

行政院國家科學委員會補助專題研究計畫 成果報告
 期中進度報告

具有教育及競賽功能的人形機器人系統之設計與開發-

子計畫三：機器人即時分散嵌入式系統之研究

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 96-2218-E-032-004-

執行期間：96年8月01日至97年7月31日

計畫主持人：李維聰 教授

共同主持人：

計畫參與人員：蔡佳良、林瑞營、黃柏昌、林弘偉

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

- 赴國外出差或研習心得報告一份
- 赴大陸地區出差或研習心得報告一份
- 出席國際學術會議心得報告及發表之論文各一份
- 國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：淡江大學電機工程學系

中華民國 97 年 5 月 30 日

0. 摘要

早期機器人這名詞曾在動畫、電影上出現，這幾年來更由於工業的發展自動化與機器手臂的帶入，更是帶領人類進入新的生活環境。自從早稻田大學的wabot-1與HONDA 的ASIMO發表後，開始引起各國與學術界的跟進，紛紛大力投入機器人的研究。

在我們的計畫案研究中，在本身機器人的運動控制外，還得因應比賽增加多機器人資源分享功能，在這裡我們用無線網路來傳輸資料，並制定一系列的API讓我們共同平台可以有很好的整合傳輸能力。

1. 前言

隨著機器人技術的發展與進步，多機器人間的溝通已形成如 Mark Weiser 這位學者所言的「無所不在的計算(Ubiquitous Computing)」的境界。對於競爭用途的人形機器人而言，若機器人可彼此分享由各種感測器所接收到的資料，將可在競賽中合作以取得勝利。而且，在人形機器人上存在著各式各樣重要的控制系統，例如行動控制系統、環境感知與活動偵測、高速視覺、人工智慧與知識管理以及智慧型控制系統等等。因此，一個良好的機器人共同平台將有助於整合所有不同的控制系統。為實現此目標，第一件事情是瞭解嵌入式硬體與軟體間如何溝通並執行有用的任務。機器人共通平台至少有無線通訊、硬體與軟體整合及客製化嵌入式作業系統等功能。

2. 關鍵字

智慧型機器人、多機器人資料分享。

3. 研究目的

本計畫的提出，是著眼於研究、設計、建置與測試總計畫(具有教育及競賽功能的人形機器人系統之設計與開發)所需之即時分散嵌入式系統(子計畫三)，實作人型機器人所需的共通平台(架構圖如圖3-1所示)，實作機器人共通平台具有下列主要目的：

- 支援各子計畫所需的硬體連接介面。
- 藉由無線網路通訊功能，使多機器人間了解彼此的意圖與行動來相互合作，實現多機器人運動策略於機器人足球賽中。
- 提供子計畫四建立相對應的機器人動作操作介面與模擬模型所需的執行環境。
- 傳送子計畫一的馬達迴授感測、壓力感測、平衡感測與加速度感測等感測器資訊及子計畫二的影像資訊並於接收子計畫四的決策判斷後，轉譯成控制命令下達至感測器與馬達，實施行為反應。

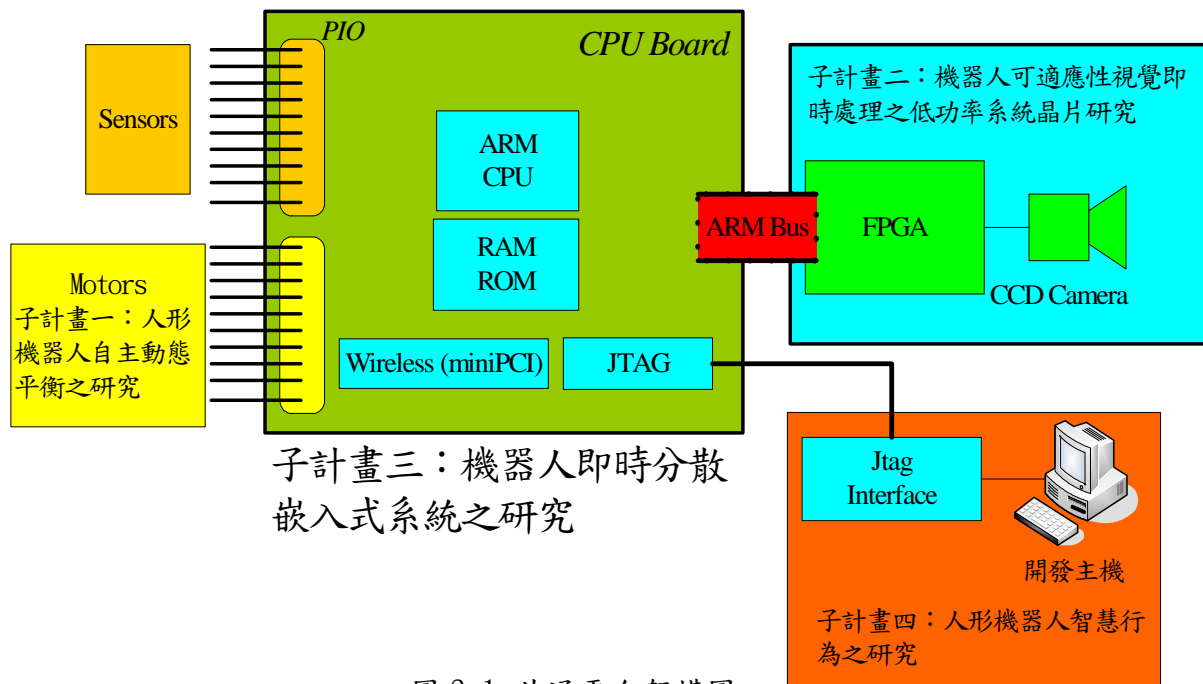


圖 3-1 共通平台架構圖

4. 研究方法

4.1 架構

分散式系統的架構可由軟體與硬體二個層面來討論。就軟體架構而言，Embedded Linux 的分散式計算架構如下圖4-1所示：

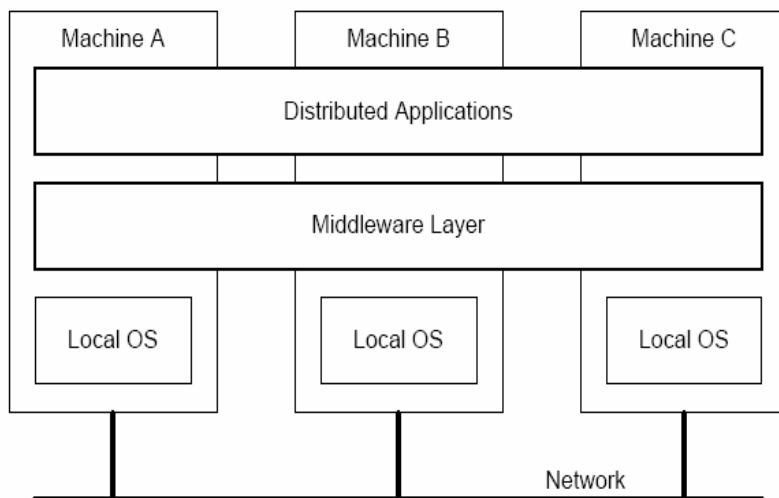


圖 4-1 分散式系統架構圖

由於嵌入式系統複雜度通常比一般的個人電腦作業系統來得簡化，並且程式碼會較小。因為它通常是應用於資訊產品設備，所以系統功能會因需求的不同，而有所不同。並且因為功能專屬，所以不需要複雜的人機介面，一般只提供文字模式供使用者使用。一般使用也不能自行開發程式。通常會提供全部的系統原始碼給系統開發者修改。在本計畫中，希望在執行上能將系統部分予以客製化，以配合日後各種不同類型之機器人的核心作業平台之使用。

4.2 研究步驟

本計畫主要的目的是建置人形機器人的即時分散嵌入式系統，支援動態感知系統、步

態規劃、視覺處理系統晶片、感測電路系統與驅動電路系統。在此提出研究的方向和想法，並期望將其用下列實作的技術達成，而其中的細節我們分成五個部份來進行：

- 建置與測試嵌入式平台。
- 建置與測試電子電路系統。
- 建置與測試無線網路系統。
- 設計、開發與測試嵌入式軟體與 API。
- 研究多機器人間自主溝通模式。

4.3 第一年預定成果

4.3.1 第一年預定工作及完成工作項目：

1. 完成CPU Board的建置：

以ARM9為核心CPU，並依據各計劃之需求，建置各種硬體設備及介面設備。

2. 完成OS的建置(含Sensor, Motor Drivers)：

目前我們以LINUX建置作業系統，執行子計畫四的策略架構平台，並收集儲存子計畫二所提供之訊息，再依據子計畫四所發出之命令，提供所需資料。此外，子計畫四亦經過此平台向子計畫一下達各種指令。

3. 完成定義API：

目前定義完成了三個API介面，分別是Sensor、Motor、FPGA及Wireless，並且針對各API的功能做不同的Function Call，以下列的是各API Function Call的名稱：

1. *struct Motor_Control(int M_ID, int M_Order, *void Data)
2. *struct Sensor_Control(int S_ID, int S_Order, *void Data)
3. *int W_Send(int R_ID, *struct Data)
*int W_Receive(*void Data)
*struct Ref_Command(*void Data)

4. 完成多機器人溝通系統用之無線網路研究：

在無線網路這方面最主要做的是多機器人資料分享及網路監控。我們使用 WiFi 做為無線傳輸的網路。在而程式設計上分兩類：傳送和接收，因此兩個命名分別是*int W_Send和*int W_Receive。在傳送上有兩個參數，原因是傳送時需要知道給哪一架機器人及資料，所以參數設定為 int R_ID、*struct Data。而在接收上，機器人只要負責接收資訊，所以參數只需設定為*struct Data 即可。

裁判箱這部份是只有在比賽時才有的，也是利用 WiFi 做無線傳輸下達命令給機器人，在程式設計上是以 struct Ref_Command 為命名。在資料傳送數據上，我們設計一個參數指標命名為*void Data，方便裁判針對所有機器人做監控。

4.4 工作結果與討論

4.4.1 CPU Broad 的建置：

我們是參考陸達半導體 STR9105 SPECIFITION，未來可能會需要擴充新的功能，所以留了一組 ISA-Bus 以防未來需要新增功能，下圖 4-2 是我們的電路板實體圖：

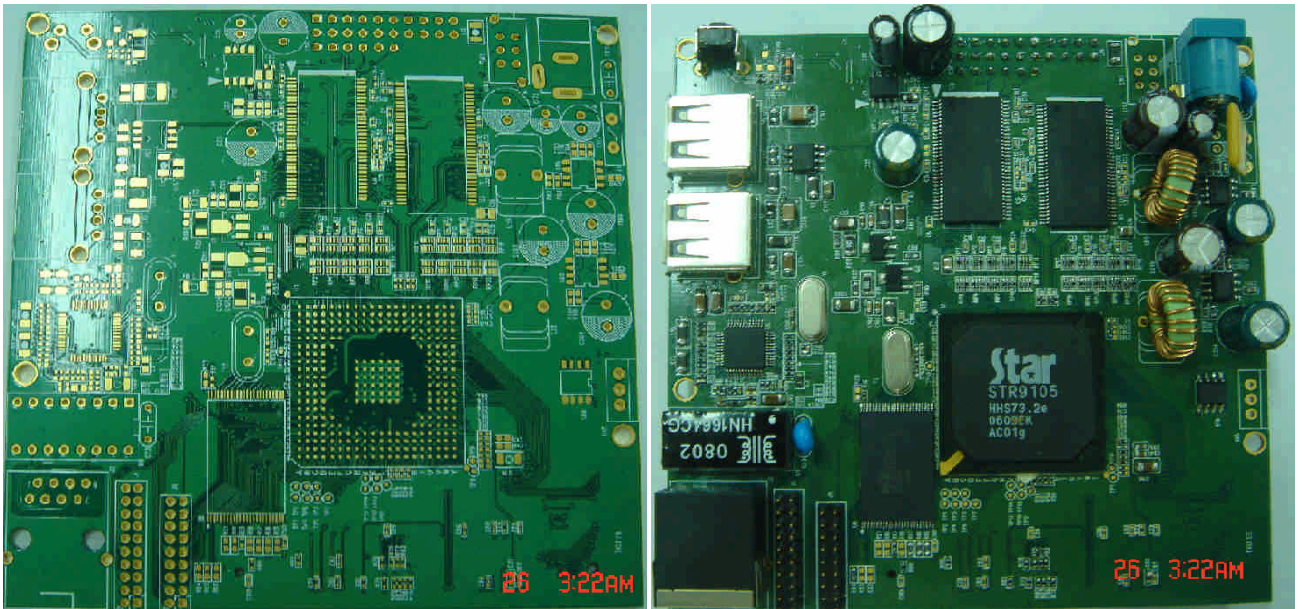


圖 4-2 電路板實體圖

4.4.2 OS 的建置:

在 Linux 系統上它的 Source Code 是 Open 的，而在 Sensor, Motor Drivers 的設計上更能有基本的 Source Code 可以參考，減少了我們在系統上出錯的問題，下 4-4 圖是我們 OS 的架構圖:

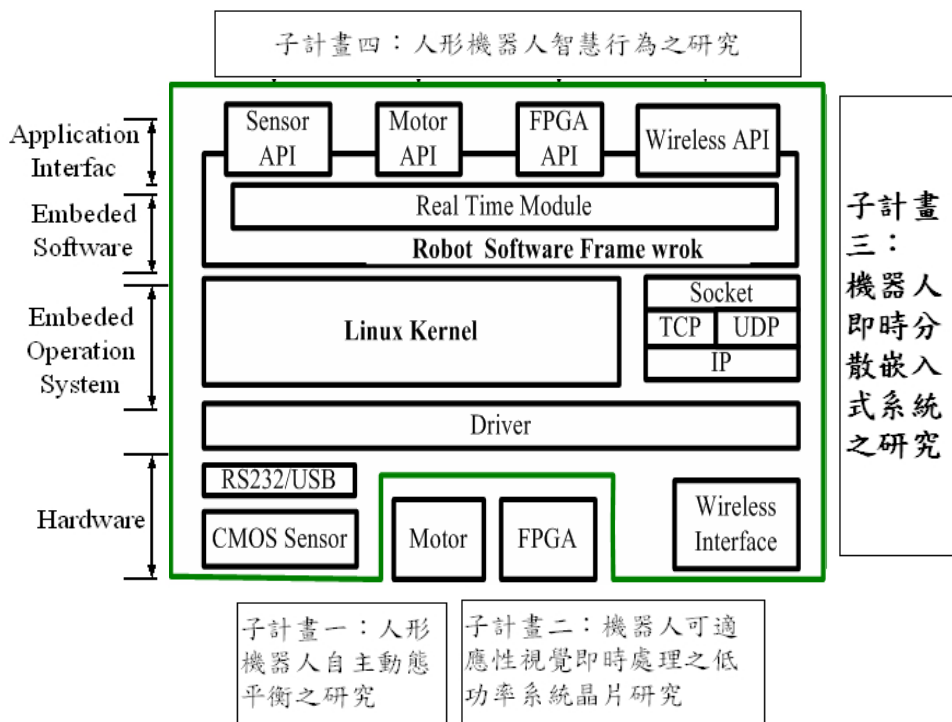


圖 4-3 OS 的架構圖

4.4.3 API 的定義:

在 API 的定義規劃上我們討論出依照功能命名及設定它的參數，由於這是各界面之間的溝通，所以需要統一制定一個資料庫，讓各界面之間都可以彼此瞭解各功能相互需求的資料，表 4-1 顯示我們規範 API 的命令及動作介紹：

4.4.3.1 馬達控制:

馬達控制在這裡最主要是控制它所轉的角度，並且控制它在轉角度時的速度，而本身馬達內部的回溯電路也讓我們容易得知目前馬達的資訊。而在程式設計上我們是以*struct Motor_Control 命名，設定三個參數為 int M_ID、int M_Order、*void Data，這分別對應 Motor ID、命令的類別以及資訊的下達。

功能	函式	內容	資料
馬達控制 P.12	*struct Motor_Control (int M_ID, int M_Order, *void Data)	ID 設定(0x03)	0~253(FD) and 254(FE) for available ID.
		鮑率設定(0x04)	Address 1~34
		回傳的延遲時間(0x05)	
		溫度限制設定(0x0B)	高溫限定與低溫限定
		最大力矩(0x0E~0F、0x22~23)	
		狀態回傳(0x10)	
		角度極限設定(0x14~17)	
		目標角度(0x1E~1F)	移動到的絕對角度
		移動速度(0x20~21)	移動時使用的速度
		鎖住(0x2f)	鎖住這個馬達
		強力(0x30~31)	強力轉動

表 4-1 馬達控制的 API

4.4.3.2 感測器控制:

這裡目前是由三個 Sensor 所組成，分別是壓力感測器、G Sensor 及電子指南針。壓力感測器是放置在腳底板上，它可以計算機器人的重心；G Sensor 是在計算機器人移動時的角速度，以上兩個 Sensor 最主要的功用是在機身平衡。最後電子指南針功用是以北極夾角角度為基準，可以設定機器人的相對座標及方向。而程式設計上我們是以*struct Sensor_Control 命名，設定三個參數為 int S_ID、int S_Order、*void Dat，分別對應 Sensor ID、命令及資訊的下達。如表 4-2 所示：

感測器控制	*struct Sensor_Control (int S_ID, int S_Order, *void Data)	壓力感測器資料	兩腳共八個壓力感測器值，以及中心點位置。 (請見資料)
		加速度計資料	三軸的角加速度(請見資料)
		電子指南針資料	角度資訊(請見資料)

表 4-2 Sensor API

4.4.3.3 無線網路:

在無線網路這方面最主要做的是多機器人資料分享及網路監控，我們使用 WiFi 做我們的無線傳輸，定義傳送和接收的命名分別是*int W_Send 和*int W_Receive，並且依照回傳的資料各制定了 struct Data int S_ID 及*void Data。如表 4-3。

無線網路	*int W_Send(int R_ID, *struct Data) *int W_Receive(*void Data)	監控端開啟	監控端開啟監控
		監控端停止	監控端停止監控
		角色定位(前鋒等)	定位機器人的角色
		隊友位置	回傳目標隊友目前位置
		自身位置	送出自身位置資訊
		下一步策略	送出機器人目前決定的動作至監控端
		特殊策略(競賽用)	監控端送出特殊策略命令至機器人

表 4-3 無線網路 API

4.4.3.4 裁判箱:

在裁判箱這裡是只有在比賽時才有的，他也是利用 WiFi 做無線傳輸下達命令給機器人，在程式設計上是以各動作的行為制定了一個 struct Ref_Command 為命名結構。如表 4-4。

裁判箱	*struct Ref_Command(*void Data)	開始命令	比賽啟動
		停止命令	比賽停止
		連結命令	與裁判箱連結
		停止連結	與裁判箱斷線
		紅牌	紅牌指令，機器人停止動作下場
		黃牌	黃牌指令，無動作
		PK	執行 PK 策略，機器人自動定位
		自由球	執行自由球策略，機器人自動定位
		球門球	執行球門球策略，機器人自動定位

表 4-4 裁判箱 API

5. 預期成果

5.1 未來第二年預定完成工作:

1.完成 FPGA 之介面建置：與子計畫二影像介面溝通模式與驅動程式撰寫。

本子計畫依第一年所研議之 FPGA 以及 ARM Bus 規格，建置並撰寫相對應之 API Driver，並與子計畫二進行測試。

2.完成 API 與驅動程式溝通的開發與測試：完成 API 與 Driver 之撰寫與建置。

本子計畫依第一年所研議之 API 與 Driver 規格，完成撰寫，並與子計畫一進行 I/O 之測試，且與子計畫四進行 Sensor 的測試，與指令的執行測試。

3.完成無線網路架構與 API 開發與測試:

本子計畫依第一年所研議多機器人溝通系統選定之網路，建置無線網路系統，並且完成 API 以及驅動程式的開發，以供第三年多機器人溝通系統需求。

5.1 未來第三年預定完成工作:

1.提供整合測試的環境：整合系統測試。

本子計畫將與其他計畫同時進行連結，測試子計畫四能否順利藉由 PIO 對子計畫一之機構以及 Sensor 進行控制，以及與子計畫二之資料轉送是否正常運行。

2.完成嵌入式作業系統客製化：評估現有嵌入式系統，研議客製化的可行性與否。

客製化是近來嵌入式系統的趨勢，本子計畫亦應對此有所研議。因此，將針對此一議題，進行是否能針對不同類型機器人的功能需求，能給予最有效的環境。

3.完成 PID 迴授控制介面之建置: 建置一 DIP 迴授控制介面。

依第二年的計畫，建置一DIP迴授控制介面，以利子計畫一能更順利處理迴授以及加強伺服韌性

4.完成多機器人間無線網路溝通系統建置與測試。

由於多機器人溝通系統是本計劃之另一個要件，因此本子計畫依第二年所建置多機器人溝通系統選定之網路，進行多機器人間的多模即時溝通測試，順利達成資料的交換。

6.結論

在第一年度的計畫案中，我們完成了電路板的設計，並規劃因應其他子計畫的 API，達到執行上，能將系統部分予以客製化，以配合日後各不同類型之機器人的核心作業平台之，未來將繼續以這平台測試各 API 的功用與整個機器人的控制，已完成計畫案的整個內容。

7. Reference

[1] Rabaey J, Ammer M, Silva J, et al., PicoRadio supports ad hoc ultra-low power wireless networking , IEEE Computer Magazine, pp.42-48, 2000