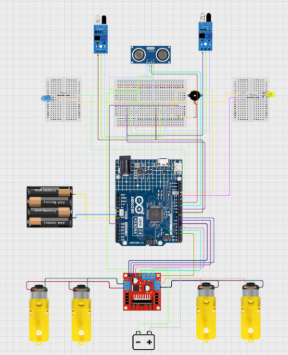
**電路電子學期末報告 自走車實作**

**組員:** **吳庭慧、曾子芸、蔡欣妤、江悅銘**

**電路圖**

****

**特點**

* 會唱生日快樂歌
* 與過去課程結合
* LED在兩側可更清楚知道轉向
* 每0.1秒偵測一次

**遇到的困難**

1.紅外線偵測到物體沒辦法動

程式碼的邏輯不正確，使紅外線有偵測到的時候停止，沒有偵測到的時候前進。

2.對程式的了解不夠熟悉

一開始在做的時候因為一直使用ChatGPT來幫忙，所以程式碼都沒有自己檢查過就上傳了。

3.車速太快 來不及轉彎

我們將程式改成每0.1秒停下來偵測一次，因為經過我們多次測試之後，發現如果使用多餘0.1秒會很容易錯過轉彎時機；小於0.1秒的話，因為時間很短，所以馬達需要很大的轉速才能使車子前進。

4.紅外線偵測不準確

我們調整了紅外線的敏感度後，發現有一邊的紅外線偵測器壞掉了，所以我們更換了紅外線偵測器。

5.生日快樂歌音調不對

參考網路上的程式和影片再進行修改。

**程式碼**

**// --- 馬達控制腳位設定區塊 ---**

**const int ENA = 3, IN1 = 2, IN2 = 4; // 左輪控制**

**const int ENB = 5, IN3 = 6, IN4 = 7; // 右輪控制**

**// --- 紅外線循跡感測腳位設定區塊 ---**

**const int IR\_L = 8, IR\_R = 9; // 左右紅外線循跡感測器（LOW表示偵測到白線）**

**// --- LED 腳位設定區塊 ---**

**const int LED\_L = 13;**

**const int LED\_R = 12;**

**// --- 超音波避障感測器腳位設定區塊 ---**

**const int Trig = 10, Echo = 11;**

**// --- 蜂鳴器腳位（DAC輸出） ---**

**const int pinBuzzer = A0; // 類比腳位 A0**

**// --- 狀態列舉定義 ---**

**enum State { IDLE, TURN\_RIGHT, TURN\_LEFT, MOVE\_FORWARD, STOP\_MOTORS };**

**State state = IDLE;**

**// --- 控制變數 ---**

**unsigned long lastToggleTime = 0;**

**bool motorsOn = false;**

**unsigned long turnStartTime = 0;**

**const unsigned long turnDuration = 800;**

**bool ledToggle = false;**

**bool turnLedToggle = false;**

**// 生日快樂歌設定（使用頻率）**

**int doremi[] = {262, 294, 330, 349, 392, 440, 494, 523, 587, 659, 698, 784, 880, 988};**

**int happybirthday[] = {**

**5,5,6,5,8,7,**

**5,5,6,5,9,8,**

**5,5,12,10,1,7,6,**

**11,11,10,8,9,8**

**};**

**int meter[] = {**

**1,1,2,2,2,4,**

**1,1,2,2,2,4,**

**1,1,2,2,2,2,2,**

**1,1,2,2,2,4**

**};**

**int melodyLength = sizeof(happybirthday) / sizeof(int);**

**int noteIndex = 0;**

**unsigned long noteStartTime = 0;**

**bool playingNote = false;**

**void setup() {**

**pinMode(ENA, OUTPUT); pinMode(IN1, OUTPUT); pinMode(IN2, OUTPUT);**

**pinMode(ENB, OUTPUT); pinMode(IN3, OUTPUT); pinMode(IN4, OUTPUT);**

**pinMode(IR\_L, INPUT); pinMode(IR\_R, INPUT);**

**pinMode(LED\_L, OUTPUT); pinMode(LED\_R, OUTPUT);**

**pinMode(Trig, OUTPUT); pinMode(Echo, INPUT);**

**pinMode(pinBuzzer, OUTPUT);**

**Serial.begin(9600);**

**noteStartTime = millis();**

**}**

**void loop() {**

**unsigned long currentMillis = millis();**

**// 超音波避障偵測**

**digitalWrite(Trig, LOW);**

**delayMicroseconds(2);**

**digitalWrite(Trig, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(Trig, LOW);**

**long dur = pulseIn(Echo, HIGH, 30000);**

**float dist = 0;**

**if (dur > 0) {**

**dist = dur / 58.0;**

**Serial.print("距離: "); Serial.print(dist); Serial.println(" cm");**

**}**

**if (dist > 0 && dist < 15 && state != STOP\_MOTORS) {**

**Serial.println("前方有障礙物，停止！");**

**stopMotors();**

**digitalWrite(LED\_L, HIGH);**

**digitalWrite(LED\_R, HIGH);**

**delay(1000);**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**state = STOP\_MOTORS;**

**motorsOn = false;**

**}**

**// 紅外線循跡感測器讀取**

**int l = digitalRead(IR\_L);**

**int r = digitalRead(IR\_R);**

**switch (state) {**

**case IDLE:**

**Serial.print("IR\_L = "); Serial.print(l);**

**Serial.print(", IR\_R = "); Serial.println(r);**

**if (l == LOW && r == HIGH) {**

**Serial.println("開始右轉");**

**turnStartTime = currentMillis;**

**state = TURN\_RIGHT;**

**motorsOn = false;**

**turnLedToggle = false;**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**} else if (l == HIGH && r == LOW) {**

**Serial.println("開始左轉");**

**turnStartTime = currentMillis;**

**state = TURN\_LEFT;**

**motorsOn = false;**

**turnLedToggle = false;**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**} else if (l == LOW && r == LOW) {**

**Serial.println("前進");**

**state = MOVE\_FORWARD;**

**motorsOn = false;**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**} else if (l == HIGH && r == HIGH) {**

**Serial.println("停止");**

**stopMotors();**

**motorsOn = false;**

**state = STOP\_MOTORS;**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**}**

**break;**

**case TURN\_RIGHT:**

**if (currentMillis - turnStartTime >= turnDuration) {**

**Serial.println("右轉完成，回到IDLE");**

**stopMotors();**

**motorsOn = false;**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**state = IDLE;**

**} else {**

**if (currentMillis - lastToggleTime >= 200) {**

**lastToggleTime = currentMillis;**

**motorsOn = !motorsOn;**

**turnLedToggle = !turnLedToggle;**

**if (motorsOn) {**

**startRightTurn(); Serial.println("右轉馬達開");**

**} else {**

**stopMotors(); Serial.println("右轉馬達停");**

**}**

**digitalWrite(LED\_L, turnLedToggle ? HIGH : LOW);**

**digitalWrite(LED\_R, LOW);**

**}**

**}**

**break;**

**case TURN\_LEFT:**

**if (currentMillis - turnStartTime >= turnDuration) {**

**Serial.println("左轉完成，回到IDLE");**

**stopMotors();**

**motorsOn = false;**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**state = IDLE;**

**} else {**

**if (currentMillis - lastToggleTime >= 200) {**

**lastToggleTime = currentMillis;**

**motorsOn = !motorsOn;**

**turnLedToggle = !turnLedToggle;**

**if (motorsOn) {**

**startLeftTurn(); Serial.println("左轉馬達開");**

**} else {**

**stopMotors(); Serial.println("左轉馬達停");**

**}**

**digitalWrite(LED\_R, turnLedToggle ? HIGH : LOW);**

**digitalWrite(LED\_L, LOW);**

**}**

**}**

**break;**

**case MOVE\_FORWARD:**

**if (!(l == LOW && r == LOW)) {**

**Serial.println("前進感測器條件不符，停止並回IDLE");**

**stopMotors();**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**motorsOn = false;**

**state = IDLE;**

**} else {**

**if (currentMillis - lastToggleTime >= 100) {**

**lastToggleTime = currentMillis;**

**motorsOn = !motorsOn;**

**ledToggle = !ledToggle;**

**if (motorsOn) {**

**moveForward(); Serial.println("前進馬達開");**

**} else {**

**stopMotors(); Serial.println("前進馬達停");**

**}**

**if (ledToggle) {**

**digitalWrite(LED\_L, HIGH);**

**digitalWrite(LED\_R, LOW);**

**} else {**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, HIGH);**

**}**

**}**

**}**

**break;**

**case STOP\_MOTORS:**

**stopMotors();**

**digitalWrite(LED\_L, LOW);**

**digitalWrite(LED\_R, LOW);**

**motorsOn = false;**

**state = IDLE;**

**break;**

**}**

**// --- 非阻塞生日快樂歌播放（偵測兩個感測器同時HIGH就停止唱並重置） ---**

**if (l == HIGH && r == HIGH) {**

**// 兩個HIGH時停止並重置**

**noTone(pinBuzzer);**

**noteIndex = 0;**

**playingNote = false;**

**noteStartTime = currentMillis;**

**} else {**

**// 其他情況持續播放（轉彎時不重置）**

**playHappyBirthdayNonBlocking(currentMillis);**

**}**

**}**

**// 馬達控制函式**

**void startRightTurn() {**

**digitalWrite(IN1, LOW);**

**digitalWrite(IN2, HIGH);**

**analogWrite(ENA, 180);**

**digitalWrite(IN3, HIGH);**

**digitalWrite(IN4, LOW);**

**analogWrite(ENB, 180);**

**}**

**void startLeftTurn() {**

**digitalWrite(IN1, HIGH);**

**digitalWrite(IN2, LOW);**

**analogWrite(ENA, 180);**

**digitalWrite(IN3, LOW);**

**digitalWrite(IN4, HIGH);**

**analogWrite(ENB, 180);**

**}**

**void moveForward() {**

**digitalWrite(IN1, HIGH);**

**digitalWrite(IN2, LOW);**

**analogWrite(ENA, 200);**

**digitalWrite(IN3, HIGH);**

**digitalWrite(IN4, LOW);**

**analogWrite(ENB, 200);**

**}**

**void stopMotors() {**

**digitalWrite(IN1, LOW);**

**digitalWrite(IN2, LOW);**

**analogWrite(ENA, 0);**

**digitalWrite(IN3, LOW);**

**digitalWrite(IN4, LOW);**

**analogWrite(ENB, 0);**

**}**

**// 非阻塞播放生日快樂歌函式**

**void playHappyBirthdayNonBlocking(unsigned long currentMillis) {**

**if (!playingNote) {**

**// 開始新音符**

**int freq = doremi[happybirthday[noteIndex] - 1];**

**int duration = meter[noteIndex] \* 150;**

**playToneStart(freq);**

**noteStartTime = currentMillis;**

**playingNote = true;**

**}**

**// 檢查是否該停止目前音符**

**int duration = meter[noteIndex] \* 150;**

**if (currentMillis - noteStartTime >= duration) {**

**playToneStop();**

**noteIndex++;**

**if (noteIndex >= melodyLength) {**

**noteIndex = 0;**

**}**

**playingNote = false;**

**}**

**}**

**void playToneStart(int frequency) {**

**tone(pinBuzzer, frequency);**

**}**

**void playToneStop() {**

**noTone(pinBuzzer);**

**}**

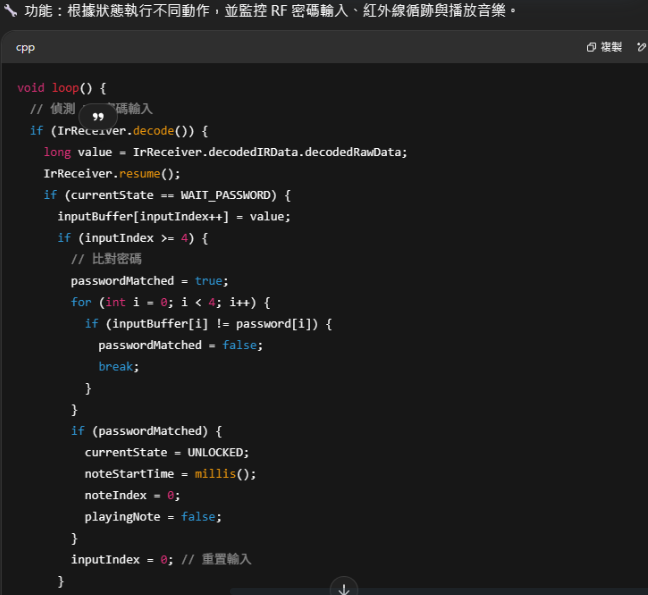
**1.初始化與腳位設定**



**2. setup() 初始化設定**

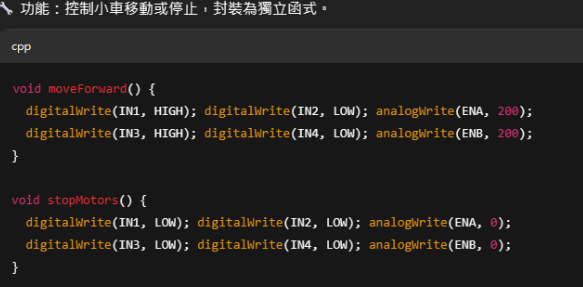
****

**3.loop() 主程式邏輯**

****

****

**4. 馬達控制函式**

****

**硬體設定與腳位定義**

* 馬達控制腳位：ENA、IN1、IN2 控制左輪，ENB、IN3、IN4 控制右輪，使用PWM控制速度與方向
* 紅外線循跡感測器：IR\_L、IR\_R，偵測是否在白線上（LOW代表偵測到白線）
* LED燈：LED\_L、LED\_R，用於顯示狀態提示
* 超音波避障感測器：Trig、Echo，測量前方障礙物距離
* 蜂鳴器：pinBuzzer（A0類比腳位），用來播放生日快樂歌

**狀態機**

* IDLE：等待判斷下一步動作。
* TURN\_RIGHT：右轉。
* TURN\_LEFT：左轉。
* MOVE\_FORWARD：前進。
* STOP\_MOTORS：停止馬達。

超音波避障判斷

* 發送超音波，計算距離。
* 如果距離小於15公分且不是停止狀態，停止馬達並閃爍LED，避免撞到障礙物。

紅外線循跡感測器讀取

* 根據左、右感測器的狀態判斷小車的行進狀態：
* 左感測到白線 (LOW) 右沒感測到白線 (HIGH)：右轉。
* 右感測到白線 左沒感測到白線：左轉。
* 兩邊都感測到白線：前進。
* 兩邊都沒感測到白線：停止。

狀態行為控制

* 根據狀態執行相對應動作（馬達控制及LED閃爍）。
* 轉彎時用定時切換馬達開關，造成間歇性轉動與LED閃爍效果。
* 前進時馬達也以固定頻率開關，LED交替閃爍。

生日快樂歌非阻塞播放

* 利用millis()控制音符的持續時間，避免阻塞主程式。
* 兩個紅外線感測器同時為HIGH時，停止音樂並重置播放。
* 其他狀態持續播放生日快樂歌。

**馬達控制函式**

* startRightTurn()：左輪反轉，右輪正轉，產生右轉動作。
* startLeftTurn()：左輪正轉，右輪反轉，產生左轉動作。
* moveForward()：兩輪正轉，往前移動。
* stopMotors()：停止兩輪馬達。

**音樂播放函式**

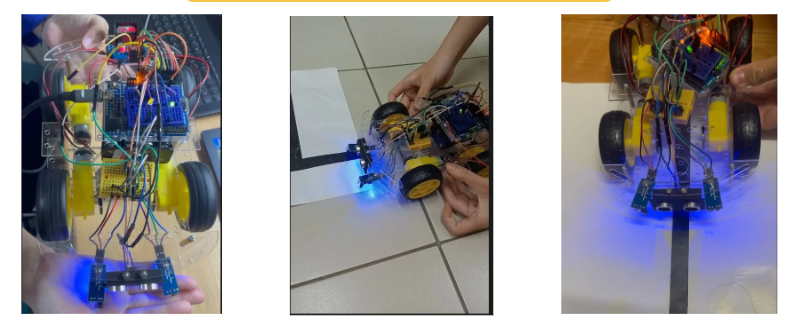
* playHappyBirthdayNonBlocking()
* 用millis()定時控制每個音符播放時長，實現非阻塞音樂播放。
* playToneStart(freq)：啟動指定頻率蜂鳴器聲音。
* playToneStop()：停止蜂鳴器聲音。

[**GitHub專案**](https://github.com/xunlu1205/vegetable_new_fish/blob/31b9d03818c6f973bfccbc203e27fdffcc143f7b/README.txt)

[**https://reurl.cc/EVd0Ag**](https://reurl.cc/EVd0Ag)

**測試影片**

[**https://reurl.cc/5KnxYM**](https://reurl.cc/5KnxYM)

****

[**成果影片**](https://reurl.cc/Dq52Ge)

[**https://reurl.cc/Dq52Ge**](https://reurl.cc/Dq52Ge)

**心得**

這次的電路電子課程，我們所製作的主題是「自走車」，一開始，我們對於感測器、馬達控制和程式邏輯都不熟悉，尤其是在紅外線循跡感測器的判斷邏輯上遇到了困難。當我們第一次將車子放上軌道時，明明已經對準了黑線，車子卻完全沒有動作，讓我們感到非常困惑。這時學長告訴我們，不能一開始就直接讓Chatgpt寫完整的循跡程式，而是應該先從簡單的偵測測試開始，確認感測器輸出的數值，了解它在偵測到黑線和白線時分別回傳的是「0」還是「1」。

後來，我們慢慢釐清了判斷邏輯，原來感測器在讀到黑線時回傳的是高電位（HIGH），而白線則是低電位（LOW）。在理解這些基礎原理後，我們才開始進一步調整控制馬達的條件與方向，讓車子可以正確地沿著埋線前進。

除了邏輯判斷上的困難，我們也遇到了其他部分的問題，例如：車速太快來不及轉彎、轉彎角度太大導致偏離軌道、馬達轉速過低導致車子無法前進等。我們透過不斷地觀察與測試，一點一點地去調整並修改程式邏輯，甚至反覆檢查每一條電路接線。過程中雖然辛苦，但每解決一個問題，我們的成就感也隨之提升。

我們也嘗試使用 AI 協助理解錯誤訊息或修正程式碼，雖然 AI 在程式語法上能提供幫助，但實體電路的狀況卻很難用 AI 完全理解與判斷，例如：電線接反、感測器模組損壞、電壓供應不足等問題，最後還是需要我們自己深入理解電路與程式邏輯去分析原因、調整設計。這讓我們深刻體會到：「AI 是工具，但真正解決問題的還是人本身。」

此外，我們特別加入了「生日快樂歌」的播放功能。我們的靈感來自於生活中的玩具車，這些玩具不僅能動，還會唱歌、閃燈，因此我們也希望自己的作品能更加生活化、更具互動性。剛好之前在課堂上學過播放旋律的程式，於是我們決定把這段經驗應用上來，成為我們作品的一大亮點。

這次的學習過程讓我們深刻理解到，解決問題需要耐心、邏輯與實作經驗。從一開始車子不會動，到最後完成一台會循跡、會唱歌的自走車，我們經歷了很多錯誤與修正。這不僅增進了我們對電子電路與程式的理解，也讓我們學會如何面對困難、不急躁，並從錯誤中學習。

從發現問題到解決問題與獨立思考的能力。透過結合過去所學與實際應用，創造出屬於自己的作品，並且從中獲得成就與樂趣。