

# 設計與實作一輛具無線傳輸能力之多功能自走車

許健平\* 李匯宗\*\*

\*國立清華大學資訊工程系

\*\*國立中央大學資訊工程研究所

sheujp@cs.nthu.edu.tw

dennis@axpl.csie.ncu.edu.tw

**摘要**-在本篇論文，我們設計一台可遠端遙控的自走車，此車上裝置可垂直、水平移動的攝影機平台，讓使用者可以利用電腦來監控遠端的景物，並命令自走車移動至適當的位置。車上也裝置超音波測距感測器、溫溼度感測器、及電子羅盤，這些感測器可用於危險環境的量測，也可作為自走車自動巡航及避障的設計，為了擴展其應用範圍，本研究在自走車上結合無線感測器節點，可進行無線感測網路的相關應用。自走車上的控制器設計採雙層式設計，上層控制器可透過無線網路與遠端使用者連絡，在下層控制器以一個 Microcontroller 控制三個負責不同功能的 Microcontrollers，而這些控制器再依照指令來控制更底層的電子元件。本研究於實作完成後，經過測試，確實可達到於遠端進行遙控、量測，在整合無線感測器節點後，可透過感測器節點取得感測資料並傳回終點端。

**關鍵字：**無線感測器網路、自走車、遠端監控、嵌入式系統、無線網路

## 1. 簡介

本文的研究目的，希望可以設計一台利用目前最為流行及普遍的無線網路來遙控遠端的機器人自走車，於車上裝置可垂直、水平移動的攝影機平台，讓使用者在遠端使用電腦來觀看遠端的時況，並可適時的做出移動方向的判斷，車上還裝置了超音波測距感測器、溫溼度感測器及電子羅盤，這些感測器可用於危險環境的量測，也可作為自走車自動巡航及避障的設計，為了擴展其應用範圍，本研究在自走車上結合無線感測器節點 (sensor node)，可進行無線感測網路 (Wireless Sensor Networks, WSNs) 的相關應用。

在現今相當熱門的 WSNs 領域裡，目前大都著重於

靜態 (static) sensor nodes 佈設、通訊協定、電源消耗等議題[6]，我們知道 WSNs 裡，除了 static sensor 外還有動態 (mobile) 的 sensor[5]，本研究是實作 mobile sensor node 於遙控自走車上，mobile sensor 具有 static sensor 所沒有的優點，移動性是它最大的優點，我們可以不需要大規模的佈設 sensor node，即可在短時間內派出 mobile sensor，去取得我們最感興趣的資料。

應用之一是在危險環境進行探勘，任何危險、狹小及人類不適於到達的地方，我們都可以藉由此自走車來取得遠端畫面、溫溼度等資料。另外一個為 WSN 的應用是在偏遠無人的環境裡[2,6,7]，例如沙漠、無人島、森林等，如 sensor 故障，或當初佈設的位置是空間狹小的時候，由人來進行維護是相當困難的，因考量到安全、成本、效率等問題，因此我們可以從遠端遙控自走車來進行維護，因其具備了 sensor node，如有故障的 sensor node 又無法透過自我組織的通訊協定來變換 routing 路徑[4]，此時我們可以透過無線訊號來逐一搜索故障的 sensor，當找到後再利用自走車的機械手臂進行更換。在 WSN 裡的定位相關議題，也是相當的重要的，由於 sensor node 講求成本低、耗電低，所以一般都不會具備定位裝置[1,3]，例如：GPS，如何讓 sensor node 可以獲得正確的位置，關係到許多應用的測量準確，例如：於草原上測量野生動物的移動路徑，錯誤的定位資訊，會使的實驗數據有很大的錯誤，在這我們可以利用自走車的定位系統取得所在地的位置資訊，並用廣播的方式，來讓附近的 sensor 獲得自己的位置。本研究於實作完成後，經過測試，確實可達到於遠端進行遙控、量測，在整合 WSN 後，可透過 Sensor node 取得感測資料。實際上已完成的自走車如圖 1 所示。



圖 1 自走車實體圖

## 2. 系統架構與背景技術

車體為整體相當重要的一環，除了必須搭載所有零組件外還需要可以靈活移動，而輕量化也是必須考慮的課題，因為過重的車體將會影響到電池的續航力，依照需求選擇了由 Lynxmotion 公司所提供的產品 4WD1Robot，其類似壓克力的材質提供足夠的支撐強度來承載 3.5 公斤的重量，以及全時四輪傳動系統，提供在惡劣的路況下也可能順利的移動。車體的動力是來自於四具減速馬達，此馬達齒輪比為 50:1，兼具速度及扭力，其傳動方式類似於一般軍用的裝甲車。馬達的驅動必須依靠 motor driver，依照 microcontroller 給予控制邏輯來控制正逆轉與停止。車體上方架設了攝影機，提供使用者可以遠端即時獲得現場畫面，對於攝影機我們採用成本低廉的 Web Cam。車體控制器是負責車上所有功能的運作，以及回應遠端使用者的需求，依照功能劃分為兩個層式設計，如圖 2 所示。

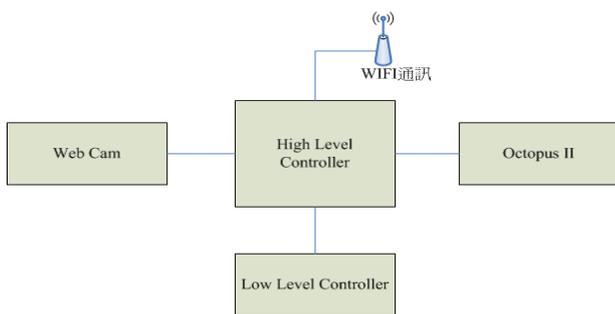


圖 2 控制器架構圖

上層控制器(High-level Controller)負責 WiFi 通訊、視覺系統、高階運算功能、WSN 通訊，最後再由一個 RS-232 通訊介面連接至下層控制器(Low-level Controller)。High-level Controller 的設計是以 single board computer 為核心控制器，整體是採階層式設計，往上可透過無線網路與遠端使用者連結，往下可透過 RS-232

介面與 Low-level Controller 連接，旁側可使用 USB 來連結各式的裝置，如圖 3 所示。

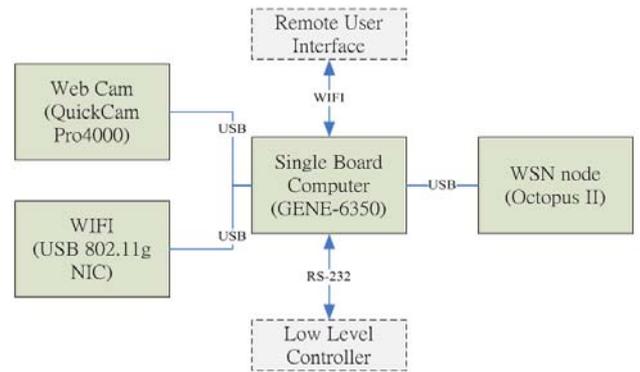


圖 3 上層控制器硬體架構圖

上層控制器各項裝置及功能如下：

- Web Cam：提供現場畫面，讓使用者不必親臨現場可以獲得資訊。
- WSN Node：這裡採用中央大學的高速通訊與計算實驗室所開發的 OctopusII。OctopusII 經由 USB 連線至 single board computer，透過 OctopusII 即可擷取 sensor node 上的資料，也可過 RF 與其他 sensor node 進行通訊。
- WiFi Interface：為了讓通訊及遙控裝置可以有更遠的控制距離以及更佳的資料傳輸能力，使用 WiFi 加上 Internet 的應用，我們甚至可以在地球的彼端也可控制此部自走車。

下層控制器主要控制車體上的電子元件，包括馬達驅動器、超音波測距感測器、溫溼度感測器、電子羅盤及伺服機控制器，如圖 4 所示。

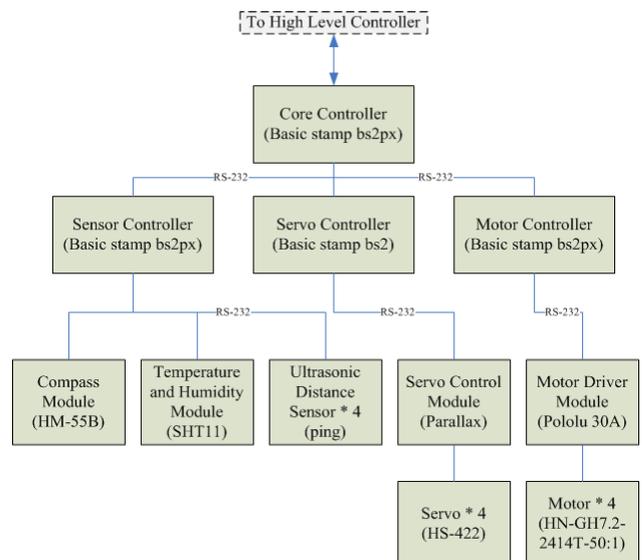


圖 4 下層控制器硬體架構圖

下層控制器主要是控制實體層電子訊號的傳遞，在

Low-level Controller 裡也是採用階層式的設計，以一個 microcontroller 控制三個負責不同功能的 microcontrollers，而這些控制器再依照指令來控制更底層的電子元件。

在供電系統方面，本文所使用的單板電腦僅需要使用 5 伏特即可運作，經過實測，運轉時須要 3.5A 才足夠推動此電腦，故購買了一個 5V5A 的穩壓器，將鋰電池的 12V 降為 5V。另外由於馬達在啟動時需要較大的電流來推動，而此部自走車又具備四個馬達及五個伺服機，所以電池的選擇就必須要能夠瞬間供應大電流，於此我們所使用的電池為遙控模型常使用的 7.2V 鎳氫電池。本文所使用的四個馬達，在全速運轉及負重在 1kg 時，耗電約為 550mA，再加上 Low-level Controller 及 sensor 的耗電量約為 3A 左右，估算可使用時間約為 1.37 小時。

### 3. 控制器軟體設計

控制器的軟體依照硬體的階層式設計，Low-level Controller 主要是由四個 Basic Stamp 2(BS2)的晶片組成，於 BS2 上是使用 PBASIC 來開發，各個 controller 都有各自負責功能，其功能說明如下：

- Core Controller：主要功能是做為與 High-level Controller 的通訊窗口。當 Core Controller 收到來自 High-level Controller 的 Command 後，會進行指令的轉換，再用 UART 介面傳送至負責該功能的控制器上執行。
- Sensor Controller: 當 Sensor Controller 收到來自 Core Controller 的 command 時，將依據要測量的 sensor 的類型來呼叫函式來測量，並將數值轉為可渡性較高的數值，並將結果回傳給 Core Controller。
- Servo Controller: 當 Servo Controller 收到來自 Core Controller 的 command 時，會依據要控制 Servo 的目前位置角度來進行增加或減少的運算，再將結果轉換成指令送至 Servo Controller Module，該 module 就會依照我們需要的角度來驅動 Servo。
- Motor Controller: 當 Motor Controller 收到來自 Core Controller 的 Command 時，依據要行進的方向，來控制四個馬達不同的轉向，再送至 motor driver module 來驅動馬達。

在 High-level Controller 上的 single board computer 的 OS 我們是採用 Ubuntu Linux 7.04 Server edition，其

kernel 為 2.6.20 版本，由於 Linux 為 open source 的 OS 所以我們可以很容易的在裡面安裝我們需要的軟體以及開發需要的程式，在一些特殊的情況下，我們還可以修改核心的程式來符合我們需要。程式開發方面我們採用 Java 語言，因為在設計自走車程式的 OS 是 Linux、使用者操作介面的 OS 是 Windows，為了讓開發的程式語言具有一致性，所以選擇了 Java 作為跨平台的開發語言，且其優良的 OO (物件導向)特性，相當適合開發自走車。

High-level Controller 上的軟體設計，是由 Core 來做為主要的邏輯控制核心，而 Data Exchange Module 則為程式的資料交換中心，由於為多執行緒的設計，所以各部份的通訊介面將有不定時的訊息傳入，而各個模組也可透過 Data Exchange Module 取得必須的資料。

各個模組說明如下：

- Core Module：Core Module 為系統的主程式，控制所有模組的邏輯，也包含管理架構裡的 Thread。
- VLC Module：VLC module 是採用 J VLC 來做為視訊多媒體的主要程式庫，包括啟用、停止、傳輸、解碼等，並提供介面讓試用者可以從 UI 來控制。
- RMI Object Module：為了簡化通訊協定的設計，選用了 Java RMI 來做為與遠端使用者通訊的主要方式，透過預先設計好的 RMI Interface 及 Method，使遠端的使用者操作介面只要呼叫這些 Interface 的 Method 即可達成資料傳輸的目的，而不必在乎底層封包的傳輸格式及方法。
- Data Exchange Module：由於控制器是屬於模組化及多執行緒的設計，在任何時候都有來自遠端使用者、感測器等，的各種通訊資料，於這裡我們設計了一個資料交換中心，每個模組及 Thread 獲得資料時會將此資料寫入，而需要資料時就會去檢查是否有自己需要的，如此可節省每個模組之間的資料交換時間以及浪費記憶體空間的問題。
- Sensor Polling Thread Module：當使用者要求 Sensor 開始量測時，Core Module 會啟動 Sensor Polling Thread，此 Thread 便開始依照順序陸續將指令寫入 Data Exchange Module，當 Low-level Controller 有資料回應時，此資料也會直接寫入 Data Exchange Module，而遠端的使用者介面只要定時來讀取即可。
- Serial Connection Thread Module：此 Module

為維護與 Low-level Controller 之間 RS-232 的連線，當有資料要寫入時，會開啟 Output Stream 將資料輸出，當有資料由 Low-level Controller 進入時，此 module 會將資料讀入，並寫入 Data Exchange Module。

- Serial Write Thread：此 Thread 主要是監看 Data Exchange Module 的 Command 資料欄位是否有指令進入，當讀到的資料為空白時，代表沒有指令，非空白時，先用正規表示式檢查此指令是否完整，正確無誤後，再呼叫 Serial Connection Thread Module 並將此指令傳送至 Low-level Controller。
- Auto Navigation Thread Module：此 Thread 是自動導航及避障的邏輯控制，當使用者介面將自動導航的功能開啟後，Core Module 會啟動此 Thread，並依照自行取得的 Sensor 資訊來判斷自走車行進的方向，當超音波測距感測器偵測到障礙物過於接近時，會開始避障的動作，並在離開障礙物後繼續前進。

在無線感測器資料讀取方面，sensor 上的程式，是開發在 TinyOS 上的應用程式。nesC 是 TinyOS 上專用的開發語言，設計的概念是採事件驅動來操作，並將不同的功能設計成 Module，在程式有使用到時才會包含在系統裡面，適合於有限資源的 Sensor Node 來使用。在 nesC 的程式裡，一開始先載入必要的 Module，包括 Timer、Communication、Temperature 等，接下來讀取 sensor 的資料，並組成封包往 UART 送出，接下來重複以上動作，所傳送的資料是包括：溫度、溼度、光合作用有效光、全光譜值、內部溫度、內部電壓等。Sensor 的讀取程式是由 Java 開發完成，於程式內部引用 TinyOS 已開發好的程式庫，就可順利的完成。程式一開始先開啟 Com Port，開始讀入由 sensor node 經由 UART 送來的封包，依據自訂的封包格式進行解讀，再將資料顯示出來，此時再將此資料寫入 Data Exchange Module，待使用者介面的程式來讀取。

為了簡化自走車與遠端使用者之間通訊協定的複雜性，本研究採用 Java Remote Method Invocation (RMI, Java 遠端方法呼叫)。RMI 的使用讓我們不必考慮各種通訊的細節，例如：資料形式與資料結構轉換、連線的管理等。當我們將這些細節簡化，則程式設計者可以把時間轉移到應用系統的需求上，這才是使用者最關心的。RMI 的基本原理，是把網路的通訊看成程式中

的程序(Procedure)，當成是需要遠端程序提供某種服務或接受訊息時，直接呼叫一個程序，而這個程序有相對應的遠端程序，可以在呼叫後於遠端的電腦上被執行。

為了讓自走車可以長距離自動行巡航以及遇到障礙物時自動閃避，以避免車體損壞。於車體前方、右方、左方，各裝置一個超音波 sensor，當偵測到其前方的 sensor 偵測出障礙物距離小於 80 公分時，會進行避障的動作。本研究實作一個簡易的避障功能，主要是利用車體上的超音波測距感測器來得知障礙物的距離，進而達到避障的目的。前方 sensor 負責偵測行進中的障礙物，左右兩旁的 sensor 則負責於避障時延牆邊行走來感測是否離開障礙物的資訊來源，於避障流程裡，分為五階段，如下：

1. 偵測前方障礙物：當自動巡航啟動時，自走車會自動開始前進，同時前方超音波不停的開始偵測，當偵測到前方障礙物小於 80 公分時停止前進。
2. 向右閃避障礙物：於此階段，向右轉 90 度，並繼續前進，前進的同時，不停的偵測左方的超音波 sensor，當偵測到障礙物距離超過 100 公分時，表示已離開障礙物，則停止前進。
3. 離開障礙物：此階段雖已離開障礙物，但必須在回覆原先的航道，故於此先左轉 90 度，完成後繼續前進，前進的同時不停偵測左方的 sensor，當偵測到障礙物距離超過 100 公分時，表示離開了障礙物，則停止前進。
4. 歸回原航道：這時算是離開障礙物，但必須回原前進路線，故開始前進一段路後停止，向右轉 90 度。
5. 繼續前進：此階段已恢復原先的航道，開始前進。

在使用者操作介面，Client 程式是採用 Java 撰寫，而 GUI Library 是使用 SWT，由於 SWT 是直接呼叫 OS 的繪圖 API，所以在執行時速度較 AWT 或 Swing 更快。Client 程式相較於自走車程式而言，其架構較為簡單，因大部分的程式邏輯皆在自走車，而 Client 主要是要與使用者有互動，並將車體實際畫面及 sensor 資料呈現在操作介面上。使用者操作介面如圖 5 所示，其介面上的操作如下：

- a. 行進控制：可控制自走車前進、後退、左轉、右轉、停止。
- b. 自動導航：當自動導航的 Check Box 被勾取後，車體會依照目前的行進方向為預設角度，於行進的同時，有導航狀態可以看出目前的執行進度。
- c. 攝影機視角控制：攝影機架設於旋轉平台，我們可以

透過此上下左右按鈕來控制攝影機的視角。

d.機械手臂控制：於車體前方的機械手臂可以用來抓取小形物體，透過操作介面，可以控制前方爪子的開合來夾取物品，垂直及水平的控制則可以決定抓取角度。

e.即時感測器資料：如要取得遠方的感測資料，只需要將 ON/OFF 的 Check Box 打勾，遠端即會進行量測，並將資料回傳，其中資料包括溫溼度、電子羅盤數值、前方超音波數值、左方超音波數值、右超音波數值。

f.設定：在設定區裡，可以控制是否要連線至車體或攝影機，並可看出目前的連線狀態。

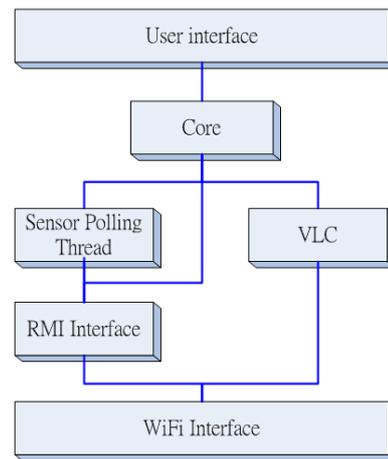


圖 5 使用者介面架構圖



圖 5 使用者操作介面

程式整體功能由 Core Module 來掌握，如圖 6，當 User Interface 有動作產生時，由 Core Module 將 UI 的動作進行判斷，如為操作自走車的要求時，就會透過 RMI 將指令送至遠端的自走車，遠端的自走車在依據指令來執行。當 UI 的 Real time Sensor Data 的功能開啟後，Sensor Polling Thread 會開始下指令要求遠端的 sensor 開始量測，並將資料寫入 Data Exchange Module，Client 端只要定時讀取 Data Exchange Module 並將資料顯示在介面上。視訊影像功能方面，在按下連線攝影機按鈕後，將會開啟一個 session，並要求自走車的 VLC Server，立即擷取 Web Cam 的畫面並壓縮後回傳。

#### 4. 系統整合與測試

本章節將實際測試自走車的性能，實驗環境如圖 7 所示，本研究使用一個無線基地台 (Access Point, AP) 將 SSID (無線基地台名稱) 設為 robot，自走車及使用者電腦的 IP 分別設為 192.168.11.1 及 192.168.11.5，由於兩個 IP 屬於相同私有 IP 網段，所以只需要透過一個 AP 就可以進行通訊，將來如要透過 Internet 來連線時，只要將 IP 設為公用 IP，中間 routing 由 ISP (網際網路服務提供者) 負責，如此可實現遙控遠在異地的自走車。

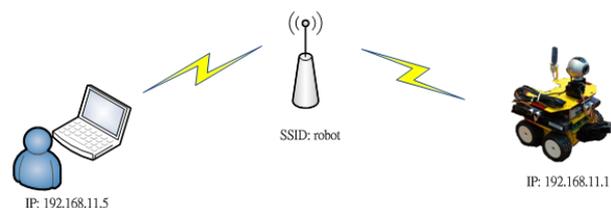


圖 6 測試架構圖

運轉測試分為兩個場景，兩個場景都分別測試手動搖控、自動避障功能以及不同環境下所感測器所測量的數值。由於文章裡無法表示動態的特性，如欲觀看測試過程，請至中央大學高速通訊及計算實驗室網站觀看測試影片，超連結為 [http://axpl.csie.ncu.edu.tw/new\\_version/wmv/Auto\\_Car\\_Demo.wmv](http://axpl.csie.ncu.edu.tw/new_version/wmv/Auto_Car_Demo.wmv)。

#### 實驗一：

實驗場地於元智大學六館一樓，並測試於行進中遭遇展示櫃所進行的自動避障，並於此進行手動遙控操作測試。實驗場地實景如圖 8。於自動避障測試中，有兩個障礙物連接在一起，一個是建築物支柱，另一個為文物展示櫃，其長寬分別為 75cm×75cm、60cm×60cm，當自走車的前方超音波感測器偵測到障礙物距離小於 80

公分時，會開始自動避障。



圖 8 實驗場地 1 之實景

於測試當時由攝影機可清晰的將畫面傳回使用者介面，並可取得感測器於當時環境所量測到的資料，實際畫面如圖。



圖 70 實驗場地 2 之實景

## 5. 結論

在整個研究實作裡，從一開始的硬體選擇到軟體的設計，到最後的整合，最困難的莫過於整合上的問題，因為有各種硬體有其特殊的控制方式，所以在選購硬體時都要非常注意其特性，以及將來成功整合的可能性，在軟體的部份，一開始就必須有良好的架構，其必須具備穩定性、擴充性、模組化，如開始的系統分析沒有做好，以及考慮不夠周延時，在開發到後期時一定會出現很多問題，導致必須重新翻修大部分的程式碼，本研究的軟體架構在一開始設計都出就考慮到未來的整合及擴充，所以並沒有太大的問題。

在軟體方面，由於許多的動作都要求即時，不論是自走車的控制、影像的傳輸、sensor 的量測與資料回傳，在這些要求裡，於設計上就設計相當多的執行緒來分工合作，在各個執行緒之間的通訊及資料交換等問題，在架構裡的 Data Exchange Module 擔任資料中心的角色，此設計的優點，讓日後不增加的執行緒可以更容易的取得資料，且更容易整合。



圖 9 實驗一使用者操作介面

## 實驗二：

實驗二場地於元智大學五館五樓，並測試於行進中遭遇紙箱，所進行的自動避障測試，並於此進行手動遙控操作測試。實驗場地實景如圖 10。於自動避障測試中，將遭遇三個紙箱所堆積成的障礙物於行進路徑中，其長寬分別為 86cm×30cm，當自走車的前方超音波感測器偵測到障礙物距離小於 80 公分時，會開始自動避障。於測試當時由攝影機可清晰的將畫面傳回使用者介面，於畫面裡可看到前方有三個紙箱的障礙物，並可取由感測器取得當時環境所量測到的資料。

## 參考文獻

- [1] J. Bruck, J. Gao and A. Jiang, "Localization and Routing in Sensor Networks by Local Angle Information," *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'05)*, pp. 181-192, Illinois, USA, May 2005.
- [2] A. Mainwaring, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pp. 88-97, Atlanta, USA, August 2002.
- [3] Y. Ohta, M. Sugano, and M. Murata, "Autonomous

Localization Method in Wireless Sensor Networks,” *Proceedings of the third IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNs’05)*, pp. 379-384, Osaka, Japan, March 2005.

- [4] K. Sohrabi, J. Gao, V. Ailawadhi and G. J. Pottie, “Protocols for Self-Organization of a Wireless Sensor Network,” *IEEE Personal Communications*, Vol. 7, pp. 16-26, October 2000.
- [5] S. Tilak, N. B. Abu-Ghazaleh and W. Heinzelman, “A Taxonomy of Wireless Micro-Sensor Network Models,” *ACM Mobile Computing and Communications Review*, Vol. 6, pp. 28-36, April 2002.
- [6] K. Holger, and W. Andreas, “Protocol and Architectures for Wireless Sensor Networks,” Wiley-Interscience Press, USA, 2005.
- [7] F. Zhao and L. Guibas, “Wireless Sensor Networks: An Information Processing Approach,” Morgan Kaufmann Press, USA, 2004.