|  |
| --- |
| 基本資料 |
| * 姓名：王豪逸 * 標題：B024020014\_Assignment 10(基本) |
| 作業目的與設計 |
| * 目標：電腦視覺辨視+語音控制LED燈開關 * 電路設計說明(總分1分)：   LED燈GPIO: 紅21  視訊鏡頭+麥克風  喇叭   電路設計照片(總分1分)：  C:\Users\ball\AppData\Local\Microsoft\Windows\INetCache\Content.Word\HW_10.jpg   * 程式邏輯設計(總分1分)：   偵測是否同時包含圓形、三角形，其中圓形不屬於多邊形要跟三角形分開處理，如果有則進入語音辨識的環節。偵測到打開則會開啟LED燈、偵測到不這個字眼則會關閉LED燈。   * 影片URL：https://youtu.be/uNudDTGFL\_Y |
| 心得(總分1分) |
| PI這個板子不知道是哪個環節出錯運算速度都不快，導致攝影機的畫面會延遲很久或是久久卡上一秒的畫面，反應速度很不優良。應該可以嘗試更換記憶卡試試。  語音辨識一開始只偵測是跟不。句子有點短反應不良加長就好些了。 |
| 程式碼 |
| 程式碼：  ====基本題===(總分1分)  # -\*- coding: utf-8 -\*-  import cv2  import speech\_recognition as sr  import RPi.GPIO as GPIO  from gtts import gTTS  import os  filename = "tmp.mp3"  led = 21  def main():  GPIO.setmode(GPIO.BCM)  GPIO.setup(led, GPIO.OUT)  GPIO.output(led, False)  r = sr.Recognizer()  r.energy\_threshold = 4000  w = WedCam()  while True:  if w.get\_frame():  Say("是否開啟紅色LED燈")  with sr.Microphone() as source:  print("Start.....")  audio = r.listen(source)  try:  stt = r.recognize\_google(audio, language="zh-tw").encode('utf-8')  print("Input: " + stt)  if stt.find("不") >= 0:  GPIO.output(led, False)  if stt.find("打開") >= 0:  GPIO.output(led, True)  except sr.UnknownValueError:  print("Google Speech Recognition could not understand audio")  except sr.RequestError as e:  print("Could not request results from Google Speech Recognition service; {0}".format(e))  if cv2.waitKey(1) == 27:  break  cv2.destroyAllWindows()  def Say(str):  aud = gTTS(text=str, lang="zh-tw")  aud.save(filename)  os.system("mpg321 "+filename)  class WedCam():  def \_\_init\_\_(self):  self.cap = cv2.VideoCapture(0)  def get\_frame(self):  tri = False  circle = False  sucess = False  sucess, frame = self.cap.read()  if not sucess:  return False  gray = cv2.cvtColor(frame,cv2.COLOR\_BGR2GRAY)  ret, binary = cv2.threshold(gray,127,255,cv2.THRESH\_BINARY)  contours, hierarchy = cv2.findContours(binary, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)  circles = cv2.HoughCircles(gray, cv2.cv.CV\_HOUGH\_GRADIENT, 1, 20)  for c in contours:  peri = cv2.arcLength(c, True)  if peri < 50:  continue  approx = cv2.approxPolyDP(c,0.1\*peri,True)  ct = len(approx)  if ct == 3:  cv2.drawContours(frame,[approx],-1,(0,0,255),3)  tri = True  if circles is not None:  for i in circles[0,:]:  cv2.circle(frame,(i[0],i[1]),i[2],(0,255,0),2)  circle = True  cv2.imshow("img", frame)  return tri and circle  if \_\_name\_\_ == '\_\_main\_\_':  main() |