

推薦序

PREFACE

為了培養機器人產業的優秀人才，近年來國內教育環境對於機器人教育的重視程度是與日俱增。從國小國中到大專院校，各種類型的比賽可說是非常豐富，學生可以從比賽中學到許多寶貴的經驗，教育單位與家長們也逐漸體認到動手做課程之重要性。

淡江大學電機系機器人研發團隊秉持著「動手做，做中學」的精神，積極參與國際與國內的各項機器人競賽，我們深刻了解機器人的設計與製作是需要整合資訊、機械、電機與控制等跨領域的技術。因此，做為一位機器人教育的推動者，本人一直在思考如何建立一個機器人的軟硬體教學環境。LabVIEW 是一個很容易上手的圖形化程式語言，是針對專業量測與儀器控制所發展的軟體，並且為產業界所普遍使用。

本人於 2009 年翻譯出版《LabVIEW 圖控程式與 LEGO MINDSTORMS NXT》之目的在於幫助 LEGOMINDSTORMS NXT 的使用者可以快速學習 LabVIEW。由於 NI 的機器人教育願景是希望建立一個從小學到業界的持續學習環境以及提供一個整合式軟硬體平台來快速整合不同領域技術，因此本人在大一的「基礎電機實驗」課程中規劃設計一份 LEGO 機器人的實驗教材，LEGO 積木組裝的方式與圖形化設計系統讓學生可以在每週 3 小時的實驗課程中學習如何設計與製作一台 LEGO 機器人。此外，藉由大一的「圖控程式設計」課程，讓學生認識與體驗 LabVIEW 圖形化編程環境。並且在高年級的「機器人實驗」課程中，讓學生學習如何將 NI 所發展的嵌入式控制平台 CompactRIO 做為機器人的控制平台。這些規劃設計讓電機系的學生可以持續的學習機器人的各種概念與工具，進而在「專題實驗」課程中可以設計製作一台機器人。

這些課程的規劃設計是需要完整且合適的教學環境與教材。本人樂見國內有愈來愈多年輕朋友投入相關教材的開發工作。本書係由國家發展儀器股份有限公司吳維翰先生與 CAVEDU 教育團隊曾吉弘先生等人所合作編寫之高階機器人程式設計教材，相信本書的出版將可以讓學生可以更快速了解 LabVIEW 與其相關的應用。

翁慶昌

淡江大學電機系暨機器人研究所教授

近年來科技發展之趨勢，均朝向互動化、人性化、直覺化發展，圖形化程式開發平台逐漸成為程式設計及開發之主流，而 LabVIEW 不僅是圖形化程式開發平台之始祖，一直以來，也幾乎是圖形化程式開發平台的代名詞。LabVIEW 初期以虛擬儀表的概念，讓許多自動化工程師可輕易透過它設計直覺且可直接控制之人機介面，其易懂簡單的圖形化程式開發介面，不僅縮短程式開發速度，也方便後續維護及更新。樂高（LEGO）MINDSTORMS NXT-G 就是以 LabVIEW 為核心架構之圖形化程式開發介面，許多初學者甚至是國小學童，經過訓練便可輕易上手，因此讓學習及設計程式變得輕鬆有趣，進而提昇許多學子動手做、著手設計程式及完成科學實驗之動力。這些年機器人教育之所以能快速發展且成功推廣，樂高與其圖形化程式介面可說是最大功勞之一。

由於使用者越來越多，許多學習者或玩家開始不滿足於簡單易懂的開發介面，他們開始探究更高階的圖形化程式開發平台，如 LabVIEW 或微軟的 MSRDS（MicroSoft Robotics Developer Studio）。然而，雖然 MSRDS 來勢洶洶但仍動搖不了 LabVIEW 的主流地位。NI（國家發展儀器股份有限公司）在 2009 年推出 LabVIEW 之 LEGO MINDSTORMS NXT 模組，提供 LEGO 愛好者可以將樂高機器人與 LabVIEW 提供之強大功能整合，開發及實現更快速、功能更強及更完整之程式。本書作者以完整功能解說、設計技巧、及許多實用應用實例，清楚鋪陳讓讀者得以輕鬆進入樂高機器人與 LabVIEW 結合之的程式世界。

近年來國內越來越多機器人競賽的舉辦，有效推動了機器人學習之風氣。由學習到參與競賽，再由競賽經驗進一步學習，似乎已成為良性循環，促進且養成參與者之技術及執行專題之實驗能力。本人參與競賽推動工作多年，看到學子在參與競賽的過程中，全力以赴且團隊合作，達成甚至超越預期之目標，這些經驗相信是一般學科教育無法養成的。

本書提供有系統的學習方法，讓許多對機器人有興趣的學子，能更進一步透過書中專題案例，衍生創意想法及實現較複雜的機器人程式開發。本人相信且期待，許多學子透過閱讀本書並動手實作，將使得機器人發展增添新的動力，也奠定未來投入機器人產業之基礎。

謝銘原

於南台科技大學機器人研究中心

推薦序

PREFACE

深耕於台灣機器人教育界的 CAVEDU 團隊，從兒童使用的樂高機器人 EV3，高中、大專生使用的 KNR 機器人平台、myRIO，甚至工業界的專業機器人系統建置等，皆是 CAVEDU 團隊所服務的對象。多年來致力於推廣機器人教育與技術的過程中，CAVEDU 教育團隊累積了在機器人領域上的扎實經驗、以及專業的教學方法，更難能可貴的是，CAVEDU 將機器人課程帶入工程教育，讓孩子們透過實際動手的過程，毫無負擔的感受科學與工程所帶來的樂趣，進而提升台灣整體的科學素養。這種精神正與 NI 近年來宣導的「Do Engineering」(實做工程教學)理念不謀而合。

此次欣聞由 CAVEDU 團隊所著作的《LabVIEW 高階機器人教戰手冊》即將進入第二刷，這對身為同樣推廣實做工程的 NI 來說，不啻是個振奮人心的好消息。藉由深入淺出的教學方式，讓即使是非理工背景的讀者，只要對機器人有興趣，也能輕鬆透過本書的教學，建立起機器人技術及 LabVIEW 程式撰寫的基礎。值得一提的是，CAVEDU 教育團隊在書中不藏私的慷慨相授，結合了多年來教學上所觀察到使用者盲點，並加上自身豐富的使用經驗，帶領讀者一步步使用 NI LabVIEW 圖形化程式語言。

由於機器人是一整合性的學科，機器人系統的建構者往往需要同時掌握機械機構、感測及資料擷取、馬達、視覺檢測等多項技術，因此 NI 提供的軟硬體整合平台式解決方案，得以幫助使用者以最快的速度建構出機器人系統，讓讀者將時間放在機器人的設計與創新上，而不是花時間在底層軟硬體設定跟連結。這點從 Lego EV3 與 NI LabVIEW 的結合，即可得到最好的體現。此外本書也結合了 NI LabVIEW 與機電方面的延伸應用案例，讓讀者可以將自本書所學的軟硬體技術，延伸至未來的實作專題，研究、甚至工作中。

多年來 NI 持續投入 R&D 資源，為業界設計出方便好用的 NXT 與 EV3 函式 API，希望藉此幫助學習建構機器人的入門者，同時也期待機器人教育能因此讓學生們以輕鬆、直覺的方式，進入機器人與工程實作領域。在此，我要特別感謝 CAVEDU 教育團隊，除了充分使用上述 API 在他們所開設的機器人推廣課程外，更用心設計出實用且豐富的文化教學內容，不僅落實 NI 圖形化系統設計的理念、更造福廣大的台灣機器人愛好者。

台灣工程的未來，需要我們一起努力推動，期待所有讀者能因為《LabVIEW 高階機器人教戰手冊》這本書，讓機器人建構功力更上層樓，進而在工程領域，激發出更多元、更創新的思維。再次感謝 CAVEDU 團隊的努力，容我邀請大家，一起藉由本書進入有趣的實做工程世界~



國家發展儀器股份有限公司 (National Instruments, NI)
大中華區行銷總經理郭皇志

Kinect 體感互動

15-1 前言

15-1-1 Kinect 介紹

Kinect 是由微軟公司針對 XBOX、PC 所開發的體感偵測裝置。打破以往僅能透過搖桿控制遊樂器的限制，Kinect 能透過手勢及語音辨識達到控制器的效果，讓玩家能夠深入其境。除了遊玩之外，Microsoft 在 2011 年發佈了 Kinect SDK，讓開發人員能夠利用 Kinect 做出更多有趣的體感應用。Kinect 已於 2014 年推了 2.0 版，本書範例適用於各版本的 Kinect。



圖 15-1 Kinect 體感偵測裝置

Kinect 有 3 個鏡頭：1 個 RGB 彩色影像鏡頭、2 個深度影像鏡頭，每秒能有 30 個畫格（30fps），透過 2 個紅外線影像能夠知道 Kinect 前方的場景深度，如果恰巧有完整的人入鏡的話，Kinect 將會取得完整人物的骨架資料。



圖 15-2 RGB 影像、深度影像

15-1-2 安裝 Kinect 驅動程式

在使用 Kinect 前追蹤人動作前，我們必須安裝 Kinect 的 Develop（開發包）、SDK（Software Development Kit，軟體開發工具包）以及 LabVIEW for Kinect 的函式庫，請依下列步驟安裝驅動程式：

STEP 1 至 Kinect for Windows 的官方網站（<http://www.microsoft.com/en-us/kinectforwindows/>）右上方搜尋列搜尋「kinect for windows」（圖 15-3）。

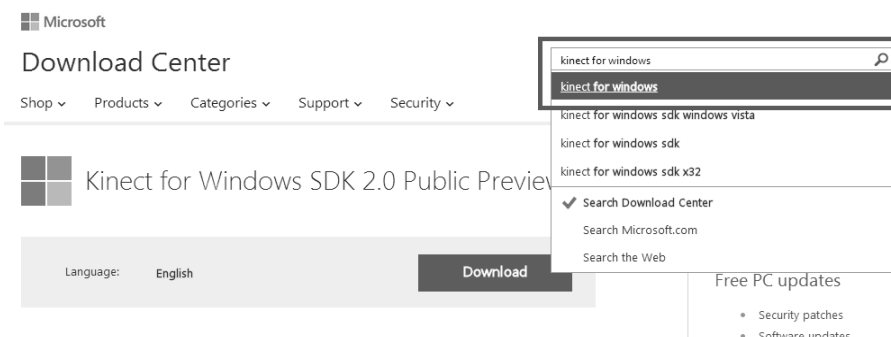


圖 15-3 在搜尋列搜尋 Kinect for windows

STEP 2 搜尋完成後在下方找尋二個安裝檔「Kinect for Windows SDK v1.8」及「Kinect for Windows Developer Toolkit v1.8」點擊連結後按下 Download 即可下載。

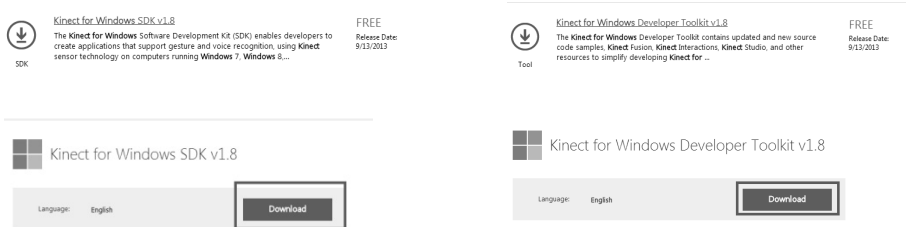


圖 15-4 Kinect SDK (左)、Kinect Toolkit (右)

STEP 3 下載完成後，點擊下載檔案進入安裝精靈（注意！需先安裝 Kinect SDK 後才可安裝 Kinect Toolkit）。

STEP 4 軟體授權頁面。請點選[I agree to the license terms and conditions]後按 Install（圖 15-5）。

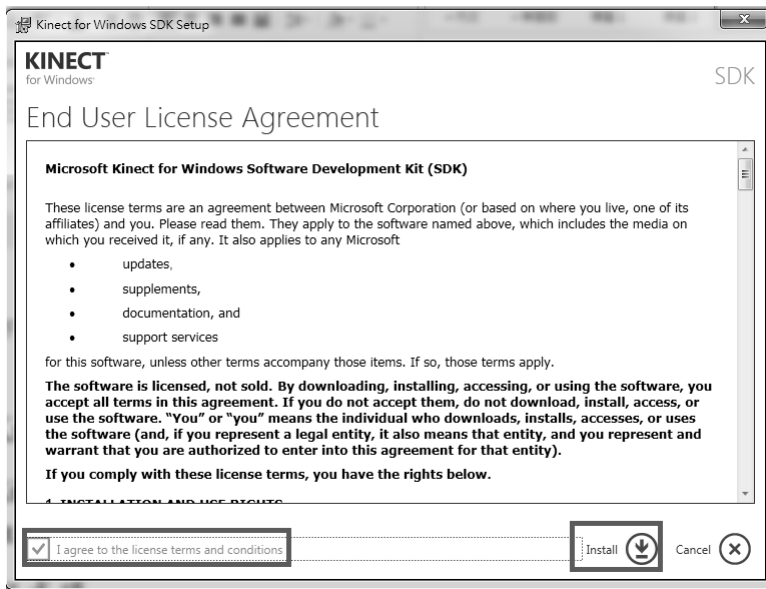


圖 15-5 進入安裝精靈

STEP5 安裝過程很快就完成了，Kinect SDK 安裝完成後接著安裝 Kinect Toolkit。

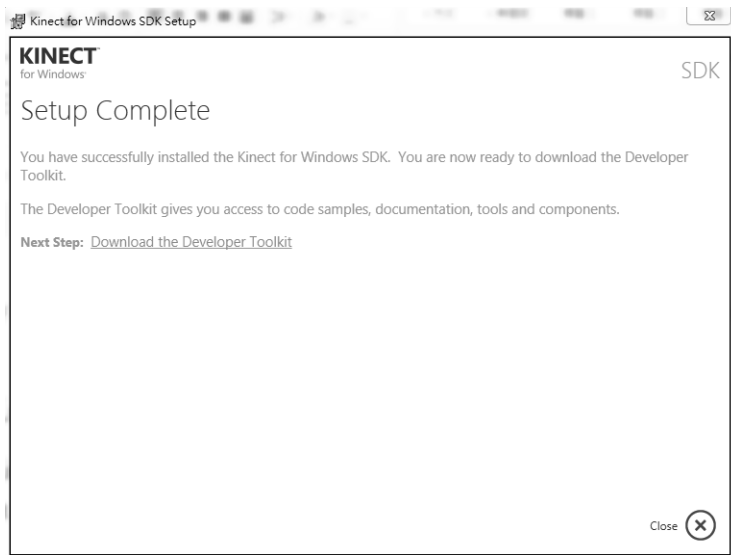


圖 15-6 Kinect SDK 完成安裝

STEP6 點擊 KinectDeveloperToolkit-v1.8.0-Setup.exe 進入安裝精靈。

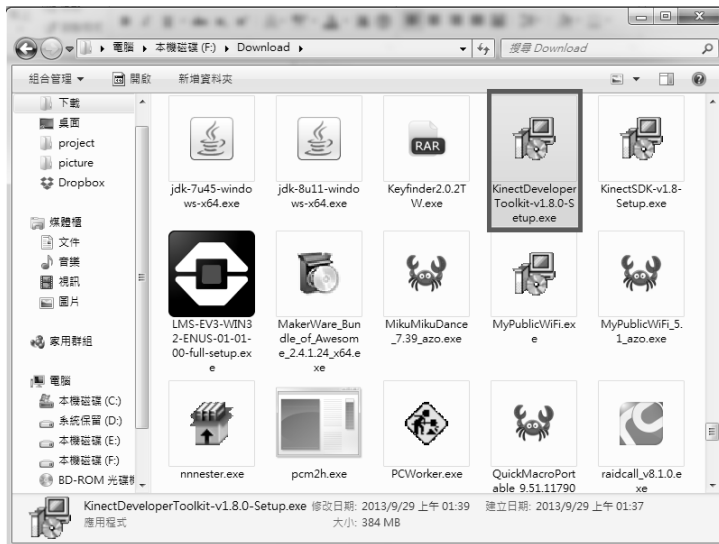


圖 15-7 點擊 KinectDeveloperToolkit-v1.8.0-Setup.exe

STEP 7 軟體授權頁面。請點選[I agree to the license terms and conditions] 後按 Install (圖 15-8)。

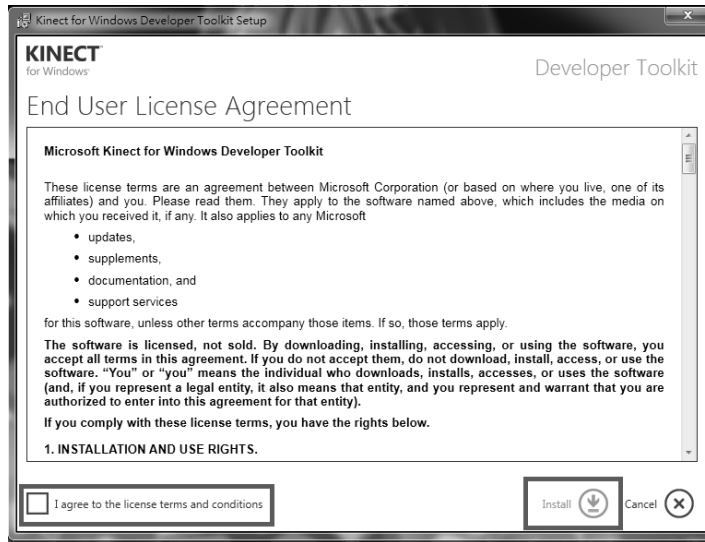


圖 15-8 進入安裝精靈

STEP 8 安裝過程也是很快就完成了 (圖 15-9)

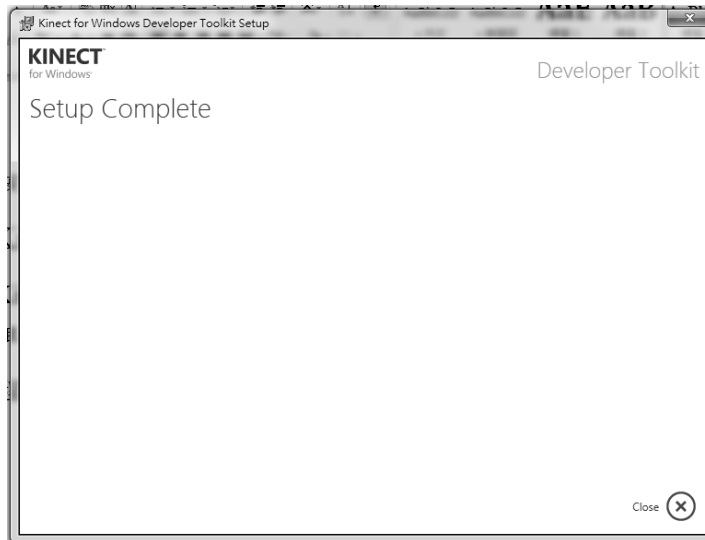


圖 15-9 Kinect Toolkit 安裝完成

15-2 Kinect 實作

15-2-1 Kinect 指令介紹

在開始撰寫程式前，我們先介紹一些常用到的 Kinect 指令。

1. Initialise：初始化 Kinect，並且取得有關 Kinect 的資訊。

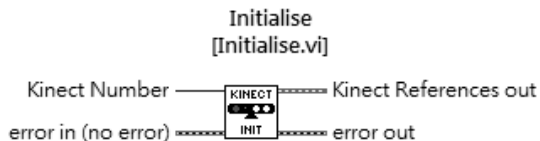


圖 15-10 Initialize

2. Configure：設定 Kinect 的模式，一共有三種類型：影像、深度、骨架。

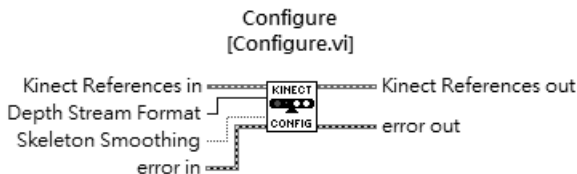


圖 15-11 Configure

3. Read：讀取 Kinect 所偵測到的數值，模式會依據 Configure.vi 的設定而有不同的讀取方式。

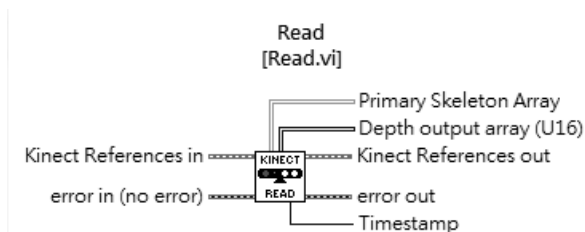


圖 15-12 Read

4. Angle Between Joints：計算 1、2 關節點與 2、3 關節點之間的夾角。

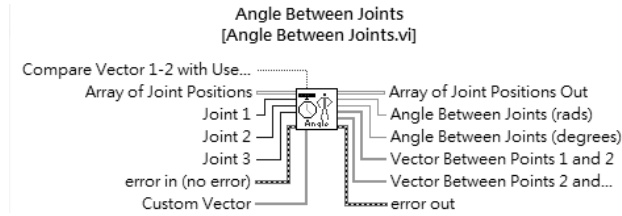


圖 15-13 Angle Between Joints

15-2-2 組裝機器人

機器人部分很簡單，僅需使用基本雙馬達車型即可，我們會在<EX15-3>中，由雙手角度來控制機器人的動作。

15-3 程式碼

15-3-1 Hello Kinect

第一次接觸 Kinect 要先熟悉它的基本架構及用法，本範例將利用 Kinect 讀取影像及深度資訊。

顯示影像及深度

EX15-1 showimage.vi

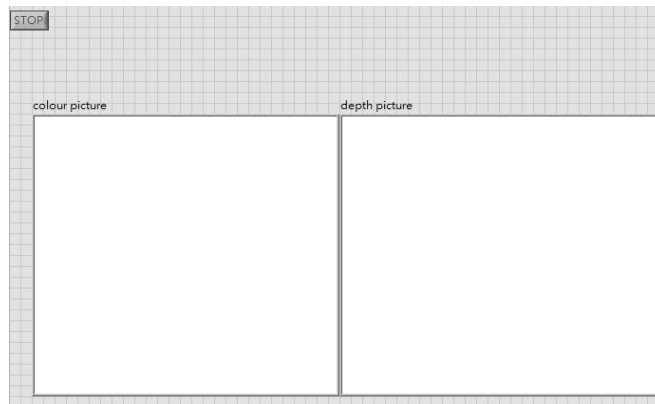


圖 15-14a <EX15-1>前方面板

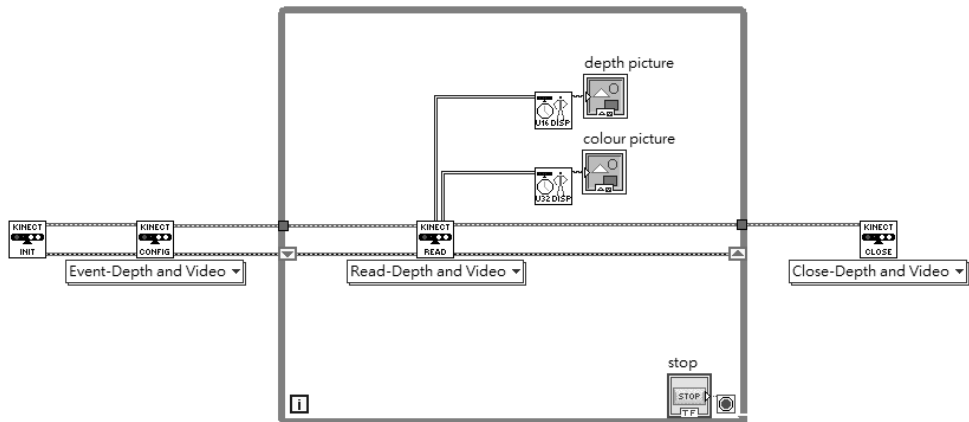


圖 15-14b <EX15-1>程式圖

- STEP 1 從程式方塊區的 Vision and Motion→Kinesthesia Toolkit for Microsoft Kinect 選單中新增 Initialise.vi。
- STEP 2 於同一選單中，新增 Configure.vi，由於這次我們只要看 RGB 影像及深度影像，所以在 Configure.vi 選擇 Event-Depth and Video。
- STEP 3 於同一選單中，新增 Read.vi，並將上述 3 個 vi 的 Kinect References out 連在一起（如圖 15-15）。



圖 15-15 將 3 個 vi 連在一起

- STEP 4 在 Vision and Motion→Kinesthesia Toolkit for Microsoft Kinect→Additional Vis 選單中，新增 Display Colour Data.vi 及 Display Depth Data.vi
- STEP 5 將 Read.vi 上的 Depth output array 及 Colour output array 分別連至 Display Colour Data.vi 及 Display Depth Data.vi，並分別在 Display Colour Data.vi 及 Display Depth Data.vi 的 new picture 端口點擊右鍵\Create\Indicator，然後分別重新命名為 Colour picture 及 Depth picture，這兩個顯示器將分別顯示彩色影像與深度影像。

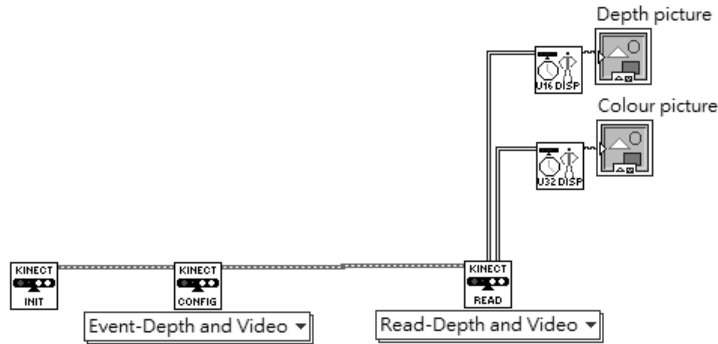


圖 15-16 新增 Display Colour Data.vi 及 Display Depth Data.vi

- STEP 6** 使用 While loop 迴圈將 Read.vi、Display Colour Data.vi、Display Depth Data.vi、Depth picture 及 Colour picture 包起來並新增一組 Shift Register。
- STEP 7** 從 Vision and Motion→Kinesthesia Toolkit for Microsoft Kinect 選單中新增 Kinect Close.vi 並將 Kinect References out 及 error out 連起來，有助於除錯。

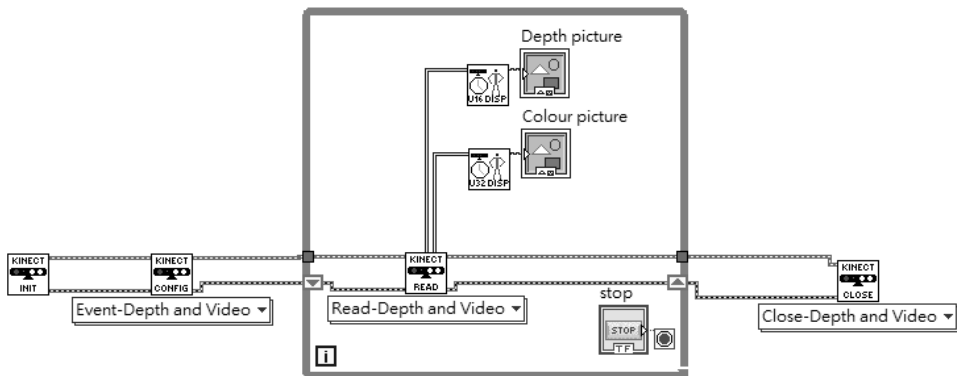


圖 15-17 <EX15-1>程式圖

完成之後，請接上 Kinect 後執行程式就可以看見即時的 RGB 影像及深度影像。

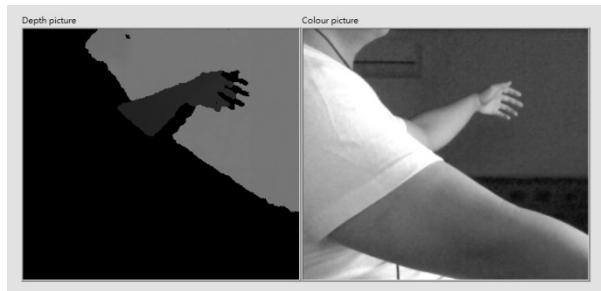


圖 15-18 深度(左)及 RGB(右)影像圖

15-3-2 顯示背景

Kinect 除了 RGB 影像及深度影像外，還可以擷取人體的骨架資訊。為了要了解 Kinect 偵測骨架的結果，我們要透過 LabVIEW 來顯示出「人」的架構。

EX15-2 skeleton.vi



圖 15-19a <EX15-2>前方面板，在此顯示骨架圖

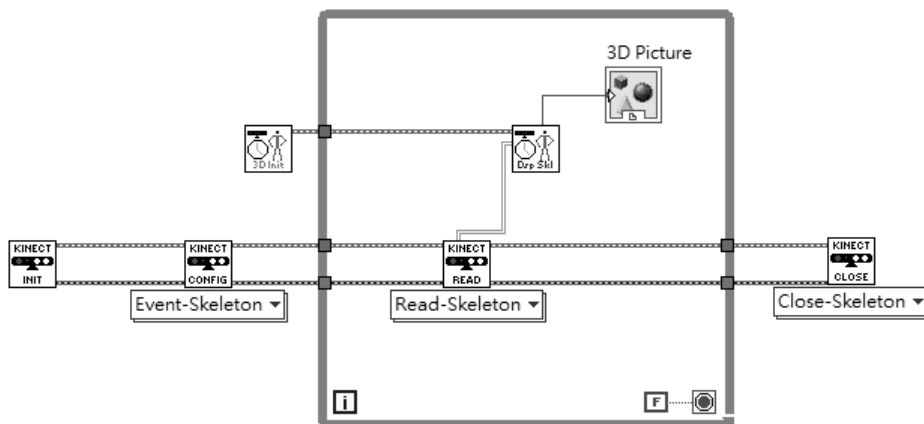


圖 15-19b <EX15-2>程式圖

- STEP 1** 從程式方塊區的 Vision and Motion→Kinesthesia Toolkit for Microsoft Kinect 選單中新增 Initialise.vi。
- STEP 2** 從同一選單中，新增 Configure.vi，並選擇 Event-Skeleton。
- STEP 3** 從同一選單中，新增 Read.vi，並將上述 3 個 vi 的 Kinect References out 連在一起（圖 15-20）。

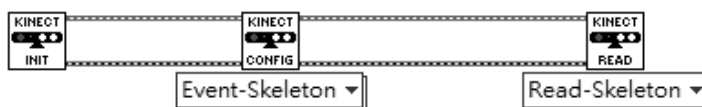


圖 15-20 將 3 個 vi 連在一起

- STEP 4** 從 Vision and Motion→Kinesthesia Toolkit for Microsoft Kinect→Additional Vis 選單中新增 Initialise 3D Skeleton.vi。
- STEP 5** 從同 STEP 3 的選單中，新增 Render 3D Skeleton.vi，並連接 Scene & Skeleton Refs In（圖 15-21）

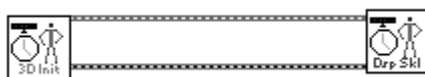


圖 15-21 3D Skeleton 指令

STEP 6 從前方面板的 Modern→Graph 中新增 3D Picture.vi，再回到程式方塊區將 Render 3D Skeleton 的 Scene Out 連至 3D Picture.vi。



圖 15-22 3D Picture

STEP 7 使用 While loop 迴圈將 Render 3D Skeleton.vi、3D Picture.vi、Read.vi 包起來，並從 Vision and Motion→Kinesthesia Toolkit for Microsoft Kinect 選單中新增 Kinect Close.vi，最後將線連在一起。

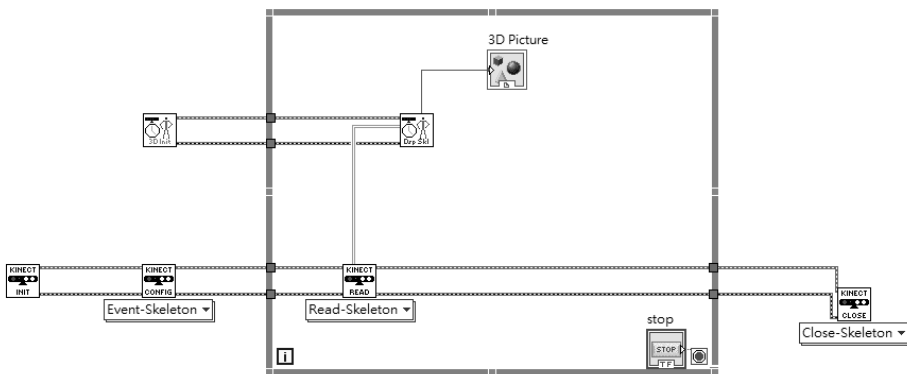


圖 15-23 完成圖

將程式碼完成後執行 LabVIEW 程式即可在前方面板看到 Kinect 所偵測的到您身體的各關節點喔！

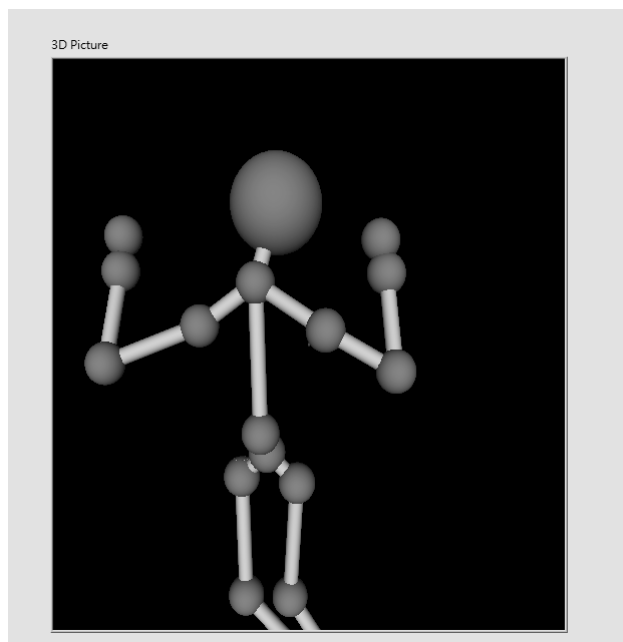


圖 15-24 人物骨架圖

15-3-3 Kinect 控制樂高 EV3 機器人

熟悉 LabVIEW for Kinect 的基本指令後，我們可以利用 Kinect 控制 EV3 車跑來跑去了！本範例可偵測雙手與身體的夾角，以控制機器人兩顆馬達的速度。

EX15-3 EV3 CAR BY KINECT

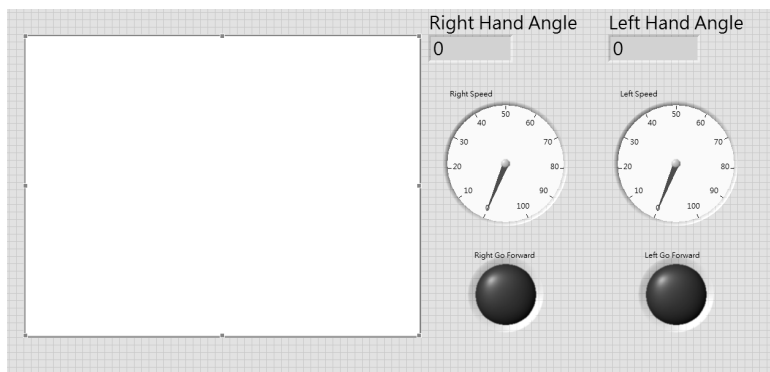


圖 15-25a <EX15-3>前方面板

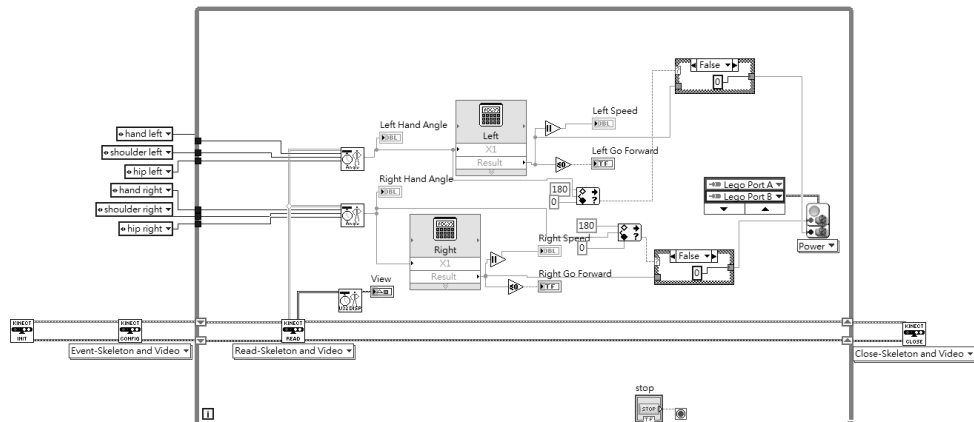


圖 15-25b <EX15-3>程式圖

- STEP 1 從程式方塊區的 Vision and Motion→Kinesthesia Toolkit for Microsoft Kinect 選單中新增 Initialise.vi。
- STEP 2 從同一選單中，新增 Configure.vi，並選擇 Event-Skeleton and Video。
- STEP 3 從同一選單中，新增 Read.vi。
- STEP 4 從同一選單中，新增 Kinect Close.vi，並將上述 3 個 vi 的 Kinect References out 連在一起（如圖 15-26）。

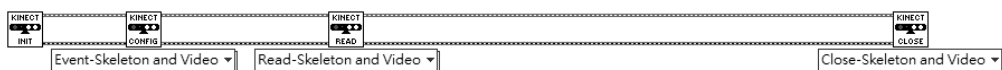


圖 15-26 Kinect 基本架構

- STEP 5 從程式方塊區的 Vision and Motion→Kinesthesia Toolkit for Microsoft Kinect→Additional Vis 選單中，新增 Display Colour Data.vi，並在前方面板顯示畫面。

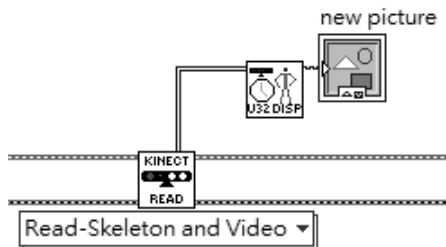


圖 15-27 顯示畫面

由於左右手的計算方式僅在於判斷部位不同而已，所以先寫右輪之後再撰寫左輪。

- STEP 6** 從 Vision and Motion → Kinesthesia Toolkit for Microsoft Kinect → Additional Vis 選單中新增 Angle Between Joints.vi，並在 Joint1、Joint2、Joint3 點擊右鍵 \Create \Constant，分別更改為 hand right、shoulder right、hip right。

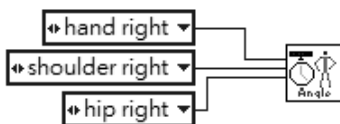


圖 15-28 新增 Angle Between Joints

- STEP 7** 從 Mathematics → Scripts & Formulas 選單中新增 Formula.vi，運算算式為 $(X1-90)/90*100$ ，並將 Angle Between Joints.vi 上的 Angle Between Joints(Degrees) 端口連至 X1。

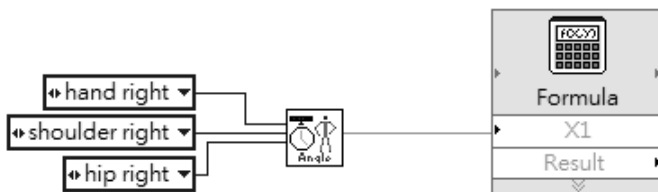


圖 15-29 新增 Formula

- STEP 8** 在前方面板新增三個元件，分別為 Modern → Numeric 中新增 Numeric Indicator 並改名為 Right Hand Angle；Express → Numeric Indicators 中新增 Gauge 並改名為 Right Speed、數值範圍為 0~100

以及 Express→LEDs 中新增 Round LED 並改名為 Right Go Forward。

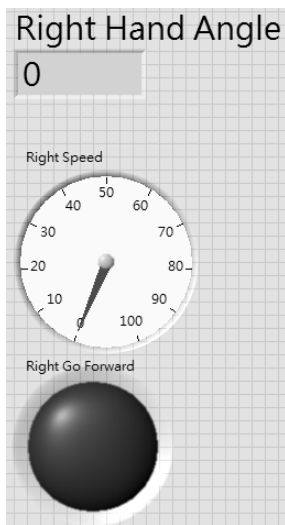


圖 15-30 觀測元件

STEP 9 將 Angle Between Joints.vi 上的 Angle Between Joints(Degrees)連至 Right Hand Angle 來顯示。

STEP 10 將 Formula 的 Result 取絕對值後連至 Right Speed 來顯示。

STEP 11 將 Formula 的 Result 判斷是否小於等於零後連至 Right Go Forward 來顯示。

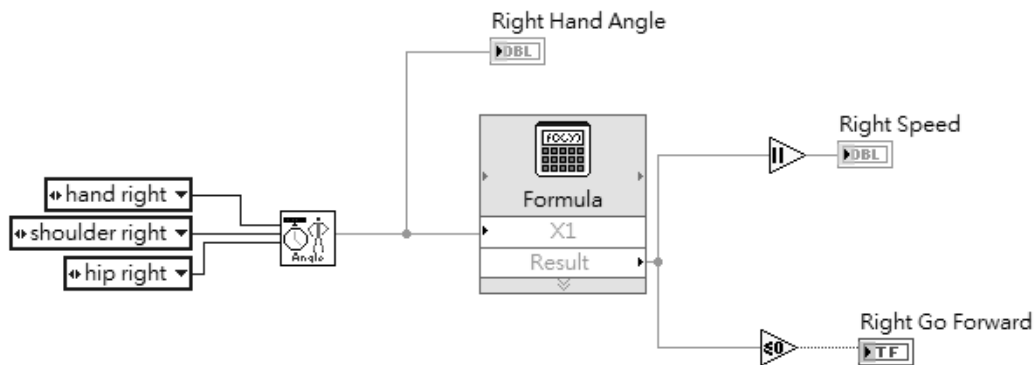


圖 15-31 顯示資料與判斷結果

STEP 12 從程式方塊區的 Programming→Comperation 選單中，新增 In Range and Coerce 指令，將 Angle Between Joints.vi 上的 Angle Between Joints(Degrees)連至 x，並設定上下限分別為 180 與 0，藉此限制資料範圍。

STEP 13 新增 Case Structure，在 False case 內放入常數「0」，再將 In Range and Coerce 的 In Range? 連至 Case Selector。將 Formula 的 Result 穿過的 True case。

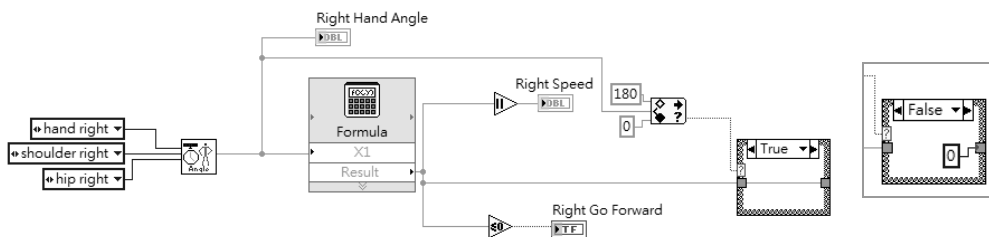


圖 15-32 連接 Case Structure

STEP 14 從 MINDSTORMS Robotics→I/O 選單中，新增 Move Motors 並在 Motors 端口點擊右鍵\Create Constant 新增 LEGO Port B 及 LEGO Port A，代表機器人的 A、B 馬達。

STEP 15 將常數 0 連至 Move Motors 的 Power/Speed 2，代表如果角度判斷結果在 0~180° 之間，則依據 STEP 7 的公式決定 A 馬達轉速；反之則停止馬達。

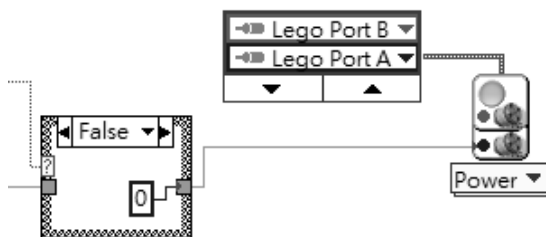


圖 15-33 超出範圍則停下馬達

STEP 16 重複 STEP 6~STEP 15 並將 Right 改成 Left，且 Case Structure 的結果連至 Move Motors 的 Power/Speed1。

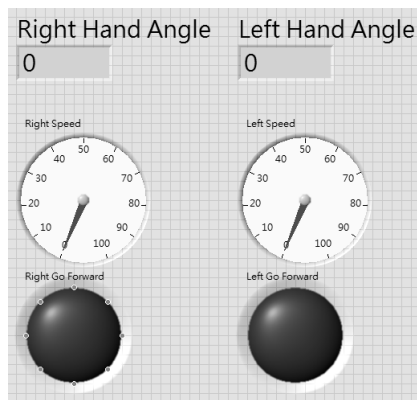


圖 15-34a 新增左手的前方面板元件

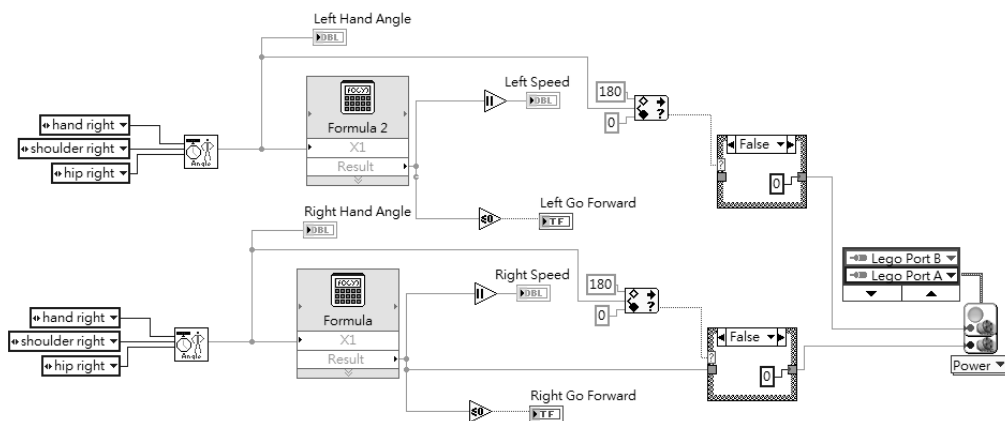


圖 15-34b 新增左手相關指令（畫面上半）

STEP 17 使用 While loop 將需要反覆執行的指令包起來，並將 Read.vi 上的 PrimarySkeletonArray 連至 Angle Between Joints.vi 上的 Array of Joint Positions，如圖 15-35。

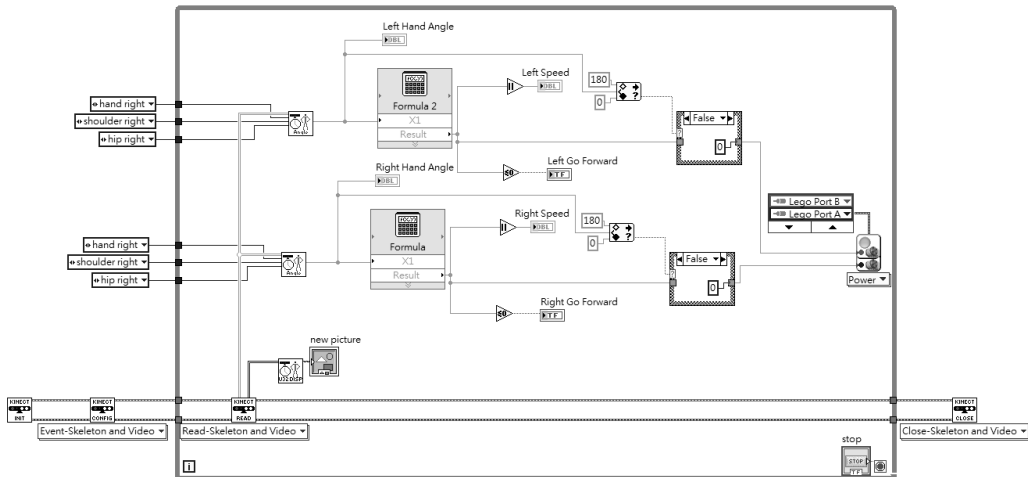


圖 15-35 <EX15-3>程式圖

程式完成後執行程式，就可以利用你的雙手控制左右馬達前進與後退的速度了！

15-4 延伸挑戰

1. 延續<EX15-3>，利用一隻手達成遙控機器人前進、後退與轉彎（提示：運用不同的關節角度）。
2. 想像自己手中握有方向盤，向前傾為車子前進、向後傾為車子後退、順時鐘轉方向盤為右轉、逆時針轉方向盤為左轉

15-5 總結

本書到此告一段落，希望您對於 LabVIEW 這套有趣又強大的程式語言已打下穩固的基礎，也能用手邊的樂高零件製作各種不同功能的自動化機器人。感謝您的支持。期待下一本書再相見！