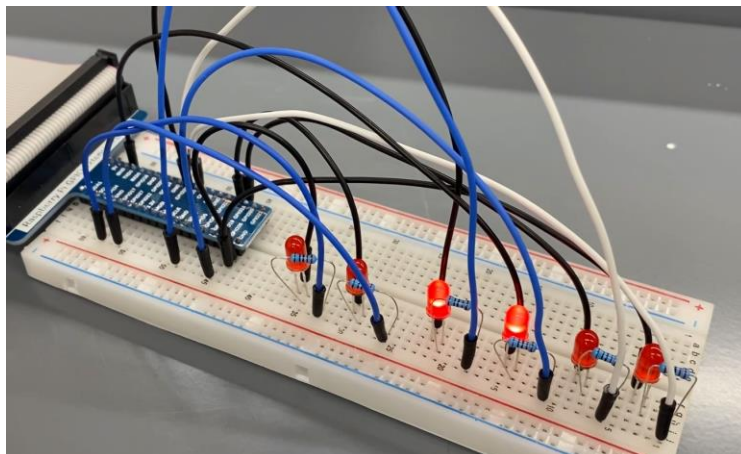
def loop():
    i = 0
    n = len(ledList)
    forth = True
    while True:
        if i == 0:
            forth = True
        if i == n - 1:
            forth = False
        ledPin = ledList[i]
        GPIO.output(ledPin, GPIO.HIGH) # make ledPin output HIGH level to turn on led
        #print ('led turned on >>>') # print information on terminal
        time.sleep(1) # Wait for 1 second
        GPIO.output(ledPin, GPIO.LOW) # make ledPin output LOW level to turn off led
        #print ('led turned off <<<')
        #time.sleep(1) # Wait for 1 second
        if forth:
            i += 1
        else:
            i -= 1

def destroy():
    GPIO.cleanup() # Release all GPIO

if __name__ == '__main__': # Program entrance
    print ('Program is starting ... \n')
    setup()
    try:
        loop()
    except KeyboardInterrupt: # Press ctrl-c to end the program.
        destroy()
```



- 2 Make 2 LEDs at the two ends of 6 LEDs on the board blink and move in different directions and back.



```

import RPi.GPIO as GPIO
import time

ledList = [11, 15, 29, 37, 18, 22]
#ledPin = 11 # define ledPin
def setup():
    GPIO.setmode(GPIO.BOARD) # use PHYSICAL GPIO Numbering
    for ledPin in ledList:
        GPIO.setup(ledPin, GPIO.OUT) # set the ledPin to OUTPUT mode
        GPIO.output(ledPin, GPIO.LOW) # make ledPin output LOW level
        #print ('using pin%d'%ledPin)

def loop():
    n = len(ledList)
    left,right = 0, n -1
    forth = True

    while True:
        if left == 0:
            forth = True
        if left == n - 1:
            forth = False

        ledPin0 = ledList[left]
        ledPin1 = ledList[right]
        GPIO.output(ledPin0, GPIO.HIGH) # make ledPin output HIGH level to turn on led
        GPIO.output(ledPin1, GPIO.HIGH) # make ledPin output HIGH level to turn on led
        time.sleep(1) # Wait for 1 second
        GPIO.output(ledPin0, GPIO.LOW) # make ledPin output LOW level to turn off led
        GPIO.output(ledPin1, GPIO.LOW) # make ledPin output LOW level to turn off led
        time.sleep(1) # Wait for 1 second

        if forth:
            left += 1
            right -= 1
        else:
            left -= 1
            right += 1

def destroy():
    GPIO.cleanup() # Release all GPIO

if __name__ == '__main__': # Program entrance
    print ('Program is starting ... \n')
    setup()
    try:
        loop()
    except KeyboardInterrupt: # Press ctrl-c to end the program.
        destroy()

```

3 Make 2 LEDs in the center blink in different directions and move back.

```
import RPi.GPIO as GPIO
import time

ledList = [11, 15, 29, 37, 18, 22]
#ledPin = 11 # define ledPin

def setup():
    GPIO.setmode(GPIO.BOARD) # use PHYSICAL GPIO Numbering
    for ledPin in ledList:
        GPIO.setup(ledPin, GPIO.OUT) # set the ledPin to OUTPUT mode
        GPIO.output(ledPin, GPIO.LOW) # make ledPin output LOW level
        #print ('using pin%d'%ledPin)

def loop():
    n = len(ledList)
    left,right = 0, n -1
    forth = True

    while True:
        if left == 0:
            forth = True
        if left == n/2 - 1:
            forth = False

        ledPin0 = ledList[left]
        ledPin1 = ledList[right]
        GPIO.output(ledPin0, GPIO.HIGH) # make ledPin output HIGH level to turn on led
        GPIO.output(ledPin1, GPIO.HIGH) # make ledPin output HIGH level to turn on led
        time.sleep(1) # Wait for 1 second
        GPIO.output(ledPin0, GPIO.LOW) # make ledPin output LOW level to turn off led
        GPIO.output(ledPin1, GPIO.LOW) # make ledPin output LOW level to turn off led
        time.sleep(1) # Wait for 1 second

        if forth:
            left += 1
            right -= 1
        else:
            left -= 1
            right += 1

def destroy():
    GPIO.cleanup() # Release all GPIO

if __name__ == '__main__': # Program entrance
    print ('Program is starting ... \n')
    setup()
    try:
        loop()
    except KeyboardInterrupt: # Press ctrl-c to end the program.
        destroy()
```

