

新唐 IoT M487 人流偵測

一、教材說明：

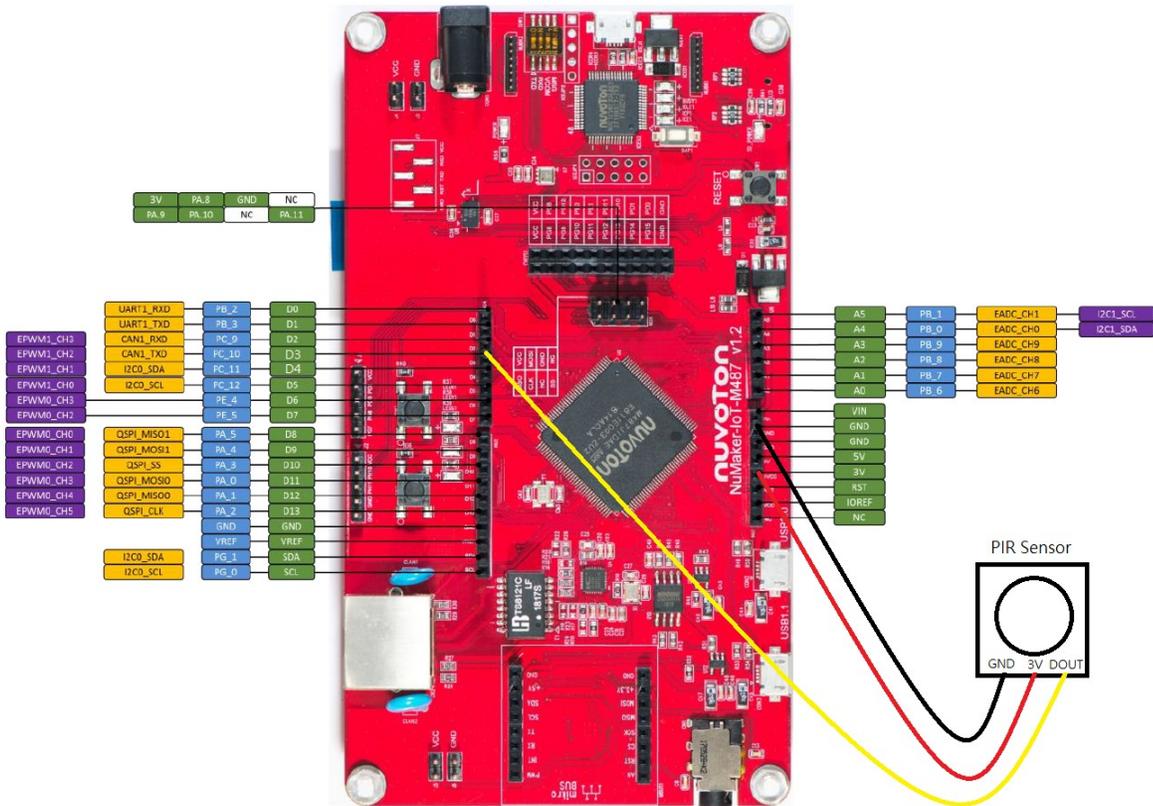
本教材使用新唐 IoT M487 開發板與人體紅外線感測器 (PIR Sensor) 來做人流的偵測。每當有人經過 PIR 前面時，PIR 偵測到溫度變化，會把 DOUT 腳的電位拉高，持續數秒。我們可從 M487 的數位輸入腳來偵測 DOUT 電位變化，若被拉高馬上透過 HTTP 送出資料給 IDEAS Chain 的數據平台。這一筆筆的資料累積在資料庫中，我們可以使用數據平台的工具來存取資料庫，分析人流變化。

二、教材材料：

- 新唐開發板:NuMaker-IoT-M487 V1.3
- 數位式人體紅外線感測器
- 杜邦線 3 條 - 黑、紅、黃
- Micro USB 線

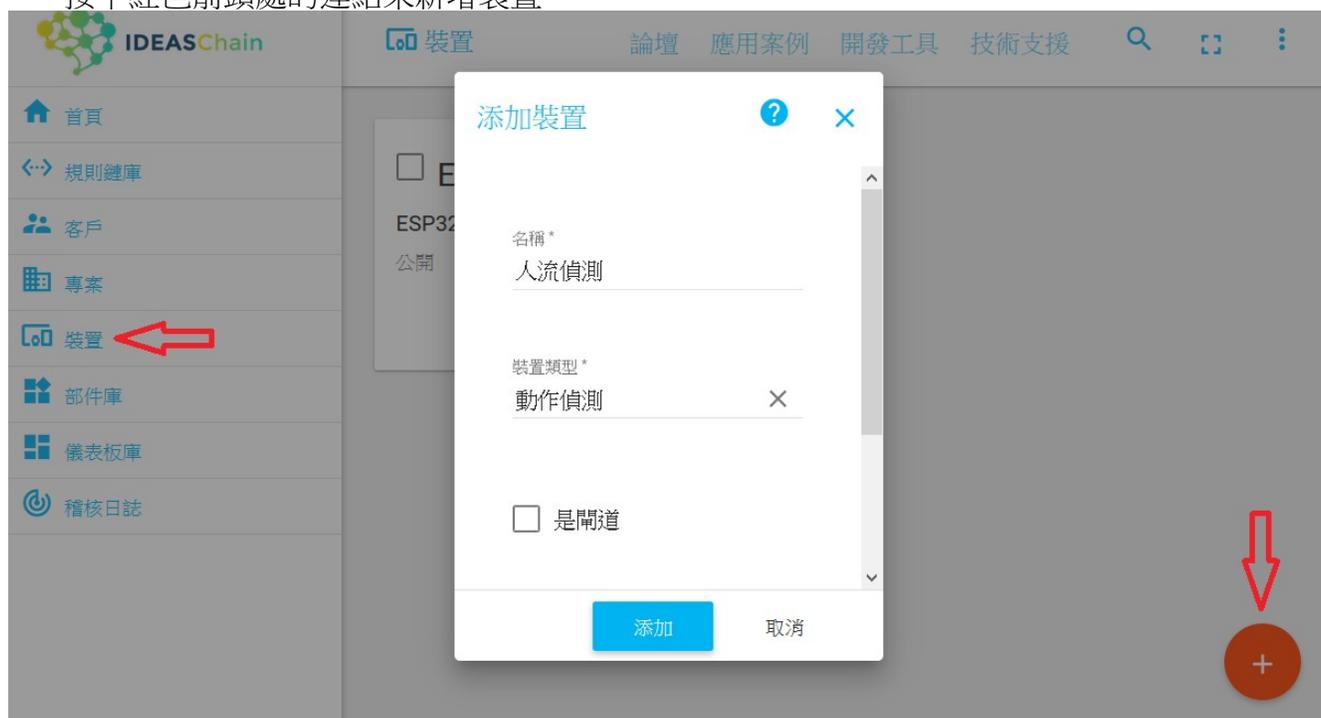
三、接線圖：

IoT M487	杜邦線	PIR Sensor
GND	黑	GND
3V	紅	VCC
D3	黃	DOUT

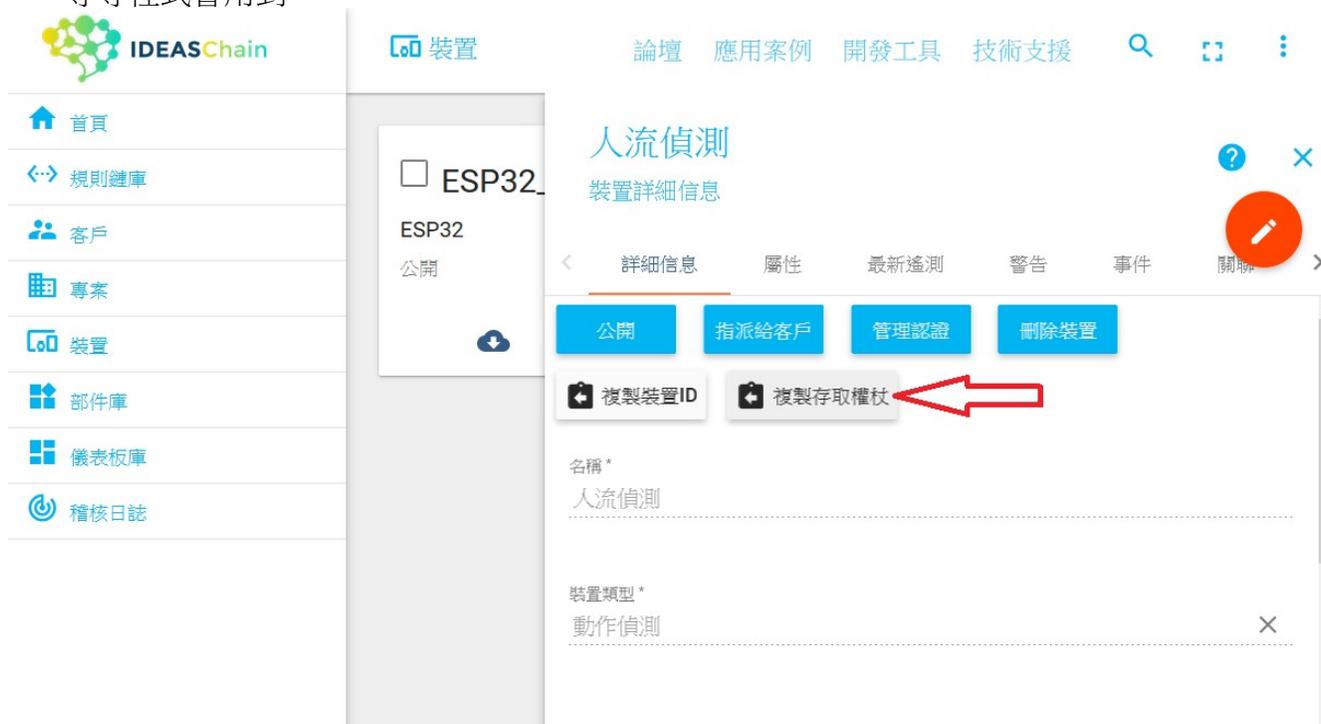


四、在 IDEAS Chain 數據平台上新增裝置：

按下紅色箭頭處的連結來新增裝置



新增完裝置後，進入裝置設定，按下複製存取權杖按鈕。請存下此英數字組成的字串備用，等等程式會用到。



五、教材程式：

第 1-6 行：導入必要的函式庫。

第 12 行：為開發板的 D3 腳位建立 Pin 物件，模式是輸入。

第 17 行：為 'led2' (綠色) 建立 LED 物件，做為狀態顯示使用。

第 22-23 行：IDEAS Chain 數據平台網路位址。

第 30 行：裝置的存取權杖 (請先到 IDEAS Chain 數據平台建立裝置)。

```
1 import pyb
2 import utime
3 from pyb import Pin
4 from pyb import LED
5 import network
6 import usocket as socket
7
8 #
9 # PIR 人體感測器接到 D3
10 # 為開發板的 D3 腳位建立 Pin 物件，模式是輸入。
11 #
12 get_in = Pin(Pin.board.D3, Pin.IN)
13
14 #
15 # 為 'led2' 建立 LED 物件，做為狀態顯示使用。
16 #
17 led_green = LED('led2')
18
19 #
20 # IDEAS Chain 數據平台位址
21 #
22 HOST = "ideaschain.com.tw"
23 API_URL = "iiot.ideaschain.com.tw"
24 print(HOST)
25 print(API_URL)
26
27 #
28 # 裝置的存取權杖 (請先到 IDEAS Chain 數據平台建立裝置)
29 #
30 DEVICE_KEY = "9b0T24hroG2zS2L56314"
31 print(DEVICE_KEY)
32
```

第 36 行：建立 WIFI 網路物件。

第 37 行：連接 WIFI AP，請修改 SSID 與 PASSWORD 以符合您的無線網路環境。

第 43 行：印出從 WIFI AP 分配到的 IP 位址。

第 48-49 行：產生準備傳送給數據平台的資料字串，格式是 JSON。

第 55 行：取得 IDEAS Chain 數據平台的 IP 位址，傳送資料時會到。

第 61 行：宣告變數，儲存上次傳送資料的時間。用來控制不要太密集傳送資料。

第 66-71 行：綠色 LED 閃爍函式，參數 count 是閃爍次數。

```
33 #
34 # 連接 WIFI AP
35 #
36 wlan = network.WLAN()
37 wlan.connect('YOUR_WIFI_SSID', 'YOUR_WIFI_PASSWORD')
38 pyb.delay(2000) # delay 2 seconds
39
40 #
41 # 印出分配到的 IP 位址
42 #
43 wlan.ifconfig()
44
45 #
46 # 產生 JSON 格式的資料字串
47 #
48 param_data = "{\"PIR\": \"1\"}"
49 param_lenth = str(len(param_data))
50 print(param_data)
51
52 #
53 # 取得 IDEAS Chain 數據平台的 IP 位址
54 #
55 addr = socket.getaddrinfo(HOST, 80)[0][-1]
56 print(addr)
57
58 #
59 # 宣告變數，儲存上次傳送資料的時間。用來控制不要太密集傳送資料。
60 #
61 previous_send_time = 0
62
63 #
64 # 綠色 LED 閃爍函式，參數 count 是閃爍次數。
65 #
66 def blink(count):
67     for i in range(count):
68         led_green.on()
69         pyb.delay(100) # delay 100 ms
70         led_green.off()
71         pyb.delay(100) # delay 100 ms
```

第 76 行：while 迴圈開始，不斷地重覆執行。

第 86-88 行：檢查 PIR DOUT 腳位狀態，若 PIR 感測器沒被觸發，則等待 1 秒，然後跳過繼續下個迴圈。

第 93-96 行：若 PIR 感測器被觸發，但是距離上次傳送資料不到 5 秒，則等待 1 秒，然後跳過繼續下個迴圈。防止短時間內傳送太多資料。

第 104 行：產生 HTTP POST 標頭，準備傳送給數據平台。

```
72
73 #
74 # while 迴圈，不斷地重覆執行。
75 #
76 while True:
77
78     #
79     # 印出 PIR 感測器的狀態
80     #
81     print('PIR Status : ', get_in.value())
82
83     #
84     # 若 PIR 感測器沒被觸發，則等待1秒，然後跳過繼續下個迴圈。
85     #
86     if get_in.value() == 0:
87         pyb.delay(1000) # delay 1 second
88         continue
89
90     #
91     # 若 PIR 感測器被觸發，但是距離上次傳送資料不到5秒，則等待1秒，然後跳過繼續下個迴圈。
92     #
93     if (utime.ticks_ms() - previous_send_time) < 5000:
94         print('Too close to previous sending, filter it.')
95         pyb.delay(1000) # delay 1 second
96         continue
97
98     print('\n')
99     print('=== PIR is Triggered ===')
100
101     #
102     # 產生 HTTP POST 標頭
103     #
104     str_request = "POST /api/v1/" + DEVICE_KEY + "/telemetry HTTP/1.1\r\nContent-Type: appl
105     API_URL + "\r\n\r\n" + param_data
106     print(str_request)
```

第 111 行：宣告 Socket 物件。

第 113 行：閃爍並等待 1.6 秒。

第 116 行：連接數據平台。

第 118 行：閃爍並等待 1.6 秒。

第 121 行：傳送資料給數據平台。

第 123 行：閃爍並等待 1.2 秒。

第 126 行：接收從數據平台傳回的資料。

第 135 行：存下這次送出資料的時間，單位：ms。

```
---
107     #
108     # 開始傳送資料
109     #
110     print("Open socket")
111     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
112
113     blink(8) # Blink green LED and delay (8 * 200) ms
114
115     print("Connecting to server")
116     s.connect(addr)
117
118     blink(8) # Blink green LED and delay (8 * 200) ms
119
120     print("Send Data")
121     s.send(str_request)
122
123     blink(6) # Blink green LED and delay (6 * 200) ms
124
125     print("Receive Data")
126     str_response = s.recv(4096)
127
128     s.close() # Close socket
129
130     print(str_response)
131
132     #
133     # 存下這次送出資料的時間，單位：ms
134     #
135     previous_send_time = utime.ticks_ms()
136
```

六、教材成果演示：

上傳程式後，斷電重新上電，並開啟 PuTTY 接收 Serial Port 訊息，沒問題的話 PuTTY 會不斷地顯示 PIR 的狀態。在 PIR 前揮手，程式被觸發，開發板上的 LED2 (綠色)會開始閃爍，代表正在傳送資料。數據平台上的**裝置-最新遙測**會出現一筆資料。



The screenshot shows the IDEASChain IoT dashboard. On the left is a navigation menu with options like 首頁, 規則鏈庫, 客戶, 專案, 裝置, 部件庫, 儀表板庫, and 稽核日誌. The main content area is titled '裝置' and shows details for an 'ESP32' device. A red arrow points to the '最新遙測' (Latest Telemetry) tab. Below this, a table displays the latest telemetry data:

最後更新時間	鍵	值
2020-05-27 10:47:58	PIR	1

At the bottom of the table, there are pagination controls: Page: 1, Rows per page: 5, 1 - 1 of 1.

使用圖表觀察人流狀況，表中的立棒代表偵測到人經過。



The screenshot shows the IDEASChain IoT dashboard with a chart titled '人流偵測圖表' (Person Flow Detection Chart). The chart is a 'Timeseries Bars - Flot' plot showing the 'PIR' sensor data over the last 30 minutes. The x-axis represents time from 12:50 to 13:15, and the y-axis represents the PIR value from 0 to 2. The chart shows several vertical bars representing PIR detections. A red arrow points to the chart area. At the bottom right of the chart, there is a legend for 'PIR' and a '平均值' (Average) of 1.

七、電路組裝完成圖片：

