

泳池管理系統

前言

室內泳池是一種提供游泳運動、教學、訓練、休閒等多元功能的運動場地，其管理涉及水質、空氣、溫度、濕度、消毒、安全、設備、人員等多方面的因素，對於維護游泳池的運作效率、服務品質、使用者滿意度和環境衛生有著重要的作用。然而，室內泳池管理的過程中，往往會遇到各種問題和挑戰，例如水質污染、空氣不良、溫度不適、濕度過高、消毒不足、安全隱患、設備故障、人員不足等，這些問題不僅影響游泳池的正常運作，也可能危害使用者的健康和 safety，甚至造成經濟損失和法律風險。因此，如何建立一套有效的室內泳池管理系統，是一個值得關注和探討的課題。

本文旨在探討室內泳池管理系統的設計與實施，主要分為以下幾個部分：第一部分介紹室內泳池自動化管理內容，並分析室內泳池管理的現狀和問題；第二部分介紹室內泳池管理系統的概念、架構和功能，並探討室內泳池管理系統的建置方法和步驟；第三部分介紹室內泳池管理系統的運用與評估，並舉例說明室內泳池管理系統的實際應用和效果。

本文的研究方法主要採用實際案例分析和比較分析，參考網路文獻和實例，從理論和實務的角度，對室內泳池管理系統進行深入的建置和評估，期望能夠為室內泳池管理的理論和實踐提供一些有益的參考和建議。

壹、動機

在台灣，每年夏天都會發生不少戲水意外，造成許多人員傷亡。尤其是花蓮，因為地處東部海岸，有許多美麗的海灘和河流，吸引了大批的遊客前來玩水。然而，這些水域往往隱藏著危險，如暗流、漩渦、水草、水中障礙物等，如果不注意安全，就可能發生意外。

為了提高人民的水域安全意識，花蓮縣政府近年來積極推動水域安全教育，並興建了多座公共游泳池，希望能將人們的戲水活動轉移到更安全的環境中。然而，游泳池也不是絕對安全的，如果管理不善，也可能發生水質污染、溺水、碰撞、滑倒等意外事故，危及人們的生命健康。

因此，我們團隊開發了一套泳池管理系統，旨在提升游泳池的安全管理水準，並減少意外事故的發生。我們的泳池管理系統主要有以下幾個功能：

- 水質監測：我們的系統可以即時監測游泳池的水質，包括濁度及溫度等指標，並將相關數據存入雲端資料庫，標準進行比對，如果發現異常，就會自動調整水質或發出警報，並通知管理人員處理。
- 泳池環境監測：我們的系統可以根據游泳池的需求設置空氣溫濕度及空氣品質等偵測，隨時偵測泳池環境，將數據存入雲端資料庫中，確保泳池區域內都有足夠的安全。
- 更衣室管理：我們的系統可以管理游泳池的更衣室部分，感測器包括溫濕度、光敏、觸摸開關、觸控開關、oled液晶螢幕、燈條等，可監控環境清潔及使用者安全，以維護泳池中更衣室。

- 使用者心律偵測：我們的系統可以預防游泳池的個人生命意外事故，例如，我們的系統可以透過心律感測器，監測游泳池內的運動人員心律動態，如果發現心律問題，就會立即通知救生員和管理人員進行救援，並啟動緊急醫療程序。此外，我們重視運動員生命安全，以減少意外事故的風險。

我們團隊相信，我們的泳池管理系統可以有效地提升游泳池的安全管理水準，並減少意外事故的發生，讓人們可以安心地享受戲水的樂趣，也讓花蓮成為一個更安全的水域休閒之都。

貳、系統

這是一個創客物聯網的作品，利用了當前流行的物聯網(IOT)技術，採用了ESP-32s物聯網晶片，並配合了多種感測器，使其能夠方便地應用於日常生活，並以低成本和高效能為設計目標。這個專題主要分為三個部分：感測器結構、資料庫與物聯網部分。

感測器結構部分，我們使用了溫濕度感測器、光敏電阻、觸控感測器、心律感測器、水質感測器、空氣感測器.....等，並將它們連接

到 E S P-32s 物聯網晶片上。這些感測器可以測量環境中的溫度、濕度、光線、安全、人體心律、空氣品質等數據，並將它們傳送到晶片上進行處理和分析。

資料庫與物聯網部分，我們使用了 mysql 作為我們的雲端資料庫，並利用了 Arduino IDE 和 C 語言來編寫晶片的程式碼。我們讓晶片能夠透過 Wi-Fi 連接到 mysql，並將感測器的數據上傳到資料庫中，同時也能從資料庫中獲取指令和訊息。我們還使用了 Mysql workbench 來管理一個資料庫應用程式，讓使用者能夠透過行動載具遠端控制和監測我們的物聯網系統。

這個創客物聯網的作品，展示了物聯網技術在生活中的應用和潛力，並且以低成本和高效能為原則，實現了一個智慧化的物聯網系統。我們希望這個作品能夠啟發更多的創意和創新，並為物聯網的發展和推廣做出貢獻。

一、系統~感測器

這個專題我們使用多種感測器，我們以下將本題會用到的感測器做說明：

1.光敏電阻:

一種利用光電導效應的特殊電阻，它的電阻和入射光的強弱有直接關係。光強度增加，則電阻減小；光強度減小，則電阻增大。光敏電阻可以用於偵測光線的變化，例如在自動控制、警報、攝影等領域，本主題主要自動開啟燈條控制。



2.溫溼度感測器:

DHT11 溫溼度感測器是一種結合溫度和濕度的感測元件，可以將量測到的數據轉換為數位訊號，並透過單線雙向傳輸方式與微控制器溝通。它的量測範圍是 0-50°C 和 20-90%RH，精確度分別是 $\pm 2^{\circ}\text{C}$ 和 $\pm 5\%\text{RH}$ 。它的優點是體積小、功耗低、成本低、可靠性高，適合用於室內環境的監測。



3.微動開關模組:

碰撞開關模組的工作原理是利用微動開關的觸點來控制電路的通斷。當有物體觸碰到開關的搖桿時，觸點會改變狀態，從而改變輸出信號的電位。

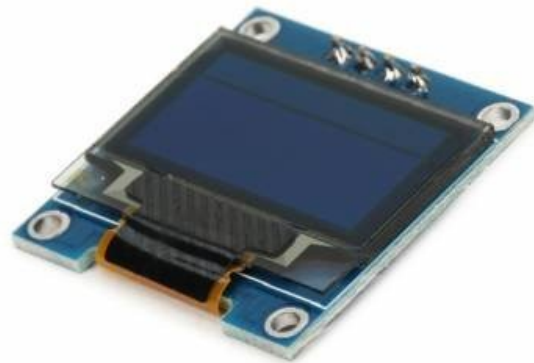
碰撞開關模組的一般接線方式是將 VCC 接到電源正極，GND 接到電源負極，OUT 接到信號輸出。當沒有碰撞時，OUT 輸出高電位，開關指示燈滅；當有碰撞時，OUT 輸出低電位，開關指示燈亮。



4.oled 液晶螢幕

OLED 液晶螢幕是一種顯示技術，它使用有機材料發光，不需要背光源，因此可以實現輕薄、可彎曲、低功耗、高對比度、廣視角等優點。OLED 液晶螢幕目前主要應用在手機、電視、筆記本電腦等產品上；本次作品我們使用 0.91 吋 64*128 液晶螢

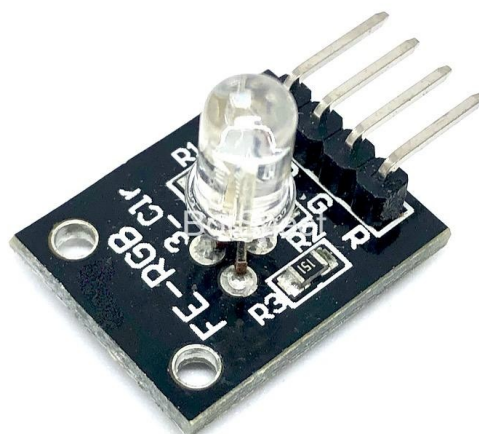
幕，做為顯示資訊用。



5.三色 led 燈

3 色 LED 模組 是一種電子元件，可以發出紅、綠、藍三種顏色的光，並通過調節每種顏色的強度來混合出不同的顏色。

3 色 LED 模組 通常由一個貼片全彩 LED 或三個單色 LED 組成，並有四個針腳，分別是 VCC、R、G、B。3 色 LED 模組可以與 Arduino、ESP32、樹莓派等開發板接口，實現各種燈光效果；本次專題利用此特性來顯示更衣室使用情形，亮綠燈無人使用，亮紅燈有人使用。



6.LED 軟燈條

5050 三晶片【RGB 七彩】LED 軟燈條是一種使用 5050 LED 晶片製成的燈條，可以發出紅、橙、黃、綠、藍、靛、紫、白等多種顏色的光。這種燈條採用軟性電路板設計，可以隨意彎曲，適合用於裝飾、照明、氣氛營造等用途。

5050 三晶片【RGB 七彩】LED 軟燈條的特點有：

- 亮度高，品質穩定，使用壽命長。
- 光色均勻，不同批生產均勻性可達 98% 以上。
- 採用 DC5V/12V/24V 低電壓設計，安全可靠。
- 可以使用遙控器、音樂控制器、手機 APP 等方式控制顏色和模式的變化。
- 可以自行裁剪，根據需要調整長度。



7. 按鈕開關

大按鍵模組和按鈕模組是一種常用的電子元件，可以用於 Arduino 等微控制器的開發和應用。它們的功能是在按下或釋放時產生不同的電平信號，以控制其他電路或裝置的動作。它們通常有不同的顏色和形狀，可以根據需要選擇和組合。



8. 水溫度感測器 DS18B20

水溫度感測器 DS18B20 是一種數位溫度感測器，它可以通過 1-Wire 總線與微處理器進行通信，並提供 9 位至 12 位的攝氏溫度測量。它的外形像一個三極管，並且具有防水、防潮、防生鏽的特性。它的測溫範圍是 -55°C 到 $+125^{\circ}\text{C}$ ，並且可以設置高溫度和低溫度觸發器。它的應用場景包括土壤溫度檢測、水箱溫度控制等等。



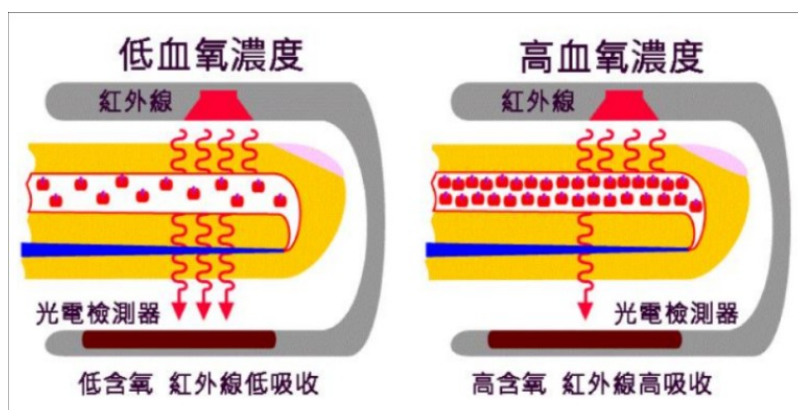
9. MAX30102 血氧、心跳感測器模組

MAX30102 是一種集成了脈搏血氧儀和心率監測器的生物感測器模組。它可以通過光反射信號（PPG）來測量血液中的氧氣飽和度和心跳頻率。它適用於可穿戴設備和健康監測應用。

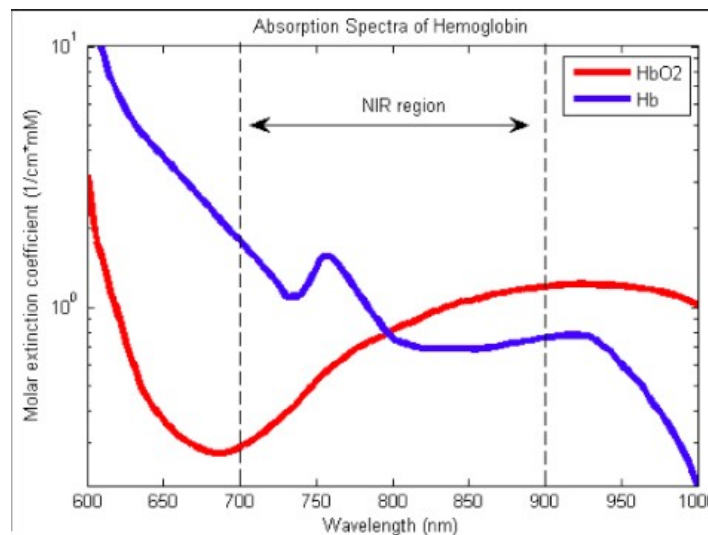
血氧感測器 Max30102 有別於以往須抽血檢驗，利用光學偵測更加方便安全。原理是利用

帶氧血紅素：紅外線吸收率高(850-1000nm)

未帶氧血紅素：紅光吸收率高(600-750nm)



當感測器將紅光與紅外光都發射打到人體內，計算被吸收的數值，就可以反推血氧的含量了。



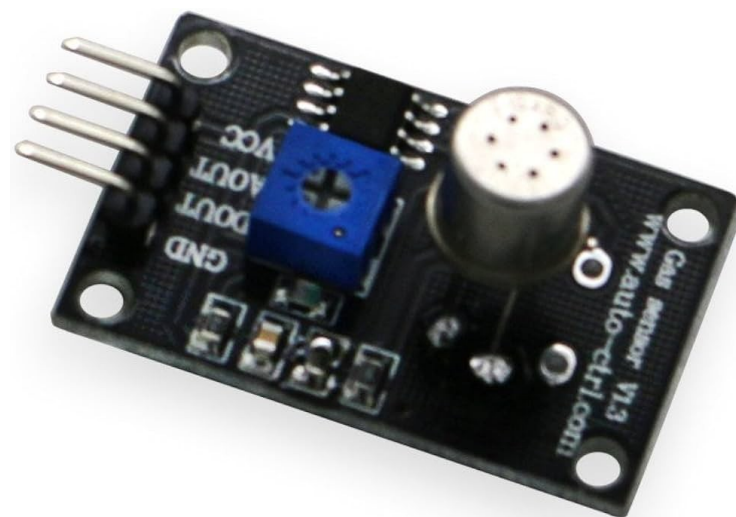
10. RFID

RFID RC522 是一個模組，可以使用 MFRC522 晶片和 Arduino 來讀寫 RFID 卡片。RFID 是無線射頻識別的簡稱，它是一種利用無線電波與標籤或卡片通訊的技術，標籤或卡片可以儲存資訊。RFID RC522 的工作頻率是 13.56 MHz，可以讀取 Mifare 卡片，這種卡片廣泛用於停車場代幣、現金卡、員工證等應用。



11. TGS2602 模組

TGS2602 模組是一種空氣品質嗅味氣體檢測感測器，可以測量空氣中的有害氣體，如氨、硫化氫、甲醛等。TGS2602 模組的工作原理是利用費加羅公司的 TGS2602 探頭，該探頭是一種半導體氣體傳感器，其電阻會隨著空氣中氣體濃度的變化而變化。TGS2602 模組可以應用於空氣清新機、空氣監測系統、空氣污染控制等領域。



12. esp32 自由硬體

ESP32 是一款由上海樂鑫科技開發的低功耗微控制器，它集成了 Wi-Fi 和雙模藍牙功能，適用於物聯網和無線通訊領域。ESP32 採用 Tensilica Xtensa LX6 雙核或單核微處理器，內置無線開關，RF 換能器，功率放大器，低噪聲接收放

大器，濾波器和電源管理模組。ESP32 是 ESP8266 的後繼產品，由台積電採用 40 奈米製程生產。

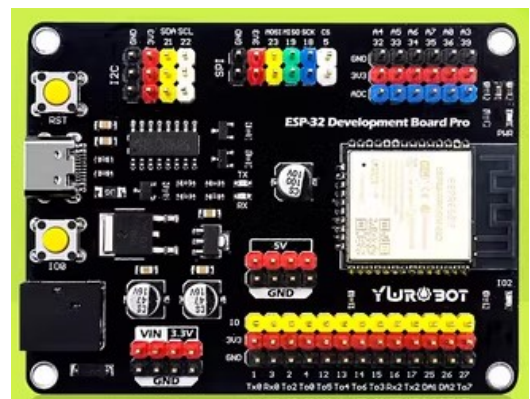
ESP32 的主要特性如下：

- 處理器：CPU: Xtensa 雙核（或單核）32 位元 LX6 微處理器，時鐘速度 160/240 MHz，算力高達 600 DMIPS
- 儲存：448 KB ROM (64KB+384KB)，520 KB SRAM
- 無線連接：Wi-Fi: 802.11 b/g/n，藍牙：v4.2 BR/EDR 和 BLE（和 Wi-Fi 共享射頻模組）
- 外設介面：34 個可程式化 GPIO s，12 位元 SAR ADC，高達 18 個通道，2 個 8 位元 DAC，10 個輕觸感測器（電容式感應 GPIOs），4 × SPI，2 × I²S 介面，2 × I²C 介面，3 × UART，SD / SDIO /CE-ATA/ MMC / eMMC 主控制器，SDIO/SPI 從控制器，具有專用 DMA 和計劃支援 IEEE 精準時間協定 3 的 乙太網路 介面，控制器區域網路 2.0，紅外控制器（TX/RX，多達 8 通道），脈衝計數器（支援 全正交 解碼），電機 PWM，LED PWM（多達 16 通道），超低功耗類比前置放大器

- 安全特性：支援全部 IEEE 802.11 標準安全功能，包括 WPA, WPA2, WPA3（取決於版本）4 以及 無線區域網路鑑別與保密基礎結構（WAPI），安全啟動，ROM 加密，1024 位元 OTP，硬體加密加速：AES, SHA-2, RSA, ECC, 亂數生成（RNG）
- 電源管理：內部 低壓差穩壓器，RTC 獨立電源域，5 μ A 深度睡眠電流，從 GPIO 中斷，定時器，ADC，電容式輕觸感測器中斷喚醒

ESP32 有多種不同的版本，包括 ESP32-S2, ESP32-C3, ESP32-S3, ESP32-C6, ESP32-H2 等，它們有不同的 CPU 和功能，但使用相同的 SDK（ESP-IDF），代碼基本相容。ESP32 還有許多不同的開發板，如 NodeMCU-32s, DOIT DEVKIT V1, ESP32-CAM, TTGO ESP32, Lolin ESP32, WeMos ESP32 mini, HELTEC WiFi LoRa 32 等，它們有不同的外設和應用場景，但核心晶片都是 ESP32。

ESP32 是一款功能強大，成本低廉，易於開發的微控制器，它支援 Arduino 開放性架構，也可以使用 Python, Lua, MicroPython 等語言進行編程。ESP32 適合用於開發智能家居，遠端控制，物聯網設備，無線網路攝像頭，穿戴式裝置，機器人，人工智能等各種創新的項目。ESP32 是一款值得學習和使用的微控制器。



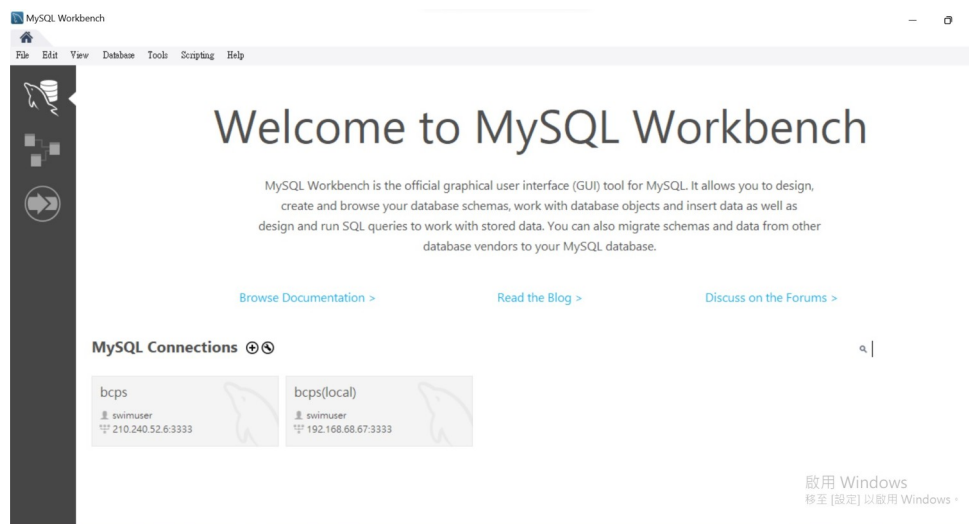
三、 系統~資料庫

本次專題團隊本來利用兩星期時間已架設乙部伺服器，系統為 freeBSD13.2 版，內容為有 nginx、mysql、php、phpmyadmin..... 等等，但因為外網無法連入 mysql 資料庫，教網中心短時間之內無法修正完畢，故只好裝在筆電中，模擬資料庫及網頁部分，所以我們選擇了 workbench 是一款專為 MySQL 設計的可視化資料庫管理工具，可以在 Windows、Linux 和 Mac OS 上使用。workbench 資料庫的

主要功能包括：

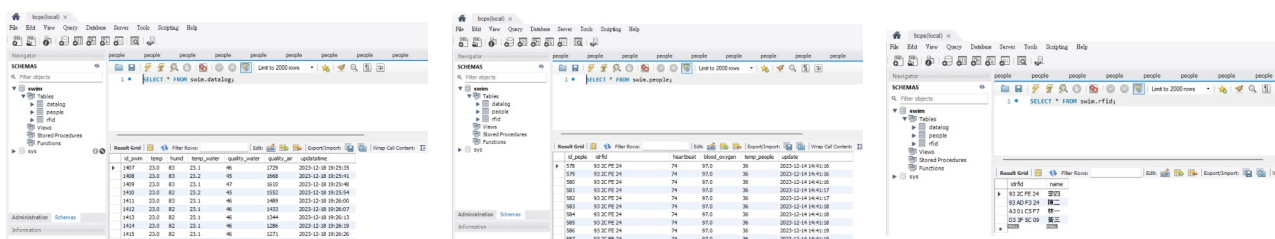
- 資料建模：你可以使用圖形化的介面來建立和修改資料庫的結構，例如資料表、欄位、關聯等。你也可以從現有的資料庫或腳本中匯入或匯出資料模型，並生成相應的 SQL 程式碼。
- SQL 開發：你可以使用內建的查詢編輯器來編寫和執行 SQL 語句，並查看資料庫中的資料和結果。你也可以使用視覺化的工具來建立和管理資料表、索引、檢視、儲存過程等物件。
- 資料庫管理：你可以使用 workbench 資料庫來監控和調整資料庫的效能，並進行資料庫的配置、備份、還原、遷移等操作。你也可以使用 workbench 資料庫來管理資料庫的使用者和權限。

它提供了資料建模、SQL 開發和資料庫管理等功能，讓你可以方便地操作 MySQL 資料庫。



workbench 安裝完畢後，我們就可以透過 workbench 進入資料庫，以進行設定，包括建立新的資料庫欄位以及開放讓 IoT 可連入更新資料，讓 ESP32 可以 WiFi 遠端更新資料庫。

這個專題我們設立三個 table，如 datalog、people 及 rfid，下圖所示：



三、系統~程式部分

本專題程式撰寫主要為 C++ 及 PYTHON 兩個部分，C++ 主要是使用在 ESP32、感測器及傳輸資料庫(mysql)中，有三個程式：

1. 泳池環境偵測

```
void loop() {
//*****TGS2602 偵測空氣中的揮發物污染*****
gasval = analogRead(gasSensor);
```

```

Serial.println( gasval );
delay(500);
//*****

//讀取 DHT11
byte temperature = 0;
byte humidity = 0;
int err = SimpleDHTErrSuccess;

sensors.requestTemperatures();
float temperatureC = sensors.getTempCByIndex(0);

if ((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
    Serial.print("Read DHT11 failed, err="); Serial.println(err); delay(1000);
    return;
}
Serial.print("Sample OK: ");
Serial.print((int)temperature); Serial.print(" *C, ");
Serial.print((int)humidity); Serial.println(" H");

//將溫濕度加入 SQL 字串
String INSERT_SQL = "INSERT INTO swim.datalog
(temp,humd,temp_water,quality_water,quality_air) VALUES ('" + String((int)temperature) + "','" +
String((int)humidity) + "','" + String((float)temperatureC) + "','" + String((int)tdsValue) + "','" +
String((int)gasval) + "')";
MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
cur_mem->execute(INSERT_SQL.c_str()); //execute SQL
delete cur_mem;
//conn.close(); // close the connection
Serial.println("Data Saved.");
delay(5000);

static unsigned long analogSampleTimepoint = millis();

if(millis()-analogSampleTimepoint > 4000){ //every 40 milliseconds, read the analog value from
the ADC
    analogSampleTimepoint = millis();
}

```

```
    analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin); //read the analog value and
store into the buffer
```

```
    analogBufferIndex++;
    if(analogBufferIndex == SCOUNT){
        analogBufferIndex = 0;
    }
}
```

```
static unsigned long printTimepoint = millis();
if(millis()-printTimepoint > 800U){
    printTimepoint = millis();
    for(copyIndex=0; copyIndex<SCOUNT; copyIndex++){
        analogBufferTemp[copyIndex] = analogBuffer[copyIndex];
```

```
    // read the analog value more stable by the median filtering algorithm, and convert to voltage
value
```

```
    averageVoltage = getMedianNum(analogBufferTemp,SCOUNT) * (float)VREF / 4096.0;
```

```
//
```

```
    //temperature compensation formula: fFinalResult(25^C) =
fFinalResult(current)/(1.0+0.02*(fTP-25.0));
    float compensationCoefficient = 1.0+0.02*(temperature-25.0);
    //temperature compensation
    float compensationVoltage=averageVoltage/compensationCoefficient;
```

```
    //convert voltage value to tds value
    tdsValue=(133.42*compensationVoltage*compensationVoltage*compensationVoltage -
255.86*compensationVoltage*compensationVoltage + 857.39*compensationVoltage)*0.5;
```

```
    //Serial.print("voltage:");
    //Serial.print(averageVoltage,2);
    //Serial.print("V ");
    Serial.print("TDS Value:");
    Serial.print(tdsValue,0);
    Serial.println("ppm");
```

```

    }
}

}

```

2. 泳者心律偵測

```

void loop() {

//*****rfid*****

if (mfr522.PICC_IsNewCardPresent() && mfr522.PICC_ReadCardSerial()) {
    // 顯示卡片內容
    Serial.print(F("Card UID:"));
    dump_byte_array(mfr522.uid.uidByte, mfr522.uid.size); // 顯示卡片的 UID
    Serial.println();
    Serial.print(F("CC:"));
    Serial.println(mfr522ReadCode);
    //   mfr522ReadCode = "";
    //   Serial.print(F("PICC type: "));
    //   MFRC522::PICC_Type piccType = mfr522.PICC_GetType(mfr522.uid.sak);
    //   Serial.println(mfr522.PICC_GetTypeName(piccType)); //顯示卡片的類型

    //   mfr522.PICC_HaltA(); // 卡片進入停止模式
    }

// *****rfid end*****

long irValue = particleSensor.getIR(); //Reading the IR value it will permit us to know if there's a
finger on the sensor or not
//是否有放手指
if (irValue > FINGER_ON ) {

    //檢查是否有心跳，測量心跳
    if (checkForBeat(irValue) == true) {

        //顯示血氧數值

```

```

//Serial.print("Bpm="); Serial.println(beatAvg);//將心跳顯示到序列
long delta = millis() - lastBeat;//計算心跳差
lastBeat = millis();
beatsPerMinute = 60 / (delta / 1000.0);//計算平均心跳
if (beatsPerMinute < 255 && beatsPerMinute > 20) {
    //心跳必須再 20-255 之間
    rates[rateSpot++] = (byte)beatsPerMinute; //儲存心跳數值陣列
    rateSpot %= RATE_SIZE;
    beatAvg = 0;//計算平均值
    for (byte x = 0 ; x < RATE_SIZE ; x++) beatAvg += rates[x];
    beatAvg /= RATE_SIZE;
}
}

```

//測量血氧

```

uint32_t ir, red ;
double fred, fir;
particleSensor.check();//Check the sensor, read up to 3 samples
if (particleSensor.available()) {
    i++;
    ir = particleSensor.getFIFOIR();//讀取紅外線
    red = particleSensor.getFIFORed();//讀取紅光
    //Serial.println("red=" + String(red) + ",IR=" + String(ir) + ",i=" + String(i));
    fir = (double)ir;//轉 double
    fred = (double)red;//轉 double
    aveir = aveir * frate + (double)ir * (1.0 - frate); //average IR level by low pass filter
    avered = avered * frate + (double)red * (1.0 - frate);//average red level by low pass filter
    sumirrms += (fir - aveir) * (fir - aveir);//square sum of alternate component of IR level
    sumredrms += (fred - avered) * (fred - avered);//square sum of alternate component of red level

    if ((i % Num) == 0) {
        double R = (sqrt(sumirrms) / aveir) / (sqrt(sumredrms) / avered);
        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(",SPO2="); Serial.println(ESpO2);
        sumredrms = 0.0; sumirrms = 0.0; SpO2 = 0;
    }
}

```



```

    i = 0;
}
particleSensor.nextSample(); //We're finished with this sample so move to next sample
}

//將數據顯示到序列
Serial.print("Bpm:" + String(beatAvg));
//顯示血氧數值，避免誤測，規定心跳超過 30 才能顯示血氧
if (beatAvg > 65) {Serial.println(",SPO2:" + String(ESpO2));
String INSERT_SQL = "INSERT INTO swim.people (idrfid,heartbeat,blood_oxygen) VALUES ('" +
String(String(mfrc522ReadCode)) + "','" + String((int)beatAvg) + "','" + String((int)ESpO2) + "')";
MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
cur_mem->execute(INSERT_SQL.c_str()); //execute SQL
delete cur_mem;
delay(500);
}
else Serial.println("無效值");
// if (beatAvg > 68) {

}
//沒偵測到手指，清除所有數據及螢幕內容顯示"Finger Please"
else {
//清除心跳數據
for (byte rx = 0 ; rx < RATE_SIZE ; rx++) rates[rx] = 0;
beatAvg = 0; rateSpot = 0; lastBeat = 0;
//清除血氧數據
avered = 0; aveir = 0; sumirrms = 0; sumredrms = 0;
SpO2 = 0; ESpO2 = 90.0;
}
}
}

```

3. 更衣室程式

```

void loop() {
    printLocalTime();
    byte temperature = 0;
    byte humidity = 0;
    int err = SimpleDHTErrSuccess;

```

```

val_1=digitalRead(TOUCH_PIN);
// start working...
if ((err = dht11.read(pinDHT11, &temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
    Serial.print("Read DHT11 failed, err="); Serial.println(err);delay(1000);
    return;
}

Serial.print("Humidity = ");
Serial.print((int)humidity);
Serial.print("% , ");
Serial.print("Temperature = ");
Serial.print((int)temperature);
Serial.println("C ");
delay(1000); //每 1 秒顯示一次
display.setCursor(0, 22);//設定游標位置
display.print("temp:");display.println((int)temperature);
display.setCursor(0, 33);//設定游標位置
display.print("humi:");display.println((int)humidity);
//小夜燈*****
int val=analogRead(light_sensor);
if (val<1200){
    digitalWrite(led, LOW);}
else digitalWrite(led, HIGH);
//*****
//使用狀況*****
if (val_1==0){
    digitalWrite(led_red, LOW);
    digitalWrite(led_green, HIGH);
    display.setTextSize(1);//設定文字大小
    display.setTextColor(WHITE);//文字顏色
    display.setCursor(0, 44);//設定游標位置
    display.print("room1:using");
}
else {
    digitalWrite(led_green, LOW);
    digitalWrite(led_red, HIGH);
    display.setTextSize(1);//設定文字大小
    display.setTextColor(WHITE);//文字顏色

```

```

display.setCursor(0, 44);//設定游標位置
display.print("room1:no use");
}
display.display();//顯示螢幕

//觸摸開關*****
int ctsValue = digitalRead(ctsPin);

if (ctsValue == HIGH){
    digitalWrite(buzzPin, HIGH); //有源蜂鳴器響起
    String message = "發生危險，目前狀態：";
    Serial.println(message);
    //*****line post*****
    if (client.connect(host, 443)) {
        int LEN = message.length();
        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();
        //等候回應
        delay(2000);
        String response = client.readString();
        //顯示傳遞結果
        Serial.println(response);
        client.stop(); //斷線，否則只能傳 5 次
    }
    else {
        //傳送失敗
        Serial.println("connected fail");
    }
}
}

```

```

else{
digitalWrite(buzzPin, LOW); //有源蜂鳴器關閉
delay(2000);
}

```

```

# 導入所需的模組
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

from matplotlib.font_manager import FontProperties

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['Microsoft JhengHei']

# 設定資料庫連線參數
db_settings = {
    "host": "127.0.0.1",
    "port": 3333,
    "user": "swimuser",
    "password": "bcps8562620",
    "db": "swim",
    "charset": "utf8"
}

# 建立資料庫連線
conn = pymysql.connect(**db_settings)

# 查詢資料庫中的溫溼度資料
sql = "SELECT updatetime, quality_water, quality_air FROM datalog"
df = pd.read_sql(sql, conn)

# 關閉資料庫連線
conn.close()

# 設定圖表標題和軸標籤
plt.title("涇湖泳池環境品質偵測折線圖")
plt.xlabel("日期")
plt.ylabel("空氣品質 / 水品質")

# 繪製溫度和濕度的折線圖，並設定顏色和樣式
plt.plot(df["updatetime"], df["quality_water"], color="red", marker="o", linestyle="--", label="空氣溫度")
plt.plot(df["updatetime"], df["quality_air"], color="blue", marker="s", linestyle="-", label="空氣濕度")
# plt.plot(df["updatetime"], df["temp_water"], color="green", marker="s", linestyle="-", label="水溫度")

# 顯示圖例
plt.legend()

# 顯示圖表
plt.show()

```

```

# 導入所需的模組
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

from matplotlib.font_manager import FontProperties

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['Microsoft JhengHei']

# 設定資料庫連線參數
db_settings = {
    "host": "127.0.0.1",
    "port": 3333,
    "user": "swimuser",
    "password": "bcps8562620",
    "db": "swim",
    "charset": "utf8"
}

# 建立資料庫連線
conn = pymysql.connect(**db_settings)

# 查詢資料庫中的溫溼度資料
sql = "SELECT updatetime, temp, humd, temp_water FROM datalog"
df = pd.read_sql(sql, conn)

# 關閉資料庫連線
conn.close()

# 設定圖表標題和軸標籤
plt.title("涇湖泳池環境偵測折線圖")
plt.xlabel("日期")
plt.ylabel("空氣溫度(°C) / 空氣濕度(%) / 水溫度(°C) /")

# 繪製溫度和濕度的折線圖，並設定顏色和樣式
plt.plot(df["updatetime"], df["temp"], color="red", marker="o", linestyle="--", label="空氣溫度")
plt.plot(df["updatetime"], df["humd"], color="blue", marker="s", linestyle="-", label="空氣濕度")
plt.plot(df["updatetime"], df["temp_water"], color="green", marker="s", linestyle="-", label="水溫度")

# 顯示圖例
plt.legend()

# 顯示圖表
plt.show()

```

```

# 導入所需的模組
import pymysql
import pandas as pd
import matplotlib.pyplot as plt

from matplotlib.font_manager import FontProperties

import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['Microsoft JhengHei']

# 設定資料庫連線參數
db_settings = {
    "host": "127.0.0.1",
    "port": 3333,
    "user": "swimuser",
    "password": "bcps8562620",
    "db": "swim",
    "charset": "utf8"
}

# 建立資料庫連線
conn = pymysql.connect(**db_settings)

# 查詢資料庫中的溫溼度資料
sql = "SELECT updata, heartbeat, blood_oxygen FROM people"
df = pd.read_sql(sql, conn)

# 關閉資料庫連線
conn.close()

# 設定圖表標題和軸標籤
plt.title("泳者心律偵測折線圖")
plt.xlabel("日期")
plt.ylabel("心跳 / 血氧")

# 繪製溫度和濕度的折線圖，並設定顏色和樣式
plt.plot(df["updata"], df["heartbeat"], color="red", marker="o", linestyle="--", label="心跳")
plt.plot(df["updata"], df["blood_oxygen"], color="blue", marker="s", linestyle="-", label="血氧")

```

肆、 總結

施作此專題遇到了一些挑戰和收穫。首先，我需要學習和使用一些相關的硬體和軟體工具，例如感測器、資料庫、雲端服務、網頁開發、伺服器架站及 python 繪出折線圖等，這些都是我之前不太熟悉的領域，但也讓我拓寬了視野和技能。其次，我需要考慮和處理一些實際的問題和細節，例如如何確保數據的準確性和安全性，如何處理異常和錯誤，如何提高程式的效能和穩定性，如何提升使用者的體驗和滿意度等，這些都是我之前沒有深入思考過的問題，但也讓我增強了邏輯和創造力。

但是專題利用一些設備和加上 IOT 來維持游泳池的水質和安全。我認為游泳池自動管理有以下幾個好處：

一、省時省力：游泳池自動管理可以減少人工的操作和監測，

讓我有更多的時間享受游泳的樂趣。我不需要每天人工測試水的濁度等指標，只要定期檢查設備的情況和水位就可以了。

二、節能節水：游泳池自動管理可以根據水的狀況和需求未

來將走向調節過濾、循環、消毒等系統的運作，避免浪費電力和水資源。我可以設定一些參數，例如水溫、水壓、過濾時間等，讓設備自動調整，達到最佳的效果和效率。

三、保持水質：游泳池自動管理可以持續地監測和調整泳

池的平衡，防止水的變質和污染。我們可以隨時隨地掌控水的濁度、細菌等問題，也有效防範水質對泳者皮膚、眼睛、頭髮等造成的刺激和傷害。

總之，游泳池自動管理讓游泳者可以更輕鬆、更安心、更舒適地享受游泳的樂趣。我覺得這是一個非常值得研究推廣的項目，也是一個符合環保和節能的選擇。如果你也有興趣，你可以參考

相關的資訊；本專題是一個充滿挑戰和收穫的過程，我從中學習到了很多知識和技能，也體驗到了很多樂趣和成就感。我希望我的程式能夠為游泳池的管理和使用帶來便利和價值，也希望能夠繼續改進和完善我的程式，讓它更加智慧和人性化。