

# 樂高保齡球機器人

專題編號：104-CSIE-S024

執行期限：103 年第 1 學期至 104 年第 1 學期

指導教授：郭忠義

專題參與人員： 101590316 陳宣佑  
101590341 江佳勳  
101590342 葉偉倫

## 一、摘要

本專題使用 leJOS 開發環境[1]，搭配 OpenCV 的影像處理功能，實做出能夠偵測周遭環境，達到自動取球、投球，並返回起始位置的”樂高保齡球機器人”。

**關鍵詞：**leJOS、OpenCV、TETRIS、LegoMindstormsNXT。

## 二、緣由與目的

無人載具的技術在這個時代越來越受重視，上至飛機下至汽車，這項技術可以適用的地方非常廣泛。在此背景下，機器人的製作及設計已成為重要的課題。

本專題想設計保齡球機器人，參加 IRHOCS 競賽，並將此作為專題的內容。

## 三、研究範圍

製作可以符合比賽的資格及具有自動行動功能(判斷出界範圍、取球、偵測球瓶、擊球、判斷球瓶是否全倒等)的機器人。

## 四、使用技術方法

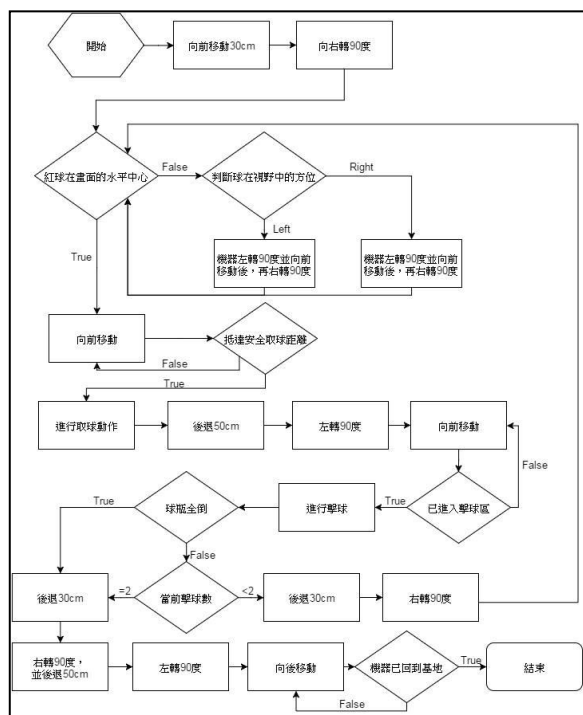
(一)、影像處理：利用影像處理的技術精確取得球及球瓶的位置，並能偵測比賽場地黑線避免出界。

(二)、藍芽傳輸：利用藍芽使 NXT 及電腦進行連接，讓電腦處理影像的運算後再利用藍芽傳輸將指令傳給 NXT 進行運算。

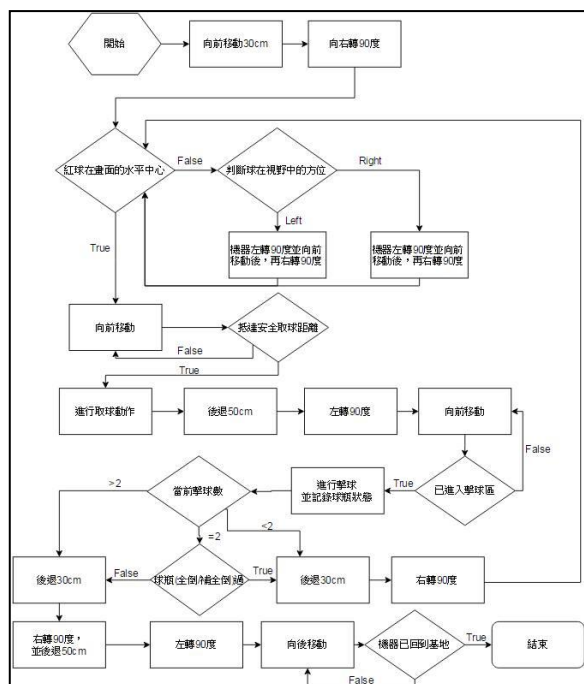
## 五、架構流程

因為局與局間有準備時間可調整程式，因此主要將程式分為兩部分，第一部分為初賽的前 1 局與決賽的前 2 局的程式(最多投兩球，全倒後回基地)及第二部分的初賽的第 2 局與決賽的第 3 局的程式(最多可投到三球)。而比賽有三種模式，第一種為正常保齡球賽(初賽第一回合、決賽第一局)、第二種為加分球瓶的特殊擺放方式(初賽第二回合、決賽第二局)及第三種的障礙球瓶(決賽第三局)，基本上我們的機器人會照著流程走，會改變的部分除了電腦端偵測方式需要改變以外，擊球點的位置(流程圖是移至場地中央)也會作些許更改。

本專題利用了 Color Sensors(顏色感應器)進行地面場地界線的判定，NXT 進行移動的時候可以利用 Color Sensors 的判斷成功到達取球區以及回到基地、Ultrasonic Sensors(超聲波感應器)進行取球距離的判定，雖然攝影機抓地到球的位置，但是機構取球的距離有限，所以利用了 Ultrasonic Sensors 能使取球更加穩定。剩餘的部分使用攝影機作為機器人的”眼睛”，因為使用了筆記型電腦做影像處理的運算，因此不需要在 NXT 上面寫太強大的影像功能，可以更精簡 NXT 主機所燒錄的程式，主要功能為尋找球以及球瓶的位置，和其他的 Sensors 搭配使用能使機器人進行一連串工作時更穩定。



圖(一)、前局流程圖



圖(二)、終局流程圖

## 六、IRHOCS 競賽規則

IRHOCS 競賽簡要規則，比賽開始後機器人於基地出發，至球座上將球取出，之後移動至投球區(以黑色電工膠帶作為分區線)，若超出分區線則視為犯規。保齡球計分規則與一般正常保齡球相同。詳細

規則請參照參考文獻[2]。

## 七、開發工具說明

### (一)、Eclipse

Eclipse 為 Open Source Community 所創建的開發軟體，被廣泛地使用在許多不同的領域。較為許多人使用的是 Java 應用程式與 Android app 的開發。大多數人知道這個開發工具是 Java 的 IDE。目前在全世界的 Java 開發環境市場已經超過 50%。

### (二)、leJOS

leJOS NXJ 是可執行於樂高 NXT 主機上的 Java Virtual Machine。leJOS 為 Java 的擴充套件，所以必須先安裝 JDK，之後可使用 Eclipse 開始進行開發。leJOS 可以在任何作業系統上進行開發，且功能及函式庫也很完善，如 TETRIS 套件、一般 Sensor、藍芽傳輸功能等均可以使用，但是必須要有一些 Java 語言的基礎。

## 八、實驗結果

在保齡球機器人的影像處理部分使用了 HSV 空間用以尋找顏色，可以利用 H(色相)範圍進行較準確的判斷。之後使用高斯濾波(模糊化)及 Sobel 使影像邊緣化，讓影像雜訊變少，之後再利用 HoughCircles (霍夫圓型偵測)抓出球的位置，這麼做能夠比較準確的抓到比賽紅球的位置。而判斷球瓶全倒的部分則是利用投球前後幾秒內去進行白色高度(白色球瓶高度)的位置判斷，高度不變代表沒有全倒；高度變低則為全倒。

## 九、結論

在進行機器人設計中也會碰到棘手

的情況，如開發軟體安裝完卻無法正常與 NXT 連接、NXTCam 無法成功抓取畫面、使用攝影機做影像辨識時光線對於顏色的影響、以及機構硬體元件數量不足的情況，以致於需要更換開發軟體、元件、影像偵測方式以及改變機構等。

除了上述問題之外，機器人需要很多的邏輯運算才能運作，而我們只是做”保齡球機器人”就用到了很多的邏輯運算及影像辨識。我們可以從此次專題中了解到機器人的製作與設計需要花很多的時間及心力，而希望我們此次的經驗能夠為未來打下一個基礎，往更高階的技術邁進。

#### 十、參考文獻

[1]以 Java 語言控制 LEGO NXT 機器人，實作「電腦與問題解決」教學單元之專題研究 - 國立台中第一高級中學 林奇鋒：

<http://icerc.tnssh.tn.edu.tw/download/epaper/epaper70/20120229.pdf>

[2]機器人保齡球賽規則：

[http://www.iceira.ntu.edu.tw/uploads/IRHOCS\\_2015\\_-\\_Robot\\_Bowling\\_rules.pdf](http://www.iceira.ntu.edu.tw/uploads/IRHOCS_2015_-_Robot_Bowling_rules.pdf)

[3]謝碧景-延伸學習：NXC > 感應器 (sensor)應用：

<https://goo.gl/0IJrUV>

[4]基峰電子書-藍芽遙控車：

<http://epaper.gotop.com.tw/pdf/acl010400.pdf>

[5]Color Detection & Object Tracking：

<http://opencv-srf.blogspot.tw/2010/09/object-detection-using-color-seperation.html>

[6]Object Detection：

<https://solderspot.wordpress.com/2014/10/18/using-opencv-for-simple-object-detection/>

[7]Blob Detection：

<http://www.learnopencv.com/blob-detection-using-opencv-python-c/>

[8]霍夫圓形偵測 HoughCircles：

<http://ccw1986.blogspot.tw/2012/12/opencv-houghcircles.html>

[9]圖像處理-Hough 線變換和圓變換：

[http://rockyhotshow.blogspot.tw/2011/10/hough\\_07.html](http://rockyhotshow.blogspot.tw/2011/10/hough_07.html)

[10]Tracking a ball with Java/OpenCV：

<http://cell0907.blogspot.tw/2013/07/tracking-ball-with-javaopencv.html>

[11]色彩表：

<http://www.ebaomonthly.com/window/photo/lesson/colorList.htm>