

行政院國家科學委員會補助
大專學生研究計畫研究成果報告

* ***** *
* 計 畫 : 結合 Android 裝置實作 Lego 機器人最短路徑記憶之探 *
* 名 稱 討 *
* ***** *

執行計畫學生： 張育誠
學生計畫編號： NSC 101-2815-C-275-003-E
研 究 期 間： 101 年 07 月 01 日至 102 年 02 月 28 日止，計 8 個月
指 導 教 授： 張安成

處理方式： 本計畫可公開查詢

執 行 單 位： 嶺東科技大學資訊科技系

中華民國 102 年 03 月 11 日

結合 Android 裝置實作 Lego 機器人最短路徑記憶之探討

學生：張育誠 指導老師：張安成

資訊科技系

摘要

本計畫目的在於能藉由智慧型手機控制機器人應用於危險任務或運輸以達到節省人力與時間之目的。目前國內許多研究 Lego 機器人者，多為應用於醫療、運動之控制與一般的避障或路徑的自走；主要為實作一個利用 HTC (Android 系統)手機結合 NXT Lego 機器人配合其感應器(Sensor)與記憶之功能並透過演算法判別最短路徑，記憶後以最短路徑走到目的地。在研究過程中將使用光感應器(Light Sensor)於 Lego 機器人上，經智慧型手機操控行走路徑並記憶路徑後，採用演算法判別其最短路徑，最後，Lego 機器人將以最短路徑走至目的地，實現利用智慧型手機來達到節省人力與時間，進而完成危險或緊急之任務。

一、前言

這幾年非常熱門於全球衛星定位系統 (Global Positioning System, GPS)，因此手持式導航裝置 (Portable Navigation Device, PND) 商品也日漸增加，許多手機也開始講究內建 GPS 功能，為为了提高準確性，因此 GPS 手機再加入輔助全球衛星定位系統 (Assisted Global Positioning System, AGPS) 功能，但手機要進行導航工作還必須搭配導航軟體和圖資[1]。由於這些皆需路徑之判斷，而本計畫就是以最短路徑為探討，因此加入自走車的技術來探討最短路徑之實驗。

隨著科技日新月異不斷的發展與進步，早期多以大量勞力完成的工作，如今為了工作效率或是要降低人事成本，已經慢慢被自動化機器取代。因此許多功能特殊的機器人，也因為特定需求正慢慢在發展中，而自走車是機器人系統中一項已經被廣泛應用的系統，一般包含了自動控制、感測元件、通訊系統與電機機械等結構。輪型機器人比起仿人型機器人或者仿動物型機器人，控制上來的簡易，如今也是應用最廣的一種機器人[2]。輪型機器人在追蹤路徑時感測器就如同機器人的眼睛，可提供機器人周圍環境的資訊，讓機器人即時依照周圍環境的不同做出正確判斷，因此，感測器提供的資訊完整度、正確性與即時性，將是機器人能否正確追蹤路徑的一大關鍵[2]。而此研究即為使用輪型的 Lego 機器人來實驗。

二、研究背景

樂高 (LEGO) 機械人 Mindstorms NXT 是樂高集團所製造的下一代的可程式即可組合機械人玩具。於 2006 年 9 月上旬推出樂高公司和美國麻省理工學院共同開發的機械人組件新款「教育用 LEGO Mindstorms NXT」[3]。採用了 32 位元的 ARM7 微處理器、256 Kbytes FLASH memory、64 Kbytes RAM，並且附有觸碰、聲音、光源、超音波等 4 種感應器，且搭配 3 個伺服馬達；通訊方式則有藍牙無線傳輸(Bluetooth Class II V2.0 與 USB 12 Mbps)可讓使用者做出透過藍牙裝置(如 PDA 或手機)控制的機器人[4]。

Lego 機器人宛如一個盲人，而感測器就像它的眼睛、耳朵等感官系統般，用來輔助 Lego 機器人的視覺、聽覺、觸覺等，而 NXT Lego 機器人基本上擁有的感應器以感官系統來區別，觸覺好比就是觸碰感應器，視覺為超音波感應器、光感應器，聽覺就如聲音感應器，而伺服馬達彷彿機器人的運動控制，可驅動機器人移動。

光感應器用來測量來自一個方向的光源強度。前方兩燈泡：發出紅光以增加反射光線之強度及接收反射回來的光。利用光源作為感應的輸入裝置，主要用作分辨顏色明暗，藉由偵測到的數值來控制樂高機器人的動作，可以利用紅外線反射光讀取數值辨識顏色深淺，顏色深淺判斷數值為最暗(0)~最亮(100)，其缺點為無法分辨相近的顏色如紅色和橘紅色[4]；本計畫使用此感應器來偵測路況並記憶路徑。

三、研究動機及目的

現代社會為人手一機之時代，許多人已開始使用智慧型手機，而目前市場上多數智慧型手機皆能支援 Android 系統。Android 為開放式的系統，而且為市場佔有率最高的之一，因為自由度高、支援多款手機，加上 Android 的應用程式通常以 Java 資料庫元為基礎編寫，恰好與本計畫之開發工具 Java 語言相符；許多有關自走車研究的專題多為一般的運動控制或迷宮等競賽之用，而現今生活中許多使用於行車之用的導航設備以及 Google Map 並非皆以最短路徑為規劃。以此為靈感，本計畫將運用目前市場上支援最多手機的 Android 系統的智慧型手機與 Lego 機器人做結合並以路徑導航之觀念來探討自動記憶路徑且計算最短路徑的自走車，未來能結合人們想去曾經去過的地點時可經此設備來達到節省時間與導航的不確定性。

四、系統架構與設計

本計畫之系統架構圖，如圖 4.1 所示。將程式寫入 NXT 主機後，利用 Android 手機藍牙控制 Lego 機器人 搜尋迷宮所有路徑，並將走過之路徑儲存到手機端，經由演

算法運算後手機端再把運算結果顯示到 Android 手機，使用者可經由 Android 手機中的介面命令 Lego 機器人自走迷宮最短路徑。

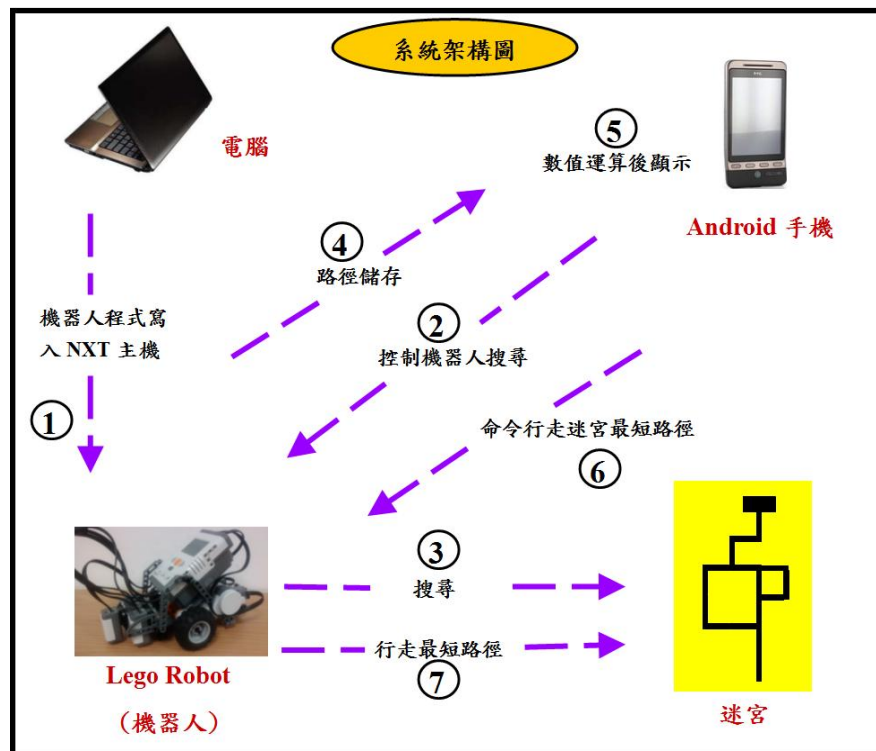


圖 4.1 系統架構圖

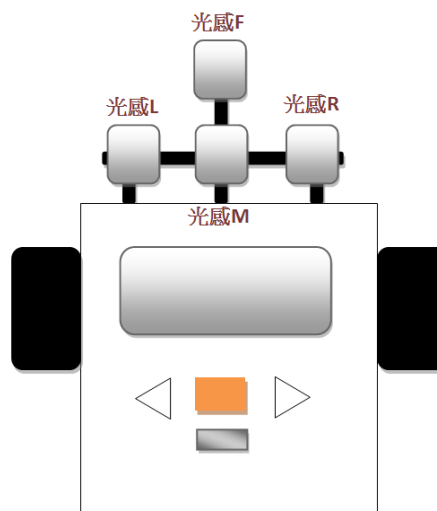


圖 4.2 光感位置擺設圖

(一) Lego 機器人 架構與感應器用途

本計畫使用 4 顆光感應器來實現循跡與判斷之用，中間光感 M 為負責循跡黑線，左右光感 L、R 負責判斷左右轉與校正，前面光感 F 將用來判斷前面是否有節點，進而使後面三顆光感來行走正確的路徑，如圖 4.2 為光感位置擺設圖。

光感 F 主要應用於 Lego 機器人自走時，用來在第一時間遇到路徑節點判斷是否有節點存在，來導引 Lego 機器人進一步的往前判斷該路況屬於何種路況，來判斷是否要前進、左轉以及右轉。

(二) 路徑圖

Lego 機器人行走道路之路徑圖，如圖 4.3 所示，其中黑色膠帶部分為 Lego 機器人要走的路徑，白色部分為路徑的節點。利用光感應器部分感測光的明暗度，在此，我使用黑色膠帶來使 Lego 機器人作判斷，Lego 機器人會照著黑色膠帶部分行走。由於從起點(IN)至終點(OUT)有多條路徑，因此先利用 Android 手機操控 Lego 機器人行走路徑圖中所有路徑並紀錄路徑中的節點後，經由記憶功能可使 Lego 機器人記憶走過之路徑，並透過演算法來實現，最後能使 Lego 機器人自走路徑圖中最短路徑到終點(OUT)。

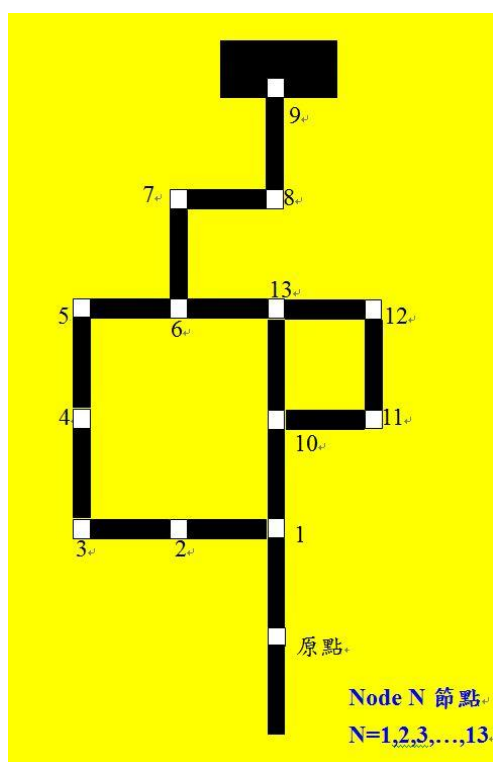


圖 4.3 路徑節點圖

(三) 光感之感測路徑與路口狀況

Lego 機器人走到節點時所會遇到的主要 6 種路況與終點路況，合計共 7 種路況，這 7 種路況的狀態如圖 4.4 所示；光感應器感測路口之狀態，如圖 4.5 所示，其中紅點表示為光感應器有感應到黑線的狀態，白點則為黑線以外未被感測到部分的光感應器之狀態。

本計畫使用路況 1 到路況 7 來記錄每條路徑的節點，由於每個節點的路況皆不同，因此，可使 Lego 機器人 在記憶路徑時能夠判斷哪些節點已經走過或需再重複走的路徑。在路徑圖中，共有三條可從起點(IN)到達終點(OUT)，且每條路徑皆有節點，而節點與節點的距離設計成均有相同之距離。















一般路況	光感測狀態
1. 左岔路 	
2. 右岔路 	
3. 右轉路 	
4. 左轉路 	
5. T 字路 	
6. 直路 	
終點路況	光感測狀態
7. 終點 	

圖 4.4 節點路況

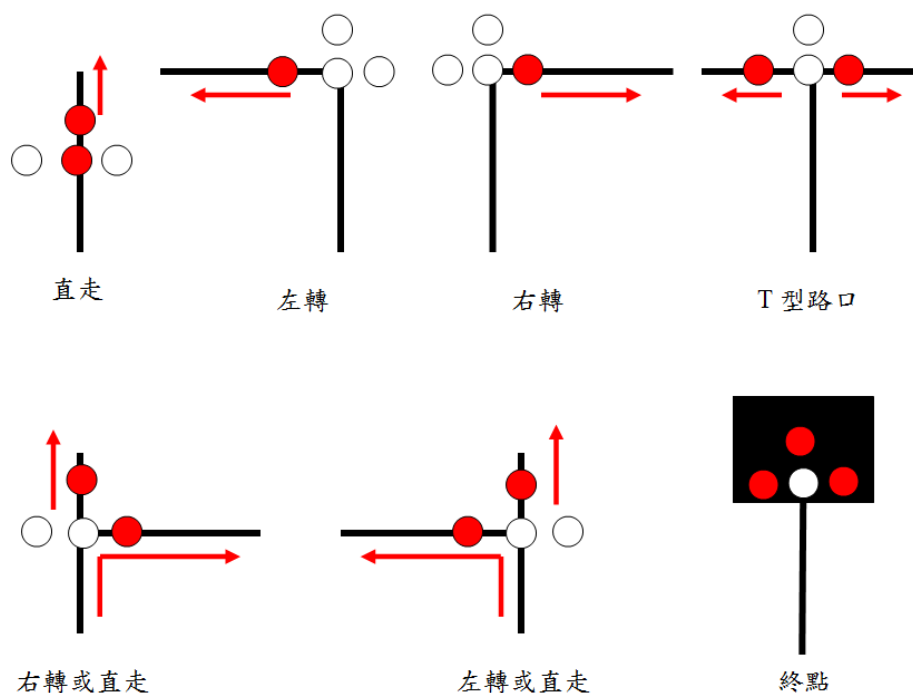


圖 4.5 光感測之路口狀態圖

五、系統實作

本計畫 Lego 機器人的實體圖，如圖 5.1，感應器部分共有 4 顆光感應器，分別為 L (左光感)、M (中間光感)、R (右光感) 和 F (前光感)；以及圖 5.2 為的路徑圖的實際製作成果。

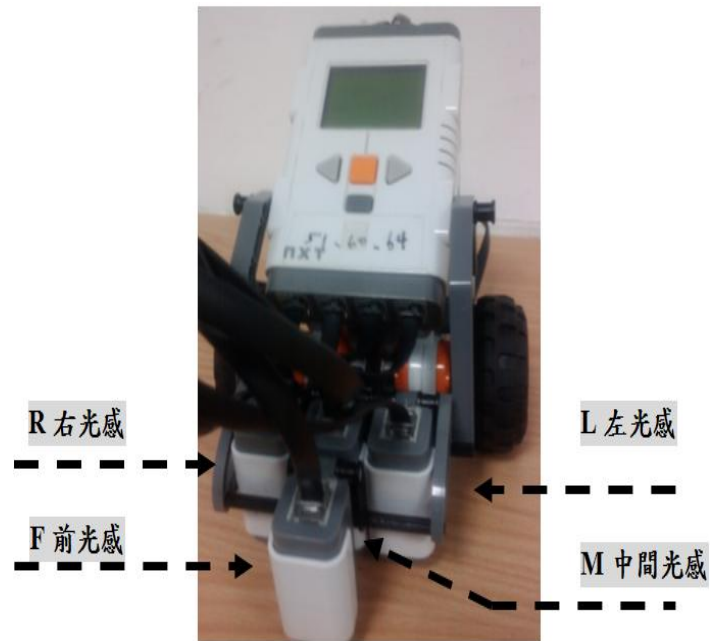


圖 5.1 Lego 機器人實體圖

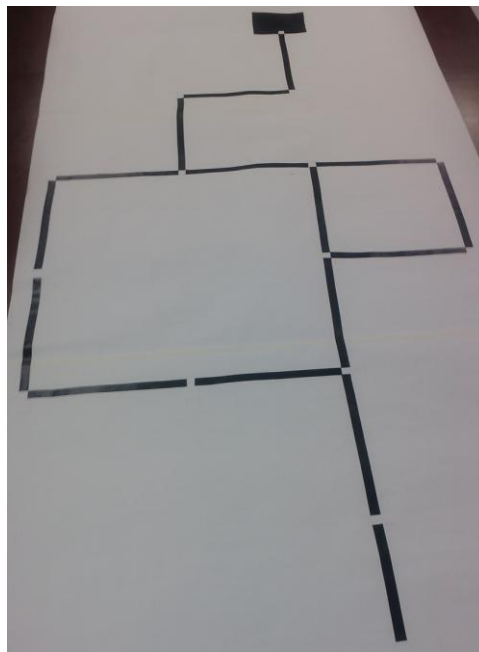


圖 5.2 路徑實體圖

(一) 路徑記憶與分析

1. 內部記憶體(Internal Storage)

資料儲存部分使用手機內部的記憶體來儲存資料，由於資料量小且為數字整數形式，因此作為本專題資料儲存之系統。

2. 路徑分析

以下實線部分分別為路徑 A、B、C 三條路徑，如圖 5.3 至 5.5 所示，從原點開始行走，路徑中共有 13 個節點，每條路徑節點的路況皆不同，三條路徑分析如下：

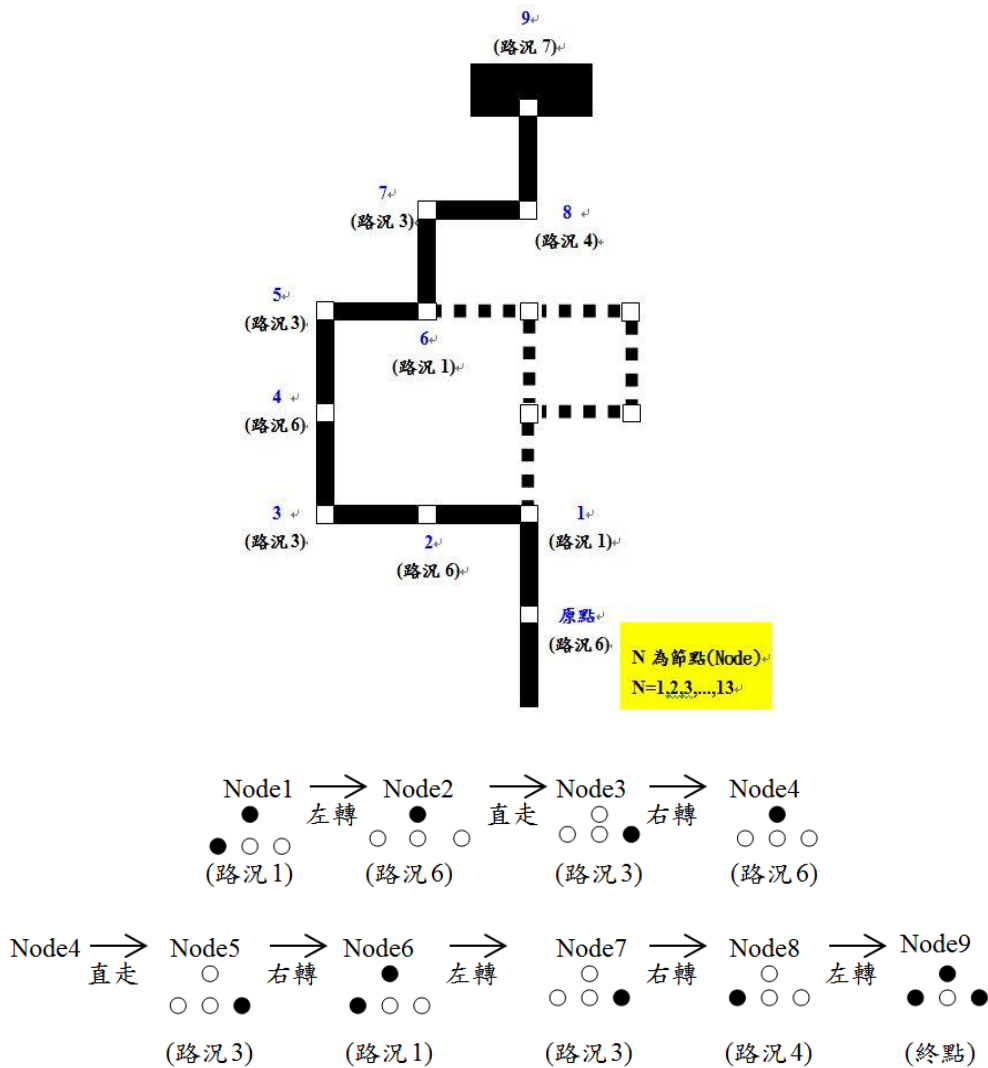


圖 5.3 路徑 A 分析

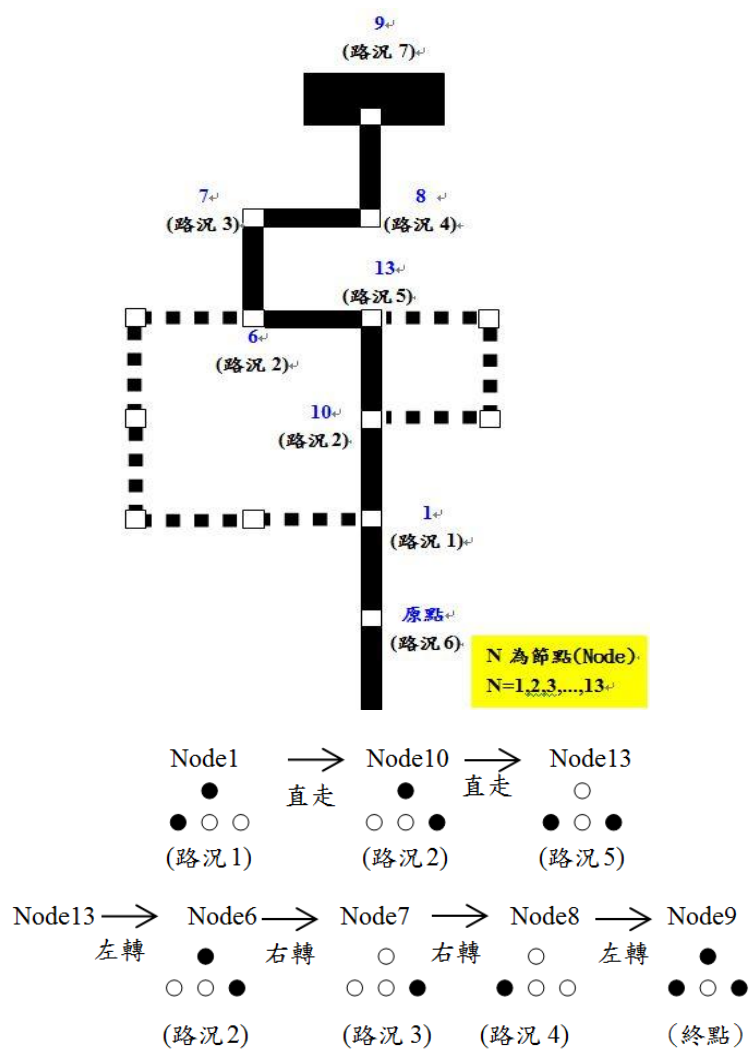
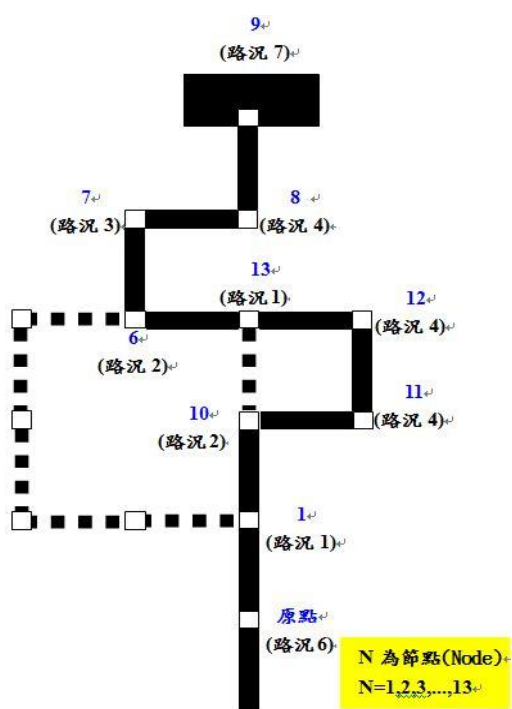
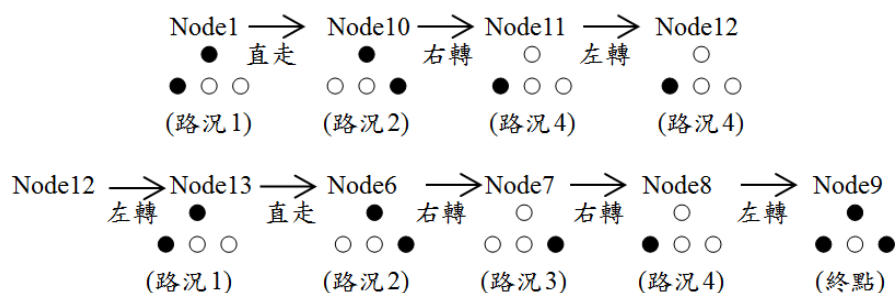


圖 5.4 路徑 B 分析





3. 最短路徑判斷

(二) 藍牙傳輸

如圖 5.6 與 5.7 分別為等待手機藍牙連結 Lego 機器人 與連結成功的畫面，當手機使用藍牙連結 Lego 機器人 時 Lego 機器人畫面會顯示 Waiting 為正在等待連結；若 Lego 機器人發出 "嗶" 一聲且畫面顯示「Connect success!」代表已連結成功。

圖 5.6 Lego 機器人等待連結畫面

圖 5.7 Lego 機器人連結成功畫面

岔路右轉」、「3.左轉」、「4.右轉」、「5. T 字路口」、「6.直線」等；每個路口都有節點，由光感 F 判斷是否有節點(白色色塊部分)，一旦光感 F 偵測到節點會直走到光感 M 在節點上時，將會開始判斷是否要進行左轉、右轉、或前進之動作，判斷完並執行程式，若為轉彎，則轉彎完會跳回直走程式繼續執行判斷，直到轉彎處再跳到轉彎副程式執行，不斷循環，直到到達終點為止，如圖 5.8 所示。

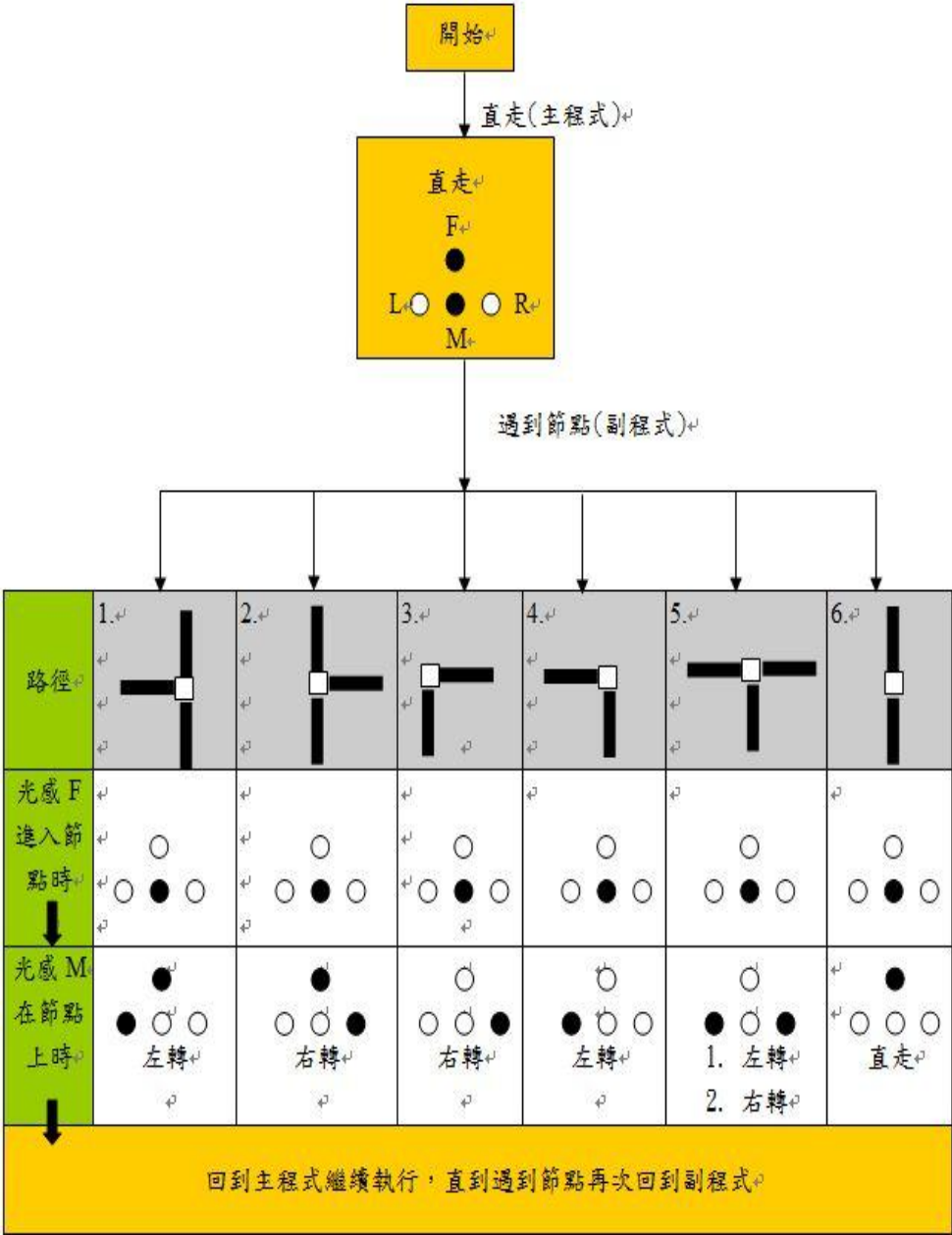


圖 5.8 Lego 機器人自走判斷

(四) 手機操作流程

本系統之手機操作流程圖如圖 5.9 所示，使用者，在手機上的介面分為開始、說明、離開三個主選單，按下開始鈕後將開啓功能選單，利用藍牙操控 Lego 機器人記錄路徑，記錄完畢後，切換到顯示結果介面，最後再使 Lego 機器人自動執行路徑結

果之動作。

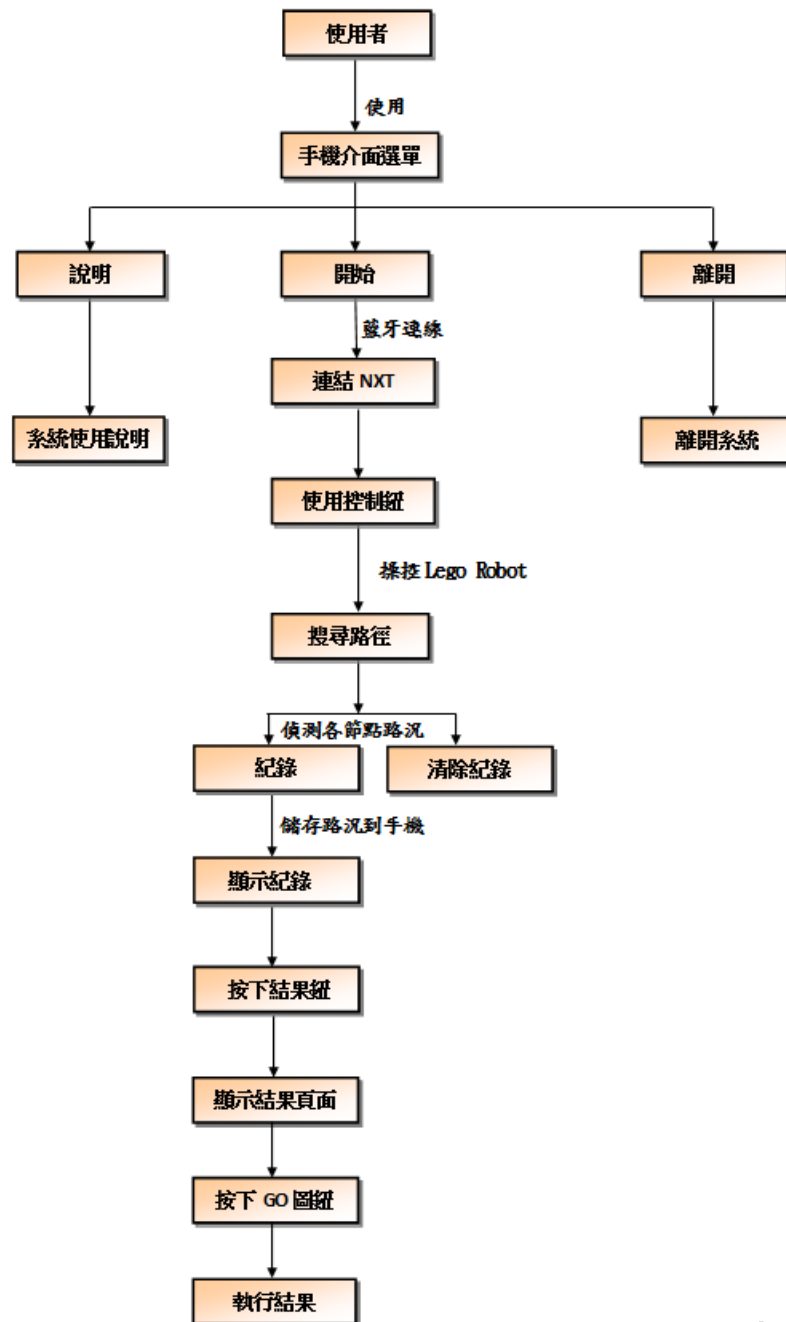


圖 5.9 手機操作流程圖

六、研究成果

(一) 手機介面與功能

手機介面的首頁畫面，如圖 6.1。介面選單中，開始按鈕為進入功能選單；若要了解此系統之操作方法則可按左下角之「說明鈕」，將會進入操作說明之頁面如圖 6.2 所示；如圖 6.3 若要離開系統則按右下「離開鈕」即可。



圖 6.1 系統首頁



圖 6.2 操作說明介面



圖 6.3 系統離開介面

1. 開啓藍牙

如圖 6.4 所示，為開啓藍牙裝置介面。由於手機與 Lego 機器人是利用藍牙來傳遞訊號，因此必須利用手機內部的藍牙裝置與 Lego 機器人進行連線後，才可操控 Lego 機器人；當使用者未開啓藍牙裝置時，按下系統首頁開始鈕，便會自動跳出是否開啓藍牙裝置的詢問訊息來提醒使用者開起來藍牙裝置，可避免使用者忘記開啓藍牙裝置。

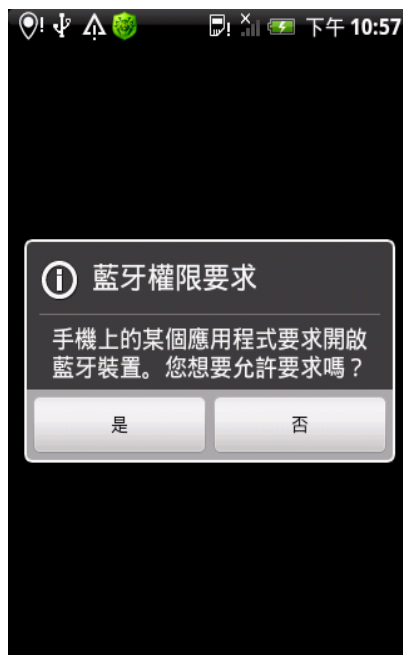


圖 6.4 開啓藍牙介面

2. 主要功能介面

主要功能介面，如圖 6.5 所示。首先必須輸入 NXT 主機的名稱，開啓 NXT 端程

式後，按下連結鈕等待 Lego 機器人連結成功，連結成功後便能開始操控 Lego 機器人來搜尋路徑，左右兩邊的紅色圖鈕能讓使用者在轉彎時能微調 Lego 機器人的方向，可方便校準 Lego 機器人在路徑上行走；當 Lego 機器人到達路徑節點時按下前進鈕後 Lego 機器人能進一步的偵測到路況並顯示出來，接著使用者可按下記錄圖鈕即可儲存到手機，若想清除紀錄也能按清除紀錄鈕來清除所有資料，在介面下方也同時能顯示剛剛記錄過的路徑路況，能讓使用者方查看；最下方的結果圖鈕為連結路徑搜尋結果的介面。



圖 6.5 主要功能介面

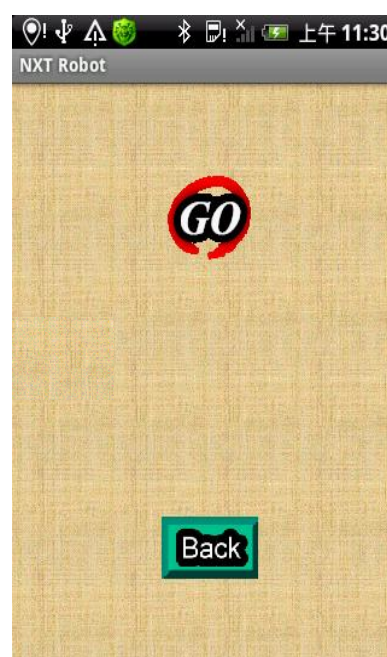


圖 6.6 結果執行介面

3. 結果執行介面

如圖 6.6 所示，結果執行介面，按下控制介面底下的結果圖鈕後將會跳至此介面，而此介面即為最短路徑執行介面，記錄完路徑後利用控制介面控制 Lego 機器人 回到路徑起點後，按下結果鈕到此介面，使用者按下 GO 執行圖鈕後 Lego 機器人 即會執行最短路徑自走到終點。

(二) 手機控制 Lego 機器人路徑記憶

如圖 6.7(a)、(b)圖，為使用 Android 手機操控 Lego 機器人來記憶路徑的路況的實作情形；首先開啓 NXT 端的本系統要控制的主程式，然後進入系統控制介面後，按下連結 NXT 的連結鈕，連結成功即能開始遙控 Lego 機器人來搜尋記憶路徑圖中每條路徑的節點，透過每一個節點的路況儲存到手機，最後執行時依照記錄的路況依序來

行走直到終點。



圖 6.7(a) 手機操控 Lego 機器人記憶路徑之路況



圖 6.7(b) 手機顯示 Lego 機器人記憶路徑之路況

(三) 執行結果.自走最短路徑

判斷完 3 條路徑後，按下結果執行介面中 GO 的圖鈕，即執行最短路徑的路徑 B 之自走。以下為自走路徑 B 之情況，如圖 6.8(a)至 6.8(f)所示

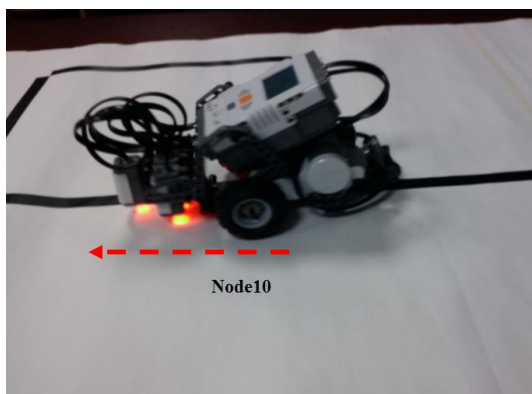


圖 6.8(a) Lego 機器人路徑 B 自走過程

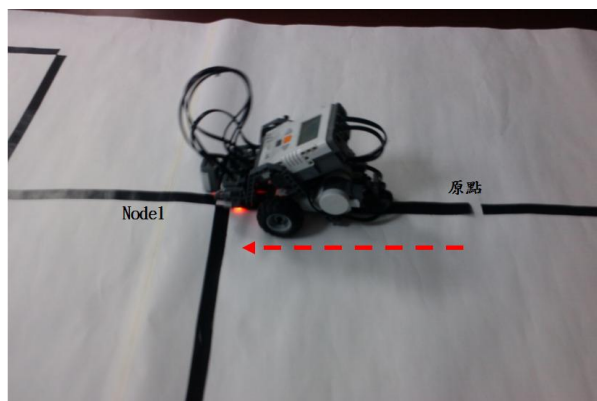


圖 6.8(b) Lego 機器人路徑 B 自走過程

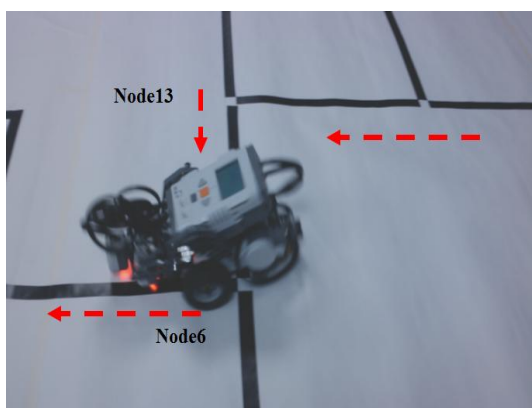


圖 6.8(c) Lego 機器人路徑 B 自走過程

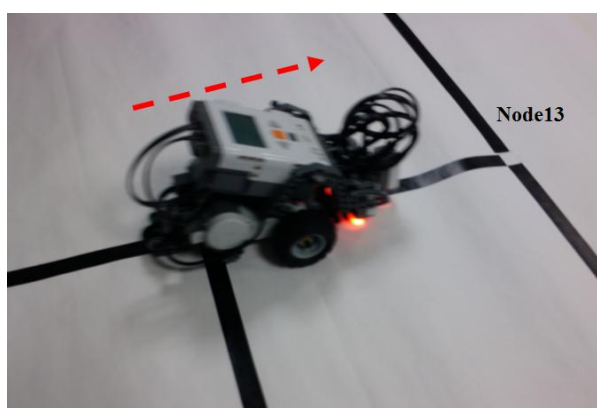


圖 6.8(d) Lego 機器人路徑 B 自走過程

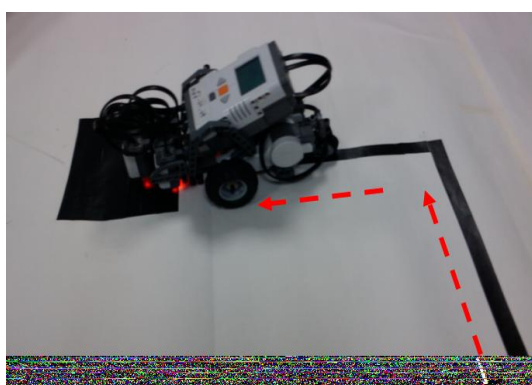


圖 6.8(e) Lego 機器人路徑 B 自走過程

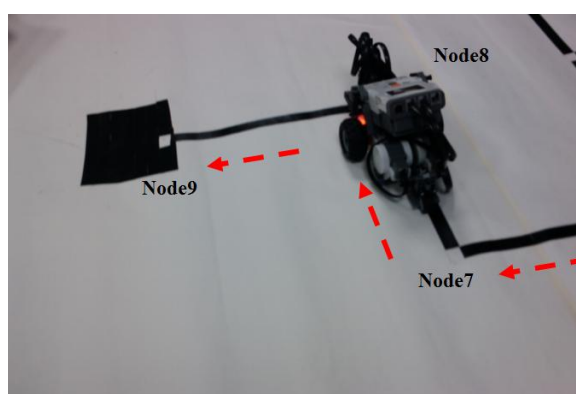


圖 6.8(f) Lego 機器人路徑 B 自走過程

六、結論與未來展望

智慧型手機已逐漸取代一般型手機之趨勢，而智慧型手機功能也愈來愈多；現代資訊化社會中，人們終日汲汲營營，常與時間競爭，尤其在上下班途中。在此，對於導航方面我利用現在許多人使用的 Android 智慧型手機與 Lego 機器人結合，並加入

了記憶之功能且模擬人們可利用手機操作交通工具記憶走過之路徑，並算出最短路徑，能達到省時之功用，未來對於緊急事件也能藉由此系統曾走過之路徑而快速到達。

本計畫主要以藍牙方式傳輸，未來我希望也能以 Wi-Fi 方式來傳輸，使本系統傳輸範圍擴大，並加入定位功能，如此一來，更能使此系統功能更加精準，並能改善人們的生活。

參考文獻

- [1] <http://www.libertytimes.com.tw/2008/new/aug/29/today.style9.htm> (自由電子報 AG PS 快速定位)
- [2] 李孟軒，輪型機器人之路徑追蹤與避障，國立中央大學，電機工程學系，碩士論文，民國 100 年
- [3] http://zh.wikipedia.org/zh.hk/%E6%A8%82%E9%AB%98Mindstorms_NXT (維基百科)
- [4] CAVE 教育團隊 曾吉弘/林祥瑞/Juan Antonio，碁峯出版社，機器人程式設計與實作.使用 Java，2010
- [5] <http://lab.cavedu.com/lejos> (leJOS . 樂高機器人的 Java 語言的 CAVE 實驗室)
- [6] 魏立和、林泓勳，行動智慧型機器人照護及娛樂系統，嶺東科技大學，專題報告，民國 97 年
- [7] 張唐禎，應用 LEGO Mindstorms NXT 於競賽機器人之動態路徑規劃，義守大學，電子工程研究所，碩士論文，民國 98 年
- [8] 徐順興，以圖控軟體學習正規程式語言之研究，大華技術學院，機電工程研究所，碩士論文，民國 100 年
- [9] <http://jcr1.jcps.kh.edu.tw/lifetype/post/3/10657>(資訊學習園)
- [10] <http://zh.wikipedia.org/wiki/Wikipedia:%E9%A6%96%E9%A1%B5> (維基百科)
- [11] <http://ozzysun.blogspot.tw/2010/11/android.html>(程式搖滾.Android 資料儲存)
- [12] http://neuron.csie.ntust.edu.tw/homework/93/csie_introduction/homework1/b9315032/BlueTooth.htm(藍牙科技)
- [13] 楊嘉隆，具搜尋最短路徑演算法之線迷宮鼠實作，龍華科技大學，電子工程研究所，碩士論文，民國 100 年