|  |
| --- |
| 基本資料 |
| * 姓名：王豪逸 * 標題：B024020014\_Assignment 5(基本) |
| 作業目的與設計 |
| * 目標：偵測環境舒適度 * 電路設計說明(總分1分)：   LED燈GPIO：  Red: (16), Yellow: (20), Green: (21)  溫溼度感測器GPIO(2):  蜂鳴器(18)  溫溼度感測器、蜂鳴器共用3V的接口。全部線路都連到同一個接 地。   電路設計照片(總分1分)：   * 程式邏輯設計(總分1分)：   主要流程會先偵測溫度，改變LED顏色決定是否撥放警告聲。沿用垃圾車撥放小星星的物件，由於他是thread寫成的所以運作過程中可以分成兩大部分偵測溫度的thread不會被撥放音樂的程式碼干擾到。   * 影片URL：https://youtu.be/jzvfgK853mo |
| 心得(總分1分) |
| * 遇到哪些困難：  1. 一度想不到要去哪找2個3V接口。 2. 蜂鳴器一直沒有聲音  * 如何解決：  1. 在同學得提點下想到在麵包版上插在同一排即可達到並聯可以供電。 2. 拿就有可以使用的程式碼下去測試發現是線路沒有插好，蜂鳴器跟溫溼度感測器差得太近差錯，重新整理了一下線路讓3V跟接地的位置特別一點比較不會差錯。 |
| 程式碼 |
| 程式碼：  ====基本題===(總分1分)  import Adafruit\_DHT  import time  import threading  import RPi.GPIO as GPIO  def main():  GPIO.setmode(GPIO.BCM)  sensor = Adafruit\_DHT.DHT11  tempGPIO = 2  leds = {'r':16, 'y': 20, 'g': 21}  for key, value in leds.items():  GPIO.setup(value, GPIO.OUT)  GPIO.output(value, False)  music = LittltStar()  music.start()  music.mute()  try:  while True:  ctime = time.strftime("%H:%M:%S")  humi, temp = Adafruit\_DHT.read\_retry(sensor, tempGPIO)  if humi is not None and temp is not None:  print(ctime, "-> Temp={0:0.1f}\*C Humi={1:0.1f}%".format(temp, humi))  for key, value in leds.items():  GPIO.output(value, False)  if temp >= 31:  GPIO.output(leds['r'], True)  music.play()  elif temp >=27:  GPIO.output(leds['y'], True)  music.mute()  else:  GPIO.output(leds['g'], True)  music.mute()  else:  print("Fail")  time.sleep(2)  pass  except KeyboardInterrupt:  pass  music.stop()  GPIO.cleanup()  class LittltStar (threading.Thread):  def \_\_init\_\_(self):  threading.Thread.\_\_init\_\_(self)  GPIO.setmode(GPIO.BCM)  self.tone = [0, 262, 294, 330, 349, 392, 440]  self.chord = [5,1,5,1,5,1,5,1]  self.buzzGPIO = 18  GPIO.setup(self.buzzGPIO, GPIO.OUT)  self.vol = 60  self.buzz = GPIO.PWM(self.buzzGPIO, 440)  self.buzz.start(self.vol)    self.playMusic = False  self.running = False    def run(self):  temple = 0  self.running = True  while self.running :    if(self.playMusic):  self.buzz.ChangeFrequency(self.tone[self.chord[temple]])  self.buzz.ChangeDutyCycle(self.vol)  time.sleep(0.2)  temple = (temple+1)%len(self.chord)  else:  time.sleep(0.4)  self.buzz.ChangeDutyCycle(0)  time.sleep(0.1)    def play(self):  self.playMusic = True  def mute(self):  self.playMusic = False  def stop(self):  self.running = False  main() |