




居家照顧_跌倒偵測

08160191 李旻珊

08160130 林采葳



目錄

- 
1. 研究目的
 2. 研究工具
 3. 流程圖
 4. 程式碼講解
 5. 問題與未來
 6. 現場展示

研究目的

台灣近年來人口老化速度越來越快，居家照顧相關的行業也越來越興盛，但是很多時候一個護理師通常會照顧3個以上的長者，因此有供不應求的情況。

我們希望利用攝影機，偵測是否有人，若有人的話則會發出警示聲音，並緩慢亮起紅燈；若沒人的話則不會有反應。更重要的是如果有人跌倒，則會發出警示聲，並快速閃爍紅燈。





研究工具

研究工具

語音卡、擴充版、喇叭



樹莓派



麵包板

攝影機



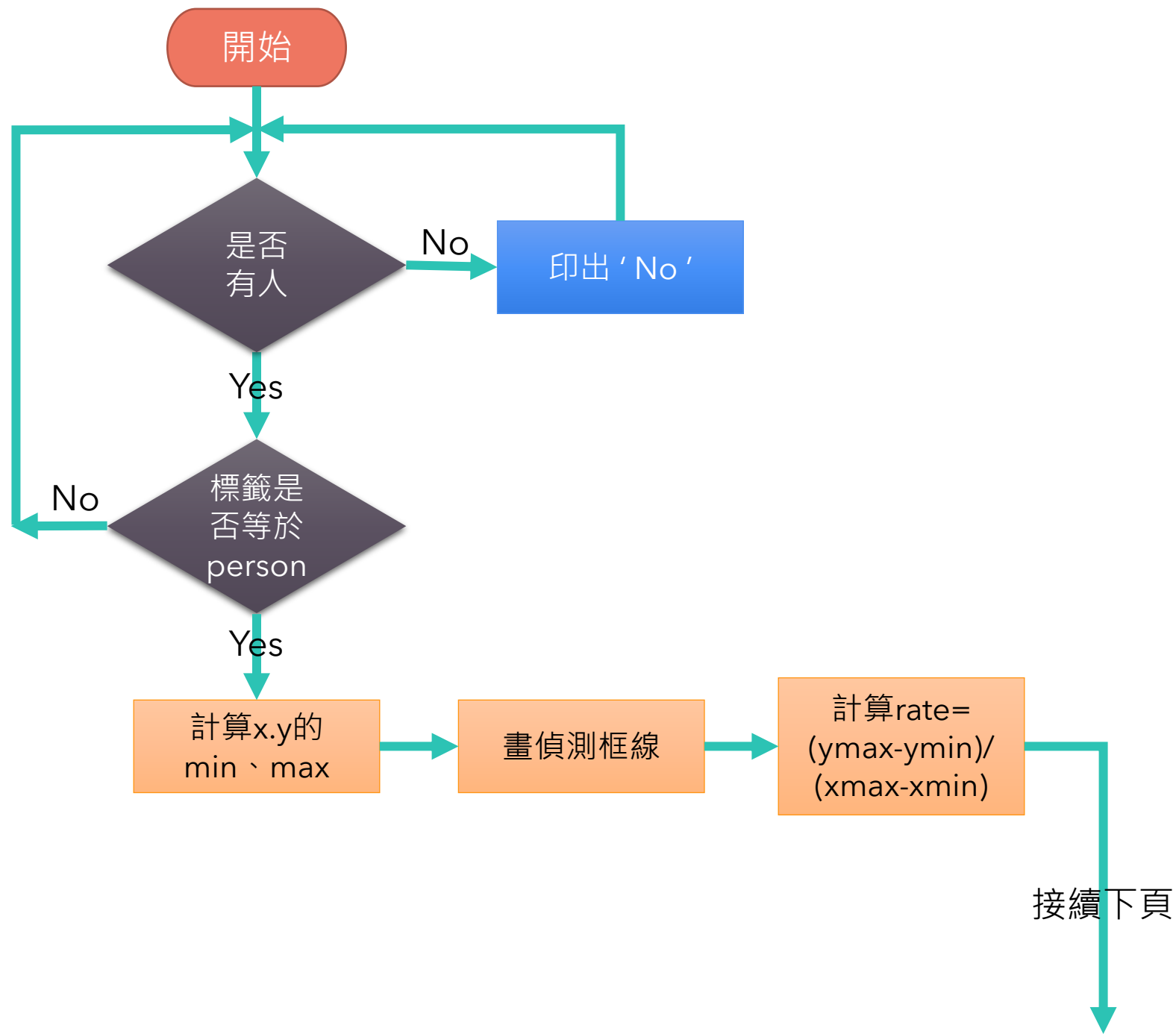
LED燈

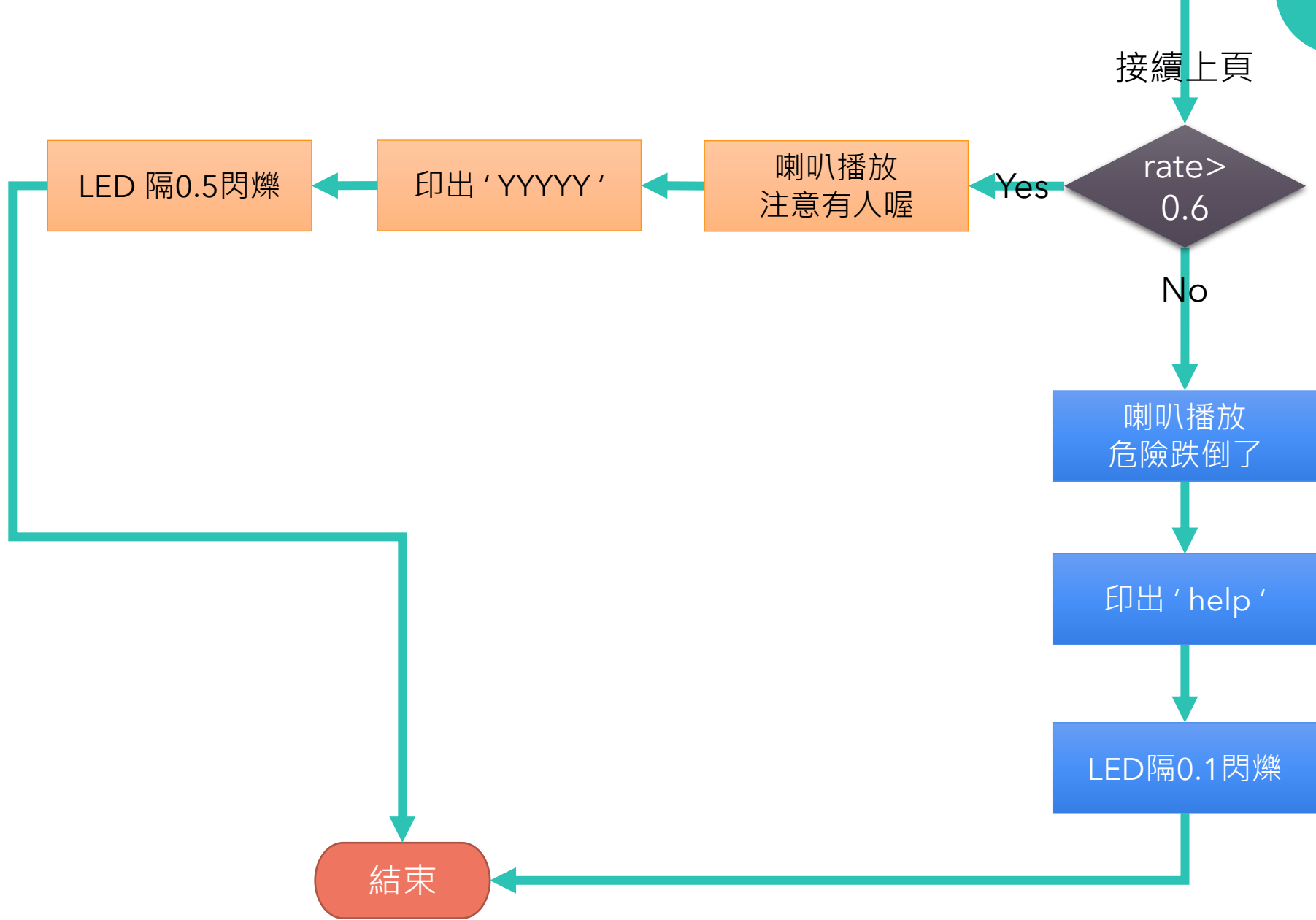




流程圖

流程圖







程式碼講解

LED 程式碼

接角設為26，接地為20

```
15
16 LED_PIN = 26
17 GPIO.setmode(GPIO.BOARD)
18 GPIO.setup(LED_PIN, GPIO.OUT)
19
```

設定間隔閃爍秒數，並印出HIGH、LOW確認是否成功執行

```
225 GPIO.output(LED_PIN, True)
226 print("HIGH")
227 time.sleep(0.5)
228 GPIO.output(LED_PIN, False)
229 print("LOW")
230 time.sleep(0.5)
```

確保程式中止時能夠執行並清掉GPIO的設定

```
283 GPIO.cleanup()
```

語音 程式碼

放置 speak 程式碼

```
21 def speak(sentence, lang, loops=1):
22     with tempfile.NamedTemporaryFile(delete=True) as fp:
23         tts=gTTS(text=sentence, lang=lang)
24         tts.save('{} .mp3'.format(fp.name))
25         mixer.init()
26         mixer.music.load('{} .mp3'.format(fp.name))
27         mixer.music.play(1)
```

設定sentence01、sentence02分別不同的語句

```
191 sentence01='注意'
192 sentence02='危險'
```

設定sentence句子為sentence01+警告語，並設置隔2秒播放一次

```
223 sentence=sentence01+'有人喔'
224 print('YYYYY')
225 GPIO.output(LED_PIN,True)
226 print("HIGH")
227 time.sleep(0.5)
228 GPIO.output(LED_PIN,False)
229 print("LOW")
230 time.sleep(0.5)
231
232 else:
233     sentence=sentence02+'跌倒了'
234     print('help')
235     GPIO.output(LED_PIN,True)
236     print("HIGH")
237     time.sleep(0.1)
238     GPIO.output(LED_PIN,False)
239     print("LOW")
240     time.sleep(0.1)
241
242     speak(sentence, 'zh')
243     time.sleep(2)
```

鏡頭 程式碼

印出classes，設定target為第0個

```
186 # Retrieve detection results
187 boxes = interpreter.get_tensor(output_details[0]['index'])[0] #
188 classes = interpreter.get_tensor(output_details[1]['index'])[0]
189 print('classes=', classes)
190 target=[0]
191 sentence01='注意'
192 sentence02='危險'
```

設定如果classes有第0個標籤(person)的話則計算x.y的min和max、並開始畫框線

```
196 # Loop over all detections and draw detection box if confidence is above minimum
197 for i in range(len(scores)):
198     if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):
199         if classes[i]==0:
200
201             # Get bounding box coordinates and draw box
202             # Interpreter can return coordinates that are outside of image dimension
203             ymin = int(max(1,(boxes[i][0] * imH)))
204             xmin = int(max(1,(boxes[i][1] * imW)))
205             ymax = int(min(imH,(boxes[i][2] * imH)))
206             xmax = int(min(imW,(boxes[i][3] * imW)))
207
208             cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (255, 151, 100), 2)
```

鏡頭 程式碼

設定框線、label、rate值，並印出x.y的min和max，以及rate值

```
207
208     cv2.rectangle(frame, (xmin,ymin), (xmax,ymax), (255, 151, 100), 2)
209
210     # Draw label
211     object_name = labels[int(classes[i])] # Look up object name from "labels" array using class index
212     label = '%s: %d%%' % (object_name, int(scores[i]*100)) # Example: 'person: 72%'
213     labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2) # Get font size
214     label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to top of window
215     cv2.rectangle(frame, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED)
216     cv2.putText(frame, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2) # Draw label text
217
218     print('ymin=',ymin,'ymax=',ymax,'xmin=',xmin,'xmax=',xmax)
219     rate=(ymax-ymin)/(xmax-xmin)
220     print('rate=',rate)
```

鏡頭 程式碼

設定rate值>0.6時，代表有偵測到人，而且是站立的，則喇叭播放'注意有人喔'，印出'YYYYY'，LED以0.5的速度閃爍
否則，代表有偵測到人，但是是跌倒的，則喇叭播放'危險跌倒了'，印出'help'，LED以0.1的速度閃爍

```
221 GPIO.output(LED_PIN,True)
222 if(rate>0.6):
223     sentence=sentence01+'有人喔'
224     print('YYYYY')
225     GPIO.output(LED_PIN,True)
226     print("HIGH")
227     time.sleep(0.5)
228     GPIO.output(LED_PIN,False)
229     print("LOW")
230     time.sleep(0.5)
231
232 else:
233     sentence=sentence02+'跌倒了'
234     print('help')
235     GPIO.output(LED_PIN,True)
236     print("HIGH")
237     time.sleep(0.1)
238     GPIO.output(LED_PIN,False)
239     print("LOW")
240     time.sleep(0.1)
241
242 speak(sentence, 'zh')
243 time.sleep(2)
```

鏡頭 程式碼

若沒有偵測到人，只會印出'NO'，LED則不會亮

```
244     .....else:
245     .....GPIO.output(LED_PIN,False)
246     .....print("NO")
```



問題與未來

問題1 rate值的設定

1. rate的值和攝影機擺放的位置也有間接的關係，需要完整的照到人，並且不斷的調整值。
2. rate值太大，可能人是坐著的狀態但卻顯示跌倒的回應。
3. rate值太小，也許人已經是平躺的狀態卻始終顯示站立的回應，因此這部分我們也測試滿久才找到一個平衡值。



問題2 攝影機的拍攝速度

1. 很多時候會是喇叭和LED燈已經有反應了，
但攝影機的畫面卻來是卡頓、延遲的狀態。
2. 透過刪減一些不必要的程式碼，降低系統的
作業時間。
3. 速度有比較快，但還是有卡頓和延遲的現象。



未來

1. 我們希望未來可以更進一步利用Line Bot的方式，連接使用者的手機，可以更及時的透過訊息通知，避免錯過急救的黃金時間。
2. 目前已經完成Line的環境建置，以及最後的測試。





現場展示



謝謝大家