

機電整合-final report:

Line tracker & maze runner

B07611002 鄭泊聲

演算法說明

Line tracker

1. 以 PWM 方式驅動馬達，基本上沿用 lab 的 code，分別寫成前進、左轉、右轉的 function。
2. 以 distance meter lab 所提供的讀值 code 來讀取 IR SENSOR 的數值，以 700 為 threshold，大於 700 視為看到黑線。

主程式最初開始執行時，先讀取當三個 IR sensor 的數值，接著首先檢查是否有中央 IR 是否有感測到黑線，若有，則觸發前進 function，且不會進入檢查左右 IR 的 else if，直接回到主程式開頭，重新讀取三個數值；若沒有，才檢查左右 IR。左 IR 感到黑線時，觸發左轉 function；右 IR 則相反。整個程式中沒有觸發馬達停止的機制，若都沒有看到黑線時，不會觸發任何 function，則車輛會以原本的方式繼續作動(維持前進、維持向左或維持向右)。

由於程式優先檢查前 IR 的數值是否觸發前進，只要中央有黑線，無論兩旁是否有黑線，就會直進，因此理論上在交叉口會選擇直進。

整個沒有使用到 delay，也不是透過 delay 來控制每次左右擺動的幅度。由於每一次主程式觸發馬達作動 function(前進、左轉、右轉)後，進會立刻回到迴圈頂端，進行 IR 的讀值，在重新判別車輛應前進方向。因此不須透過 delay 來控制擺動幅度，若車輛偏移，那麼就持續修正，直到在迴圈中中央 IR 讀到黑線時(也就是已經修正回到中央時)，車輛自然會回到前進狀態。

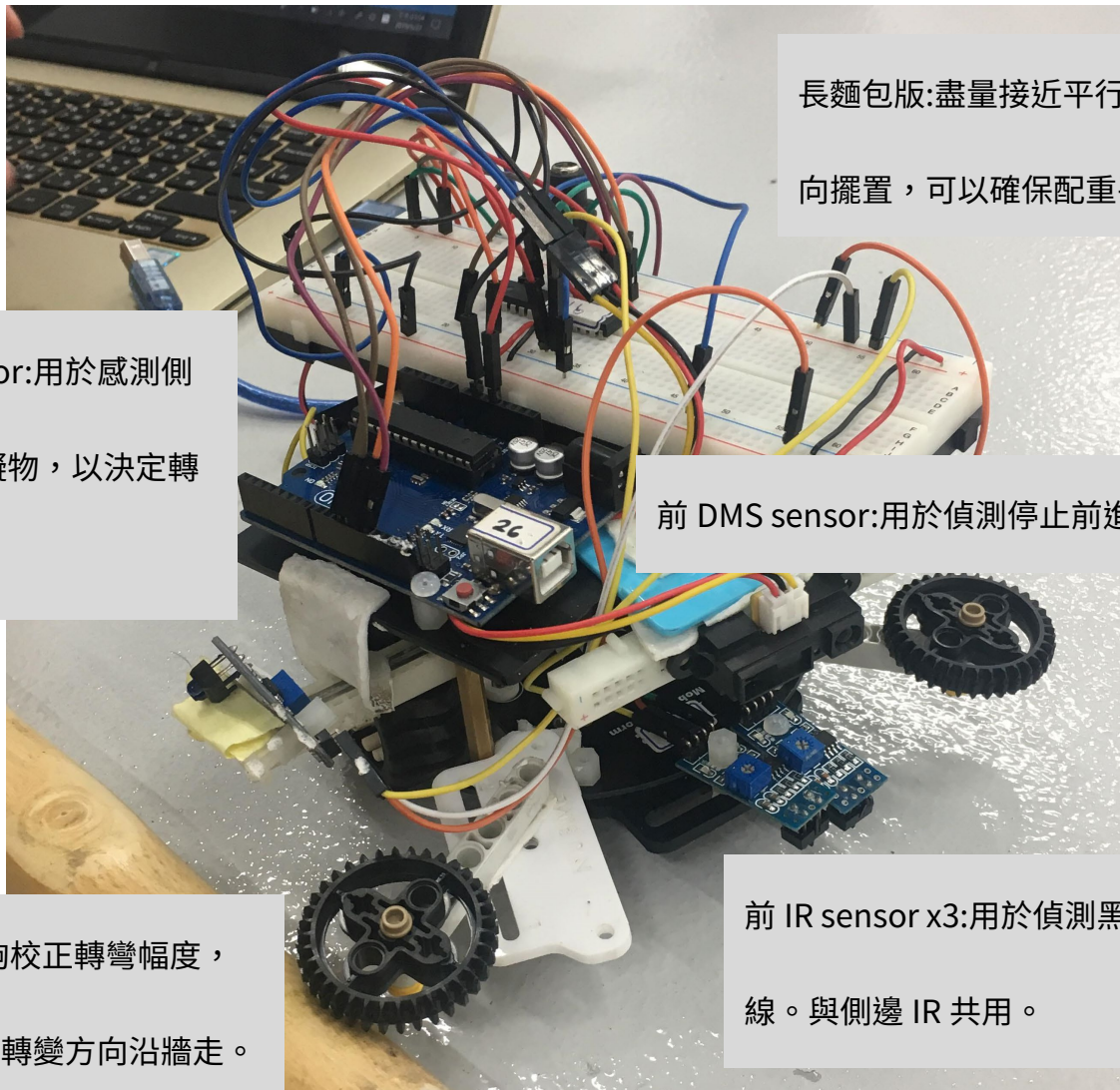
Maze runner

以 distance meter lab 所提供的讀值 code 來讀取 IR SENSOR 及 DMS sensor 的數值，以 800 為側邊 IR 的 threshold，大於 800 視為目前側邊有障礙物；以 350-400 左右(每次數值略會微調)為前方 DMS 的 threshold，大於 350-400 視為目前前面有障礙物。

程式的運作方式是，當偵測到前方有障礙物時，就停止，並且轉 90 度，到平行障礙物的方向，以側邊 IR 面對障礙物。接著開始前行，直到側邊 IR 偵測到側邊已經沒有障礙物時，車輛停止，再轉 90 度重新回到面向終點的方向。接著前行，直到下一次遇到障礙物時，就重複以上動作。

惟在最後的深 v 障礙處，車輛偵測到前方障礙時，會先倒退，接著轉彎一定角度，前進直到撞上終點處的最後一道牆時，再以上述方式尋找出口。

硬體設計



長麵包版:盡量接近平行前後方向擺置，可以確保配重平衡。

側邊 IR sensor:用於感測側邊是否有障礙物，以決定轉彎時機。

前 DMS sensor:用於偵測停止前進時機。

避障輪:能夠校正轉彎幅度，撞牆時自動轉變方向沿牆走。

前 IR sensor x3:用於偵測黑線。與側邊 IR 共用。

1. 採用 6 科 1.5v 鹼性電池驅動，沒有使用其他形式電力。
2. 全部使用課堂提供之感應器，沒有另外購買。
3. 使用的 ATmega 328p 在每天使用的第三個小時後，出現 timer 故障的情形。
4. 採用課堂提供的車輛平台，馬達經過更換，確保是一模一樣的馬達。

問題意識及心路歷程

硬體比軟體可靠

在 maze runner 部分，一開始花了很多時間在調整每次轉彎的 delay 值，但車輛每次走的路線都不太一樣，很難用一個固定的轉彎量來寫。後來裝上避障輪，如果轉的不準確，在撞到障礙物時車輛可以自行校正回來，就省去了一個一個轉彎都得一直調整的問題。這給我很大的啟發是，可以用好的硬體設計來解決的問題，就盡量不要用軟體來彌補。硬體的運作看得很清楚，可以直接用手調整，有好的硬體，會讓軟體的運作順暢很多。

類似的例子，也發生在 line tracker 的車輛配重等方面，把配重固定好，每次轉彎擺動才會穩定。又或者是 maze runner 側邊 IR 的裝設，用比較外凸的位置來確保 IR 會靠的障礙物近，不容易偵測錯誤，就能減少很多調整 threshold 的時間。

盡量自動化

自動化的程度，主要是做 maze runner 時我很猶豫的部分。一開其實我想要透過固定的路線，讓車輛直接走到最側邊，跳過第一排障礙，減少遇到障礙物的次數。但是車輛每次走的角度都會有點不同，很容易在長直進路線歪掉，撞到障礙物或牆壁。後來我決定改回來，在前半部全面自動化，讓車輛自己判斷何時停止、何時轉彎、何時繞過障礙物等等。改變策略之後，只經過幾次調整，車輛就幾乎都能順利的到達第二個 check point。

Maze runner 的後半部分(check point 2 之後)，我也曾經試過全部寫死，但嘗試幾次之後，發現結果非常不 constant，所以後來我還是讓車輛感測器介入，自行決定停止的時機、轉彎的時機、還有最後找到出口的時機等等。也是在經過這些調整後，我的車輛才在測試時成功走完迷宮。

結果不是我能掌控的

花了很多時間，經過很多調整，不過最後我們的分數根本相當於我們甚麼都沒做。我覺得在這裡繼續花篇幅闡釋我最後一天如何失敗，沒有太大的意義，就算我成功說服你我真的在測試時都成功、軟硬體設計真的都可行、最後一天是一連串的衰事毀滅了我，你也沒辦法幫我加分。反正分數不是全部，但我只是想說，我真的很難過、很難過。