

2021.7.9



DSI2598+ 線上工作坊



關於黑趣

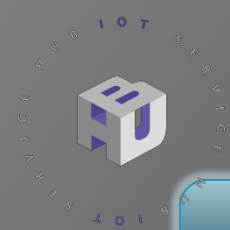
黑趣活動

黑趣爆導

物聯網智造基地

A detailed scene of miniature figures working on a green printed circuit board (PCB). The figures are dressed in various work clothes: a yellow shirt and green pants, a green shirt and grey pants, a blue shirt and red cap, and a grey shirt. They are surrounded by numerous black integrated circuits (chips) mounted on the board. One figure in the foreground is pushing a red wheelbarrow. The background is a dark teal gradient with a faint city skyline.

ideas Hatch
智造創業加速器



DSI2598+ 線上工作坊

益師傅



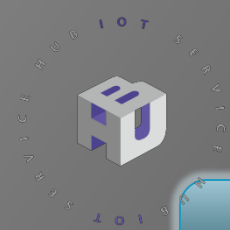
MQTT 與 IoT 整合運用
公開社團

E-mai: pos_soft_king@yahoo.com.tw

程式與資料下載:

<https://pse.is/3jt4ad>





NB-IoT 說明

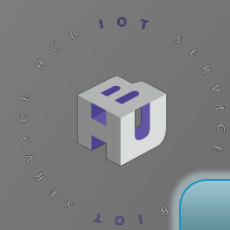
根據IoT應用場景不同，物聯網的需求可以分為**高速率**、**中速率**、**低功率**耗覆蓋三層，而**NB-IoT**在的**LPWAN**層級中又具備覆蓋 **90%**的佔有能力。

LPWAN（Low Power WideArea Network）：低功耗廣域網絡，就是專為低功耗、遠距離、數量多連接的物聯網應用而設計。

短距離無線網絡包含**WIFI**、**藍牙**、**ZigBee**等多種技術。

LPWAN也包含多種技術，如**LoRa**、**Sigfox**、**NB-IoT**等。

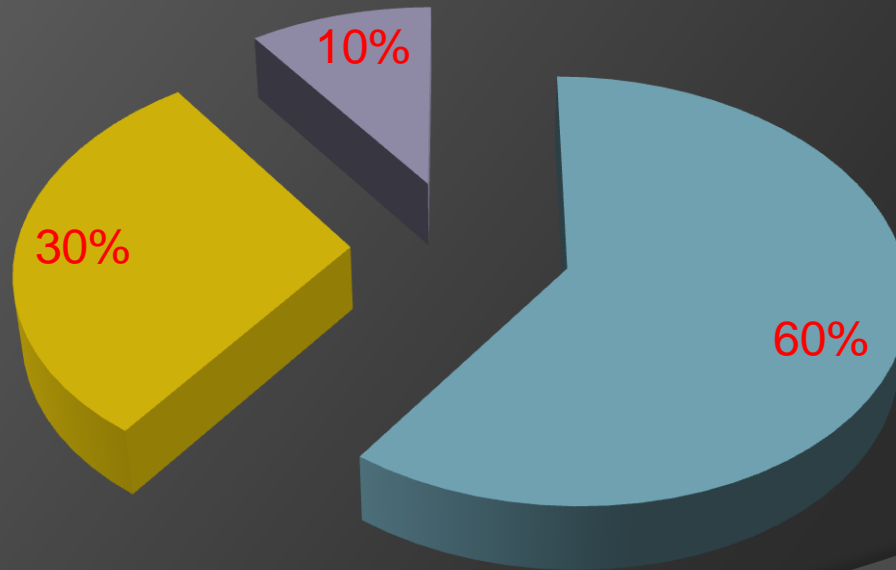
目前 **LPWAN** 技術可被分為授權頻段(**NB-IoT**)的廣域網技術及非授權頻段(**Sigfox** **LoRa**)的廣域網技術兩類，不同的 **LPWAN** 技術在接入網絡、部署方式、技術特點、功耗性能及服務模式上都有所差異。



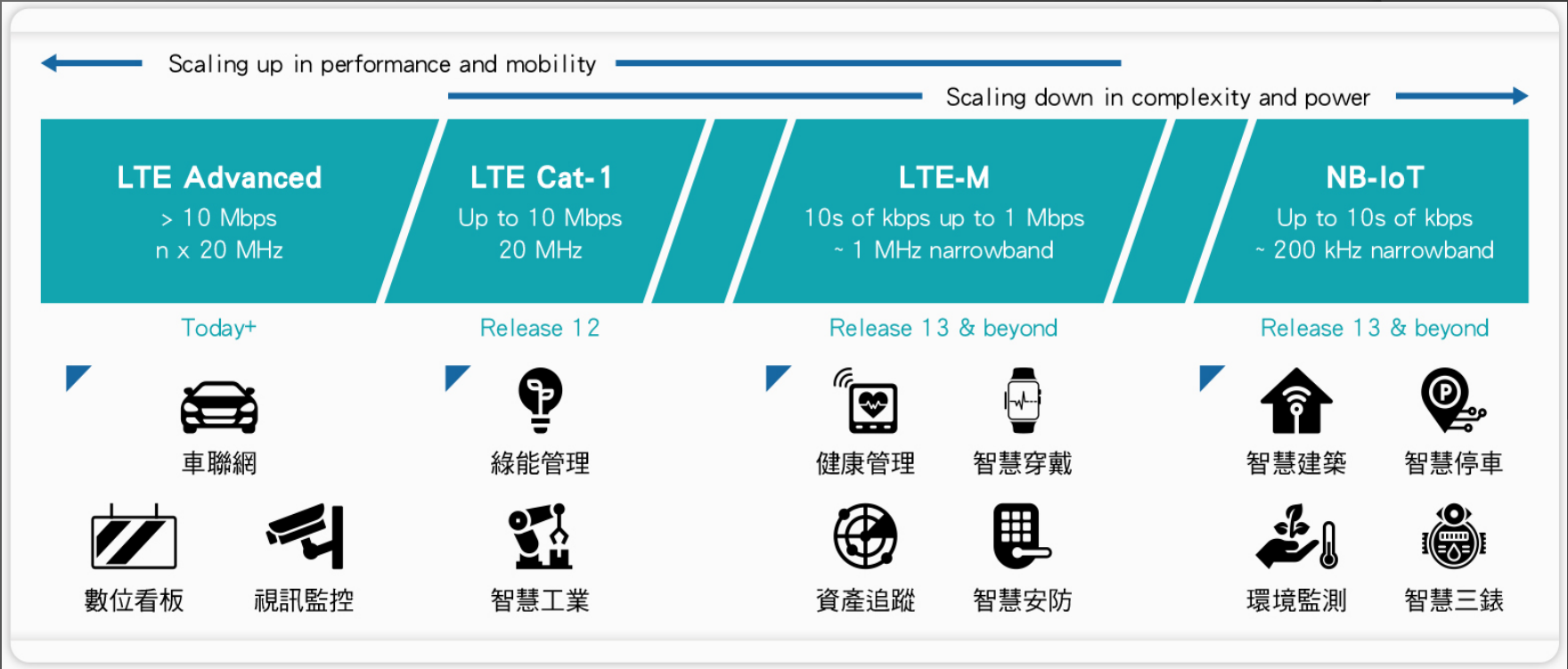
NB-IoT 說明

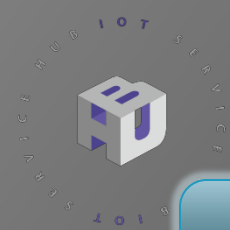
IoT速率運用占比

■ LPWAN ■ 中速率 ■ 高速率



NB-IoT 說明





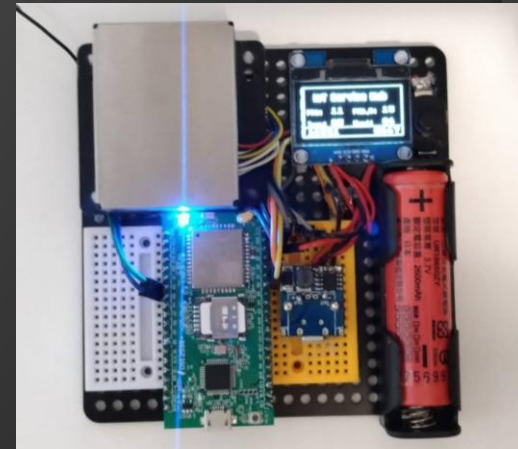
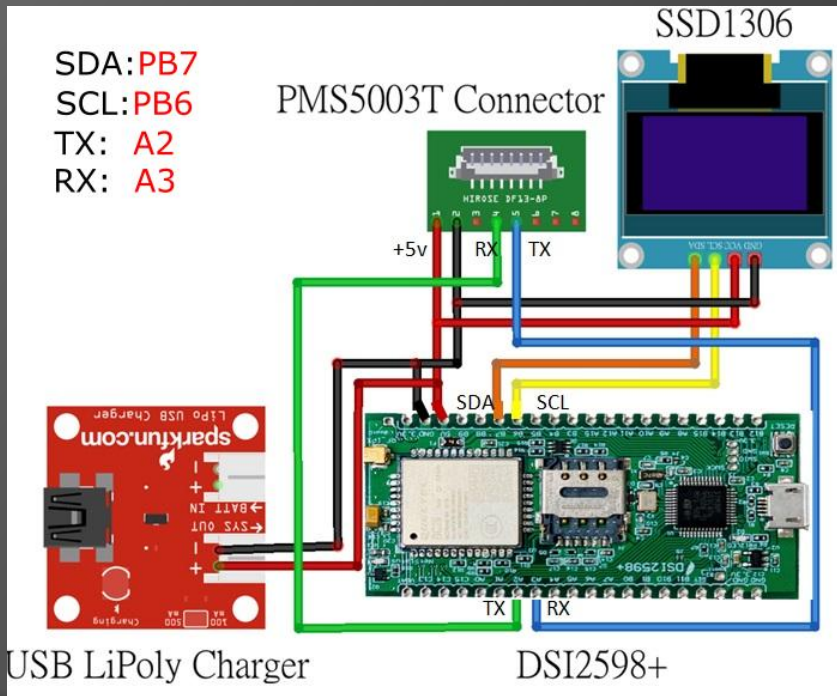
NB-IoT 說明

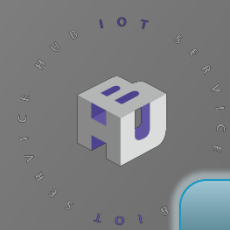
NB-IoT 與 WiFi 之差異：

	NB-IoT	WiFi
上網方式	NB-IoT 利用電信基地台連上網際網路	透過無線基地台連上網際網路
傳遞資料量	適用小資料量傳輸	適用傳輸大量資料的訊息
連線距離	由於全台基地台涵蓋率夠高，幾乎無死角	連接無線基地台的距離較短
耗損功率	採用低功率晶片，使用一般 AA 電池可達 3-5 年以上	晶片耗用功率較高
IP位址	使用的電信基地台提供的IP 位址大都為 虛擬 IP 網段	連網較易取得真實IP 位址

DSI2598+ 實作案例

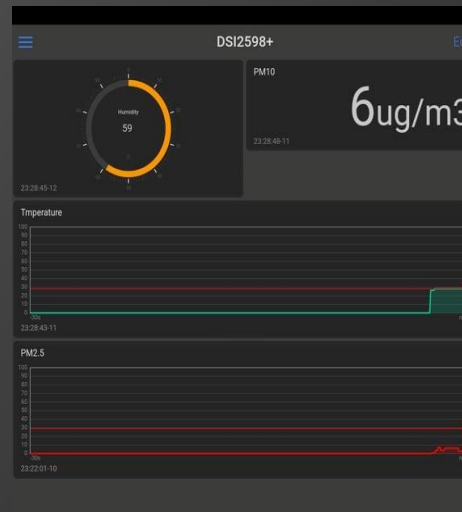
Particle detection

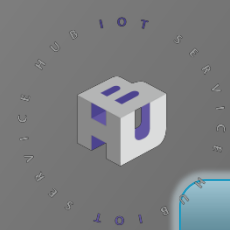




DSI2598+ 實作案例

MQTT Data Dashboard





DSI2598+ 實作案例

Emergency call system

前言:

由於台灣已邁向老人社會,有些老人習慣獨自居住生活,但難免生活上總是會有些突發狀況或是擔心有陌生人入侵狀況.

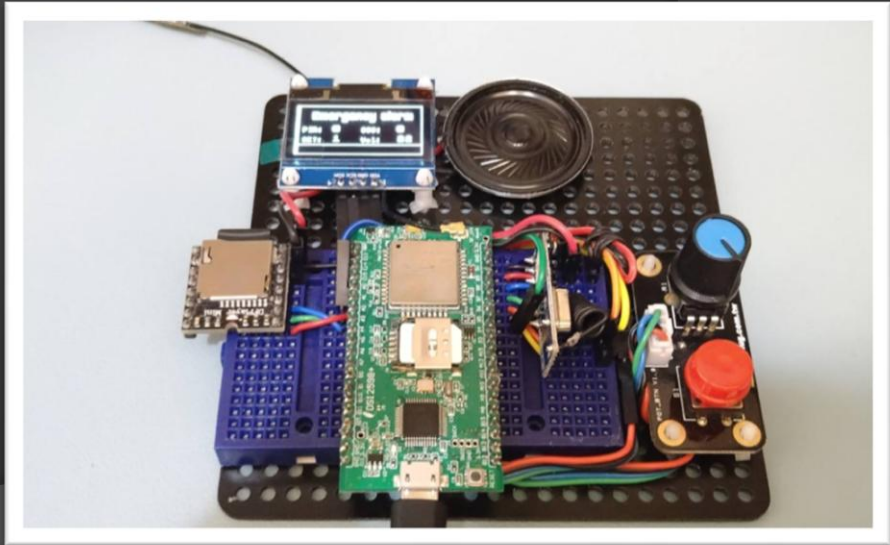
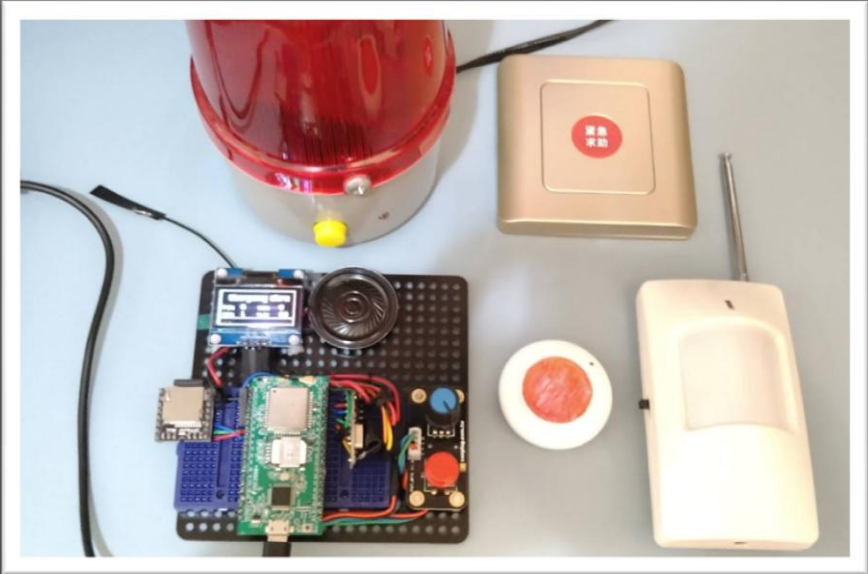
所以便能透過 **NB-LoT** 以 **MQTT** 傳遞少量資料到雲端 (呼叫/入侵偵測),讓其他家人知道長輩的狀況. 因為也是只有傳遞緊急狀況資訊,所以 **NB-LoT** 的運用機制就會比 **wifi** 的運用更合適,不用每個月花費過多的資訊費用.

系統功能:

- 1.可以根據哪個按鈕被按下會進行呼叫功能.會透過 **MQTT** 傳送資料到數據平台, 緊急求救系統也會發出呼救警報聲音.
- 2.透過系統上警戒啟動或解除按鈕,可以偵測異常入侵 (**PIR** 偵測),偵測異常入侵會發出警報聲響,會透過 **MQTT** 傳送資料到數據平台.
- 3.遠端 **MQTT** 警示燈 會閃爍,智能手錶也會震動並發出聲響,會說: "家人緊急呼救" / "有人入侵" 的警語.
- 4.系統上的旋鈕可以控制音量大小聲.

DSI2598+ 實作案例

Emergency call system



DSI2598+ 實作案例

Emergency call system

	第一代	第二代
開發板	DSI5168	DSI2598+
通訊方式	wifi	NB-IoT
數據平台	IDEASChain	New IDEASChain
無線按鈕	RF 315M	RF 315M
MQTT支援	無	有
顯示支援	無	1.3 吋 OLED
入侵偵測	無	有

作品圖示





物聯網智造基地

I O T S E R V I C E H U B

國產IC開發板系列

量產導向的物聯網開發板解決方案

DSI 2598+ 說明

DSI2598+ 介紹

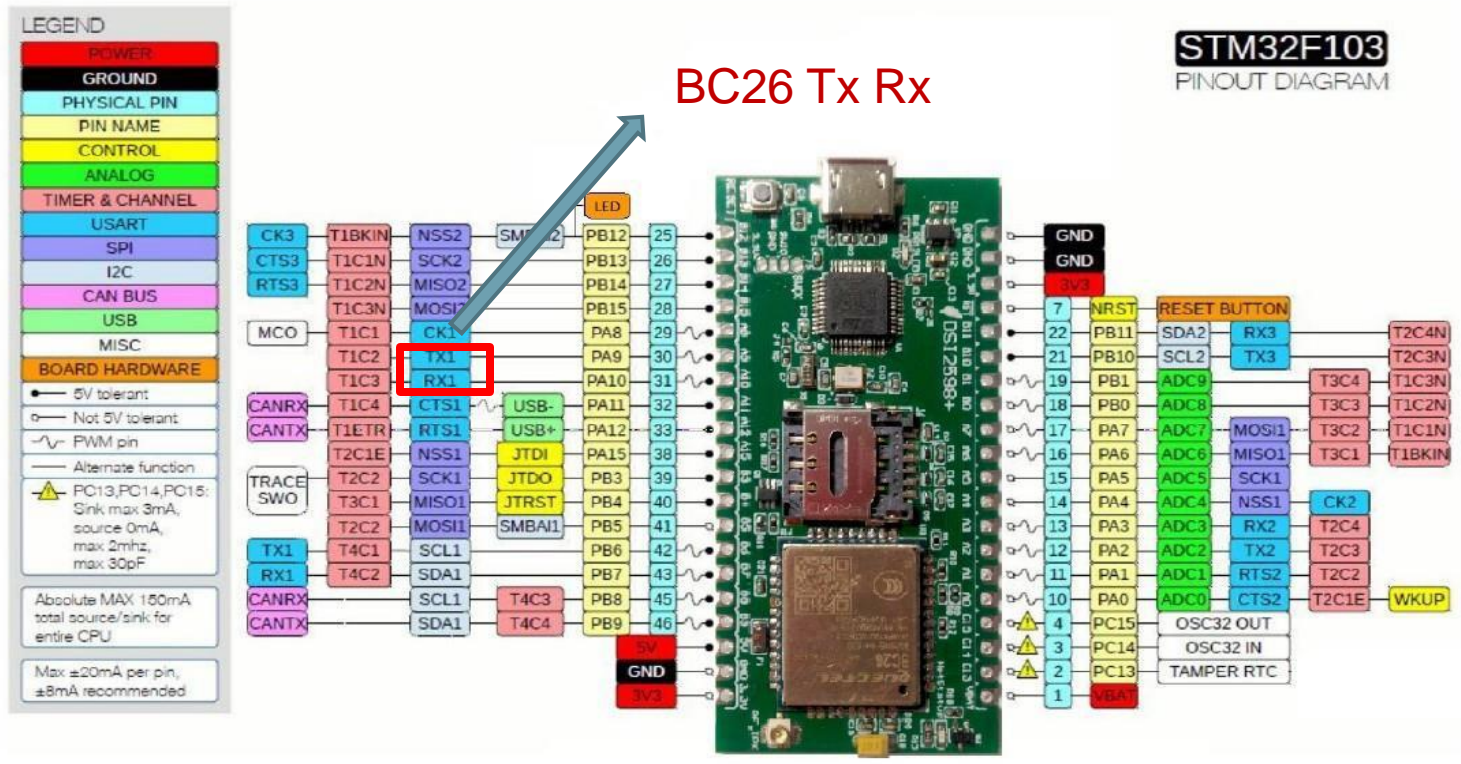
硬體功能	第一代	第二代	第三代
NB-IoT 晶片	MT2625	MT2625	MT2625
MCU	ATMEGA328	ATMEGA328	STM32F103C8T6
I/O	16	16	37
EEPROM	1KB	1KB	可用flash模擬
SPRAM	2KB	2KB	20KB
Flash Memory	32KB	32KB	64KB
ADC	3	2	10
SPI	1	1	2
UART	1	1	3
I2C	1	1	2
PWM	5	2	15



DSI2598+ 介紹

DSI2598+ Arduino 開發環境

接腳及硬體功能說明:

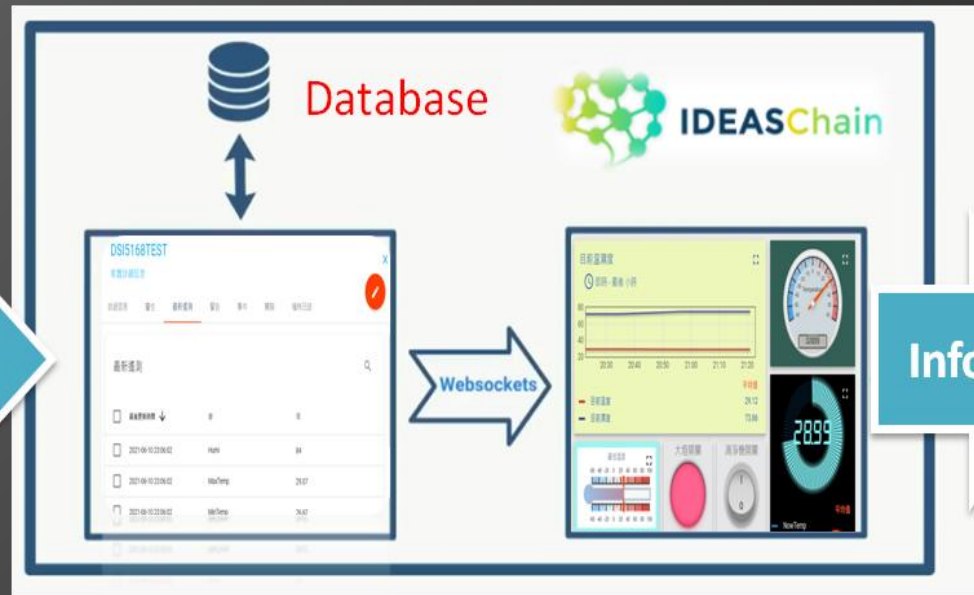


IDEASChain 介紹

Sensor readings



MQTT
HTTP



user





MQTT解析說明

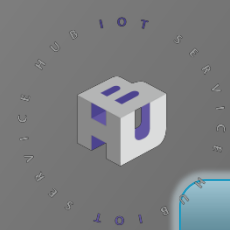
MQTT是一種基於「發布 **publish** / 訂閱 **subscribe**」機制的訊息傳輸協定，我們可以把它想成 **YouTube** 影片發行和訂閱的機制。

MQTT 訊息發送端，相當於 **YouTuber** 拍攝影片上傳到 **YouTube** 後，若有開啟訂閱小鈴鐺時，便會收到發片通知可以讓訂閱者觀看，而 **YouTube** 角色就如同代理人（**broker**），來統籌管理發行和訂閱事宜。
每一個訊息來源（影片）都有個唯一的主題名稱（哪個 **YouTuber** 的哪個影片）。

代理人是個伺服器軟體(我們這裡都採用 **broker.emqx.io**)，向伺服器發送主題的一方是發布者（**publisher**），從伺服器獲取主題的一方則是訂閱者（**subscriber**）。

傳送感測器資料的一端是所謂的發佈者(**publish**)，接收感測器資料的另一端則是訂閱者(**subscribe**)。每個 **Sensor/MCU** 的訊息都需要有個主題（**Topic**）名稱以方便識別。

MQTT的主題（**Topic**）：**MQTT**主題名稱可以是 **UTF-8**（萬國碼）編碼的字串，我們可以自行決定主題名稱，舉例來說：傳送溫度的訊息主題可命名成{**Temp**}、傳送濕度的訊息主題叫做{**Humi**}、傳送亮度的訊息主題叫做{**Lux**}...等等。
主題名稱也支援類似檔案路徑的階層式命名方式，例如：**wanhua/room1/temp**



MQTT 整合運用

安裝 MQTTlens

chrome 線上應用程式商店 -> MQTTLens

參考資料:

<https://swf.com.tw/?p=1009>

MQTTLens

MQTT LENS

Connections +

- broker.emqx.io
- Local
- IDEASChain
- Mobile

Add a new Connection

Connection Details

broker.emqx.io

connection name

Connection color scheme

Hostname Port

Client ID

Session Clean Session Automatic Connection Automatic Connection Keep Alive seconds

Credentials

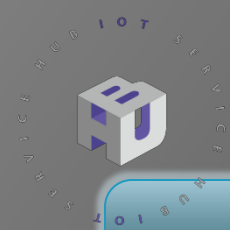
Username Password

version 0.0.14

most once

Retained

Messages: 0/2



MQTT Lens

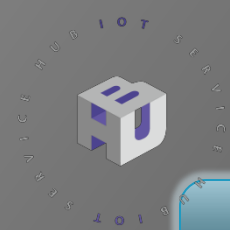
The screenshot displays the MQTT Lens interface with the following elements:

- Connections:** A sidebar on the left shows two connections: `broker.emqx.io` (highlighted with a yellow box) and `IDEASChain`.
- Connection: broker.emqx.io:** The main panel shows the configuration for this connection. The `Subscribe` field contains the topic `wanhua/room1/temp` (highlighted with a red box). The `PUBLISH` button is also highlighted with a blue box.
- Message:** A message with the value `26.5` is displayed below the subscription field.
- Subscriptions:** A section titled "Subscriptions" shows the topic `"wanhua/room1/temp"` with a message count of `0/1`. Below this is a table of messages:

#	Time	Topic	QoS
0	12:31:02	wanhua/room1/temp	0

Below the table, the message content is shown as `Message: 26.5`.

At the bottom of the interface, the `IDEASChain` connection is visible with its `Subscribe` field.



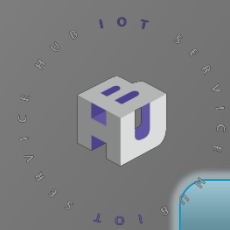
JSON整合運用

JSON 解析與練習

Node-RED 的核心就是 **Node.js** , 而他實現 **IoT** 就是用 **MQTT**, 各位若之前沒接觸過 **Node-RED** 而先接觸 **MQTT** 的人, 就會發現 其實 **Node-RED** 裡頭就是 **JSON** 結構, 用的都是 **payload** 字眼.

所以建議各位 一定先清楚 **JSON** 結構再來使用他, 會覺得很輕鬆上手 .

各位可以多試試下面的連結, 來驗證自己是否了解



JSON整合運用

JSON 解析與練習

參考資料:

[http://JSON 教學](http://JSON教學)

工具

<http://jsonviewer.stack.hu/>

練習

<https://reurl.cc/Dvq0zR>

arduino 環境設定

20210709_classroom | Arduino 1.8.13
檔案 編輯 草稿碼 工具 說明

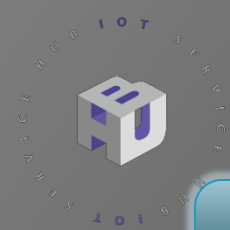
```
1 #include "BC26Init.h"  
2 #include <U8g2lib.h>  
3 #include <RTClock.h>  
4 #include <TimeLib.h>  
5  
6 #define DELAYTIME 720  
7 //-----  
8 String mqttServer  
9 String mqttPort  
10 String clientId  
11 String username  
12 String password
```

草稿碼使用了 30604 bytes (24%) 的全域變數使用了 5000 bytes (24%) 的動態記憶體，剩餘 15400 bytes 給區域變數。上限為 20480 bytes。

參考步驟: <https://pse.is/3jnj8h>

填入:

http://dan.drown.org/stm32duino/package_STM32duino_index.json



程式解說

JSON Serialize

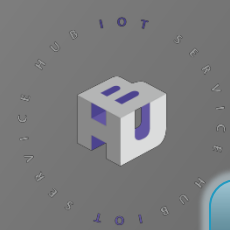
IDEASChain MQTT 傳送

Sleep Mode

程式解說

IDEASChain MQTT Topic

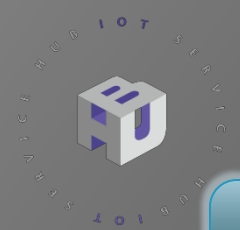
```
7 //-----  
8 String mqttServer      = "iiot.ideaschain.com.tw"; // new ideaschain dashboard MQTT server  
9 String mqttPort        = "1883";  
10 String clientId        = "DSI2598TEST";           // MQTT client ID. it's better to use uni  
11 String username        = "fasrSjaAogrg6QU02JeQ"; // device access token (存取權杖)  
12 String password        = "";                       // no need password  
13 String subscribeTopic  = "v1/devices/me/telemetry";  
14 String publishTopic    = "v1/devices/me/telemetry"; // Fixed topic. ***DO NOT MODIFY***  
15 String subscribeTopicRPC = "v1/devices/me/rpc/request/+"; // RPC MQTT  
16 String publishPayload  = "{\"ForeheadTemp\":\"30\"}"; // String of stringified JSO  
17  
18 #define MQTT_RECONNECT_INTERVAL 100                // millisecond
```

程式解說

Serialize

```
88 Buff = "";  
89 Buff += pm25 ;  
90 json_doc["PM25"] = Buff;  
91  
92 Buff = "";  
93 Buff += pm1 ;  
94 json_doc["PM1"] = Buff;  
95  
96 connect_MQTT(mqttServer, mqttPort, username, password); //開啟 MQTT 連線  
97 serializeJson(json_doc, json_output);  
98 Publish MQTT(publishTopic, json_output); //MQTT 傳回PM10
```



程式解說

Sleep Mode

```
103 Close_MQTT(); //關閉 MQTT 連結
104 //Send_ATcommand("AT+QPOWD=0", 1); //關閉 BC26 電源
105 Send_ATcommand("AT+CFUN=0",1); //讓 BC26 進入省電模式
106 sleepAndWakeUp(STANDBY, &rt, 1*60); //1分鐘傳輸一次
107 nvic_sys_reset();
```





MQTT Dash 說明

MQTT Dash 僅適用於 **Android** 設備，可用於 **智能手機** 和 **平板電腦**。它包含一個儀表板，可以在其中註冊多個專案，例如針對不同的房間。使用 **Icon**，**文字**，**圖片** 和類似內容，可以訂閱或發佈不同的主題。它支持 **Raspberry Pi**，並且可以 **24/7 全天候工作**。

安裝和配置 **MQTT Dash App**

於 **Google Play** 商店搜索 “ **MQTT Dash** ”



”並且安裝

安裝後打開 “ MQTT Dash 應用程序 ”
點擊右上角的 “ **加入** ” 符號。

Name: 輸入名稱 ,這是可自訂。

Address: 輸入位址 `broker.emqx.io`

Port: 端口可以保留 `1883` 值。

MQTT Dash

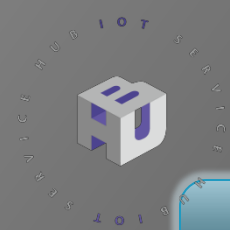
- have just one connection configured.
- If you have more than one connection, you can create home screen shortcut for every connection.
To create shortcut long press on any connection in connections list.
- Keep screen on when connected to this broker
- Allow metrics management. If disabled, you can't add, edit, delete or rearrange metrics.
This serves as protection from unintentional metrics changing.

Name

Address

Port

1883



MQTT Dash 說明

MQTT Dash

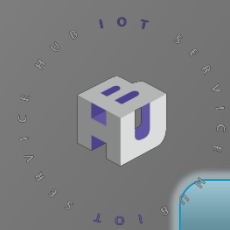


JavaScript 語法運用

ON RECEIVE
ON DISPLAY
ON TAP

參考資料:

<https://reurl.cc/8yDDmM>



MQTT Dash 練習

練習一:

從 **DSI2598+** (60秒傳送一次) 模擬的測試資料

最高溫度 (Text)	Topic :	v1/devices/me/telemetry
	JSON:	MaxTemp.value
最低溫度 (Text)	Topic :	v1/devices/me/telemetry
	JSON:	MinTemp.value
目前溫度 (Text)	Topic :	v1/devices/me/telemetry
	JSON:	NowTemp.value
濕度 (Range)	Topic :	v1/devices/me/telemetry
	JSON:	Humi.value

