物聯網期末專題

智能風扇

Smart Fan

組員: 4109056001 施昶宇

4109056008 陳政鴻

4109056018 蔡明賢

4109056030 馬國維

目錄

1. 摘要………………………………………………………………..……..1
2. 專題研究動機與目的………………………………………………..…..1

三、 設計原理、研究方法與步驟……………………………………….........2

3.1 機器學習原理……………………………………………………......2

3.2 Q-learning原理…………………………………………………..….3

3.3藉由PWM控制風扇轉速………………………………………..….4

3.4 選擇強化學習― Q-learning……………………………………..…..5

3.5設計主要的訓練演算法……...…………………………………..…..6

四、系統實現與實驗………………………………………………..…….7

4.1 硬體方面…………………………………………………….…..…7

4.1.1 Arduino Mega2560…………….......................................7

4.1.2 ESP8266……………………….......................................8

4.1.3 L298N………………………………………………….9

4.1.4 DS18B20……………………………………………....10

4.2風扇和各個模組的實際接線圖…………………………………....11

4.3 軟體方面……………………………………………………………12

4.3.1 使用者登入………………………………………………..12

4.3.2 爬取天氣…………………….……………………………..12

4.3.3 ESP8266連上WIFI、TCP連線………………………....13

4.3.4 連線資料庫取得資料……………………………………...14

4.3.5 用q-learning開始訓練………………………………...…..14

4.3.6 訓練結果更新資料庫…………………………………..….15

4.3.7整體流程圖……………………………………………..…15

4.4 實驗設計……………………………………………………………16

五、效能評估與成果……………………………………………………….…17

六、結論…………………………………………………….…………………18

七、參考文獻……………………………………………………………….…19

1. **摘要**

**一般市面上的風扇只能固定轉速，並且沒有考慮人體溫度，缺乏便利和舒適性，為了解決這個問題，我們使用Q-Learning結合電風扇，實現出個人化的電扇，並利用ESP8266將風扇連上WIFI，建立與資料庫的連線，將q\_table上傳至資料庫儲存，讓下一位使用者登入時，能夠從資料庫存取自己的q\_table，經過長時間的訓練之後，能夠根據使用者狀態自動調整到最佳的狀態。**

**關鍵詞:** **Reinforcement learning、智能風扇、ESP8266、資料庫。**

**二、專題研究動機與目的**

**電風扇是日常生活必備的家電，但是市面上大多僅考慮環境溫度而沒有考慮使用**

**者體溫的風扇。夏天在冷氣房時，一般風扇只能固定風速，剛開始覺得很熱，但因熟睡後風速不變，體溫下降，經常因為冷醒才把電扇關掉，十分缺乏便利和舒適性。因此我們希望能設計一個能根據人體溫度自動調整風速的電扇，來讓使用者能維持在最舒適的溫度。**

**三、設計原理、研究方法與步驟**

**3.1****機器學習原理**

**機器學習是一門開發演算法和統計模型的科學，旨在讓電腦從大量的數據中學習並自主地進行預測、分類、集群等任務，而不需要明確的程序或規則。主要分成三大類，監督式學習、非監督式學習和強化學習三種類型，如圖3.1。**

|  |
| --- |
|  |
| **圖3.1 機器學習架構圖[11]** |

**本次專題主要利用其中的強化學習，在強化學習中，agent通過觀察環境的狀態，進行行動，並根據環境的反饋學習最佳行動策略。強化學習不需要正確的輸出輸入，而是強調如何基於環境而行動，以取得最大化的預期利益，十分符合我們專題所需，圖3.2為題例說明。**

|  |
| --- |
|  |
| **圖3.2 強化學習主要流程[10]** |

**3.2 Q-learning原理**

**Q-learning中會需要一個代理人(Agent)，用來表示在不同環境下做出決定的角色，而我們需要決定角色面對到的狀態(State)，和定義面對此狀態所做出的動作(Action)，最後每個state和action對應到的即為Q值，Q值越大Agent越容易以其**

**action做為下一步的動作，圖3.3為Q-learning演算法。**

|  |
| --- |
|  |
| **圖3.3 Q-learning演算法[8]** |

**其中的核心公式Q-function由圖3.4所示，Q(s, a)是在狀態s下採取行動a的Q值，α是學習率控制了每次更新Q值時新信息的貢獻程度。如果學習率太高，更新Q值時可能會太過於激進，導致算法無法收斂或收斂緩慢。r是在狀態s下採取行動a後收到的，γ是衰減因子，γ值越大，未來回報的價值越高，Agent更加重視未來的回報，將更加注重長期利益，maxQ(s', a')是在狀態s'下所有可能的行動a'的最大Q值。**

|  |
| --- |
|  |
| **圖3.4 Q-function[9]** |

**3.3****藉由PWM控制風扇轉速**

**脈波寬度調變( Pulse Width Modulation ) 簡單 PWM ，是一種控制脈波工作週期(duty cycle)來達到控制輸出電壓的一種控制方式。由下方公式可知方波的直流值是與工作週期成正比，因此我們只要改變工作週期，就可以改變輸出的電壓(圖3.5)進而改變風扇的轉數，在arduino中可以設定的週期範圍為0到255。**

**VDC= Max Vin x Duty Cycle**

|  |
| --- |
|  |
| **圖3.5 輸出電壓對工作週期折線圖** |

**3.4選擇強化學習― Q-learning**

**我們主要考量人體溫度和風扇轉速的關係，溫度的變化是連續不斷的，而且對於使用者來說，冷熱的感受是十分主觀的，我們無法事先將資料做標記，因此我們選強化學習當中的Q-learning，就由讓風扇探索環境累積經驗，在未來遇到類似經驗時就能選出最優的決策。**

**建構Q-table**

**我們將action從-100每格遞增10到100，總共21格，以及state為根據目前體溫和目標溫度的差值以每格0.2℃的差距，從+8.0℃到-8.0℃，一共81格如表4.1所示。**

**表3.1 風扇之Q-table**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PWM** | | | | | | | | | |
| **溫差** |  | **-100** | **-90** | **…** | **-10** | **0** | **+10** | **…** | **+90** | **+100** |
| **+8.0** |  |  |  |  |  |  |  |  |  |
| **...** |  |  |  |  |  |  |  |  |  |
| **+0.4** |  |  |  |  |  |  |  |  |  |
| **+0.2** |  |  |  |  |  |  |  |  |  |
| **0.0** |  |  |  |  |  |  |  |  |  |
| **-0.2** |  |  |  |  |  |  |  |  |  |
| **-0.4** |  |  |  |  |  |  |  |  |  |
| **…** |  |  |  |  |  |  |  |  |  |
| **-8.0** |  |  |  |  |  |  |  |  |  |

**3.5設計主要的訓練演算法**

**(1)偵測溫度**

**(2)查詢q-table並選擇最大q值的action**

**(3)檢查溫度是否有往目標溫度前進有則加分，反之扣分**

**(4)利用圖5.4中的公式更新Q-table**

**回到(1)繼續訓練直到結束，圖3.6為其流程圖**

|  |
| --- |
|  |
| **圖3.6 訓練驗算法流程圖** |

**四、系統實現與實驗**

**4.1硬體方面:**

* + **Arduino Mega2560**
  + **ESP8266**
  + **L298N**
  + **DS18B20**

**4.1.1 Arduino Mega2560**

**於Arduino系列開發板的一員(圖4.1)。它擁有54個數字I/O引腳（其中15個可作為PWM輸出），16個類比輸入引腳，4個UART串口，16 MHz的晶體振盪器，256 KB的Flash存儲器，8 KB的SRAM和4 KB的EEPROM。它支持USB接口和外部電源供應，可通過USB接口與電腦連接進行溝通和數據傳輸。應為其儲存空間較大，因此我們使用此來當作風扇和軟體之間的橋樑。**

|  |
| --- |
|  |
| **圖4.1 Arduino mega 2560** |

**4.1.2 ESP8266**

**此模組能讓上述的arduino開發版能連上wifi，ESP8266有許多系列，我們是使用ESP-01s來讓我們的風扇能跟資料庫連結。ESP-01s模組具有8個引腳，包括2個GPIO、UART串口通訊口和ADC（類比數位轉換器）引腳，並且具有內置的WiFi天線。它可以透過AT指令進行控制，支持STA（station）和AP（access point）模式，並且可以通過TCP/IP協議進行無線通訊。此外，ESP-01s還具有支持WPA/WPA2安全協議的加密功能，可保護無線通訊的安全性，圖4.2、圖4.3為其與arduino 開發板的接腳圖。**

|  |
| --- |
|  |
| **圖4.2 Arudino 和 ESP01s的接線圖[12]** |

|  |
| --- |
|  |
| **圖4.3 Arudino 和 ESP01s的實際接線圖** |

|  |  |  |  |
| --- | --- | --- | --- |
| **4.1.3 L298N**  **L298N(圖4.4)是馬達驅動模組，可用於控制直流馬達和步進馬達等負載，電壓範圍為5V至35V，它可以通過控制輸入端的信號控制馬達的轉向和速度，正好符合我們挑整風扇轉數的需求。** | |  | | --- | |  | | **圖4.4 L298N** | |
|  |  |

**表4.1為L298N和Arduino mega 2560的接線關係，其中L298N需要外接12V的電源，並將L298N上的GND接到Arduino mega 2560上，使其能夠順利運行。**

**表4.1 mega 對L298N腳位**

|  |  |
| --- | --- |
| Arduino mega 2560 | L298N |
|  | 12V外部電源 |
| GND | GND |
| Pin8(PWM) | IN1 |
| Pin9(PWM) | IN2 |

**4.1.4 DS18B20**

**它使用1-Wire（One-Wire）資料傳輸協定，資料傳輸只需一條線即可，需要三個腳位（電源、資料、接地）。此溫度感測元件溫度感測範圍-55°C到+125°C，而且在 -10 °C～85 °C 這個範圍內保證 ±0.5°C 的精確度，偵測的溫度讀數已在內部校正為攝氏刻度，因此我們選擇此元件來偵測人體的體溫，如圖4.5。**

|  |
| --- |
|  |
| **圖4.5 DS18B20** |

**表4.2 mega 對L298N腳位**

|  |  |
| --- | --- |
| **Arduino mega2560** | **DS18B20** |
| GND | GND |
| 5V | VCC |
| Pin5 | Data |

**4.2風扇和各個模組的實際接線圖**

**圖4.6、圖4.7為風扇的實際接線圖**

|  |
| --- |
|  |
| **圖4.6風扇實際接線圖** |
|  |
| **圖4.7 風扇實際接線圖** |

**4.3軟體方面**

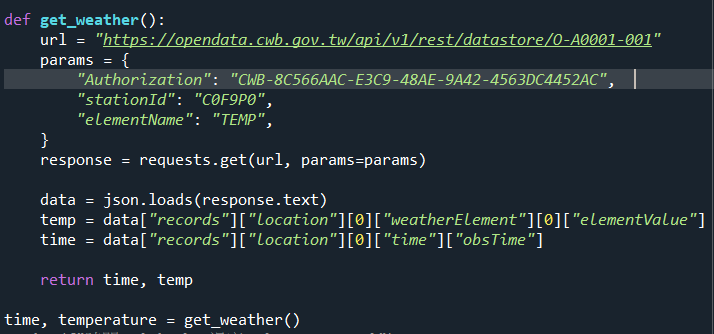
**4.3.1 使用者登入**

**使用者可利用QR code掃描登入， 用戶可註冊新帳號或登入舊有的帳號如圖4.8。**

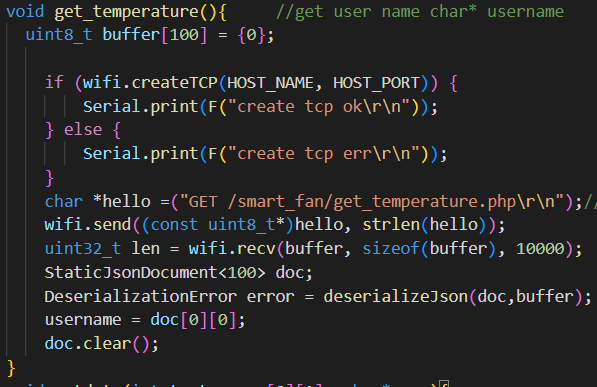
|  |
| --- |
|  |
| **圖4.8 登入畫面** |

**4.3.2 爬取天氣溫度**

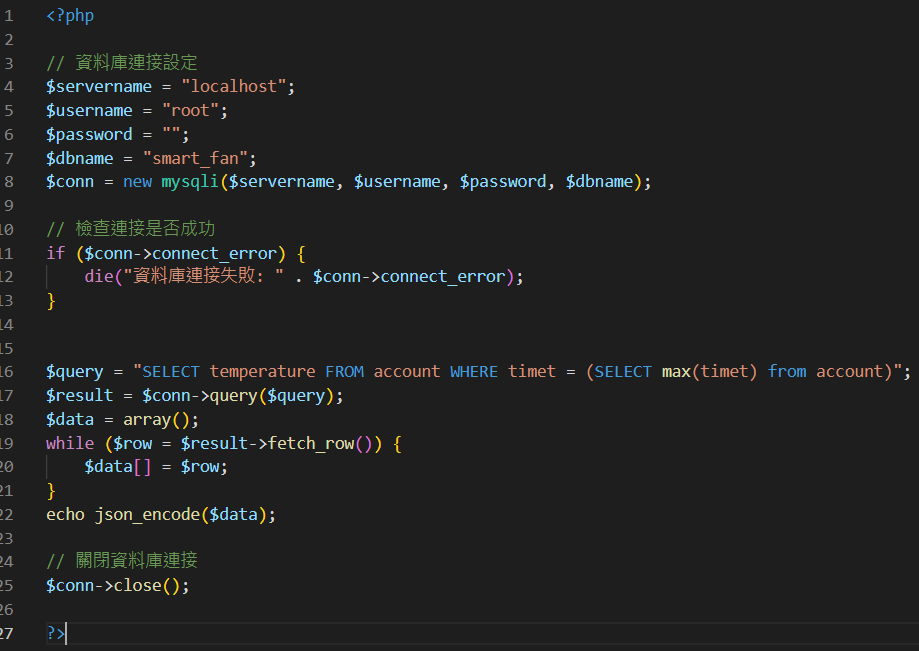
**1.利用python爬取溫度存入資料庫**



**2.向伺服器要求天氣資訊**



**3.PHP發送天氣資料**



**4.3.3 ESP8266連上WIFI、TCP連線**

**利用ESP8266函式庫連到指定的WIFI，風扇和伺服器建立TCP連線。**

**4.3.4連線資料庫取得資料**

**使用者登入後，系統會抓取之前的q-table，並將其傳送給風扇，如未登入過，將初始化新的table再傳給風扇，如圖4.9所示。**

|  |
| --- |
|  |
| **圖4.9 登入及傳送資料示意圖** |

**4.3.5用q-learning開始訓練**

**一開始會設定一個基本風速，記錄使用者的感受並改變風速，例如感覺冷會使風速降低，反之風速增加，當使用者感受到一次冷和一次熱時，利用二分法，推斷使用者最適溫度落在此區間內，藉由兩個溫度取中間值當目標溫度開始訓練。**

**每3秒偵測溫度，從q-table中選出q值最高的action，並計算reward，計算方式如下:**

* **判斷此action是否有使溫度靠近目標溫度，如果是，reward為此溫度差乘10，不是則為負的。**
* **溫度不變但溫度為目標溫度則reward為20，反之如果不是目標溫度則reward為-20**

**最後根據q-function來更新q-table。**

**4.3.6訓練結果更新資料庫**

**利用ESP8266跟資料庫建立TCP連線，將q-table傳給資料庫，如圖4.10 所示。**

|  |
| --- |
|  |
| **圖4.10 Q\_table上傳示意圖** |
|  |

**4.3.7整體流程圖**

**圖4.11 為風扇整體架構的流程圖**

|  |
| --- |
|  |
| **圖4.11 整體流程圖** |

**4.4實驗設計**

**將風扇放置於離受試者約40cm處，吹向胸口至脖子的位置，檢測溫度變化，最後根據其測量值來觀看效果，用同樣方式測量一般風扇對溫度的影響，並比較兩者間的差異。**

**五、效能評估與成果**

**圖5.1為使用q-learning訓練的結果，Y軸為溫度和X軸為訓練次數的關係圖，一開始epsilon值較高，會採取較多隨機action，造成前期溫度較不穩定，隨著訓練次數增加，降低epsilon的比率，讓風扇可以根據q-table來選擇action，溫度也因此逐漸朝目標溫度收斂。**

|  |
| --- |
|  |
| **圖5.1 溫度對次數的折線圖，紅線為目標溫度32.78°C** |

**圖5.2為PWM值和次數的關係圖，相關係數為-0.16，從圖中可得知PWM值隨著訓練次數增加， PWM值越小，由圖5.3可看出呈現負相關。根據圖5.1，溫度最後會逐漸在目標溫度附近，因此風扇會選擇PWM較小的值來穩定溫度。**

|  |
| --- |
|  |
| **圖5.2 PWM對次數的折線圖，紅線為PWM和times的線性關係** |

|  |
| --- |
|  |
| **圖5.3 PWM對溫度的點陣圖，紅線為PWM和times的線性關係** |

**圖5.4為一般固定轉速之風扇，隨著次數的增加，體溫逐漸下降，偏離目標溫度**

|  |
| --- |
|  |
| **圖5.4溫度對次數的折線圖，紅線為目標溫度32.78°C** |

**六、結論:**

**有q-learning訓練的風扇在一開始訓練時雖然會花較多時間在訓練，但是隨著時間推移，很明顯較一般市售風扇在維持最適溫度有更好的表現。不過因為風扇無法對環境溫度和濕度有太大的影響，因此在不同的室溫和環境下，都可能會影響最適溫度的決定。**

**七、參考文獻:**

[1] ITEADLIB\_Arduino\_WeeESP8266 <https://github.com/itead/ITEADLIB_Arduino_WeeESP8266>

[2] esp8266 基本測試

<http://maker.tn.edu.tw/modules/tad_book3/page.php?tbsn=21&tbdsn=390>

[3] ESP8266 WiFi 模組(ESP-01)韌體更新

<http://www.twbts.work/2019/05/esp8266-wifi-esp-01.html?m=1>

[4] 溫度感測器 DS18B20

<https://shop.mirotek.com.tw/arduino/arduino-adv-1/>

[5] 強化學習導論六：時間差分學習(TD Learning)

<https://www.getit01.com/p2018020833426502/>

[6] Q-Learning Algorithm: From Explanation to Implementation

<https://towardsdatascience.com/q-learning-algorithm-from-explanation-to-implementation-cdbeda2ea187>

[7] 使用 L298N 模組控制直流馬達

<http://coopermaa2nd.blogspot.com/2012/09/l298n.html>

[8] Open AI Gym 簡介與 Q learning 演算法實作

<https://blog.techbridge.cc/2017/11/04/openai-gym-intro-and-q-learning/>

[9] qlearning基於價值的強化學習算法

<https://blog.csdn.net/weixin_26729375/article/details/109070321>

[10] 強化學習：入門指南

<https://smartauto.ctimes.com.tw/DispArt-tw.asp?O=HK3AT6N81A2ARASTDX>

[11] 探討機器學習與深度學習之差異

<https://www.wpgdadatong.com/tw/blog/detail/40355>

[12] Test Arduino Mega 2560 and ESP8266 (AT Commands)

<https://pdacontrolen.com/prueba-arduino-mega-2560-y-esp8266/>