

2024仰望盃全國科學HomeRun實作大賽

決賽成果報告書

隊伍名稱： 馬克烘焙坊

作品名稱： 蛋願我們的感情滯留在這一刻

科學概念 1： 黏滯係數

黏滯係數(黏度)為剪應力以及流體中速度梯度的比值，當流體黏性越高則黏滯係數將會越大，呈現正向相關性，而在牛頓流體中，黏滯係數將會是一個使剪應力以及速度梯度呈現線性正比相關性的常數，在本研究中將對蛋白的黏滯係數相關流體性質做探討以及研究其在烘焙上的自動化應用可能性。

科學概念 2： 轉動力學

在本研究中大量使用定軸且單軸的轉動裝置，因此會使用到轉動力學的計算，物體的轉動源於合力矩的不為零，因而造成轉動的產生，而因本研究使用定軸轉動裝置，故將研究在力偶情況下兩轉動系統透過流體耦合造成角速度之間比例的差異探討，進而對流體性質做進一步討論，並延伸推廣至實際應用層面。

決賽成果報告書內文

1. 發想動機與定性的現象展示：

我是一個對烹飪抱持高度興趣的人，從以前就經常學習製作各式各樣的甜食，而在西式甜點中，不難發現許多的甜點製作過程中都摻雜有「蛋」的攪拌及打發的相關元素。各項甜點所需的蛋打發程度不同可以創造出各種不同的口感，像是鬆餅漿就不須過度打發，但是需要打至起泡以創造鬆軟口感，但是在棉花糖、蛋白霜中，蛋液就必須被打發至完全。而一般打發狀態的判斷是需要大量烘焙經驗累積而成，對我這種新手就不甚友好。同時，作為物理研究者，我不喜歡定性化的答案，因此我想追求**定量描述蛋的打發程度之可能**。

而上述的問題困擾我許久，直到最近製作瑞士馬林糖時，因為需要將蛋白以及砂糖在隔水加熱的情況下拌至溶解並打發，所以觀察到了一個有趣的現象。蛋白的黏稠度會隨著打發時間越久而增加，從一開始水狀的蛋白液，到最後直接變成一團白色綿密不易散開的流體，而當打蛋器轉軸垂直於鍋底並運作時，打蛋器本身的**轉動力矩會以蛋白為介質傳遞部分至鍋子上**，而又因鍋子外是一大鍋用於隔水加熱的熱水，所以提供了非常小的阻力，讓鍋子能因此力矩而轉動。而我簡單做了實驗觀察得出定性猜想—隨著蛋白的打發，其**流體黏滯係數隨打發時長而增加，讓打蛋器轉軸力學能耦合到鍋子轉動之轉換率提高**。

藉由此現象我覺得應該能解決一直困擾我的蛋液打發程度無法定量的問題，透過**感測器測量角速度**就可以以不同能量轉換率，也就是其黏滯係數的增加率來定義打發程度，而這就是本研究所想探討之事，在後續篇幅將具體說明理論依據以及執行方式與應用藍圖。



圖一、背景實驗執行圖

(此為背景實驗影片，能看的出來在蛋白流體本身不同狀態下其打蛋器與鍋子之間轉動系統之能量耦合的能量轉換率的定性關係)

影片連結：<https://youtu.be/THoNY9WfHEU>

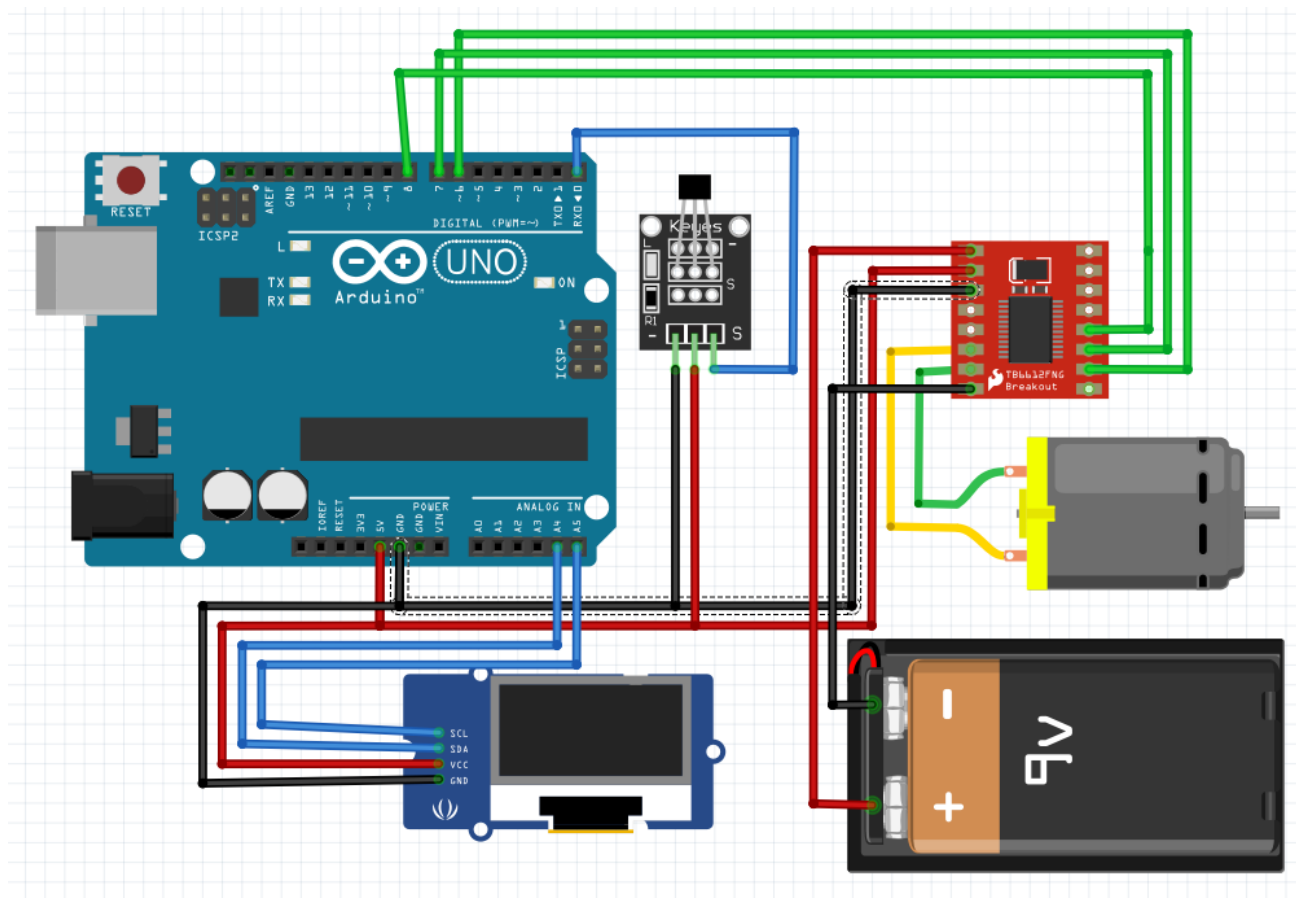
2. 作品創意性：

本作品從研究者本人不經意間的烘焙過程發想，以物理的角度思考打發蛋白一事，利用流體力學中之流體黏滯係數會是打發時間的函數一事。並由此推斷出觀察打蛋器轉動系統以及鍋子轉動系統間能量耦合的轉換率能用來定義蛋白打發程度的猜想，並進行初步實驗，觀察出原先定性的假說是符合猜想的。在實際作品中本研究結合了霍爾磁力感測器用於進行實驗以及測量角速度，結合馬達驅動板與OLED螢幕實際製作出一個自動化的打蛋器，透過物理與工程上的結合解決生活中所遇見的問題，為本研究具創意之處。

3. 硬體及電路架構圖：

本作品使用：

UNO R3主板、KY-024霍爾磁力感測器、L9110馬達驅動板、OLED螢幕(0.96吋)、碳刷馬達



圖二、硬體電路架構示意圖

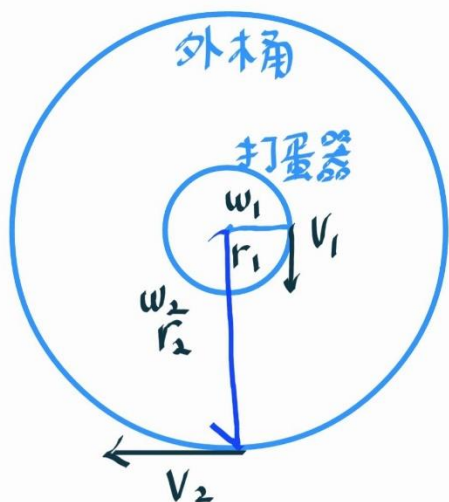
4. 理論模型：

簡單論述流體力學，黏度在其中扮演不可或缺之角色，其定義為：

$$\eta = \frac{\tau}{(dv/dy)} = \frac{(F/A)}{(dv/dy)}, \quad \eta: \text{黏滯係數}、\tau: \text{剪應力}、F: \text{受力}、A: \text{受力面積}、v: \text{速度}、y: \text{面間距}$$

也就是黏滯係數是剪應力以及速度梯度的比例常數，也是本研究的導出量(應變變因)

而根據以上定義可對本研究中流體之研製係數做推導：



(先將打蛋器近似成圓柱狀)

$$\eta = \frac{\tau / r_1 A}{(r_1 \omega_1 - r_2 \omega_2) / (r_2 - r_1)}$$

$$\text{又 } A = \int dA = \int_0^{2\pi} l r_1 d\theta = 2\pi r_1 l \quad \text{且 } \tau = r_1^2 \omega_1^2$$

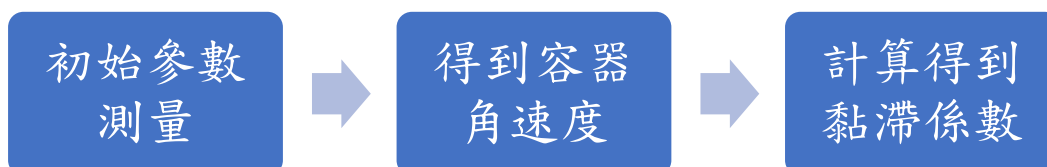
故黏滯係數可推導為以下形式：

$$\eta = \frac{m r_1^2 \omega_1^2 / 2\pi r_1^2 l}{(r_1 \omega_1 - r_2 \omega_2) / (r_2 - r_1)} = \frac{m \omega_1^2 / 2\pi l}{(r_1 \omega_1 - r_2 \omega_2) / (r_2 - r_1)}$$

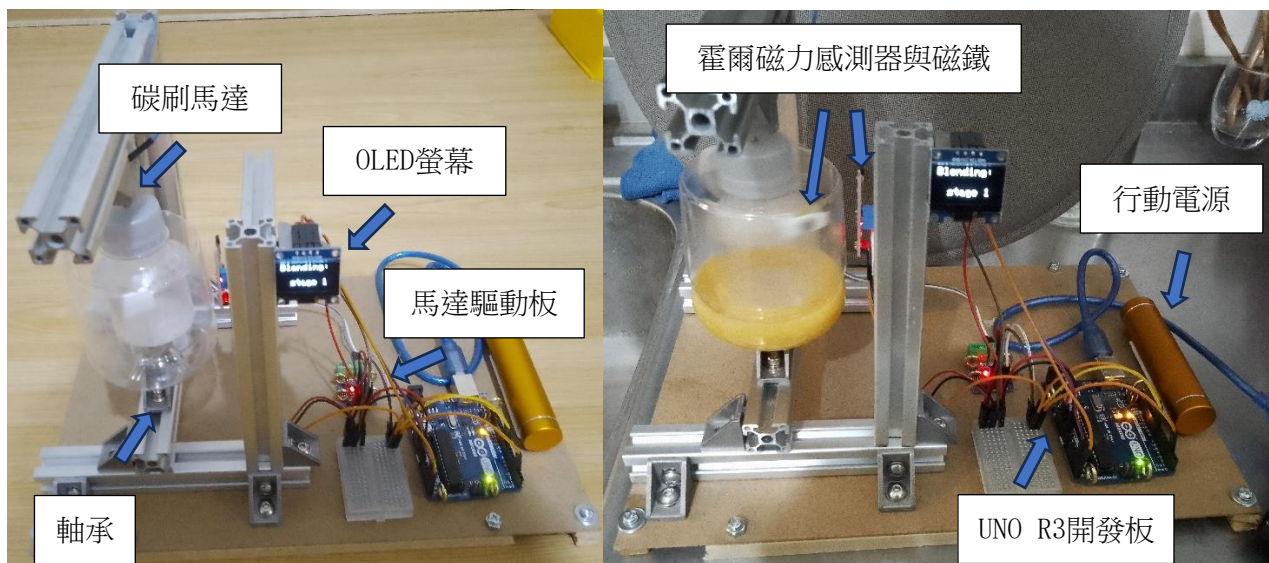
圖三、理論示意圖

(τ : 旋轉力矩、 l : 外層容器高度、 r : 半徑、 ω : 角速度)

而此處我們就可以得到內部馬達轉速(已用馬達配電板定速)以及因為馬達給予流體剪力造成對外層容器的靜力矩因而產生的旋轉角速度，並利用此二參數算出蛋液當下的黏滯係數，進而推估蛋液打發程度。



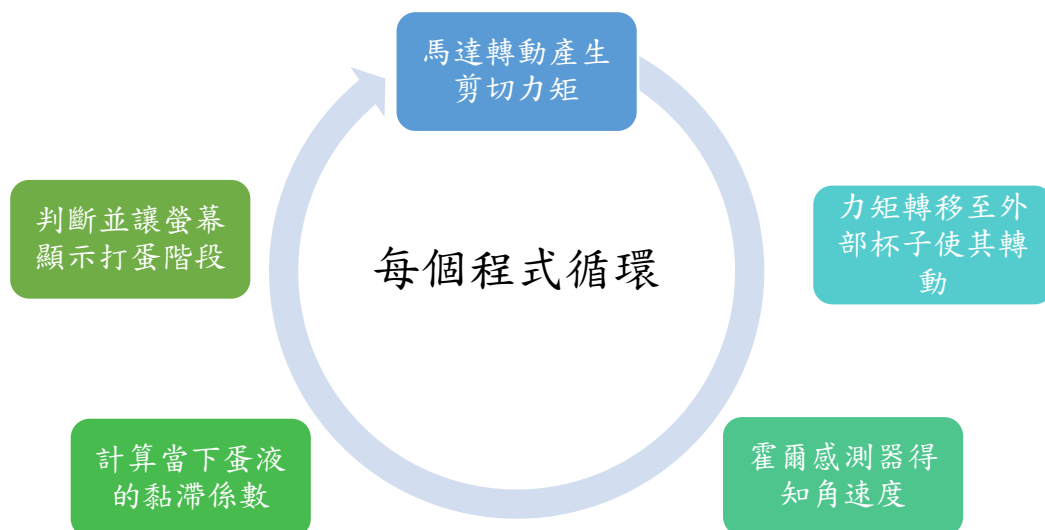
5. 作品成果說明：



圖四、自動打蛋器(未加蛋液)(左圖)與實際打蛋之運作照片(右圖)

運作邏輯簡介：

一開始給予上方馬達穩定轉速(馬達驅動板)，此時旋轉攪拌裝置開始旋轉並給予蛋液切向作用力，而又因外部容器底部裝設軸承可自由轉動，故受蛋液影響，轉動力矩轉移至外部容器，而外部容器上有磁鐵，在靠近霍爾磁力感測器時會產生數位訊號，透過程式編寫得知旋轉一圈的週期，以此計算外部容器角速度，並透過此角速度與馬達轉速及內外半徑等幾何性質計算當下蛋液黏滯係數，當其值到達一定範圍時，界定為一個階段，顯示在OLED螢幕上(此處以Stage+數字表示)，透過此種方式打造出一全自動打蛋機，使用時只需要放入蛋液並在達到所需打蛋階段時關閉機器即可。



6. 實驗與裝置改良

做為一個結合物理概念的產品研發，研發過程中最重要的必然是對該產品的測試與將參數做出報表，並對其進行改良，因此我們進行了以下實驗測試：

A. 驗證理論模型與修正：

作為所有驗證程序的第一步，驗證理論模型的可用性是非常重要的，因為在後續的所有黏度計算都將以此為基礎，所以作為背景實驗，首先來討論黏度模型計算出的值與實際值之間的偏差，並修正模型。

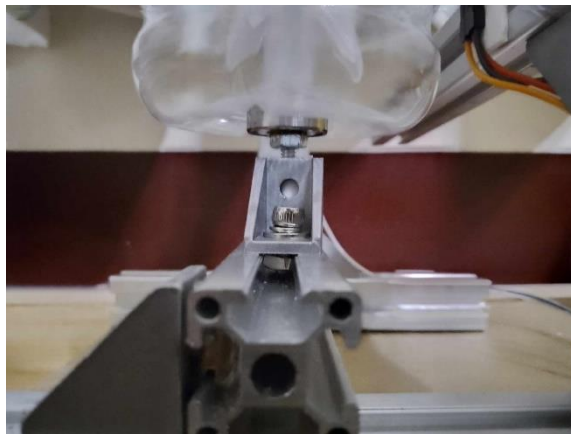
在黏度計算理論模型中，假設前提是內外接觸面皆為圓環狀，因此首先我們使用了圓柱體攪拌器來驗證理論模型，詳細參數如下：

攪拌器半徑	容器半徑	馬達角速度	容器角速度	蛋液高度	攪拌器質量
41 mm	62 mm	27 rad/s	1.56 rad/s	18.7 mm	13 g

表一、實驗參數

我們後來計算出的蛋液黏滯係數為 1.62 帕斯卡除以秒，但實際的值卻只有 1.32，足足差了 0.3，而我們後來發現此誤差為來自容器底部軸承的潤滑油阻力，因此我們雖知道黏滯係數理論模型是可用的，但必須做修正，最終我們將它加上了軸承阻力常數項 -0.3，作為理論模型的修正項，如下所示：

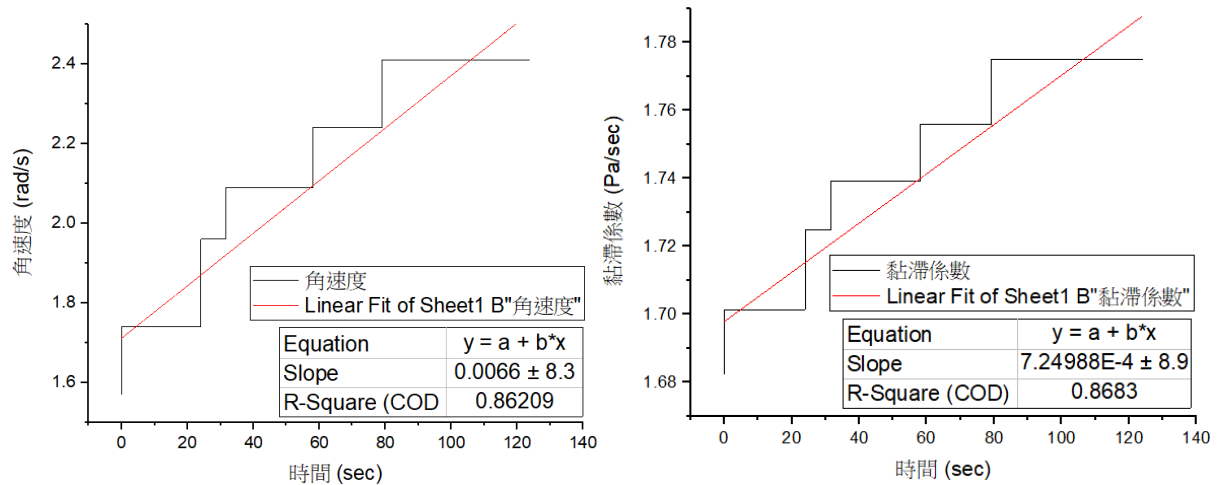
$$\eta = \frac{m\omega_1^2/2\pi l}{(r_1\omega_1 - r_2\omega_2)/(r_2 - r_1)} - 0.3$$



圖五、軸承造成阻力示意圖

B. 打發效率測定

在驗證過模型並進行修正後，下一步就是要測定打發效率作為產品參數的測量，而首先我們先進行圓柱狀打蛋器的測定，而得到如下數據：



表二、角速度與黏滯係數與時間的關係

在此處因為我們假設打發為一個空氣等速混入蛋液的過程，因此將打發程度的增長率視為常數，以一定斜率線性成長，而數據之所以是呈現階梯狀是因為霍爾磁力感測器得到的數據是**數位訊號**，同時我們訊號處理時在接收到下一個數據前將持續輸出上一個數據，一次會得到較大量的相等數據，故圖表呈現此種結果。

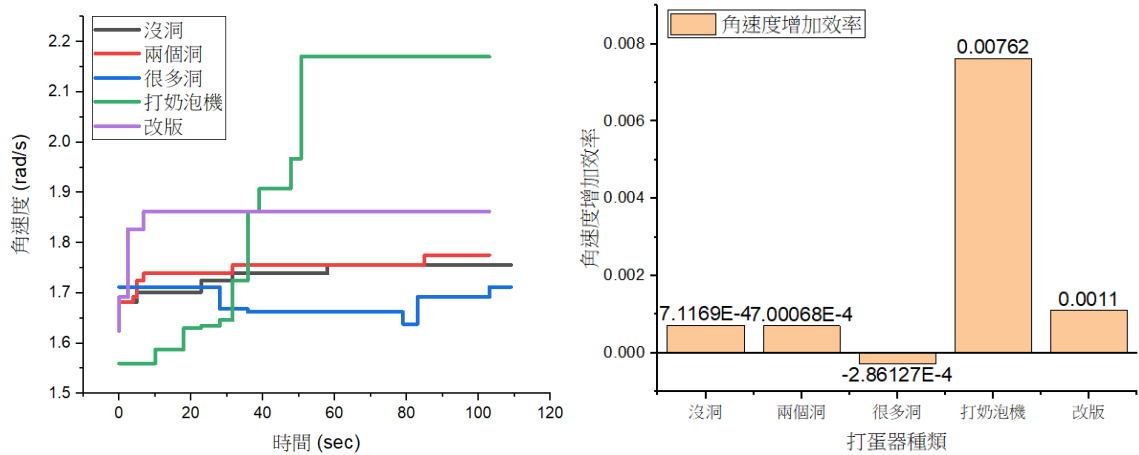
而在根據我們假設的線性擬合中，得到的黏度隨時間成長的斜率其實非常小，也代表此種打蛋器效率非常低，而我們認為主要是因為打蛋器的主要機制是讓蛋液混入空氣，但圓柱體混入空氣的效率非常低，故斜率如此的低，而在後續我們也將再做改良，而我們仍可得到所欲求之產品參數，也就是**圓柱形打蛋器的黏度增加量將與經過時間有0.000725的比例常數關係**。



圖六、圓柱形打蛋器

C. 多種打蛋器構造與其效率

因應前述所遇到的問題，圓柱形打蛋器效率實在太低，無法堪用為一個好的打蛋器，因此我們決定在打蛋器主體開孔，試試看能否在顧及理論本身假設的同時也能做出有效率的打蛋器，而結果卻不如我們所料。



表三、多版本打蛋器角速度隨時間變化及其效率比較

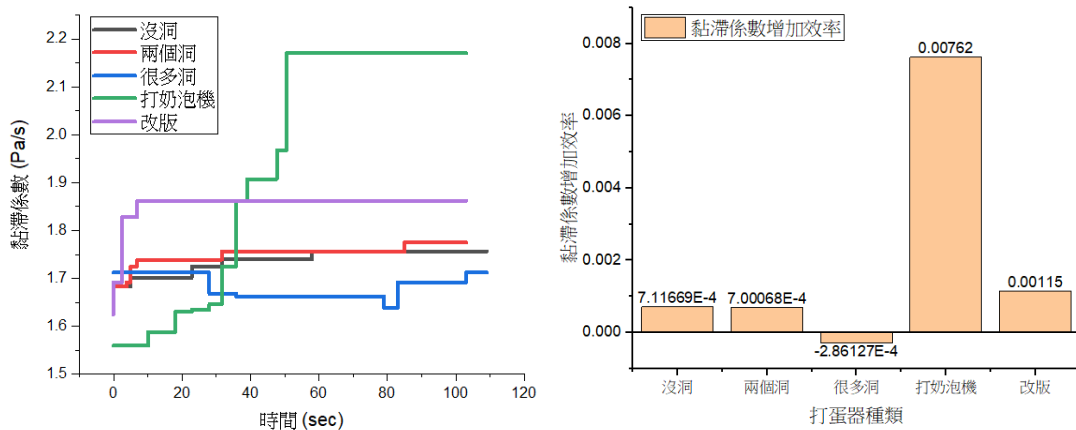
在打兩個洞的版本上，效率確實小小增加了一點，但效率仍然不夠高，而我們後來發現，因為在打蛋時，洞內大氣壓力會讓蛋液進不去，故空氣實際上沒有與蛋液混合量增加太多，因此幾乎沒增長。

後來我們又在兩個洞的版本上多用電鑽鑽了幾個孔，原本希望能讓空氣與蛋液互動更頻繁，但後來發現效率反而減少非常多，主要是因為在轉動時，蛋液會流進打蛋器中，在其中也因為所受到的角動量而旋轉，所以吸收了更多能量，造成打蛋效率不僅沒上升反而下降了非常多，實際情況如右圖所示。



圖七、多個孔的打蛋器

在比較過後我們發現圓柱狀打蛋器似乎不太可行，因此我們再度改版，將其改成無底圓柱體，並且將邊緣用成條狀，而結果似乎如我們所料的，效率提高了許多，這讓我們決定在最後以此為最終版本當作打蛋機的主體，作為最後的改良版裝設上去再打蛋機的主體上。



表四、多版本打蛋器黏滯係數隨時間變化及其效率比較



圖八、多版本打蛋器

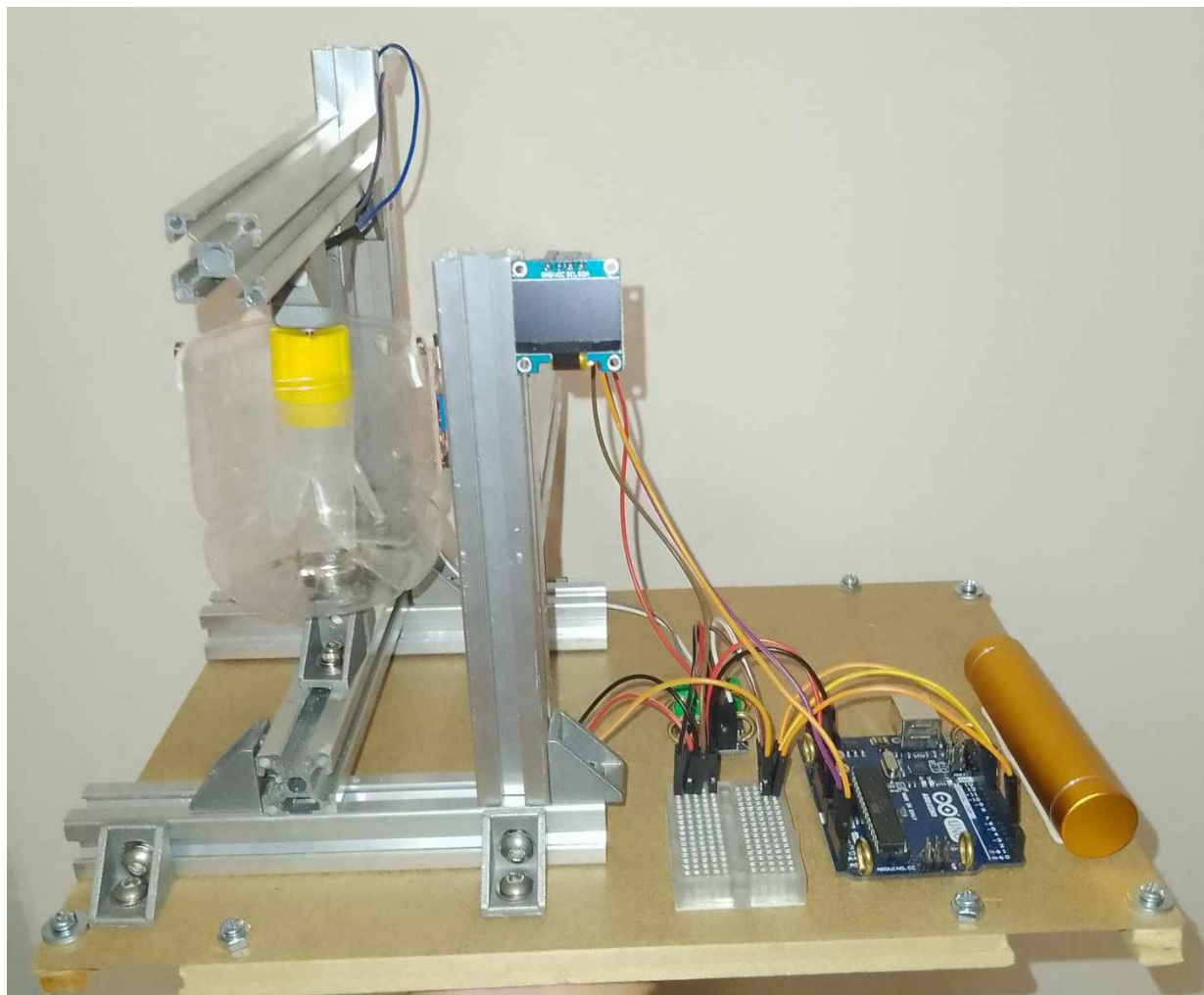
7. 結論

從一開始，我們從日常生活中觀察到了隔水加熱打蛋這件事會造成力矩從馬達透過蛋液傳遞到外部鍋子，首先進行了背景實驗，得到了黏滯係數會隨打發時間增加的結論，而後我們設計了穩定的裝置以及推導出圓環容器以及打蛋器的黏度計算理論模型，以此作為基礎做後續研究。

在研究此自製產品的參數及規格時，首先發現了我們的黏滯係數計算理論模型須修正，算出來的值與實際值差了 0.3 帕斯卡除以秒左右，而透過觀察發現其誤差原因是因為底部軸承中有潤滑油，潤滑油的阻力項在一開始計算時被我們所忽略，故此誤差應更正為軸承的阻力修正常數項，其值 0.3 帕斯卡除以秒。

在完成理論模型的修正後，我們探討了不同打蛋器模型的效率，發現圓柱形的效率極差，鑽兩個孔後情況沒太大改變，而鑽很多孔後反而更糟，因為蛋液在內部也產生旋轉，消耗能量，最終改良後發現將環狀邊切成條狀效果最好，擁有不遜色於專門打奶泡器的效率。

8. 最終產品改良



圖九、最終改版樣式

9. (附錄) Arduino 程式碼

定義與初始化:

```
1 //variables
2 unsigned long period = 0;
3 float angularVelocity = 0.0;
4 bool prestate = 0;
5 bool state = 0;
6 float t1=0;
7 float t2;
8 float viscosity;
9 float mass;
10 float internalomega;
11 float internalradius;
12 float externalradius;
13 float fluidheight;
14
15
16 //libraries
17 #include <SPI.h>
18 #include <Wire.h>
19 #include <Adafruit_GFX.h>
20 #include <Adafruit_SSD1306.h>
21
22 //initial setups
23 #define A_1A 8
24 #define A_1B 9
25 #define Speed 10
26 #define SCREEN_WIDTH 128
27 #define SCREEN_HEIGHT 64
28 #define OLED_RESET -1
29 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

顯示螢幕畫面設定:

```
31 //display setups
32 void stageone(void) {
33
34     display.clearDisplay();
35     display.setTextSize(2);
36     display.setTextColor(1);
37     display.setCursor(5,0);
38     display.print("Blending:");
39     display.setCursor(26,40);
40     display.print("stage 1");
41     display.display();
42 }
```

主程式(初始設定):

```
66 //main program
67 void setup() {
68     Serial.begin(9600);
69     if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
70         Serial.println(F("SSD1306 allocation failed"));
71         for(;;);
72     }
73     pinMode(A_1A,OUTPUT);
74     pinMode(A_1B,OUTPUT);
75     digitalWrite(A_1A,LOW);
76     digitalWrite(A_1B,LOW);
77     stageone();
78     pinMode(2, INPUT);
79 }
```

主程式(執行迴圈):

```
81 void loop() {
82
83     //hall sensor section
84     state=digitalRead(2);
85     if(prestate == 1 && state == 0)
86     {
87         t1=millis();
88     }
89     if(prestate == 0 && state == 1)
90     {
91         t2=millis();
92     }
93     if(t2-t1>0)
94     {
95         float period = (t2-t1)/1000;
96         angularVelocity = 3.14/period;
97     }
98     Serial.println(angularVelocity);
99     prestate=state;
100
101     //viscosity calculation
102     viscosity = (mass*internalomega*internalomega/6.28*fluidheight)/(internalradius*internalomega-externalradius*externalomega)*(externalradius-internalradius)-0.3
103
104     //display section
105     if(viscosity > 1.3 && viscosity < 1.7){
106         stageone();
107     }
108     if(viscosity > 1.7 && viscosity < 2){
109         stagetwo();
110     }
111     if(viscosity > 2){
112         stagethree();
113     }
114
115     //data collection delay
116     delay(100);
117 }
118
119
120
121 }
```

10. 參考文獻:

Let there be math. (2017, October). *Fluid Mechanics*. YouTube.

<https://www.youtube.com/playlist?list=PLw7aJpYJwwNUAKpfiDIwikNnvfVsGw2tr>

黏滯係數之測定. (n.d.). 國立中央大學物理學系.

http://generalphysicslab.phy.ncu.edu.tw/phy_lab/viscosity.htm