|  |
| --- |
| 基本資料 |
| * 姓名：王豪逸 * 標題：B024020014\_Assignment 3(基本+加分) |
| 作業目的與設計 |
| * 目標：模擬紅綠燈顯示倒數時間 * 電路設計說明(總分1分)：   1. 1燈組的GPIO接法為：   LED： 綠(2)、黃(3)、紅(4)   * 1. 2燈組的GPIO接法為：   LED： 綠(22)、黃(27)、紅(17)  C 七段顯示器的接法:  由左上開始順時針依序為26,19,13,6,12,16,20,21  D 蜂鳴器(18)(基本+加分)  E 音樂的LED(加分題):  由於小星星只有六個音所以只放六個，由低音到高音依序為14,2,3,4,17,27。   電路設計照片(總分1分)：       * 程式邏輯設計(總分1分)：   基本題:  大體上也是改進上一份作業，這次的修改點在閃完黃燈後要加速鳴叫。我採用的方法為基本一聲時間為0.8秒，下次每聲的長度都為前一次的0.85倍已達到節奏變快的效果。每次都會累加這些時間，超過五秒則繼續後面的程式。  加分提:  一般的音樂中間是都有間隔的，所以在每次的節奏中我都會等待一小段時間在彈下個節奏。小星星在每7個音符為2拍這時鳴叫的時間有特別拉長。LED的部份是跟的蜂鳴器運作蜂鳴器鳴叫時會順便一起打開屬於他的LED燈。   * 影片URL：<https://youtu.be/7_TxYDBwD7w>(基本)   <https://youtu.be/22RDfmp8PoY>(加分) |
| 心得(總分1分) |
| * 遇到哪些困難：一開始還不知道怎麼讓蜂鳴器暫時停止聲音。   很難處理如何加快節奏，時間也要控制在五秒鐘。   * 如何解決：   看了一助教給的程式碼知道PWM初始的440是頻率，後面有關PWM還有一個是Start(50)猜想這應該是可以測試的部分，根據呼吸燈得程式碼把它弄成0蜂鳴器就真的沒聲音了。  時間長度要越來越短就會想到等差數列，但是用減法可能會產生負數導致程式出錯，為了避免產生負數就想到類似等差數列的等比級數。時間的部分我想到以前使用UNITY的經驗他間隔都是必須加總每次呼叫他的間隔所得到，就把這方法套用在這上面。 |
| 程式碼 |
| 程式碼：  ====基本題===(總分1分)  import RPi.GPIO as GPIO  import time  l = {"g": 2, "y": 3, "r": 4}  r = {"g": 22, "y": 27, "r": 17}  gpios = [26,19,13,6,12,16,20,21]  buzzGP = 18  numboard = [[0,1,1,1,0,1,1,1],  [0,0,0,1,0,1,0,0],  [1,0,1,1,0,0,1,1],  [1,0,1,1,0,1,1,0],  [1,1,0,1,0,1,0,0],  [1,1,1,0,0,1,1,0],  [1,1,1,0,0,1,1,1],  [0,0,1,1,0,1,0,0],  [1,1,1,1,0,1,1,1],  [1,1,1,1,0,1,0,0],  [0,0,0,0,0,0,0,0]]  def main():  GPIO.setmode(GPIO.BCM)    for key, value in l.items():  GPIO.setup(value, GPIO.OUT)  GPIO.output(value, False)  for key, value in r.items():  GPIO.setup(value, GPIO.OUT)  GPIO.output(value, False)  for i in gpios:  GPIO.setup(i, GPIO.OUT)  setNumber(10)  GPIO.setup(buzzGP, GPIO.OUT)  buzz = GPIO.PWM(buzzGP, 440)  buzz.start(0)  try:  gside = r  rside = l  GPIO.output(rside["r"], True)  time.sleep(5)  GPIO.output(gside["g"], True)  while True:  time.sleep(10)  GPIO.output(gside["g"], False)  for i in range(5,0,-1):  setNumber(i)  GPIO.output(gside["y"], True)  time.sleep(0.5)  GPIO.output(gside["y"], False)  time.sleep(0.5)  setNumber(10)  GPIO.output(gside["r"], True)  buzz.ChangeFrequency(1000)    intertime = 0.4  totaltime = 0  while totaltime < 5:  buzz.ChangeDutyCycle(50)  time.sleep(intertime\*0.7)  buzz.ChangeDutyCycle(0)  time.sleep(intertime\*1.3)  totaltime += 2\*intertime  intertime \*= 0.85    GPIO.output(rside["r"], False)  GPIO.output(rside["g"], True)  tmp = gside  gside = rside  rside = tmp  except KeyboardInterrupt:  GPIO.cleanup()  def setNumber(num):  for i in range(0,len(gpios)):  GPIO.output(gpios[i], numboard[num][i])  main()  ====加分題===  import RPi.GPIO as GPIO  import time  tone = [[0,0,0,0,0],  [66,131,262,523,1046],  [74,147,294,587,1175],  [83,165,330,659,1318],  [88,175,349,698,1397],  [98,196,392,784,1568],  [110,220,440,880,1760],  [124,247,494,988,1976]]  playtone = [tone[0][2],tone[1][2],tone[2][2],tone[3][2],  tone[4][2],tone[5][2],tone[6][2]]  led = [14,2,3,4,17,27,22]  chord = [1,1,5,5,6,6,5,4,4,3,3,2,2,1,  5,5,4,4,3,3,2,5,5,4,4,3,3,2,  1,1,5,5,6,6,5,4,4,3,3,2,2,1]  def main():  GPIO.setmode(GPIO.BCM)  for i in led:  GPIO.setup(i, GPIO.OUT)  GPIO.output(i, False)  GP18 = 18  GPIO.setup(GP18, GPIO.OUT)  buzz = GPIO.PWM(GP18, 440)  buzz.start(50)  try:  for i in range(0, len(chord)):  c = chord[i]  buzz.ChangeFrequency(playtone[c])  GPIO.output(led[c], True)  if i % 7 == 6:  time.sleep(1)  else:  time.sleep(0.5)  buzz.ChangeDutyCycle(0)  GPIO.output(led[c], False)  time.sleep(0.2)  buzz.ChangeDutyCycle(50)  except KeyboardInterrupt:  pass  GPIO.cleanup()  main() |