

中華民國第 57 屆中小學科學展覽會

作品說明書

高級中等學校組 工程學科(一)科

佳作

052320

自製 Arduino 比色計

學校名稱：國立新竹女子高級中學

作者： 高二 洪如瑩 高二 康邁文	指導老師： 徐以誠
---------------------------------	------------------

關鍵詞：Arduino、色彩感測器、比色計

壹、摘要：

本研究利用單晶片微控制器 Arduino、色彩感測器 TCS3200、散光型白光 LED、電阻、電容、電晶體、筆記型電腦等器材製作出一簡易的比色計。先透過限流電阻及補償電容的更換交叉測試，尋找到系統所能提供最穩定光源的組合，然後利用一些已知濃度溶液的紅色、綠色、藍色、強度偵測值，用多元多項式迴歸分析法尋找出該溶液的最佳濃度估測函數，以後只要測得未知濃度溶液的偵測值，即可以此估測函數計算出該溶液的濃度。經由硫酸銅及硫氫鐵離子等溶液的實際測試，平均的估測誤差分別為 2.75% 及 3.77%，可見我們自製的比色計確實精確有效。

貳、研究動機與目的：

每次到化學實驗室時，總會看見桌上擺放著許多的硫酸銅廢液。但往往因溶液放置過久或其他因素，導致濃度改變。因此，為了減少此種浪費，我們決定測出其濃度，以利重新利用。但若是用滴定法，會改變其溶液內容，所以我們想藉其顏色變化來推測其濃度。可是市面上之光譜儀價格昂貴且攜帶不便，剛好日前資訊課提到 Arduino 之相關應用，所以我們想用 Arduino 自製一個精準、簡易又便宜的比色計。

參、器材及軟體：

- 一、筆記型電腦。
- 二、Visual Basic.Net 2012。
- 三、Arduino Uno 板、色彩感測器 TCS3200、杜邦線。
- 四、電路板、白光 LED、電阻、NPN 電晶體、電解電容、烙鐵、焊錫、木板。
- 五、藥品： $CuSO_4$ 、 $Fe(NO)_3$ 、 $KSCN$ 、 HNO_3 、 HCl 、蒸餾水。
- 六、電子天秤、燒杯、容量瓶、漏斗、玻棒、刮勺、滴管、秤量紙、鋁箔紙。

肆、研究過程：

一、相關研究

歷屆的科學展覽作品中，關於自製比色計的共有四篇（參考資料[6][7][8][9]），分別提出了不同的裝置設計及估測方法來偵測有色溶液的濃度，綜觀這些報告所使用的策略，都是在暗箱中，先以光源來照射比色管中已知濃度的有色溶液，並以光敏電阻為感測器，將測得的電阻值紀錄下來，然後找出溶液濃度與電阻值之間的關係（趨勢線或檢量線），之後，就可以利用此關係式，由感測器所測得之電阻值來估測未知濃度溶液的濃度。

然而，這些方法在操作上並不是很方便，需要許多額外的輔助器材，如光源裝置、三用電表、電源供應器等，除了接線較為繁複外，還需透過關係式的計算，無法直接判讀待測溶液的濃度。雖然如此，我們可由這些研究報告中得知，溶液的偵

測環境、光源的種類、濃度感測裝置及濃度估測函數是影響有色溶液濃度偵測精確度的關鍵。

二、比色計的設計與實作

最近高中校園的科學教育正吹起一股自己動手做、改造世界的自造者（Maker）風潮，其中單晶片微控制器 Arduino 正是這波風潮中的最主要元件，可讓青年學子發揮創意作出許多讓人眼睛一亮的作品，創造出更多不同的可能性。

Arduino 可以搭配各種類型的外接感測器及電子元件，透過程式上傳至晶片後，而設計出與真實世界互動的應用作品。在眾多的外接感測器中，TCS3200 引起了我們的興趣，它能夠偵測出外界物體的色彩值，似乎可以稍做修改來當做比色計的濃度感測裝置，而且 Arduino 亦可透過 USB 界面與電腦做連線互動，意謂著能夠即時地顯示待測溶液的色值並進而估測其濃度。

1. 單晶片微控制器 Arduino

Arduino 是由義大利米蘭互動設計學院所設計出來的開放原始碼的單晶片微控制器電路板，使用類似於 C 語言的程式開發環境，其目的是希望計師能夠透過 Arduino 板及外接元件，快速地開發出可與真實世界互動的應用產品。圖 1 為 Arduino Uno 板，這是大多數入門者擁有的第一塊控制板，它使用了 ATMEL 公司的 ATmega328 當作微控制器，具有 14 支數位輸出入腳及 6 支類比輸入腳，可以透 USB 連接埠與電腦連接，除了供電外也可與電腦進行訊息互動。

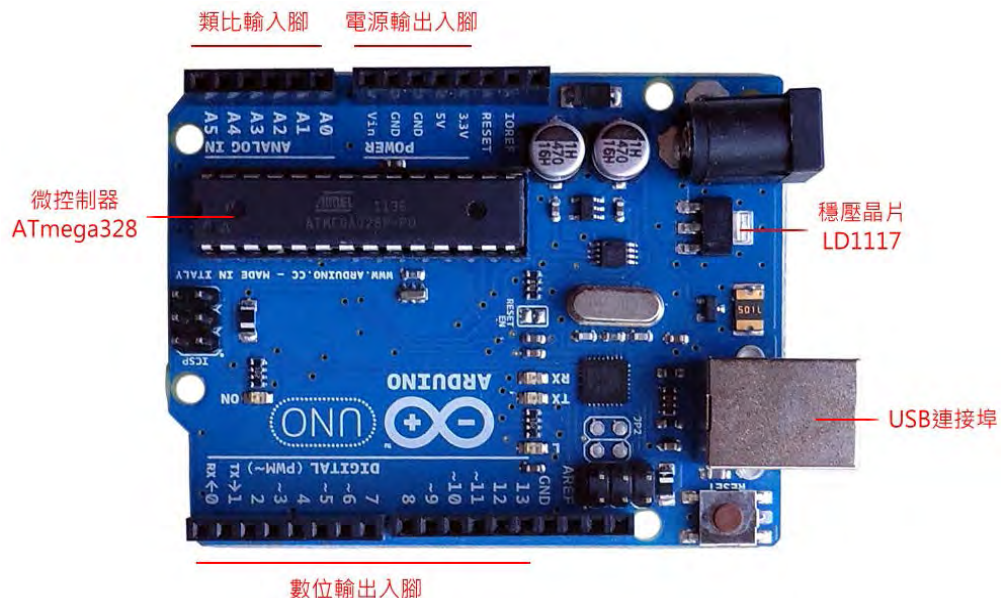


圖 1 Arduino UNO 板（圖片取自 Arduino 官方網站 www.arduino.cc）

2. 色彩感測器 TCS3200

色彩感測器 TCS3200 的主要元件有 TAOS 公司（現屬於 AMS 奧地利微電子公司）的 TCS3200 晶片和 4 個白光 LED，晶片中設置有紅、藍、綠三種濾波器，能夠分析物體反射之光線色彩值，並回傳至單晶片微控制器 Arduino。



圖 2 色彩感測器 TCS3200 之正反面（CJMCU 的 TCS3200 產品，圖片為自行拍攝）

依照參考資料[5]的產品規格說明書，我們可以藉由改變 S0~S3 接腳的高低電位來設定偵測的頻率及濾波器類型，並由 OUT 接腳來分別取得反射光線的紅、綠、藍、強度等四個成分值，相關的設定方式如表格 1 及表格 2 所示。

S0	S1	OUTPUT 偵測頻率比例
Low	Low	關機
Low	High	2%
High	Low	20%
High	High	100%

表格 1 偵測頻率的選擇模式

S2	S3	濾波器類型
Low	Low	紅色
Low	High	藍色
High	Low	強度（無濾波器）
High	High	綠色

表格 2 濾波器的選擇模式

3. 比色計的構想與實作

比色計的設計構想圖如圖 3 所示，我們使用最常見的 Arduino Uno 板當作比色計的微控制器，其主要的工作就是等待電腦透過 USB 界面傳送過來的訊號，再控制其外接的色彩感測器 TCS3200 去讀取有色溶液的色值，然後將讀取到的色值回傳給電腦，電腦端的應用程式即可用濃度估測函數來判讀溶液的濃度。



圖 3 比色計的設計構想圖

色彩感測器 TSC3200 上的白光 LED 是用來發射白光照射在待測物體上面，然後偵測反射光線的色值來判定物體的顏色，這與一般比色計採用光線穿透式的偵測方法有所不同，所以我們先將色彩感測器上的白光 LED 取下，將其焊接在電路板上，為配合比色管的高度，僅使用了三顆白光 LED，最後再以導線將白光 LED 與其原先的焊點連接即可，整個比色計的接線圖如圖 4 所示，色彩感測器 TSC3200 的各個接腳則使用杜邦線與 Arduino Uno 板連接。

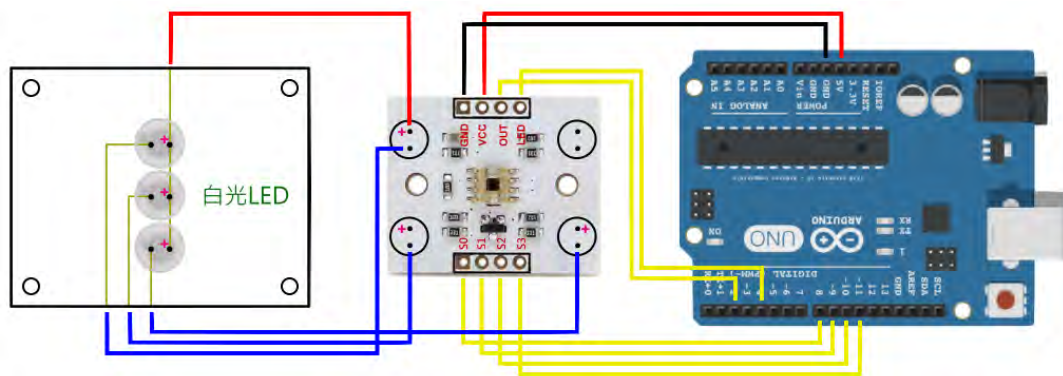


圖 4 比色計接線圖

另外，為了避免外界光線的干擾，我們將這些元件置入木製的暗箱中，僅留比色管抽換孔及 USB 插座以利更換比色管及與電腦連接，圖 5 為完成後的自製 Arduino 比色計。

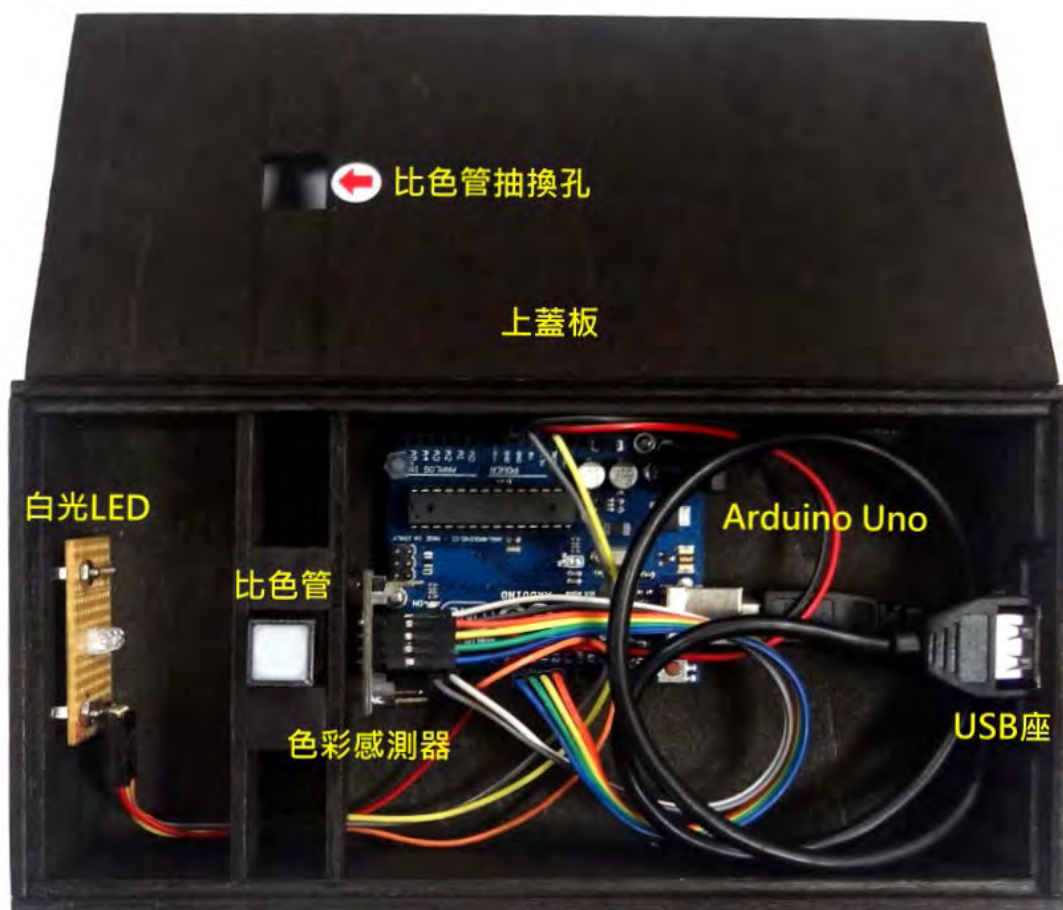


圖 5 自製 Arduino 比色計裝置圖

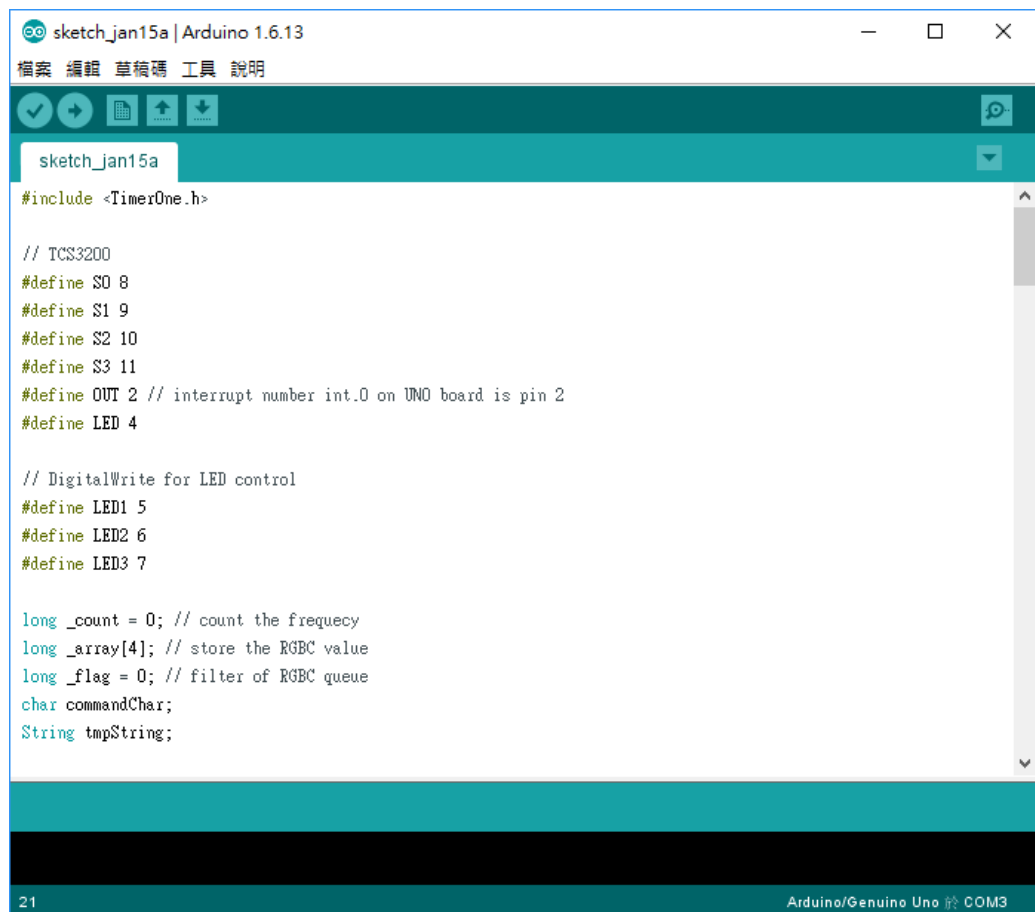
三、穩定性測試與光源的設計

要能夠精準的判定待測溶液的濃度端賴色彩偵測值的準確性，能夠搭配色彩感測器的穩定白光 LED 光源是不可或缺的要件。我們藉由自行撰寫的程式，先測試前面實作出來的比色計（光源由色彩感測器 TCS3200 供電），然後再嘗試著尋找是否有更穩定色彩偵測的元件組合。

1. 穩定性測試之程式設計

(1) Arduino 端的程式設計

Arduino 端的程式分成兩個函式，`setup()` 函式主要是用來初始化變數、初始化串列埠、定義與設定 Arduino 的數位接腳。`loop()` 函式則設計成等待電腦端透過串列埠傳送過來的訊息，依訊息指示而開始或是停止色彩偵測。我們設計成每 2.5 秒鐘完成一次溶液穿透光線的色彩偵測，紅色、綠色、藍色、強度各測 0.5 秒，休息 0.5 秒鐘，然後將測得的色彩值透過串列埠傳送給電腦端，如此循環偵測與傳送，直到電腦端傳送過來「停止偵測」的訊息為止，程式碼詳細請見附錄一「穩定性測試 Arduino 端的程式設計」。



```
sketch_jan15a | Arduino 1.6.13
檔案 編輯 草稿碼 工具 說明

sketch_jan15a

#include <TimerOne.h>

// TCS3200
#define S0 8
#define S1 9
#define S2 10
#define S3 11
#define OUT 2 // interrupt number int.0 on UNO board is pin 2
#define LED 4

// digitalWrite for LED control
#define LED1 5
#define LED2 6
#define LED3 7

long _count = 0; // count the frequency
long _array[4]; // store the RGB value
long _flag = 0; // filter of RGB queue
char commandChar;
String tmpString;
```

21 Arduino/Genuino Uno 於 COM3

圖 6-1 使用 Arduino IDE 開發 Uno 板上的色彩偵測程式

(2) 電腦端的程式設計

我們以 VB.NET 來開發電腦端的程式，執行畫面如圖 6-2 所示，先將電腦系統中有裝置連接的串列埠加入下拉式選單，可供使用者選擇本程式要與哪一個串列裝置連接；另外提供了 3 個 **RadioButton**，供使用者挑選要進行的色彩偵測次數；開始偵測後，**TextBox** 可以顯示 **Arduino** 回傳過來的色彩偵測數據資料；當偵測完成，按下 **Save** 按鈕即可將 **TextBox** 中的數據資料存成文字檔，以利進行後續的穩定度分析。程式碼請見附錄二「穩定性測試電腦端的程式設計」。



圖 6-2 電腦與 Arduino 的訊息交換程式

(3) 穩定性測試的注意事項

在比色管中裝入蒸餾水後，即可進行元件的穩定性測試，每個新元件進行 3 回合的測試，每回合測試各進行 300 次的連續色彩偵測，各回合間需關機休息 30 分鐘。元件在偵測溶液色彩值時是否穩定，有下列兩項重要的觀察指標：

- 同一回合測試的每個偵測值 x_i 與平均值 \bar{x} 間的絕對誤差 $|x_i - \bar{x}|$ 之平均值 $\sum_{i=1}^{300} |x_i - \bar{x}| / 300$ 越小越好，表示在偵測的過程中，感測器偵測數據的能力很穩定。
- 經過 3 回合的測試後，3 個偵測平均值間的標準差越小越好，表示每次開機，感測器在偵測同樣濃度液體時，數據一致性很高。

2. 光源由色彩感測器 TCS3200 供電的穩定性測試

本節將進行實作之比色計的穩定性測試，其白光 LED 光源由色彩感測器 TCS3200 供電（接線如前面圖 4 所示），由於 TCS3200 均使用表面安裝元件（SMD，Surface Mount Devices），體積小而不易識別其種類與參數，經查詢參考資料[5]後，白光 LED 的電路設計如圖 7 所示。

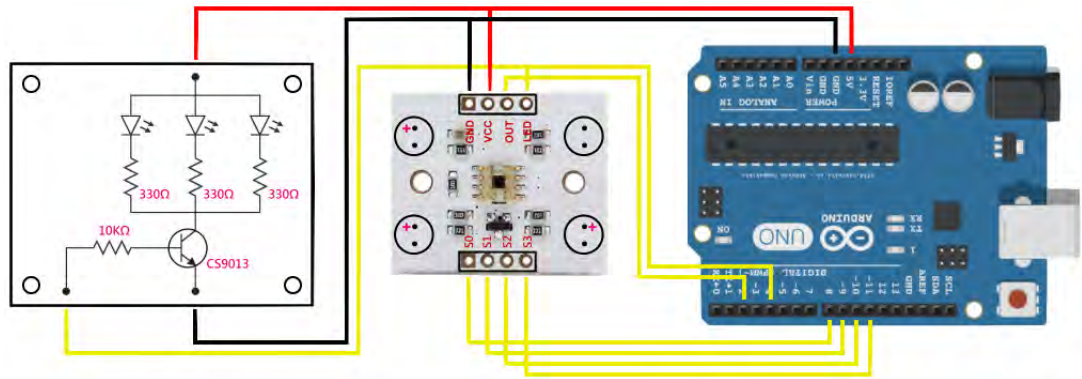


圖 7 實作之比色計的等效電路圖

穩定性測試的結果數據列於表格 3，三個回合之平均絕對誤差的平均值百分比為 0.0204%，三回合偵測平均值間標準差的百分比為 0.0219%。三個回合測試之色彩偵測值的折線圖如圖 8 所示，在此我們有將色彩偵測值做正規化處理，即每個偵測值均除以該回合 300 次偵測的平均值，即 x_i / \bar{x} ，這樣比較容易看出每個偵測值與平均值間的比例關係，而且也不受偵測平均值大小不一致的影響。

由測試結果看來，整個色彩偵測系統算是穩定的，三個回合各 300 次的偵測值幾乎都介於平均值的 $\pm 0.1\%$ 之間。

回合	項目	R	G	B	I	平均
第 1 回合	300 次偵測平均值	2306.1	2238.9	7216.1	2826.2	
	絕對誤差平均值	0.44	0.46	1.71	1.11	
	絕對誤差平均值百分比	0.0192%	0.0206%	0.0237%	0.0392%	0.0257%
第 2 回合	300 次偵測平均值	2306.7	2239.1	7216.8	2826.4	
	絕對誤差平均值	0.41	0.34	0.96	0.71	
	絕對誤差平均值百分比	0.0177%	0.0151%	0.0133%	0.0252%	0.0178%
第 3 回合	300 次偵測平均值	2305.6	2238.3	7214.1	2825.1	
	絕對誤差平均值	0.49	0.43	0.62	0.62	
	絕對誤差平均值百分比	0.0211%	0.0193%	0.0086%	0.0218%	0.0177%
三個回合之平均絕對誤差的平均值		0.45	0.41	1.10	0.81	
三個回合之平均絕對誤差的平均值百分比		0.0193%	0.0183%	0.0152%	0.0288%	0.0204%
三回合偵測平均值的平均值		2306.1	2238.8	7215.7	2825.9	
三回合偵測平均值間的標準差		0.58	0.41	1.43	0.68	
三回合偵測平均值間標準差的百分比		0.0254%	0.0183%	0.0199%	0.0240%	0.0219%
供電電壓 VCC：4.43V LED 順向電壓：2.81V、2.84V、2.79V 電阻電壓降：1.62V、1.59V、1.64V 供電電流 I：14.70mA（4.91mA、4.82mA、4.97mA）						

表格 3 光源由色彩感測器 TCS3200 供電之穩定性測試數據

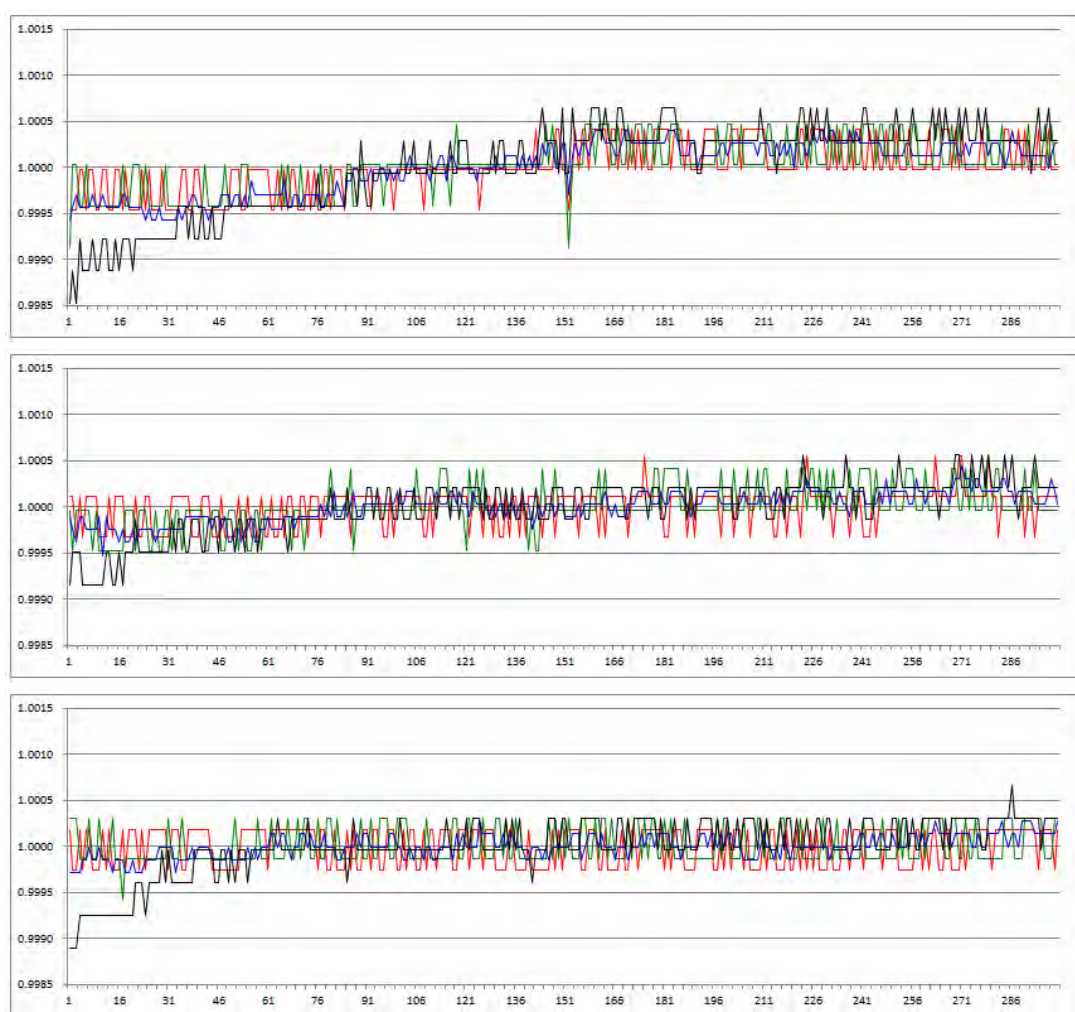


圖 8 光源由色彩感測器 TCS3200 供電的之正規化色彩偵測值的折線圖

3. 光源由 Arduino VCC 供電的穩定性測試

光源由色彩感測器 TCS3200 負責供電時，其色彩偵測的穩定性似乎不錯，但是經由測量 LED 的順向電壓分別為 2.81V、2.84V、2.79V，工作電流則為 4.91mA、4.82mA、4.97mA，這與一般白光 LED 的工作電壓 2.9V~3.4V，工作電流 10mA~20mA，有不小的差距，意味著白光 LED 並沒有完全地發揮其亮度，這在測量濃度較高、顏色較深的溶液時，光源會無法穿透而影響到感測器的色彩偵測。

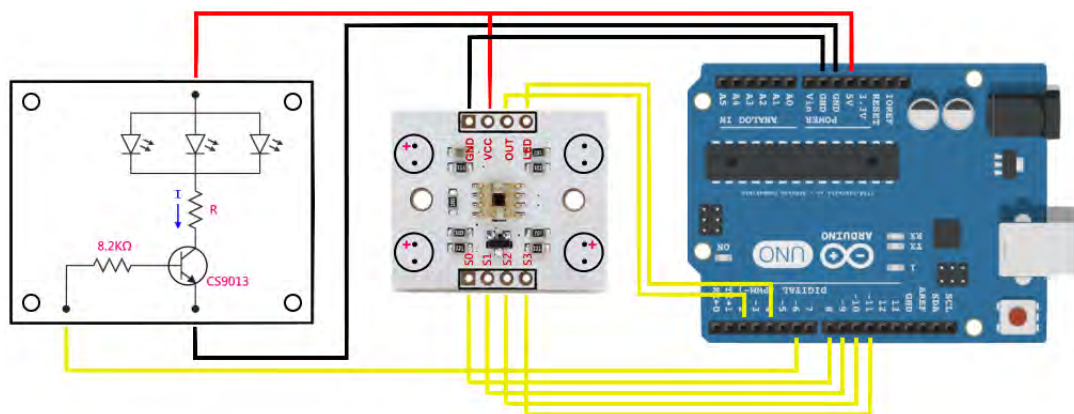


圖 9 修正後的比色計接線圖

要提高 LED 的順向電壓及工作電流，最簡單的方法就是降低限流電阻的電阻值，但要在色彩感測器 TCS3200 上更換體積很小的 SMD 電阻，是一件非常困難的工作，所以比較簡便的作法是將限流電阻及作為開關的 CS9013 電晶體直接與 LED 放在同一塊電路板上，光源由 Arduino 板上的 VCC 直接供電，電晶體的基極開關由 Arduino 板的數位輸出腳控制，修正後的比色計接線圖如圖 9 所示，圖 10 為實作出來的電路板。

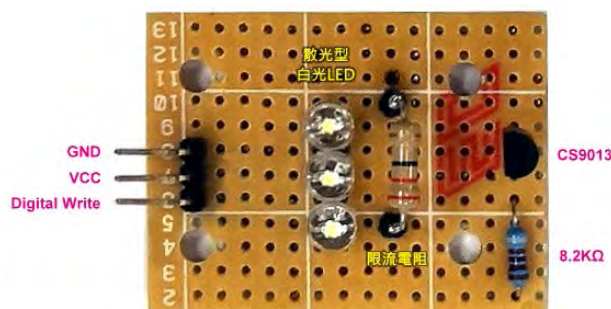


圖 10 光源電路板實作

為了瞭解白光 LED 在什麼樣的順向電壓時，可與 TCS3200 搭配而有最穩定的色彩偵測能力，我們置換了不同電阻值的限流電阻（110 Ω 、51 Ω 、39 Ω 、33 Ω 、30 Ω 、20 Ω ），依序進行穩定性測試，實測數據列於表格 4（最後一欄為前

節的測試結果，參考比較用），整體看來 30 Ω 的限流電阻具有較佳的表現，三個回合之平均絕對誤差的平均值百分比為 0.0100%，三回合偵測平均值間標準差的百分比為 0.0206%。三個回合測試之色彩偵測值的折線圖如圖 11 所示。

項目 \ 電阻 R	110 Ω	51 Ω	39 Ω	33 Ω	30 Ω	20 Ω	TCS3200
供電電壓 VCC	4.42V	4.39V	4.38V	4.37V	4.36V	4.31V	4.43V
LED 順向電壓	2.86V	2.94V	2.98V	3.02V	3.03V	3.10V	2.81V 2.84V 2.79V
電阻電壓降	1.56V	1.45V	1.40V	1.35V	1.33V	1.21V	1.62V 1.59V 1.64V
供電電流 I	14.18mA	28.43mA	35.90mA	40.91mA	44.33mA	60.50mA	14.70mA
三個回合平均絕對誤差的平均值	0.0230% (7)	0.0158% (4)	0.0161% (5)	0.0121% (2)	0.0100% (1)	0.0156% (3)	0.0204% (6)
三回合偵測平均值的平均值	R: 2194.0 G: 2150.6 B: 6929.8 I: 2729.2	R: 3856.7 G: 3765.4 B: 12258.8 I: 4896.1	R: 4563.4 G: 4443.9 B: 14551.2 I: 5838.5	R: 5028.1 G: 4896.2 B: 16013.9 I: 6459.5	R: 5399.8 G: 5263.2 B: 17264.9 I: 6986.5	R: 6642.3 G: 6455.2 B: 21326.8 I: 8688.6	R: 2306.1 G: 2238.8 B: 7215.7 I: 2825.9
三回合偵測平均值間標準差	0.0251% (5)	0.0292% (6)	0.0794% (7)	0.0224% (4)	0.0206% (2)	0.0194% (1)	0.0219% (3)

表格 4 比色計的穩定性測試數據（不同限流電阻）

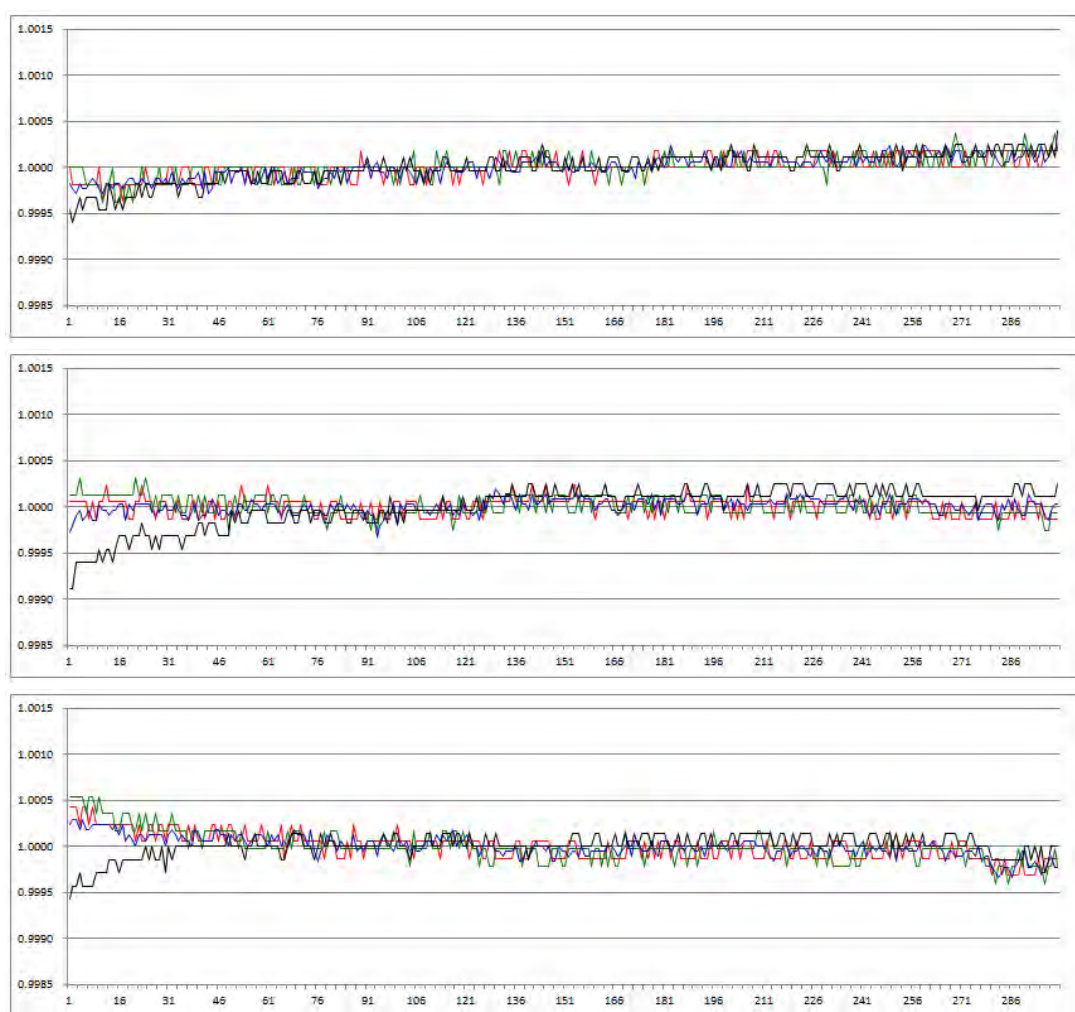


圖 11 光源由 Arduino VCC 供電限流電阻 30Ω 之正規化色彩偵測值的折線圖

4. 並聯電容的穩定性測試

電容器在電子電路中具有許多用途，其中濾波及儲能的功能是否可以讓比色計在色彩偵測上更加穩定呢？而且電容的種類繁多，那一種類的電容會比較適合呢？我們在白光 LED 的旁邊並聯了一個小電容來進行測試，接線圖如圖 12 所示。

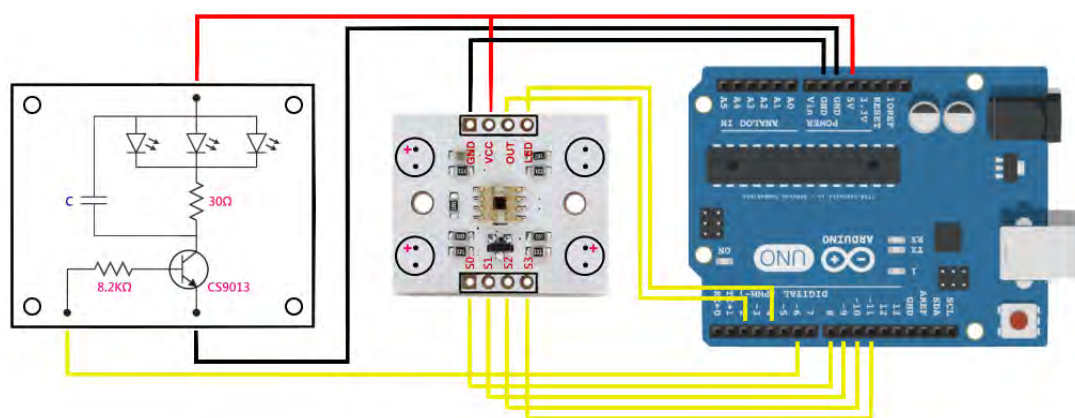


圖 12 修正後的比色計接線圖

進行實測的電容種類計有電解電容 $1\mu\text{F}$ 、陶瓷電容 $0.22\mu\text{F}$ 、MEF 電容 $0.47\mu\text{F}$ 、PPN 電容 $0.47\mu\text{F}$ 、鉭質電容 $1\mu\text{F}$ 、積層電容 $1\mu\text{F}$ ，測試所得的數據如表格 5 所

列，整體看來，鉭質電容確實讓比色計在色彩偵測上更穩定些，三個回合之平均絕對誤差的平均值百分比為 0.0093%，三回合偵測平均值間標準差的百分比為 0.0185%，其三個回合測試之色彩偵測值的折線圖如圖 13 所示。

電容 C 項目	無電容	電解電容 1 μ F	陶瓷電容 0.22 μ F	MEF 電容 0.47 μ F	PPN 電容 0.47 μ F	鉭質電容 1 μ F	積層電容 1 μ F
電容外觀							
三個回合平均絕對誤差的平均值	0.0108% (2)	0.0112% (3)	0.0148% (6)	0.0191% (7)	0.0134% (4)	0.0093% (1)	0.0147% (5)
三回合偵測平均值的平均值	R: 5134.8 G: 5000.5 B: 16408.4 I: 6615.0	R: 5134.2 G: 4999.8 B: 16407.2 I: 6613.4	R: 5138.6 G: 5003.9 B: 16420.3 I: 6618.0	R: 5122.1 G: 4988.0 B: 16366.3 I: 6596.5	R: 5127.8 G: 4993.5 B: 16386.7 I: 6604.9	R: 5129.7 G: 4995.6 B: 16392.0 I: 6608.5	R: 5128.6 G: 4994.5 B: 16387.4 I: 6605.6
三回合偵測平均值間標準差	0.0194% (3)	0.0242% (7)	0.0239% (6)	0.0086% (1)	0.0206% (4)	0.0185% (2)	0.0216% (5)

表格 5 比色計的穩定性測試數據（並聯不同的電容）

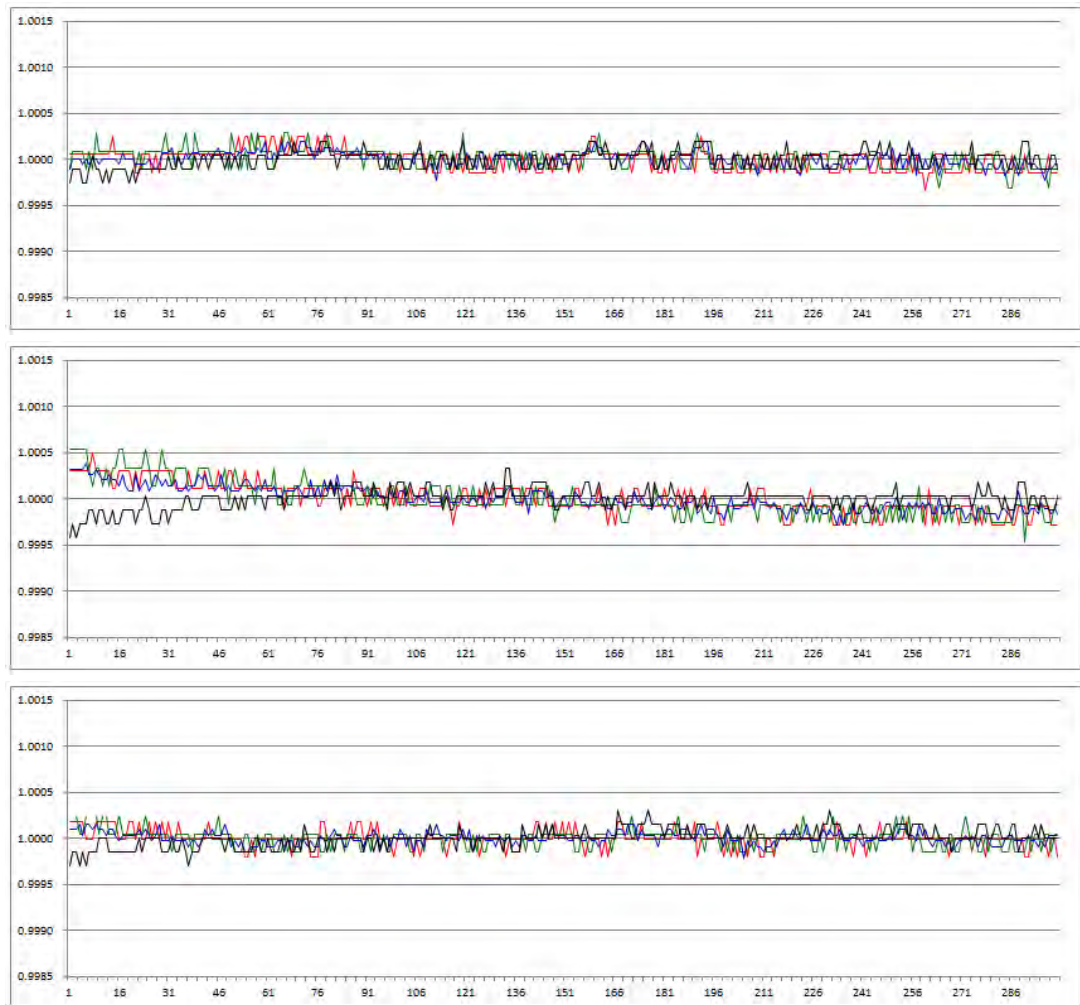


圖 13 並聯鉭質電容之正規化色彩偵測值的折線圖

四、溶液的色值偵測程式

經由多次的穩定性測試後，我們獲得了許多有關單晶片微控制器 Arduino 及色彩

感測器 TCS3200 之硬體特性，可做為溶液色彩值偵測程式的設計參考，主要的考量有下列五點，以 VB.NET 實作的溶液色彩值偵測程式，執行畫面如圖 14 所示。

- 熱機

由色彩偵測值的折線圖可以看出，Arduino、TCS3200 及 LED 光源需要一段熱機的時間，其偵測值就會漸趨穩定。

- 採用多次偵測的平均值取代一次偵測

由色彩偵測值的折線圖可以看出，偵測值會在平均線的上下一直跳動，雖然幅度不是很大，但為了避免取到極端值，我們可以進行多次偵測，然後使用平均值當做本次色彩偵測的結果。

- 縮短每次色彩偵測的時間間隔

若是採用多次偵測的平均值取代一次偵測，會使得偵測的花費時間增加，我們可以縮短 Arduino 端各色頻的偵測時間，讓使用者不至於等的不耐煩。

- 空白測試

由穩定性測試得知，每次開機後進行色彩偵測，就算偵測同樣濃度的液體時，其數據多少會不太一樣，我們可以建議使用者每次開機後都進行一次空白測試（白平衡），所得的數據就當成本次測試過程中，色彩感測器能夠偵測到各色頻的最大值（光源白點）。

- 偵測數據正規化

我們可將偵測所得的色彩值除以空白測試的色彩值，也就是進行正規化處理，這樣比較容易看出偵測值與光源白點的比例關係，而且也不受色彩感測器能夠偵測到各色頻的最大值大小不一致的影響。



圖 14-1 比色計熱機中



圖 14-2 空白測試進行中



圖 14-3 有色溶液色彩值偵測

五、濃度估測函數的尋找

參考資料[6][7][8][9]均是利用光線穿透待測溶液後，因透光強度的不同而使得接收端光敏電阻之電阻值而有所不同，再依實驗所得之檢量線來估測溶液濃度。僅使用電阻值的大小來判定溶液的濃度，雖然簡單便利，但是卻容易受到溶液吸收光

譜的特性及光敏電阻的物理性質影響而造成不小的誤差。

我們使用參考資料[2]中的多元多項式迴歸分析法來尋找最佳的濃度估測函數，可由色彩感測器 TCS3200 接收到的四種訊號（紅色、綠色、藍色、強度）來估測出溶液的濃度。四元 n 階多項式的濃度估測函數定義如下，其中 $c_{\alpha\beta\gamma\delta}$ 為多項式的係數， R 、 G 、 B 、 I 為色彩感測器 TCS3200 接收到的訊號， M' 為有色離子溶液的估測濃度， n 為多項式的階數。

$$M' = f(R, G, B, I) = \sum_{\alpha=0}^n \sum_{\beta=0}^n \sum_{\gamma=0}^n \sum_{\delta=0}^n c_{\alpha\beta\gamma\delta} R^{\alpha} G^{\beta} B^{\gamma} I^{\delta}, \quad 0 \leq \alpha + \beta + \gamma + \delta \leq n$$

我們可以用矩陣的型態來表示四維 n 階多項式估測函數，然後再利用多元多項式迴歸分析法來求得最佳的估測函數，以四元 2 階的多項式估測函數為例，式子如下：

$$M' = c_0 + c_1 R + c_2 G + c_3 B + c_4 I + c_5 RG + c_6 RB + c_7 RI + c_8 GB + c_9 GI + c_{10} BI + c_{11} R^2 + c_{12} G^2 + c_{13} B^2 + c_{14} I^2$$

以矩陣的表示法為

$$M' = [1 \quad R \quad G \quad B \quad I \quad RG \quad RB \quad RI \quad GB \quad GI \quad BI \quad R^2 \quad G^2 \quad B^2 \quad I^2] \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \\ c_{10} \\ c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \end{bmatrix}$$

利用多元多項式迴歸分析法求得最佳的估測函數

$$M = \begin{bmatrix} M'_1 \\ M'_2 \\ \vdots \\ M'_s \end{bmatrix}_{s \times 1}$$

$$P = \begin{bmatrix} 1 & R_1 & G_1 & B_1 & I_1 & R_1 G_1 & R_1 B_1 & R_1 I_1 & G_1 B_1 & G_1 I_1 & B_1 I_1 & R_1^2 & G_1^2 & B_1^2 & I_1^2 \\ 1 & R_2 & G_2 & B_2 & I_2 & R_2 G_2 & R_2 B_2 & R_2 I_2 & G_2 B_2 & G_2 I_2 & B_2 I_2 & R_2^2 & G_2^2 & B_2^2 & I_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & R_s & G_s & B_s & I_s & R_s G_s & R_s B_s & R_s I_s & G_s B_s & G_s I_s & B_s I_s & R_s^2 & G_s^2 & B_s^2 & I_s^2 \end{bmatrix}_{s \times 15}$$

$$C = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{14} \end{bmatrix}_{15 \times 1}$$

$$M \approx PC$$

則表示最佳估測函數的係數矩陣為

$$C = (P^T P)^{-1} P^T M$$

以重複組合的觀點來看，四維 n 階的多項式估測函數的總項數計有 $H_n^4 + H_{n-1}^4 + \cdots + H_1^4 + H_0^4 = C_3^{n+3} + C_3^{n+2} + \cdots + C_3^4 + C_3^3$ 項。 $n=1$ 時，有 $C_3^4 + C_3^3 = 5$ 項； $n=2$ 時，有 $C_3^5 + C_3^4 + C_3^3 = 15$ 項； $n=3$ 時，則有 $C_3^6 + C_3^5 + C_3^4 + C_3^3 = 35$ 項。

六、CuSO₄溶液的濃度偵測實驗

1. 實驗過程與步驟

(1) 實驗步驟

- 配製 18 種不同濃度的硫酸銅溶液，各 100ml。
- 將自製比色計開機並進行空白測試。
- 將硫酸銅溶液注入比色管，約九分滿。
- 將比色管放入自製比色計中，並蓋上遮光罩。
- 以自行撰寫的程式記錄下偵測所得的溶液正規化色彩值。
- 將硫酸銅由比色管中倒出，清洗比色管並擦拭乾淨。
- 更換另一種濃度的硫酸銅溶液進行偵測，重複上述步驟，直到 18 種不同濃度的硫酸銅溶液均完成測量。
- 利用自行撰寫的程式功能，將實驗結果存檔。

(2) 實驗記錄

濃度(M)	R	G	B	I	濃度(M)	R	G	B	I
0.5000	0.321905	0.773024	0.689247	0.862367	0.1250	0.608254	0.874918	0.822839	0.899926
0.4500	0.323492	0.780536	0.699194	0.873487	0.1000	0.665714	0.900065	0.854173	0.921176
0.4000	0.357143	0.799804	0.718094	0.881394	0.0875	0.704444	0.920314	0.879439	0.939708
0.3500	0.395238	0.826911	0.746245	0.900667	0.0750	0.744127	0.939582	0.905103	0.958982
0.3000	0.451111	0.866754	0.790908	0.935014	0.0625	0.756190	0.926845	0.895056	0.940944
0.2500	0.495873	0.886022	0.811400	0.938967	0.0500	0.790794	0.934357	0.908187	0.946627
0.2000	0.555873	0.909210	0.841639	0.952805	0.0313	0.847937	0.946440	0.929275	0.954781
0.1750	0.551429	0.870999	0.808614	0.908821	0.0125	0.936508	0.983997	0.976425	0.990363
0.1500	0.572698	0.870346	0.813389	0.903385	0.0050	0.923810	0.942521	0.939023	0.943909

表格 6 各種濃度硫酸銅溶液的色值偵測值

2. CuSO₄ 溶液的濃度估測函數

利用自行撰寫的程式（圖 15）載入色彩偵測之實驗結果，可以計算出一階及二階的估測函數如下。各階估測函數計算出來的估測濃度與真實濃度列於表格 7-1，平均絕對誤差及相關性如表格 7-2 所示，可以看出二階的估測函數具有較佳的估測能力。

$$f_1(R, G, B, I) = -6.105337 \times 10^{-4} + 5.657205 \times R + 1.231387 \times 10^1 \times G - 2.658432 \times 10^1 \times B + 8.656363 \times I$$

$$f_2(R, G, B, I) = -3.837654 \times 10^{-9} + 1.195951 \times 10^1 \times R + 3.533276 \times 10^1 \times G - 4.863890 \times 10^1 \times B + 2.258947 \times I - 5.985871 \times 10^2 \times RG + 6.440552 \times 10^2 \times RB + 6.246528 \times 10^1 \times RI + 2.469060 \times 10^3 \times GB - 5.639375 \times 10^2 \times GI + 2.952478 \times 10^1 \times BI - 6.152268 \times 10^1 \times R^2 - 6.834009 \times 10^2 \times G^2 - 1.537499 \times 10^3 \times B^2 + 2.388791 \times 10^2 \times I^2$$

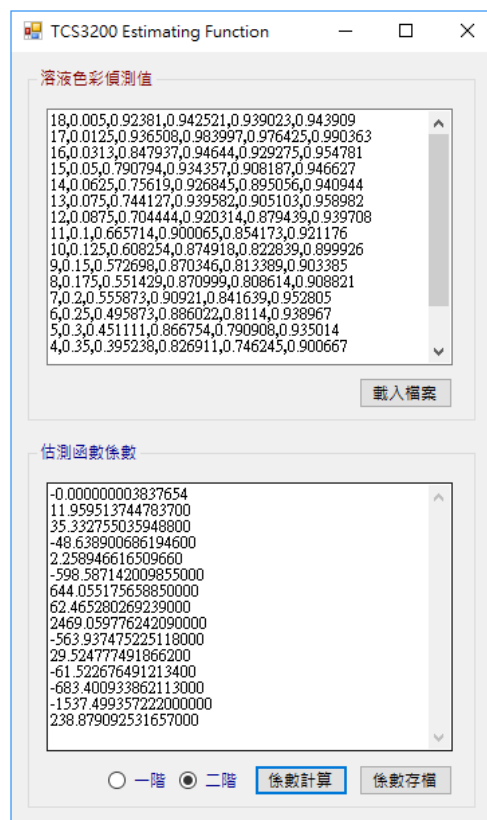


圖 15 濃度估測函數的係數計算程式

真實濃度 (M)	0.5000	0.4500	0.4000	0.3500	0.3000	0.2500	0.2000	0.1750	0.1500
一階 $f_1(R, G, B, I)$ 估測濃度	0.4812	0.4145	0.4081	0.3759	0.2926	0.2725	0.2134	0.2149	0.1532
二階 $f_2(R, G, B, I)$ 估測濃度	0.4995	0.4501	0.3992	0.3531	0.2991	0.2467	0.2029	0.1743	0.1496
真實濃度 (M)	0.1250	0.1000	0.0875	0.0750	0.0625	0.0500	0.0313	0.0125	0.0050
一階 $f_1(R, G, B, I)$ 估測濃度	0.1295	0.1152	0.0724	0.0187	0.0411	0.0294	0.0115	0.0296	0.0392
二階 $f_2(R, G, B, I)$ 估測濃度	0.1258	0.1003	0.0874	0.0753	0.0607	0.0499	0.0320	0.0122	0.0051

表格 7-1 各階函數的濃度估測值

n 階多項式函數	誤差與相關性	平均絕對誤差	相關係數平方
$n = 1$		0.021050	0.97354
$n = 2$		0.000956	0.99991

表格 7-2 四維 n 階估測函數的估測能力

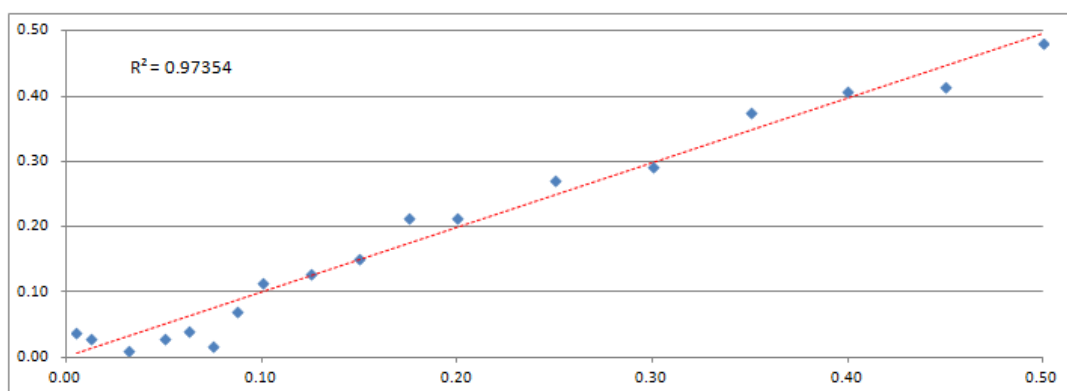


圖 16-1 一階估測函數的估測濃度與真實濃度的相關性（相關係數的平方 0.97354）

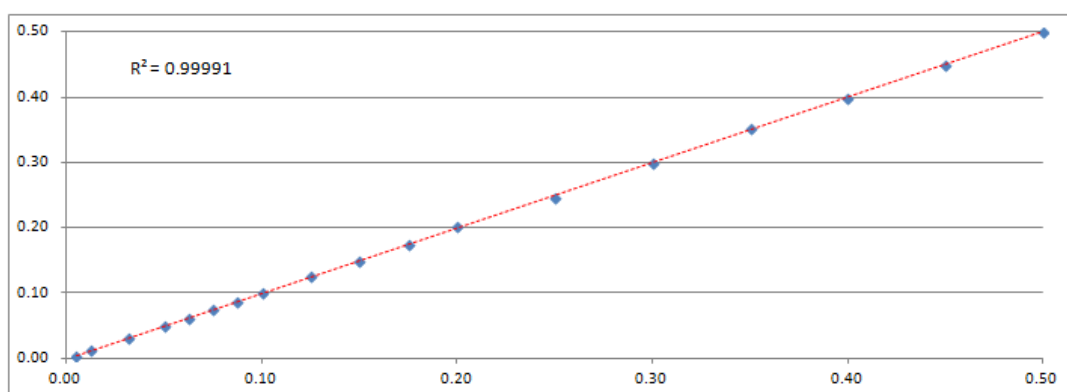


圖 16-2 二階估測函數的估測濃度與真實濃度的相關性（相關係數的平方 0.99991）

3. 濃度估測函數的精確度

另外配製了 5 種不同濃度的硫酸銅溶液，利用尋找到的二階估測函數來進行測試，實際測試結果如表格 8 所示，平均誤差為 2.75%，表示估測函數對於其他濃度的硫酸銅溶液也可以準確的估測。

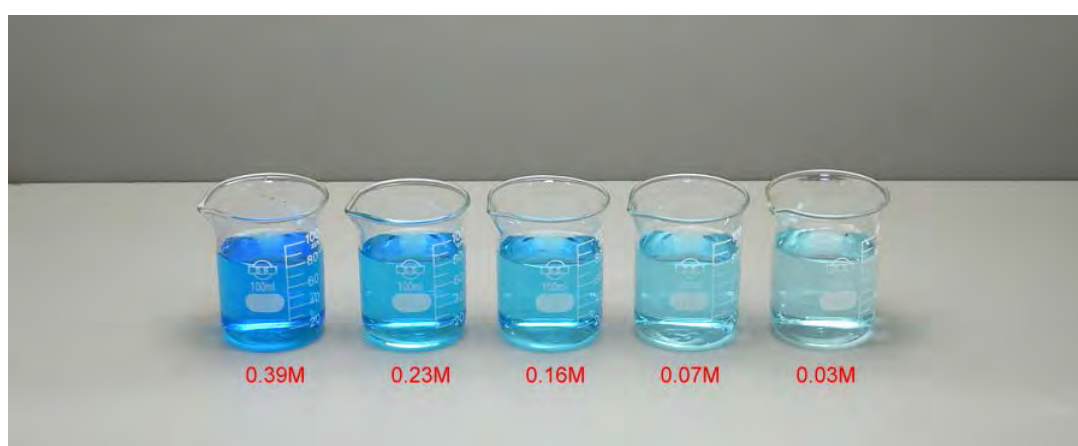


圖 17 精確度測試用的 5 種不同濃度硫酸銅溶液

真實濃度(M)	0.3900	0.2300	0.1600	0.0700	0.0300
估測濃度(M)	0.3966	0.2257	0.1556	0.0669	0.0309
誤差(%)	1.70%	1.89%	2.72%	4.47%	2.94%

表格 8 估測函數的估測值與真實濃度間的誤差（平均誤差為 2.75%）

七、FeSCN²⁺ 離子溶液的濃度偵測實驗

1. 實驗過程與步驟

(1) 實驗步驟

- 配製 18 種不同濃度的硫氰鐵離子溶液，各 100ml。

FeSCN ²⁺ (M)	配 製 方 法
0.00070000	0.00140000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00062500	0.00125000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00050000	0.00100000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00044000	0.00088000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00035000	0.00070000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00031250	0.00062500M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00025000	0.00050000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00022000	0.00044000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00017500	0.00035000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00015625	0.00031250M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00012500	0.00025000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00011000	0.00022000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00008750	0.00017500M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00007813	0.00015626M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00007000	0.00014000M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00004375	0.00008750M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00003906	0.00007812M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合
0.00002750	0.00005500M Fe(NO ₃) ₃ 50ml 與 0.2M KSCN 50ml 混合

- 每杯各滴入一滴 HNO₃，讓溶液維持酸性，防止變色。
- 其餘同硫酸銅的實驗。
- 利用自行撰寫的程式功能，將實驗結果存檔。

(2) 實驗記錄

濃度(M)	R	G	B	I	濃度(M)	R	G	B	I
0.00070000	0.387861	0.057981	0.133935	0.057330	0.00015625	0.717624	0.308564	0.378282	0.212472
0.00062500	0.375527	0.054982	0.128156	0.054564	0.00012500	0.785381	0.362140	0.436554	0.263882
0.00050000	0.443591	0.074417	0.159791	0.068241	0.00011000	0.812750	0.431070	0.491384	0.325376
0.00044000	0.481308	0.085734	0.176084	0.072133	0.00008750	0.842259	0.545818	0.585015	0.443299
0.00035000	0.549497	0.124958	0.216668	0.087503	0.00007813	0.843233	0.582806	0.613809	0.483782
0.00031250	0.567997	0.136954	0.229038	0.093538	0.00007000	0.874020	0.579572	0.620085	0.479148
0.00025000	0.614486	0.160494	0.255875	0.108459	0.00004375	0.933788	0.792403	0.803609	0.723912
0.00022000	0.649866	0.198217	0.288251	0.130514	0.00003906	0.918208	0.781406	0.792152	0.716118
0.00017500	0.694255	0.274242	0.349285	0.184310	0.00002750	0.954940	0.843621	0.852846	0.792164

表格 9 各種濃度硫氰鐵離子溶液的色相偵測值

2. FeSCN²⁺ 離子溶液的濃度估測函數

利用自行撰寫的程式（圖 18）載入色彩偵測之實驗結果，可以計算出一階及二階的估測函數如下。各階估測函數計算出來的估測濃度與真實濃度列於表格 10-1，平均絕對誤差及相關性如表格 10-2 所示，可以看出二階的估測函數具有較佳的估測能力。

$$f_1(R, G, B, I) = 9.066677 \times 10^{-8} + 1.798648 \times 10^{-2} \times R + 2.811345 \times 10^{-2} \times G - 6.971823 \times 10^{-2} \times B + 2.346425 \times 10^{-2} \times I$$

$$f_2(R, G, B, I) = 3.389000 \times 10^{-12} + 4.168884 \times 10^{-1} \times R + 7.569237 \times 10^{-1} \times G - 1.687888 \times B + 5.058949 \times 10^{-1} \times I - 1.904719 \times 10^1 \times RG + 4.490276 \times 10^1 \times RB - 1.421238 \times 10^1 \times RI + 7.111005 \times 10^1 \times GB - 2.282976 \times 10^1 \times GI + 5.364932 \times 10^1 \times BI - 5.945116 \times R^2 - 1.486529 \times 10^1 \times G^2 - 8.430116 \times 10^1 \times B^2 - 8.453928 \times I^2$$



圖 18 濃度估測函數的係數計算程式

真實濃度 (M)	0.00070000	0.00062500	0.00050000	0.00044000	0.00035000	0.00031250
一階 $f_1(R, G, B, I)$ 估測濃度	0.00061041	0.00064145	0.00053040	0.00048345	0.00034585	0.00029583
二階 $f_2(R, G, B, I)$ 估測濃度	0.00067616	0.00063982	0.00052339	0.00042176	0.00035319	0.00030999
真實濃度 (M)	0.00025000	0.00022000	0.00017500	0.00015625	0.00012500	0.00011000
一階 $f_1(R, G, B, I)$ 估測濃度	0.00027343	0.00023062	0.00017287	0.00019641	0.00006789	0.00011635
二階 $f_2(R, G, B, I)$ 估測濃度	0.00025808	0.00021471	0.00018432	0.00014717	0.00011682	0.00012287
真實濃度 (M)	0.00008750	0.00007813	0.00007000	0.00004375	0.00003906	0.00002750
一階 $f_1(R, G, B, I)$ 估測濃度	0.00011123	0.00011065	0.00002602	0.00003553	0.00006156	0.00002547
二階 $f_2(R, G, B, I)$ 估測濃度	0.00007594	0.00008383	0.00007262	0.00004268	0.00003982	0.00002758

表格 10-1 各階函數的濃度估測值

誤差與相關性 n 階多項式函數	平均絕對誤差	相關係數平方
$n = 1$	0.0000265634	0.97138
$n = 2$	0.000008577	0.99666

表格 10-2 四維 n 階估測函數的估測能力

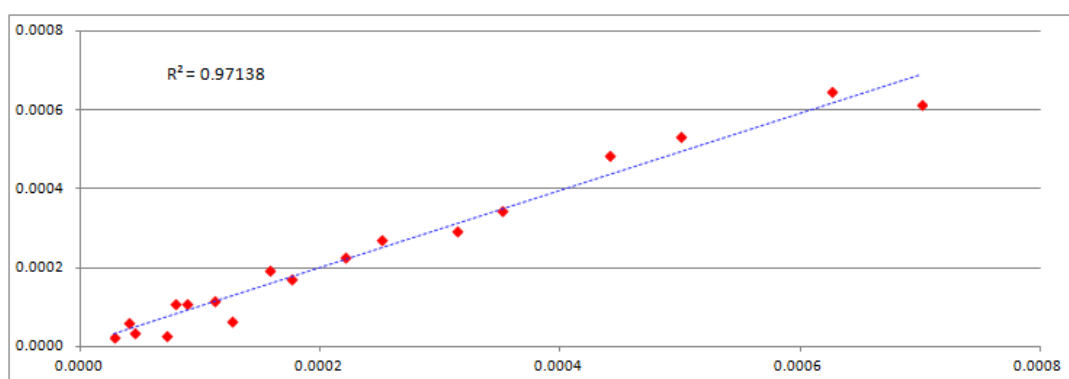


圖 19-1 一階估測函數的估測濃度與真實濃度的相關性（相關係數的平方 0.97138）

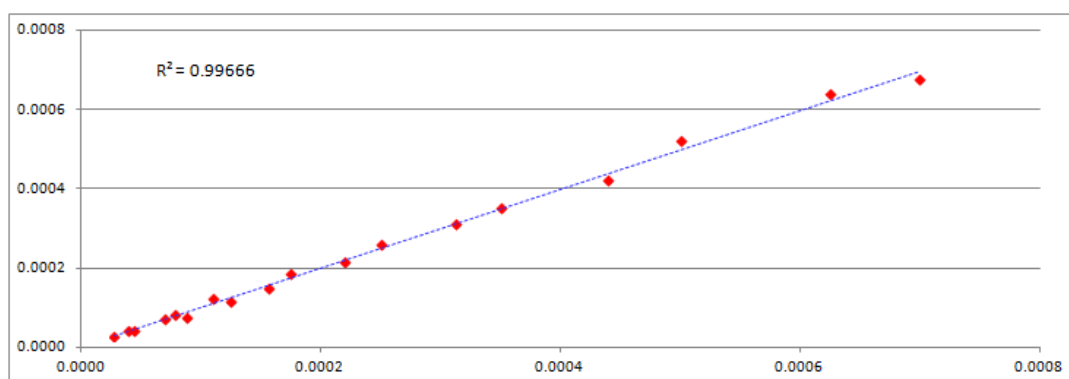


圖 19-2 二階估測函數的估測濃度與真實濃度的相關性（相關係數的平方 0.99666）

3. 濃度估測函數的精確度

另外配製了 5 種不同濃度的硫氰鐵離子溶液，利用尋找到的二階估測函數來進行測試，實際測試結果如表格 11 所示，平均誤差為 3.77%，表示估測函數對於其他濃度的硫氰鐵離子溶液也可以準確的估測。



圖 20 精確度測試用的 5 種不同濃度硫氰鐵離子溶液

真實濃度(M)	0.00052000	0.00030000	0.00017000	0.00006300	0.00004000
估測濃度(M)	0.00053982	0.00029873	0.00018092	0.00005894	0.00004070
誤差(%)	3.81%	0.42%	6.42%	6.45%	1.76%

表格 11 估測函數的估測值與真實濃度間的誤差（平均誤差為 3.77%）

伍、結果與討論：

一、溶液濃度偵測程式實作

我們以 VB.Net 來實作溶液的濃度偵測程式，程式的執行畫面如圖 21 所示，經過熱機、空白測試及載入濃度估測函數的係數後，即可開始進行溶液濃度偵測，每次的偵測結果顯示在 TextBox（文字框）中。



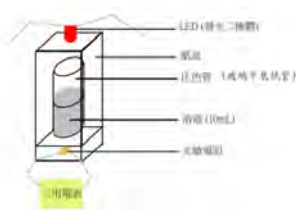
圖 21 溶液濃度偵測程式

二、自製比色計之優點探討

1. 方便性



參考資料[6]裝置



參考資料[7]裝置



參考資料[8]裝置



參考資料[9]裝置



我們的裝置

我們使用平價的單晶片微控制器 **Arduino** 及色彩感測器 **TCS3200** 所自製的比色計，透過 **USB** 介面傳送過來的色彩偵測值，即可透過程式即時的估測溶液之濃度，使用起來非常便利，與參考資料[6][7][8][9]的裝置比較起來，顯得乾淨俐落，不需要準備額外的電源與器材，及記錄數據後的計算工作。

2. 精確度

參考資料[6][7][8][9]均是利用光敏電阻所接收的亮度訊息來估測溶液的濃度，僅依據單一的訊息資料來估測溶液的濃度，雖不失為一種簡單俐落的方法，但是卻容易失之偏頗而造成不小的誤差。我們的估測函數以紅色、綠色、藍色、強度等 4 個變數所組成，除了適用於各種顏色的溶液外，與使用光敏電阻的參考資料[6][7][8][9]比較起來，誤差較小，準確度較高。

三、實驗結果討論

在以檢量線推得未知濃度之後，硫酸銅溶液和硫氰鐵溶液尚有 2.75% 以及 3.77% 的誤差。我們認為是因為電子秤僅能測量至小數點後三位，而且在濃度極低的狀況下，誤差也會在稀釋後放大。且硫氰鐵在 20℃ 的狀態下，會較容易解離回硫氰酸鉀和硝酸鐵，因此我們推測這也是造成實驗誤差的原因之一。

陸、結論：

- 一、從實驗結果來看，以 **Arduino** 及 **TCS3200** 確實可以測量到準確的溶液濃度，進而達到減少溶液浪費之目的。且所有實驗器材均容易取得、價格便宜，裝置簡單易於操作，使結果和實驗目的相符。
- 二、本實驗裝置簡單俐落，均直接透過 **USB** 介面與電腦連接，不須要額外的輔助器材，如暗箱、電源供應器、三用電表等，可以快速讀取資料，並加以分析，便利性極高。

- 三、為了使 LED 燈亮度增強，我們改變其電路並發現當電阻為 30Ω 時，可使亮度提高也依舊維持其穩定性。而為使自製比色計之色彩偵測更加精準，需要並聯電容來穩定。而相較於其他電容，鉭質電容可以更有效地穩定光源的強度。
- 四、色彩偵測程式方面，我們根據 Arduino 及 TCS3200 之特性，在每次偵測前給予熱機時間，並以空白測試測到的值使每次測量的數據正規化。這樣不僅容易看出其比例關係，也讓裝置較不受白平衡不同影響，且以多次偵測之平均值取代單次偵測，避免取得極端值，能使測量結果更準確。
- 五、使用紅色、綠色、藍色、強度等四種偵測值，搭配多元多項式迴歸分析法所尋找的濃度估測函數，可以提高未知溶液的濃度偵測精確度。
- 六、若先以標準液求得估測函數，此裝置亦可推廣至其他有色溶液濃度的估測。

柒、參考資料：

- 一、普通高中基礎化學及選修化學課本。
- 二、余桂霖著，“多元迴歸分析，” 五南文化, 2012。
- 三、楊明豐著，“ARDUINO 最佳入門與應用，” 基峰資訊, 2014。
- 四、陳會安著，“Microsoft Visual Basic 2015：程式設計範例教本，” 旗標科技, 2015。
- 五、<https://www.parallax.com/sites/default/files/downloads/28302-TCS3200-doc.pdf> 色彩感測器 TCS3200 產品規格說明書, PARALLAX 網站。
- 六、李銘哲、陳意曉、陳佑銘、李景煊著，“大家一起來比色—比色法實驗器材的創新及改良，” 中華民國第 43 屆中小學科學展覽會佳作作品, 2003。
- 七、林育任著，“自製多功能光度計之研究，” 第 4 屆旺宏科學獎優等獎作品, 2005。
- 八、沈冠宇、彭裕文、楊富雄、高維鴻著，“自製濃度觀測工具及其在高中化學實驗的應用，” 中華民國第 46 屆中小學科學展覽會佳作作品, 2006。
- 九、邱秀玲、李妍禎著，“設計、色計—自行設計微型比色計改良比色法實驗，” 中華民國第 46 屆中小學科學展覽會第三名作品, 2006。

捌、附錄：

一、穩定性測試 Arduino 端的程式設計

```
#include <TimerOne.h>

#define S0 8
#define S1 9
#define S2 10
#define S3 11
#define OUT 2 //interrupt number int.0 on UNO board is pin 2
#define LED 4 //TCS3200 LED pin
#define TRANSISTOR 6 //Transistor switch

long _count = 0; //count the frequency
long _array[4]; //store the RGBC value
```



```

long _flag = 0; //filter of RGBC queue
char commandChar;
String tmpString;

//-----

void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(OUT, INPUT);
  pinMode(LED, OUTPUT);
  pinMode(TRANSISTOR, OUTPUT);
  digitalWrite(S0, HIGH); //output frequency scaling 20%
  digitalWrite(S1, LOW);
  digitalWrite(LED, LOW); //TCS3200 LED off
  digitalWrite(TRANSISTOR, LOW); //LED off
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0)
  {
    commandChar = Serial.read();
    delay(100);
    switch (commandChar)
    {
      case 'S': //Stability_Test_Start
        digitalWrite(LED, HIGH); //TCS3200 LED on
        digitalWrite(TRANSISTOR, HIGH); //LED on
        Timer1.attachInterrupt(TCS_Callback); //attaches callback() as a timer overflow interrupt
        attachInterrupt(0, TCS_Count, RISING); //interrupt number int.0 on UNO board is pin 2
        Timer1.initialize(500000); //set a timer of length 500000 microseconds ( 0.5sec )
        Timer1.setPeriod(500000); //set 0.5sec period
        _flag = 4; //Idle
        _count = 0;
        break;
      case 'P': //Stability_Test_Stop
        digitalWrite(LED, LOW); // TCS3200 LED off
        digitalWrite(TRANSISTOR, LOW); //LED off
        Timer1.stop();
        Timer1.detachInterrupt();
        detachInterrupt(0); //Stability Test Stop !
        break;
      default:
        tmpString = String("default : not a commandChar");
        Serial.println(tmpString);
        break;
    }
  }
}

//-----

void TCS_Count()
{
  _count ++ ;
}

void TCS_Callback()
{
  switch (_flag)
  {
    case 0: //第 1.0s, 3.5s, 6.0s, ....
      _array[0] = _count; //紀錄紅色
      _flag = 1; //next step : Filter without Green
      digitalWrite(S2, HIGH);
  }
}

```

```

        digitalWrite(S3, HIGH);
        Timer1.setPeriod(500000); //set 0.5sec period
        _count = 0;
        break;
    case 1: //第 1.5s, 4.0s, 6.5s, ....
        _array[1] = _count; //紀錄綠色
        _flag = 2; //next step : Filter without Blue
        digitalWrite(S2, LOW);
        digitalWrite(S3, HIGH);
        Timer1.setPeriod(500000); //set 0.5sec period
        _count = 0;
        break;
    case 2: //第 2.0s, 4.5s, 7.0s, ....
        _array[2] = _count; //紀錄藍色
        _flag = 3; //next step : Clear(no filter)
        digitalWrite(S2, HIGH);
        digitalWrite(S3, LOW);
        Timer1.setPeriod(500000); //set 0.5sec period
        _count = 0;
        break;
    case 3: //第 2.5s, 5.0s, 7.5s, ....
        _array[3] = _count; //紀錄光的強度
        _flag = 4; //next step : Idle
        Timer1.setPeriod(500000); //set 0.5sec period
        _count = 0;
        tmpString = String();
        tmpString = tmpString + _array[0] + ',' + _array[1] + ',' + _array[2] + ',' + _array[3];
        Serial.println(tmpString); //輸出色彩偵測值到串列埠
        break;
    case 4: //第 0.5s, 3.0s, 5.5s, ....
        _flag = 0; //next step : Filter without Red
        digitalWrite(S2, LOW);
        digitalWrite(S3, LOW);
        Timer1.setPeriod(500000); //set 0.5sec period
        _count = 0;
        break;
}
}

```

二、穩定性測試電腦端的程式設計

```

Imports System.IO
Imports System.IO.Ports

Public Class Form1

    Dim COMport As String
    Dim count, stopCount As Integer

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load '程式一開始執行時
        COMport = ""
        For Each port As String In My.Computer.Ports.SerialPortNames
            ComboBox.Items.Add(port)
        Next
        btnConnect.Enabled = False
        rad200.Enabled = False : rad300.Enabled = False : rad400.Enabled = False
        btnSwitch.Enabled = False
        btnSave.Enabled = False
        TextBox.Text = ""
    End Sub

    Private Sub ComboBox_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
        ComboBox.SelectedIndexChanged '下拉式選單中串列埠改變時
        If (ComboBox.SelectedItem <> "") Then
            COMport = ComboBox.SelectedItem
            btnConnect.Enabled = True
        End If
    End Sub

```

End Sub

Private Sub **btnConnect_Click**(sender As Object, e As EventArgs) Handles btnConnect.Click

 If (btnConnect.Text = "Connect") Then '--- 按下 Connect

 If (COMport <> "") Then

 If SerialPort.IsOpen Then

 SerialPort.Close()

 End If

 Try

 With SerialPort

 .PortName = COMport

 .BaudRate = 9600

 .DataBits = 8

 .Parity = System.IO.Ports.Parity.None

 .StopBits = System.IO.Ports.StopBits.One

 .Handshake = System.IO.Ports.Handshake.None

 .Encoding = System.Text.Encoding.Default

 End With

 SerialPort.Open()

 btnConnect.Text = "disConnect"

 rad200.Enabled = True : rad300.Enabled = True : rad400.Enabled = True

 btnSwitch.Enabled = True

 lblCount.Text = "0"

 TextBox.Text = ""

 btnSave.Enabled = False

 Catch ex As Exception

 MessageBox.Show(ex.Message)

 End Try

 Else

 MsgBox("Select a COM port first !")

 End If

 Else '--- 按下 disConnect

 btnConnect.Text = "Connect"

 btnSwitch.Text = "Start"

 btnSwitch.Enabled = False

 rad200.Enabled = False : rad300.Enabled = False : rad400.Enabled = False

 SerialPort.Write("P") 'Stability Test Stop

 SerialPort.Close()

 End If

End Sub

Private Sub **btnSwitch_Click**(sender As Object, e As EventArgs) Handles btnSwitch.Click

 If (btnSwitch.Text = "Start") Then '--- 按下 Start

 If SerialPort.IsOpen Then

 btnSwitch.Text = "Stop"

 lblCount.Text = "0"

 count = 0

 btnSave.Enabled = True

 TextBox.Text = ""

 If rad200.Checked = True Then stopCount = 200

 If rad300.Checked = True Then stopCount = 300

 If rad400.Checked = True Then stopCount = 400

 rad200.Enabled = False : rad300.Enabled = False : rad400.Enabled = False

 SerialPort.Write("S") 'Stability Test Start

 Else

 MsgBox("Open Serial port first !")

 End If

 Else '--- 按下 Stop

 btnSwitch.Text = "Start"

 rad200.Enabled = True : rad300.Enabled = True : rad400.Enabled = True

 SerialPort.Write("P") 'Stability Test Stop

 End If

End Sub

Private Sub **SerialPort_DataReceived**(sender As Object, e As SerialDataReceivedEventArgs) Handles

 SerialPort.DataReceived '當串列埠接收到資料時

 Dim Incoming As String

 Try

```

Incoming = SerialPort.ReadLine()
Incoming = Incoming.Trim()
If Incoming Is Nothing Then
    TextBox.Text &= "nothing" & vbCrLf
Else
    count = count + 1
    If count <= stopCount Then
        lblCount.Text = count & " / " & stopCount
        TextBox.Text &= Incoming & vbCrLf
    Else
        btnSwitch.Text = "Start"
        rad200.Enabled = True : rad300.Enabled = True : rad400.Enabled = True
        SerialPort.Write("P") 'Stability Test Stop
    End If
End If
Catch ex As TimeoutException
    TextBox.Text &= "Serial Port read error !" & vbCrLf
End Try
End Sub

Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click
    SaveFileDialog.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
    SaveFileDialog.Title = "儲存文字檔"
    SaveFileDialog.ShowDialog()
End Sub

Private Sub SaveFileDialog_FileOk(sender As Object, e As System.ComponentModel.CancelEventArgs)
    Handles SaveFileDialog.FileOk
    Dim FS As FileStream = SaveFileDialog.OpenFile
    If FS IsNot Nothing Then
        Dim SW As New StreamWriter(FS)
        SW.Write(TextBox.Text.Trim())
        SW.Close()
    End If
    FS.Close()
End Sub

Private Sub Form1_FormClosed(sender As Object, e As FormClosedEventArgs) Handles Me.FormClosed
    If SerialPort.IsOpen Then
        SerialPort.Close()
    End If
End Sub

End Class

```

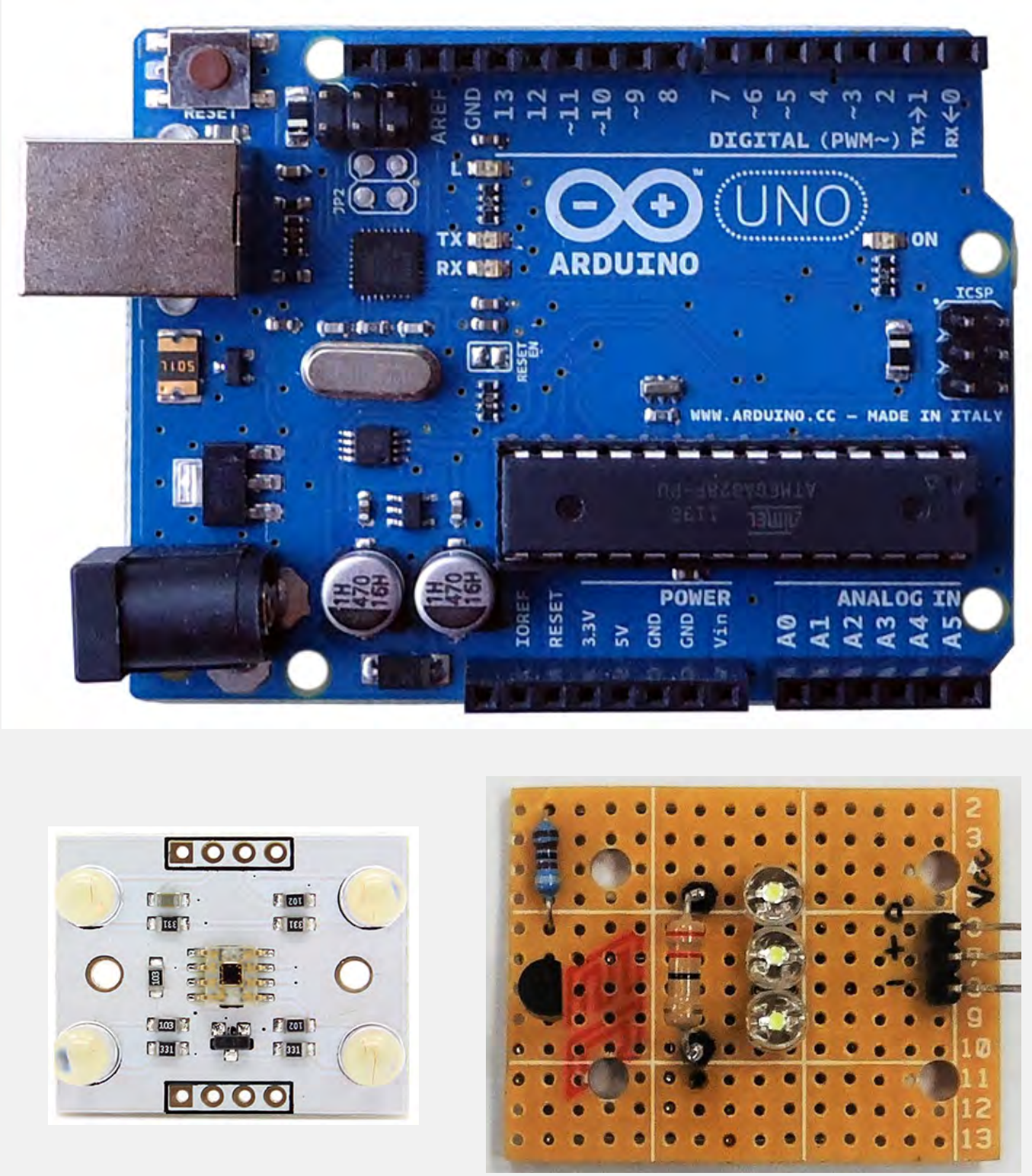
【評語】 052320

本作品為利用比色計來判斷有色液體濃度的應用，實驗系統包括單晶片微控制器 Arduino、色彩感測器 TCS3200、散光型白光 LED、電阻、電容、電晶體、筆記型電腦等元件。完成度很好，也有不錯的準確度，是不錯的實作作品。

作品海報

摘要

本研究利用單晶片微控制器Arduino、色彩感測器 TCS3200、筆記型電腦等器材製作出一簡易的比色計。先透過限流電阻及補償電容的更換交叉測試，尋找到系統所能提供最穩定光源的組合，然後利用一些已知濃度溶液的彩色值，用多元多項式迴歸分析法尋找出該溶液的最佳濃度估測函數，以後只要測得未知濃度溶液的偵測值，即可以此估測函數計算出該溶液的濃度。



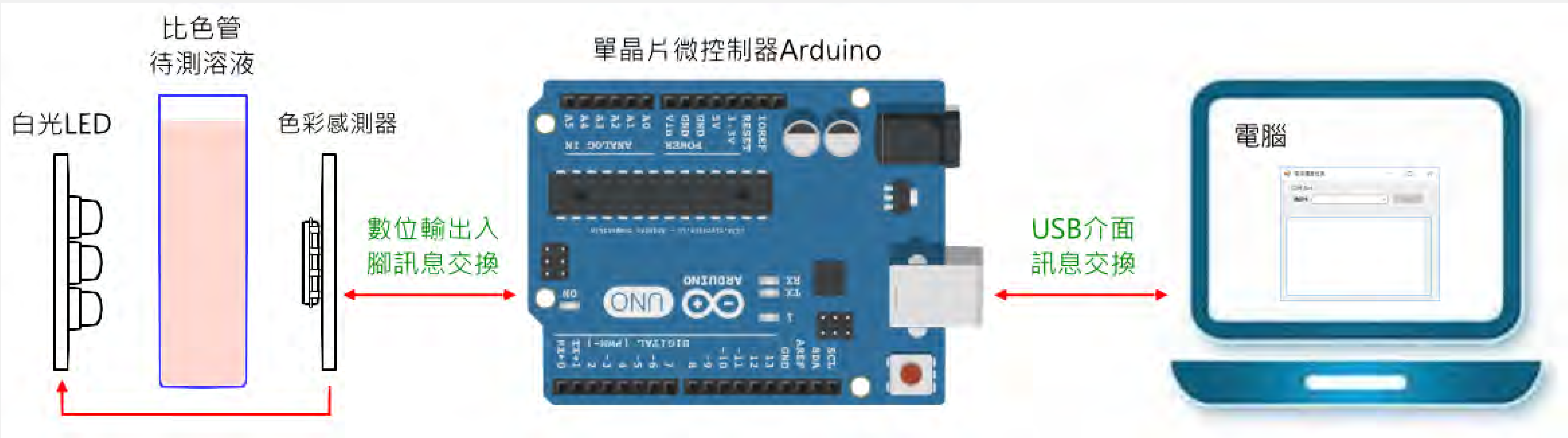
器材及軟體

Visual Basic.Net 2012、Arduino Uno板、色彩感測器TCS3200、筆記型電腦、杜邦線、電路板、白光LED、電阻、NPN電晶體、電解電容、焊錫、烙鐵、木板、電子天秤、容量瓶、玻棒、刮勺、滴管、漏斗、秤量紙、燒杯、Parafilm。
藥品：CuSO₄、Fe(NO)₃、KSCN、HNO₃、HCl、蒸餾水。

研究過程

相關研究 → 比色計的設計與實作 → 穩定性測試與光源的設計
→ 溶液的彩色值偵測程式 → CuSO₄溶液及FeSCN²⁺ 離子溶液的濃度偵測實驗

(一)比色計的構想與實作



比色計的設計構想圖

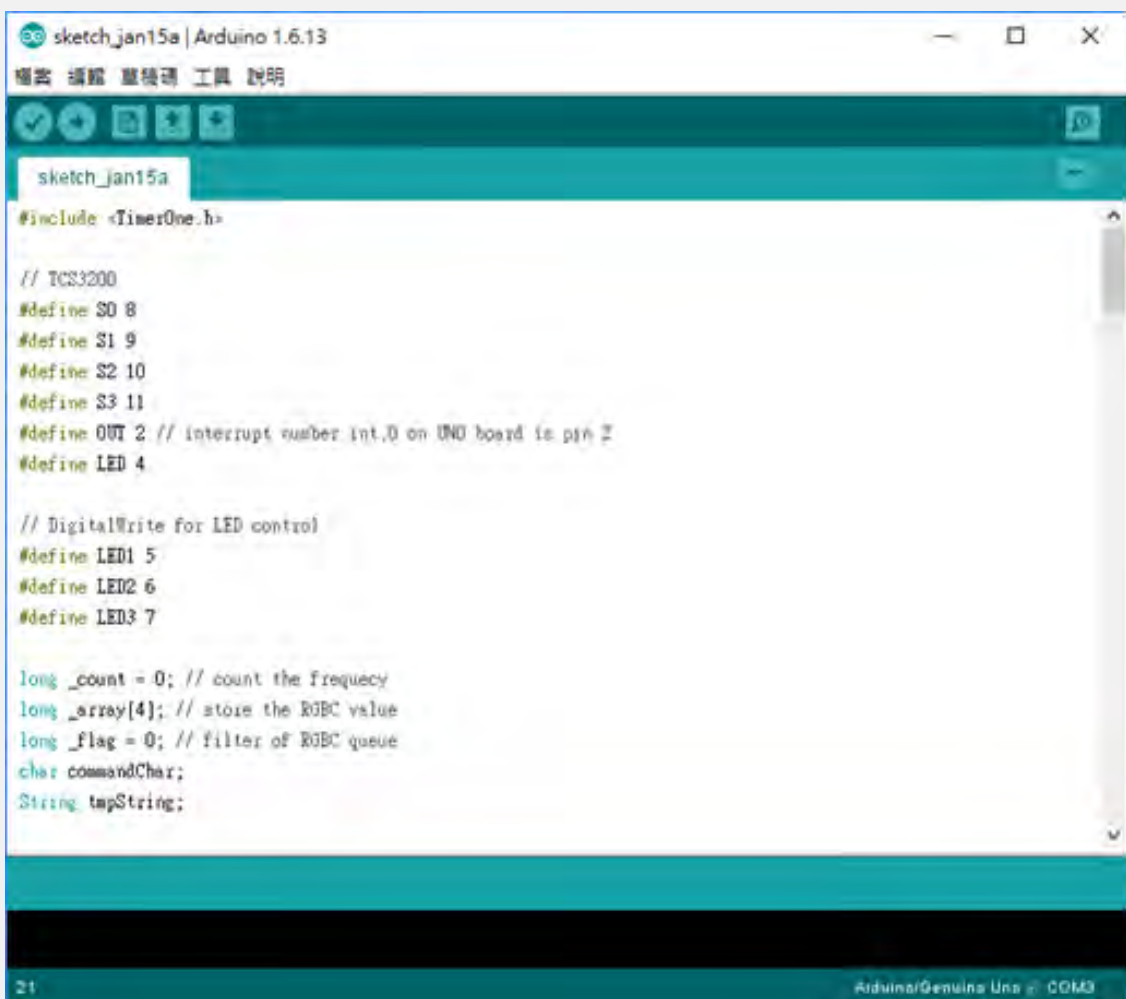


比色計連接筆電圖



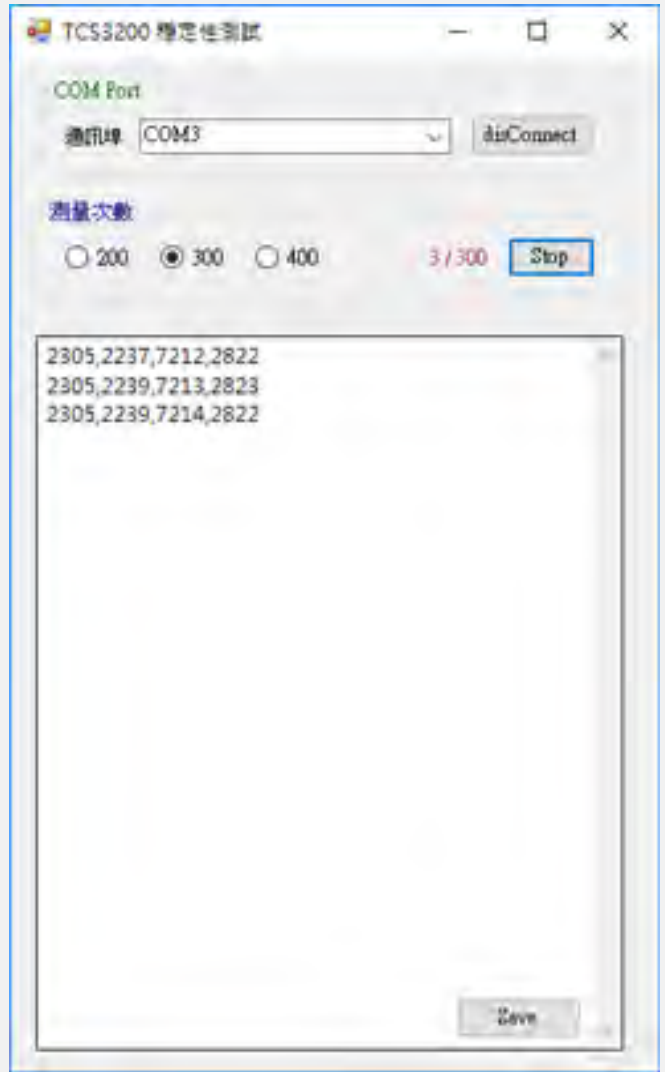
自製Arduino比色計裝置圖

(二)穩定性測試與光源的設計



使用Arduino IDE開發Uno板上的色彩偵測程式

每 2.5 秒鐘完成一次溶液穿透光線的彩色偵測，R、G、B、I 各測 0.5 秒，休息 0.5 秒鐘。



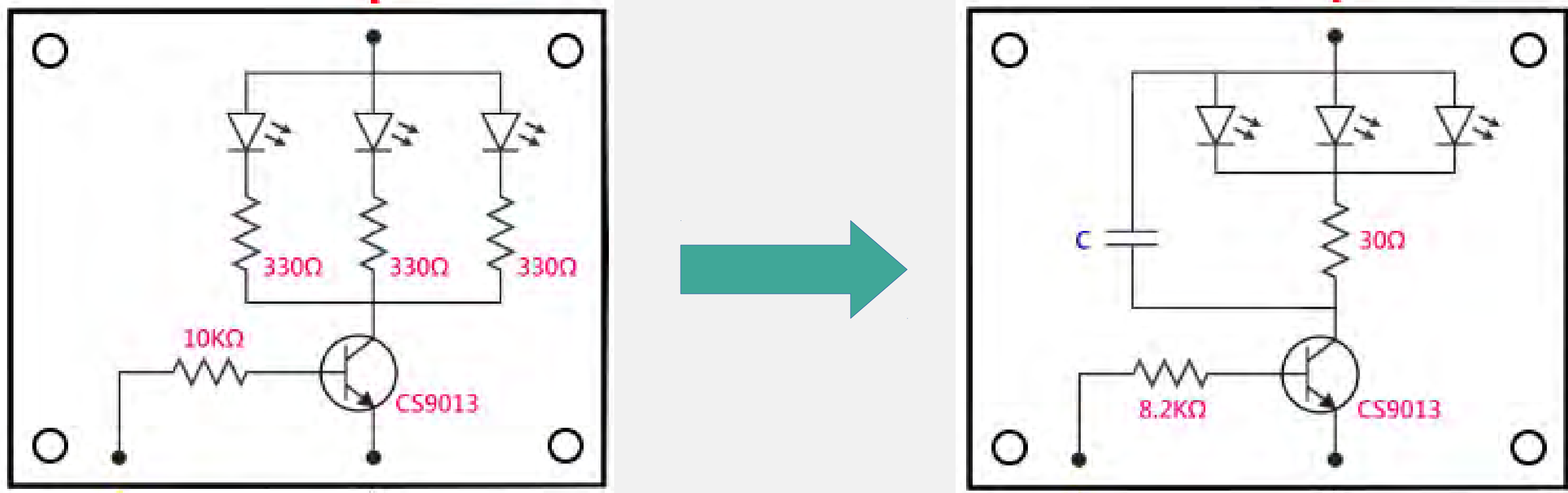
電腦與Arduino的訊息交換程式

以 VB.NET 來開發電腦端的程式。開始偵測後， TextBox 可以顯示 Arduino 回傳過來的色彩偵測數據資料。當偵測完成，按下 Save 按鈕即可將 TextBox 中的數據資料存成文字檔。

(二)穩定性測試與光源的設計

光源由色彩感測器TCS3200供電的穩定性測試

→由測試結果看來，整個色彩偵測系統算是穩定的，三個回合各 300 次的偵測值幾乎都介於平均值的 $\pm 0.1\%$ 之間。



光源由 Arduino VCC 供電的穩定性測試

→ 結果發現， 30Ω 的限流電阻具有較佳的表現，三個回合之平均絕對誤差的平均值百分比僅 0.0100% ，在所有測量的電阻值中誤差最小。

項目 \ 電阻R	110 Ω	51 Ω	39 Ω	33 Ω	30 Ω	20 Ω	TCS3200
供電電壓VCC	4.42V	4.39V	4.38V	4.37V	4.36V	4.31V	4.43V
LED順向電壓	2.86V	2.94V	2.98V	3.02V	3.03V	3.10V	2.81V 2.84V 2.79V
電阻電壓降	1.56V	1.45V	1.40V	1.35V	1.33V	1.21V	1.62V 1.59V 1.64V
供電電流I	14.18mA	28.43mA	35.90mA	40.91mA	44.33mA	60.50mA	14.70mA
三個回合平均絕對誤差的平均值	0.0230% (7)	0.0158% (4)	0.0161% (5)	0.0121% (2)	0.0100% (1)	0.0156% (3)	0.0204% (6)
三回合偵測平均值間標準差	0.0251% (5)	0.0292% (6)	0.0794% (7)	0.0224% (4)	0.0206% (2)	0.0194% (1)	0.0219% (3)

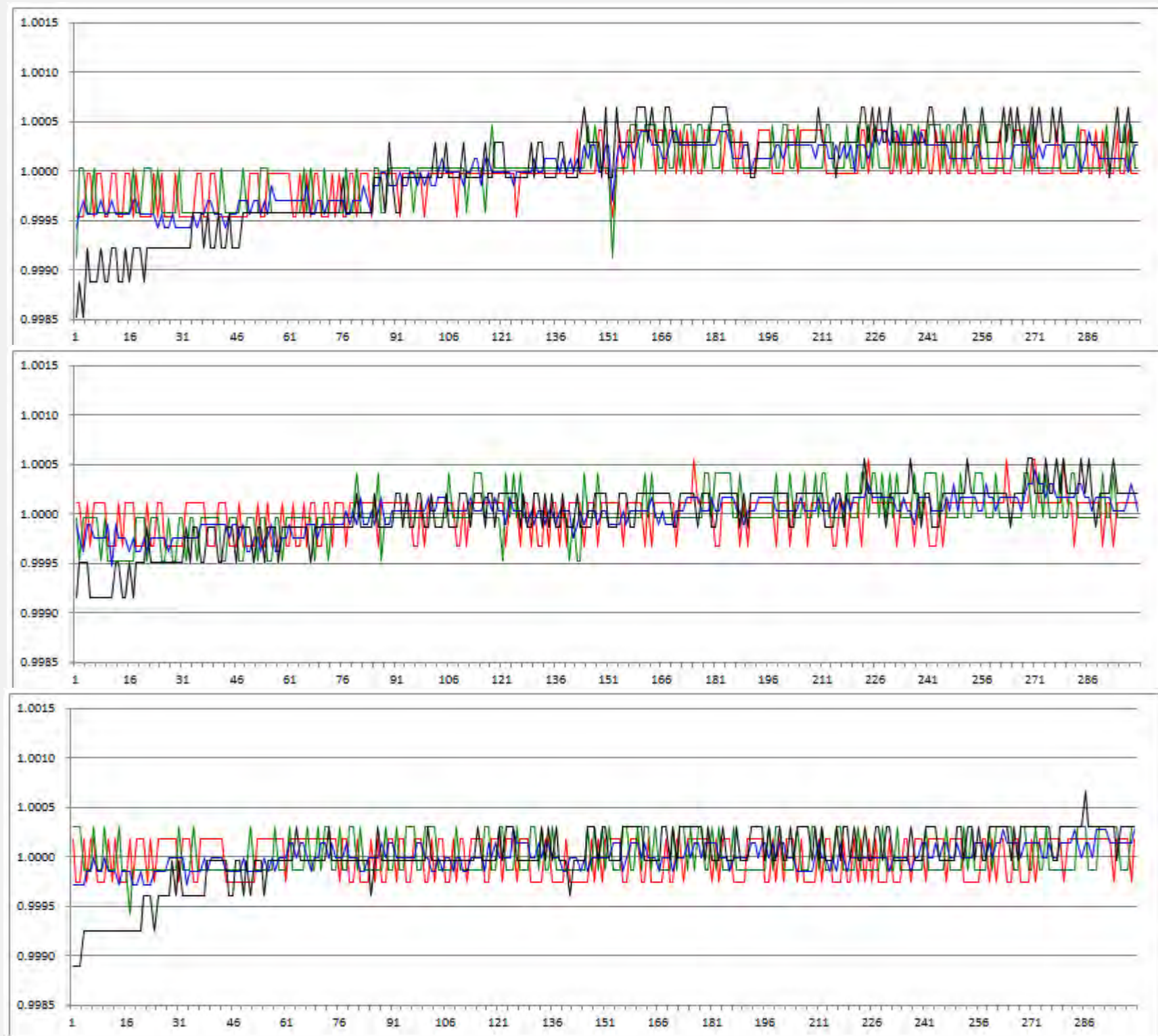
比色計在不同限流電阻下的穩定性測試數據

並聯電容的穩定性測試

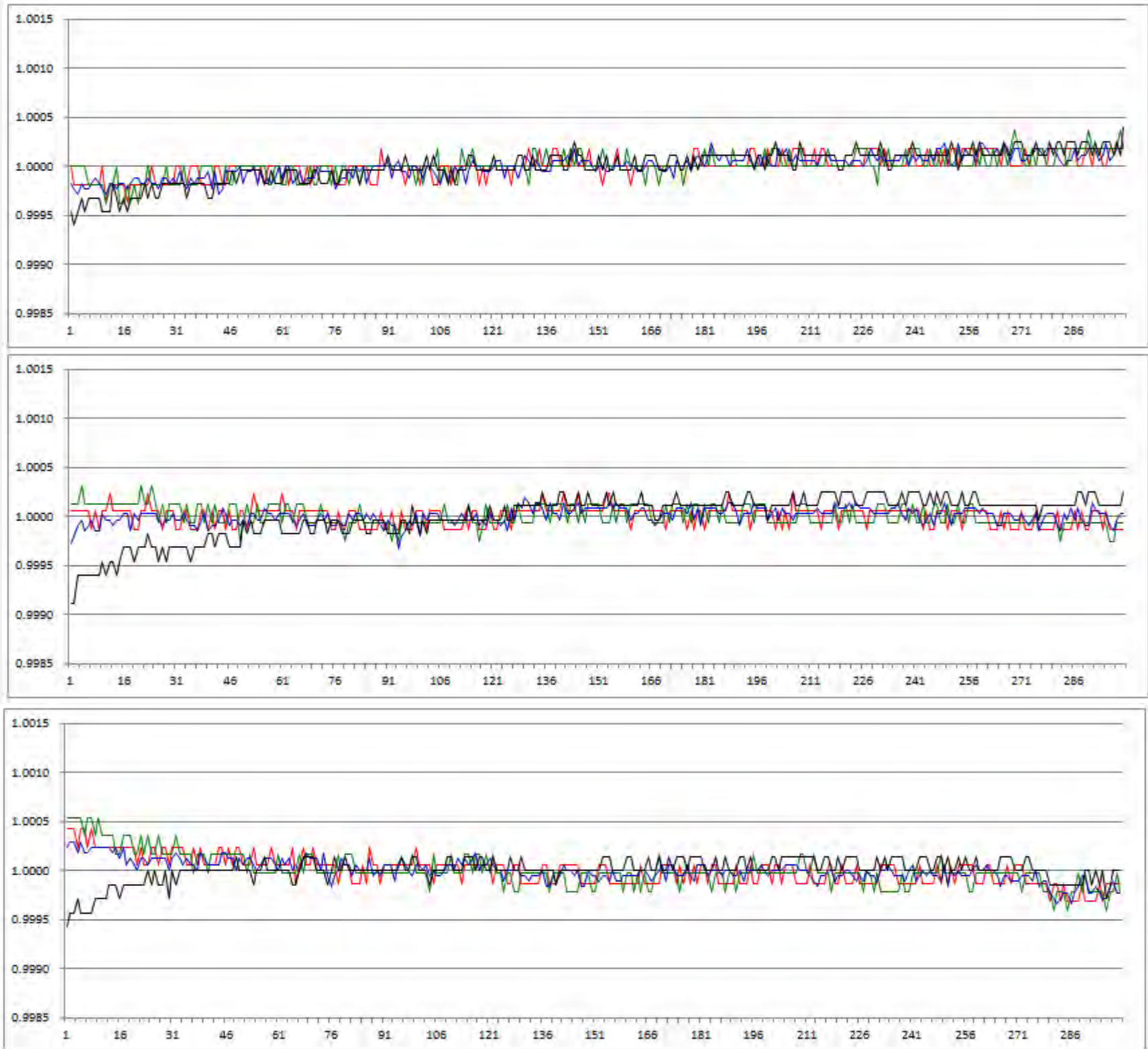
→結果發現，鉅質電容可以使比色計在色彩偵測上更為穩定，三個回合之平均絕對誤差的平均值百分比僅為 0.0093% 。

項目 \ 電容C	無電容	電解電容 1 μ F	陶瓷電容 0.22 μ F	MEF電容 0.47 μ F	PPN電容 0.47 μ F	鉅質電容 1 μ F	積層電容 1 μ F
電容外觀							
三個回合平均絕對誤差的平均值	0.0108% (2)	0.0112% (3)	0.0148% (6)	0.0191% (7)	0.0134% (4)	0.0093% (1)	0.0147% (5)
三回合偵測平均值間標準差	0.0194% (3)	0.0242% (7)	0.0239% (6)	0.0086% (1)	0.0206% (4)	0.0185% (2)	0.0216% (5)

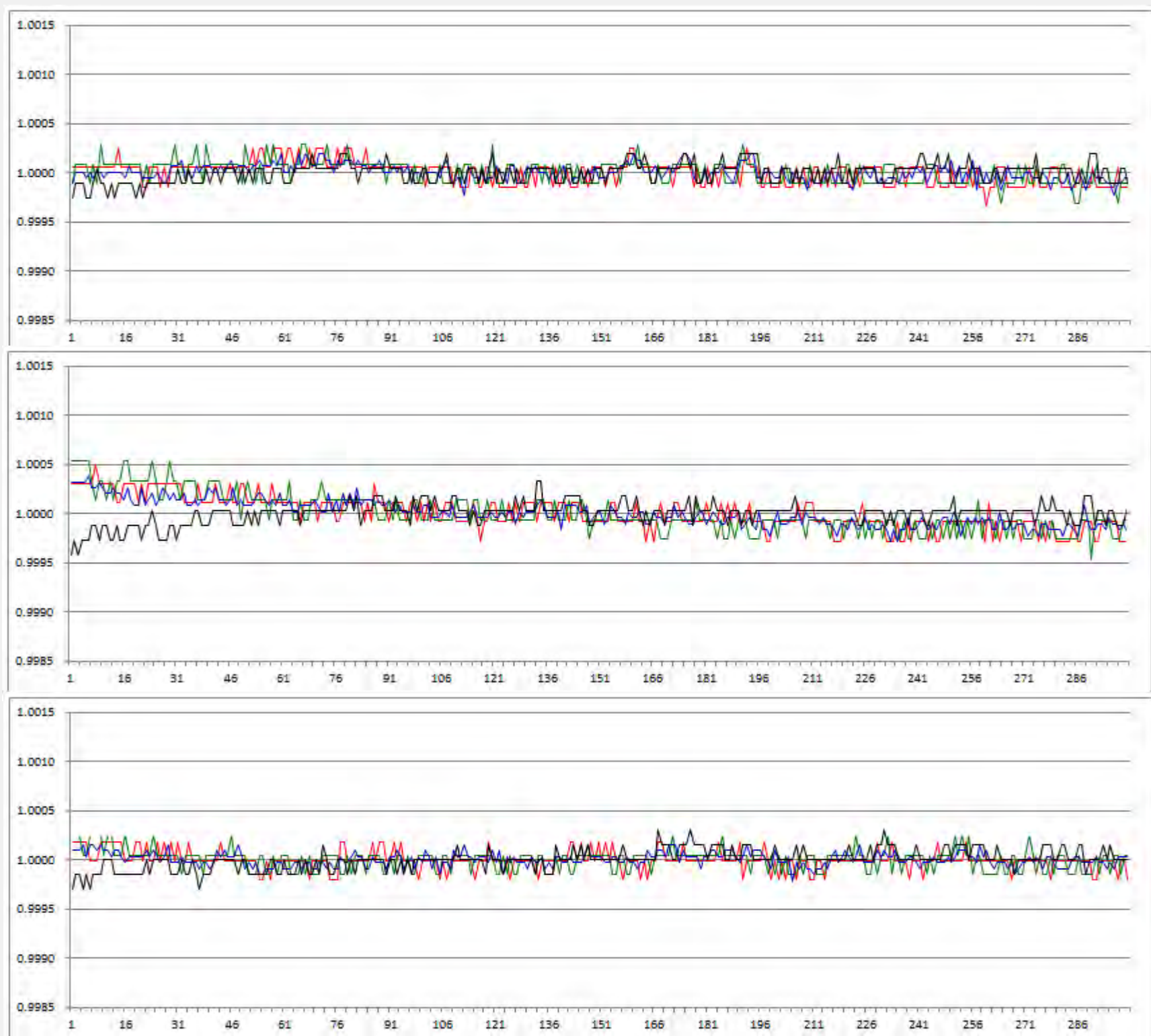
比色計並聯不同電容的穩定性測試數據



光源由色彩感測器TCS3200供電的之正規化色彩偵測值的折線圖



光源由 Arduino VCC 供電限流電阻 30Ω 之正規化色彩偵測值的折線圖



並聯鉅質電容之正規化色彩偵測值的折線圖

(三) 溶液的彩色值偵測程式

熱機

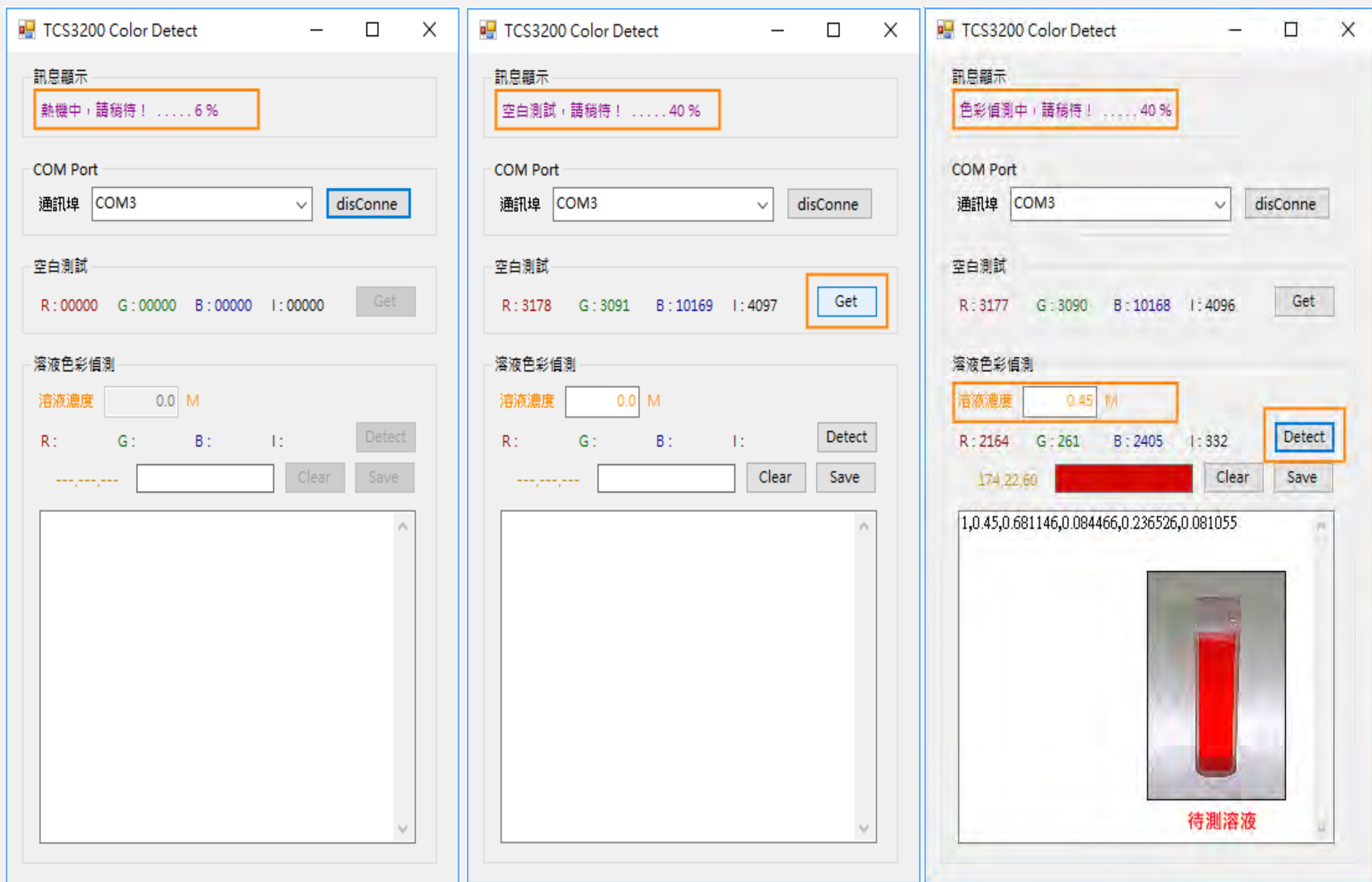
由色彩偵測值的折線圖可以看出，Arduino、色彩感測器及LED光源需要一段熱機的時間，其偵測值就會漸趨穩定。

空白測試

每次開機後都需以蒸餾水進行一次空白測試（白平衡），所得的數據就當成本次測試過程中，色彩感測器能夠偵測到各色頻的最大值（光源白點）。

偵測數據正規化

較容易看出偵測值與光源白點的比例關係，且也不受色彩感測器能夠偵測到各色頻的最大值大小不一致的影響。



比色計熱機中

空白測試進行中

有色溶液彩色值偵測

濃度估測函數的尋找

以多元多項式迴歸分析法來尋找最佳的濃度估測函數，由色彩感測器 TCS3200 接收到的四種訊號（紅色、綠色、藍色、強度）來估測出溶液的濃度。

以硫酸銅的二階估測函數為例：

$$M' = f(R, G, B, I) = -3.84 \times 10^{-9} + 1.2 \times 10^1 \times R + 3.53 \times 10^1 \times G - 4.86 \times 10^1 \times B + 2.26 \times I - 5.99 \times 10^2 \times RG + 6.44 \times 10^2 \times RB + 6.25 \times 10^1 \times RI + 2.47 \times 10^3 \times GB - 5.64 \times 10^2 \times GI + 2.95 \times 10^1 \times BI - 6.15 \times 10^1 \times R^2 - 6.83 \times 10^2 \times G^2 - 1.54 \times 10^3 \times B^2 + 2.39 \times 10^2 \times I^2$$

用矩陣的型態來表示四維 n 階多項式估測函數，然後再利用多元多項式迴歸分析法來求得最佳的估測函數，以四元 2 階的多項式估測函數為例，以矩陣的表示法為：

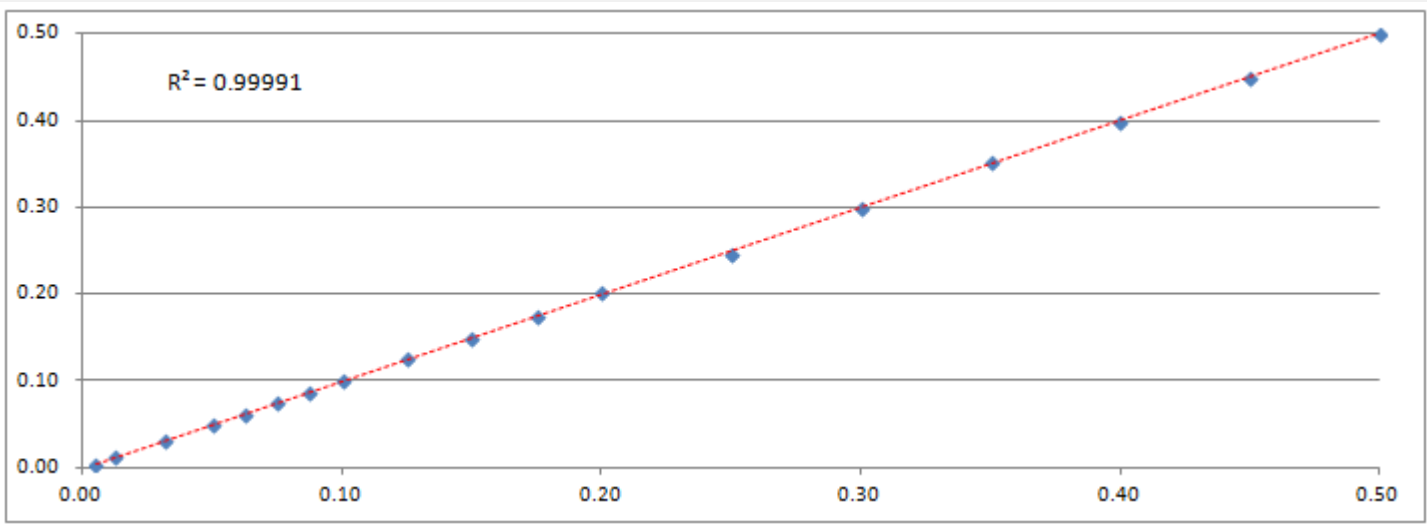
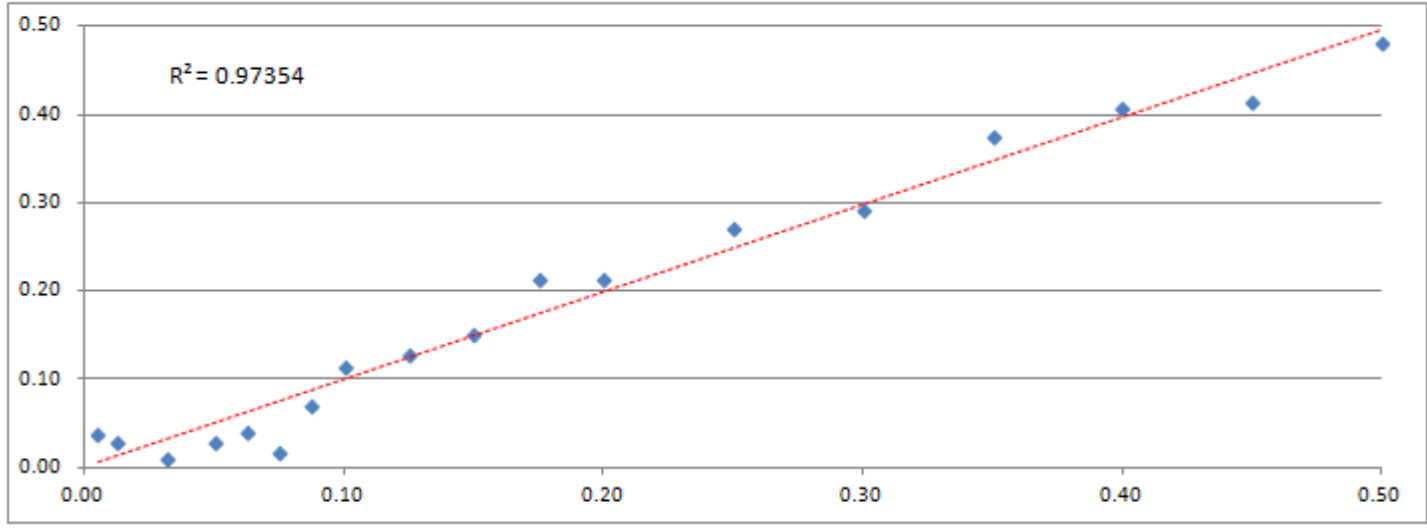
$$M' = \begin{bmatrix} 1 & R & G & B & I & RG & RB & RI & GB & GI & BI & R^2 & G^2 & B^2 & I^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{14} \end{bmatrix}$$

利用多元多項式迴歸分析法求得最佳的估測函數，則表示最佳估測函數的係數矩陣為

$$C = (P^T P)^{-1} P^T M$$

CuSO₄ 及 FeSCN²⁺ 離子溶液的濃度偵測

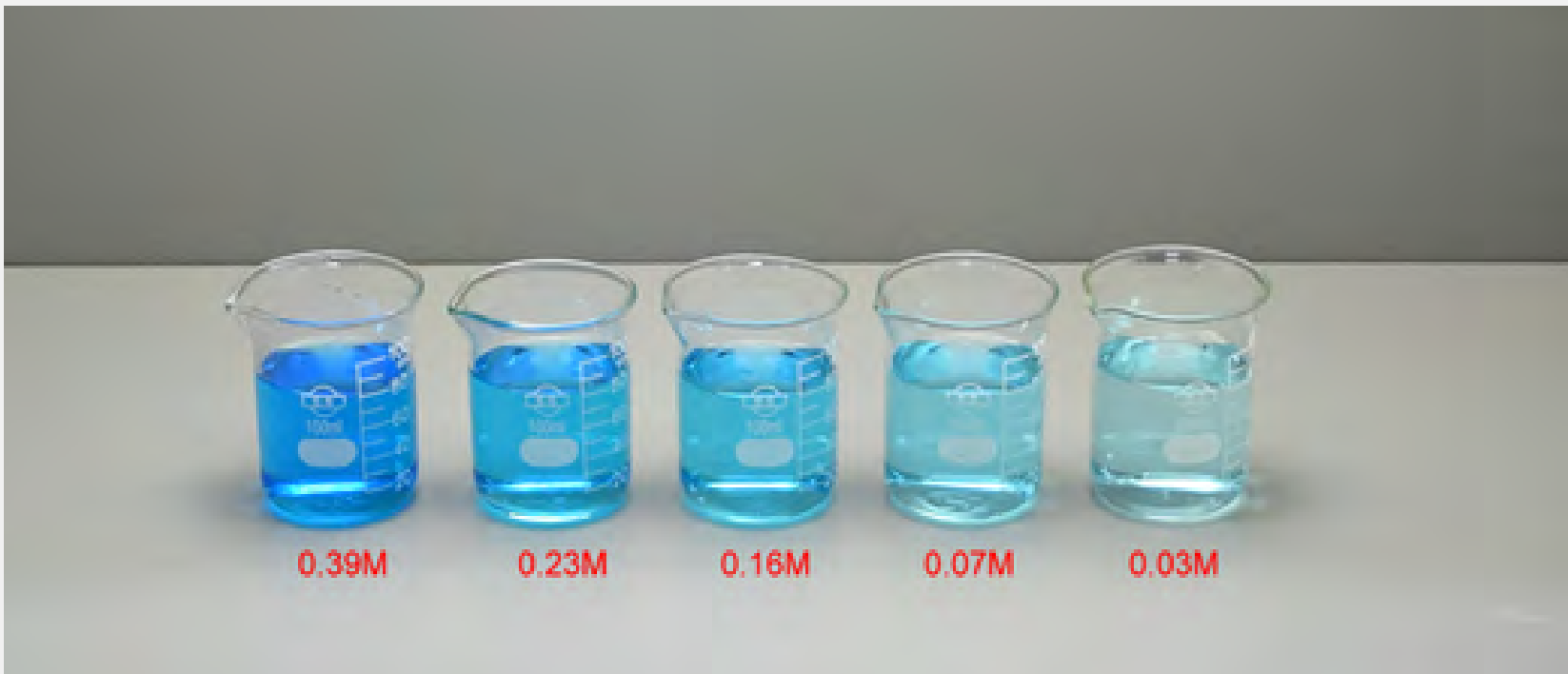
- 配製18種不同濃度的溶液
- 將自製比色計開機並進行空白測試。
- 以自行撰寫的程式記錄下18種不同濃度的溶液均液正規化色彩值。
- 計算出一階及二階的估測函數。



→可看出二階濃度估測函數具有較佳估測能力。

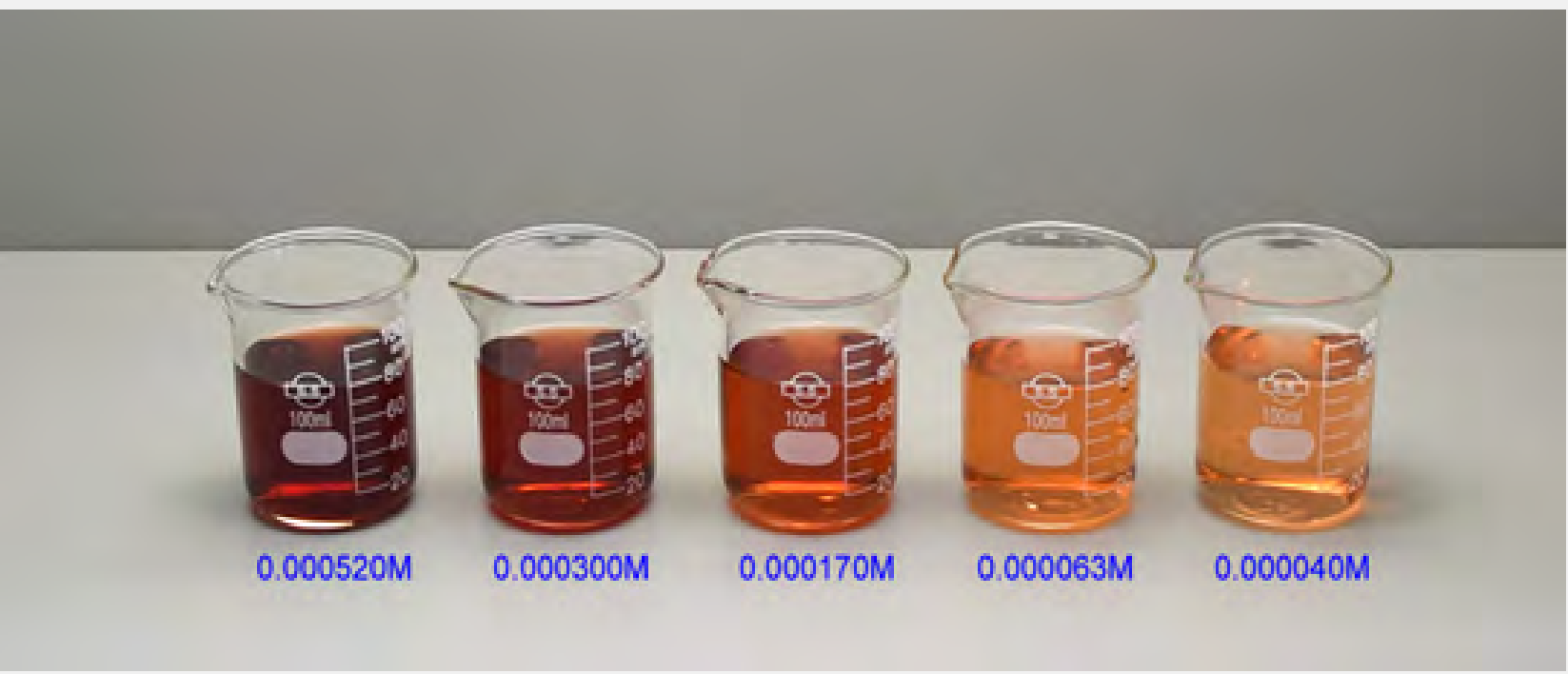
- 另外配製5種不同濃度的溶液，以尋找到的二階估測函數來進行測試。

真實濃度 (M)	0.3900	0.2300	0.1600	0.0700	0.0300
估測濃度 (M)	0.3966	0.2257	0.1556	0.0669	0.0309
誤差(%)	1.70%	1.89%	2.72%	4.47%	2.94%



平均誤差為 2.75%

真實濃度 (M)	0.0005200	0.0003000	0.0001700	0.0000630	0.00004000
估測濃度 (M)	0.00053982	0.00029873	0.00018092	0.00005894	0.00004070
誤差(%)	3.81%	0.42%	6.42%	6.45%	1.76%



平均誤差為 3.77%

結論

- 一、從實驗結果來看，以Arduino及TCS3200確實可以測量到準確的溶液濃度，進而達到減少溶液浪費之目的。且所有實驗器材均容易取得、價格便宜，裝置簡單易於操作，使結果和實驗目的相符。
- 二、使用紅色、綠色、藍色、強度等四種偵測值，搭配多元多項式迴歸分析法所尋找的濃度估測函數，可以提高未知溶液的濃度偵測精確度。
- 三、為使LED燈亮度增強，我們改變其電路並發現當電阻為30Ω時，可使亮度提高也依舊維持其穩定性。而為使自製比色計之色彩偵測更加精準，需要並聯電容來穩定。而相較於其他電容，鉭質電容可以更有效地穩定光源的強度。
- 四、色彩偵測程式方面，我們根據Arduino及TCS3200之特性，在每次偵測前給予熱機時間，並以空白測試測到的值使每次測量的數據正規化。這樣不僅容易看出其比例關係，也讓裝置較不受白平衡不同影響，且以多次偵測之平均值取代單次偵測，避免取得極端值，能使測量結果更準確。