

資訊專題競賽報告

強化學習智慧風扇

專題題目說明、價值與貢獻自評（限100字內）：

市面上的電扇大多只能手動調整風速大小，但若於不方便調整風速大小的環境下(如睡著)會相當不方便。因此我們藉由qllearning學習使用者可能的風速需求，設計一個自動化改變風速的智慧風扇。

專題隊員：

姓名	E-mail	負責項目說明	專題內貢獻度(%)
劉之岳	jeff87905@gmail.com	程式設計、概念構想	50%
劉易鑫	hahappast@gmail.com	硬體組裝、實驗	50%

【說明】上述表格之專題內貢獻度累計需等於100%。

目錄

- 一、 摘要（含關鍵詞）
- 二、 專題研究動機與目的
- 三、 專題重要貢獻
- 四、 團隊合作方式
- 五、 設計原理、研究方法與步驟
- 六、 系統實現與實驗
- 七、 效能評估與成果
- 八、 結論
- 九、 參考文獻

一、摘要

是不是常常爬起棉被開關電風扇呢? 明明剛開電風扇時相當舒服，但吹了一陣子後一陣涼意襲來，又不得已把棉被闔上。無奈的是，闔上棉被後卻又覺得悶熱煩躁不安，難以入眠。

每年 5、6 月是台灣的梅雨季，溫度變化大，一下子艷陽高照、悶熱異常，一下子又烏雲密布、寒風刺骨。每到這個時節，流行性感冒最容易趁虛而入。而在這種季節交替的環境下，人們往往在入夜時打開電風扇吹著強風，貪婪著一時的舒爽，卻沒想到在凌晨時，一場雷陣雨，帶走地表上的高溫，因而受了風寒。

人體生理機制與體溫的調節需要時間來因應環境溫度的變化，因此如果外部環境溫度變化過大，身體溫度調節機制反應不及或反應過度時，即可能對身體造成傷害或致命。

為了解決上述的情況，我們使用 Q-learning 設計了一個智慧型風扇，在熱時自動吹強，在冷時自動吹弱，解決睡眠上的困擾。

關鍵詞:Q-learning，智慧電風扇，個人化，藍芽

二、專題研究動機與目的

每個家庭多多少少都會有老人、嬰兒的存在，老人們的行動不便、嬰兒對外界的感知能力尚未成熟……等，對於短時間內動盪的環境溫度，無法像一般身強體壯的人們一樣做出最快的反應。就算聘請看護，由於每個人對於舒適溫度，感知上存在著根本差異，無法藉由人力上解決社會上的問題。

而市面上的電風扇大多只能手動控制風速，雖有些有遙控器可遠距改變電風扇轉速，但於睡覺時仍有困擾與不便。

此時便產生初步的構想，若電風扇能自動根據你的心聲改變風速的話那該有多好啊! 因此我們變設計出一個可自動根據你的需求改變風速的電風扇，並可適應多個不同的環境，讓你在不同的季節、地點皆可獲得最舒適的感覺。

三、專題重要貢獻

我們專題的研究成果可用於社會的普羅大眾，睡著後半夜溫度驟降，風扇會感測到溫度，調整大小，減少感冒機會。而由此種構想，可以進一步發展出將每個人對於舒適溫度，及電風扇吹風後的溫度變化存進資料庫中。可以使多個使用者在登入帳號後，能調用過去資料，在最短時間內，抵達舒適溫度。

而對於醫療的貢獻上，無論是在醫院的病房中，又或是在家不便行動患者。病患若無法下床或按遙控器調整風扇開關，可藉由此風扇自動感知周遭環境變數，避免病人进一步的惡化、減少醫療資源的浪費。

由電子機械類切入，若功能強大化後，可用 Q-learning 系統對機房的電量及溫度控制，達到一個最佳化的結果。

四、團隊合作方式

前期:

合作想出大架構，根據架構分成硬、軟體。兩人分別實驗不同元件並共同測試元件能不能符合程式的需求。

後期:

共同撰寫演算法模型及實驗方式，一人負責程式設計的架構及流程圖，另一人負責將概念架構實作化。

共同實驗風扇與環境溫度感測器間的互動關係、觀察、解決問題。

五、設計原理研究方法與步驟

大綱:

兩個 arduino 微處理器。並根據藍芽分為主、從端。主端搭載感測器獲得溫濕度，並在調控風扇風速大小獲得風速與體感溫度變化的關係後便進行 qlearning。qlearning 完畢後獲得 qtable，根據 qtable 選出最適合的風速，完成智能風扇。

步驟 1

將 DHT11 感測器裝置於 Arduino 主端

原理: 使用感測器監測環境變數。

研究方法: 1. 使用體感溫度公式替代傳統溫度感測貼片

2. 利用 dht11、風速計感測器測量環境變數再轉化為體感溫度

3. 測量使用者覺得舒適的體感溫度

步驟 2

於 Arduino 主端強化學習

原理: Q-learning

研究方法: 1. 在環境中隨機改變風扇大小，並藉此獲得哪個體感溫度吹哪個風扇大小會

跑到哪個體感溫度。例如: 攝氏 28.3 度 吹強 會跑到攝氏 27.5 度

2. 根據使用者舒適溫度以及體感溫度和風速大小之間的關係建立 reward table

3. 根據 rewardtable 的獎賞制度進行 Q-learning 演算法: $Q_{new}[s_t, a_t] = (1 - \alpha) * Q$

$[s_t, a_t] + \alpha * (r_t + \gamma * \max_a Q(s_{t+1}, a))$

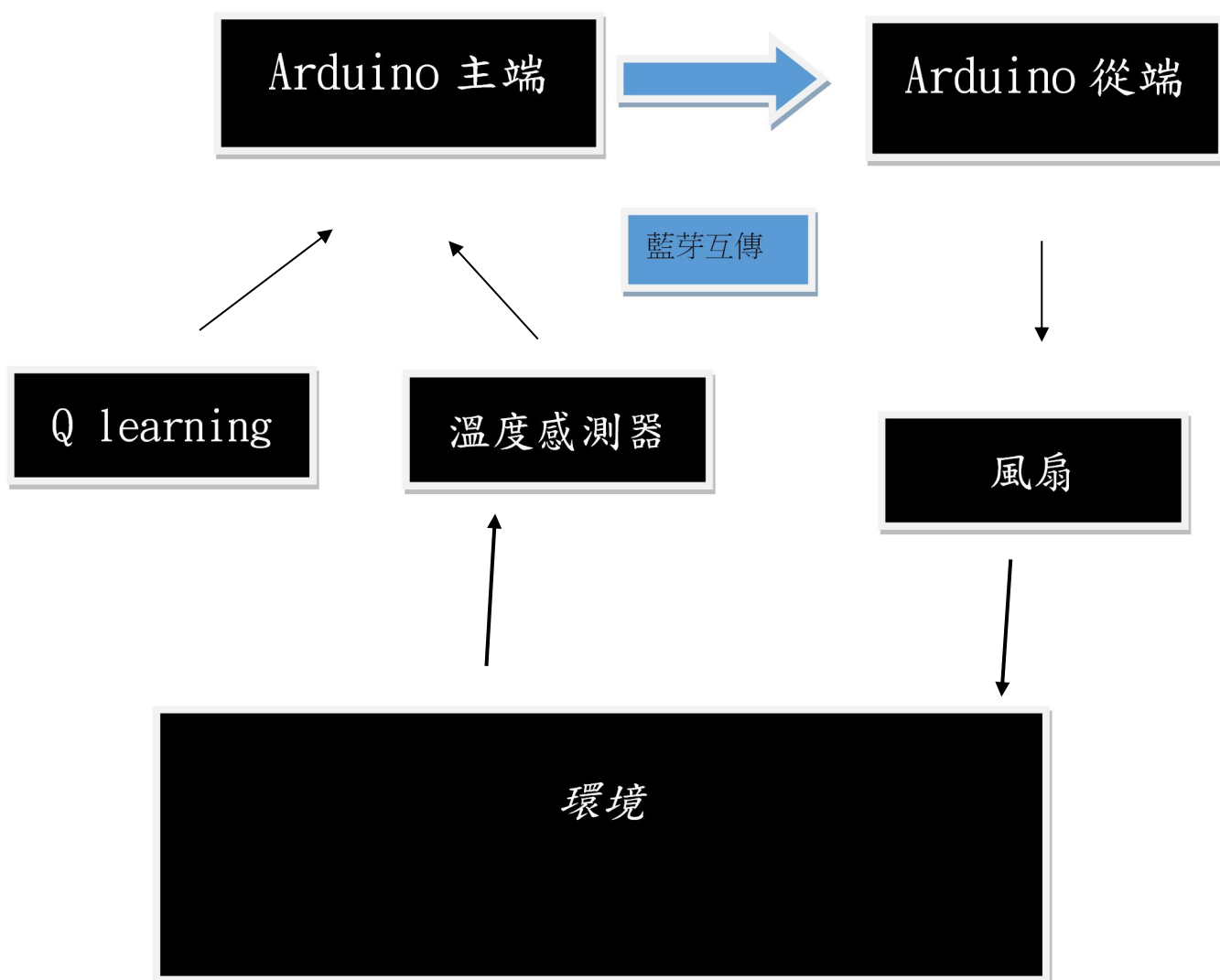
4. 獲得 qtable。

步驟 3

智慧風扇階段: Arduino 主端與 Arduino 從端連線

原理:arduino 主端根據 qtable 傳送轉動作(即弱、中、強)給從端改變風扇轉速。

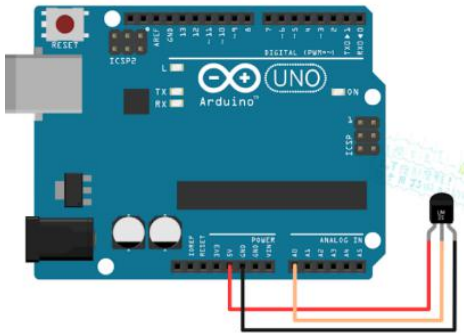
架構圖如下圖所示



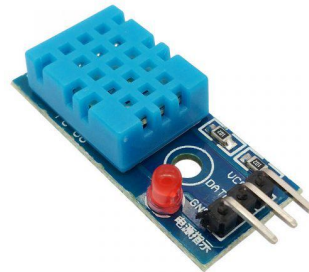
六、系統實現與實驗

實驗材料:

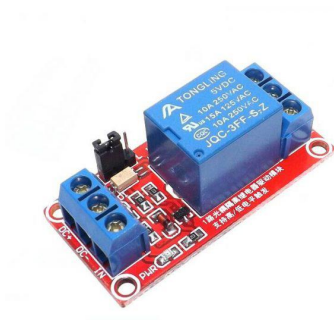
Arduino UNO *2



DHT11*1



Relay*1



bluetooth*2(一為主端，另一個為從端)



Cpu 風扇*1



風速計*1



系統實現:

1.

體感溫度公式:

$$AT=1.07T+0.2e-0.65V-2.7$$

$$e=(RH/100)*6.105*\exp(17.27T/(237.7+T))$$

其中 AT 為體感溫度 (°C)、T 為氣溫 (°C)、e 為水氣壓 (hPa)、

V 為風速 (m/sec)、RH 為相對濕度 (%)

氣溫:dht11 感測

水氣壓:dht11 感測

風速:風速計感測，其中弱風 4 (m/s), 中風 4.4(m/s), 強風 5(m/s)

2.

獲得使用者舒適溫度: 使用者按按鈕調控風扇大小。若在當前環境下滿意此風速則

按下確定鈕，我們便可獲得使用者舒適溫度。

3.

建立 reward_table: 在當前環境下改變風扇風速(弱、中、強)，若越接近舒適溫度，

其 reward 越大。建立 state_to_state_table 紀錄每個體感溫度操作哪個風扇動作後，會到達哪一個體感溫度。

4.

reward 機制的建立:

規則如下:

若使用者在目前狀態(當下溫度)下，操作完某一動作(電風扇轉速:弱、中、強)後，而新的狀態接近目標狀態(即舒適溫度)，則該事件我們給予“正數分數”的獎勵(即鼓勵)。

反之弱遠離目標狀態，則該事件我們給予“負獎勵”(即懲罰)

Reward 正數演算法:若目前狀態，操作該動作後，至目標狀態則給予滿分(100 分)的獎勵，每距離目標狀態 1 格距離，則依次遞減 10 分。

Reward 負數: 依照遠離的幅度給予值，遠離越遠給越大的負值(即懲罰越大)，遠離的越小給予的負值越小(即懲罰越小)。每遠離使用者舒適溫度一格，則多加一次-10

負數獎勵階梯化									
reward_table					state_to_state_table				
	狀態	動作				狀態	動作		
	溫度	弱	中	強		溫度	弱	中	強
	30.97	0	0	0		30.97	0	0	0
	31.17	0	0	0		31.17	0	0	0
	31.37	0	0	0		31.37	0	0	0
	31.57	80	30	30		31.57	32.57	31.57	31.57
	31.77	90	60	40		31.77	32.77	32.17	31.77
	31.97	80	60	50		31.97	32.57	32.17	31.97
	32.17	90	60	60		32.17	32.77	32.17	32.17
	32.37	80	40	40		32.37	32.57	31.77	31.77
	32.57	90	-60	-40		32.57	32.77	32.17	31.77
	32.77	90	-50	-50		32.77	32.77	31.97	31.97
goal	32.97	90	0	0	goal	32.97	32.77	0	0
	33.17	0	0	0		33.17	0	0	0
	33.37	0	0	0		33.37	0	0	0
	33.57	0	0	0		33.57	0	0	0
	33.77	0	0	0		33.77	0	0	0
	33.97	0	0	0		33.97	0	0	0
	34.17	0	0	0		34.17	0	0	0
	34.37	0	0	0		34.37	0	0	0
	34.57	0	0	0		34.57	0	0	0
	34.77	0	0	0		34.77	0	0	0
	34.97	0	0	0		34.97	0	0	0

5.

建立 qtable 的方法:

實作 Q-learning 演算法: $Q_{\text{new}}[s_t, a_t] = (1 - \alpha) * Q[s_t, a_t] + \alpha * (r_t + \gamma * \max Q(s_{t+1}, a))$:

其中 Q 為 qtable、s 為狀態、a 為動作、r 為獎勵值、 α 為學習率 ($0 < \alpha \leq 1$), γ 為衰退值 ($0 < \gamma \leq 1$)。

在 epsilon greedy=0.9 情況下操作 Q-learning

即有 0.9 的機率操作 Q-learning 另外 0.1 的機率使用 argument_max 的方法

Q-learning: qtable 的更新方法為少量學習過去的舊資料以及預測大量預測未來的資料。

其中未來資料的預測包含 3. reward_table, 而在 $\max Q(s_{t+1}, a)$ 中下一步 state 使用隨機變數預測。

Argument_max: 在 $\max Q(s_{t+1}, a)$ 中下一步 state 使用當前 qtable 最大值。

將 qtable 值寫入 eeprom, 以便 arduino 斷電後 qtable 不會消失。

6.

建立智慧風扇:

透過 5. 完成的 qtable 找出最短循環之風扇動作

使用者溫度初值(state) 其對應之 qtable 有 3 個 action(分別為弱、中、強)將其中最大的 qtable[state][action]值找出，並根據此動作藉由 delta table $\delta[state][action]$ 跳往下一個溫度，不斷循環。

實驗:

1.實驗環境:將 arduino 主端、從端至於一半封閉箱子內，隔絕非風扇風速導致的氣流影響。

2..實驗過程:量測當前環境下使用者舒適溫度後量測體感溫度與風速之間的關係 30 分鐘，並據此建立 reward_table。根據 reward_table 獲得 qtable，風扇依據 Qtable 改變風扇大小。並觀察有無達成智慧功能，比起直接用 reward_table，經過 qlearning 後更能將體感溫度控制在使用者舒適溫度左右。

3.實驗結果:

現象 1:

觀察實驗結果，如下:

於量測使用者舒適溫度階段下，若使用者按“確定鍵”確認舒適溫度的風扇大小為“弱”，則建置 reward_table 有值的部分偏向高於使用者舒適溫度。

若風扇大小為“中”時，整個 reward_table 有值的部分較為均勻的分佈在使用者舒適溫度附近。

若風扇大小為“強”時，整個 reward_table 有值的部分偏向低於使用者舒適溫度

圖表：

user選擇:弱					user選擇:中					user選擇:強				
reward_table	溫度	弱	中	強	reward_table	溫度	弱	中	強	reward_table	溫度	弱	中	強
	29.59	50	10	0		32.06	0	0	0		30.35	0	0	0
	29.74	50	10	0		32.21	0	0	0		30.55	0	0	0
	29.89	40	-10	-10		32.36	0	0	0		30.75	0	0	0
	30.04	70	-20	0		32.51	0	0	0		30.95	0	0	0
	30.19	50	-20	0		32.66	0	0	0		31.15	0	0	0
	30.34	70	-10	-10		32.81	0	0	0		31.35	0	0	0
	30.79	60	-20	-40		32.96	0	0	0		31.55	0	0	0
	30.49	70	-20	-40		33.11	80	80	80		31.75	0	0	0
	30.64	90	-60	-50		33.26	60	70	-70		31.95	0	0	0
	30.79	-80	-50	-50		33.41	60	90	-80		32.15	0	0	0
goal	30.94	80	0	0	goal	33.56	70	90	80	goal	33.35	60	100	100
	31.09	90	0	0		33.71	-70	100	80		32.55	-40	100	100
	31.24	90	0	0		33.86	-70	80	90		32.75	-40	100	80
	31.39	0	0	0		34.01	70	100	80		32.95	70	100	80
	31.54	0	0	0		34.16	60	90	90		33.15	70	90	90
	31.69	0	0	0		34.31	60	0	0		33.35	50	80	80
	31.84	0	0	0		34.46	0	0	0		33.55	60	80	70
	31.99	0	0	0		34.61	0	0	0		33.75	0	0	0
	32.14	0	0	0		34.76	0	0	0		33.95	0	0	0
	32.29	0	0	0		34.91	0	0	0		34.15	0	0	0
	32.44	0	0	0		35.06	0	0	0		34.35	0	0	0

實驗結果解讀：

若使用者在該環境下選擇弱風作為舒適溫度的話。我們觀察到建立 reward_table 階段較難將溫度吹高於使用者舒適溫度，會有一個上限存在。根據這個現象我們發現風速對於實際環境溫度的改變並不大，但對於體感溫度的差異卻是巨大的。

現象 2:

當我們改變 Y(耗減率)的值時，發現耗減率 0.2 會忽略前期看似不是最優選，但後期值卻一帆風順到達使用者舒適溫度的情形。

而 $Y = 0.9$ 卻較能反映此種情況

alpha=0.3,gamma=0.9															
reward_table					state_to_state_table					qtable					
	狀態	動作				狀態	動作				狀態	動作			
	溫度	弱	中	強		溫度	弱	中	強		溫度	弱	中	強	
	32.06	0	0	0		32.06	0	0	0		32.06	7	0	0	
	32.21	0	0	0		32.21	0	0	0		32.21	7	0	0	
	32.36	0	0	0		32.36	0	0	0		32.36	7	0	0	
	32.51	0	0	0		32.51	0	0	0		32.51	7	0	0	
	32.66	0	0	0		32.66	0	0	0		32.66	7	0	0	
	32.81	0	0	0		32.81	0	0	0		32.81	0	7	0	
	32.96	0	0	0		32.96	0	0	0		32.96	0	7	0	
	33.11	80	80	80		33.11	33.86	33.86	33.26		33.11	643.31	667.91	722.14	
	33.26	60	70	-70		33.26	34.16	34.01	33.11		33.26	733.33	742.73	570.64	
	33.41	60	90	-80		33.41	34.16	33.71	33.26		33.41	686.16	729.67	543.04	
goal	33.56	70	90	80	goal	33.56	34.01	33.71	33.26	goal	33.56	737.05	746.23	730.54	
	33.71	-70	100	80		33.71	34.01	33.56	33.26		33.71	587.37	755.26	738.75	
	33.86	-70	80	90		33.86	34.01	33.86	33.41		33.86	583.72	659.86	691.17	
	34.01	70	100	80		34.01	34.01	33.56	33.26		34.01	744.26	755.34	746.45	
	34.16	60	90	90		34.16	34.16	33.71	33.41		34.16	722.42	763.01	723.44	
	34.31	60	0	0		34.31	34.16	0	0		34.31	0	0	7	
	34.46	0	0	0		34.46	0	0	0		34.46	0	0	7	
	34.61	0	0	0		34.61	0	0	0		34.61	0	0	7	
	34.76	0	0	0		34.76	0	0	0		34.76	0	0	7	
	34.91	0	0	0		34.91	0	0	0		34.91	0	0	7	
	35.06	0	0	0		35.06	0	0	0		35.06	0	0	7	

表 1(耗減率 = 0.9)

alpha=0.3,gamma=0.2																
reward_table	state_to_state_table								qtable							
	狀態	動作				狀態	動作				狀態	動作				
	溫度	弱	中	強		溫度	弱	中	強		溫度	弱	中	強		
	32.06	0	0	0		32.06	0	0	0		32.06	7	0	0		
	32.21	0	0	0		32.21	0	0	0		32.21	7	0	0		
	32.36	0	0	0		32.36	0	0	0		32.36	7	0	0		
	32.51	0	0	0		32.51	0	0	0		32.51	7	0	0		
	32.66	0	0	0		32.66	0	0	0		32.66	7	0	0		
	32.81	0	0	0		32.81	0	0	0		32.81	0	7	0		
	32.96	0	0	0		32.96	0	0	0		32.96	0	7	0		
	33.11	80	80	80		33.11	33.86	33.86	33.26		33.11	102.26	102.12	98.8		
	33.26	60	70	-70		33.26	34.16	34.01	33.11		33.26	82.91	94.58	-49.55		
	33.41	60	90	-80		33.41	34.16	33.71	33.26		33.41	82.79	113.51	-61.05		
goal	33.56	70	90	80	goal	33.56	34.01	33.71	33.26	goal	33.56	94.58	114.57	98.92		
	33.71	-70	100	80		33.71	34.01	33.56	33.26		33.71	-45.42	122.9	98.91		
	33.86	-70	80	90		33.86	34.01	33.86	33.41		33.86	-45.42	102.32	112.34		
	34.01	70	100	80		34.01	34.01	33.56	33.26		34.01	94.58	122.91	98.92		
	34.16	60	90	90		34.16	34.16	33.71	33.41		34.16	82.88	114.58	112.52		
	34.31	60	0	0		34.31	34.16	0	0		34.31	0	0	7		
	34.46	0	0	0		34.46	0	0	0		34.46	0	0	7		
	34.61	0	0	0		34.61	0	0	0		34.61	0	0	7		
	34.76	0	0	0		34.76	0	0	0		34.76	0	0	7		
	34.91	0	0	0		34.91	0	0	0		34.91	0	0	7		

表 2 (耗減率 = 0.2)

現象 2 之實驗結果與分析:

比較表 1 與表 2:

於狀態(33.11 度)，Y=0.2 在智慧風扇階段下選擇為弱; Y=0.9 在智慧風扇階段下選擇為強

比較過程:

Step1:

選定相同溫度比較，表一與表二的選擇為狀態(33.11 度)

Step2:

表一在狀態(33.11 度)中，選擇 qtable 分數最高的值，所以表一的第一個選擇為“強”，觀察 state_to_state table 中狀態(33.11 度)選擇動作(“強”)，會到達新的狀態(33.26 度)。

Step3:

持續迭代 Step2 直到到達目標狀態為止。

比較結果:

狀態(33.11 度)的第一個動作選擇“強”，會擁有最短路徑，因此效能評比: $Y=0.9 > Y=0.2$

實驗分析:

根據上圖 qtable，同一狀態的 3 個動作中若出現相近的數字，表示自該狀態到達目標狀態的步數有高機率相同。

實驗結論: 因耗減率 $Y = 0.9$ 較能找出真正最快到達使用者舒適溫度的吹法，故而採用 $Y = 0.9$ 作為我們 Qlearning 演算法的使用

現象 3:

qtable 出現某一個狀態，是沒被更新到的，如下圖:

reward table與state to state table頻繁更新															
reward_table					state_to_state_table					qtable					
	狀態	動作				狀態	動作				狀態	動作			
	溫度	弱	中	強		溫度	弱	中	強		溫度	弱	中	強	
	30.97	0	0	0		30.97	0	0	0		30.97	0	0	0	
	31.17	0	0	0		31.17	0	0	0		31.17	0	0	0	
	31.37	0	0	0		31.37	0	0	0		31.37	0	0	0	
	31.57	80	30	30		31.57	32.57	31.57	31.57		31.57	0	0	0	
	31.77	90	60	40		31.77	32.77	32.17	31.77		31.77	703.53	709.16	579.48	
	31.97	80	60	50		31.97	32.57	32.17	31.97		31.97	759.29	775.1	722	
	32.17	90	60	60		32.17	32.77	32.17	32.17		32.17	804	779.36	778.84	
	32.37	80	-40	-40		32.37	32.57	31.77	31.77		32.37	0	0	0	
	32.57	90	-60	-40		32.57	32.77	32.17	31.77		32.57	780.39	613.84	563.39	
	32.77	90	-50	-50		32.77	32.77	31.97	31.97		32.77	796.48	639.18	630.29	
goal	32.97	90	0	0	goal	32.97	32.77	0	0	goal	32.97	27	0	0	
	33.17	0	0	0		33.17	0	0	0		33.17	0	0	0	
	33.37	0	0	0		33.37	0	0	0		33.37	0	0	0	
	33.57	0	0	0		33.57	0	0	0		33.57	0	0	0	
	33.77	0	0	0		33.77	0	0	0		33.77	0	0	0	
	33.97	0	0	0		33.97	0	0	0		33.97	0	0	0	
	34.17	0	0	0		34.17	0	0	0		34.17	0	0	0	
	34.37	0	0	0		34.37	0	0	0		34.37	0	0	0	
	34.57	0	0	0		34.57	0	0	0		34.57	0	0	0	
	34.77	0	0	0		34.77	0	0	0		34.77	0	0	0	
	34.97	0	0	0		34.97	0	0	0		34.97	0	0	0	

實驗結果解讀:

在 reward_table 階段，後來更新的值將前個值替換掉，導致某些 state 於 qlearning 階段無法被走訪到，因為能走到他的 state 被刷新成走到其他的 state。

實驗結論: 不要採用最近更新，而是採用若該 state 做哪個動作到哪個 state 已有資料，則不更新它。

七、效能評估與成果

效能評估:

1.智慧風扇於實際環境吹送效果:

因礙於 CPU 風扇葉片大小因此風力有限，導致實際情況下，得限制特定條件才有較顯著的效果，例如限定智慧風扇得離人體三公尺內、或是環境不能劇烈改變，使得 Qtable 難以實行。但若於外在氣流缺少干擾且距離較近的情形下，AI_Fan 確實能達成市售風扇難以實現的智慧效果，因為於該環境下智慧風扇 qlearning 過後，智慧風扇就獲得了預測未來情況能力。

2. 有無做 qlearning 效能差異

若無做 qlearning 而是根據使用者回饋“這樣吹會冷還是會熱”來當作依據的話，則因為無法預測後續變化，所以體感溫度(可當作使用者感受溫度)於使用者舒適溫度範圍的浮動較大。若是有做 qlearning 的話則能一定程度預測後續的改變，進而選擇較佳的動作控制體感溫度於使用者舒適溫度附近。

八、結論

自 20 世紀中後期，嵌入式系統就已漸漸踏入人們的生活，小至觸控式電燈、電子錶，大至汽車、冷氣其背後都有嵌入式系統的影子。時至今日，人工智慧、大數據，蔚為主流的年代，2016 年電腦圍棋軟體 AlphaGo 挑戰韓國棋王李世石藉由人工智慧預判棋路。因此，利用演算法預測未來可能發生的事件並非不可能的任務。

本研究將嵌入式系統結合人工智慧，使產品具有智慧效能管理。在研究過程中發現人類舒適溫度其對應結果應為舒適溫度，包含許多不同的環境變數。而 Q-learning 模型可以有效率的實現自我學習並分析數據，藉由表格將體感溫度最佳路徑循環具象化。

九、參考文獻

Francisco S. Melo, "[Convergence of Q-learning: a simple proof](#)" Robert G. Steadman , 1984, J Clim Appl Meteorol 23:1674–1687.

Hasselt, Hado van. Reinforcement Learning in Continuous State and Action Spaces. Wiering, Marco; Otterlo, Martijn van. [Reinforcement Learning: State-of-the-Art](#). Springer Science & Business Media. 5 March 2012: 207–251.