

臺北市立大安高級工業職業學校

電子科

多功能心跳血氧機

Multifunction Heartbeat Oximeter

指導老師:

楊仁元老師

組長:

電子三甲 余駿凱(0804102)

組員:

電子三甲 曾熙倫(0804127)

電子三甲 楊宇詮(0804132)

電子三甲 楊適齊(0804133)

摘要

在科技飛速發展的時代，身心健康監測與科技相結合勢在必行。

心跳和血氧是人體最重要的部分。未來，每個人都可以更輕鬆地檢查自己和家人的健康狀況，並與以往的記錄進行比較，輕鬆掌握健康狀況。

結合 LINE Notify 可以增加被關注的概率。建立網站，可以更方便地查詢監測值，讓人們省心省力，簡單方便地獲取信息，從容面對自己和家人的健康。

關鍵字: 多功能、心跳、血氧機、LINE Notify

Abstract

In an era of rapid technological advancement, it is imperative to combine physical and mental health monitoring with technology.

Heartbeat and blood oxygen are the most important parts of the human body. In the future, everyone will be able to check the health of themselves and their family members more easily, and compare with previous records to easily grasp the health status.

Combining with LINE Notify can increase the probability of being noticed. Setting up a website makes it easier to query monitoring values, allowing people to spend less effort to obtain information simply and conveniently, and to face the health of themselves and their families in a relaxed manner.

Keywords: multifunction 、 Heartbeat 、 blood oxygen 、 Line Notify

目錄

摘要	I
ABSTRACT	II
目錄	III
表目錄	V
圖目錄	VI
第 1 章 前言	1
1-1 背景與目的	1
1-2 預期成果	1
1-3 組員貢獻度	2
第 2 章 理論探討	3
2-1 硬體	3
2-1-1 ESP32-TTGO T-Display	3
2-1-2 OLED	4
2-1-3 MAX30102	4
2-1-4 微動開關	5
2-2 軟體	6
2-3 電路製作	6
2-3-1 上層	14
2-3-2 下層	15
2-4 外殼	16
2-4-1 上層蓋子	16
2-4-2 上層內板	17
2-4-3 下層內板	19
2-4-4 下層底蓋	21
第 3 章 專題準備	22
3-1 系統架構	22
3-2 甘特圖	23
3-3 流程圖	24
第 4 章 專題成果	25

4-1	成果.....	25
4-1-2	LINE Notify.....	29
4-2	問題與解決.....	33
4-2-1	卡榫斷裂.....	33
4-2-2	接腳不支援低態動作.....	33
第 5 章	結論與建議.....	34
5-1	結論.....	34
5-2	建議.....	34
	參考文獻.....	35
	附錄.....	36
	材料清單.....	36
	設備清單.....	37
	成員簡歷.....	38

表目錄

表 1 甘特圖	23
表 2 功能比對	25
表 3 心跳量測精度比較.....	26
表 4 血氧量測精度	27
表 5 量測精度(血氧機).....	28
表 6 材料清單	36
表 7 設備清單	37
表 8 成員簡歷-余駿凱	38
表 9 成員簡歷-曾熙倫	39
表 10 成員簡歷-楊宇詮.....	40
表 11 成員簡歷-楊適齊.....	41

圖目錄

圖 1 成員貢獻度.....	2
圖 2 ESP32-TTGO.....	3
圖 3 OLED	4
圖 4 MAX30102	4
圖 5 D2F-01F.....	5
圖 6 上層電路圖	14
圖 7 上層佈線圖	14
圖 8 下層電路圖.....	15
圖 9 下層佈線圖	15
圖 10 上層蓋子(繪圖軟體).....	16
圖 11 上層蓋子(實際).....	16
圖 12 上層內板(繪圖軟體)底部.....	17
圖 13 上層內板(繪圖軟體)上方.....	17
圖 14 上層內板(實際)底部	18
圖 15 上層內板(實際)上方	18
圖 16 下層內板(繪圖軟體)底部.....	19
圖 17 下層內板(繪圖軟體)底部.....	19
圖 18 上層內板(實際)底部	20
圖 19 下層內板(實際)上方.....	20
圖 20 下層底蓋(繪圖軟體).....	21
圖 21 下層底蓋(實際).....	21
圖 22 系統架構.....	22
圖 23 流程圖	24
圖 24 心跳量測(穿戴式裝置).....	26
圖 25 血氧量測(穿戴式裝置).....	27
圖 26 心跳血氧量測(血氧機).....	28
圖 27 LINE NOTIFY	29

第1章 前言

1-1 背景與目的

2021 年疫情肆虐全球，人們開始注重自身的健康問題，由於全球出現多起隱形缺氧案例，心跳血氧機成了重點關注的產品，我們以市面上的血氧機為基礎並製作。

多功能心跳血氧機(Multifunction Heartbeat Oximeter，以下簡稱本專題)主要目的為檢測自身的各項數值，如:心跳、血氧，並製成圖表。

本專題也可從網站查看，亦可以多個人使用同一台血氧機並且擁有各自的紀錄，也結合 Line Notify 當數值異常時可做到警示的效果。

希望可以藉由本專題學習到不同的知識與團隊合作的精神。

1-2 預期成果

本專題預期能使用 MAX30102 感測心跳血氧，並用 OLED 顯示。有別於一般的血氧機，我們結合了 ESP32 的 EPROM 記下過去的紀錄，並利用程式轉換為圖表。

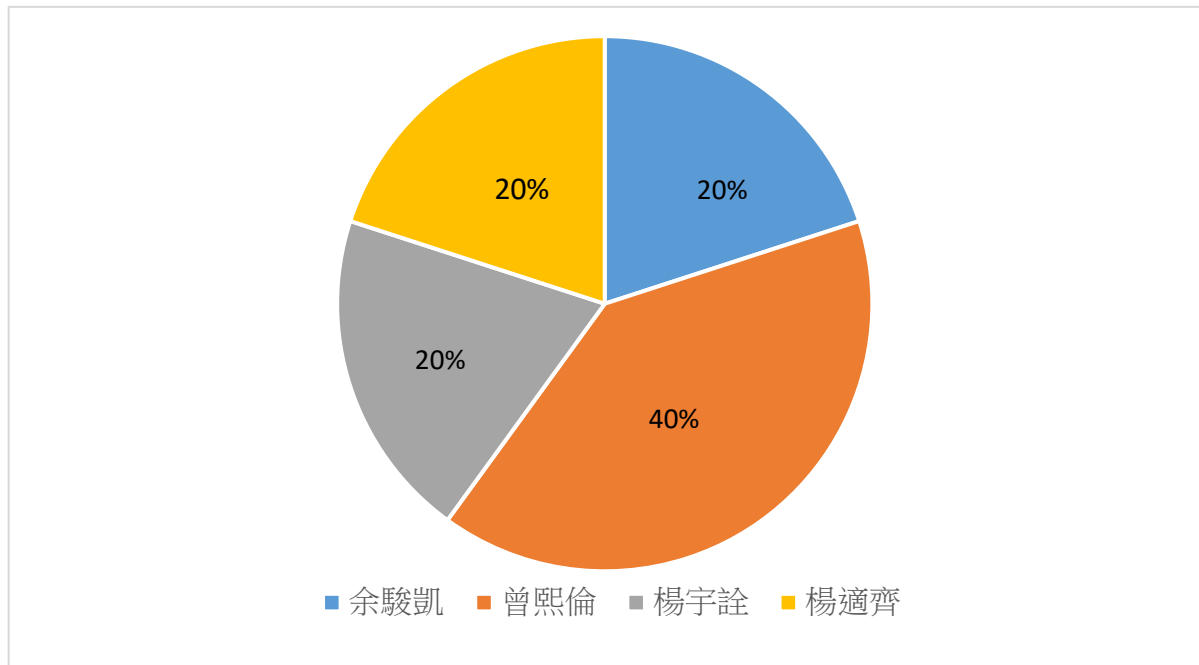
利用 HTML 製作網站架構並顯示 QR CODE 讓使用者可以快速連上網站，也使用 Line Notify 讓使用者的家人迅速得知使用者的身體狀況。

最後用 3D 列印外殼將全部結合起來。

1-3組員貢獻度

本專題以程式為核心軟體撰寫的曾熙倫同學 40%貢獻度其餘成員各 20%。
貢獻圖表如下

圖 1 成員貢獻度



第2章 理論探討

2-1 硬體

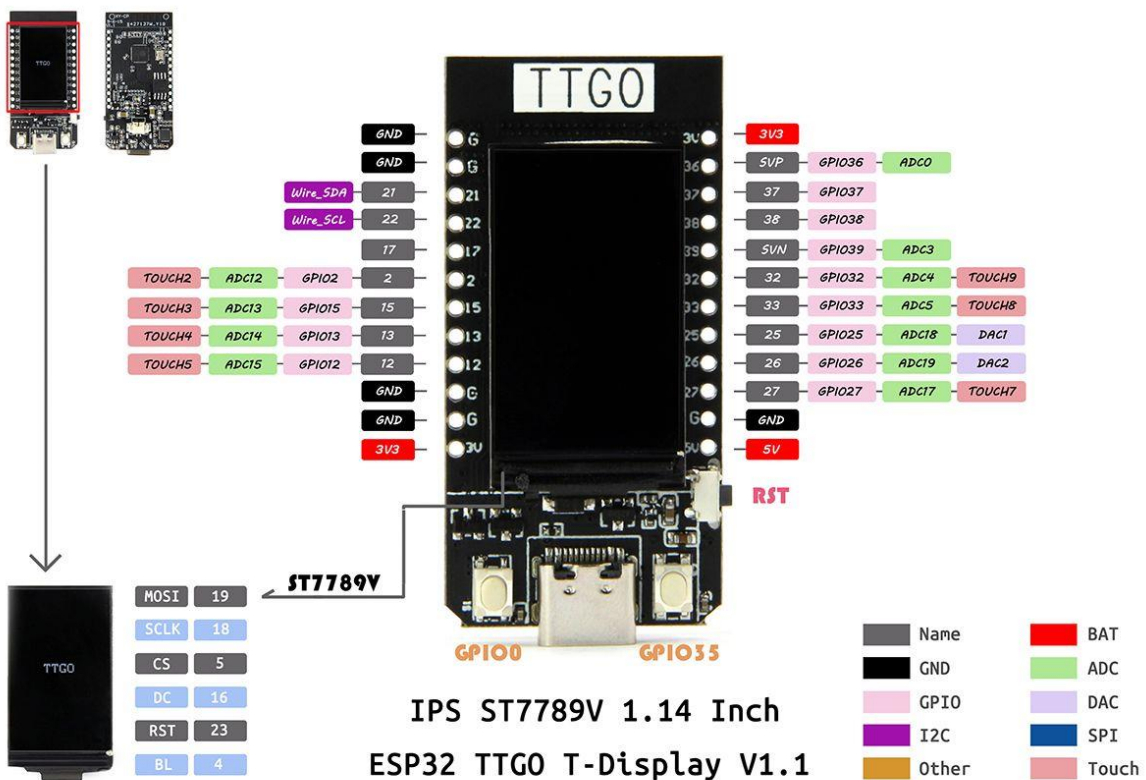
2-1-1 ESP32-TTGO T-Display

ESP32-TTGO 為一微控制器結合 1.14 吋 ST7789V、240x135 的 TFT LCD，體積小。以 Arduino IDE 函式庫進行撰寫，降低使用門檻，底部配有莫式座電池插槽提升了可攜性。

ESP32-TTGO 配有 16 支 GPIO、可支援 I²C 協定，2 支 3.3 伏特及 1 支 5 伏特與 5 支 GND 接腳，接腳詳情見下圖 2，內建 WIFI 及藍芽模組。

圖 2 ESP32-TTGO

圖片來源: <https://blog.jmaker.com.tw/ttgo-t-display/>



2-1-2 OLED

0.96 吋 OLED 螢幕，4 支 I^2C 接腳讓使用者可以簡單方便的連接到控制板，解析度 128*64 可清楚顯示各種圖形，支援多種晶片控制如:Arduino、STM32、ESP32 等，功耗低，螢幕全亮時僅 0.08W，體積小。

由於顯示螢幕將放置於手指上方，重量越輕越好，OLED 為本專題的不二首選。外觀見下圖 3

圖 3 OLED

圖片來源: <https://www.elecrow.com/096-oled-128x64-blue-p-751.html>

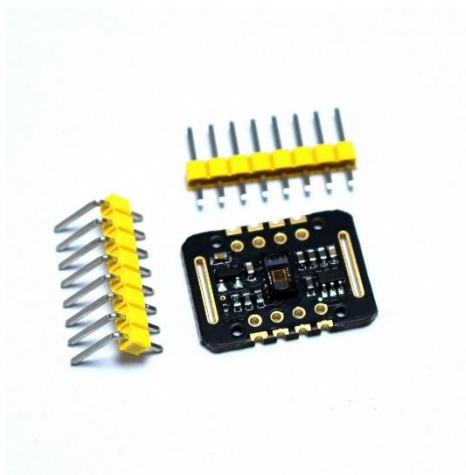


2-1-3 MAX30102

MAX30102 帶有脈搏血氧儀與心律傳感器，會發射紅光及紅外光通過光電探測器測量脈衝血液的吸光度，通過 I^2C 接口與目標 MCU 連接。其體積小且可透過指尖量測為選擇的主要原因。外觀見下圖 4

圖 4 MAX30102

圖片來源: <https://www.taiwaniot.com.tw/wp-content/uploads/2020/08/MAX30102.jpg>



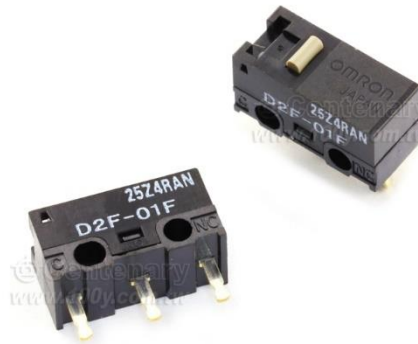
2-1-4 微動開關

採用微動開關而非傳統按鈕主要原因是想讓使用者能更輕鬆按下控制按鈕，因此我們選用 D2F-01F，由日本 OMRON 公司製造。

手感較為柔軟。三支接腳，分別為共用、常開即常關，將常開連至 5 伏特並串連一顆提升電阻，共用腳接至 MCU 作為輸入。外觀見下圖 5

圖 5 D2F-01F

圖片來源: http://images.100y.com.tw/images/product_jpg_original/A040297.jpg



2-2 軟體

2-2-1 量測心跳

檢測心跳的程式使用了 Sparkfun MAX30105 函式庫來撰寫，透過讀取紅外光的信號強度並計算心跳差來推算出平均心跳(BPM)，並透過刪去最大值以及最小值來使資料更佳準確，並且當心跳低於 50 或高於 100 時，反白畫面警示使用者。詳細心跳量測程式如下圖 6 警示程式如下圖 7

圖 6

```
void loop() {
  long irValue = particleSensor.getIR(); //讀取紅外光數值
  if (irValue > FINGER_ON ) {           //數值大於特定數值 即感測到手指
    OLED_Display_Icon(1);                //小心跳圖示
    if (checkForBeat(irValue) == true) { //檢測心跳
      OLED_Display_Icon(2);              //大心跳圖示
      tone(Tonepin,1000,10);             //發出1KHz 持續 10ms
      long delta = millis() - lastBeat; //計算心跳差
      lastBeat = millis();
      beatsPerMinute = 60 / (delta / 1000.0); //計算平均心跳
      Serial.print("BPM= "); Serial.println(beatsPerMinute); //將當次心跳顯示到序列
      if (beatsPerMinute < 255 && beatsPerMinute > 20) { //當心跳在20-255之間
        rates[ratesSpot++] = (byte)beatsPerMinute; //儲存心跳數值陣列
        ratesSpot %= RATE_SIZE; //rateSpot= 0~RATE_SIZE
        beatAvg = 0; //平均值歸零
        int min,max; //最小值與最大值
        for (byte x = 0 ; x < RATE_SIZE ; x++) { //找出最小值與最大值
          min=0;
          max=0;
          if(rates[min]>rates[x]) min=x;
          if(rates[max]<rates[x]) max=x;
        }
        for (byte x = 0 ; x < RATE_SIZE ; x++)
          if(x!=max and x!=min) beatAvg += rates[x]; //剔除最小值與最大值
      }
    }
  }
}
```

圖 7

```
if (beatAvg > 20) OLED_Display_Text(2,2,WHITE,42,15,String(beatAvg) + " BPM",42,44,String(Esp02) + "%",0,0,1);
else if(beatAvg>20 and beatAvg<50 or beatAvg>100)display.setTextColor(INVERSE); //反白畫面
else OLED_Display_Text(2,2,WHITE,42,15,"-- BPM",42,44,"---- %",0,0,1);
```

2-2-2 量測血氧

血氧部分則是透過讀取紅光及紅外光的信號強度來判別血氧濃度，當兩道光穿透手指時，血液中的帶氧血紅素跟去氧血紅素，兩者對紅外光的吸收度皆不大，但對紅光就差很多，去氧血紅素容易吸收紅光，並根據紅外光以及紅光的吸收落差計算出血氧濃度。詳細血氧量測程式如下圖 8

圖 8

```
//計算血氧
uint32_t ir, red ;
double fred, fir;
particleSensor.check(); //Check the sensor, read up to 3 samples
if (particleSensor.available() ) {
  i++;
  red = particleSensor.getFIFOIR(); //讀取紅光
  ir = particleSensor.getFIFORed(); //讀取紅外線
  fred = (double)red; //轉double
  fir = (double)ir; //轉double
  avered = avered * frate + (double)red * (1.0 - frate); //average red level by low pass filter
  aveir = aveir * frate + (double)ir * (1.0 - frate); //average IR level by low pass filter
  sumredrms += (fred - avered) * (fred - avered); //square sum of alternate component of red level
  sumirrms += (fir - aveir) * (fir - aveir); //square sum of alternate component of IR level
  if ((i % Num) == 0) {
    double R = (sqrt(sumredrms) / avered) / (sqrt(sumirrms) / aveir);
    SpO2 = -23.3 * (R - 0.4) + 100;
    ESpO2 = FSpO2 * ESpO2 + (1.0 - FSpO2) * SpO2; //low pass filter
    if (ESpO2 <= MINIMUM_SPO2) ESpO2 = MINIMUM_SPO2; //indicator for finger detached
    if (ESpO2 > 100) ESpO2 = 99.9;
    Serial.print("Oxygen % = "); Serial.println(ESpO2); //將血氧顯示到序列
    sumredrms = 0.0; sumirrms = 0.0; SpO2 = 0;
    i = 0;
  }
  particleSensor.nextSample(); //We're finished with this sample so move to next sample
}
```

2-2-3 儲存資料

資料部分，由於 esp32 內部的 EEPROM 大小有限，能儲存的資料也就相對地有限，這邊透過使用使用者代碼來減少使用的儲存空間，共儲存 20 筆資料。詳細程式如下圖 9

圖 9

```
String savedata;
if(Wifi_Connected==1) {           //如果有連上網路
  printLocalTime();              //取得時間
  Plus_Zero(savedata,month);      //偵測數值是否<10
  Plus_Zero(savedata,day);        //是的話前方補0並存入字串中
  Plus_Zero(savedata,hour);       //
  Plus_Zero(savedata,minute);    //
}
else
  savedata+="00000000";           //若無法取得時間(無網路) 則存入00000000
savedata+=user;                   //依序將使用者、心跳(BPM)、血氧濃度存入字串中
savedata+=beatAvg;
savedata+=ESpO2;
Serial.print("savedata= ");
Serial.println(savedata);
WriteData(172+16*OFFL.number,172+16*OFFL.number+16,savedata,1); //從位置172開始 每16bit儲存一筆資料
EEPROM.write(510,OFFL.number);    //寫入當前資料儲存格位置
EEPROM.commit();                  //確認儲存
Serial.print("OFFL.number= ");
Serial.println(OFFL.number);
OFFL.number=((OFFL.number+1)%20); //資料儲存格位置+1 (0~19)
//Save Offline Data -- Total 20 sets of data
Measure_time=millis();           //量測時間歸零
Idle_Measure_Time=Measure_time;  //閒置時間歸零
```

2-2-4 Line Notify

Line Notify 部分則是簡單將數值透過 Line Notify 傳送給使用者，如果心跳過緩或過快，會給予警示警示使用者心跳異常。詳細程式如下圖 10

圖 10

```
//Notify
String message = "平均心跳為:";
message += "\n" + String(beatAvg) + " BPM";
message += "\n" + String(ESpO2) + " Percent";
if(beatAvg<50 or beatAvg>100)
    message+= "\n 注意!";
if (client.connect(host, 443)) {
    Serial.println("Client Connected !");
    Serial.println(message);
    int LEN = message.length();
    //傳遞POST表頭
    String url = "/api/notify";
    client.println("POST " + url + " HTTP/1.1");
    client.print("Host: "); client.println(host);
    //權杖
    client.print("Authorization: Bearer "); client.println(Linetoken);
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.print("Content-Length: "); client.println( String((LEN + 8)) );
    client.println();
    client.print("message="); client.println(message);
    client.println();
    client.stop(); //斷線，否則只能傳5次
}
else {
    //傳送失敗
    Serial.println("connected fail");
}
```

2-2-5 線上網站

網站部分則是透過 WebServer 函式庫和 HTML 以及 CSS 來完成。最上方顯示四位使用者名稱，功能包含顯示目前心跳以及血氧，和顯示過去的量測記錄。

詳細程式如下圖 11 圖 12 圖 13 圖 14

圖 11

```
void CreateWebServer()  
{  
    server.on("/Spo2", []() {  
        String html_1 = R"====(  
        <!DOCTYPE html>  
<html>  
  
<head>  
    <meta charset="UTF-8"/>  
<style>  
  
div {  
    display: block;  
}  
#navigation_header {  
    padding-top: 0px;  
    border-left: 8px solid #CE0000;  
    border-bottom: 5px solid #CE0000;  
    background-color: #000;  
    zoom: 1;  
}  
#navigation_header a {  
    font-family: Arial;  
    font-size: 20pt;  
    font-weight: bold;  
    color: #FFF;  
    text-decoration: none;  
}  
#navigation_header:after {  
    content: ".";  
    display: block;  
    visibility: hidden;  
    clear: both;  
    height: 0;  
    font-size: 0.1em;  
    line-height: 0;  
}  
#navigation_header ul {  
    width:100%;  
    float:left;  
    /*Removes bullets*/  
    margin:0 -40px;  
    list-style-type:none;  
}  
  
#navigation_header li {  
    float:left;  
    text-align:center;
```

圖 12

```
width:135px;
}
.center{
  text-align: center;
  border: 1px solid black;
  background-color: black;
  color: white;
  font-size: 30px;
}
.circle {
  width: 100px;
  height: 100px;
  line-height: 100px;
  border-radius: 50%;
  font-size: 40px;
  color: #fff;
  text-align: center;
  background: red
}
</style>
<meta charset="UTF-8"/>
<title>多功能心跳血氧機</title>
<body>
  <div class="center">
    <span class="dot"></span>
    <h1>
      Spo2
    </h1>
  </div>
  <div id="navigation_header">
    <ul>
      <li>
        <a class="current" href="
          )=====";
html_1 +="/";
html_1 += "Harry";
html_1 +=R"=====("
>
      <span id="MenuLinkBasic">
        )=====";
html_1 += "Harry";
html_1 +=R"=====("
</span>
</a>
</li>
<li>
      <a class="current" href="
        )=====";
html_1 +="/";
html_1 += "Harry";
html_1 +=R"=====("
>
      <span id="MenuLinkBasic">
        )=====";
html_1 += "Harry";
html_1 +=R"=====("
</span>
</a>
</li>
</ul>
</div>
</body>
</html>
```

圖 13

```
        )====="";
html_1 += "/";
html_1 += "yu14785";
html_1 += R"====="("
>
        <span id="MenuLinkBasic">
)====="";
html_1 += "yu14785";
html_1 += R"====="("
</span>
</a>
</li><li>
        <a class="current" href="
)====="";
html_1 += "/";
html_1 += "firesu";
html_1 += R"====="("
>
        <span id="MenuLinkBasic">
)====="";
html_1 += "firesu";
html_1 += R"====="("
</span>
</a>
</li><li>
        <a class="current" href="
)====="";
html_1 += "/";
html_1 += "tsuki";
html_1 += R"====="("
>
        <span id="MenuLinkBasic">
)====="";
html_1 += "tsuki";
html_1 += R"====="("
</span>
</a>
</li>
</ul>
</div>
</body>
</head>
<div style="position:relative; left:30%; top:100px;">
    <div class="circle ">)====="";
    if (ESp02 <= 90)html_1 += "--";
    else html_1 += beatAvg;
    html_1+=R"====="(</div>
</div>
```

圖 14

```

<div style="position:relative; left:60%;top:auto">
    <div class="circle ">=====";
    if (beatAvg <= 20 or beatAvg >= 220)html_1 += "--";
    else html_1 += ESpO2;
    html_1+=R"=====</div>
</div>
</div>
</div>
</div>
    )=====";
/*if (beatAvg <= 20 or beatAvg >= 220)html_1 += "--";
else html_1 += ESpO2;
html_1 += " %</ol>";*/
for(int i=0;i<20;i++){
    html_1+="<p>";
    String _savedata=" ";
    ReadData(172+16*i,172+16*i+2,_savedata);
    html_1 += _savedata;
    html_1+="/";
    ReadData(172+16*i+2,172+16*i+4,_savedata);
    html_1 += _savedata;
    html_1 += " ";
    ReadData(172+16*i+4,172+16*i+6,_savedata);
    html_1 += _savedata;
    html_1 += ":";
    ReadData(172+16*i+6,172+16*i+8,_savedata);
    html_1 += _savedata;
    html_1 += " ";
    ReadData(172+16*i+8,172+16*i+16,_savedata);
    html_1 += _savedata;
    html_1+="</p>";
    }
html_1 += "</body>";
html_1 += "</html>";
server.send(200, "text/html", html_1);
}
);
}

```

2-3 電路製作

此電路使用 Altium Designer 電路設計軟體所製作，Altium Designer 是一個容易上手的設計程式，人性化的介面能夠讓人直觀的看懂並使用各種功能，很適合新手的電路設計程式。我們的專題電路板總共分成上下兩層，上層是使用者介面，下層是負責主控板與心跳血氧的感測。

2-3-1 上層

上層板子包刮 OLED 顯示器與微動開關，使用時 OLED 會顯示心跳速率與血氧濃度等資訊，微動開關則是負責喚醒休眠的裝置與控制。

圖 15 上層電路圖

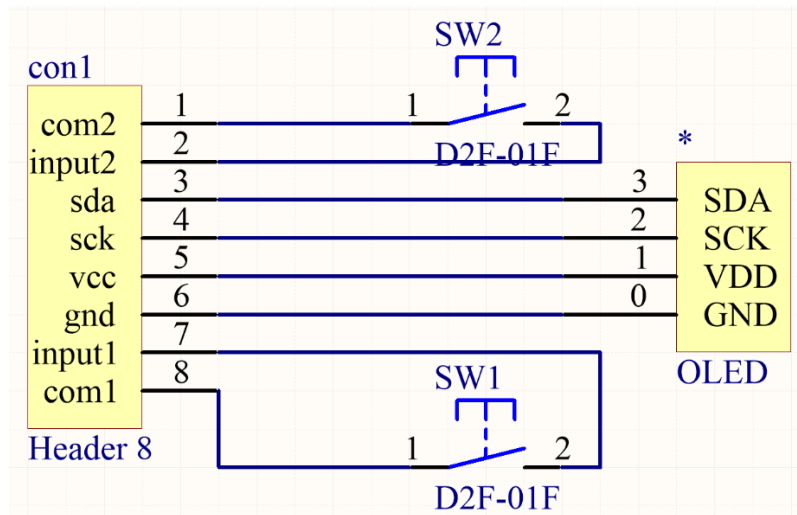
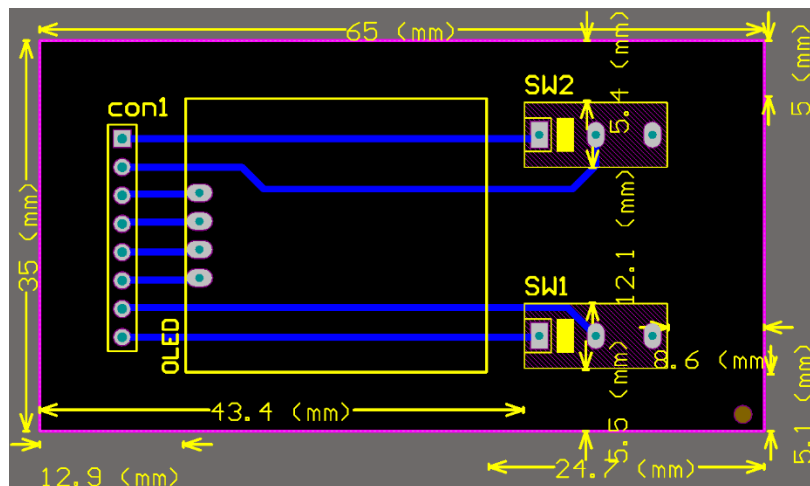


圖 16 上層佈線圖



2-3-2 下層

下層電路板包含主控板 ESP32 與感測器 MAX30102 兩個主要元件，而我們為了增加音效提醒而又外接了一顆蜂鳴器，讓使用更人性化。

圖 17 下層電路圖

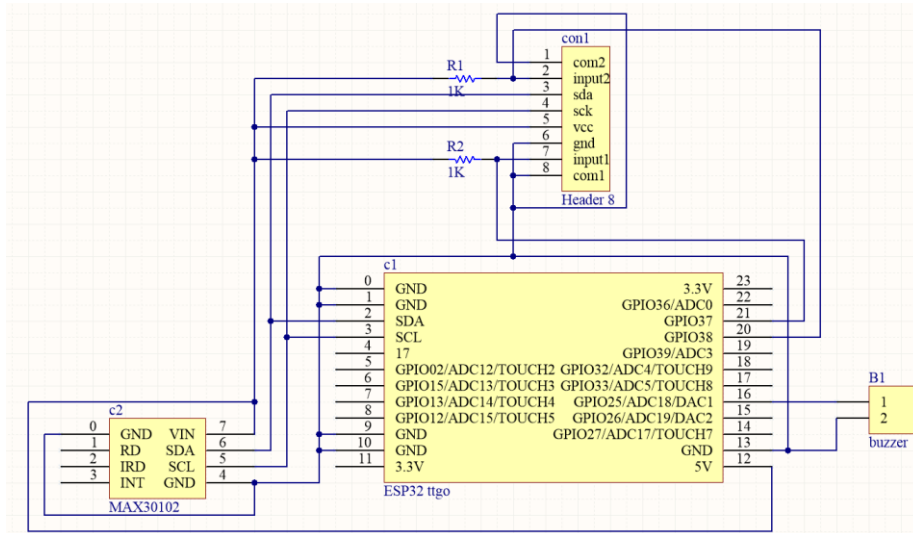
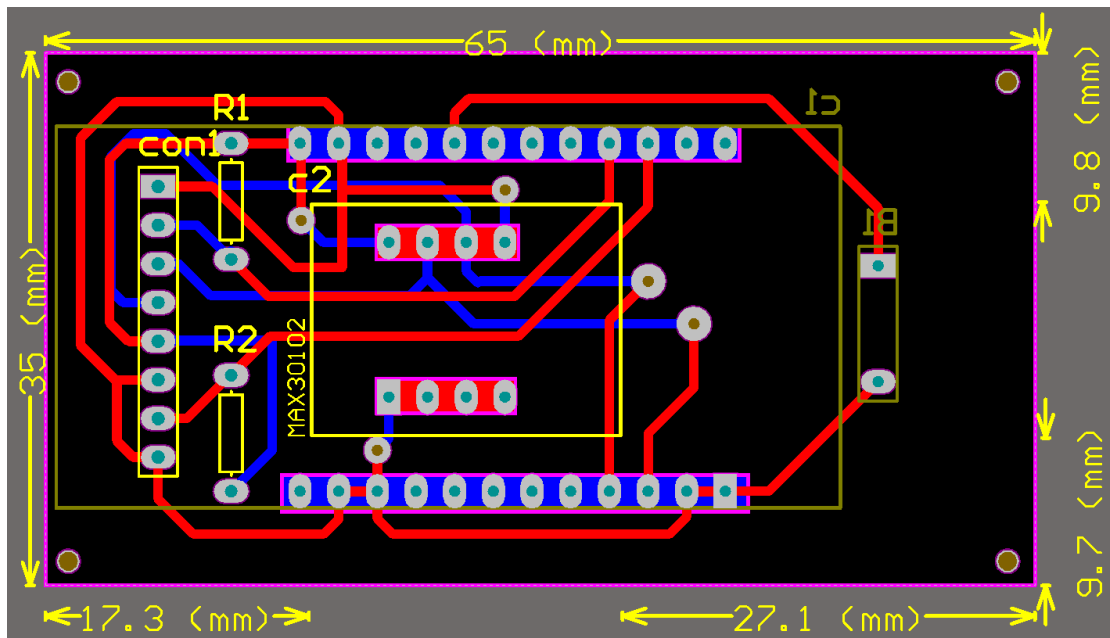


圖 18 下層佈線圖



2-4 外殼

所有部件皆使用 Blender 繪製，此軟體應用方式多元，且擁有相當多插件供使用者使用。

外殼分為 4 個部件，上下部份各有一個夾層以放置元件。

2-4-1 上層蓋子

上層可見三個空格，大的方形空格為 OLED 的顯示口，下方小型長方形空格為微動開關按鍵的卡槽。

外觀繪製軟體內外觀見下圖 19 上層蓋子(繪圖軟體)
圖 20 上層蓋子(實際)

實際外觀見下

圖 19 上層蓋子(繪圖軟體)

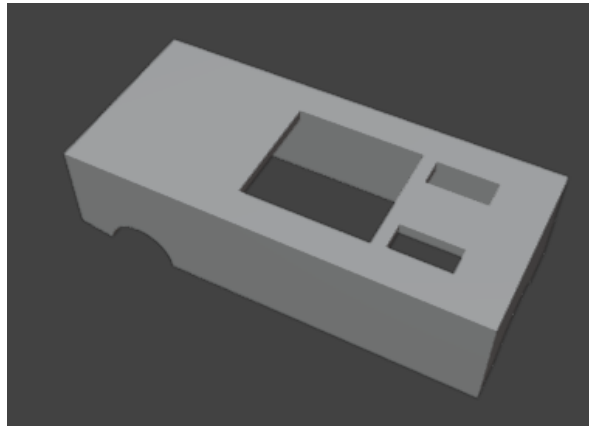


圖 20 上層蓋子(實際)



2-4-2 上層內板

前方凹槽為手指上緣放置處，而中間的空洞部分為連接上下电路板的線路，後方凹槽為彈簧放置處，但基於無法找到合適的彈簧而放棄了，改為利用重量使其自動闔上。

在繪製軟體上可見部件的上方有 4 支固定卡榫，由於其寬度太細導致斷裂，所以後期決定使用快乾膠固定，將卡榫拆除。

觀繪製軟體內外觀見下圖 21 上層內板(繪圖軟體)底部及圖 22 上層內板(繪圖軟體)上方 實際外觀見下圖 23 及圖 24 圖 20

圖 21 上層內板(繪圖軟體)底部

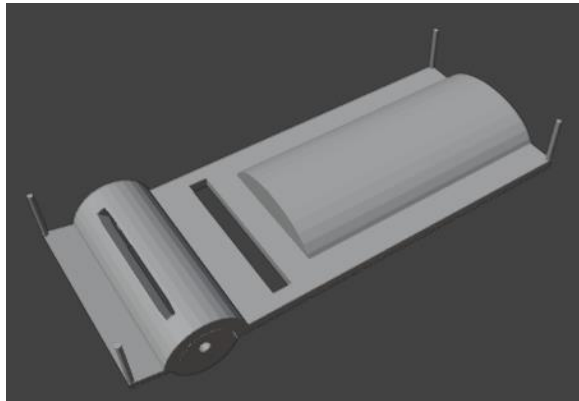


圖 22 上層內板(繪圖軟體)上方

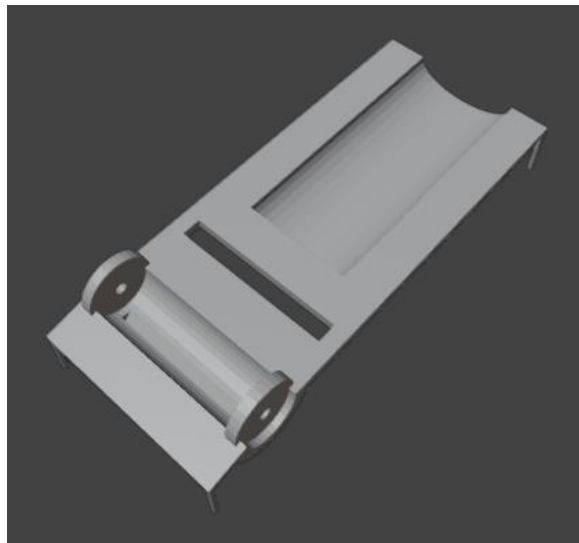


圖 23 上層內板(實際)底部



圖 24 上層內板(實際)上方



2-4-3 下層內板

下層與上層幾乎無異，前方為手指下緣放置處，方形空洞為 MAX30102 感測孔，後方凹槽同為彈簧放置處。

在板子的後方原先應有一斜板，但列印途中發生不可預期之導致斷裂，在討論之後認為並非為主要結構即不予理會。

觀繪製軟體內外觀見下圖 25 及圖 26 實際外觀見下圖 27 及圖 28 下層內板 (實際)上方圖 20

圖 25 下層內板(繪圖軟體)底部

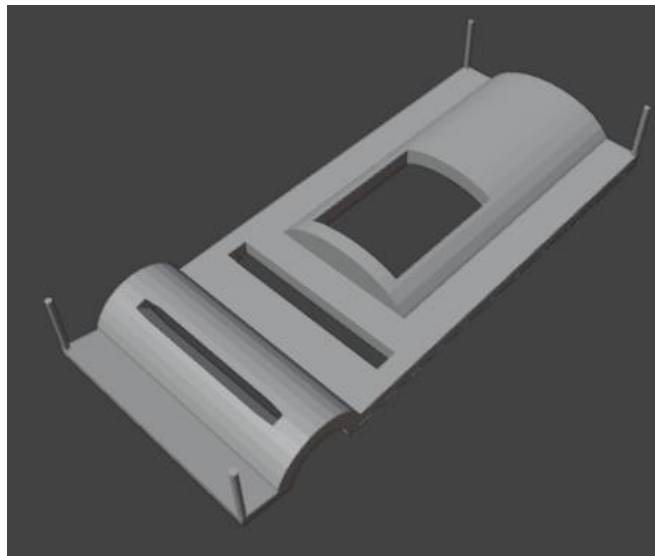


圖 26 下層內板(繪圖軟體)底部

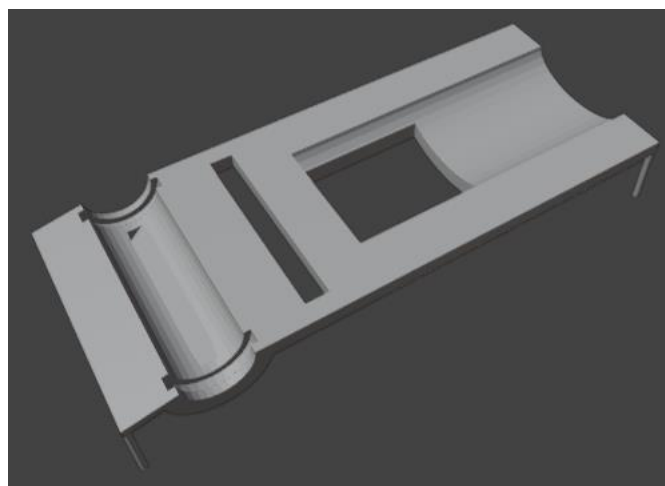


圖 27 上層內板(實際)底部

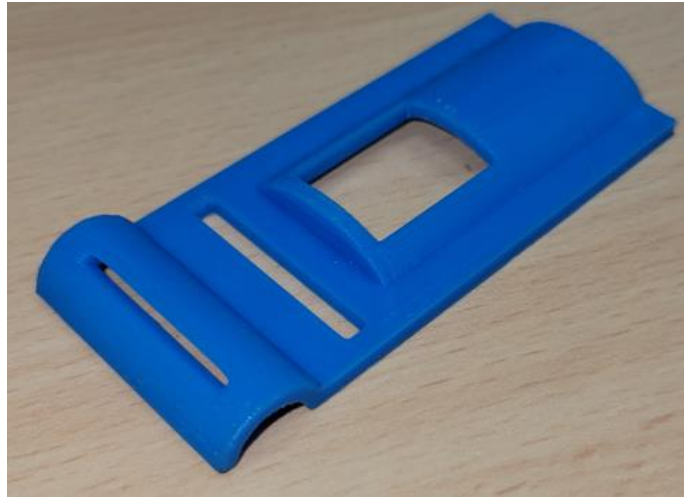


圖 28 下層內板(實際)上方



2-4-4 下層底蓋

下層高度較上層高，因為要放置電池、MCU 等等導致在不使其長度增加的情況下增加了高度，側面各有 9 個小孔放出蜂鳴器的聲音，後方有一長方形小孔為充電線輸入口。觀繪製軟體內外觀見下圖 29 實際外觀見下圖 30

圖 29 下層底蓋(繪圖軟體)

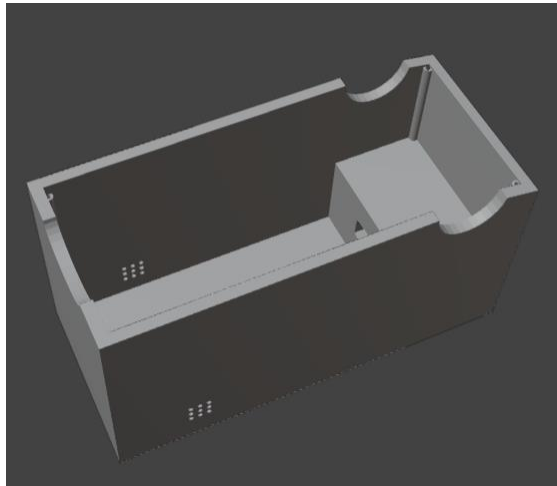


圖 30 下層底蓋(實際)



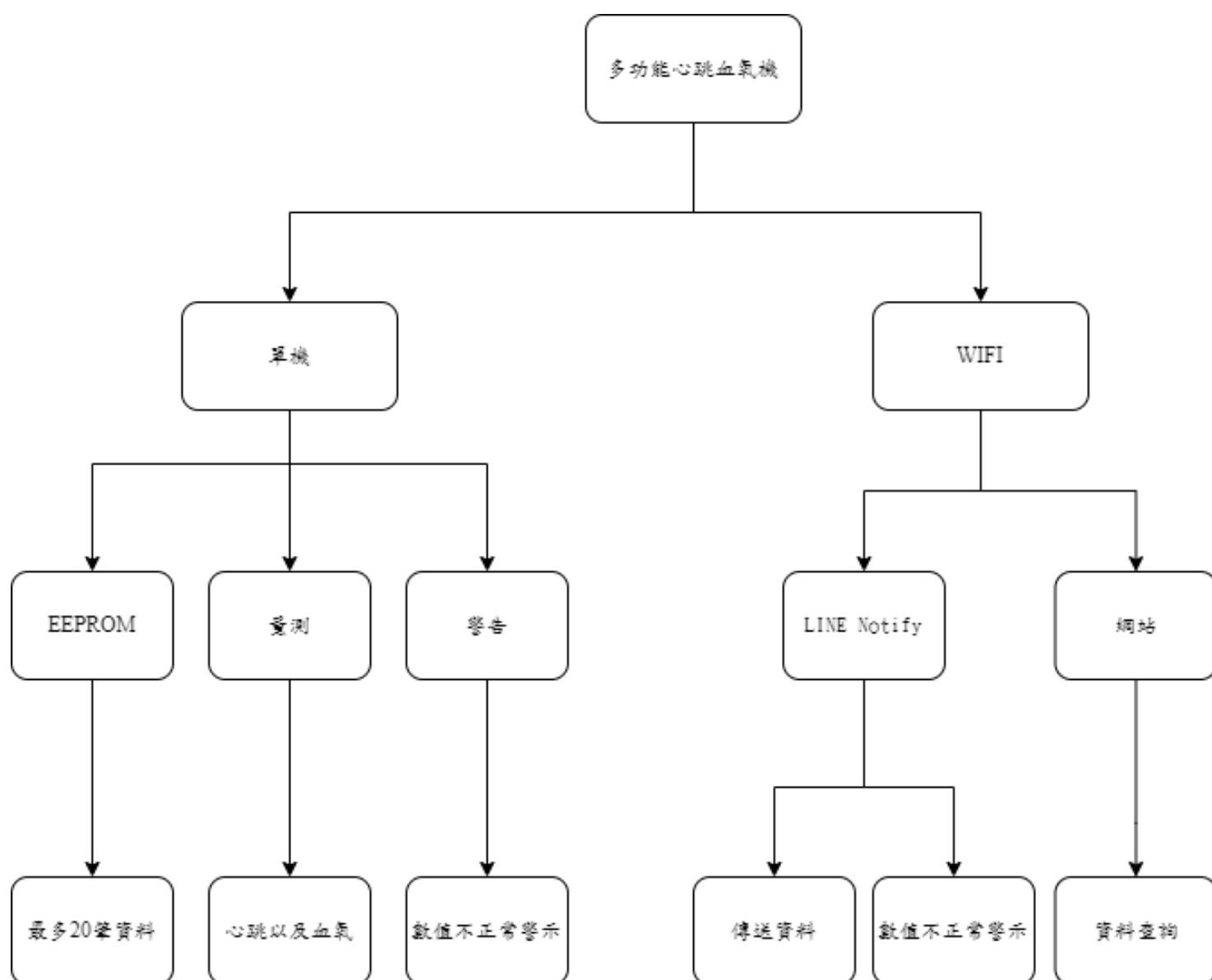
第3章 專題準備

3-1 系統架構

主要功能能分為單機與使用 WIFI，如無使用 WIFI 能做到最基本的量測心跳血氧，使用 EEPROM 紀錄與警示，如有連上 WIFI 就可使用 LINE Notify 傳送資料與警示，也可經 QRCORD 連上網站查看過去資料與修改設定。

系統架構圖見下圖 31 系統架構

圖 31 系統架構



3-2 甘特圖

負責工作並未平均分配，而是按照擅長項目即組員意願分配，使成員做力所能及且盡其所能。「負責成員」欄位中，皆為成員名字後兩字。

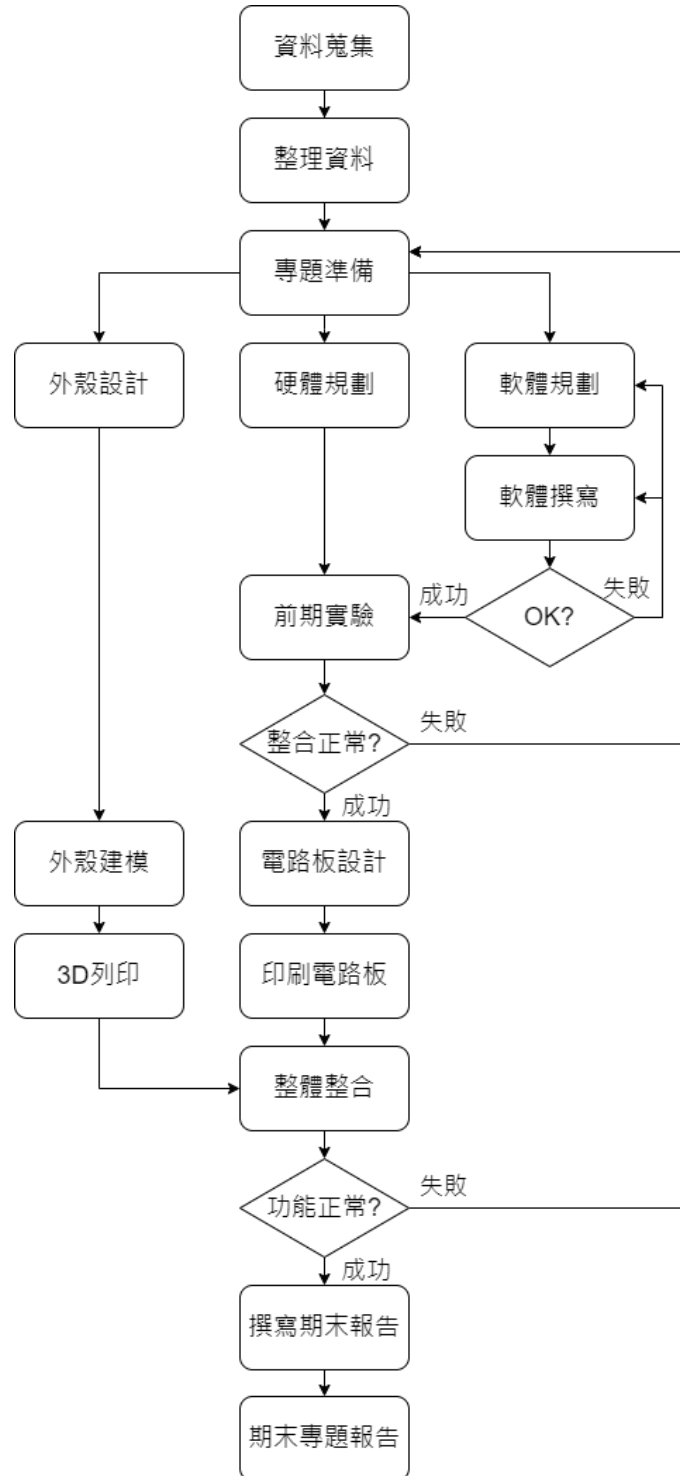
表 1 甘特圖

工作項目	週次 (日期)																		負責成員
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
資料收集																			駿凱、熙倫、 宇詮、適齊
資料整理																			駿凱、熙倫、 宇詮、適齊
專題準備																			駿凱、熙倫、 宇詮、適齊
製作計畫書																			駿凱
PPT 製作																			駿凱、熙倫、 宇詮、適齊
歷程報告																			駿凱、熙倫、 宇詮、適齊
硬體規劃																			宇詮、適齊、 駿凱
軟體規劃																			熙倫
程式撰寫																			熙倫
前期測試																			駿凱、熙倫、 宇詮、適齊
外殼設計																			駿凱、適齊
外殼建模																			駿凱、宇詮
電路板設計																			宇詮、適齊
電路板印刷																			宇詮、適齊
3D 列印																			宇詮、適齊
綜合測試																			駿凱、熙倫、 宇詮、適齊
撰寫期末報告																			駿凱、熙倫、 宇詮、適齊
預定進度	5	10	15	20	30	35	40	50	60	65	70	80	85	90	93	96	99	100	累 積 百分比%

3-3 流程圖

本專題先蒐集資料後，將軟體、硬體及外殼初步設計同時進行，再將麵包板與軟體前期實驗，完成之後繪製電路板，掌握大小及位置後繪製外殼最後整合。

圖 32 流程圖



第4章 專題成果

4-1 成果

在成果方面本專題比其他市面上能測量心跳血氧的儀器多上許多功能，但在心跳量測精準度方面仍有待校正，血氧的量測比較下來本專體可以顯示更詳細的數據。

本專題透過 WIFI 連接 LINE Notify，每 30 秒傳送一次訊息，讓使用者可以看到過去使用的紀錄，也可加入群組讓使用者的家人也可以看見。

表 2 展示本專題較其他儀器更「多功能」的部分

表 2 功能比對

功能比對	心跳	血氧	多人使用	Line Notify	網站	警示
本專題	V	V	V	V	V	V
Apple	V	V	X	X	X	X
小米	V	X	X	X	X	X
他牌血氧機	V	V	X	X	X	X

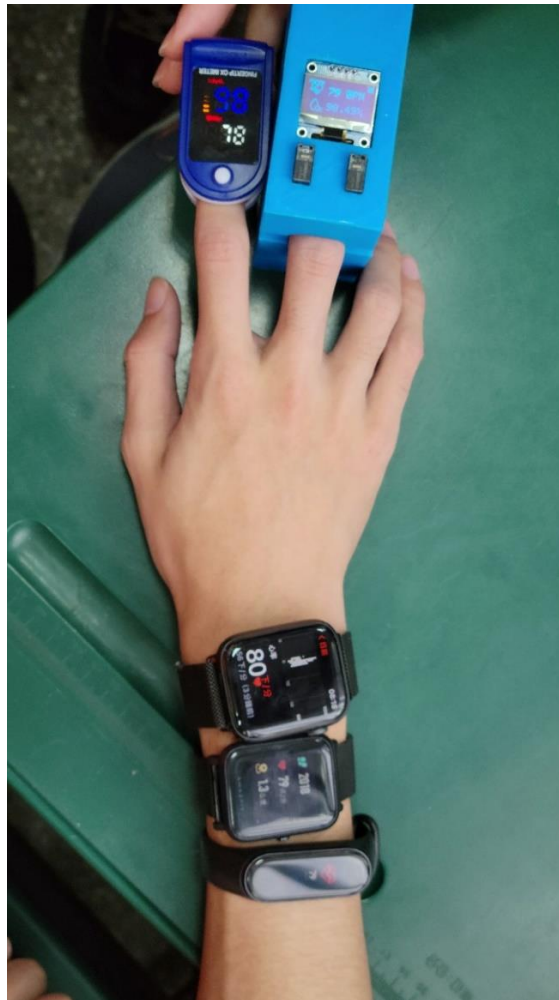
4-1-1 量測

本專題與市面上的穿戴式裝置做對比，因現行穿戴式裝置大部分僅有量測心跳的功能，因此拿心跳做比較，發現本專題時常出現量測不穩定的情況，並且有時數值偏差較大。詳細數據見下表 3 量測時照片見下圖 33

表 3 心跳量測精度比較

心跳量測精度	第一次	第二次	第三次	第四次	第五次
本專題	82BPM	78BPM	69BPM	64BPM	74BPM
apple	79BPM	79BPM	79BPM	79BPM	80BPM
小米	79BPM	79BPM	79BPM	79BPM	77BPM
他牌血氧機	79BPM	78BPM	78BPM	78BPM	79BPM
Amazfit	82BPM	79BPM	79BPM	80BPM	80BPM

圖 33 心跳量測(穿戴式裝置)



目前穿戴式裝置僅有 apple watch 擁有血氧量測功能，因此只使用 apple watch 做比較。

詳細數據見下表 4 量測時照片見下圖 34

表 4 血氧量測精度

血氧量測精度	血氧
本專題	98.62%
Apple	97%
他牌血氧機	98%

圖 34 血氧量測(穿戴式裝置)



為比照本專題與其他市面血氧機精度所以使用他牌血氧機做對照組

(上方數據為心跳，下方數據為血氧)

詳細數據見下表 5 量測時照片見下圖 35

表 5 量測精度(血氧機)

量測精度	第一次	第二次	第三次	第四次	第五次
本專題	52BPM 98.95%	50BPM 99.90%	60BPM 99.67%	64BPM 99.53%	58BPM 99.93%
他牌血氧機	64BPM 99%	67BPM 99%	67BPM 99%	68BPM 99%	67BPM 99%
ACARE(小)	65BPM 99%	68BPM 99%	68BPM 99%	68BPM 99%	68BPM 98%
ACARE(大)	65BPM 99%	69BPM 99%	67BPM 99%	68BPM 99%	68BPM 99%

圖 35 心跳血氧量測(血氧機)



+

4-1-2 LINE Notify

使用者只要將 LINE 加入 LINE Notify 即可實時收取資料
實際情形如下圖 36

圖 36 LINE Notify



4-1-3 錯誤警示

當使用者之心跳或血氧數值出現異常時，在 OLED 方面會透過反白數值而非閃爍吸引使用者注意，在 LINE Notify 則會傳送警告訊息請使用者多加注意，異常數值之範圍為:心跳每分鐘 50 以下或 100 以上、血氧則為 95%以下，為了方便拍攝成果我們將範圍縮小。成果如下圖 37 圖 38

圖 37



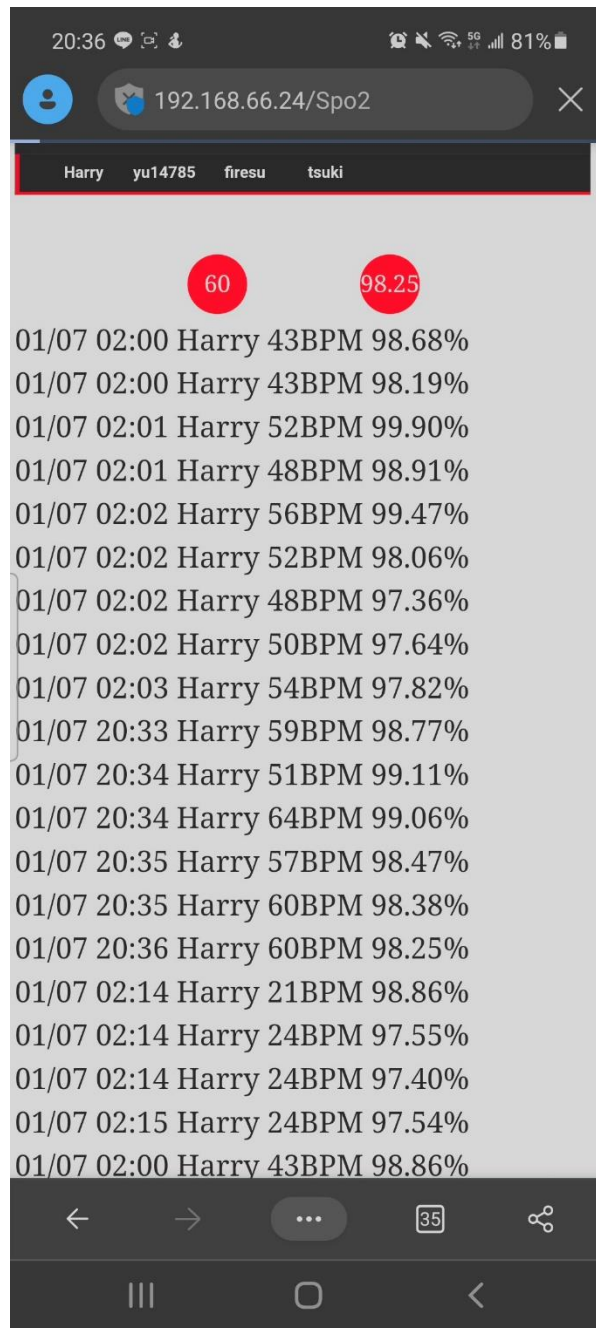
圖 38



4-1-4 網站量測顯示與資料查詢

在網站方面製作的較為陽春，最上方會顯示使用者們的名稱，上方紅圈左側代表心跳右側則為血氧，下方會顯示最多 20 比過去紀錄，經程式修改為每 30 秒傳送一次數值。成果如下圖 39

圖 39



4-2 問題與解決

4-2-1 卡榫斷裂

原先在外殼內四角設計了直徑 1mm 的卡榫以利固定，由於 1mm 實在太過細小導致列印完之後就斷裂了，最後是用快乾膠將內部外科接合固定。

4-2-2 接腳不支援低態動作

由於原本按鈕所接的接腳 2 有 touch2 的功能導致其不能為低態，所以無法用微動開關控制，後來改用 37 號接腳來做為按鈕輸入。

4-2-3 心跳量測不穩

因心跳血氧感測方式與市面上的不同與外殼無固定手指，導致微小位移就有可能造成測值不准，在修改感測器亮度使其更容易接收反射光後縮小了測值不穩定的機率。

第5章 結論與建議

5-1 結論

本專題融合了市面上血氧機與 ESP32 的控制，整體有很多進步的空間，例如外殼的細緻度、整體的體積...等等，而各種功能方面也較為陽春，整體都有待優化，做出更人性化的設計。

就整體來說比市面其他血氧機多出一些功能，但在舒適度、速度、精度都較為遜色，不過鑑於許多製作都是第一次嘗試，能做到這種程度已經算是不錯了。

5-2 建議

在最外圍應用倒角使四周更為柔順，充電的方式、位置應多加考慮，閉合的方式、量測的觸感應採用更佳人性化及更加合理的方式，在各元件凸出的部分應採用卡榫或其他方式隱藏及包裝。

本專題還可加入手機控制如 APP inventor2 等，讓使用更為便捷，可在網站加入讓使用者能修改血氧機設定的功能使其更加順手。

- 。

參考文獻

附錄

材料清單-

表 6 材料清單

類別名稱	材 料 名 稱	單位	數量	應 用 說 明	備註
文具	A3 再生紙張	張	2	1.繪製流程圖	
文具	A4 再生紙張	張	N	1 繪圖 2.討論	
單晶片 微控制器	ESP32-TTGO	個	1	主控制器:(OLED、 MAX30102、line Notify)	
感測器	MAX30102	個	2	感測心跳與血氧	
顯示器	OLED	個	2	顯示數值	
耗材	PCB 板	張	2	電路板	
耗材	3D 列印材料	卷	N	外殼	
耗材	杜邦線	卷	4	連接元件	
耗材	顯影劑	袋	3	顯影	
微動開關	D2F-01F	個	3	控制	

設備清單

表 7 設備清單

類別	設備、軟體名稱	應用說明
硬體	PC	1.3D 外殼設計 2.進行電路板繪製 3.報告撰寫
硬體	手機	1. 紀錄製作過程 2. 紀錄成果 3. 遠距離討論及檔案傳輸
硬體	3D 列印機	印製外殼
硬體	蝕刻機	蝕刻電路板
硬體	曝光機	曝光電路板
軟體	Altium Designer	繪製電路板
軟體	Microsoft word	資料書寫
軟體	Microsoft Powerpoint	簡報製作
軟體	Blender	3D 建模
軟體	Cura	將 STL 轉換為 Gcode 檔，以匯入 3D 列印機

成員簡歷

表 8 成員簡歷-余駿凱

姓名	余駿凱	班級	電子三甲	
曾修習專業科目	基本電學(含實習) 電腦輔助設計實習 電子學(含實習) 程式設計實習 數位邏輯 行動裝置應用實習 微處理機(含實習) 可程式邏輯設計實習			
參與專題工作項目	資料收集 資料整理 專題準備 製作計畫書 PPT 製作 歷程報告 軟體規劃 程式撰寫 前期測試 外殼建模 電路板設計 電路板印刷 綜合測試 撰寫期末報告			
經歷簡介	工業電子丙級檢定			

表 9 成員簡歷-曾熙倫

姓名	曾熙倫	班 級	電子三甲	
曾修習專業科目	基本電學(含實習) 電腦輔助設計實習 電子學(含實習) 程式設計實習 數位邏輯 行動裝置應用實習 微處理機(含實習) 可程式邏輯設計實習			
參與專題工作項目	資料收集 資料 word 整理 製作計畫書 PPT 製作 歷程報告 軟體規劃 程式撰寫 前期測試 綜合測試 撰寫期末報告			
經歷簡介	工業電子丙級檢定			

表 10 成員簡歷-楊宇詮

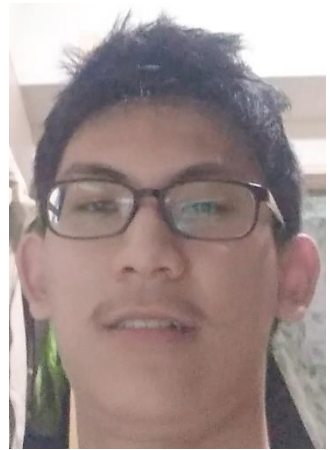

姓 名	楊宇詮	班 級	電子三甲	
曾修習 專業科目	基本電學(含實習) 電腦輔助設計實習 電子學(含實習) 程式設計實習 數位邏輯 行動裝置應用實習 微處理機(含實習) 可程式邏輯設計實習			
參與專題 工作項目	資料收集 資料 word 整理 專題準備 PPT 製作 歷程報告 硬體規劃 前期測試 外殼設計 外殼建模 3D 列印 or 雷射雕刻 綜合測試 撰寫期末報告			
經歷簡介	工業電子丙級檢定			

表 11 成員簡歷-楊適齊

姓名	楊適齊	班級	電子三甲	
曾修習專業科目	基本電學(含實習) 電腦輔助設計實習 電子學(含實習) 程式設計實習 數位邏輯 行動裝置應用實習 微處理機(含實習) 可程式邏輯設計實習			
參與專題工作項目	資料收集 資料 word 整理 專題準備 PPT 製作 歷程報告 硬體規劃 前期測試 外殼設計 電路板設計 電路板印刷 3D 列印 or 雷射雕刻 綜合測試 撰寫期末報告			
經歷簡介	工業電子丙級檢定			