

A futuristic, metallic humanoid figure is shown in profile, facing right. The figure has a highly detailed, segmented body with glowing blue circuitry and data points. The background is a dark teal color with a network of glowing blue lines and nodes, suggesting a digital or artificial intelligence theme. The figure's head is particularly prominent, with a large, circular, glowing blue element on the side, possibly representing a sensor or a data processing unit. The overall aesthetic is sleek and high-tech.

DSI5168

長照中心老人離床監測

設計者：李奕賢

指導單位：經濟部工業局

執行單位：資策會數位服務創新研究所

內容大綱

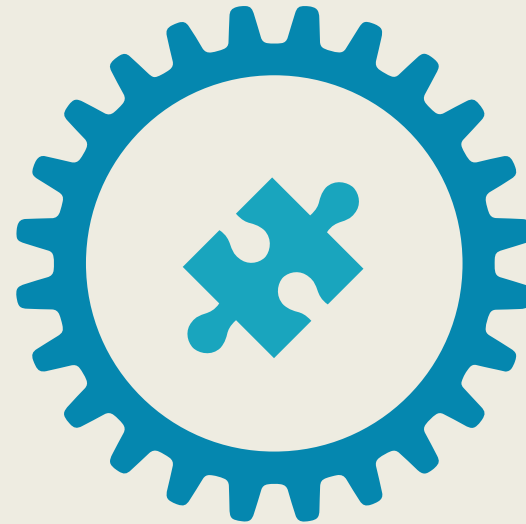
Chapter1

摘要



Chapter2

硬體介紹



Chapter3

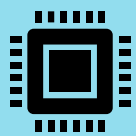
軟體介紹



Chapter4

成果展示





➤ Chapter 1 摘要

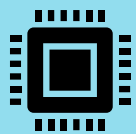
設計目的:

隨著台灣高齡人口比例不斷上升，扶養比亦同步上升，但對於扶養人力不足的情況下，遠端掌握長者行動之需求逐漸出現。因此，此設計目的是希望能達成長者、行動不便者離床、臥床監測警示，即便扶養人力不在現場，以可隨時掌握臥床者即時資訊，此專案是以物聯網智造基地所提出之企業出題所發想，對象為「美麗家園老人長照中心」。

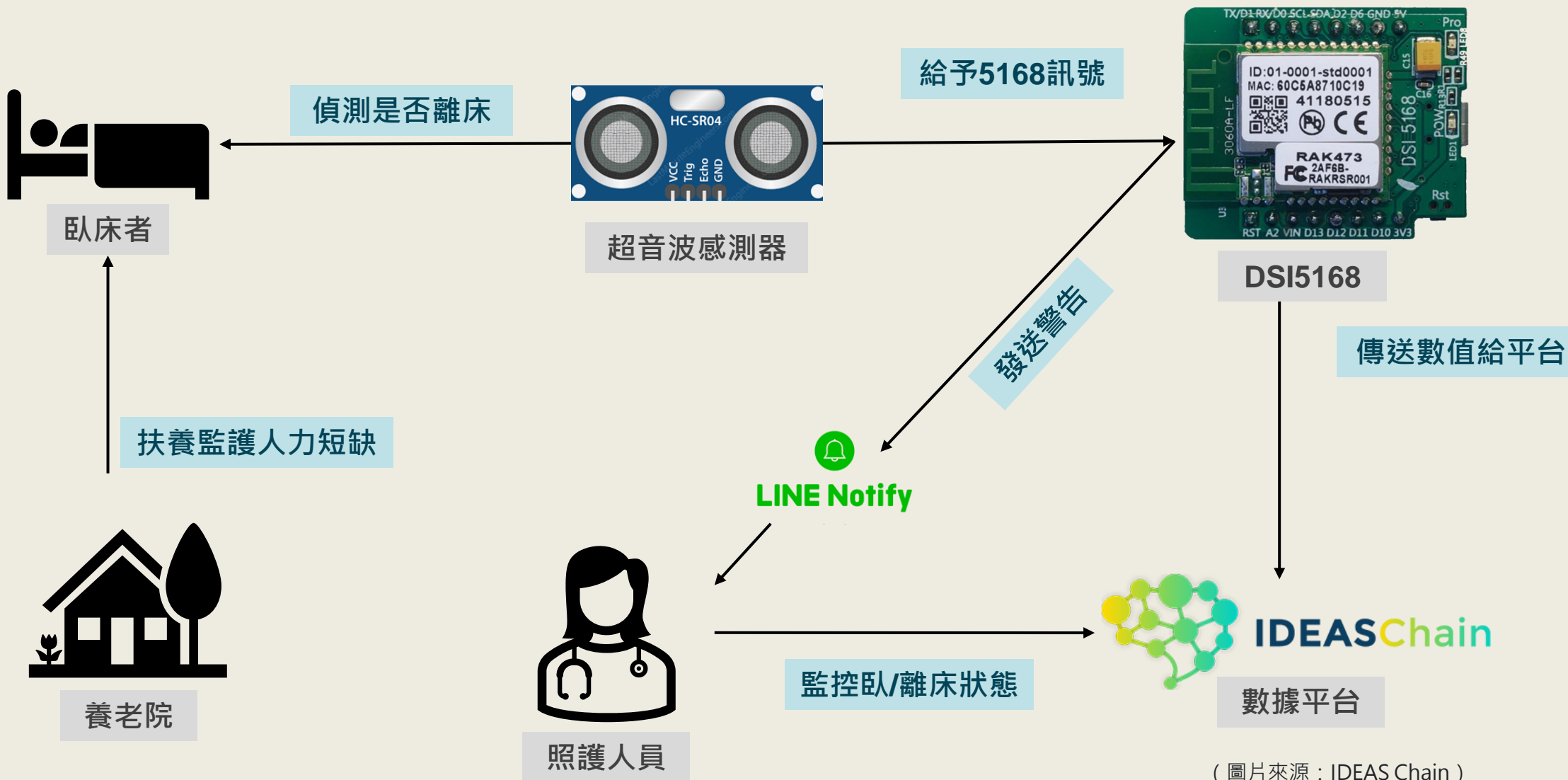
設計方法:

此應用是將超音波感測器裝設在床頭，在臥床者平躺狀態下，量測臥床者上方床頭至床板的距離，若臥床者起身，量取到的距離將會變小，在接收到數值改變後，主動傳送LINE的警示訊息給照護人員。

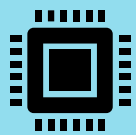




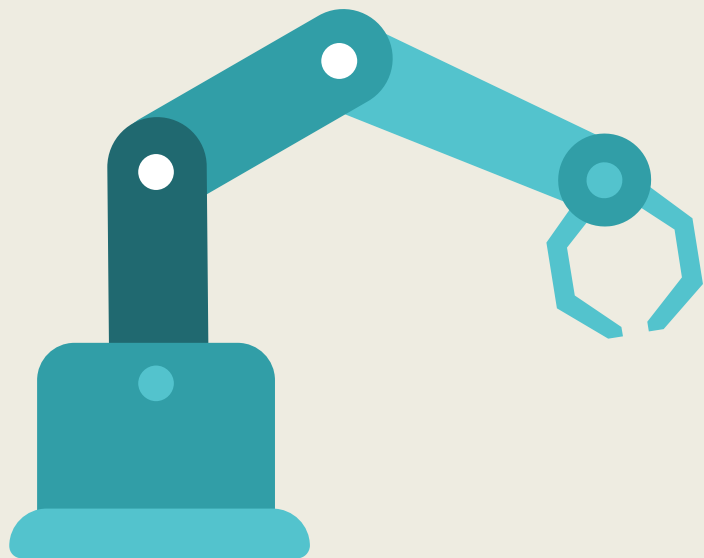
➤ 1-1 案例應用示意圖



(圖片來源 : IDEAS Chain)



➤ Chapter 2 硬體介紹



2-1

DSI5168 國產IC開發板簡介

2-2

DSI5168腳位、規格

2-3

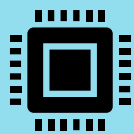
RTL8711AM 晶片介紹

2-4

電路接線介紹

2-5

感測器介紹



➤ 2-1 DSI5168 國產IC開發板簡介



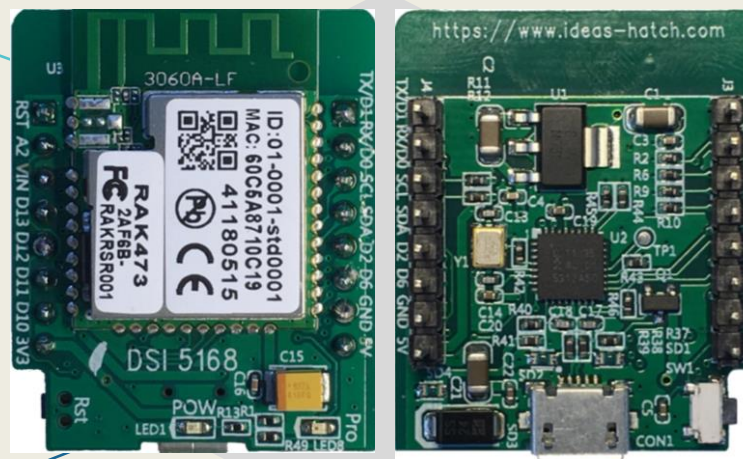
Micro Control Unit

可輸出PWM訊號、亦支援I²C、UART、SPI介面傳輸。



WiFi connect

支援Wi-Fi 802.11b 傳輸速率達11 Mbps。



Development enviroment

可直接使用Arduino IDE編譯燒錄，並完整兼容Arduino開發功能。



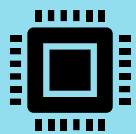
資策會服創所，以國內半導體大廠「瑞昱」所研發的Ameba 系列晶片 - RTL8711AM為核心，設計出「DSI5168」這塊物聯網國產IC開發板，完全兼容Arduino開發特性，還一舉整合MCU、Wi-Fi及豐富的外圍設備，搭配提供標準化的Arduino函式庫，可謂完整的物聯網方案。

IOT

可使用MQTT、http傳輸協定，即時聯網監測及控制。



(圖片來源：資策會服創所)

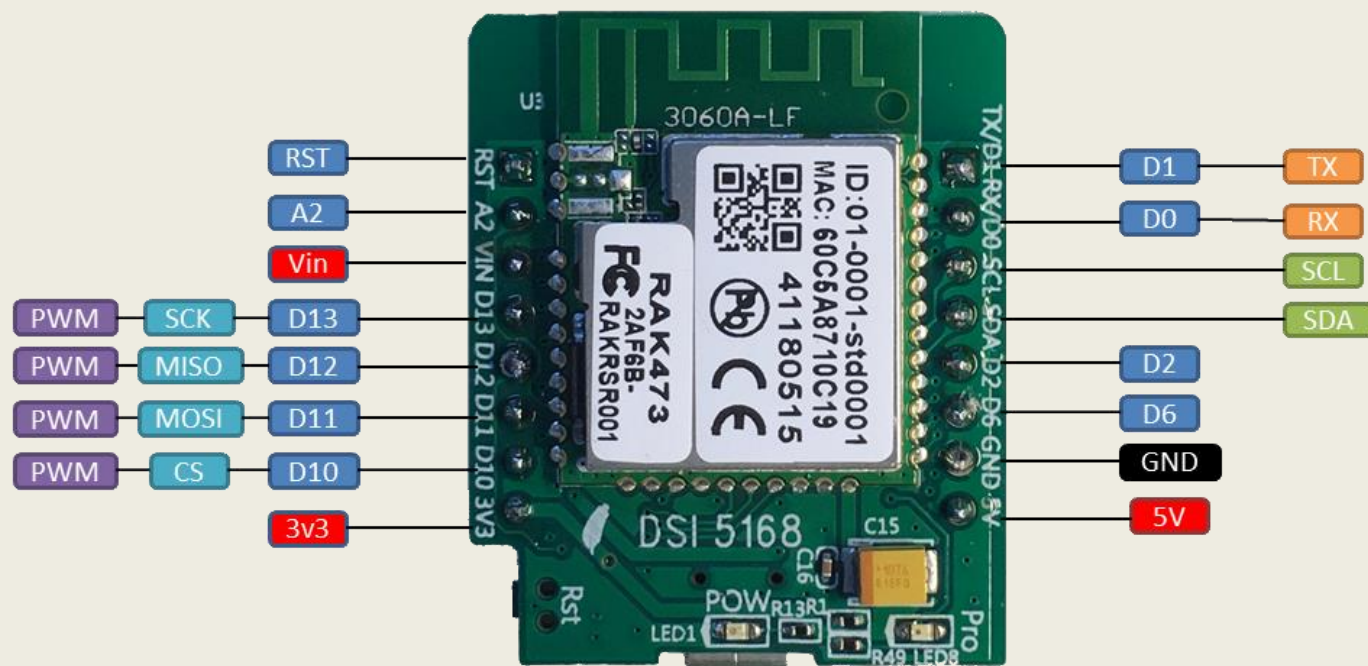


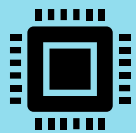
➤ 2-2 DSI5168規格、腳位

開發板規格

硬體功能	規格
Chipset	RTL8711AM
MCU	ARM M3/166MHz
I/O	12
ROM	1MB
SRAM	512KB
Internal Flash	N/A
External Flash	2MB
ADC	1
SPI	1
UART	1
I2C	1
I2S	N/A
PWM	4
SSL	Support

- UART function
- I2C definition
- Arduino definition
- SPI definition
- PWM function





2-3

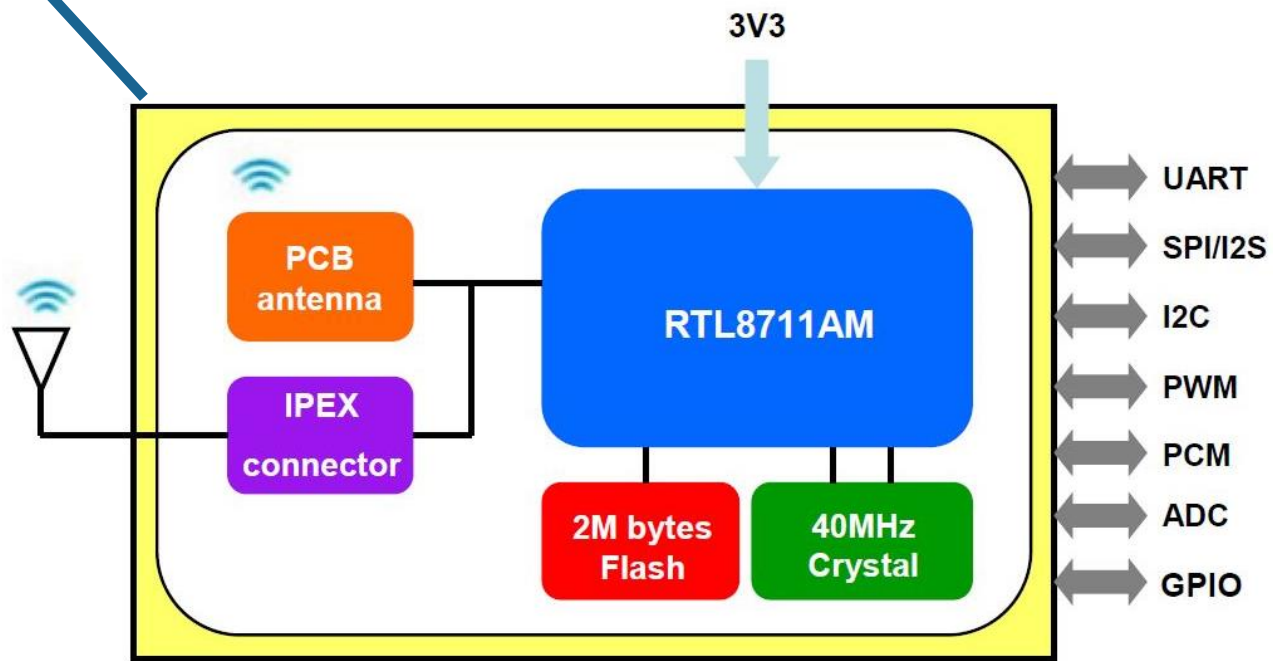


REALTEK

Ameba RTL8711AM介紹

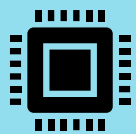
晶片介紹

此晶片擁有GPIO、PWM功能及SPI, I2C, I2S, PCM, UART, ADC等介面。裝載於DSI5168以PCB antenna天線連網，支援休眠功能，能有效降低功耗。



技術規格

1. 晶片尺寸:19×22.25×2.3 mm
2. 區域網路協定:IEEE 802.11
3. 支援天線:PCB Antenna、I-PEX
4. Flash:2Mb
5. GPIO腳位:19 pins



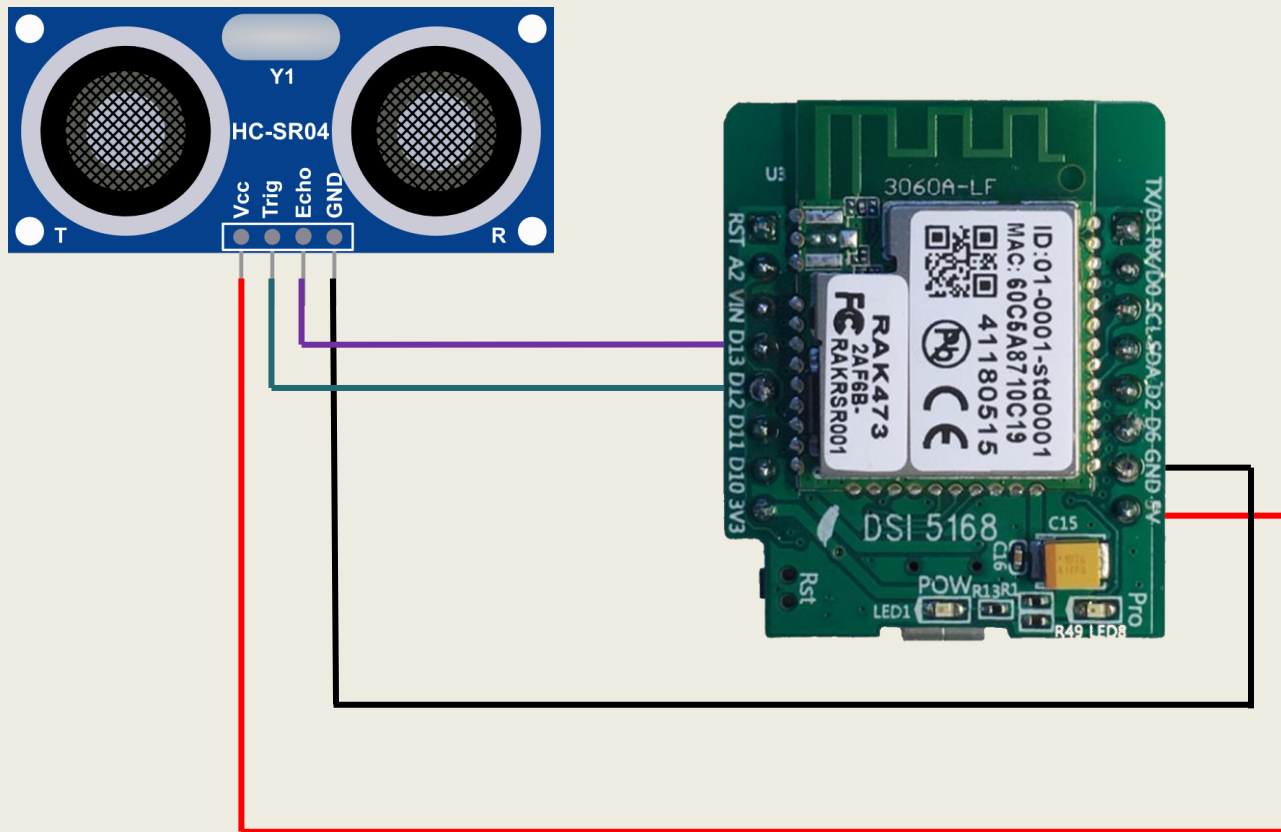
➤ 2-4 電路接線介紹

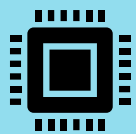
所需材料

1. DSI5168開發板
2. 超音波模組 HC-SR04
3. 杜邦排線 *4

腳位連接

- DSI5168接上micro usb(接上後電源燈恆亮)
- 超音波模組:
 - Vcc → 5V
 - trig → D12
 - echo → D13
 - GND → GND

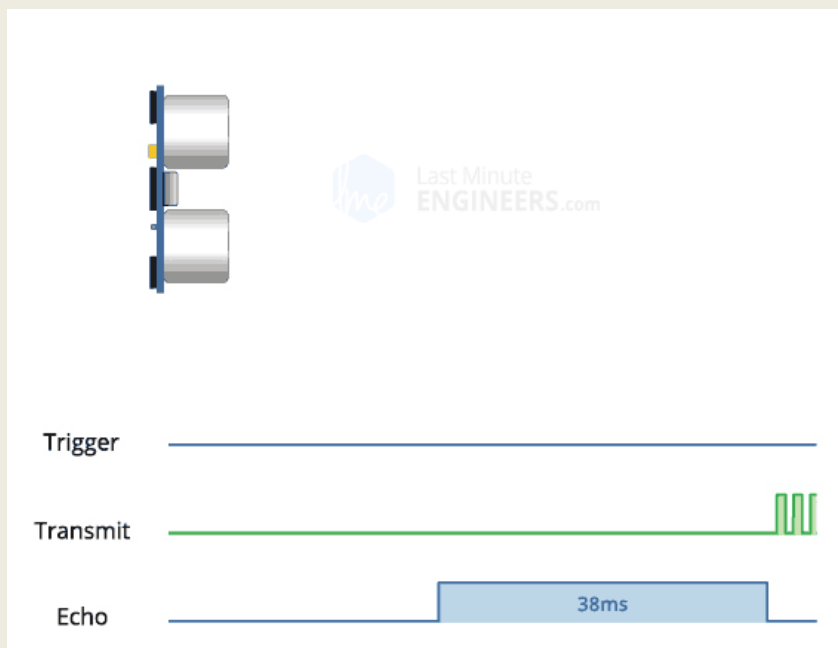




➤ 2-5 感測器介紹

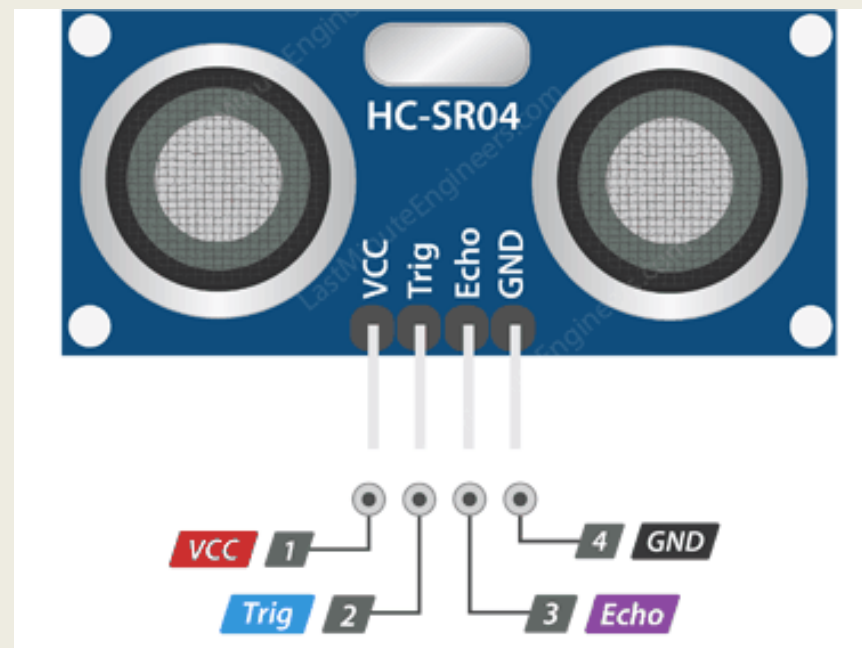
超音波感測器HC-SR04

由超音波發射器、接收器所組成。當它被觸發的時候，會發射一連串 40 kHz 的聲波並且從離它最近的物體接收回音。如下圖所示，超音波測量距離的方法，是測量聲音在感測器與物體之間往返經過的時間。

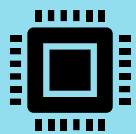


技術規格

工作電壓：5V
感測距離：2cm-450cm (注意: 2cm以內有盲區)
感測角度：15度
感測精度: : 0.3mm
工作電流：15mA
尺寸：45mm x 20mm x 15mm

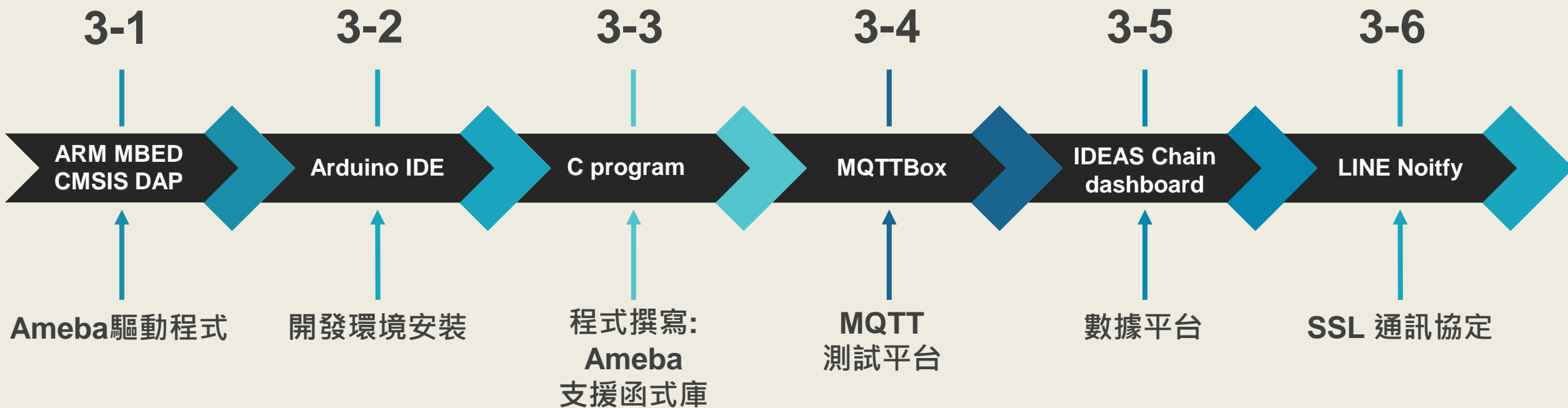


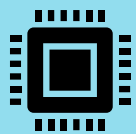
(圖片來源:lastminuteengineers.com)



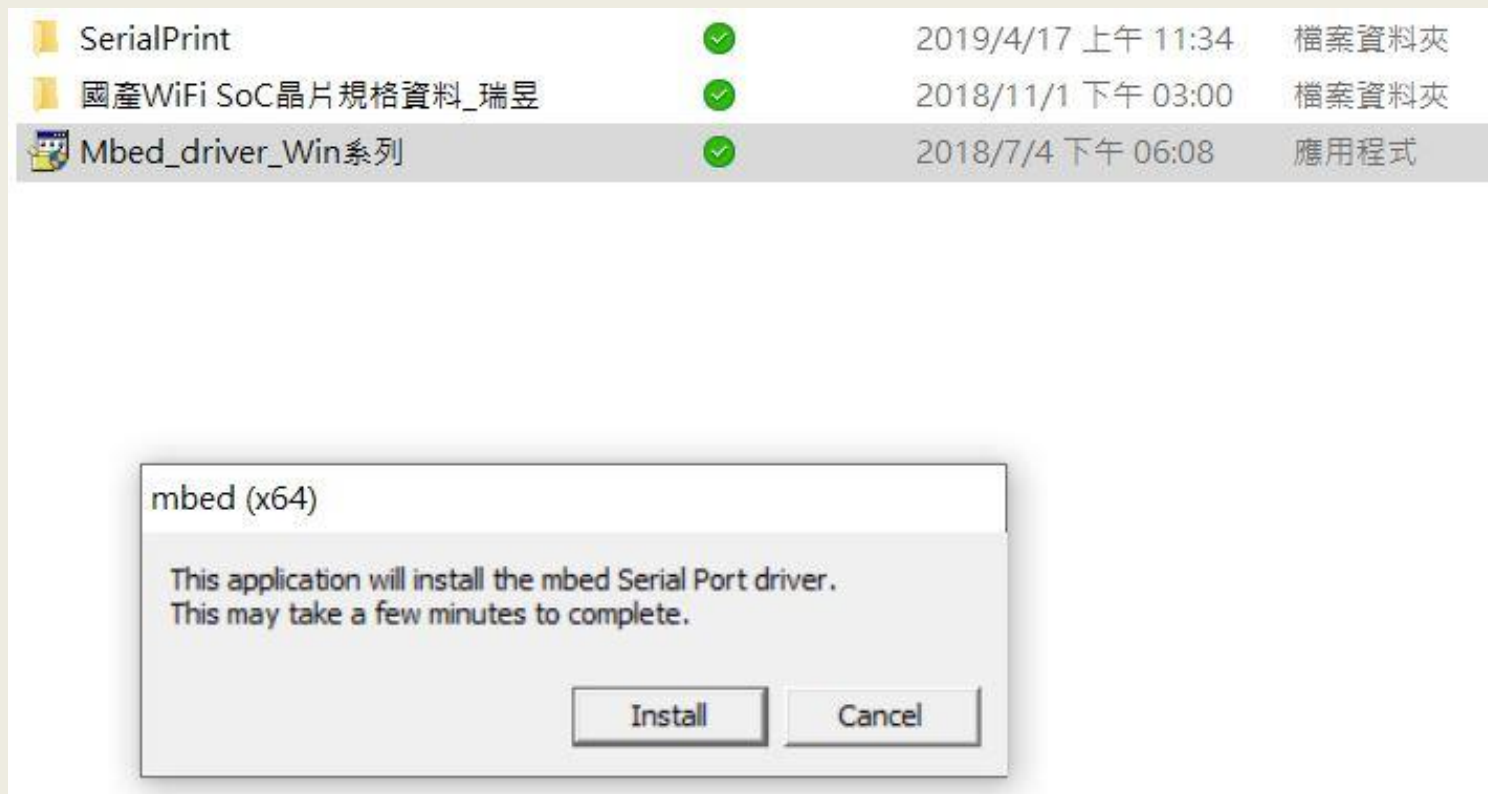
Chapter 3 軟體介紹

For Windows 10

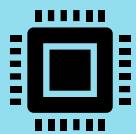




➤ 3-1 安裝Ameba晶片驅動程式



將DSI5168連接到 Windows的電腦上，
執行mbedWinSerial_16466.exe驅動程式，即可看到 Mbed的磁碟及新增的Com Port 序列埠，可
至裝置管理員查看USB連接埠，即已安裝完成。



➤ 3-2-a 安裝Arduino IDE

HARDWARE SOFTWARE CLOUD DOCUMENTATION ▾ COMMUNITY ▾ BLOG ABOUT

Downloads



Arduino IDE 1.8.13

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.


Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file

Windows app Win 8.1 or 10 

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

[Release Notes](#) [Checksums \(sha512\)](#)


HARDWARE SOFTWARE CLOUD DOCUMENTATION ▾ COMMUNITY ▾ BLOG ABOUT

Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **50,614,238** times — impressive! Help its development with a donation.

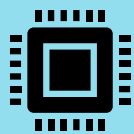
\$3 \$5 \$10 \$25 \$50 Other

JUST DOWNLOAD **CONTRIBUTE & DOWNLOAD**

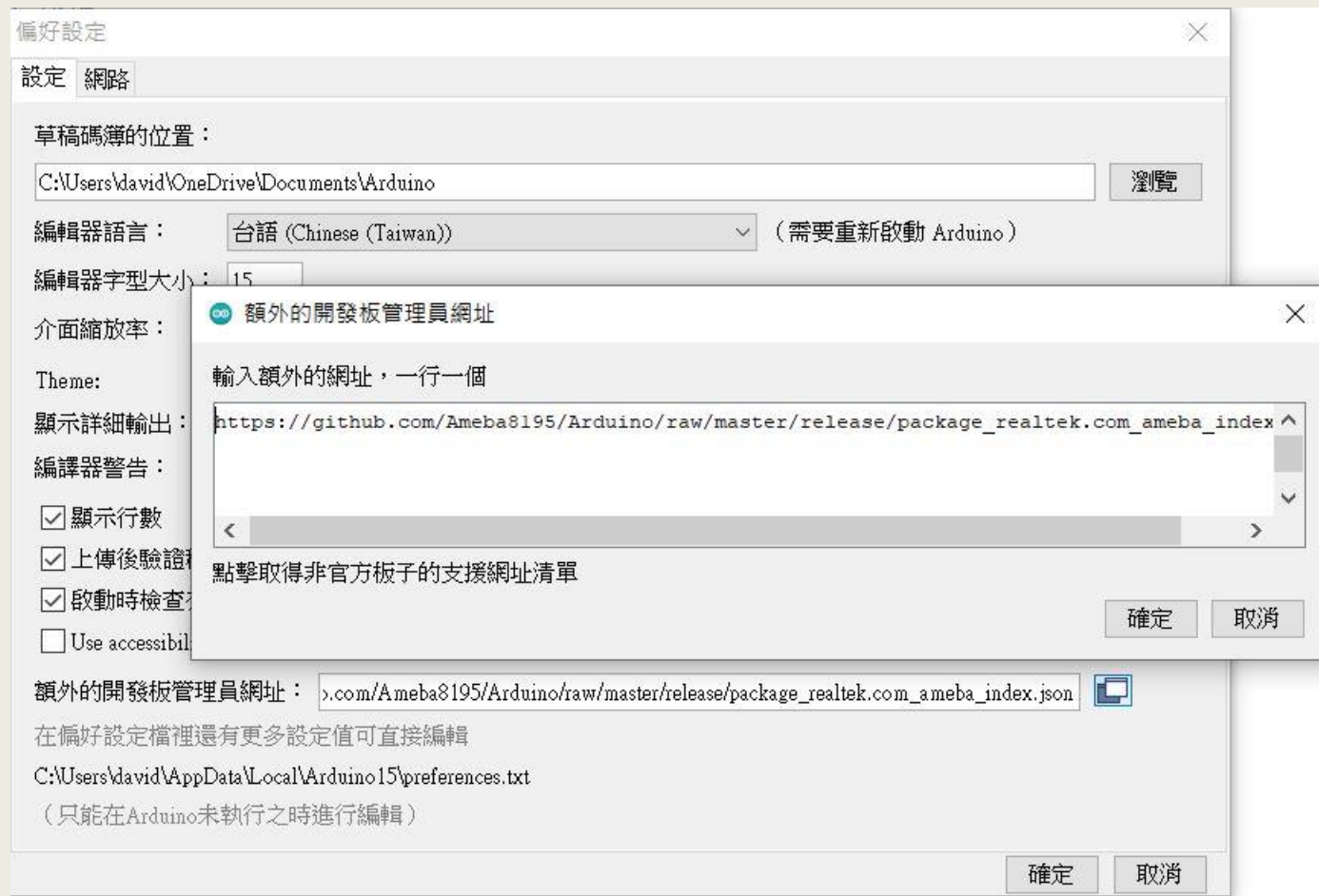


至以下網址下載Arduino IDE:
<https://www.arduino.cc/en/software>
並選擇符合您電腦的版本

可自行選擇是否贊助Arduino，若暫且不贊助，點選JUST DOWNLOAD即可

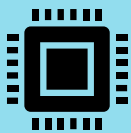


3-2-b Arduino IDE環境設定

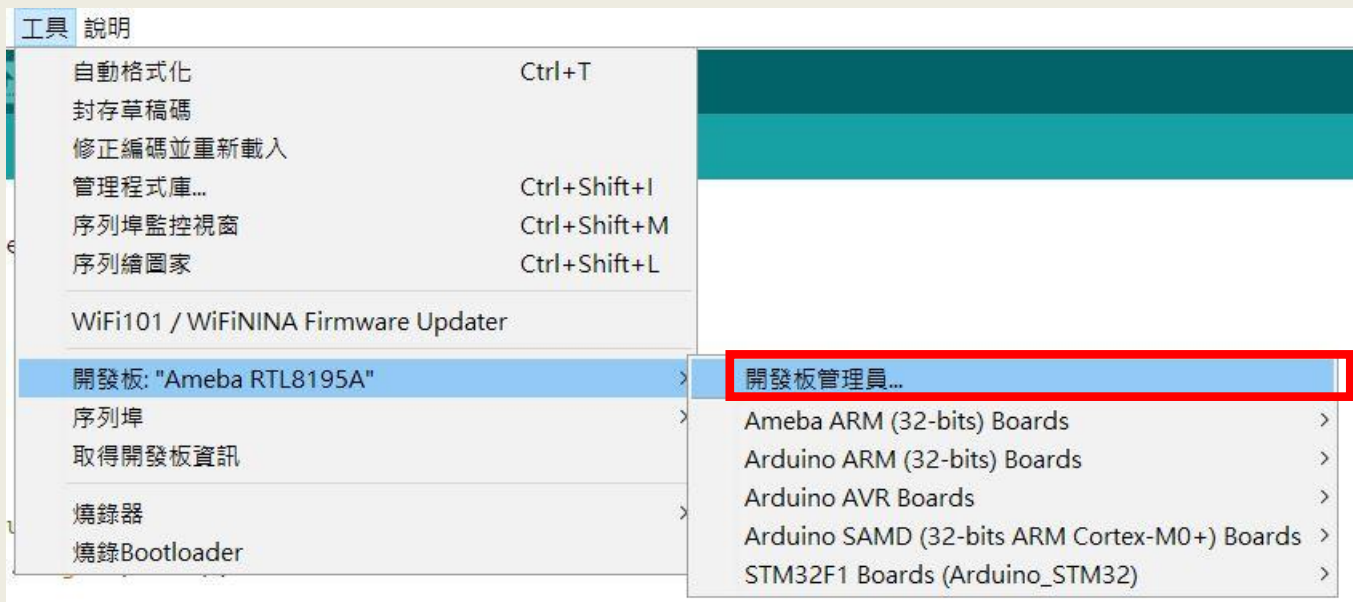


在額外的開發板管理員網址中輸入以下網址：

https://github.com/Ameba8195/Arduino/raw/master/release/package_realtek.com_ameba_index.json

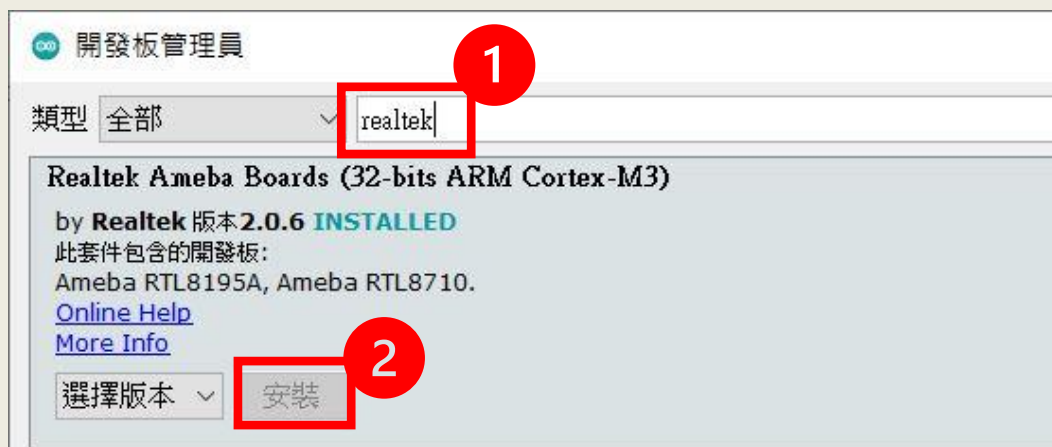


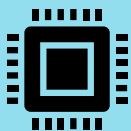
3-2-c Arduino IDE環境設定



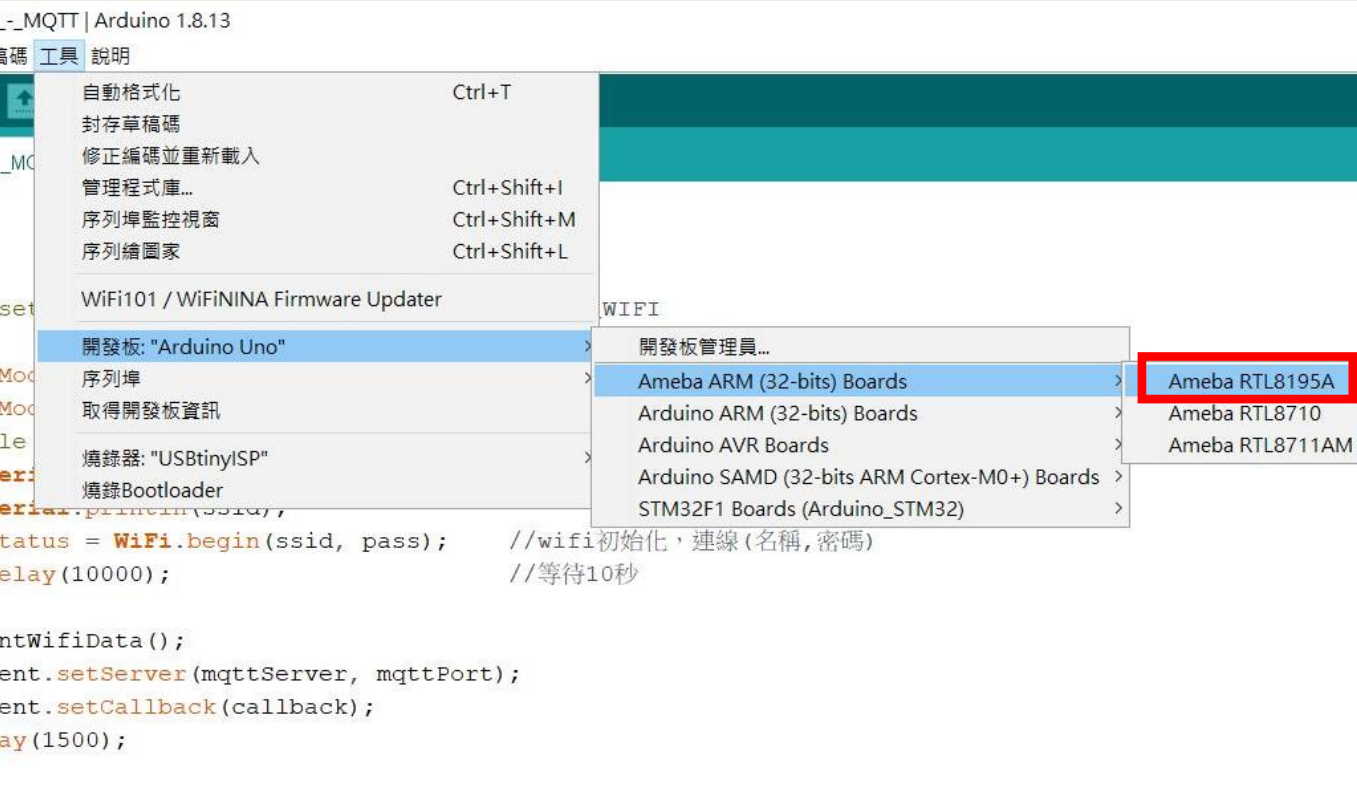
在Arduino IDE 功能列的”工具”中，選擇”開發板管理員”

輸入:realtek，並選擇Realtek Ameba Boards (32-bit ARM Cortex-M3) 安裝



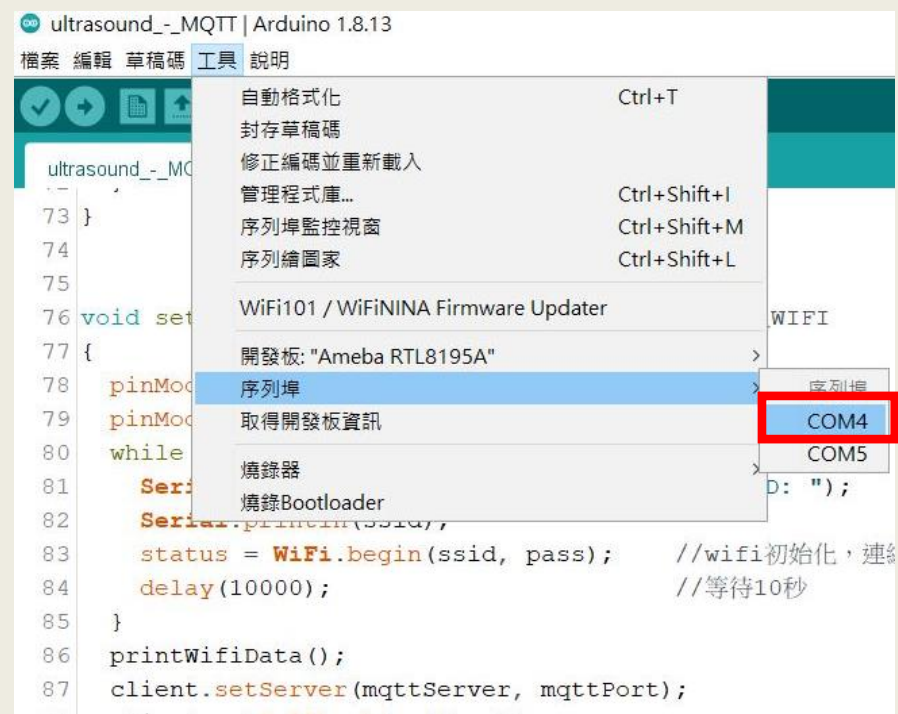


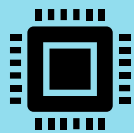
➤ 3-2-d Arduino IDE環境設定



選擇開發板:Ameba RTL8195A

選擇連接埠:COM X
(可從電腦裝置管理員中查看port)





➤ 3-3 程式撰寫

使用Arduino IDE



Ameba 支援函式庫



<WiFi.h>

使用此Arduino內建函式庫可直接使用固定函式連接WiFi，僅需在 `WiFi.begin(ssid, pass)` 中設定路由器之名稱及密碼，便可輕鬆連上網路。



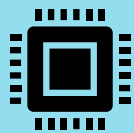
<PubSubClient.h>

使用此函式庫，可直接套用MQTT專用函式，並對Client端、Subscribe及Publish進行設定。



<AmebaServo.h>

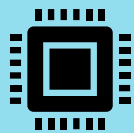
使用此函式庫，可以對伺服馬達進行精確的角度控制，其功能與Arduino內建的 `<servo.h>` 相同



> 3-3-a 程式撰寫

程式碼下載: https://github.com/Lys-0929/DSI5168_MQTT-SSL

```
#include <WiFi.h>           //WIFI library
#include <PubSubClient.h>   //MQTT library
char ssid[] = "*****";    // SSID:router name
char pass[] = "*****";    // pass:router password
String Linetoken = "*****";
int status = WL_IDLE_STATUS; // keep connecting
char mqttServer[] = "iiot.ideaschain.com.tw"; // take ideaschain as server
int mqttPort = 1883;
char clientId[] = "*****"; // MQTT client ID. Create an unique ID.
char username[] = "*****"; // device access token(change your own access token of IDEASChain)
char password[] = ""; // don't need to set up
char subscribeTopic[] = "v1/devices/me/telemetry"; //fixed topic, do not modify
char publishTopic[] = "v1/devices/me/telemetry"; //as the same as subscribeTopic
String payload_string;
char publishTopicStr1[] = "sensorDist";
char publishPayload[] = "{\"sensorDist\":\"30\"}";
WiFiSSLClient client1;
char host[] = "notify-api.line.me"; //LINE Notify API URL
int trigPin = 12;
int echoPin = 13;
```

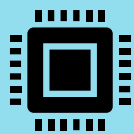



3-3-b 程式撰寫

```
void callback(char *topic, byte *payload, unsigned int length) { //recept the data from server
    Serial.print("Message arrived in topic: ");
    Serial.println(topic);
    Serial.print("Message:");
    for (int i=0; i<length; i++) {
        Serial.print( (char) payload[i]);          // convert *byte to string
    }

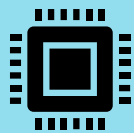
    Serial.println();
    Serial.println("-----");
}

WiFiClient wifiClient;
PubSubClient client(wifiClient);                //define the client's name
```



3-3-c 程式撰寫

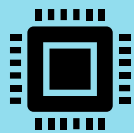
```
void publishData( char*publishTopicStr, float sensorValue){
char sensorDist [30];
sprintf (sensorDist , "{\">%s \":\">%2f \"}", publishTopicStr,sensorValue);
Serial.println(sensorDist);
while(!client.connected()){
  Serial.println("Attempting MQTT connection Attempt to connect...");
  if(client.connect (clientId, username, password)){
    Serial.println("MQTT connected");
    client.publish(publishTopic,sensorDist);
    client.subscribe(subscribeTopic);
  }
  else{
    Serial.print("failed rc= ");
    Serial.print(client.state());
    Serial.println("try again in 5 seconds ");
    delay(5000);
  }
}
```



> 3-3-d 程式撰寫

```
void reconnect() {                // client connect to the MQTT server

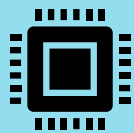
while (!client.connected()) {    //while(disconnect),then run the loop continually
  Serial.println("Attempting MQTT connection...");
  if (client.connect(clientId, username, password)) { // try to connect
    Serial.println("MQTT connected");                //after connected, publish the topic & payload
    client.publish(publishTopic, "payload_string");
    client.subscribe(subscribeTopic);                //resubscribe the topic
  } else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    delay(5000);                                     //wating for 5 second to reconnect
  }
}
}
```



3-3-e 程式撰寫

```
void setup()           //set up the pinmode and WIFI
{
  pinMode(trigPin ,OUTPUT);
  pinMode(echoPin ,INPUT);
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass); //initialize wifi setting
    delay(10000); //wating for WiFi connecting for 10 second
  }
  printWifiData();
  client.setServer(mqttServer, mqttPort);
  client.setCallback(callback);
  delay(1500);
}

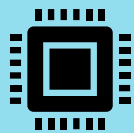
void printWifiData() {
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
}
```



3-3-f 程式撰寫

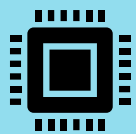
```
void loop()
{

digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
float duration_us = pulseIn(echoPin, HIGH); //Measure the integral time of pulse triging and receiving.
float distance_cm = 0.017 * duration_us; //Change the integral into distance.
Serial.print("distance: ");
Serial.print(distance_cm);
Serial.println(" cm");
delay(500);
if(isnan(duration_us) ){
  Serial.println("Failed to read from sensor!");
  return;
}
```

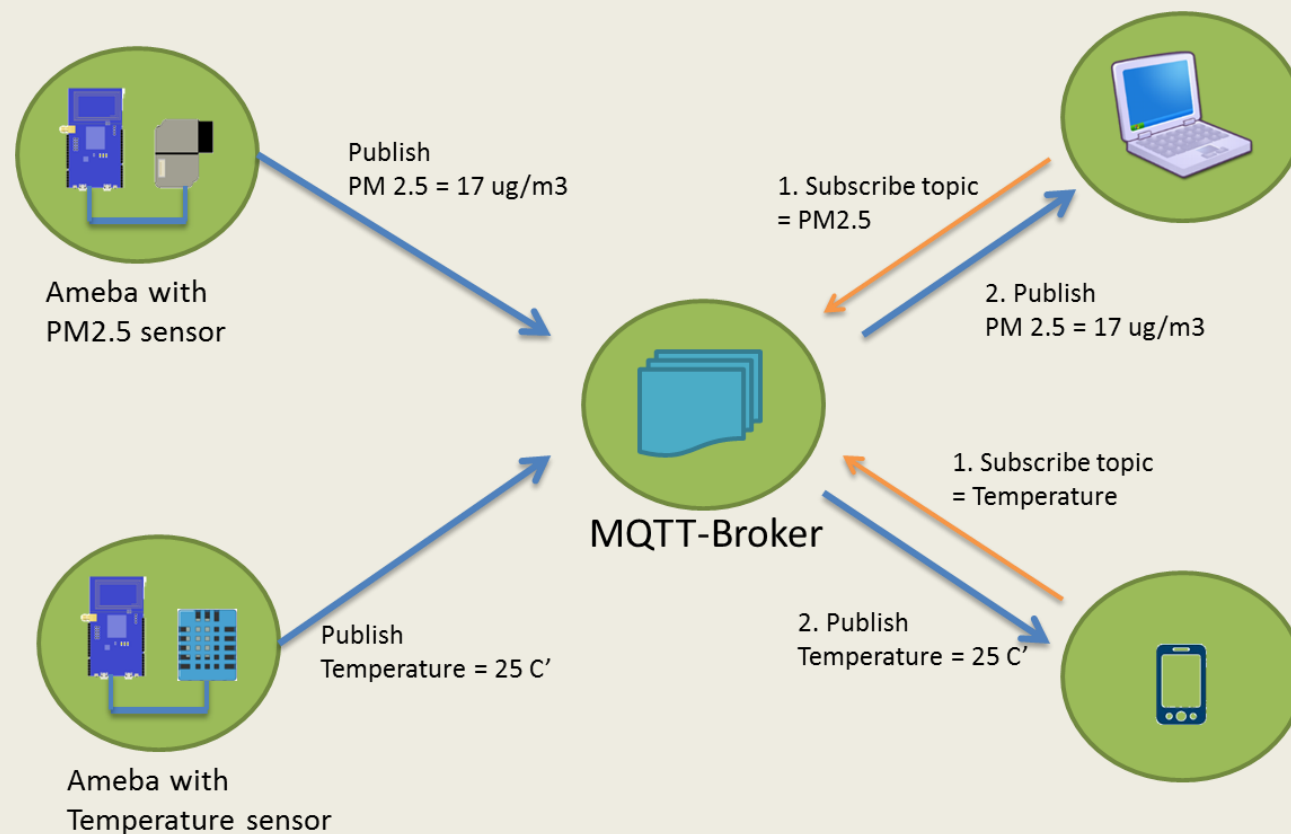
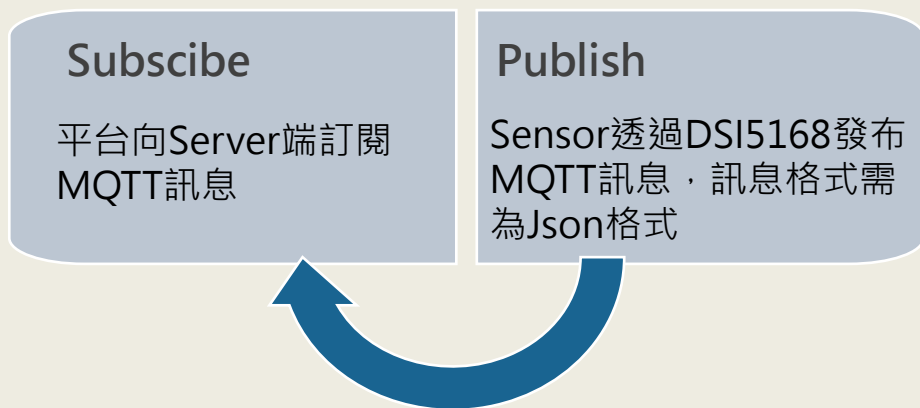
3-3-g 程式撰寫

```
if (distance_cm <= 50) {
  String message = " <Warning>: The patient is getting out of bed!"; //The content of line message.
  message += "\n The distance between patient and bed head=" + String(((float)distance_cm) + " cm";
  Serial.println(message);
  if (client1.connect(host, 443)) {
    int LEN = message.length();
    String url = "/api/notify"; //POST header
    client1.println("POST " + url + " HTTP/1.1");
    client1.print("Host: "); client1.println(host);
    //Access token
    client1.print("Authorization: Bearer "); client1.println(Linetoken);
    client1.println("Content-Type: application/x-www-form-urlencoded");
    client1.print("Content-Length: "); client1.println( String((LEN + 8) ));
    client1.println();
    client1.print("message="); client1.println(message);
    client1.println();
    delay(2000);
    String response = client1.readString();
    Serial.println(response); //Display the result of responding
    client1.stop(); //Disconnecting
  }
  else {
    Serial.println("connected fail");
  }
}
delay(5000);
if(!client.connected()){
  reconnect();
  delay(2000);
}
client.disconnect();
client.loop();
delay(300);
publishData(publishTopicStr1,distance_cm);
client.loop();
}
```

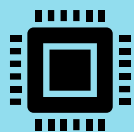


➤ 3-4-a MQTT server 測試工具

使用MQTT server測試工具，目的是為了要測試你所使用的server，可以順利地透過你所設定的topic來subscribe和publish你所要傳送的payload。



(圖片來源 : amebaiot.com)



➤ 3-4-b MQTT server 測試工具

Step 1

使用google chrome 應用程式搜尋MQTT的測試工具，如MQTTBox、MQTTLens等

Step 2

在此使用MQTTBox來示範，將此應用程式下載至電腦。



chrome 線上應用程式商店

首頁 > 應用程式 > MQTTBox



MQTTBox

來源網站: workswithweb.com

★★★★☆ 28 | 擴充功能 | 30,000+ 位使用者

總覽

評論

支援

相關項目

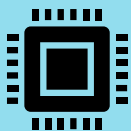
MQTT LOAD [Start Load Test] [View Graph] [View Data]

Load test 1 - ws://iot.eclipse.org:80/ws - Test Type:publishing, Msg Count:20, Instances:4, Topic:

Name	Status	Published Time
Instance 1	Done	4.7530s
Published Messages: 20	QoS Response: 20	QoS Time: 4.8520s
Load test completed successfully		Sep-28-2016 03:22:41:904 PM
Connection to broker closed		Sep-28-2016 03:22:41:890 PM
Saving data...		Sep-28-2016 03:22:41:887 PM
Waiting for QoS responses...		Sep-28-2016 03:22:41:787 PM
Publishing completed		Sep-28-2016 03:22:41:787 PM
Publishing messages to topic...		Sep-28-2016 03:22:37:034 PM
Connected to broker		Sep-28-2016 03:22:37:034 PM

client 4 error
ws://iot.eclipse.org:80/ws
Connection Error

client 5
ws://iot.eclipse.org:80/ws
Not Connected



3-4-c MQTTBox 設定

MQTTBox Edit Help

Menu MQTT CLIENT SETTINGS

MQTT Client Name Ultrasound	MQTT Client Id 830346d0-896e-11eb-8e26-2532a0ef1bf0	Append timestamp to MQTT client id? <input checked="" type="checkbox"/> Yes	Broker is MQTT v3.1.1 compliant? <input checked="" type="checkbox"/> Yes
Protocol mqtt / tcp	Host iiot.ideaschain.com.tw	Clean Session? <input checked="" type="checkbox"/> Yes	Auto connect on app launch? <input checked="" type="checkbox"/> Yes
Username xfJsQcKkZWGG8mgd05YE	Password Password	Reschedule Pings? <input checked="" type="checkbox"/> Yes	Queue outgoing QoS zero messages? <input checked="" type="checkbox"/> Yes
Reconnect Period (milliseconds) 1000	Connect Timeout (milliseconds) 2000	KeepAlive (seconds) 5	Will - Payload {\"on\":1}
Will - Topic v1/devices/me/telemetry	Will - QoS 0 - Almost Once	Will - Retain <input type="checkbox"/> No	

Save Delete

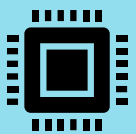
Clientname:任意指定

Username:平台的存取權杖(可參考教學之3-5-e)

MQTT Client id:指定一個獨一無二的名稱

Host: <https://ideaschain.com.tw>

(使用IDEAS Chain作為Server及數據平台)



3-4-c MQTTBox收發測試

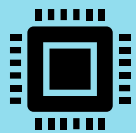
The screenshot shows the MQTTBox web interface. The top navigation bar includes 'MQTTBox', 'Edit', and 'Help'. Below this is a status bar with a 'Menu' button, a 'Connected' indicator, and buttons for 'Add publisher' and 'Add subscriber'. The main content area is split into two panels. The left panel, titled 'Topic to publish', has a text input field containing 'v1/devices/me/telemetry', a 'QoS' dropdown set to '0 - Almost Once', a 'Retain' checkbox, a 'Payload Type' dropdown set to 'Strings / JSON / XML / Characters', and a 'Payload' text area containing '{"distance":30}'. A 'Publish' button is at the bottom. The right panel, titled 'Topic to subscribe', has a text input field containing 'v1/devices/me/telemetry', a 'QoS' dropdown set to '0 - Almost Once', and a 'Subscribe' button. Below the 'Subscribe' button is a text area containing the JSON payload '{"distance":{"ts":1617887661806,"value":30}}'. At the bottom of the right panel, there is a detailed log entry: 'qos : 0, retain : false, cmd : publish, dup : false, topic : v1/devices/me/telemetry, messageid : , length : 69, Raw payload : 1233410010511511697110991013458123341161153458495449555655554544956485444341189710811710134585148125125'.

Topic:為IDEAS Chain所指定之 路徑:v1/devices/me/telemetry

Payload:需為Json格式

設定完成後先按下Subscribe再按下Publish

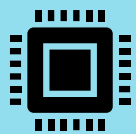
若有成功訂閱到payload訊息，表示成功透過server收發MQTT訊息



3-5-a MQTT數據平台

Step1. 至IDEAS Chain並點選數據平台: <https://iiot.ideaschain.com.tw/home> (請先建立帳號)
在此平台建立專屬專案，並連接儀表板

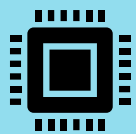
The screenshot displays the IDEASChain MQTT data platform dashboard. The interface includes a top navigation bar with the IDEASChain logo, a home button, and links for 論壇 (Forum), 應用案例 (Application Cases), 開發工具 (Development Tools), 技術支援 (Technical Support), and a user profile for 租戶管理員 (Tenant Manager). A left sidebar menu lists various functions: 首頁 (Home), 規則鏈庫 (Rule Chain Library), 客戶 (Customers), 專案 (Projects), 裝置 (Devices), 描述檔 (Description Files), 部件庫 (Component Library), 儀表板庫 (Dashboard Library), and 稽核日誌 (Audit Log). The main content area is divided into four management sections: 規則集管理 (Rule Set Management) with a 規則鏈庫 (Rule Chain Library) button; 客戶管理 (Customer Management) with a 客戶 (Customers) button; 專案管理 (Project Management) with a 專案 (Projects) button; and 裝置管理 (Device Management) with 裝置 (Devices) and 描述檔 (Description Files) buttons.



➤ 3-5-b MQTT數據平台

Step2. 點選左側:<專案>，再點選右下角的+，添加專案，填寫名稱類型後，點選添加

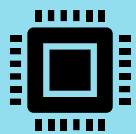
The screenshot displays the IDEASChain MQTT data platform interface. On the left, a sidebar menu contains several options: 首頁 (Home), 規則鏈庫 (Rule Chain Library), 客戶 (Customers), 專案 (Projects), 裝置 (Devices), 描述檔 (Description Files), 部件庫 (Component Library), 儀表板庫 (Dashboard Library), and 稽核日誌 (Audit Log). The '專案' (Projects) option is highlighted with a red box and a red circle containing the number '1'. In the top right corner, there is a navigation bar with links for 論壇 (Forum), 應用案例 (Application Cases), 開發工具 (Development Tools), and 技術支援 (Technical Support), along with search and refresh icons. A modal dialog box titled '添加專案' (Add Project) is open in the center. It has a close button (X) in the top right corner. The dialog contains two input fields: '名稱*' (Name*) with the value 'Ultrasound' and '專案類型*' (Project Type*) with the value 'sensor'. Below these fields is a '描述' (Description) field. At the bottom of the dialog, there are two buttons: '添加' (Add) and '取消' (Cancel). The '添加' button is highlighted with a red box and a red circle containing the number '4'. In the bottom right corner of the main interface, there is a red circle containing a plus sign (+), which is highlighted with a red box and a red circle containing the number '2'. The background of the interface is dimmed.



➤ 3-5-c MQTT數據平台

Step3. 點選左側:<裝置>，再點選右下角的+，添加專案，填寫名稱類型後，點選添加

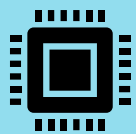
The screenshot displays the IDEASChain MQTT data platform interface. On the left, a navigation menu includes options like '首頁', '規則鍵庫', '客戶', '專案', '裝置', '描述檔', '部件庫', '儀表板庫', and '稽核日誌'. The '裝置' (Devices) option is highlighted with a red box and a red circle containing the number '1'. In the center, a '添加裝置' (Add Device) dialog box is open. It contains a '名稱*' (Name) field with 'Ultrasound' entered, a '裝置類型*' (Device Type) field with 'sensor' entered, and a '是開道' (Is Open) checkbox. The '添加' (Add) button is highlighted with a red box and a red circle containing the number '4'. In the bottom right corner of the main interface, a '+' button is highlighted with a red box and a red circle containing the number '2'. The background shows a list of devices, including one named 'Ultrasound' with a 'SENSOR' type and '公開' (Public) status.



➤ 3-5-d MQTT數據平台

Step4. 點選左側:<裝置>，再點選剛才新增的裝置，並複製存取權杖，貼上於程式碼中(或MQTTBox)

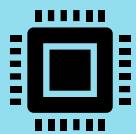
The screenshot displays the IDEASChain MQTT data platform interface. On the left, a navigation menu includes '裝置' (Devices), which is highlighted with a red box and a red circle containing the number '1'. The main content area shows a list of devices, with 'Ultrasound' selected and highlighted with a red circle containing the number '2'. Below the device name, there are several icons for device management. On the right, the 'ULTRASOUND' device details page is shown. At the top, there are tabs for '詳細信息', '屬性', '最新遙測', '警告', '事件', '關聯', and '稽核日誌'. Below these tabs, there are three buttons: '私人', '管理認證', and '刪除裝置'. The '複製存取權杖' button is highlighted with a red box and a red circle containing the number '3'. Below the buttons, there are fields for '裝置名稱' (Ultrasound) and '裝置類型' (sensor). At the bottom, there is a checkbox labeled '是閘道'.



3-5-e MQTT數據平台

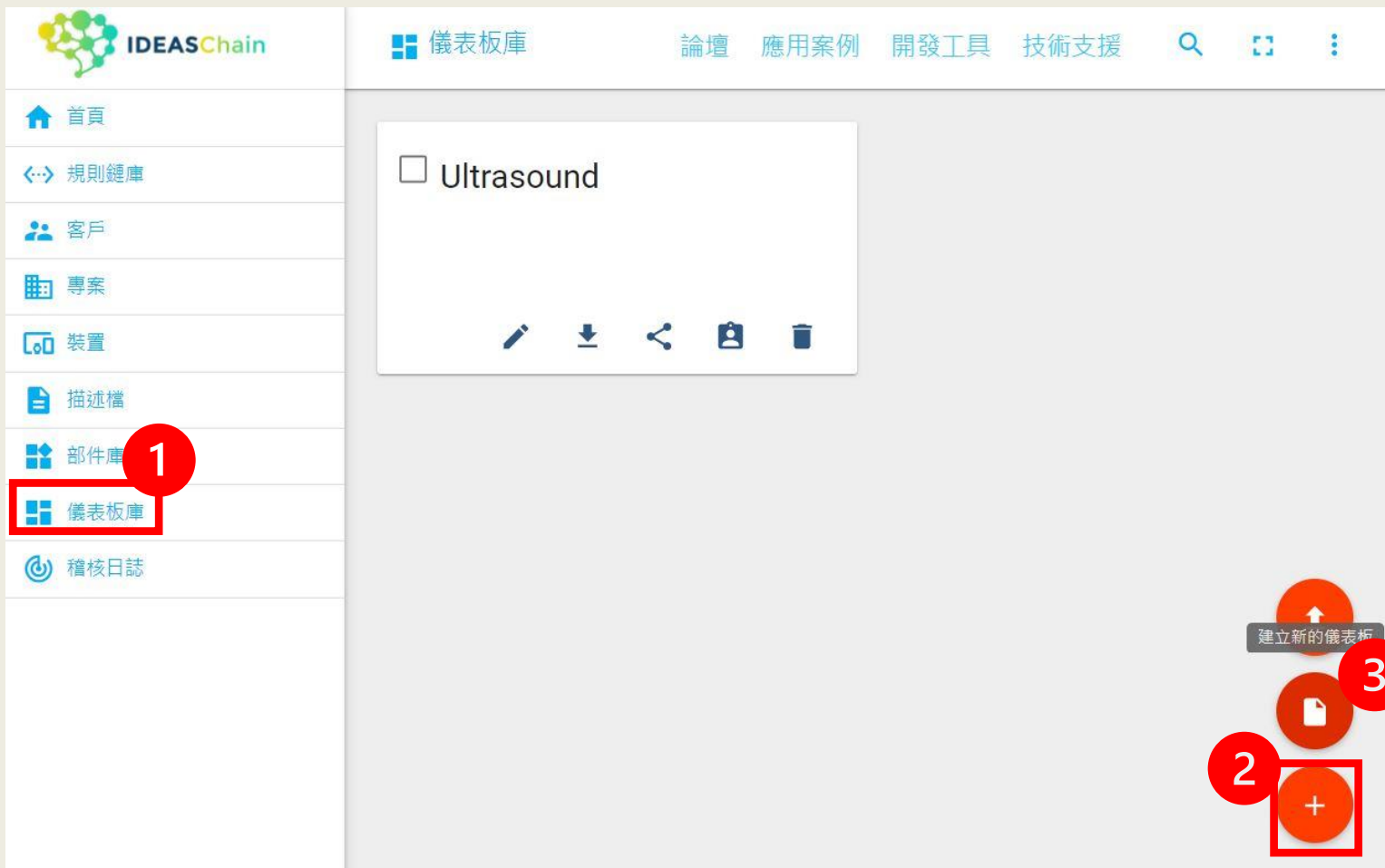
Step5. 點選左側:<裝置>，再點選剛才新增的裝置，接著點選關聯，關聯類型填寫“Contains”後，類型點選<裝置>，並填寫剛才加入的裝置名稱，最後點選<添加>。

The screenshot displays the IDEASChain MQTT data platform interface. The left sidebar contains navigation options: 首頁, 規則鏈庫, 客戶, 專案, 裝置 (highlighted with a red box and '1'), 描述檔, 部件庫, 儀表板庫, and 稽核日誌. The main content area shows a '裝置' (Device) section with 'Ultrasound' selected (highlighted with a red box and '2'). A modal dialog titled '添加關聯' (Add Relationship) is open, showing the relationship type 'Contains' (highlighted with a red box), the target entity 'Ultrasound' (highlighted with a red box and '4'), and a '添加' (Add) button. The background interface shows a table with columns for '事件', '關聯' (highlighted with a red box and '3'), and '稽核日誌'. The table is currently empty, and the pagination shows 'Page: 1', 'Rows per page: 5', and '0-0 of 0'.

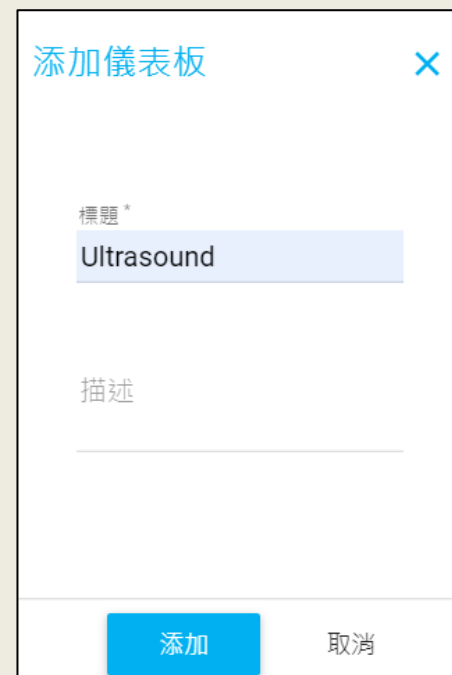


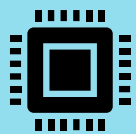
3-5-f MQTT數據平台

Step6. 點選左側:<儀表板庫>，再點選右下角的+，建立新的儀表板



Step7. 添加儀表板並新增標題





➤ 3-5-g MQTT數據平台

Step8. 點選左側:<儀表板庫>，再點選剛才新增的儀表板，建立新的儀表板點選添加

IDEASChain

儀表板庫 > Ultrasound 論壇 應用案例 開發工具 技術支援

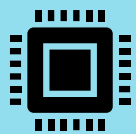
Ultrasound

沒有配置部件

進入編輯模式

Step9. 點選添加新的部件



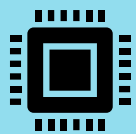


➤ 3-5-h MQTT數據平台

Step10. 點選當前包，在選擇合適的圖表，在這裡使用“Charts”，並在圖表處點一下新增

The screenshot shows the IDEASChain dashboard interface. On the left is a navigation menu with options like '首頁', '規則鏈庫', '客戶', '專案', '裝置', '描述檔', '部件庫', '儀表板庫', and '稽核日誌'. The main area displays a dashboard titled 'Ultrasour' with a '選擇部' (Select Component) dropdown menu. The menu is open, showing options: 'Analogue gauges', 'Cards', 'Charts', 'Control widgets', and 'Date'. The 'Charts' option is highlighted with a red box and a circled '2'. Below the menu, a 'Timeseries E' chart is displayed, also highlighted with a red box and a circled '3'. The chart shows two data series: 'First' (blue) and 'Second' (yellow). The x-axis represents time from 22:20:20 to 22:21:10. The y-axis represents values from 0 to 400. A legend at the bottom left identifies the series. A summary table at the bottom right shows the average values for each series.

Series	平均值
First	161.23
Second	42.64



➤ 3-5-i MQTT數據平台

Step11. 類型:點選實體，參數:輸入sensorDist

添加部件

數據 設定 進階 動作

使用儀表板的時間窗口 時間窗口 即時 - 最後分

資料來源

1.	類型	參數	時間序列
	實體	sensorDist	

沒有找到 'sensorDist' 別名。 建立新別名

實體別名必填。 需要裝置時間序列。

+ 添加

添加 取消

Step12. 類型:點選實體，參數:輸入sensorDist

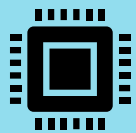
添加別名

別名* sensorDist

過濾類型* 單個實體

類型 裝置* Ultrasound

添加



➤ 3-6 LINE Notify

Step1. 進入LINE Notify 網頁，登入後在右上角點選“個人頁面”

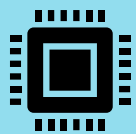
Step2. 點選發行權杖，複製到程式碼的 String Linetoken 中

Generate access token (For developers)

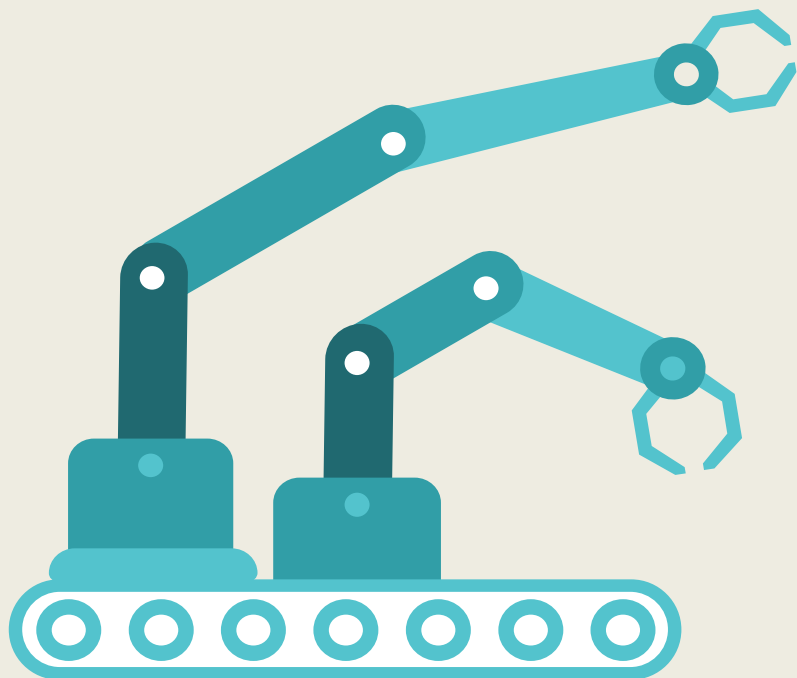
By using personal access tokens, you can configure notifications without having to add a web service.

Generate token

LINE Notify API Document



➤ Chapter 4 成果展示



4-1

WiFi連線-序列埠監控

4-2

MQTT連線-序列埠監控

4-3

模型展示

4-4

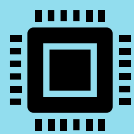
情境模擬

4-5

IDEAS Chain dashboard 成果

4-6

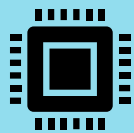
LINE Notify 通知



4-1 WiFi連線-序列埠監控

於Arduino IDE 將程式碼上傳並燒錄，待燒錄完成後，用手按下DSI5168上的RST鍵，接著開始進行WiFi連網，可於序列埠視窗中查看連接狀態。

```
COM7
23:49:10.709 -> addr:1002DFAC size:00042054
23:49:10.709 -> addr:3003DE1C size:001C21E4
23:49:10.709 -> Attempting to connect to SSID: lysiPhone
23:49:10.755 -> interface 0 is initialized
23:49:10.755 -> interface 1 is initialized
23:49:10.755 ->
23:49:10.755 -> Initializing WIFI ...
23:49:11.219 -> WIFI initialized
23:49:11.219 ->
23:49:11.219 -> RTL8195A[Driver]: set ssid [lysiPhone]
23:49:11.219 ->
23:49:13.879 -> RTL8195A[Driver]: start auth to 76:b4:e3:d7:1e:62
23:49:13.879 ->
23:49:13.879 -> RTL8195A[Driver]: auth success, start assoc
23:49:13.879 ->
23:49:13.925 -> RTL8195A[Driver]: association success(res=1)
23:49:13.925 ->
23:49:13.925 -> RTL8195A[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 TKI
23:49:13.972 ->
23:49:13.972 -> RTL8195A[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2
23:49:13.972 ->
23:49:15.515 -> Interface 0 IP address : 172.20.10.12
< >
 自動捲動  Show timestamp NL(newline) 9600 baud Clear output
```

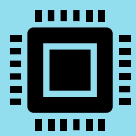


4-2 MQTT連線-序列埠監控

於WiFi成功連網後，可於序列埠視窗中查看MQTT連接狀態，並同時查看超音波感測器的數值。

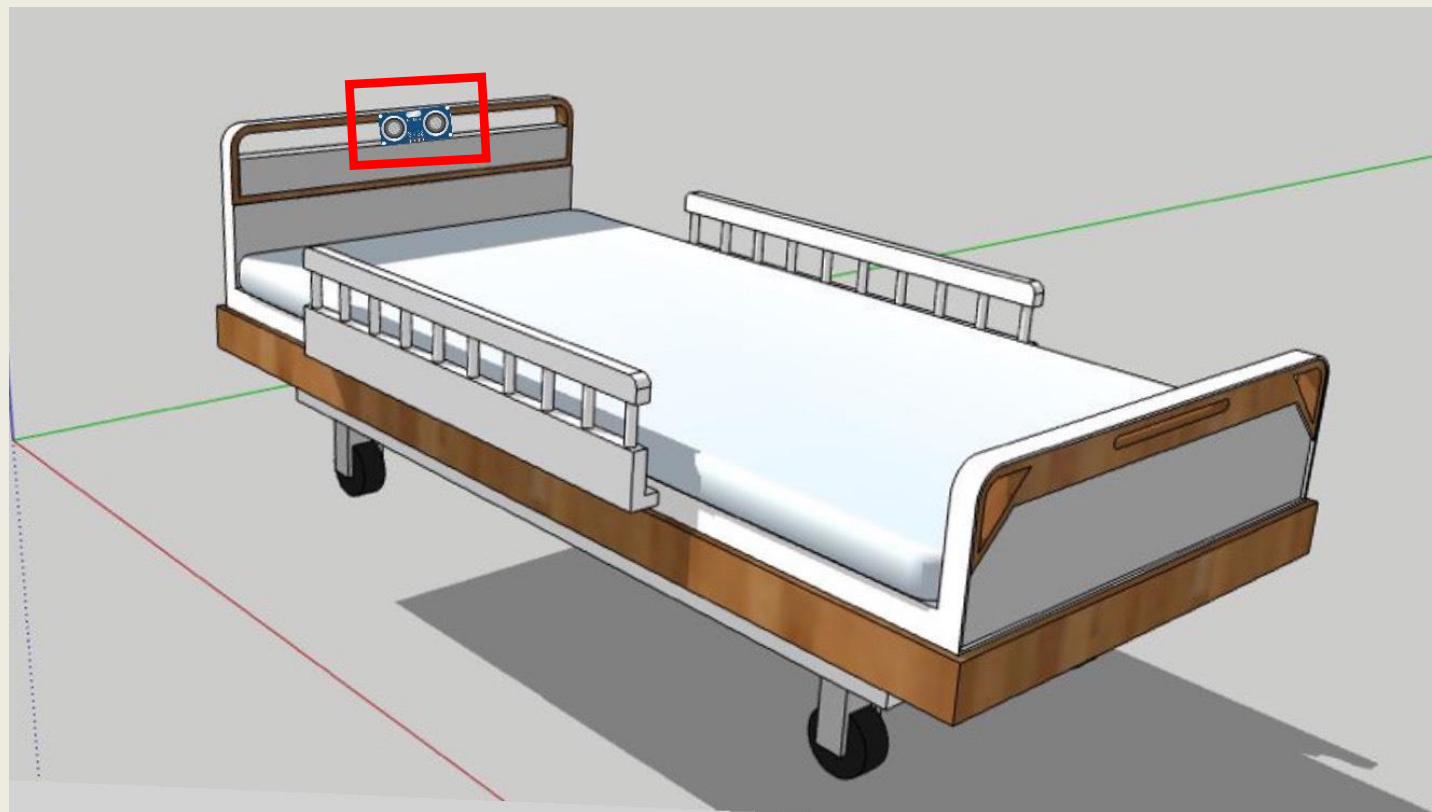
```
COM7
23:49:51.840 -> Connect to Server successful!
23:49:51.887 -> MQTT connected
23:49:52.351 -> distance: 203.37 cm
23:49:53.188 -> {"sensorDist ":"203.37 "}
23:49:53.188 -> Attempting MQTT connection Attempt to connect...
23:49:53.235 ->
23:49:53.235 -> Connect to Server successful!
23:49:53.235 -> MQTT connected
23:49:53.421 -> distance: 207.52 cm
23:49:54.298 -> {"sensorDist ":"207.52 "}
23:49:54.298 -> Attempting MQTT connection Attempt to connect...
23:49:54.345 ->
23:49:54.345 -> Connect to Server successful!
23:49:54.345 -> MQTT connected
23:49:54.481 -> distance: 164.46 cm
23:49:55.362 -> {"sensorDist ":"164.46 "}
23:49:55.362 -> Attempting MQTT connection Attempt to connect...
23:49:55.408 ->
23:49:55.408 -> Connect to Server successful!
23:49:55.408 -> MQTT connected
23:49:55.548 -> distance: 94.42 cm
```

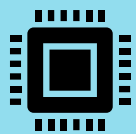
自動捲動 Show timestamp NL(newline) 9600 baud Clear output



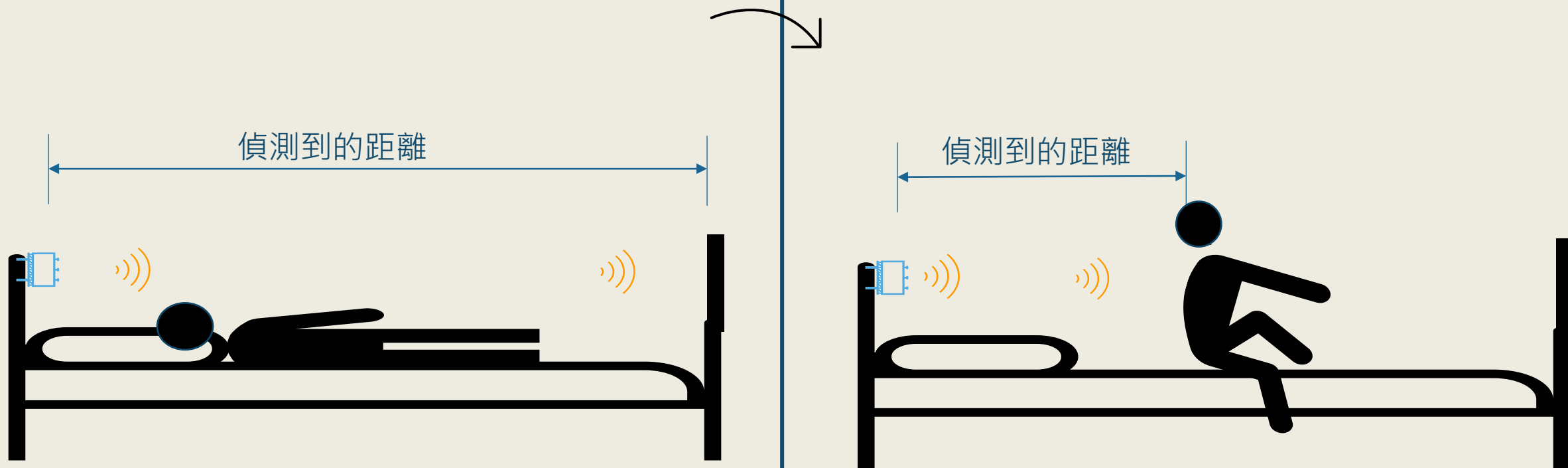
➤ 4-3 模型展示

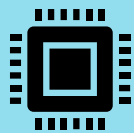
將超音波模組架設於於床頭，若有臥床者起身，超音波模組所讀取到的距離會因此減小。



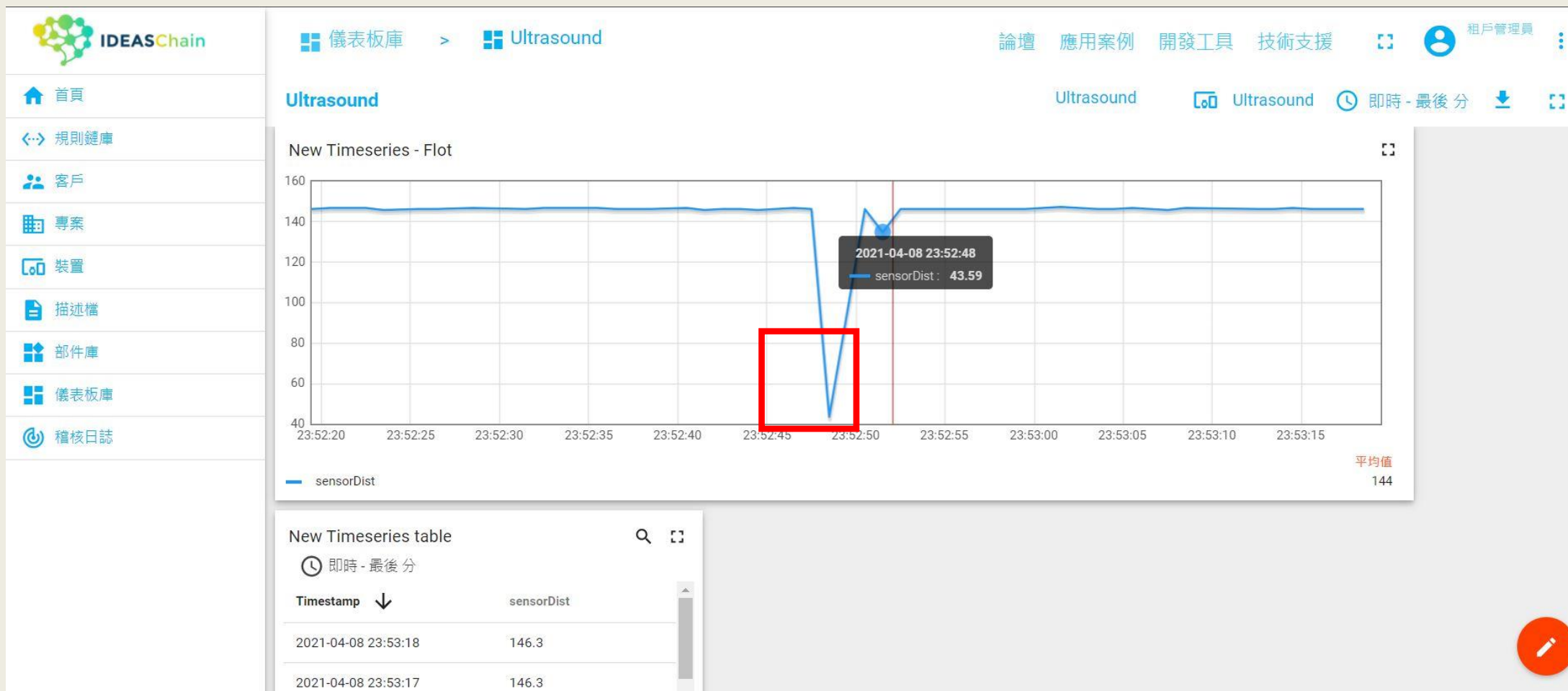


➤ 4-4 情境模擬

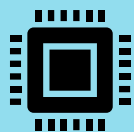




4-5 IDEASChain dashboard 成果

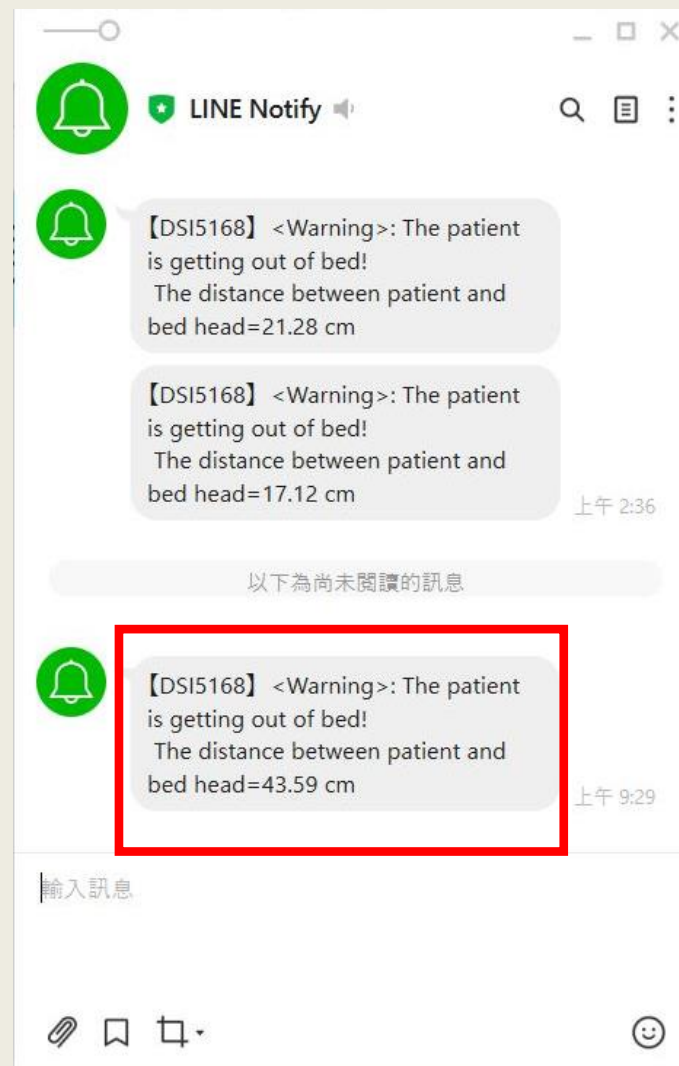


打開IDEAS Chain數據平台中的“儀表板庫”，可以看見超音波所讀取到的數值，如圖若是偵測到數值突然變小，表示臥床者起身，即可立即做出應對反應。



➤ 4-6 LINE Notify

若是感測到的數值變小，直到小於50cm，則會透過DSI5168的SSL(http)通訊協定，發送訊息給Line Notify。如此，即便養老院的照護人員無法隨時監控臥床者，在臥床者起身離床時，亦可收到Line的訊息主動提醒。



<< Thank you >>

