

智慧杯墊



報告者：石庭偉、柯智程、謝肇員、林建安
指導老師：張洧

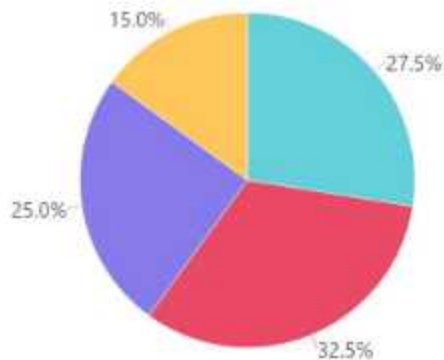
目錄

- ❖ 分工圖
- ❖ 動機
- ❖ 流程圖
- ❖ 影片展示
- ❖ 硬體
- ❖ 軟體
- ❖ 杯墊外型設計
- ❖ 未來展望
- ❖ 參考資料



分工圖表

分工圖表



· 石庭偉 · 柯智程 · 林建安 · 謝肇員

石庭偉(組長): App inventor

柯智程: Arduino

林建安: 雷射切割、外殼設計

謝肇員: 外殼組裝

動機

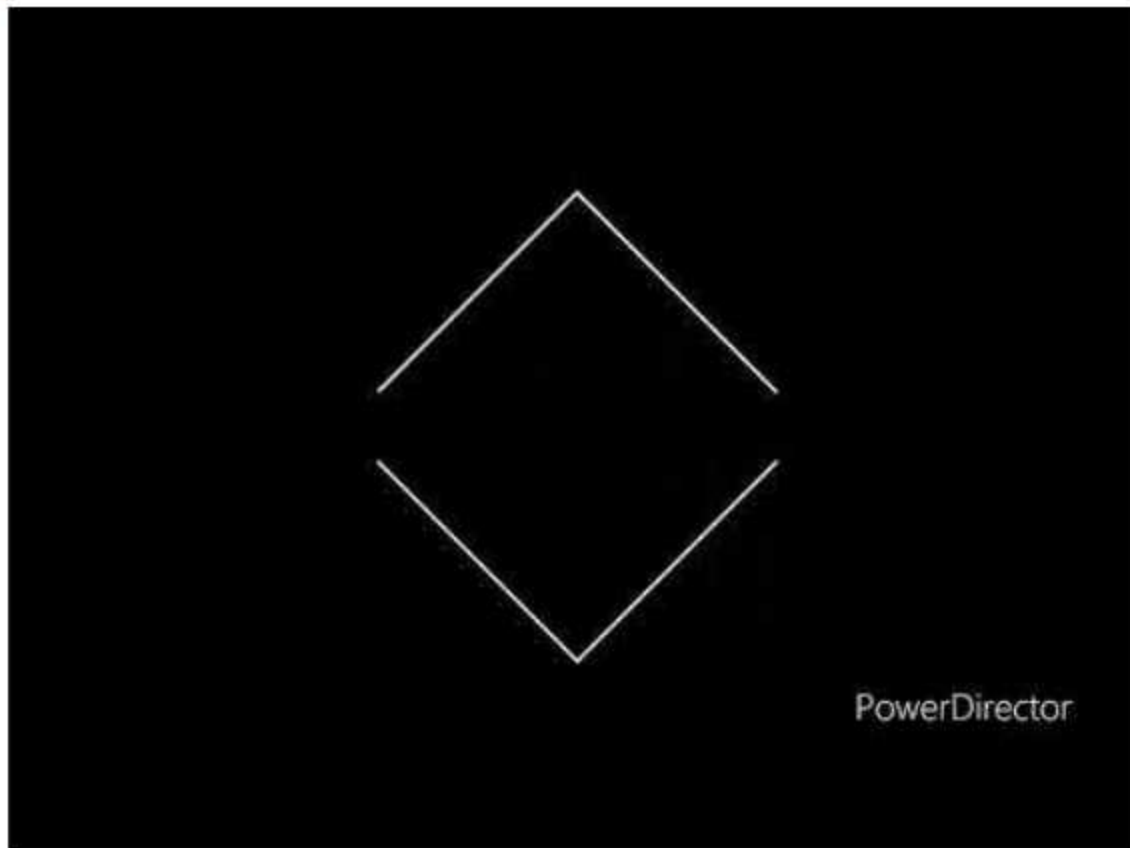
現代都市生活忙碌，不論是年輕學生亦或是上班族都長期處於快節奏的生活中，常常因為攝取水份不足或飲食不均導致健康失衡，然而對人體來說水分是不可或缺的，人體7成由水份構成，水分攝取不足可能影響尿素代謝，累積在體內傷害腎臟，也增加泌尿道的感染風險，嚴重甚至影響整體免疫力。

因此我們從每日攝水量著手，製作一個智慧杯墊，除了提醒人們喝水以外，也讓喝水這看似微不足道的日常增添幾分趣味，並且結合物聯網趨勢，透過藍芽連結至手機，使用者輸入自身身高體重可以計算出每日應攝取水量，手機也會顯示目前杯子內的水量讓使用者方便記錄自己每日的喝水狀況，以此達到我們希望人們每日攝取足夠水分的目的。

流程圖



影片展示



硬體零件

- 重量感測套件
- LCD
- 藍芽模組

主要硬體:重量感測套件



重量感測套件:零件介紹



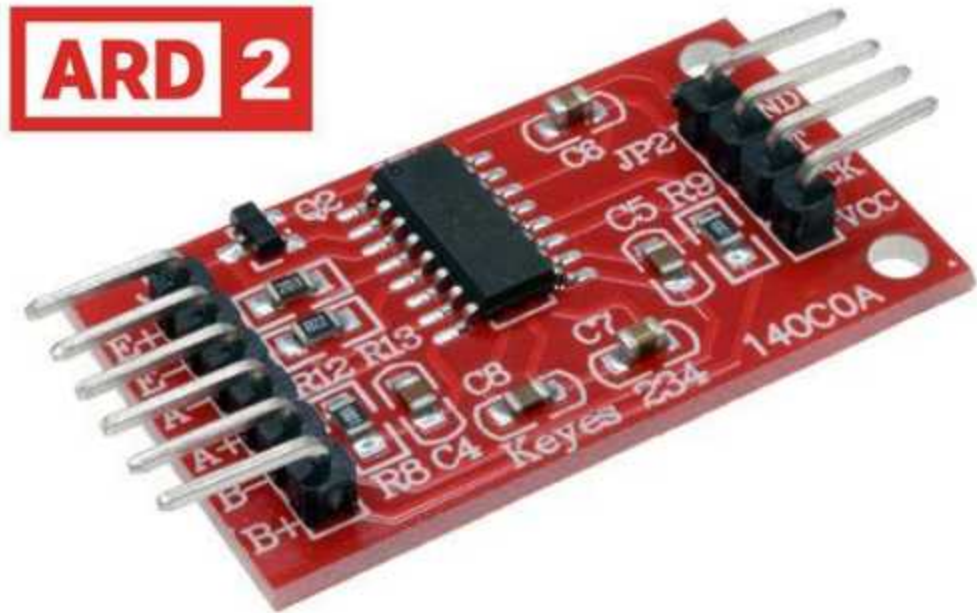
HX711重量感測模組

HX711:專為電子秤設計的24位元

A/D轉換芯片, 包括穩壓電路,

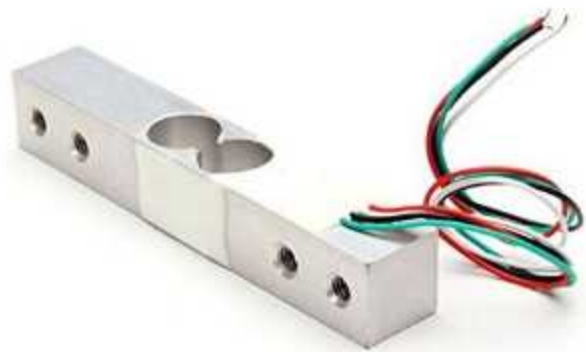
時鐘震盪器且具有集成度高,

響應快, 抗干擾強等優點。



荷重元(Load cell)

一種特殊形式的力量感測器，由應變計和橋式電路組成，當其受到壓力時將產生和作用力成正比的輸出電壓。



LCD 1602

LCD1602顧名思義就是一種小型液晶顯示器，可以顯示兩列16個字元的長度

Arduino整合1602相較簡單且內含許多程式庫，其中較著名的有LiquidCrystal.h



LCD擴充I/O PCF8574

採用PCF8574AT的芯片，具備8位輸出/輸入(I/O)
雙向擴展功能，最多允許8個模組
鍊式連接無須額外的接線

IIC地址：

PCF8574AT 0x20~0x27

PCF8574T 0x38~0x3F



HC-05 藍芽模組

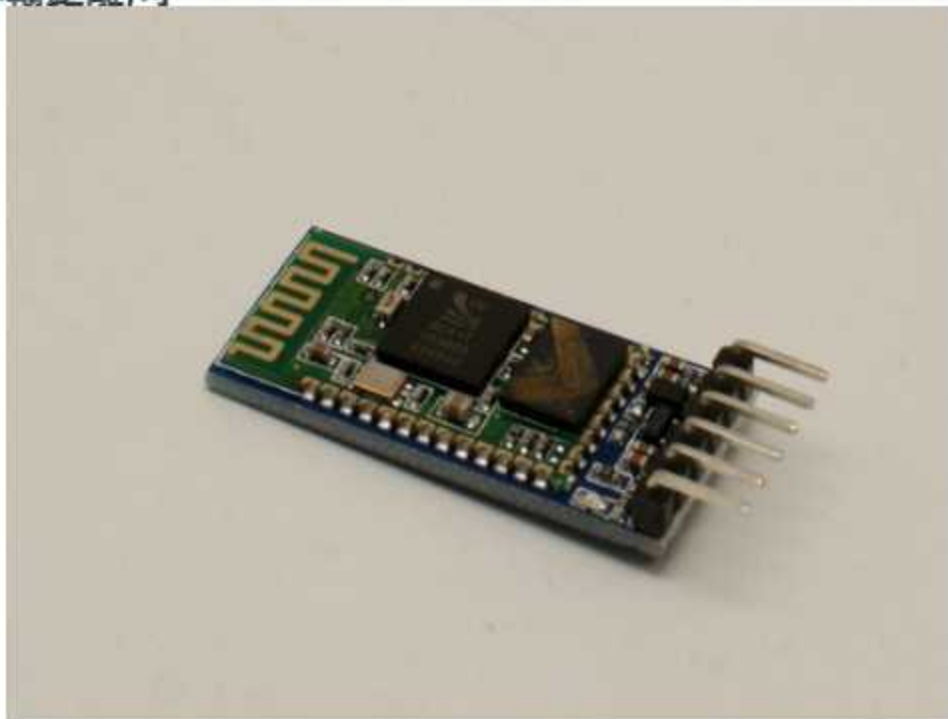
藍2.0+EDR 2Mbps~3Mbps調變深度 10m傳輸距離,可用ATcommand切換主/從模式

預設Baud9600

工作電壓3.6~6V

工作電流30~40mA 通訊8mA

工作溫度-25~75度



軟體程式

- Arduino程式
- APP Inventor程式



Arduino 程式



- 校正
- 測重
- 藍牙(設定)
- 藍芽(傳輸)
- LCD

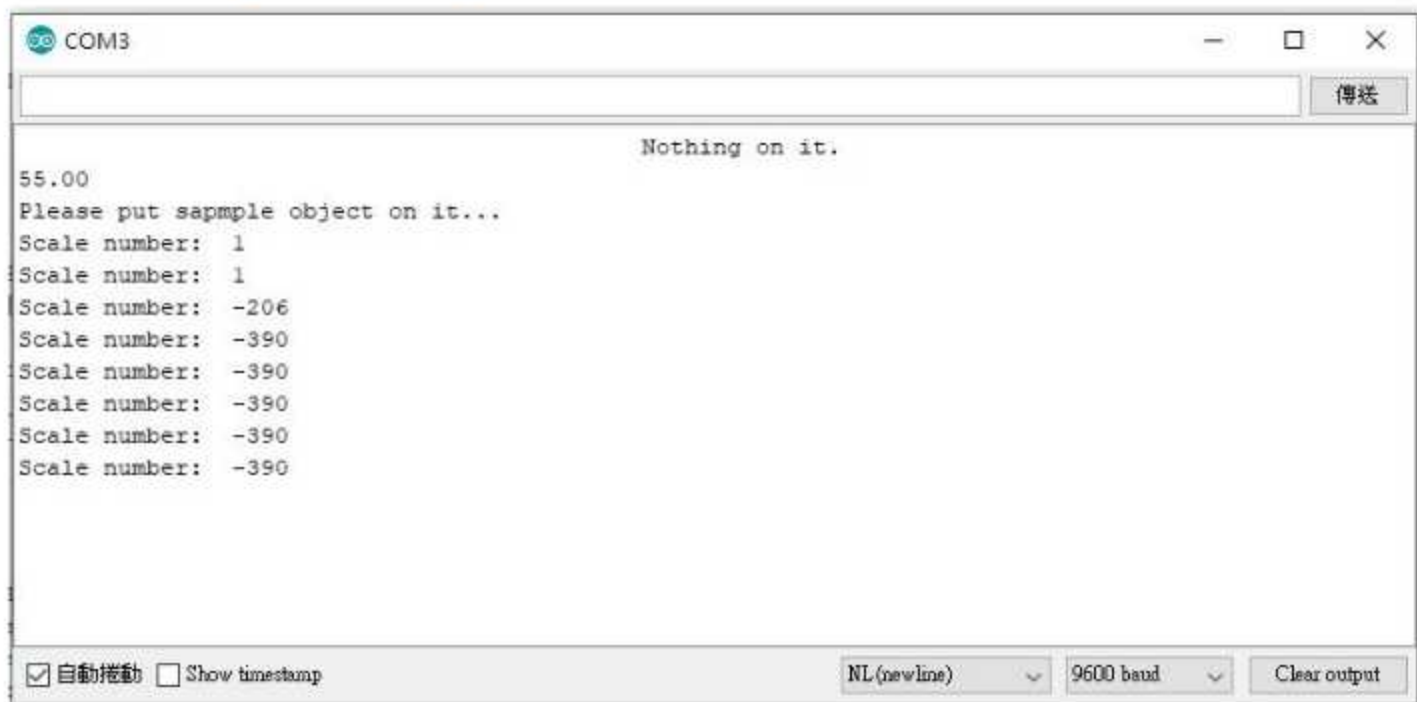


校正程式

```
HX711 scale;
```

```
void setup() {  
  Serial.begin(9600);  
  scale.begin(DT_PIN, SCK_PIN);  
  scale.set_scale();  
  scale.tare();  
  Serial.println("Nothing on it.");  
  Serial.println(scale.get_units(10));  
  Serial.println("Please put sample object on it...");  
}  
void loop() {  
  float current_weight=scale.get_units(10);  
  float scale_factor=(current_weight/sample_weight);  
  Serial.print("Scale number: ");  
  Serial.println(scale_factor,0);  
}
```

校正程式(結果)



The screenshot shows a terminal window titled "COM3" with a "傳送" (Send) button in the top right. The terminal output is as follows:

```
Nothing on it.  
55.00  
Please put sample object on it...  
Scale number: 1  
Scale number: 1  
Scale number: -206  
Scale number: -390  
Scale number: -390  
Scale number: -390  
Scale number: -390  
Scale number: -390  
Scale number: -390
```

At the bottom of the window, there are several controls: a checked checkbox for "自動捲動" (Auto scroll), an unchecked checkbox for "Show timestamp", a dropdown menu set to "NL(newline)", a dropdown menu set to "9600 baud", and a "Clear output" button.

測重程式

從校正程式計算出的比例參數套用到測重程式，之後就可得出結果

```
const int scale_factor = -390; //比例參數，從校正程式中取得
```

```
Serial.println(scale.get_units(5), 0); //設定比例參數後的數值
```

```
float weight = scale.get_units(10);
```

```
//避免出現負數
```

```
Serial.println(scale.get_units(10), 0);
```

```
if(weight<=0){
```

```
    weight = 0;
```

測重程式(使用100g砝碼)

```
COM3
99
Initializing the scale
Before setting up the scale:
24168
After setting up the scale:
-0
Readings:
-0
-0
54 ← 54是放下時瞬間感測到的重量
99
99
99
99
99
99
99
 自動捲動  Show timestamp
```



藍芽(HC-05)AT模式設定

```
#include <SoftwareSerial.h>
```

剛買來時的藍芽是原廠的設定，需設定成我們要的模式及鮑率才可使用

```
Serial.begin(9600);
```

```
Serial.println("Enter AT commands:");
```

```
BTSerial.begin(38400);
```

```
}
```

```
void loop()
```

```
{
```

```
if (BTSerial.available())
```

```
Serial.write(BTSerial.read());
```

```
if (Serial.available())
```

```
BTSerial.write(Serial.read());
```

```
}
```

持續讀取HC-05 並發送到
Arduino 串行監視器

持續從Arduino串行監視器讀取並發
送到 HC-05

設定AT模式

The image shows a serial terminal window titled "COM5". The input field contains "AT" and a "傳送" (Send) button is visible. The terminal output displays "Enter AT commands:" followed by "OK".

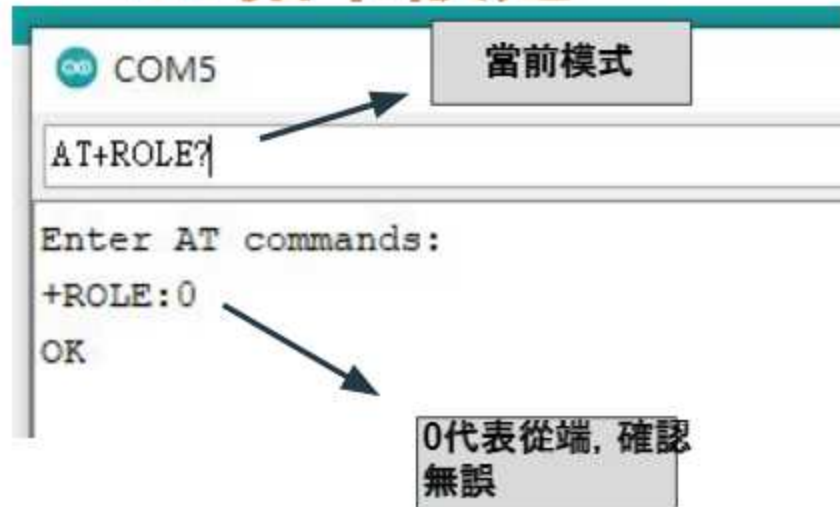
Annotations:

- A box above the input field says "輸入AT指令, 確認是否進入AT模式" (Enter AT command, confirm if entering AT mode). An arrow points from this box to the "AT" text in the input field.
- A box below the terminal output says "收到OK代表已成功進入AT模式" (Receiving OK represents successful entry into AT mode). An arrow points from this box to the "OK" text in the terminal output.
- A box to the right says "已進入AT模式代表可進行後續的AT指令設定" (Entered AT mode represents that subsequent AT command settings can be performed). An arrow points from the "OK" text to this box.

Terminal controls at the bottom include:

- 自動捲動 (Auto scroll)
- Show timestamp
- NL & CR (dropdown)
- 9600 baud (dropdown)
- Clear output

AT指令設定



COM5

當前模式

AT+ROLE?

Enter AT commands:

+ROLE:0

OK

0代表從端, 確認無誤

Detailed description: This screenshot shows a terminal window for COM5. At the top right, a grey box labeled '當前模式' (Current Mode) has an arrow pointing to the command 'AT+ROLE?' entered in the input field. Below the input field, the prompt 'Enter AT commands:' is visible. The response '+ROLE:0' is shown, with an arrow pointing to a grey box at the bottom right labeled '0代表從端, 確認無誤' (0 represents the slave end, confirmed no error). The response 'OK' is also visible.



COM5

設定鮑率

AT+UART=9600,0,0

Enter AT commands:

OK

得到回應OK

Detailed description: This screenshot shows a terminal window for COM5. At the top right, a grey box labeled '設定鮑率' (Set Baud Rate) has an arrow pointing to the command 'AT+UART=9600,0,0' entered in the input field. Below the input field, the prompt 'Enter AT commands:' is visible. The response 'OK' is shown, with an arrow pointing to a grey box at the bottom right labeled '得到回應OK' (Received response OK).

藍芽傳輸程式

```
byte Data[3];  
int insize;  
int i= weight;  
Data[0]=0;  
Data[1]=0;  
Data[2]=0;  
sprintf(str, "id 0: %d 1: %d 2: %d", i, Data[0], Data[1], Data[2]);  
//Serial.println(str);  
if (serialA == 49){  
    for(int j=0;j<3;j++)  
        I2CBT.write(Data[j]);  
  
    serialA=0;  
}
```

LCD顯示程式



W	A	T	E	R	:				9	6			C	.	C
---	---	---	---	---	---	--	--	--	---	---	--	--	---	---	---



```
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Water: ");
```

```
lcd.setCursor(9, 0);  
lcd.print(weight, 0);
```

```
lcd.setCursor(13, 0);  
lcd.print("C.C");
```

APP Inventor



APP Inventor目錄

- 第一頁版面配置
- 第二頁版面配置
- 每日應攝取水量
- 藍芽連結
- LINE Notify
- 語音功能展示
- 喝水量

第一頁版面配置



第二頁版面配置



每日應攝取水量

初始化全域變數 CM 為 0

初始化全域變數 KG 為 0

當 Screen1 初始化

執行 設 water 啟用計時 為 假

當 calculate 被點選

執行 設置 全域 KG 為 kilograme_1 文字

設置 全域 CM 為 centermeter_1 文字

設 shouldwatervalue 文字 為 $\frac{\text{取得 全域 CM}}{1000} + \text{取得 全域 KG} \times 10$

設 water 啟用計時 為 真

呼叫 對話框1 顯示警告訊息
通知 計算成功!

當 water 計時

執行 設 cup 計時間隔 為 1000

設置 全域 timer_2 為 $\text{取得 全域 timer_2} + 1$

如果 $\text{取得 全域 timer_2} = 5$

則 設 centermeter_1 文字 為 0

設 kilograme_1 文字 為 0

設 shouldwatervalue 文字 為 0

設置 全域 timer_2 為 0

設 water 啟用計時 為 假

設 shouldwatervalue 文字 為 $\frac{\text{取得 全域 CM}}{1000} + \text{取得 全域 KG} \times 10$

藍芽

當 Screen1 初始化

```
執行 設 bluetooth 啟用 為 真
    設 cup 啟用計時 為 假
    設 disconnect 啟用 為 假
```

當 bluetooth 掃描選擇

```
執行 設 bluetooth 地址 為 藍牙客戶端1 初始及名稱
```

當 bluetooth 選擇完成

```
執行 如果 呼叫 藍牙客戶端1 連接
    地址 bluetooth 藍牙客戶端1
    則 設 bluetooth 啟用 為 假
        設 disconnect 啟用 為 真
        設 cup 啟用計時 為 真
        設 bluetooth 可見性 為 假
        設 disconnect 可見性 為 真
    呼叫 藍牙客戶端1 顯示警告訊息
        通知 藍牙連接成功!
```

當 cup 計時

```
執行 如果 藍牙客戶端1 連接狀態
    則 設 nowater_value 文字 為 呼叫 藍牙客戶端1 接收文字
        位元組數 呼叫 藍牙客戶端1 取消接收位元組數
```

當 disconnect 被點選

```
執行 呼叫 藍牙客戶端1 斷開連線
    呼叫 對話框1 顯示訊息對話框
        訊息 你已中斷藍芽連線!
        標題 中斷連線
        按鈕文字 好的
    設 disconnect 啟用 為 假
    設 bluetooth 啟用 為 真
    設 bluetooth 可見性 為 真
    設 cup 啟用計時 為 假
    設置 全感 timer 為 0
```


LINE Notify



LINE

If Maker Event "LINE", then Send message

by ahlechia4412

1

Connected

Connected Oct 31, 2022
Last activity Oct 31, 2022
Run 27 times

Notify me when this runs



Check the log of your
Applet runs

View activity

Realtime Applets usually
run within 10 seconds

Check now

Your key is: `nq5CPcPQ3NkjsxHpGmfMs96GdEvEr52b50vx1KJZRt-x`

[Click to copy](#)

To trigger an Event with an arbitrary JSON payload

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/{event}/{key}/webhooks/v1/line/notify
```

*Note the `event` / `key` path element in this trigger.

With any JSON body. For example:

```
{ "text": [ [ "line", "line" ], [ "line", "line" ] ] }
```

You can also try it with `curl` from a command line.

```
curl -X POST -H 'Content-Type: application/json' -d '{"text": [ [ "line", "line" ], [ "line", "line" ] ] }' https://maker.ifttt.com/trigger/{event}/{key}/webhooks/v1/line/notify
```

Please read our [FAQ](#) on using Webhooks for more info.

Test it

To trigger an Event with 3 JSON values

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/{event}/{key}/webhooks/v1/line/notify/3
```

With an optional JSON body of:

```
{ "value1": " ", "value2": " ", "value3": " " }
```

The data is completely optional, and you can also pass `value1`, `value2`, and `value3` as query parameters or form variables. This content will be passed on to the action in your Applet.

You can also try it with `curl` from a command line.

```
curl -X POST -H 'Content-Type: application/json' -d '{"value1": " ", "value2": " ", "value3": " " }' https://maker.ifttt.com/trigger/{event}/{key}/webhooks/v1/line/notify/3
```

Please read our [FAQ](#) on using Webhooks for more info.

LINE Notify程式

初始化全域變數 key 為 模糊文字 "nq5CPcPQ3NkxHpGmfMs96GdEvEr52b50vx1KJZRt-x"

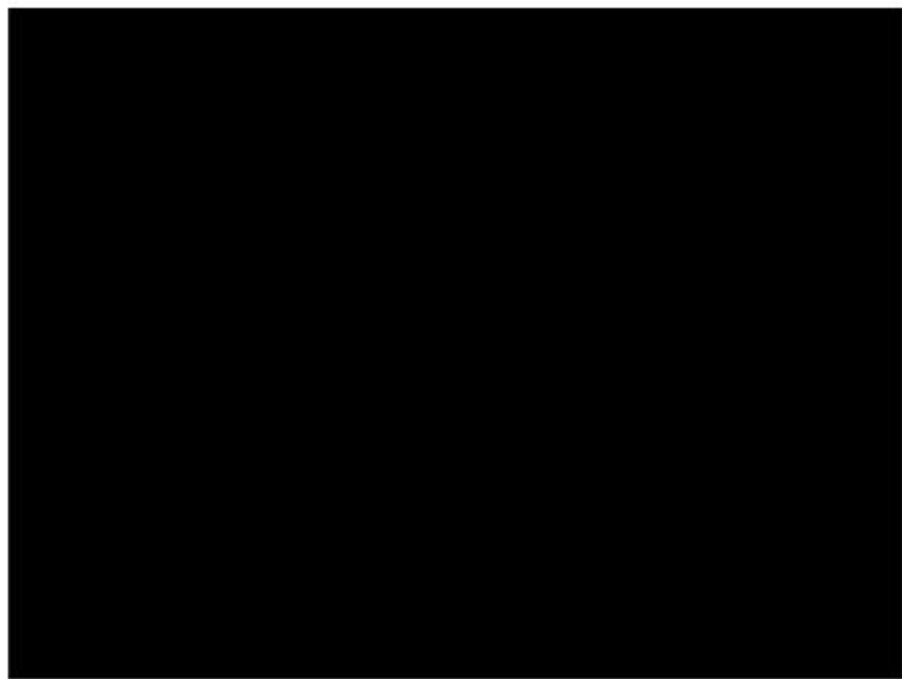
初始化全域變數 event 為 "LINE"



The image shows a Scratch script for a LINE Notify program. The script is contained within a 'when green flag clicked' event block. It begins with a 'say' block for 2 seconds. A 'when green flag clicked' block is followed by a 'set text input 1 to 0' block. A 'when timer starts' block is set to 1000. A 'set global timer to 1' block is followed by an 'if global timer is 6000' block. If true, it calls a custom block 'send message' with the message '出鼻按過水囉'. It then sets 'url 1' to a concatenation of 'https://maker.ifttt.com/trigger/' and 'global event'. It then calls 'url 1' to execute a POST text request with the text 'value1=' and '送過水囉'. Finally, it sets 'global timer' to 0.

```
當綠旗被點選時  
執行  
說 2 文字 為 取得 全域 timer  
如果 文字輸入盒1 文字 = 0  
則  
設 line 計時開端 為 1000  
設置 全域 timer 為 取得 全域 timer + 1  
如果 取得 全域 timer = 6000  
則  
呼叫 語音批露 處出文字  
消息 出鼻按過水囉  
設 網路1 網址 為 合併文字  
https://maker.ifttt.com/trigger/  
取得 全域 event  
/with/key/  
取得 全域 key  
呼叫 網路1 執行POST文字請求  
文字 合併文字  
value1=  
送過水囉  
設置 全域 timer 為 0
```

LINE Notify和語音功能展示



喝水量

初始化全域變數 `veryrealwater` 為 0

初始化全域變數 `喝後水量` 為 0

初始化全域變數 `realwater` 為 0

初始化全域變數 `模價` 為 0

初始化全域變數 `捐款水量` 為 0

初始化全域變數 `現水種` 為 0

計時

執行

- 如果 取得 全域 `water` 為 0
- 則 設置 全域 `realwater` 為 取得 全域 `water`
- 設置 全域 `veryrealwater` 為 取得 全域 `Text` + 取得 全域 `realwater`
- 說 文字輸入盒1 文字 為 取得 全域 `veryrealwater`
- 呼叫 初始水量
- 呼叫 取水
- 呼叫 喝水後水量
- 設置 全域 `realwater` 為 0

喝水量(定義過程)



確認杯墊上的水量



確定水杯被拿起來

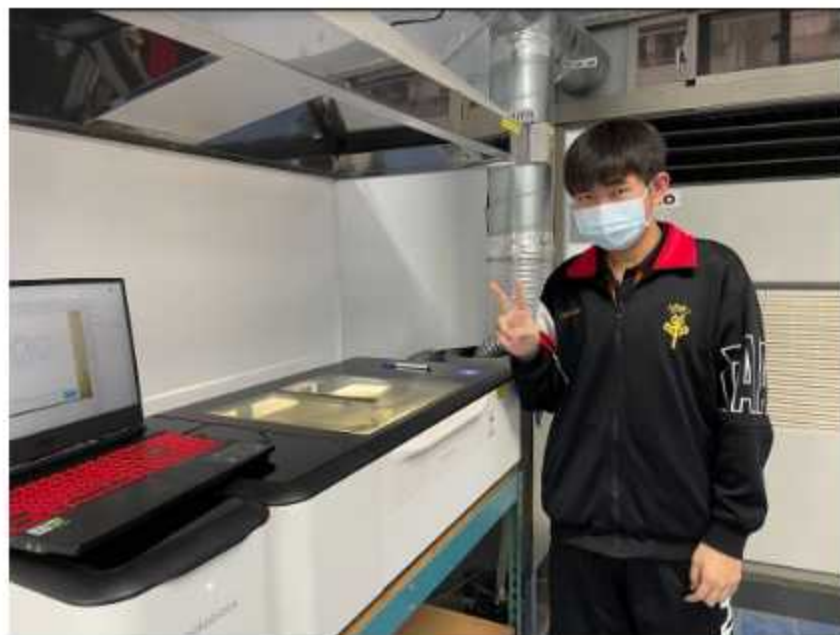
喝水量(定義過程)

The image shows a Scratch script for calculating water intake. The script is organized into several nested blocks:

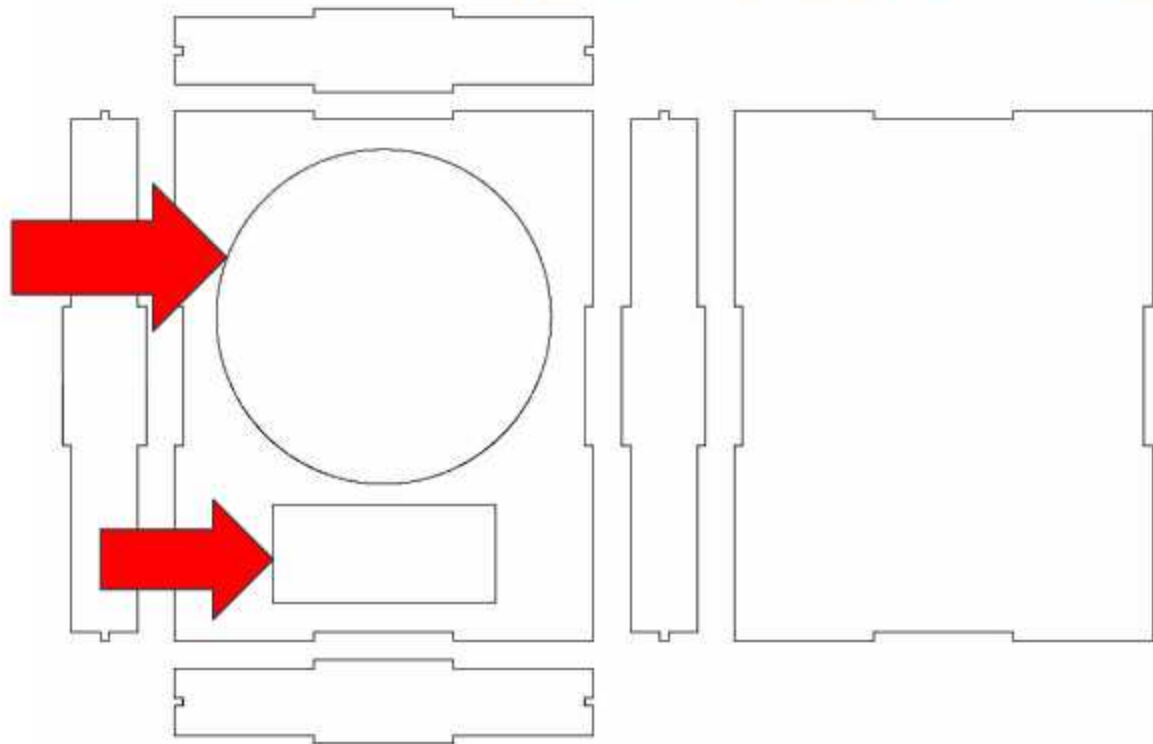
- Define Function:** A purple block labeled "定義程序" (Define Function) with the parameter "喝水量" (Water Intake). It starts with an "執行" (Execute) block.
- Conditional Logic:** An "如果" (If) block that checks if the value of "veryrealwater" is greater than 5. If true, it sets "喝水量" to 2.
- Increment Counter:** A "別" (Else) block that increments "喝水次數" (Number of drinks) by 1.
- Score Check:** A "如果" (If) block that checks if "得分" (Score) is equal to 5.
- Reset Values:** A "別" (Else) block that resets "喝水量" to the value of "veryrealwater", "喝水次數" to 0, and "喝水量" to 0.
- Comparison and Calculation:** A "如果" (If) block that checks if 0 is less than the difference between "喝水量" and "喝水量". If true, it calculates the difference between "喝水量" and "喝水量" and sets "喝水量" to that value.
- Final Calculation:** A "別" (Else) block that calculates the difference between "喝水量" and "喝水量" and sets "喝水量" to the sum of "喝水量" and "喝水量".
- Output:** A "別" (Else) block that sets "喝水量" to 1, "喝水量" to "喝水量", and "喝水量" to "喝水量".
- Input and Output:** A "別" (Else) block that sets "文字輸入量2" to "文字" and "喝水量" to "喝水量".

計算喝水量

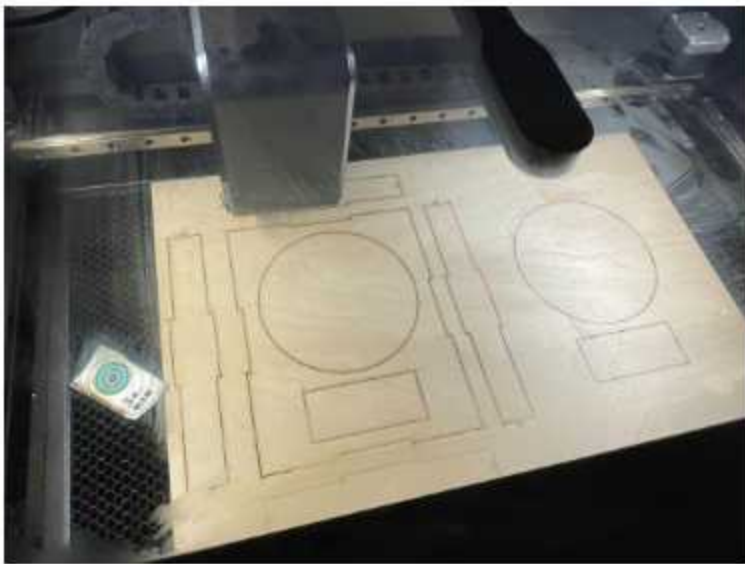
杯墊外殼設計



Laserbox設計圖(第一版)



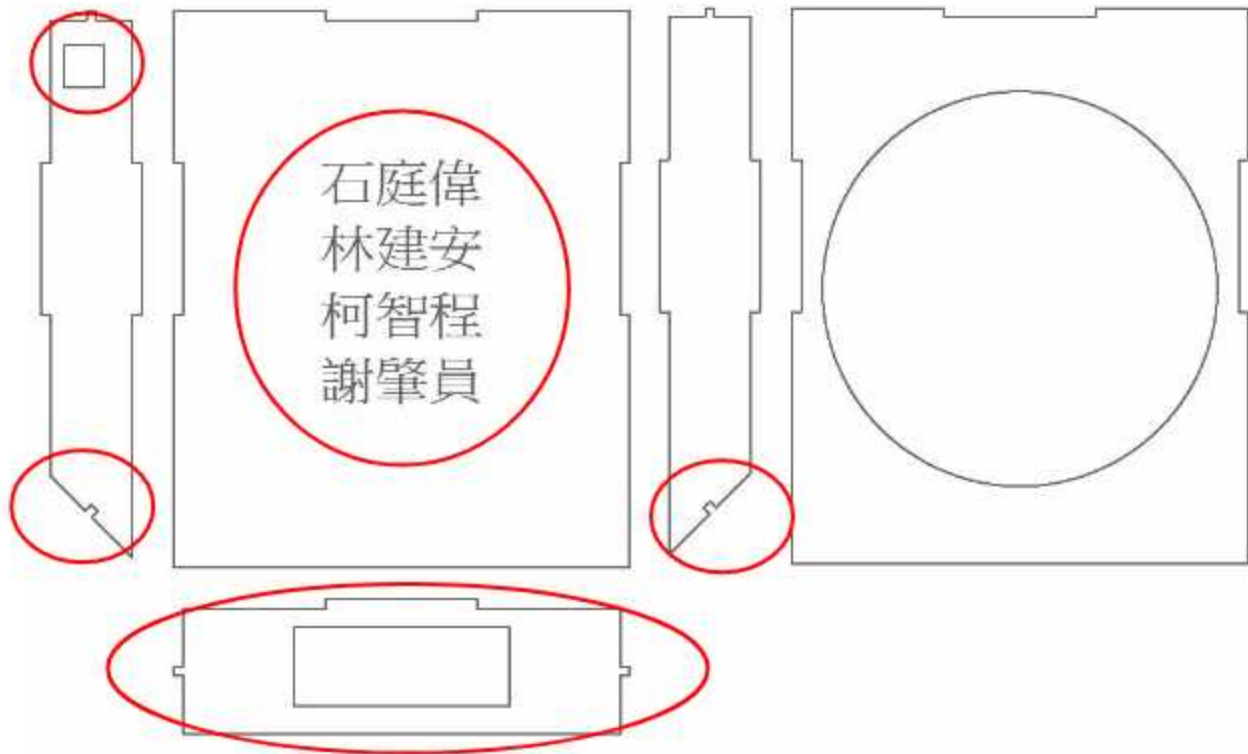
雷射場地&過程



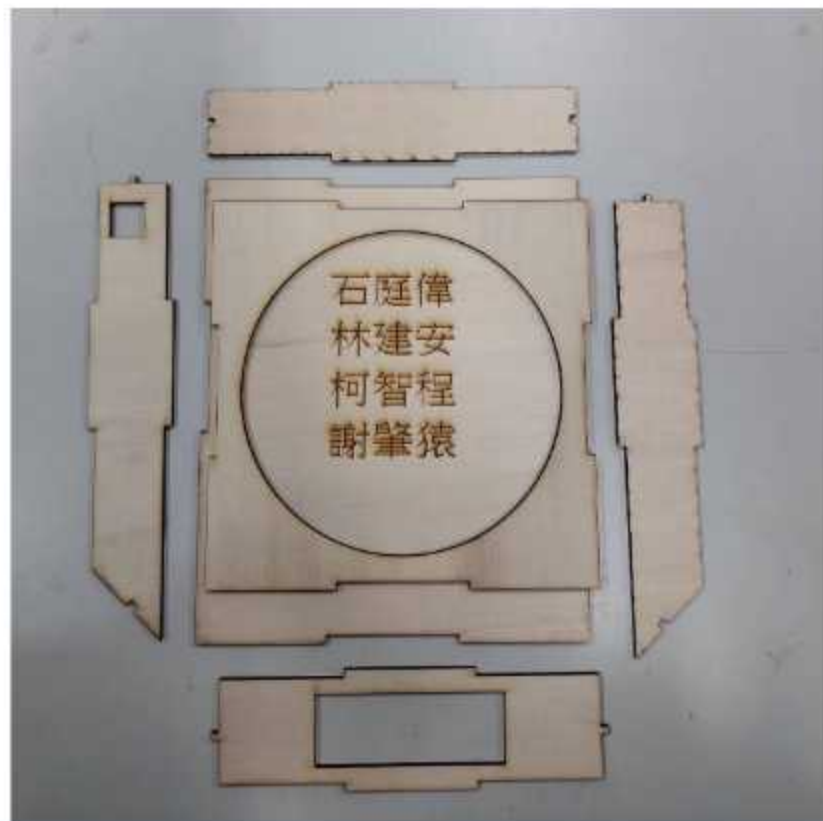
(一)完成圖



Laserbox設計圖(第二版)



(二) 完成圖



未來展望

- 1.設計為防水杯墊
- 2.外觀再修飾
- 3.可以增加紅外線偵測溫度
- 4.設計為可攜式, 用可重複充電電池供電

參考資料

https://blog.jmaker.com.tw/hx711_set/

<https://blog.jmaker.com.tw/lcd1602/>

<https://jbprogramnotes.com/2021/09/%E4%BD%BF%E7%94%A8-app-inventor-%E5%82%B3%E9%80%81%E8%A8%8A%E6%81%AF%E5%88%B0-line/>

<https://swf.com.tw/?p=712>

<https://blog.cavedu.com/2014/01/02/appinventorandarduinowithbluetooth3/>

<https://blog.cavedu.com/2017/10/18/hc05-hc06/>

<https://www.makeblock.com/cn/maker-tools/laserbox>

Q&A提問時間

報告結束