

中華民國第 60 屆中小學科學展覽會 作品說明書

高級中等學校組 工程學(一)科

佳作

052321

Google 眼鏡電錶

學校名稱：國立苗栗高級農工職業學校

作者： 職二 李俊德 職三 李健銘 職三 彭聖凱	指導老師： 羅昌財 林己超
---	-----------------------------

關鍵詞：Google 眼鏡、Arduino

摘要

本作品的特色是利用具有藍芽傳輸功能的三用電錶，透過藍芽傳輸資料到自製的 Google 眼鏡上，再透過 OLED 顯示器和鏡子、凸透鏡呈像於一塊透明面板上，透過眼睛的視線即可直接讀取到由藍芽電錶傳輸過來的數據，讓使用者可以更加方便又安全的使用三用電錶。

壹、研究動機

從高一開始就經常使用到三用電錶，發覺三用電錶在電路檢測上真的非常重要又實用，但是很常在量測電壓時，就會遇到一個問題，那就是一邊要注意量測的點，又要轉頭看一下錶頭上的指示值，往往頭一轉過去看錶頭的指示值時，量測的測試棒又在不注意的情況下鬆脫了，結果又常常量測不到正確的數值，總是要一邊注意測試棒有沒有固定好，一邊又要讀取電錶的數值，有時不小心還會觸電。

到底有沒有什麼三用電錶？可以讓視線在看著測試棒的同時，不用把頭轉來轉去就可以看到電錶的指示值呢？如果三用電錶可以像 Google 眼鏡一樣，將量測到的數據顯示在 Google 眼鏡上，一定帥呆了，就可以解決常久以來使用三用電錶的困擾，於是有了想要自己動手作一個 Google 眼鏡電錶的動機。

貳、研究目的

有薦於 Google 眼鏡的價格太過於昂貴，於是本研究的目的為利用學校所學，以有限的資源自製學生級的 Google 眼鏡，具體設計特色為：

- 一、參考 Google 眼鏡的圖片，利用 SolidWorks 繪製原型圖。
- 二、利用 3D 列印機，列印可完成的作品框架。
- 三、利用藍芽模組與市售的藍芽電錶連結即可完成資料的讀取。
- 四、讓有興趣自製作品的人也可以參考本研究的作品順利完成製作或改良。

參、研究設備及器材

本研究使用的設備及材料，如表 1。

表 1. 使用設備與工具

		
1. Arduino Nano 開發板	2. 藍芽 4.0 模組	3. 0.66 英吋 OLED
		
4. 3.7V 鋰電池	5. 鋰電池專用充電電路板	6. 放大鏡
		
7. 鏡子	8. OW18B 數位電表	9. 壓克力板

肆、研究過程或方法

研究方法為先由線上查詢、圖書館及老師的建議，搜集相關知識後，決定系統架構，並且依據研究目的，利用電腦繪圖軟體進行機構與電路設計，而整個系統以 Arduino 單晶片為系統核心，經由實際性的操作以測試系統的完整性，再進行修改，最後完成研究報告，研究方法流程如圖 1.所示。

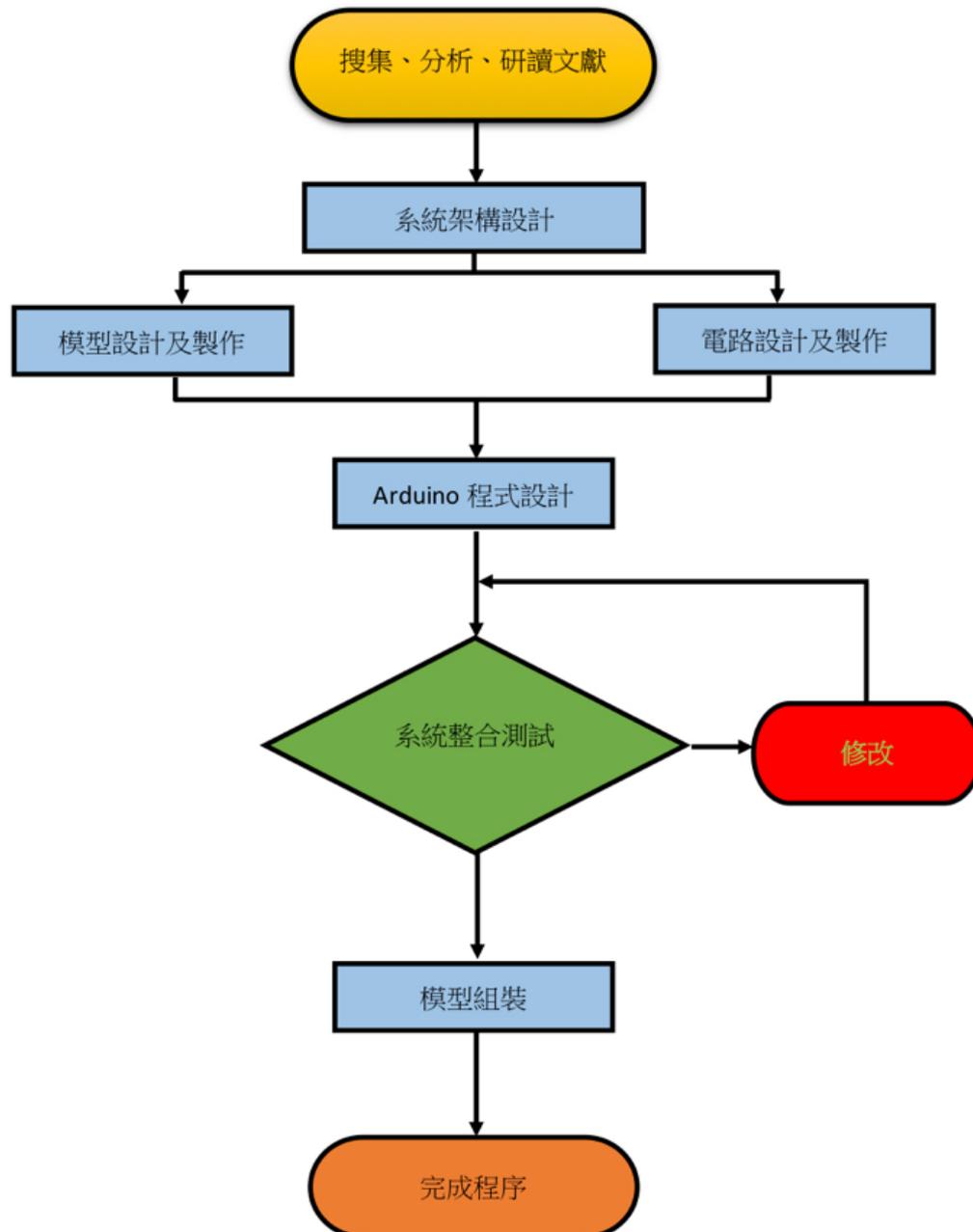


圖 1. 研究方法流程圖

一、系統架構設計

本研究發想的作品為利用 Arduino 開發板整合測試的方便特性，配合電腦軟體、藍芽電錶、…等的機構與電路設計，自製學生級的 Google 眼鏡，系統設計架構圖，如圖 2.所示。

完整系統設計模型如圖 3.所示，為利用利用藍芽無線傳輸資料至系統裝置上的 Arduino 單晶片，由 Arduino 單晶片再依據所接收的資料，將量測出的指示值，由 OLED 顯示，在透過反射鏡、放大鏡、壓克力板，將量測數值顯示出來。

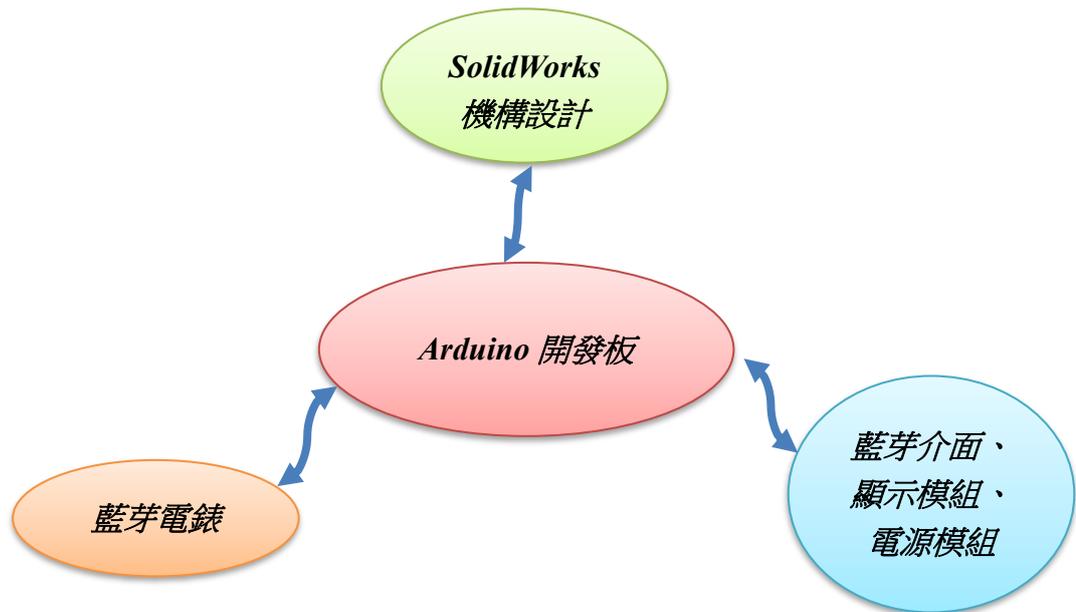


圖 2. 系統設計架構圖

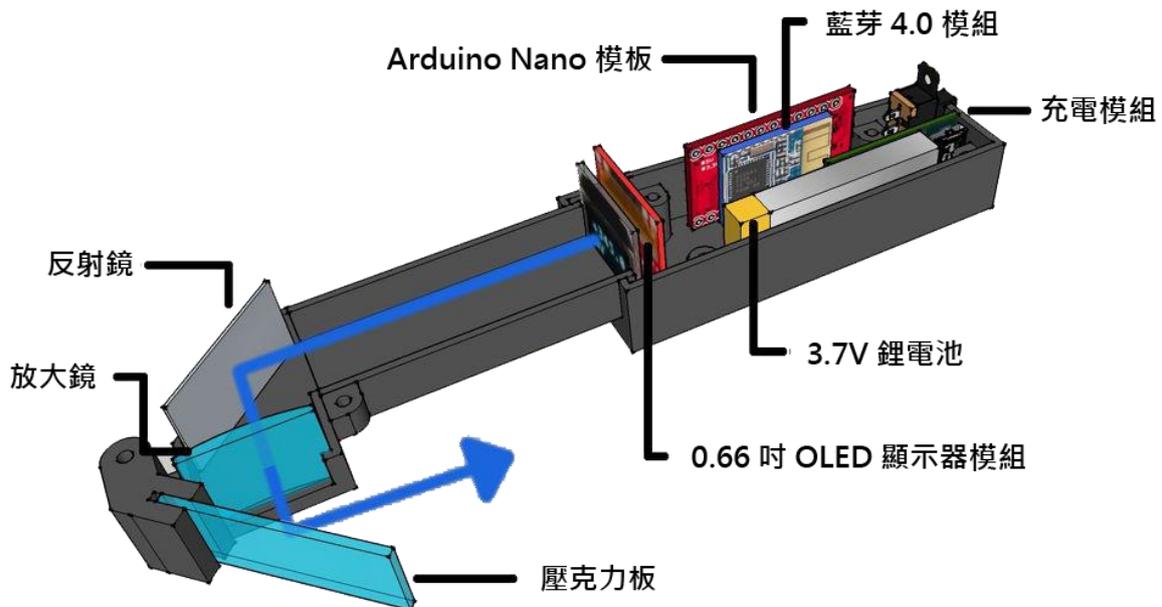


圖 3. 完整系統設計模型

二、控制模板選擇

為了讓模板可以放入設計的眼鏡空間又不需要花費太多的費用，選擇目前較常使用的 Arduino 模組會比較適合，下表 2.為常用 Arduino 模組比較表。

表 2. 常用 Arduino 模組比較表

	UNO	Nano 3.0	mini
MCU	ATmega328P	ATmega328P	ATmega328
工作電壓	5V	5V	5V
輸入電壓	7-12V	7-12V	7-9V
數位 IO 接腳	14	14	14
類比 IO 接腳	D(6)/S(8)	8	8
PWM	6	6	6
時脈	16M	16M	16M
Flash	32K	32K	32K
SRAM	2K	2K	2K
EEPROM	1K	1K	1K
USB 晶片	ATmega16u2	FT232	無
特色	目前使用最多，穩定性最好	功能和 UNO 一樣，並且體積更小	最小控制板，燒錄程式需要外接 USB 轉 TTL

考量了尺寸、價位和程式燒錄方式等因素，本次的研究選擇 Arduino Nano3.0 開發板為主要的控制模板。

三、模型設計及 3D 繪圖

此機構模型設計是參考 Google 眼鏡的原型加以改良設計，利用 SolidWorks 軟體，繪製機構組件及模擬組合圖，如圖 4.所示。

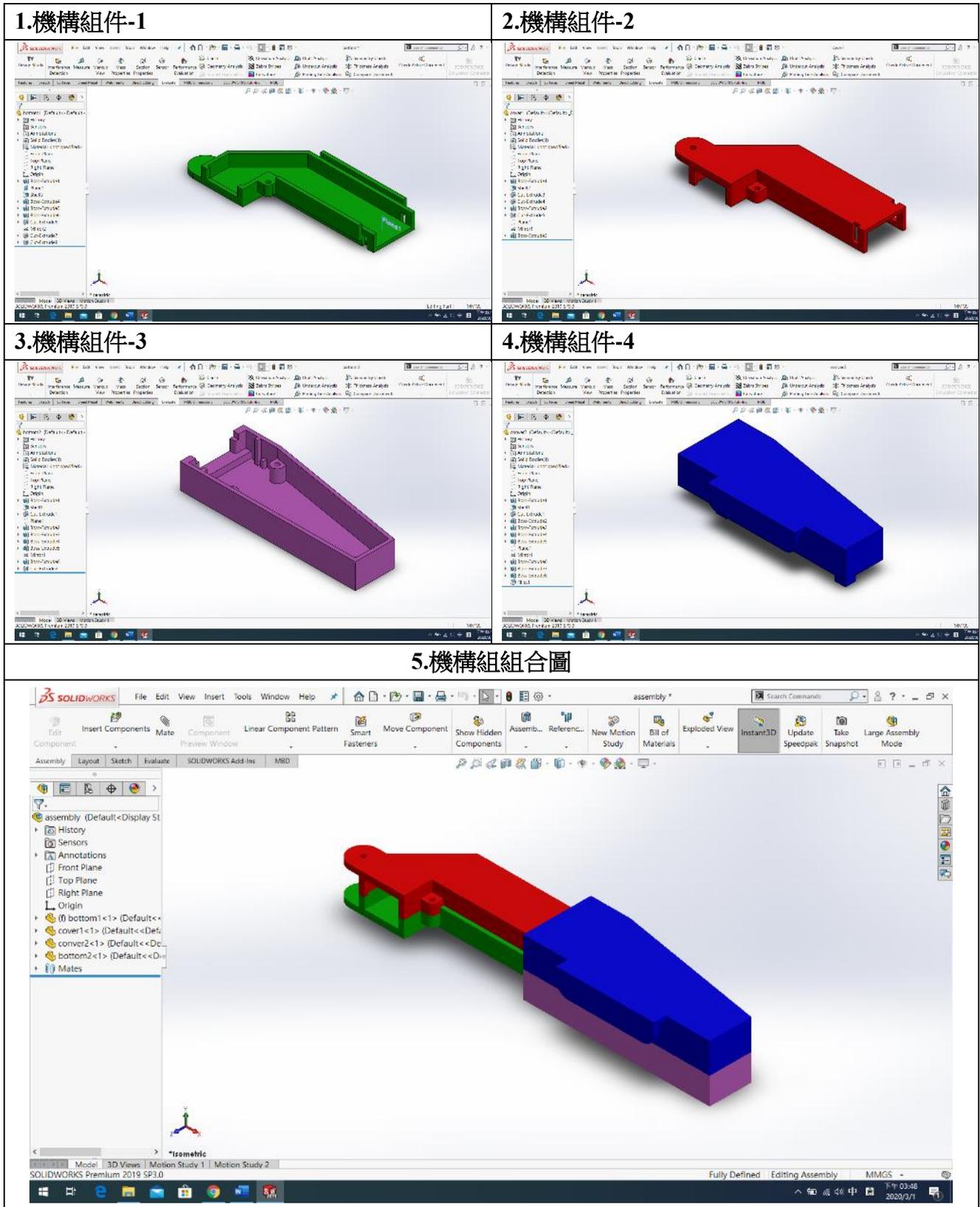


圖 4. 機構組件及模擬組合圖

四、電路設計及製作

(一)、Arduino Nano 開發板概要

Arduino Nano 就是迷你版的 Arduino UNO，如圖 5.所示，擁有同樣的規格如表 3.所示，且增強版在保證 Nano 的性能不變原則下，更換了 USB 晶片，以提高下載的速度及增加在 win7 win8 的穩定性。

Nano 的體積小巧，可以用來開發出獨立運作具有互動效果的電子產品，也可與電腦相連接開發出與 Flash,Processing 等軟體一同合作完成的互動作品。

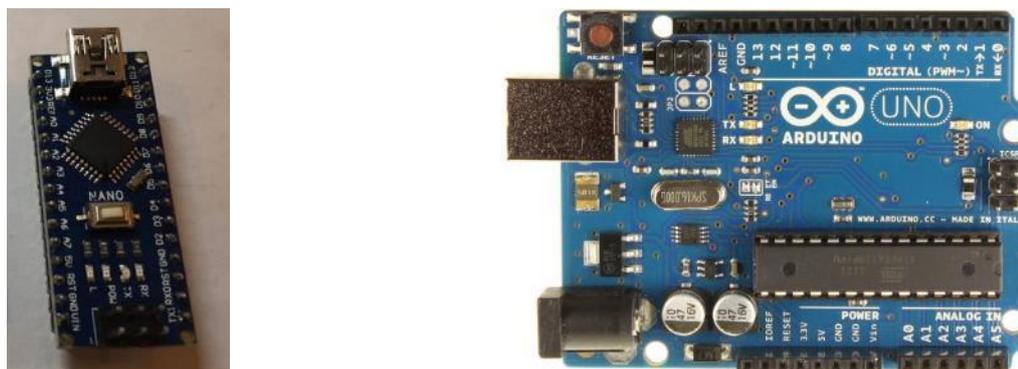


圖 5. Arduino Nano 板與 Arduino UNO 板

表 3. Arduino Nano 規格表

名稱	說明
數位輸出入	14 個 Digital I/O (包括 TX, RX, D2~D13；其中 TX、RX 為 TTL 電位之串口收發埠)
類比輸入	8 個 Analog I/O (A0~A7)
PWM 輸出	6 個 PWM 埠 (D3, D5, D6, D9, D10, D11)
MU 晶片	採用原廠 Atmel Atmega328P-AU 單晶片
燒錄方式	支援 ISP(In System Programmer 線上燒錄)下載功能
供電方式	1. 支持 USB 下載及供電 2. 支持外接 5V~12V 直流電源供電 (若工作電流較大，建議外接輸入 Vin=5~7V) 3. 支援 9V 電池供電 4. 三種供電方式：USB、VIN、外部 5V 輸入
輸出特性	輸出電壓：5V 直流電壓輸出和 3.3V 直流電壓輸出。 (注意：當電路需求較大輸出電流時，例如超過 0.2A，請另備電源供給)
尺寸	長-45mm，寬-18mm。
注意：Nano 之 Analog I/O 多出了 A6、A7，A6 和 A7 只能純當 Analog I/O 使用，無法像 A0~A5 也可以充當 Digital I/O 使用	

(二)、0.66 吋 OLED

0.66 吋 OLED 使用的介面為 I2C 介面，只需使用 SDA 及 SLC 線連接 Arduino 電路板的 A4 及 A5 兩條線即可，大大減少 Arduino 輸出及輸入接腳的使用，其詳細的規格如表 4。

表 4. 0.66 吋 OLED 規格表

項目	規格說明	單位
點陣(解析度)	64 × 48	Dots
模組尺寸	19.9 × 23.1 × 2.4	mm
有效區域	13.42 × 10.06	mm
像素大小	0.185 × 0.185	mm
像素間距	0.210 × 0.210	mm
顯示模式	Passive Matrix 被動矩陣	
顯示顏色	單色	
驅動方式	1/48 Duty	
IC	SSD1306BZ	
通訊介面	I2C	

(三)、藍芽 4.0 BLE 模組

本研究的藍芽模組支持 UART 接口，並支持 SPP 藍牙串口協議，具有成本低、體積小、功耗低、收發靈敏性高等優點，只需配備少許的外圍元件就能實現其強大功能，其特點為：

- 1.藍牙協議：Bluetooth Specification V4.0 BLE,收發沒有字節限制。
- 2.工作頻率：2.4GHz ISM band。
- 3.調製方式：GFSK(Gaussian Frequency Shift Keying)。
- 4.靈敏度：≤-84dBm at 0.1% BER。
- 5.傳輸速率：Asynchronous: 6 kbps Synchronous: 6 kbps。
- 6.安全特性：Authentication and encryption。
- 7.支持服務：Central & Peripheral UUID FFE0,FFE1。
- 8.功耗：自動休眠模式下，待機電流 400uA~1.5mA,傳輸時 8.5mA。

最後本研究整合所需之模組及電源充電模組設計出完整系統所需之電路硬體設計，如圖

6.所示。

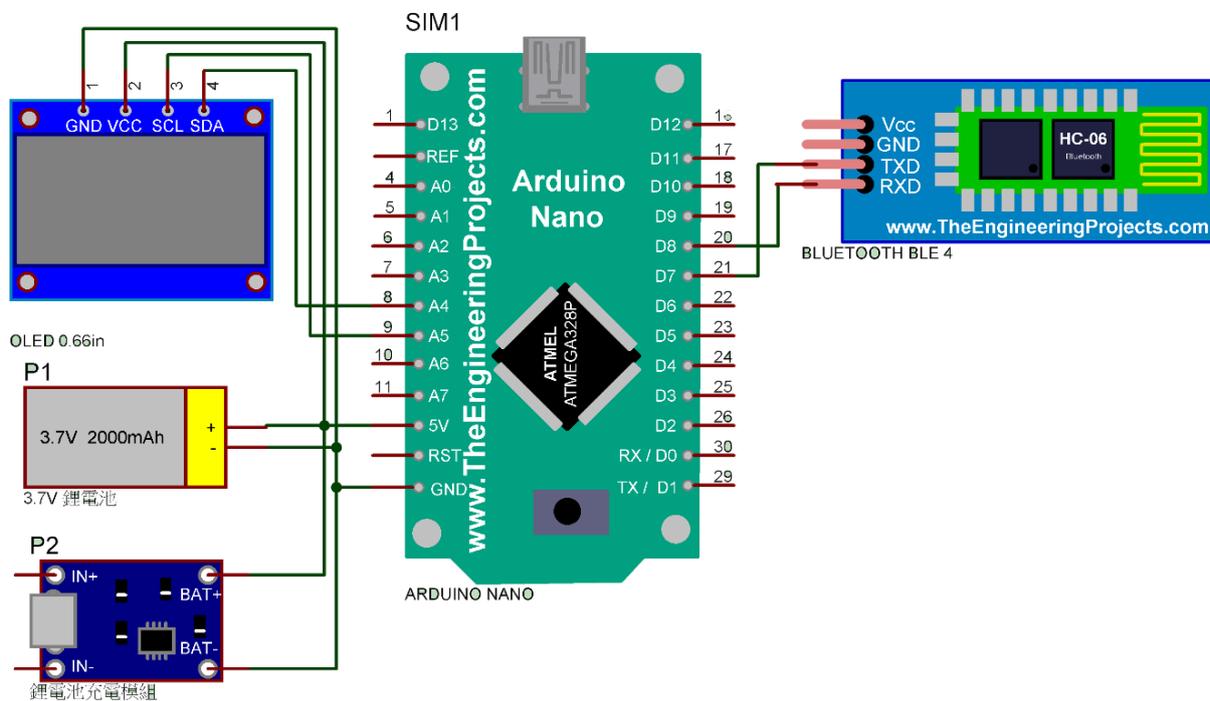


圖 6. 電路硬體設計圖

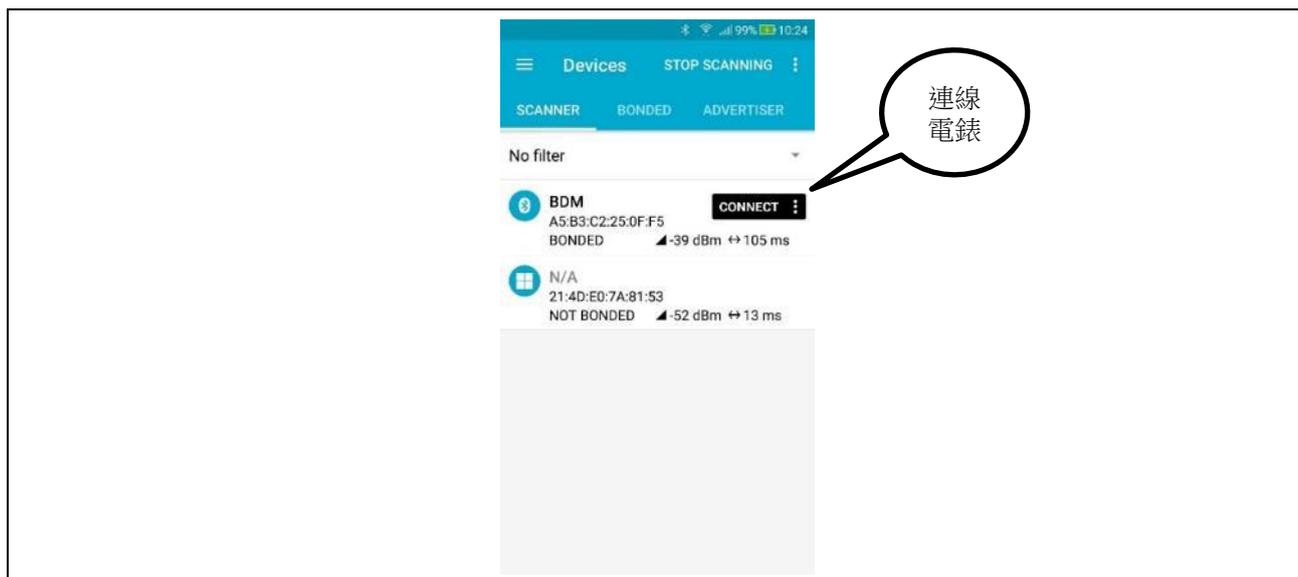
五、OW18B 藍芽電錶傳送資料偵測記錄實驗

使用設備：OW18B 藍芽電錶、手機(安裝 nRF Connect APP)，測試步驟：

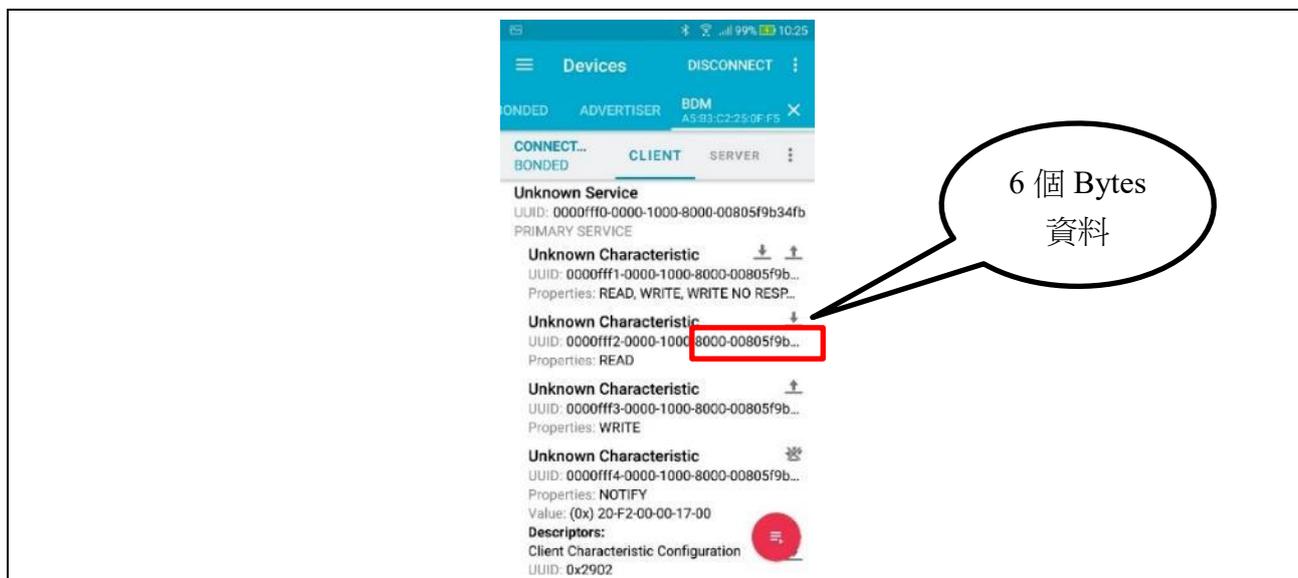
1. 開啟 OW18B 電源並啟動藍芽功能。



2. 啟動手機 nRF Connect APP，再與電錶連線。



3. 轉動電錶檔位，監看記錄資料變化。



經由多次的實驗測試，從手機藍芽接收到由藍芽電錶測量後所傳輸出來的完整數據如表 5.所示，再依據我們研究分析得知參數與藍芽電錶目前的檔位與數值之關係，可知接到的數據中有 6 個 Bytes 的參數值，是主要電錶顯示測量數值與檔位的資料。

表 5. 藍芽電錶記錄資料表

檔位		Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
AC 電壓	Manual	mV	59	F0	手動 0 自動 4	0	Low Byte High Byte
		無小數點	60	F0		0	
		小數點 1 位	61	F0		0	
		小數點 2 位	62	F0		0	
		小數點 3 位	63	F0		0	

DC 電壓	Manual	mV	19	F0		0	Low Byte	High Byte	
	V	無小數點	20	F0		0		負值 8X	
	V	小數點 1 位	21	F0		0		負值 8X	
	V	小數點 2 位	22	F0		0		負值 8X	
	V	小數點 3 位	23	F0		0		負值 8X	
電阻	手動 Ω	小數點 1 位	21	F1		0	Low Byte	High Byte	
	手動 K Ω	小數點 3 位	2B	F1		0			
		小數點 2 位	2A	F1		0			
		小數點 1 位	29	F1		0			
	手動 M Ω	小數點 3 位	33	F1		0			
		小數點 2 位	32	F1		0			
	M Ω OL			37	F1				
	Ω OL			27	F1				
K Ω OL			2F	F1					
二極體	導通電壓	小數點 3 位	A3	F2	0	0			
	未導通	.OL	A7	F2	0	0	FF	7F	
是否短路	短路		E1	F2	0	0	電阻低於 30 會響		
	未短路		E7	F2	0	0	FF	7F	
電容	nF	小數點 2 位	4A	F1	4	0			
頻率	Hz	小數點 1 位	A1	F1	4	0			
		小數點 2 位	A2	F1	4	0			
		小數點 3 位	A3	F1	4	0			
溫度	攝氏		20	F2	0	0			
	華氏		60	F2	0	0			
無接觸電壓偵測			60	F3	0	0	01 有電		
hfe			20	F3					
DC 電流	uA	小數點 1 位	91	F0	0	0	Low Byte	High Byte	
		無小數點	90	F0	0	0		負值 8X	
	mA	小數點 1 位	99	F0	0	0		負值 8X	
		小數點 2 位	9A	F0	0	0		負值 8X	
	A	小數點 2 位	A2	F0	0	0		負值 8X	
AC 電流	uA	無小數點	D0	F0	0	0	Low Byte	High Byte	
		小數點 1 位	D1	F0	0	0			
	mA	小數點 1 位	D9	F0	0	0			
		小數點 2 位	DA	F0	0	0			
	A	小數點 2 位	E2	F0	0	0			

由實驗數據中，分析出各位元組所代表數值的說明如下：

- (1) Byte1、Byte2：表示為檔位與所測量的解析度單位。
- (2) Byte3：為手動(0)或自動(4)。
- (3) Byte4 無作用。
- (4) Byte5、Byte6 為測量的高低位元數值。

例如：測量交流 220V 電壓值，電壓錶設定：檔位(ACV)、無小數點、手動。

手機接收到的藍芽傳送訊息： **UUID：0000fff2-0000-1000-00dc-0000f060**

Byte6	Byte5	Byte4	Byte3	Byte2	Byte1
00	DC	00	00	F0	60

我們就可以利用實驗分析所得的參數來建立 Arduino 程式所需要的資料庫，做為資料轉換，以用來在 OLED 顯示測量出來的實際測量數據。

六、Arduino 程式設計

(一)、藍芽設定

1. 首先利用 Arduino SoftwareSerial 的函數建立序列埠實體。

```
/*-----產生藍芽通訊實體-----*/  
SoftwareSerial mySerial(7, 8); // RX, TX  
/*-----*/
```

2. 再利用 Arduino 送出 AT+CON 指令至藍芽模組，連線方式採用連接藍芽模組 MAC 位址的方式連接。

```
mySerial.begin(9600);  
mySerial.println("AT+CONA0xa5b3c2250ff5");
```

3. 自建函數讀取由電錶傳送至 Arduino 模組的資料。

```
/*-----讀取由電錶傳送過來的資料-----*/  
getString = "";  
do {  
  a++;  
  datas[a] = mySerial.read();  
  getString += (char)datas[a];  
  delay(4);  
} while (datas[a - 1] != 13 && datas[a] != 10 && mySerial.available() != 0);  
mySerial.read();  
mySerial.read();  
/*-----*/
```

(二)、OLED 設定

1. OLED 採用 SSD1306 晶片控制，首先要載入 Adafruit 的 SSD1306 函數庫。

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 0 // GPIO0
Adafruit_SSD1306 display(OLED_RESET);
```

2. 建立自建顯示字型資料。

```
/*-----OLED自建顯示字型檔-----*/
static const unsigned char PROGMEM words[] =
{ //電-1
  0x7F, 0xC0, 0x04, 0x00, 0xFF, 0xF0, 0xA5, 0x10, 0x94, 0x90, 0x7F, 0xC0, 0x44, 0x40, 0x7F, 0xC0,
  0x44, 0x40, 0x7F, 0xD0, 0x04, 0x10, 0x07, 0xF0, 0x00, 0x00,
  //壓-2
  0x40, 0x20, 0x7F, 0xF0, 0x52, 0x40, 0x5E, 0x50, 0x5B, 0xF0, 0x56, 0x40, 0x5A, 0xA0, 0x67, 0x10,
  0x42, 0x00, 0x5F, 0xE0, 0x82, 0x00, 0xBF, 0xF0, 0x00, 0x00,
  //阻-3
  0x94, 0x40, 0xF7, 0xE0, 0x94, 0x40, 0xA4, 0x40, 0xA7, 0xC0, 0x94, 0x40, 0x94, 0x40, 0x97, 0xC0,
  0xF4, 0x40, 0x84, 0x40, 0x84, 0x40, 0x9F, 0xF0, 0x00, 0x00,
  //容-4
  0x04, 0x00, 0x42, 0x20, 0x7F, 0xF0, 0xD0, 0xA0, 0x26, 0x40, 0x49, 0x20, 0x10, 0x80, 0x20, 0x40,
  0xDF, 0xB0, 0x10, 0x80, 0x10, 0x80, 0x1F, 0x80, 0x00, 0x00,
```

3. 利用 display 函數將要顯示的文字及圖案顯示於 OLED 上。

```
/*-----*/
String dig1 = "";
String dig2 = "";
if (datas[5] < 16) dig1 = "0";
if (datas[6] < 16) dig2 = "0";
/*取得資料的第5和第6個位元的值*/
String val = dig1 + String(datas[5], HEX) + dig2 + String(datas[6], HEX);
char buf[5];
/* FSB buf[2] buf[3] buf[0] buf[1] */
val.toCharArray(buf, 5);
int values;
/* 如果出現真的直流電壓 */
if (buf[2] == '8') {
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 24);
  display.println("-");
  values = charToInt(buf[0]) * 16 + charToInt(buf[1]) + charToInt(buf[3]) * 256;
} else {
  values = charToInt(buf[0]) * 16 + charToInt(buf[1]) + charToInt(buf[2]) * 4096 + charToInt(buf[3]) * 256;
}
display.setTextSize(2);
display.setTextColor(WHITE);
display.setCursor(4, 20);
byte val3 = values % 10;
byte val2 = (values / 10) % 10;
byte val1 = (values / 100) % 10;
```

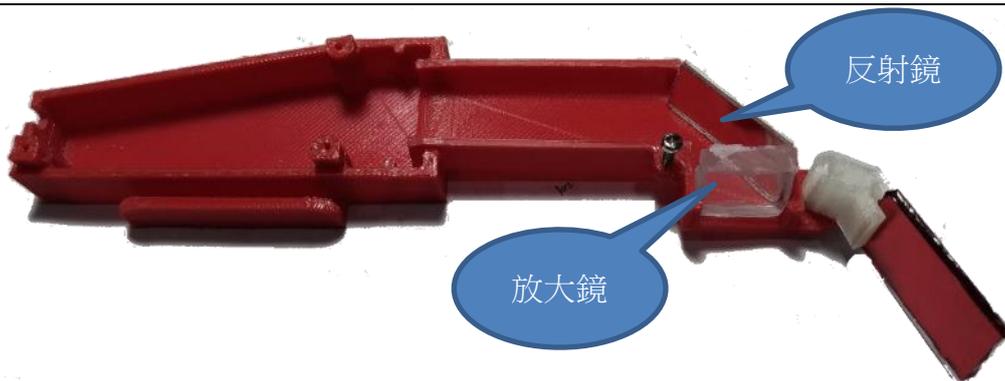
```

byte val0 = values / 1000;
String dispVal = "";
if (divide == 0) dispVal = String(val0) + String(val1) + String(val2) + String(val3);
if (divide == 1) dispVal = String(val0) + String(val1) + String(val2) + "." + String(val3);
if (divide == 2) dispVal = String(val0) + String(val1) + "." + String(val2) + String(val3);
if (divide == 3) dispVal = String(val0) + "." + String(val1) + String(val2) + String(val3);
banner(datas[1], datas[2]);
display_banner(num1, num2, num3, num4);
if (val == "ff7f") display.println("O.L");
else if (datas[1] == 0x60 && datas[2] == 0xf3 ) {
    if (buf[1] == '0') display.println(" EF");
    if (buf[1] == '1') display.println("-");
    if (buf[1] == '2') display.println("--");
    if (buf[1] == '3') display.println("----");
    if (buf[1] == '4') display.println("-----");
    unit = "NCV";
}
else display.println(dispVal);
display.setTextSize(1);
display.setCursor(42 , 40);
display.println(unit);
if (resistor == 1) {
    display.drawBitmap(55, 40, omor, 16, 11, 1); //歐姆符號
    if (values <= 30) display.drawBitmap(10, 40, horn, 16, 11, 1); //喇叭圖示
}
display.display();
delay(2);

```

七、硬體組裝

1.組合外殼、反射鏡、放大鏡



2.置入硬體電路



3.蓋上上蓋完成組裝



圖 7. 硬體組裝流程

伍、研究結果

一、作品測試結果

1.穿戴上自製 Google 眼鏡電錶	2.自動與電錶建立藍芽連線
	
3.測量所需檢測之電壓端點	4.顯示交流電壓值於視線前方
	

圖 8. 作品測試結果

二、作品特點

1. 本研究自製的 Google 眼鏡電錶，讓使用者在使用電錶時，可以更方便的操作和讀取數值，避免掉人為操作上的危險，特別是高電壓及狹窄工作空間時，除方便測量，更可以保障使用者的安全。
2. Google 眼鏡電錶在外型雖無法與市售的商品相比，但自製時所花費的費用卻不到市售商品的十分之一，對於沒有經費預算的人來說，不失為一種極佳的選擇。
3. Google 眼鏡電錶在設計外型開發的過程中，是利用 SolidWorks 軟體設計，再實際由 3D 列印機印出，成品若不理想也可以即時透過軟體修改，除加速了開發的速度，也省下了很多額外的費用。

三、自製作品費用

表 6. 自製 Google 眼鏡電錶費用表

名稱	數量	價格	合計
Arduino NANO	1 個	100 元	100 元
0.66 吋 OLED	1 個	120 元	120 元
藍芽 4.0BLE 模組	1 個	100 元	100 元
鋰電池充電模組	1 個	50 元	50 元
3.7V 鋰電池	1 個	150 元	150 元
3D 列印耗材、鏡片等	1 套	約 500 元	500 元
總費用合計約：			1020 元

陸、討論

一、Google 眼鏡電錶重量與大小問題

本研究自製 Google 眼鏡電錶在穿戴上，是掛載在使用者本身所戴的眼鏡側邊，如圖 9.所示，在穿戴上會讓使用者感覺重量太重尺寸太大的問題。由於本研究設計主要考量為利用有限資源為出發點來創作，故在電路硬體的選擇上已盡量選擇較小尺寸，在機構的材質上亦是利用學校所提供 3D 列印機的 PLA(Polylactic Acid 聚乳酸)線材列印而成，故在後續的改良上建議可將電路整合縮小尺寸，如此在機構的設計才可進一步的減少所需放置硬體電路的空間，而機構的材質上，可朝向利用輕量金屬的材料設計外殼，如：鋁、鈦…等。

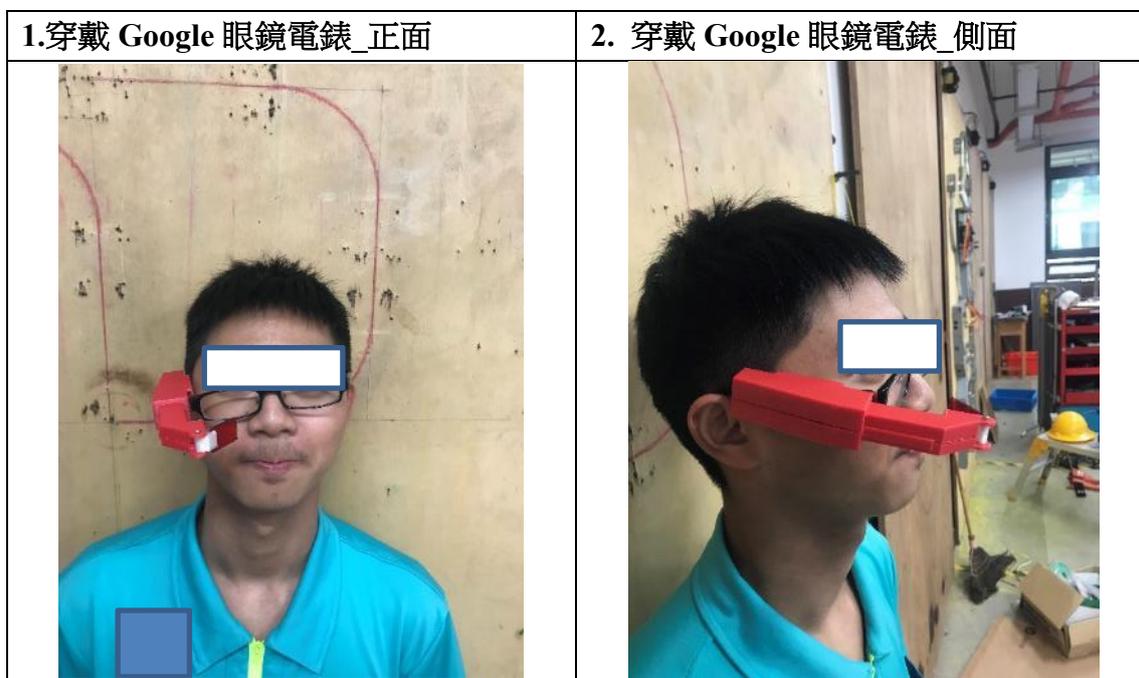


圖 9. 穿戴 Google 眼鏡電錶

二、Google 眼鏡焦距問題

由於自製的 Google 眼鏡電錶是利用反射鏡、放大鏡將 OLED 顯示的數值投射到透明壓克力板上，讓使用者的視線可以同時看到電錶數值及視線前方景物，但在掛載在使用者本身所戴的眼鏡側邊時，投射出來的數值，因每位使用者眼鏡不同及自製 Google 眼鏡機構問題會有不容易對焦的問題如圖 10.所示，目前的改良方式為在 Google 眼鏡外殼上，利用線槽製作可調整掛載間隙的 Google 眼鏡外殼，方便使用者調整焦距，如圖 11.所示。另一種改良方式為利用雙面膠將 Google 眼鏡黏貼在安全帽的側邊，除了方便調整焦距，同時也解決 Google 眼

鏡掛載在眼鏡上造成使用者覺得重量太重問題，如圖 12 所示。

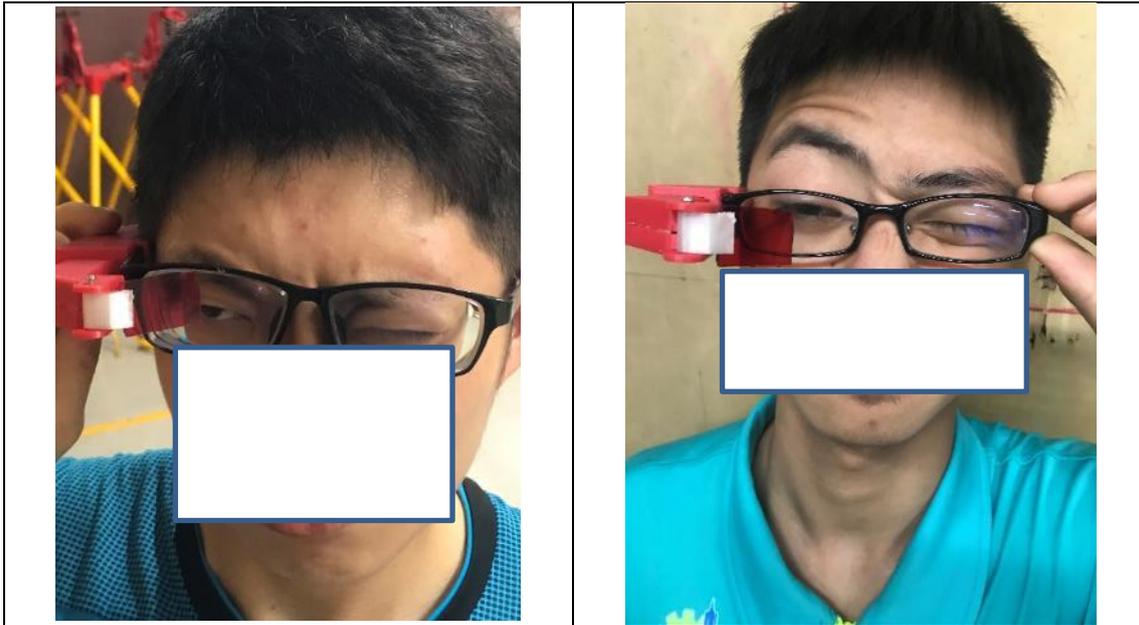


圖10. Google眼鏡焦距問題

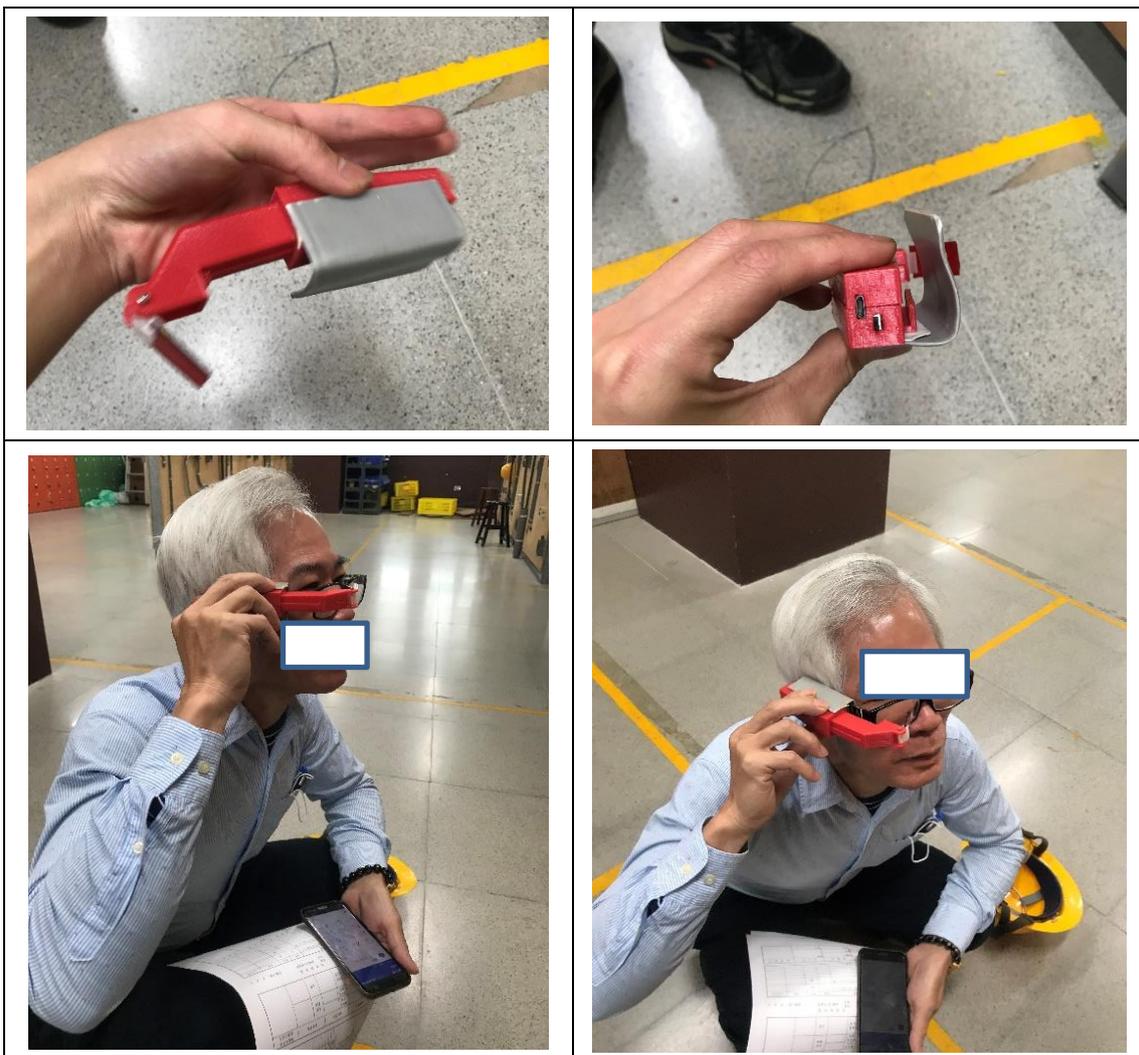


圖11. Google眼鏡可調整掛載間隙

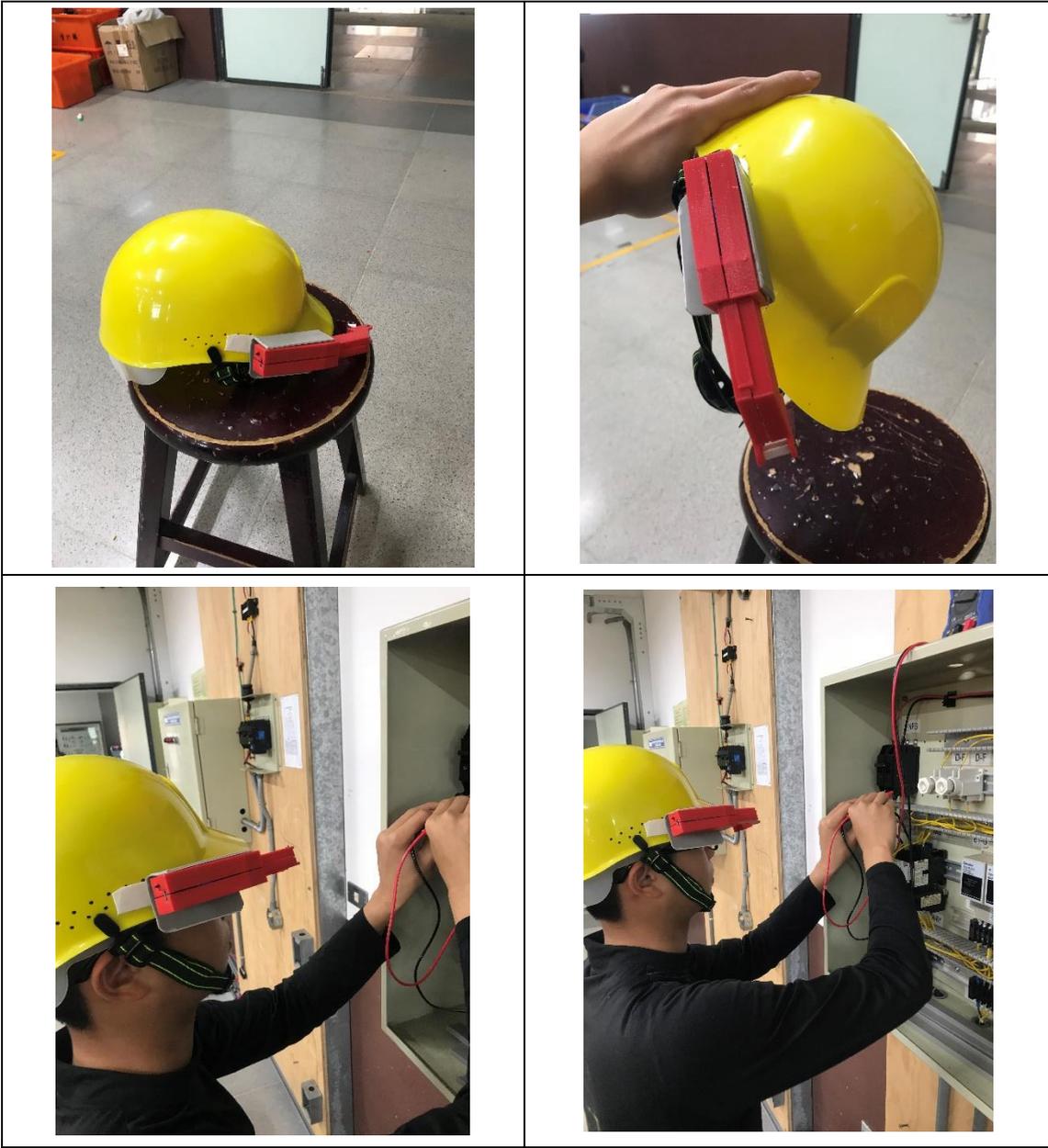


圖12. Google眼鏡黏貼於安全帽

柒、結論

本研究確實能達成研究的目的，自行設計出所需的硬體與機構，創作出類似 Google 眼鏡的測量輔助工具。使用者只需具備有藍芽功能的電錶，就可以透過藍芽傳輸資料到自製的 Google 眼鏡上，再透過 OLED 顯示器，利用反射原理，將畫面呈像於一塊透明面板上，透過眼睛的視線即可輕易讀取到由藍芽電錶傳輸過來的數據，讓使用者可以更加方便又安全的使用三用電錶。

這次的研究學會了很多事，這些都可能是我們高中三年來都很難觸碰到的事物，我們在製作的過程中從最先前的設計概念到裝置外觀都經過了一番討論，在製作過程中遇到了許多瓶頸，例如如何把電腦繪製的裝置外觀實現、材質的選擇、還有內部電路的配置，顯示的調校等等，讓我們體會到一個產品從設計到製作是需要投入非常多的心血，也體會到團隊合作是多麼多麼的重要，到最後看到成品能順利、完美的啟動、使用我們這幾個月來的辛苦一瞬間都值得了。

捌、參考資料及其他

1. 艾迪諾(2015)。Arduino 全能微處理機實習。台北：全華。
2. 徐德發(2012)。Arduino 錦囊妙計。台北：碁峰資訊。
3. 趙英傑(2013)。超圖解 Arduino 互動設計入門。台北：旗標。
4. 孫駿榮、蘇海永(2015)。用 Arduino 全面打造物聯網。台北：碁峰資訊。

【評語】 052321

本作品為藍芽電錶之使用添加新功能，透過藍芽傳輸資料到自製的 Google 眼鏡上，再透過 OLED 顯示器和鏡子、凸透鏡呈像於一塊透明面板上，透過眼睛的視線即可直接讀取到由藍芽電錶傳輸過來的數據，讓使用者可以更加方便又安全的使用三用電錶，具有創意性。

Google眼鏡電錶

摘要

本作品的特色是利用具有藍芽傳輸功能的三用電錶，透過藍芽傳輸資料到自製的Google眼鏡上，再透過OLED顯示器和鏡子、凸透鏡呈像於一塊透明面板上，透過眼睛的視線即可直接讀取到由藍芽電錶傳輸過來的數據，讓使用者可以更加方便又安全的使用三用電錶。

研究動機

三用電錶在電路檢測上是最常用的儀錶，但是很常在量測電壓時，一邊要注意量測的點，一邊又要低頭看看錶頭上的數值，往往頭一轉動去看錶頭時，量測的測試棒又在不注意的情況下鬆脫了，結果常常量測不到正確的數值，總是要隨時注意測試棒有沒有固定好，一邊又要讀取電錶的指示值，一不小心還會觸電。

到底有沒有什麼三用電錶？可以讓視線在看著測試棒的同時，不用把頭轉來轉去就可以看到電錶的指示值呢？如果三用電錶可以像Google眼鏡一樣，將量測到的數據顯示在Google眼鏡上，一定帥呆了，就可以解決長久以來使用三用電錶的困擾，於是有了想要自己動手作一個Google眼鏡電錶的動機。

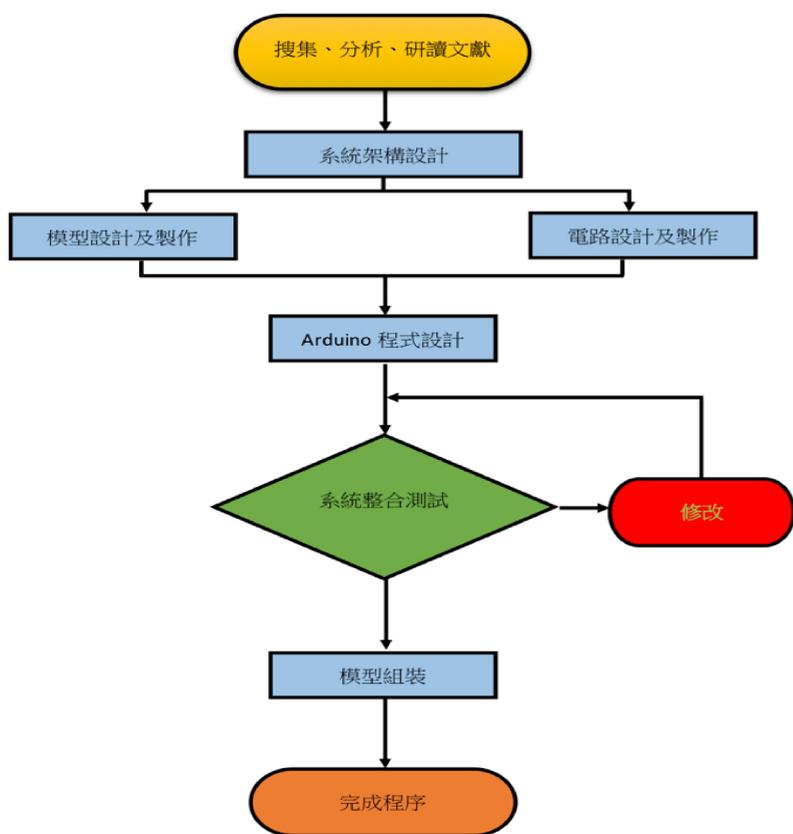
研究目的

有薦於Google眼鏡的價格太過於昂貴，於是本研究的目的為利用學校所學，以有限的資源自製學生級的Google眼鏡，具體設計特色為：

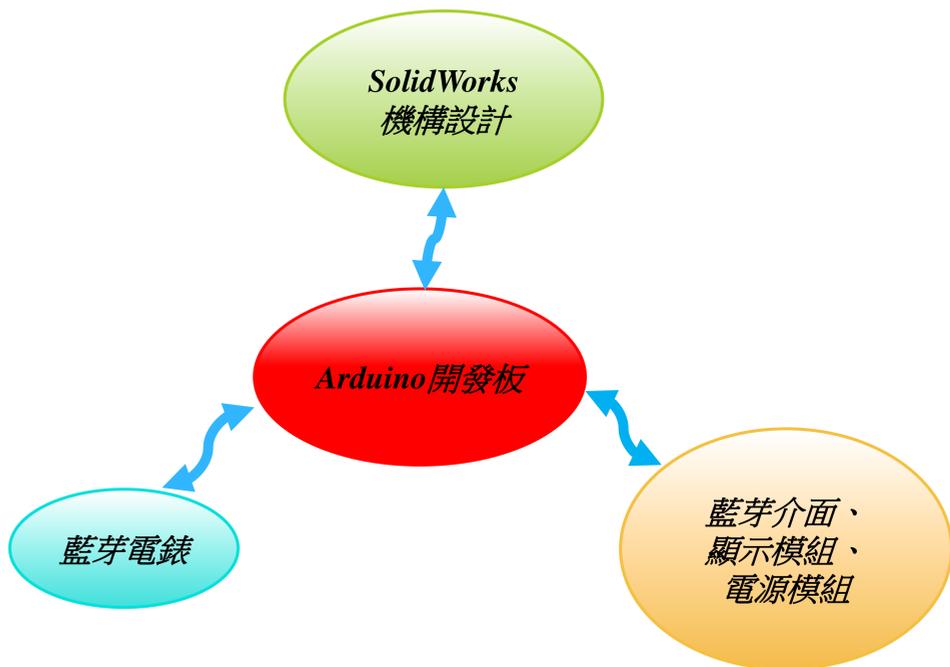
- 一、參考Google眼鏡的圖片，利用SolidWorks繪製原型圖。
- 二、利用3D列印機，列印可完成的作品框架。
- 三、利用藍芽模組與市售的藍芽電錶連結即可完成資料的讀取。
- 四、讓有興趣自製作品的人也可以參考本研究的作品順利完成製作或改良。

研究過程或方法

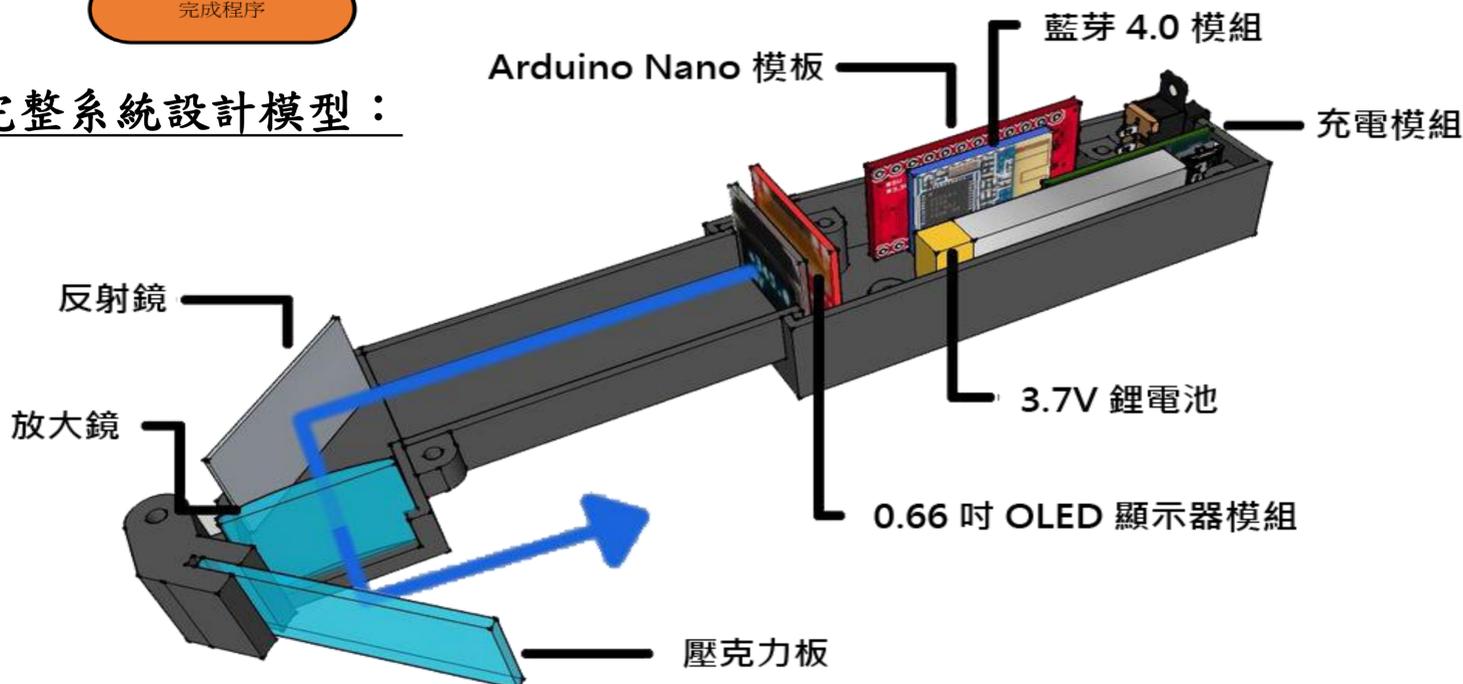
一、研究流程圖：



二、系統設計架構圖：

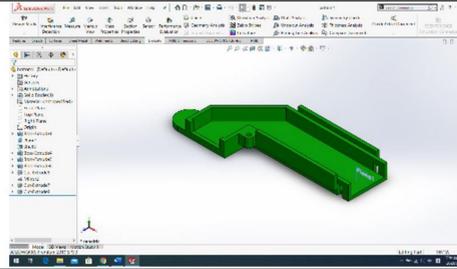


三、完整系統設計模型：

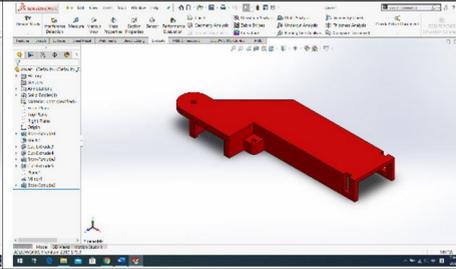


四、模型設計及3D繪圖：

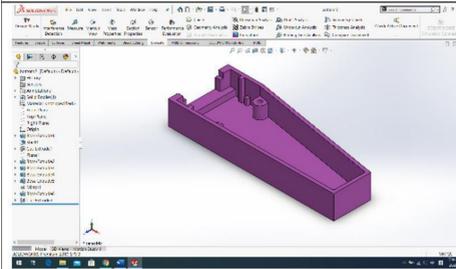
1.機構組件-1



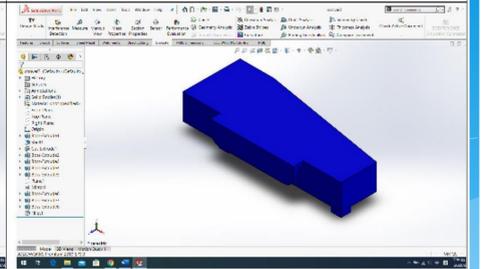
2.機構組件-2



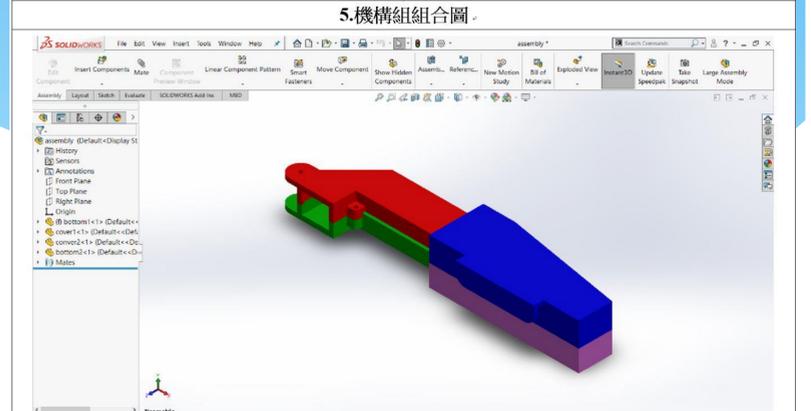
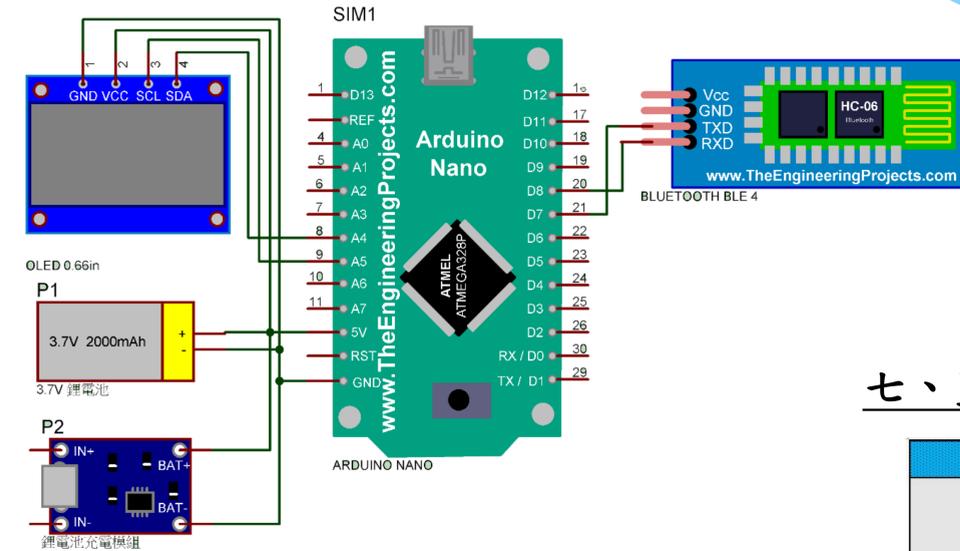
3.機構組件-3



4.機構組件-4



五、電路硬體設計圖：

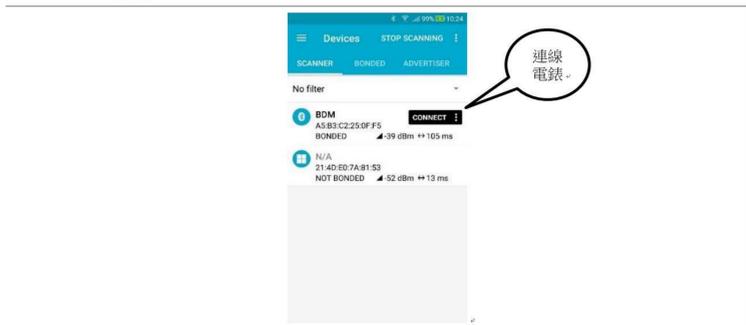


六、藍芽電錶傳送資料記錄實驗：

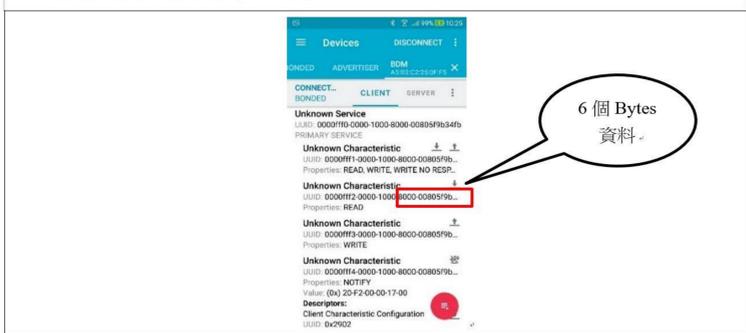
1. 開啟 OW18B 電源並啟動藍芽功能。



2. 啟動手機 nRF Connect APP，再與電錶連線。



3. 轉動電錶檔位，監看記錄資料變化。

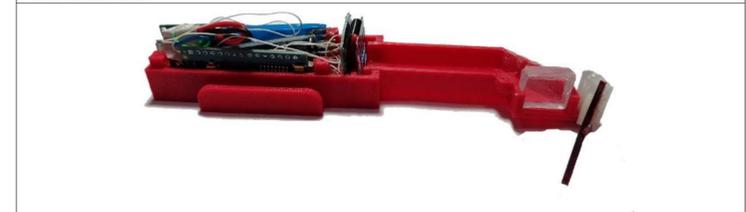


八、硬體組裝：

1.組合外殼、反射鏡、放大鏡。



2.置入硬體電路。



3.蓋上上蓋完成組裝。



七、藍芽電錶記錄資料表：

		檔位	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
AC 電壓	Manual	mV	59	F0	手動0 自動4	0	Low Byte	High Byte
		無小數點	60	F0		0		
		小數點1位	61	F0		0		
		小數點2位	62	F0		0		
DC 電壓	Manual	mV	19	F0		0	Low Byte	High Byte
	V	無小數點	20	F0		0		負值 8X
	V	小數點1位	21	F0		0		負值 8X
	V	小數點2位	22	F0		0		負值 8X
電阻	手動 Ω	小數點1位	21	F1		0	Low Byte	High Byte
		小數點3位	2B	F1		0		
	手動 KΩ	小數點2位	2A	F1		0		
		小數點1位	29	F1		0		
	手動 MΩ	小數點3位	33	F1		0		
		小數點2位	32	F1		0		
二極體	導通電壓	小數點3位	A3	F2		0	0	
	未導通	OL	A7	F2		0	0	FF 7F
	是否短路	短路		E1	F2		0	0
	未短路		E7	F2		0	0	FF 7F
電容	nF	小數點2位	4A	F1		4	0	
	Hz	小數點1位	A1	F1		4	0	
頻率		小數點2位	A2	F1		4	0	
		小數點3位	A3	F1		4	0	
溫度	攝氏		20	F2		0	0	
	華氏		60	F2		0	0	
	無接觸電壓偵測		60	F3		0	0	01 有電
	hfe		20	F3				
DC 電流	uA	小數點1位	91	F0		0	0	Low Byte High Byte
		無小數點	90	F0		0	0	負值 8X
	mA	小數點1位	99	F0		0	0	負值 8X
		小數點2位	9A	F0		0	0	負值 8X
AC 電流	A	小數點2位	A2	F0		0	0	負值 8X
	uA	無小數點	D0	F0		0	0	Low Byte High Byte
		小數點1位	D1	F0		0	0	
	mA	小數點1位	D9	F0		0	0	
	小數點2位	DA	F0		0	0		
	A	小數點2位	E2	F0		0	0	

由實驗數據中，分析出各位元組所代表數值的說明如下：

- (1) Byte1、Byte2：表示為檔位與所測量的解析度單位。
- (2) Byte3：為手動(0)或自動(4)。
- (3) Byte4 無作用。
- (4) Byte5、Byte6 為測量的高低位元數值。

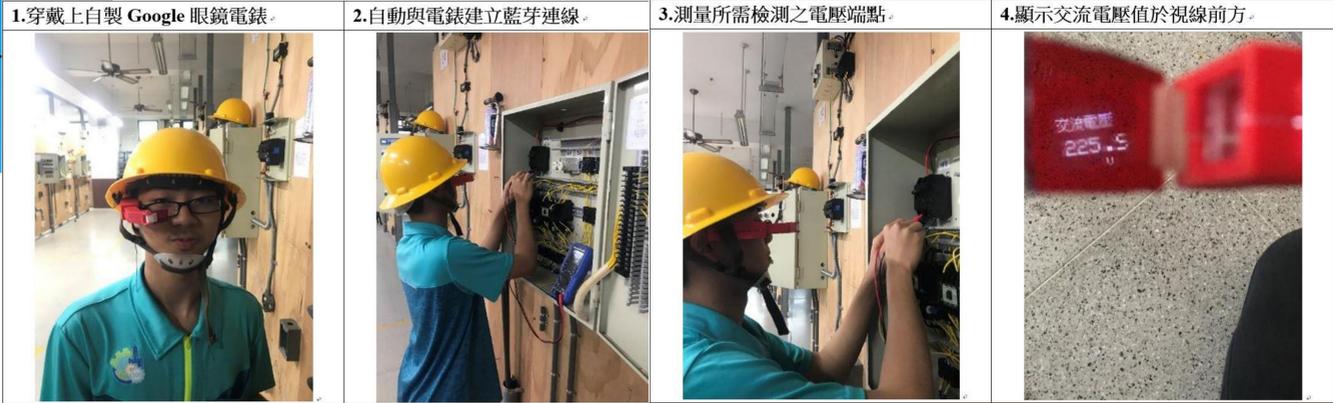
例如：測量交流 220V 電壓值，電壓錶設定：檔位(ACV)、無小數點、手動。

手機接收到的藍芽傳送訊息： UUID：0000fff2-0000-1000-00dc-0000f060。

Byte6	Byte5	Byte4	Byte3	Byte2	Byte1
00	DC	00	00	F0	60

研究結果

一、作品測試結果：



二、作品特點：

1. 本研究自製的Google眼鏡電錶，讓使用者在使用電錶時，可以更方便的操作和讀取數值，避免掉人為操作上的危險，特別是在高電壓及狹窄工作空間時，除方便測量，更可以保障使用者的安全。
2. Google眼鏡電錶在外型上雖無法與市售的商品相比，但自製時所花費的費用卻不到市售商品的十分之一，對於沒有經費預算的人來說，不失為一種極佳的選擇。
3. Google眼鏡電錶在設計開發的過程中，是利用SolidWorks軟體設計，再實際由3D列印機印出，成品若不理想也可以即時透過軟體修改，除加速了開發的速度，也省下了很多費用。

三、自製作品費用：

名稱	數量	價格	合計
Arduino NANO	1 個	100 元	100 元
0.66 吋 OLED	1 個	120 元	120 元
藍芽 4.0BLE 模組	1 個	100 元	100 元
鋰電池充電模組	1 個	50 元	50 元
3.7V 鋰電池	1 個	150 元	150 元
3D 列印耗材、鏡片等	1 套	約 500 元	500 元
總費用合計約：			1020 元

討論

一、Google眼鏡電錶重量與大小問題：

本研究自製Google眼鏡電錶在穿戴上，是掛載在使用者本身所戴的眼鏡側邊，如圖1所示，在穿戴上會讓使用者感覺重量太重尺寸太大的問題。由於本研究設計主要考量為利用有限資源為出發點來創作，故在電路硬體的選擇上已盡量選擇較小尺寸，在機構的材質上亦是利用學校所提供3D列印機的PLA(Polylactic Acid聚乳酸)線材列印而成，故在後續的改良上建議可將電路整合縮小尺寸，如此在機構的設計才可進一步的減少所需放置硬體電路的空間，而機構的材質上，可朝向利用輕量金屬的材料設計外殼，如：鋁、鈦...等。

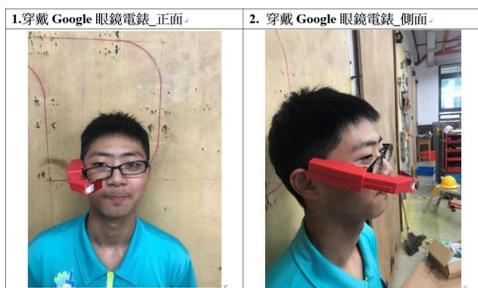


圖1.



圖2.



圖3.



二、Google眼鏡焦距問題：

由於自製的Google眼鏡電錶是利用反射鏡、放大鏡將OLED顯示的數值投射到透明壓克力板上，讓使用者的視線可以同時看到電錶數值及視線前方景物，但在掛載在使用者本身所戴的眼鏡側邊時，投射出來的數值，因每位使用者眼鏡不同及自製Google眼鏡機構問題會有不容易對焦的問題如圖2所示，目前的改良方式為在Google眼鏡外殼上，利用線槽製作可調整掛載間隙的Google眼鏡外殼，方便使用者調整焦距，如圖3所示。另一種改良方式為利用雙面膠將Google眼鏡黏貼在安全帽的側邊，除了方便調整焦距，同時也解決Google眼鏡掛載在眼鏡上造成使用者覺得重量太重問題，如圖4所示。



圖4.



結論

本研究確實能達成研究的目的，自行設計出所需的硬體與機構，創作出類似Google眼鏡的測量輔助工具。使用者只需具備有藍芽功能的電錶，就可以透過藍芽傳輸資料到自製的Google眼鏡上，再透過OLED顯示器，利用反射原理，將畫面呈像於一塊透明面板上，透過眼睛的視線即可輕易讀取到由藍芽電錶傳輸過來的數據，讓使用者可以更加方便又安全的使用三用電錶。

參考資料

1. 艾迪諾(2015)。Arduino全能微處理機實習。台北：全華。
2. 徐德發(2012)。Arduino錦囊妙計。台北：基峰資訊。
3. 趙英傑(2013)。超圖解Arduino互動設計入門。台北：旗標。
4. 孫駿榮、蘇海永(2015)。用Arduino全面打造物聯網。台北：基峰資訊。