**110 學年度教育部國民及學前教育署科技教育創意實作競賽(新竹縣初賽)**

創意企劃書

隊伍編號：國中資訊010

作品名稱：居家防蚊通知系統

組別：□ 國小資訊組 ■國中資訊組

目錄

[一、作品名稱：居家防蚊通知系統 3](#_Toc89892225)

[二、問題解析與解決策略 3](#_Toc89892226)

[三、作品說明 4](#_Toc89892227)

[四、事件流程圖 4](#_Toc89892228)

[五、程式碼 5](#_Toc89892229)

[六、設備應用 11](#_Toc89892230)

[七、材料清單 11](#_Toc89892231)

[八、團隊分工 12](#_Toc89892232)

[九、參考資料 12](#_Toc89892233)

# 一、作品名稱

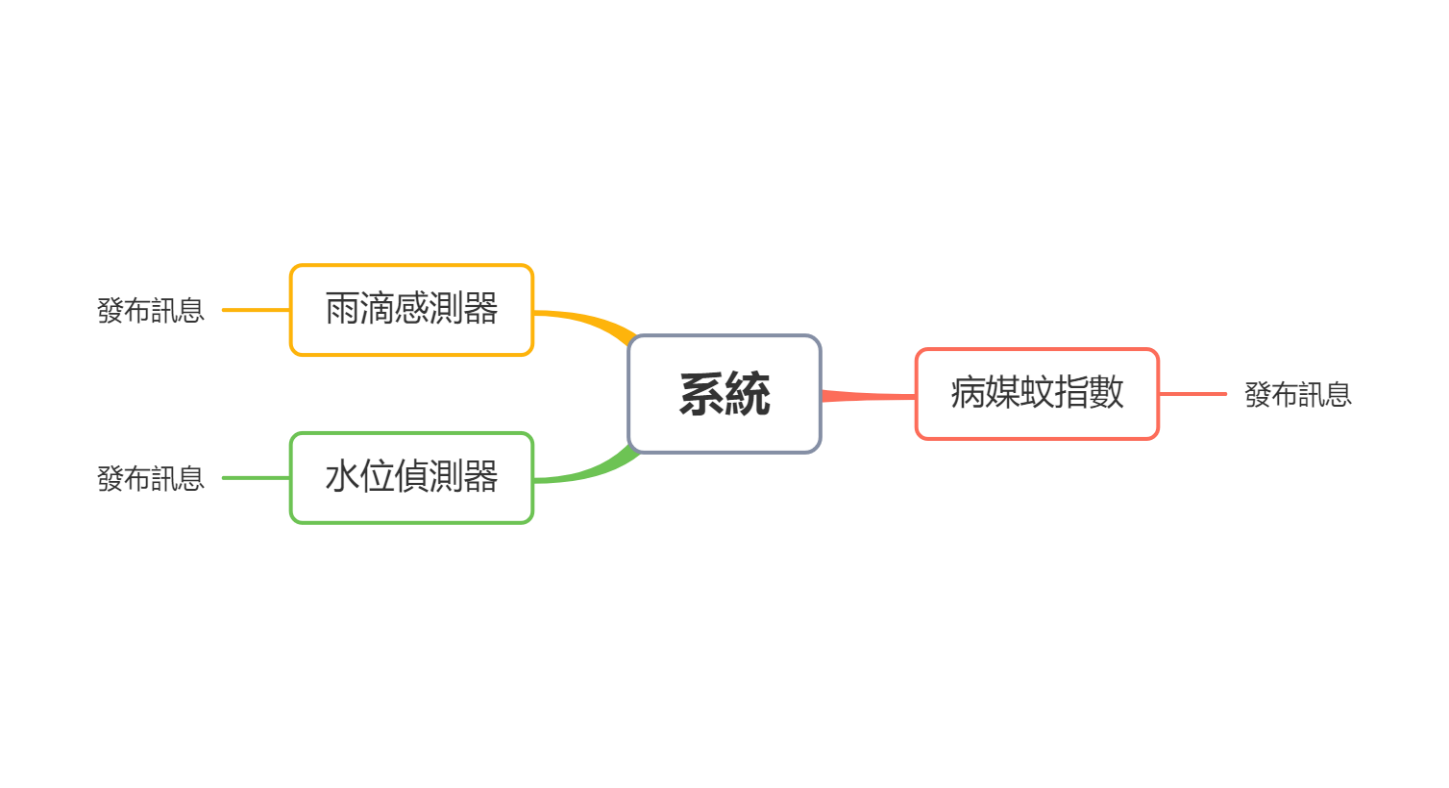
**居家防蚊通知系統**

# 二、問題解析與解決策略

登革熱（Dengue fever），是一種由登革病毒所引起的急性傳染病，這種病毒會經由蚊子傳播給人類。全球登革熱的好發地區，主要集中在熱帶、亞熱帶等有埃及斑蚊和白線斑蚊分布的國家，隨著全球化發展逐漸便利，各國之間相互流通及往返也趨於頻繁，自1980年代之後，登革熱也開始向各國蔓延，成為嚴重的公共衛生問題。臺灣位於亞熱帶地區，像這樣有點熱、又有點溼的環境，正是蚊子最喜歡的生長環境，為登革熱流行高風險地區。(衛福部疾病管制局)

如何能有效預防病媒蚊的滋生就成為一個重要課題。亞洲地區屬於家棲性，幼蟲孳生於住宅內或其周圍之容器內，但在其它地區亦有野棲特性，可以生長於各類型的天然積水容器。 社區巷弄、居家環境潛藏各式各樣民眾意想不到的病媒孳生源，近日中央與地方衛生及環保單位於登革熱個案活動地周邊查核，發現最容易被忽略的孳生源有冷氣機集水桶、盆栽底盤、屋簷排水槽（俗稱天溝）、懸吊電線塑膠管及塑膠椅凹陷處等容易積水處。提醒民眾務必落實「巡、倒、清、刷」，仔細巡視家戶內外積水容器，將積水倒掉，不要的容器予以清除，若有大型廢棄容器如廢棄浴缸、水族箱等，可連絡清潔隊協助清運，留下的器物也要刷洗以去除斑蚊蟲卵，並妥善收拾或予以倒置，使用中的儲水容器則務必每週檢查、刷洗並加蓋或加掛細網。澈底清除孳生源是登革熱防治的根本，並配合各項防治工作，主動清除居家內外的積水容器，才能有效阻斷疫情傳播。

居家積水容器的清理是防止病媒蚊孳生的重要方法之一，為了提醒居家積水容器的清理，本系統利用地方政府所提供的病媒蚊指數作為數據依據，透過系統發出訊息提醒注意已進入登革熱高峰期，要注意個人防蚊以及居家清潔。現代人有時過於忙碌常常疏於清理戶外。因下雨或其他情況所造成的積水而未清理，這是造成病媒蚊孳生的原因，所以系統也提供水位偵測監控居家容器是否積水要及時清理以防病媒蚊的孳生。另外也透過雨滴偵測，在下雨後也提醒要注意並清理積水容器。由於地方政府提供病媒蚊指數，並未及時通知要去清理居家的積水環境，希望透過本系統的提醒去清理居家的積水環境，達到消除病媒蚊的生長環境，達到有效阻斷疫情傳播。

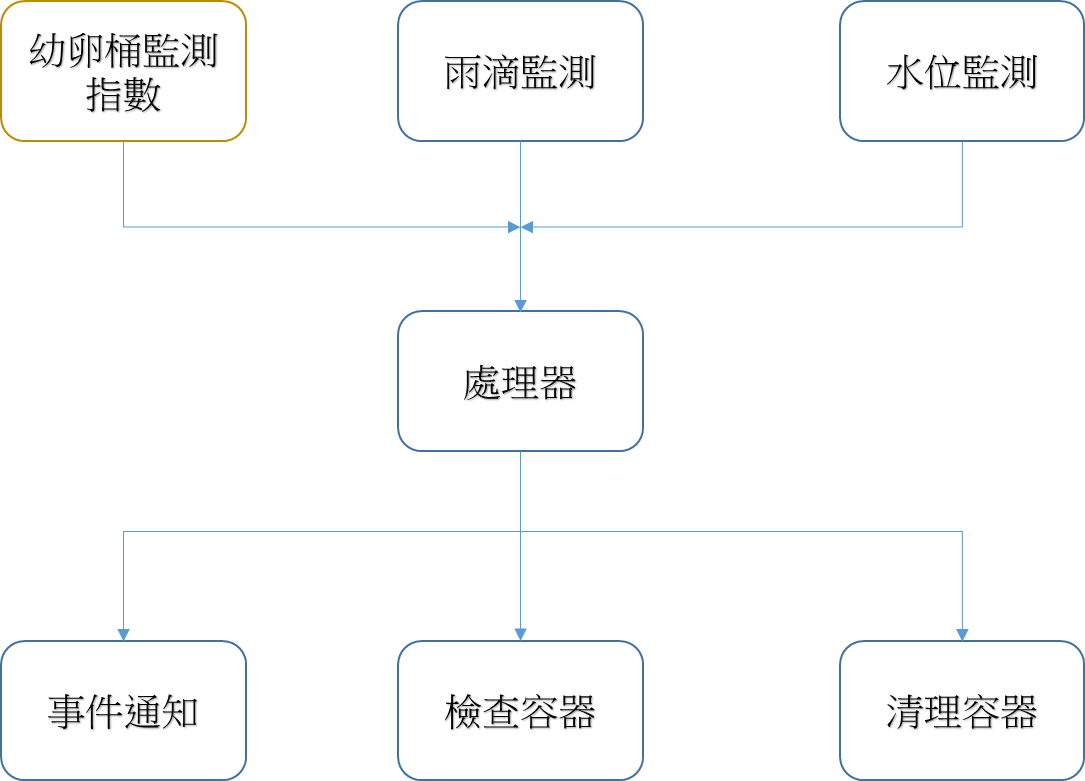


# 三、作品說明

系統分為三大部分:ESP32、MQTT訊息中心、感測器。

1. ESP32: ESP32 是一系列高效能雙核心、低功耗、整合 Wi-Fi 與藍牙的 32 位元微控器, 適合物聯網、可穿戴設備與行動裝置應用。因為ESP32 的功能強大，所以選用ESP32做為系統的核心，負責主控所有的偵測、數據判讀以及訊息的傳遞，更可利用Arduino IDE來開發相關程式。
2. MQTT訊息中心：接收及發布相關信息。利用地方政府提供的誘卵桶監測指數，並依據衛服部疾病管制署建議執行原則緊急成蟲化學防治**-**活動地點其布氏指數在2級(含)以上，或成蚊指數在0.2以上之地點，分析這些指數或數據並依據結果發布相關訊息，也收集感測器數據，將訊息推撥至MQTT 伺服器，讓使用可以訂閱訊息。
3. 感測器：利用水位監測器及雨滴感測器來偵測相關數據再交由系統來處理。平時利用水位監測器監控容器水量提醒積水並清理積水容器。下雨時利用雨滴感測器隨時提醒要注意清理居家的積水容器。

# 四、事件流程圖



# 五、程式碼

#include "FS.h"

#include "SD.h"

#include "SPI.h"

#include <WiFi.h>

#include <PubSubClient.h>

char ssid[] = "ssid"; // 請修改為您連線的網路名稱

char password[] = password"; // 請修改為您連線的網路密碼

int get\_D3 = 36;

int get\_A0 = 39;

int get\_water = 34;

// ------ 以下修改成你MQTT 設定 -------

char\* MQTTServer = "mqtt.eclipseprojects.io"; // 免註冊MQTT 伺服器

int MQTTPort = 1883; // MQTT Port

char\* MQTTUser = ""; // 不須帳密

char\* MQTTPassword = ""; // 不須帳密

// 推播主題1

char\* MQTTPubTopic1 = "YourTopic/class205/temp";

// 推播主題2

char\* MQTTPubTopic2 = "YourTopic/class205/humi";

// 訂閱主題1:

char\* MQTTSubTopic1 = "YourTopic/class205/led";

long MQTTLastPublishTime; // 此變數用來記錄推播時間

long MQTTPublishInterval = 10000; // 每10 秒推撥一次

WiFiClient WifiClient;

PubSubClient MQTTClient(WifiClient);

//讀取SD卡檔案

void readFile(fs::FS &fs, const char \* path){

Serial.printf("Reading file: %s\n", path);

File file = fs.open(path);

if(!file){

Serial.println("Failed to open file for reading");

return;

}

Serial.print("Read from file: ");

while(file.available()){

Serial.write(file.read());

}

file.close();

}

// 開始WiFi 連線

void WifiConnecte() {

// 開始WiFi 連線

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("WiFi 連線成功");

Serial.print("IP Address:");

Serial.println(WiFi.localIP());

}

// 開始MQTT 連線

void MQTTConnecte() {

MQTTClient.setServer(MQTTServer, MQTTPort);

MQTTClient.setCallback(MQTTCallback);

while (!MQTTClient.connected()) {

// 以亂數為ClietID

String MQTTClientid = "esp32-" + String(random(1000000, 9999999));

if (MQTTClient.connect(MQTTClientid.c\_str(), MQTTUser, MQTTPassword)) {

// 連結成功，顯示「已連線」。

Serial.println("MQTT 已連線");

// 訂閱SubTopic1 主題

MQTTClient.subscribe(MQTTSubTopic1);

} else {

// 若連線不成功，則顯示錯誤訊息，並重新連線

Serial.print("MQTT 連線失敗, 狀態碼=");

Serial.println(MQTTClient.state());

Serial.println("五秒後重新連線");

delay(5000);

}

}

}

// 接收到訂閱時

void MQTTCallback(char\* topic, byte\* payload, unsigned int length) {

Serial.print(topic); Serial.print("訂閱通知:");

String payloadString; // 將接收的payload 轉成字串

// 顯示訂閱內容

for (int i = 0; i < length; i++) {

payloadString = payloadString + (char)payload[i];

}

Serial.println(payloadString);

// 比對主題是否為訂閱主題1

if (strcmp(topic, MQTTSubTopic1) == 0) {

Serial.println("改變燈號：" + payloadString);

if (payloadString == "ON") {

digitalWrite(15, HIGH);

}

if (payloadString == "OFF") {

digitalWrite(15, LOW);

}

}

}

void setup() {

// put your setup code here, to run once:

Serial.begin(115200);

if(!SD.begin()){

Serial.println("Card Mount Failed");

return;

}

uint8\_t cardType = SD.cardType();

if(cardType == CARD\_NONE){

Serial.println("No SD card attached");

return;

}

Serial.print("SD Card Type: ");

if(cardType == CARD\_MMC){

Serial.println("MMC");

} else if(cardType == CARD\_SD){

Serial.println("SDSC");

} else if(cardType == CARD\_SDHC){

Serial.println("SDHC");

} else {

Serial.println("UNKNOWN");

}

uint64\_t cardSize = SD.cardSize() / (1024 \* 1024);

Serial.printf("SD Card Size: %lluMB\n", cardSize);

readFile(SD, "/hello.txt");

// 開始WiFi 連線

WifiConnecte();

// 開始MQTT 連線

MQTTConnecte();

pinMode(get\_D3, INPUT);

}

void loop() {

// 如果WiFi 連線中斷，則重啟WiFi 連線

if (WiFi.status() != WL\_CONNECTED) {WifiConnecte();}

// 如果MQTT 連線中斷，則重啟MQTT 連線

if (!MQTTClient.connected()) { MQTTConnecte();}

// 如果距離上次傳輸已經超過10 秒，則Publish 溫濕度

if ((millis() - MQTTLastPublishTime) >= MQTTPublishInterval ) {

// ------ 將DHT11 溫度送到MQTT 主題 ------

//MQTTClient.publish(MQTTPubTopic1, String(temperature).c\_str());

//MQTTClient.publish(MQTTPubTopic2, String(humidity).c\_str());

Serial.println("溫濕度已推播到MQTT Broker");

MQTTLastPublishTime = millis(); // 更新最後傳輸時間

}

water();

rain();

MQTTClient.loop(); // 更新訂閱狀態

delay(50);

}

void water() {

//從A0讀取輸入

int value = analogRead(get\_water);

Serial.print(value);

Serial.print(" ");

if (value > 480) {

Serial.println("High Level");

}

else if ((value > 340) && (value <= 480)) {

Serial.println("Middle Level");

}

else if ((value > 100) && (value <=340)){

Serial.println("Low Level");

}

else if (value <=100){

Serial.println("NO Water");

}

delay(500);

}

void rain() {

int value = analogRead(get\_A0); //從A0讀取感測值

Serial.print(value);

Serial.print(" ");

if(digitalRead(get\_D3) == LOW){

Serial.println("Digital value : wet");

}

else{

Serial.println("Digital value : dry");

}

delay(1000);

}

# 六、設備應用

1.電腦

2.焊接工具

3.麵包版(初步接線測試)

4.單芯線

# 七、材料清單

1.Esp32 x1

2.雨滴感測器 x1

3.水位感測器 x1

4.電路板 x1

5.積木組合屋 x1

# 八、團隊分工

1.程式撰寫：由丁向恩及黃文顯負責Arduino 程式。

2.硬體組裝：由黃文顯及許宸棨負責硬體組裝，及接線測試。

3.文件整理：由許宸棨、丁向恩、黃文顯共同編寫文件。

# 九、參考資料

登革熱病媒蚊指數。衛福部：疾病管制局。

Arduino。<https://www.arduino.cc/>

IoT 物聯網應用-使用ESP32 開發版與Arduino C 程式語言。尤濬哲。台科大。

ESP32程式設計。曹永忠。千華駐科技。

超圖解物聯網IoT實作入門。趙英傑。旗標出版股份有限公司。

超圖解 ESP32 深度實作。趙英傑。旗標出版股份有限公司。

第四篇 ESP32數位讀取(感測)digitalRead-夜市小霸王。尤濬哲。

https://youyouyou.pixnet.net/blog/category/3441000