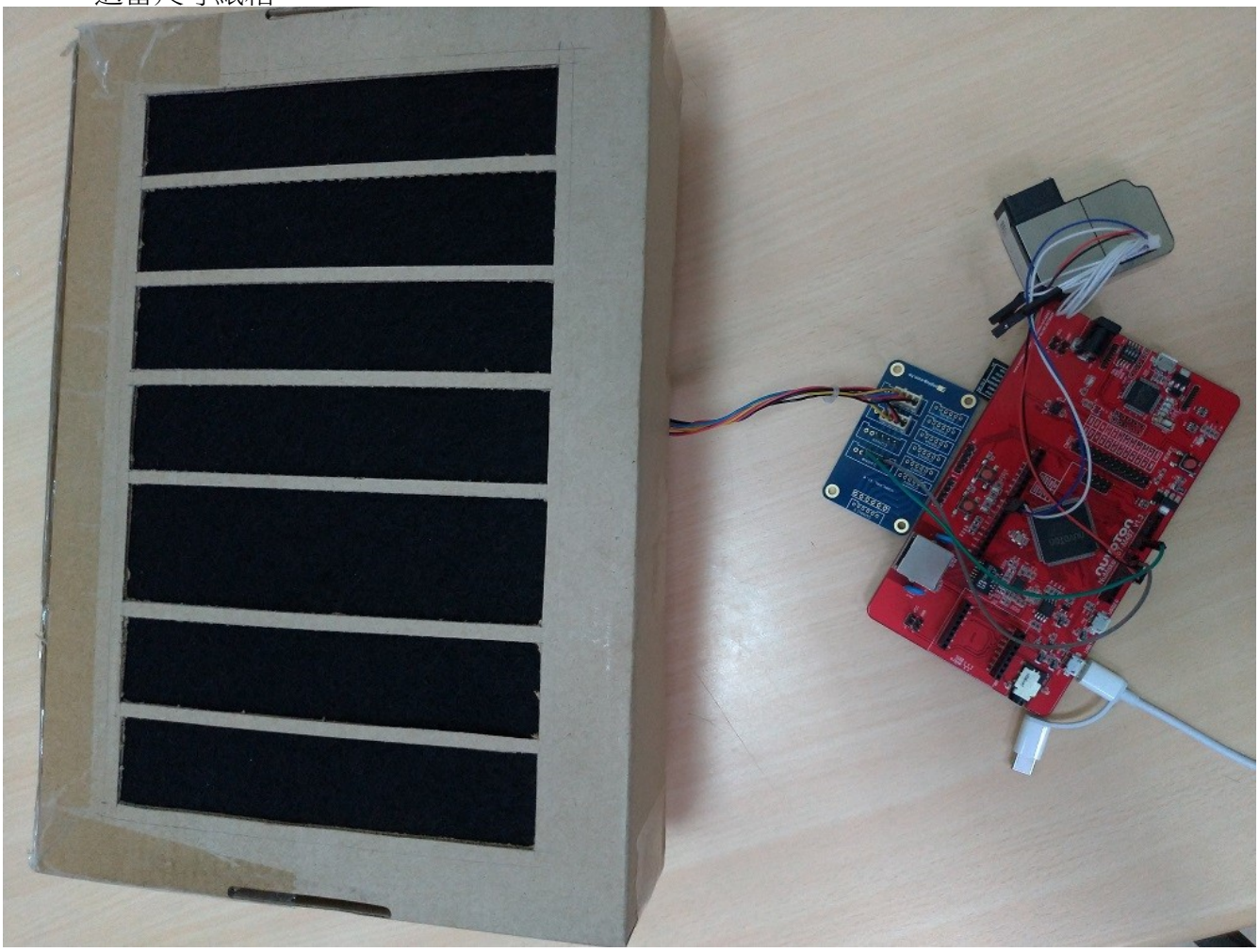
新唐 IoT M487 空氣清淨機

一、教材說明：

本案例使用新唐 IoT M487 開發板，藉由它豐富的輸入輸出，結合雷射粉塵感測器來偵測空氣中的 PM2.5 濃度，並直接由 GPIO 輸出 PWM 來控制風扇轉速，節約電力消耗。最後會上傳空氣品質與風扇轉速的數據到 IDEAS Chain 數據平台。

二、教材材料：

- 新唐開發板 : NuMaker-IoT-M487 V1.3
- 攀藤雷射粉塵感測器 (PMS3003)
- 4 線式節能風扇
- Honeywell HEPA 濾網
- Honeywell 活性碳濾網
- 連接線
- Micro USB 線
- 適當尺寸紙箱

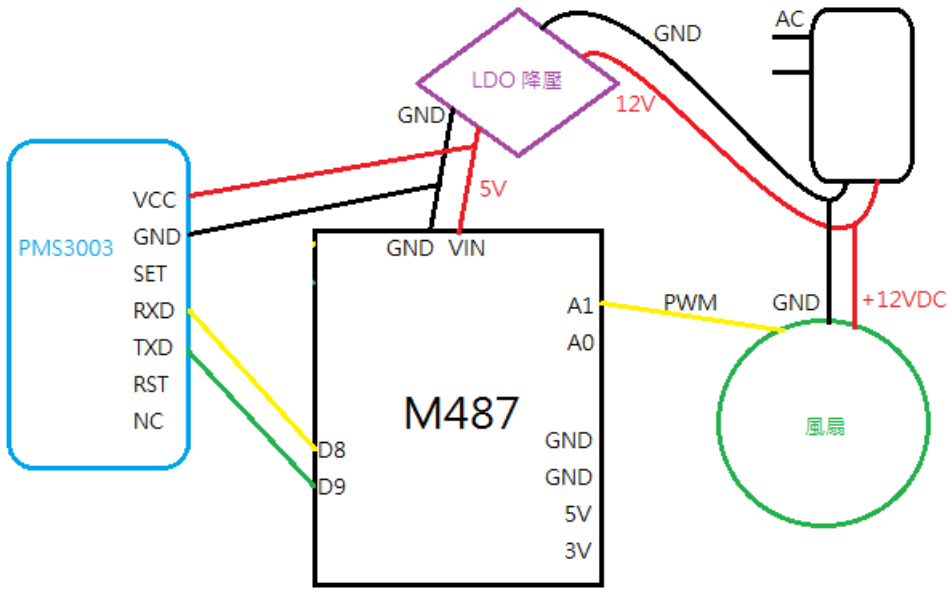


三、接線圖：

IoT M487		攀藤 PMS3003
GND		GND
5V		VCC
D8 (UART5_TXD)		RXD
D9 (UART5_RXD)		TXD

IoT M487		4 線式風扇 (註 1)
GND		GND
-		+12VDC
-		Sense
A1		PWM

註 1 : 風扇的 +12VDC 需另外用 12V 的電源供應器供電，風扇的 GND 與 M497 開發板共地。



四、教材程式：

第 13-14 行：PM2.5 與 PWM 的對應表。例: PM2.5 低於 5 對應 PWM 50， PM2.5 在 5~12 對應 PWM 59，PM2.5 在 12~15 對應 PWM 69 等等等...

第 19-20 行：IDEAS Chain 數據平台網路位址

第 27 行：裝置的存取權杖 (請先到 IDEAS Chain 數據平台建立裝置)

第 33-41 行：函式，輸入 PM2.5 ，傳回對應的 PWM 值。

```
10 #
11 # PM2.5 與 PWM 的對應表
12 #
13 PmTable =  [ 5, 12, 15, 20, 35.5] # PM2.5(µg/m3)
14 FanTable = [50, 59, 69, 79, 89  ] # PWM(%) Range : 50 ~ 99
15
16 #
17 # IDEAS Chain 數據平台位址
18 #
19 HOST = "ideaschain.com.tw"
20 API_URL = "iiot.ideaschain.com.tw"
21 print("HOST =", HOST)
22 print("API_URL =", API_URL)
23
24 #
25 # 裝置的存取權杖 (請先到 IDEAS Chain 數據平台建立裝置)
26 #
27 DEVICE_KEY = "YQMz9z8u6WUoNoFAYjPw"
28 print("DEVICE_KEY =", DEVICE_KEY)
29
30 #
31 # 依據 PM2.5 的數值在 "PmTable" 與 "FanTable" 尋找相對應的 PWM(%) 值。
32 #
33 def getFanPwm(pm2_5):
34     for pi in range(len(PmTable)):
35         if pi == 0:
36             if pm2_5 < PmTable[pi]:
37                 return FanTable[pi]
38         else:
39             if(pm2_5 >= PmTable[pi - 1]) and (pm2_5 < PmTable[pi]):
40                 return FanTable[pi]
41     return 99 # 99%
```

第 46-51 行：LED 閃爍函式，參數 count 是閃爍次數。

第 56,61 行：宣告 PWM 變數，設定脈衝寬度百分比為 0 在 A1 腳位上

第 66-71 行：宣告 LED 變數，做為狀態顯示使用。

```
43 #
44 # LED 閃爍函式，參數 count 是閃爍次數。
45 #
46 def blink(led, count):
47     for i in range(count):
48         led.on()
49         pyb.delay(100) # delay 100 ms
50         led.off()
51         pyb.delay(100) # delay 100 ms
52
53 #
54 # 宣告 PWM 變數
55 #
56 PwmA1 = PWM(1, freq = 2)
57
58 #
59 # 設定脈衝寬度百分比為 0 在 A1 腳位上
60 #
61 PwmA1Ch = PwmA1.channel(mode = PWM.OUTPUT, pulse_width_percent = 0, pin = Pin.board.A1)
62
63 #
64 # 宣告 LED 變數，做為狀態顯示使用。
65 #
66 LedR = LED('led0') # Red LED
67 LedY = LED('led1')
68 LedG = LED('led2')
69 LedR.off()
70 LedY.off()
71 LedG.off()
```

第 73-88 行：連接 WIFI AP，宣告變數 - ‘WlanReady’ 儲存連線狀態。

```
73 WlanReady = True
74 try:
75     #
76     # 連接 WIFI AP
77     #
78     Wlan = network.WLAN()
79     Wlan.connect('WIFI_SSID', 'WIFI_PASSWORD')
80     pyb.delay(2000) # delay 2 seconds
81
82     #
83     # 印出分配到的 IP 位址
84     #
85     print(Wlan.ifconfig())
86 except:
87     WlanReady = False
88     print("Unexpected error while connecting to WIFI AP.")
```

第 93 行：宣告變數，儲存上次傳送資料的時間。用來控制每？秒傳送資料一次。

第 96 行：宣告變數，儲存上次讀取 sensor 的時間。用來控制每？秒讀取 sensor 一次。

第 103-106 行：儲存上次 PM1.0, PM2.5 and PM10 的數值。用來查看數值是否出現變化。

第 111 行：初始化 PMS3003 變數

```
 90 #
 91 # 宣告變數，儲存上次傳送資料的時間。
 92 #
 93 previous_send_time = 0
 94
 95 #
 96 # 宣告變數，儲存上次讀取 sensor 的時間。
 97 #
 98 previous_read_time = 0
 99
100 #
101 # 儲存上次 PM1.0, PM2.5 and PM10 的數值
102 #
103 pre_pm1_0 = 0
104 pre_pm2_5 = 0
105 pre_pm10 = 0
106 data_changed = False
107
108 #
109 # 初始化 PMS3003 變數
110 #
111 Pms = PmsX003() # PMS5003T or PMS3003
112 Pms.begin()
```

第 118-120 行：宣告變數，用來儲存目前的 PM1.0, PM2.5 and PM10 數值。

第 129 行：讀取 snesor - PMS3003

第 130-132 行：取得目前的 PM1.0, PM2.5 and PM10 數值

第 137-144 行：檢查 PM1.0, PM2.5 and PM10 的數值是否有變化，若有變化則把 data_changed 設成 True

```
114 #
115 # 不斷地重覆執行迴圈
116 #
117 while True:
118     pm1_0 = pre_pm1_0
119     pm2_5 = pre_pm2_5
120     pm10 = pre_pm10
121 
122     #
123     # 每 2 秒讀取 sensor 一次
124     #
125     if (utime.ticks_ms() - previous_read_time) >= 2000:
126         #
127         # 讀取 sensor
128         #
129         if(Pms.readAll() == 0): # Success
130             pm1_0 = Pms.getPm1_0()
131             pm2_5 = Pms.getPm2_5()
132             pm10 = Pms.getPm10()
133             print(Pms.getPmsType())
134             print("pm1_0 =", pm1_0)
135             print("pm2_5 =", pm2_5)
136             print("pm10 =", pm10)
137             if pm1_0 != pre_pm1_0 or pm2_5 != pre_pm2_5 or pm10 != pre_pm10:
138                 #
139                 # 比較目前的數值與上次的數值，若有變化，把 data_changed 設成 True。
140                 #
141                 data_changed = True
142                 pre_pm1_0 = pm1_0
143                 pre_pm2_5 = pm2_5
144                 pre_pm10 = pm10
145         else:
146             print("Data format error")
147         previous_read_time = utime.ticks_ms() # 儲存這次讀取 sensor 時間
```

第 157-172 行：依據 PM2.5 的數值來改變 LED 的燈號，綠色 LED 代表空氣品質良好，黃色 LED 代表空氣品質尚可，紅色 LED 代表空氣品質不良。

第 177 行：依據空氣品質來取得相對應的定脈衝寬度百分比(風扇轉速)

第 183 行：設定脈衝寬度百分比為 pwm_percent

```
151         #
152         # 依據 PM2.5 的數值來改變 LED 的燈號
153         # 綠色 LED 代表空氣品質良好
154         # 黃色 LED 代表空氣品質尚可
155         # 紅色 LED 代表空氣品質不良
156         #
157         curr_led = LedG
158         if pm2_5 <= 12:
159             curr_led = LedG
160             LedG.on() # Turn on green LED
161             LedY.off()
162             LedR.off()
163         elif pm2_5 > 12 and pm2_5 <= 55:
164             curr_led = LedY
165             LedG.off()
166             LedY.on() # Turn on yellow LED
167             LedR.off()
168         else:
169             curr_led = LedR
170             LedG.off()
171             LedY.off()
172             LedR.on() # Turn on red LED
173
174         #
175         # 依據空氣品質來取得相對應的定脈衝寬度百分比(風扇轉速)
176         #
177         pwm_percent = getFanPwm(pm2_5)
178         print("FanPwm =", pwm_percent)
179
180         #
181         # 設定脈衝寬度百分比為 pwm_percent
182         #
183         PwmA1Ch = PwmA1.channel(mode = PWM.OUTPUT, pulse_width_percent = pwm_percent, pin = Pin.board.A1)
```

第 192-193 行：產生 JSON 格式的資料字串，準備上傳至 IDEAS Chain 數據平台。

第 199 行：產生 HTTP POST 標頭

第 206-223 行：傳送資料至 IDEAS Chain 數據平台

第 229-234 行：萬一傳送發生錯誤，則把 data_changed 設成 True，下次迴圈會重新傳送。

```
185         #
186         # 每 10 秒上傳一次
187         #
188         if WlanReady == True and (utime.ticks_ms() - previous_send_time) >= 10000:
189             #
190             # 產生 JSON 格式的資料字串
191             #
192             param_data = "{\"PM_1_0\":\"" + str(pm1_0) + "\",\"PM_2_5\":\"" + str(pm2_5) + "\",\"PM_10\":\"" + str(pm10) + "\",\"FAN_PWM\":\"" + s
193             param_lenth = str(len(param_data))
194             print(param_data)
195 
196             #
197             # 產生 HTTP POST 標頭
198             #
199             str_request = "POST /api/v1/" + DEVICE_KEY + "/telemetry HTTP/1.1\r\nContent-Type: application/json\r\nContent-Length: " + param_lenth
200             print(str_request)
201 
202             try:
203                 #
204                 # 開始傳送資料
205                 #
206                 print("Open socket")
207                 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
208                 blink(curr_led, 8) # Blink the LED and delay (8 * 200) ms
209                 curr_led.on()
210                 print("Connecting to server")
211                 addr = socket.getaddrinfo(HOST, 80)[0][-1] # 取得 IDEAS Chain 數據平台的 IP 位址
212                 print("addr =", addr)
213                 s.connect(addr)
214                 blink(curr_led, 8) # Blink the LED and delay (8 * 200) ms
215                 curr_led.on()
216                 print("Send Data")
217                 s.send(str_request)
218                 blink(curr_led, 6) # Blink the LED and delay (6 * 200) ms
219                 curr_led.on()
220                 print("Receive Data")
221                 str_response = s.recv(4096)
222                 s.close() # Close socket
223                 print(str_response)
224 
225                 #
226                 # 存下這次送出資料的時間，單位 : ms
227                 #
228                 previous_send_time = utime.ticks_ms()
229             except:
230                 #
231                 # 傳送資料發生錯誤
232                 #
233                 print("Unexpected Error while sending data to server.")
234                 data_changed = True # Set the flag to resend again
235 
```

五、教材成果演示：

系統上電，開發板上的 3 顆 LED (綠黃紅) 會依照目前的空氣品質狀態發亮。

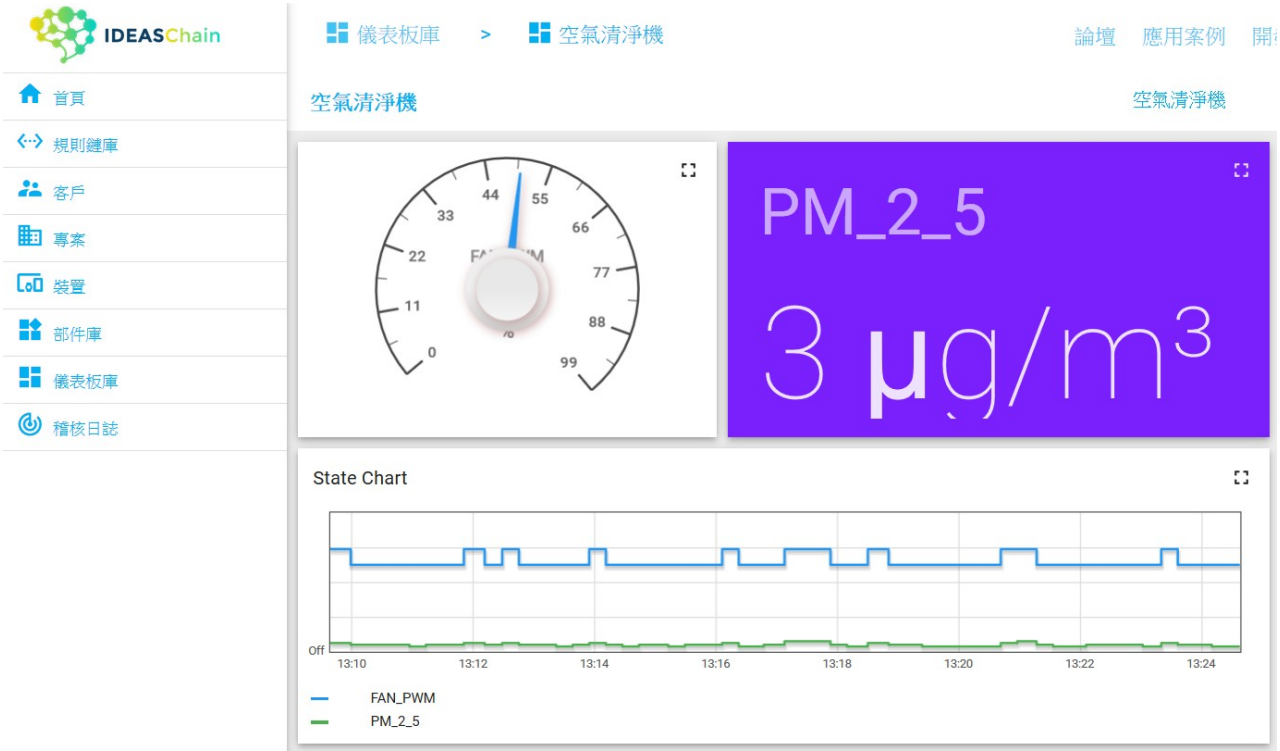
PM2.5 濃度 (μg/m³)	LED
0 to 12	綠 (空氣品質良好)
12 to 55	黃 (空氣品質尚可)
55 to ∞	紅 (空氣品質不良)

程式會每 10 秒左右上傳目前的 PM1.0, PM2.5, PM10 and Fan PWM 數值到 IDEAS Chain 的數據平台上。

程式依據 PM2.5 的數值來調整給風扇的 PWM 輸出，控制風扇轉速， PM2.5 的數值越高，風扇轉速越快。不同的風扇對 PWM 可能會有不同的轉速特性，通常 PWM 設定成 0 也不會完全靜止，因此需要通過測試來找出一組較適當的數值。

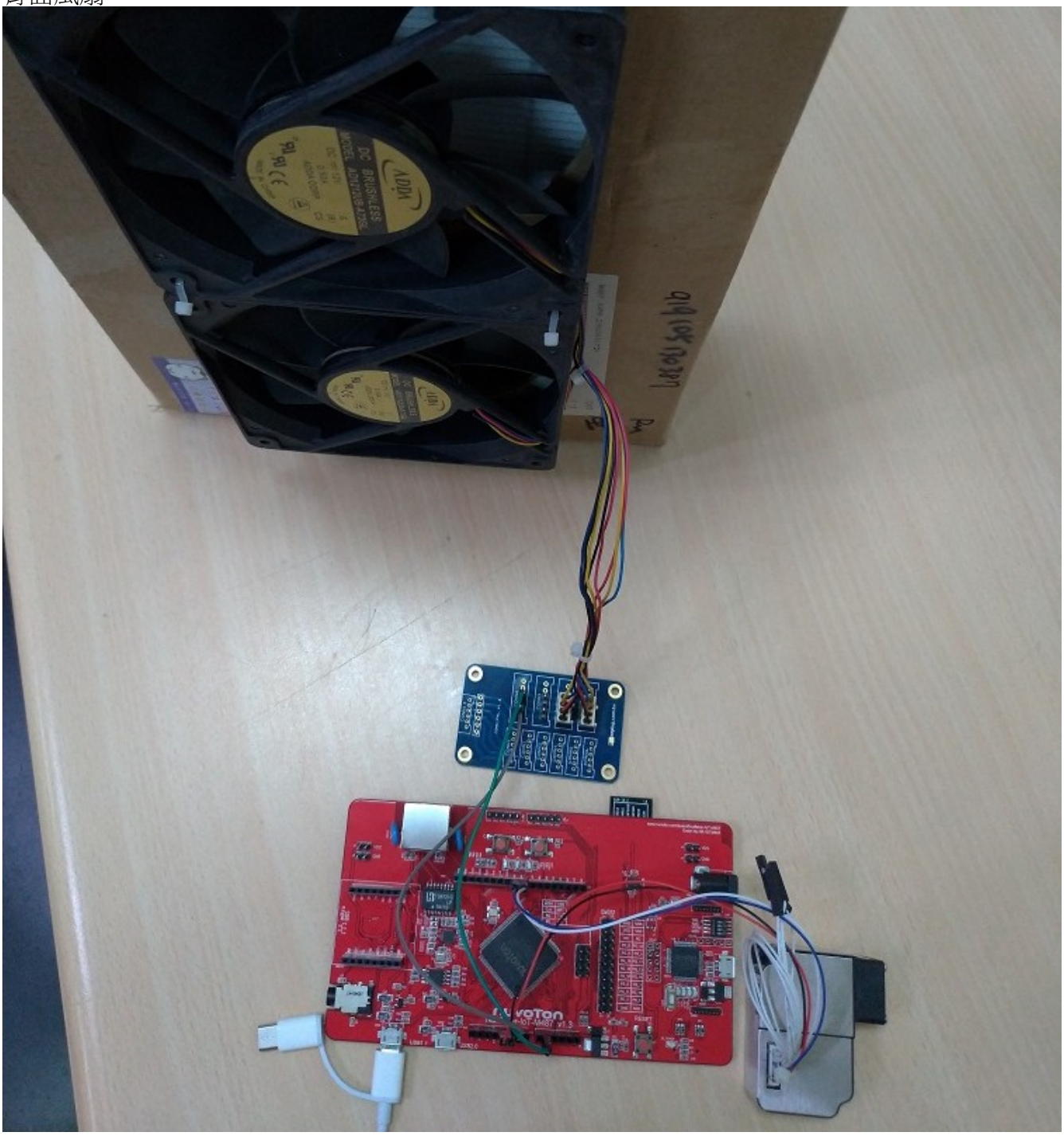
正常狀態下， PM2.5 數值短時間不會有太大變化，可以點 1 柱香放在 PMS3003 前來誘發 PM2.5 數值提高，觀察風扇轉速變化。

下圖是使用 IDEAS Chain 數據平台來顯示數值

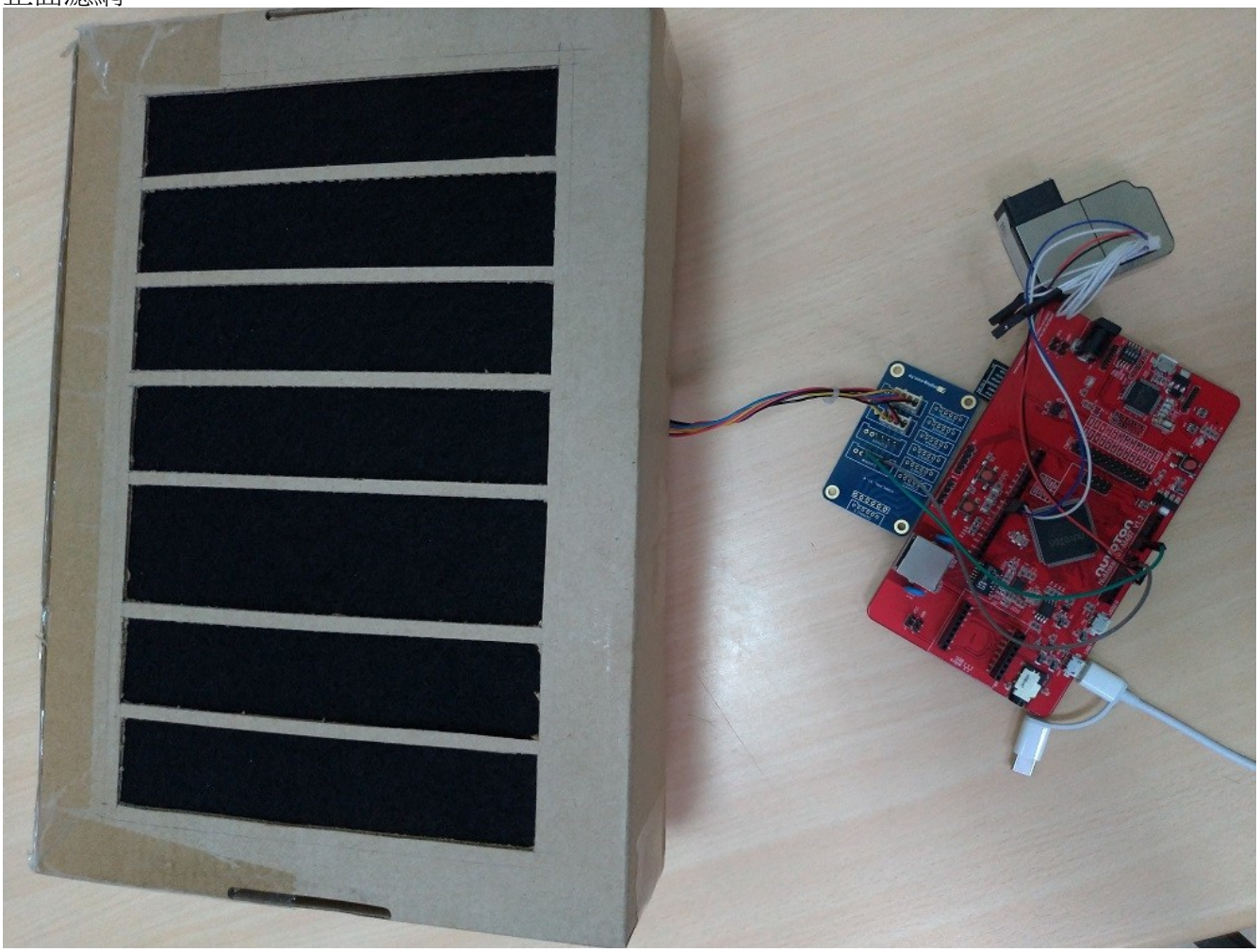


六、電路組裝完成圖片：

背面風扇



正面濾網



濾箱內部第一層活性炭濾芯

第二層 HEPA 濾芯

使用洗車用海綿固定，風扇兩側的橡膠條在合上時會把濾網壓緊，務必密合，過濾效果才會好。