

14. 無人割草機器人之電控系統研製

劉昭恕

國立高雄應用科技大學 機械工程系助理教授

摘要

無人割草機器人是一種具備自動割草功能之服務型機器人。在無人的狀態下，可在周圍佈上邊界線的區域中自動割草，當電源為弱電狀態時，可依邊界線自動搜尋路徑回去充電埠補充電源。此外，無人割草機器人甚至可以在有障礙物散佈的區域中割草，而碰撞感測器則可以確保當其碰到障礙物時，得以正確地離開障礙物。當工作進行至割草模式時，無人割草機器人可立即切換成割草模式來進行割草工作，當工作中偵測到弱電狀態時，無人割草機器人就會停止割草工作，切換到回程充電的模式，依佈線搜尋路徑回到充電埠充電，待完成充電後再繼續其未完成的割草工作。在上述的所有功能動作中，感測器也扮演著相當重要的角色，而無人割草機器人需要的感測訊息包括馬達位置訊息、無人割草機器人方位偵測訊息、障礙物偵測訊息、弱電訊息及割草判斷訊息等。本計劃乃依廠商的實際需要，針對無人割草機器人之所有電控系統的硬體及操控策略來加以開發，設計一組全新且屬於廠商自我版本之無人割草機器人之電控系統。

關鍵字：無人割草機器人、自動割草、電控系統

一、前言

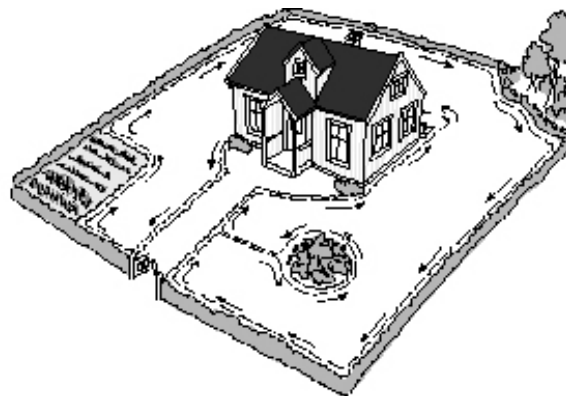
無人割草機器人是一種具備自動割草功能之服務型機器人(圖一)。在無人的狀態下，可在周圍佈上邊界線的區域中自動割草，當電源為弱電狀態時，可依邊界線自動搜尋路徑回去充電埠補充電源，至於割草高度是可以事先加以設定的。



圖一、無人割草機器人

此外，無人割草機器人甚至可以在有障礙物散佈的區域中割草，而碰撞感測器則可以確保當無人割草機器人碰到障礙物時，得以正確地離開障礙物，至

於割草區域佈線的範圍及形狀是可以任意規劃的(圖二)。當工作進行至割草模式時，無人割草機器人可立即切換成割草模式來進行割草工作，當工作中偵測到弱電狀態時，無人割草機器人就會停止割草工作，切換到回程充電的模式，依佈線搜尋路徑回到充電埠充電，待完成充電後再繼續其未完成的割草工作。而所有的工作都是可事先規劃及設定，當發生無法排除之狀況時，此機器人亦會發出警訊以警示使用者，一段時間未排除時，則將自動斷電，待攜回充電埠重新啟動後才恢復正常工作狀態。



圖二、自動割草機器人工作區示意圖

也就是說，無人割草机器人的主要任務為儘可能以最短的時間完成每次的割草任務。然而，受限於割草區域狀況的不確定性，無人割草机器人就必須依各種狀況來做動作。原則上，完整的割草運動模式大體可以區分成幾種：(1) 依循邊界線之循軌控制，(2) 正常割草狀態之模式跟隨運動控制，(3) 碰到障礙物之障礙物閃避運動控制，(4) 超越邊界線時之轉向運動控制，(5) 以上幾種狀態接續或同時發生之運動控制。因此，若不能將上述之運動控制模式適切的設計，也就是運動控制器的設計，將會嚴重影響無人割草机器人的整體性能表現。

在上述的所有功能動作中，也可以發現感測器扮演著相當重要的角色，而無人割草机器人需要的感測訊息包括馬達位置訊息、無人割草机器人方位偵測訊息、障礙物偵測訊息、弱電訊息及割草判斷訊息等。這些感測器中以馬達位置及割草邊界感測器最為重要，而且目前市面上這兩類的感測器價位也非常高，這對產品成本競爭力而言其影響是很大的。

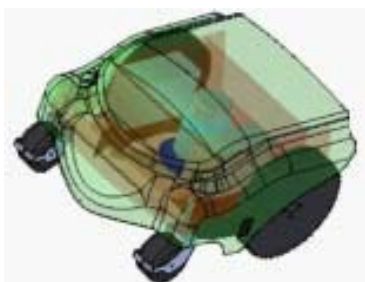
由於計劃委託廠商為目前世界上供應此類機器人的生產廠家之一，且已規劃開發此類自我版本的相關產品，因此，為設計出一符合功能需要，甚或是更勝既有產品的新版本設計，且得以擁有自我核心的電控系統及使其具備優異的工作性能，是目前委託廠商急欲完成的工作。本計劃乃依廠商的實際需要，針對無人割草機器人之感測器及電控系統的軟、硬體來開發，設計一組全新且屬於廠商自我版本之無人割草機器人系統。

二、研究目的

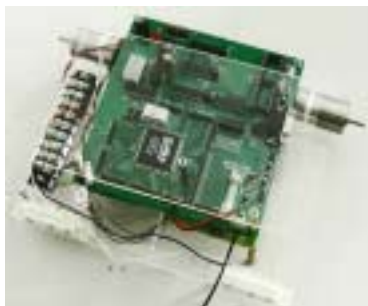
無人割草機器人系統，主要由運動用馬達、割草用馬達、定位感測器、方位感測器、充電模組、控制器及人機界面所組成，且搭配機構、軟體設計及路徑規劃後可達到自動搜尋回程充電、割草、閃避障礙物及最佳割草模式的目的。

因此，針對上述之功能要求，在硬體規劃上，必須具有多顆馬達驅動、控制及感測等設計，尤其是各種工作模式及路徑之規劃設計，不管是硬體腳位數、電位轉換及系統頻寬等，都需要加以設計及驗證。

本開發計劃的目標在於建立一屬於廠商自我版本之無人割草機器人之軟、硬體系統(可見下圖三及圖四)，同時滿足上述之各項功能及動作，其中包括操控軟體的設計、路徑的規劃、控制器設計及相關馬達驅動硬體的設計開發等。藉此，希望得以取代目前現有之無人割草機版本，推出屬於廠商自我版本之全新的無人割草機器人。



圖三、無人割草機器人之機構外觀圖



圖四、無人割草機器人之控制器模組雜型

有關廠商自我版本之感測器模組，包括馬達位置感測器及割草機器人割草邊界感測器。而馬達位置感測器如圖五所示，由偵測馬達連結之光盤之光匝數來決定馬達的角度，此感測器也就是所謂的 Encoder。



圖五、Encoder

有關無人割草機器人割草邊界的感測器，乃藉由偵測割草邊界線所發出之磁場來決定離邊界的距離，如此，可避免機器人超出割草邊界，也可讓機器人在弱電時，循跡回到充電埠充電。目前市售的磁感應感測器如圖六所示：



圖六、磁場感測器

除了感測器外，其界面電路設計及訊號解譯也相當重要，因此，本計劃也針對上述之感測器來加以設計，以滿足委託廠商自我產品開發的需求，同時達到自我核心技術掌控的目的。

此外，基於無人割草機器人之運動模式的多樣性需要，本計劃也對於各種運動模式之控制加以規劃設計，讓無人割草機器人員備優異的運動操控性能。

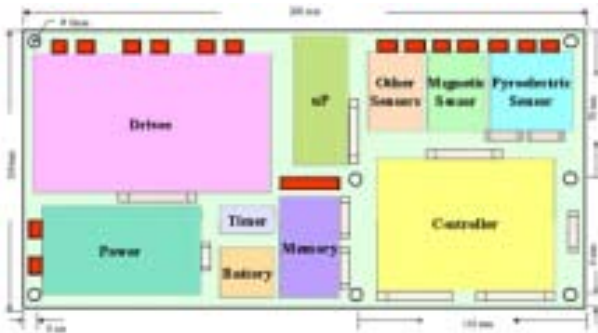
三、研究方法

本開發計劃在於建立一屬於廠商自我版本之無人割草機控制器之軟、硬體系統，也就是核心控制器的設計，主要功能有馬達驅動控制、馬達定位及速度控制、感測器訊號處理及訊息識別、人機界面設定等，以及各種工作模式的設計，包括循跡回程充電、障礙物閃避及割草路徑規劃等。

整個設計程序，將先行完成馬達的選取及驅動電路的設計，同時決定感測器的種類及訊號處理，然後依功能需要來規劃硬體的腳位及其定義，也因

此可以決定出整個控制器的大概硬體架構。在完成初步硬體的輸出入功能設計後，接著依規格要求來做韌體的流程設計及動作模式的規劃，也就是完成整個系統工作的狀態機圖的規劃及設計。

整個控制器硬體架構，基於上述之硬體規格需求，經初步功能驗證及控制選定後，基本上整個硬體包括單晶片、CPLD、記憶體及輸出入界面的電路設計，同時考量訊號的放大及濾波要求，以及電流放大之驅動電路設計。有關整個系統中之控制器的方塊圖如圖七所示：



圖七、系統之控制器方塊圖

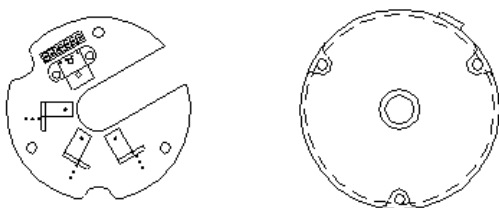
而對於Encoder 及割草邊界磁感測器，其設計方法及原理將如下所述：

(1) Encoder：其原理乃藉由光盤旋轉，由光匝產生 AB 相位的訊號，其相位差為90 度。其波形如圖八所示。



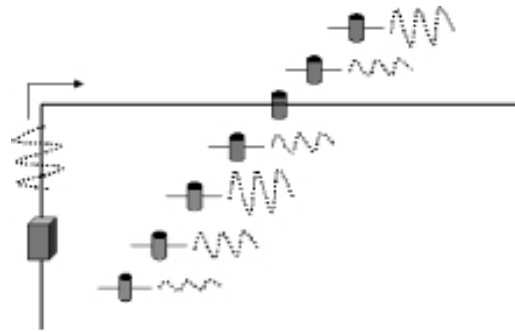
圖八、Encoder 之AB 相波形圖

有關Encoder 機構部份之初步設計構想，如圖九所示：



圖九、Encoder 之機構圖，左邊為電路板，右邊為光盤

(2) 磁感應感測器：其原理乃在割草邊界線中通上交流訊號，再以適當繞組的線圈來接收感應磁場以產生感應電壓，只要將感應線圈上的感應電壓適當的處理，即可得到因距離變化而產生之電壓大小變化的有效訊號，而相關原理可見圖十。



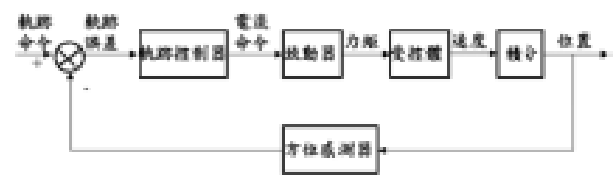
圖十、磁感應器之磁感應原理示意圖

本計劃將依上述之構想加以修正使感測器其動作更為確實。同時設計相關界面電路來轉換及修正訊號，使其得以應用在無人割草機器人之各項功能上。

而有關機器人的控制系統，大致可分成以下三大部份：

1. 運動之控制迴路

機器人系統在行進間，就形成一運動之控制迴路(見圖十一)。機器人系統將依所指示之軌跡命令來加以控制，原則上，將儘可能將運動的行為控制平順，因此，必須搭配適當的方位感測器及軌跡控制器。



圖十一、運動之控制迴路

2. 工作之位置控制迴路

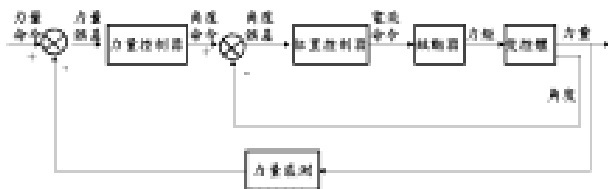
機器人系統在工作時，就形成一工作之位置控制迴路(見圖十二)。機器人系統將依所指示之工作角度命令來加以控制，原則上，也將儘可能將工作運動的行為控制平順，因此，必須搭配適當的位置控制器。



圖十二、工作之位置控制迴路

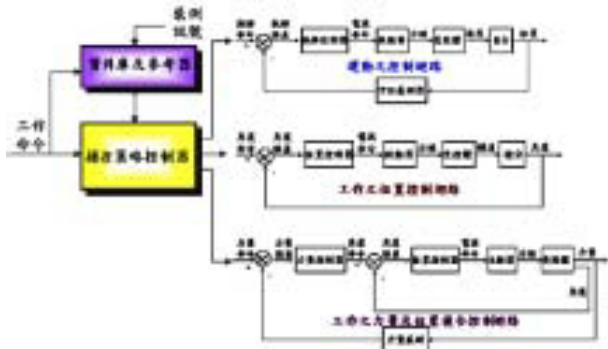
3. 工作之力量及位置複合控制迴路

機器人系統在工作時，若有夾持的需要且必須保持適當的力量時，就形成一工作之力量及位置複合控制迴路(見圖十三)。機器人系統將依所指示之工作力量命令來加以控制且移動角度，原則上，也將儘可能將工作運動的行為控制平順，因此，必須搭配適當的力量感測器、力量控制器及位置控制器。



圖十三、工作之力量及位置複合控制迴路

若將上述之三個控制系統合併，將可得到完整之控制系統方塊圖(見圖十四)。

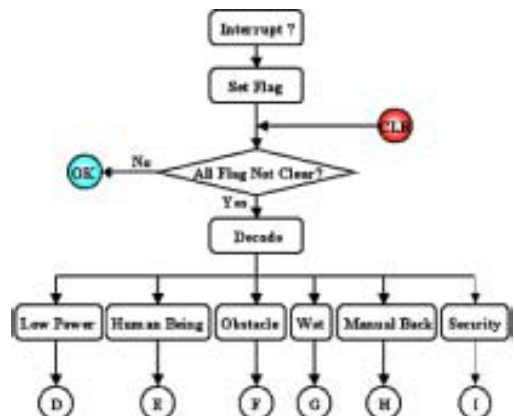
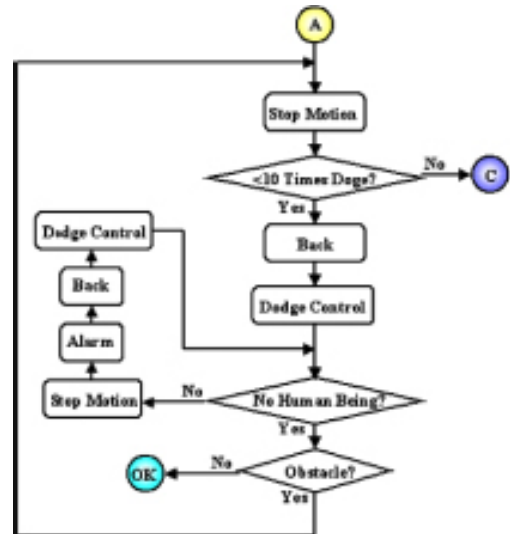
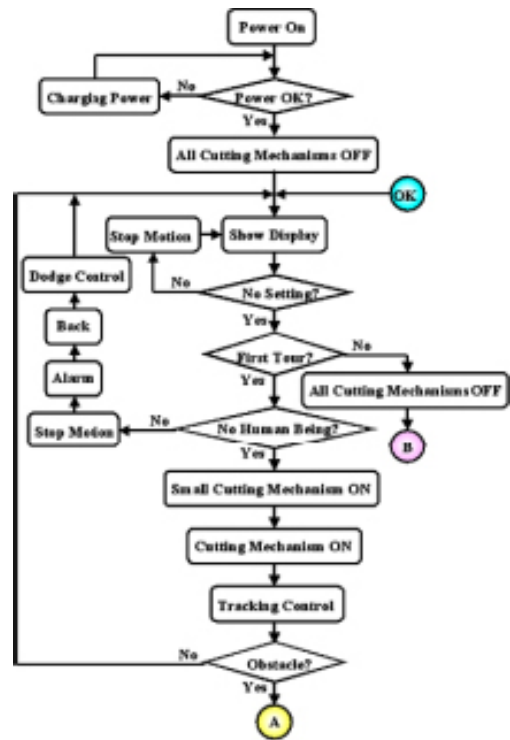


圖十四、完整之控制系統方塊圖

由圖十四可知，要整個機器人控制系統控制順暢，就必須搭配資料庫參考器及操控策略控制器，如此，才有可能將整機器人控制系統的性能表現出來。

整個設計程序，將先行依動作要求來做控制流程的設計及動作模式的規劃，也就是完成整個系統工作的狀態機圖的規劃設計。接著搭配整個控制器硬體的架構，包括單晶片及FPGA，做韌體及硬體描述語言的綜合設計。

有關整個無人割草機器人之運動控制之操控狀態機的主要流程，依實際需要及硬體規劃所設計，其運動、割草、警報及閃避障礙物之動作的安排及判斷將如下所示：

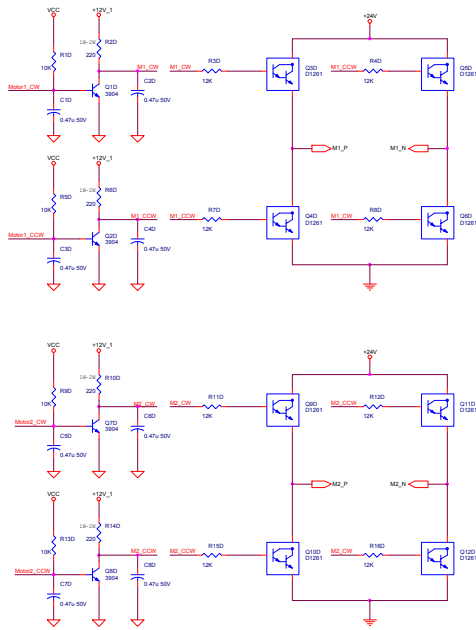


圖十五、運動控制之流程圖

四. 系統設計

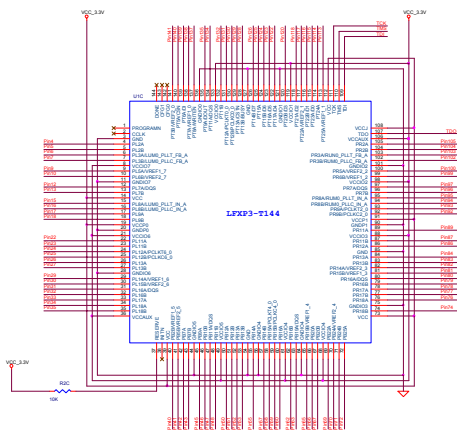
本計劃乃設計一無人割草機器人之感測器及電控系統的軟、硬體，依規劃分別設計馬達驅動電路、控制器電路及感測器電路，也分別加以驗證其功能及電氣特性。

有關馬達驅動電路，由於所選定的馬達為直流馬達，同時考量電氣的穩定及反應速度，因此，採用H-bridge 及電晶體的組成電路來驅動馬達，而相關的驅動電路如圖十六所示：



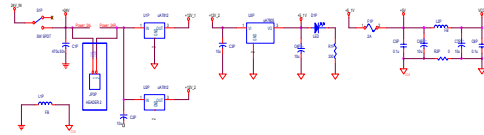
圖十六、馬達驅動電路

有關控制器部份，為了硬體腳位的數量及硬體反應頻寬的要求，同時考量保密性及使用彈性，因此，採用硬體描述語言(VHDL)及CPLD 來加以設計實現，相關的電路圖如圖十七所示：

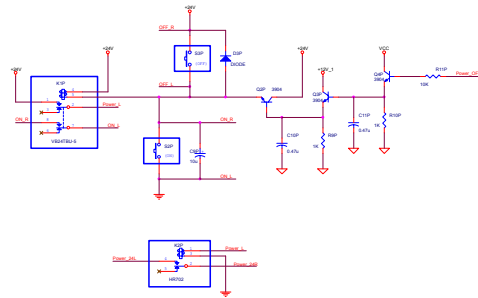


圖十七、控制器核心硬體及電路

此外，由於馬達驅動電壓為24VDC，而且因電路設計需求，需要使用12VDC、5VDC 及3.3.VDC各種不同的電壓源，且在電源啟動及關閉部份，因功能需要手動及自動的功能設計，因此，在電源電路部份，必須加以設計，而相關電路如圖十八及圖十九所示：

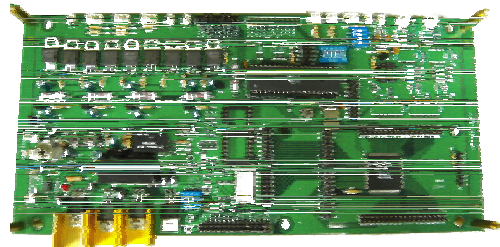


圖十八、電源規劃電路



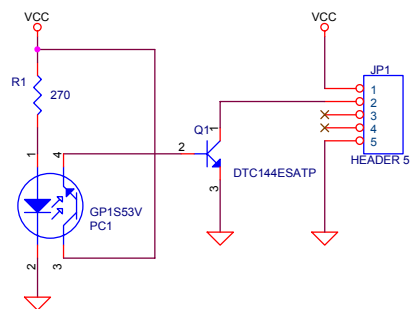
圖十九、電源控制電路

各部份電路經設計驗證後，再經佈線走線及備料完成，且經板廠洗板後，再次地焊接驗證後，及可得圖二十之電控模組電路板成品。



圖二十、電控模組之電路板成品

有關馬達Encoder 電路，經設計及驗證後可得圖二十一所示之電路圖：



圖二十一、Encoder 之電路圖

將實現後，可見圖二十二之成品圖及圖二十三之波形圖，從波形中可見A及B相位差為90度。



圖二十二、Encoder 之成品圖



圖二十三、Encoder 之波形圖

有關割草邊界之磁場感應感測器部份，感應磁場用之感應線圈裝置可見圖二十四，而感應線圈主要裝置於機器人前下端位置，可見圖二十五所示。

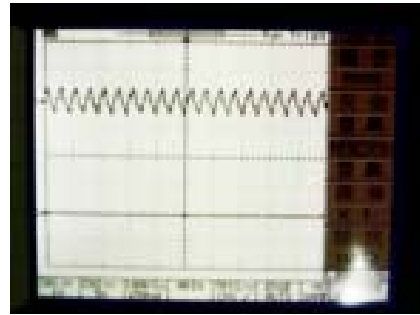
在未做任何電氣訊號處理之磁場感應波形，可見圖二十六所示。



圖二十四、感應磁場用之感應線圈裝置

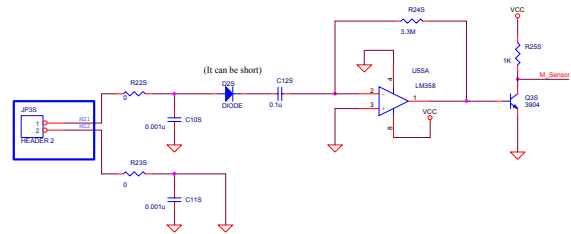


圖二十五、磁場感應線圈裝置位置

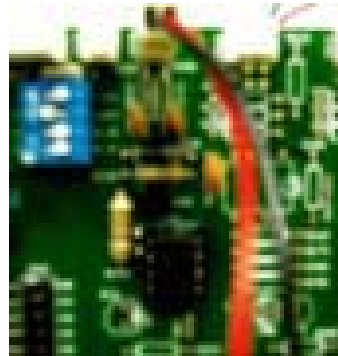


圖二十六、磁場感應波形

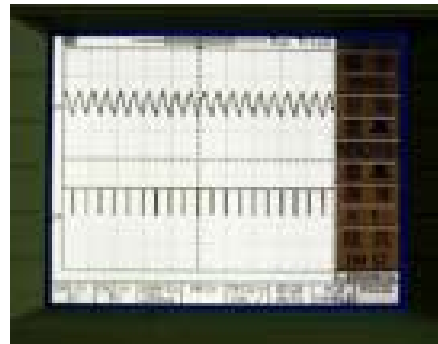
為了得到數位化之距離對磁場感應變化數值，於是開發設計一電路(圖二十七)且將其硬體電路實現(圖二十八)，各距所測得之實際波形及訊號處理後之數位化波形，可見圖二十九至圖三十二所示之結果。



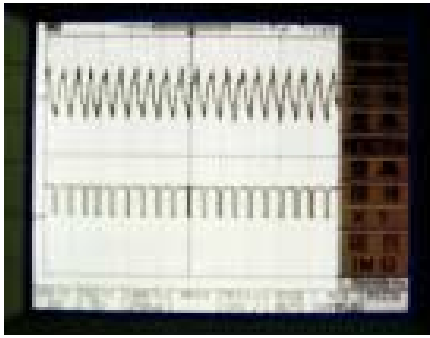
圖二十七、磁場感應之訊號處理電路



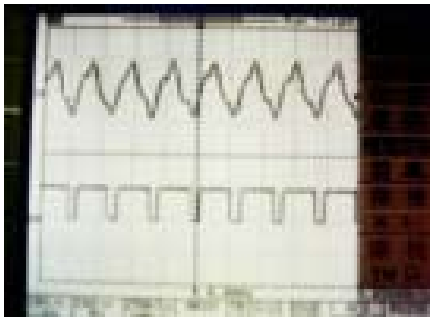
圖二十八、磁場感應模組之電路實體



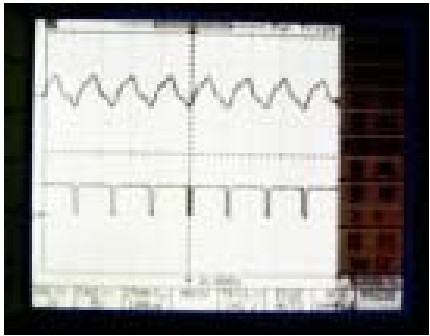
圖二十九、磁場感應及訊號處理後之波形



圖三十、磁場感應最強及訊號處理後之波形



圖三十一、感應訊號強時之訊號放寬波形



圖三十二、感應訊號弱時之訊號放寬波形

此外，無人割草機器人之硬體規劃及運動控制器之動作要求乃動作流程來設計其核心硬體的程式，核心硬體乃以Lattice 所出之FPGA(如圖三十三)來執行，程式乃以硬體描述語言(VHDL)來加以設計。



圖三十三、控制器核心硬體

在程式中，分別設計且整合包裝計時器、感測器處理及馬達控制等，分述如下：

1. 計時器設計單體：

```
entity Timer is
    port(
        clk_in      :in  std_logic;
        clk_out     :out std_logic;
        clk_2ms    :out std_logic;
        clk_8ms    :out std_logic;
        clk_16ms   :out std_logic;
        clk_20ms   :out std_logic;
    );
end Timer;
```

2. 感測器處理程序：

```
Sensor_detector:process(Rst_a, clk)
begin
    if Rst_a='0' then
        sensor_detect <= '0';
        sensor_count <= 0;
    elsif clk'event and clk = '1' then
        if Sensor = '0' then
            sensor_count <= sensor_count + 1;
        else
            sensor_count <= 0;
        end if;
        if sensor_count >= 300 then
            sensor_detect <= '1';
        else
            sensor_detect <= '0';
        end if;
    end if;
end process Sensor_detector;

Distance_calculate:process(Rst_a, clk, clk_2ms)
begin
    if (Rst_a='0' or clk_2ms='0') then
        distance_count <= 0;
    elsif clk'event and clk = '1' then
        if Sensor = '0' then
            distance_count <= distance_count +
        else
            distance_count <= distance_count;
        end if;
    end if;
end process Distance_calculate;

Distance_value:process(Rst_a, clk_2ms)
begin
    if Rst_a='0' then
        distance_value1 <= 3000;
        distance_value2 <= 3000;
    elsif clk_2ms'event and clk_2ms = '0' then
        distance_value2 <= distance_value1;
        distance_value1 <= distance_count;
    end if;
end process Distance_value;
```

3. 馬達控制程序：

```
motor_C:process(Rst_a, clk)
begin
  if Rst_a = '0' then
    state_now <= 0;
    Rst_wait <= '1';
    CW_dir <= "11";
    Light <= '0';
    Buzzer_Gate <= '0';
  elsif clk'event and clk='1' then
    case state_now is
      when 0 =>
        Motor1_CW_tmp <= 'Z';
        Motor1_CCW_tmp <= 'Z';
        Motor2_CW_tmp <= 'Z';
        Motor2_CCW_tmp <= 'Z';
        Rst_wait <= '1';
        state_now <= 1;
        Light <= '0';
        CW_dir <= "11";
        Buzzer_Gate <= '0';
      when 1 =>
        Motor1_CW_tmp <= 'Z';
        Motor1_CCW_tmp <= 'Z';
        Motor2_CW_tmp <= 'Z';
        Motor2_CCW_tmp <= 'Z';
        Rst_wait <= '0';
        Light <= '0';
        CW_dir <= "11";
        Buzzer_Gate <= '0';
        state_now <= 1;
        if clk_4s = '1' then
          state_now <= 2;
        end if;
      when 2 =>
        Motor1_CW_tmp <= '0';
        Motor1_CCW_tmp <= 'Z';
        Motor2_CW_tmp <= '0';
        Motor2_CCW_tmp <= 'Z';
        Rst_wait <= '1';
        Light <= '0';
        CW_dir <= "01";
        Buzzer_Gate <= '0';
        state_now <= 2;
        if sensor_detect = '1' then
          state_now <= 3;
          Light <= '1';
        end if;
      when 3 =>
        Motor1_CW_tmp <= 'Z';
        Motor1_CCW_tmp <= 'Z';
        Motor2_CW_tmp <= 'Z';
        Motor2_CCW_tmp <= 'Z';
        Rst_wait <= '1';
        Light <= '1';
        CW_dir <= "11";
        Buzzer_Gate <= '1';
        state_now <= 4;
    ...
  when 11 =>
    Motor1_CW_tmp <= 'Z';
    Motor1_CCW_tmp <= 'Z';
    Motor2_CW_tmp <= '0';
    Motor2_CCW_tmp <= 'Z';
    Rst_wait <= '1';
    Light <= '0';
    CW_dir <= "10";
    Buzzer_Gate <= '1';
    state_now <= 11;
    if (distance_value1=0 and distance_value2=0) then
      state_now <= 7;
    elsif distance_value1 > distance_value2 then
      state_now <= 12;
    end if;
  when others =>
    Motor1_CW_tmp <= 'Z';
    Motor1_CCW_tmp <= 'Z';
    Motor2_CW_tmp <= 'Z';
    Motor2_CCW_tmp <= 'Z';
    Rst_wait <= '1';
    Light <= '0';
    CW_dir <= "10";
    Buzzer_Gate <= '1';
    state_now <= 7;
  end case;
end if;
end process motor_C;
```

將其架設於無人割草機器人機構上(見圖三十四)，即可啟動及驗證所有之功能及動作。



圖三十四、無人割草機器人之成品圖

該計劃之無人割草機器人之感測器、電控模組及控制器模組的工程雛型之軟、硬體已開發完成，各項軟硬體之初步功能驗證也已完成，只待廠商最後要求之控制模式確認後，在軟體上稍加修改即可。若經產品包裝及試產後，再配合機構組裝即可量產銷售，相信在不久的將來，就可看見該款機器人在市面上大量銷售。

五、結論

無人割草機器人之感測器、電控系統設計及開發之工程雛型已經順利完成。本計劃所設計之軟、硬體皆為自我設計開發，尤其在系統架構及控制核心部份都是有相當高之技術門檻及價值。因此，不僅可大幅降低廠商之產品成本而且提昇廠商之技術水準及確保該核心技术之保密。

而本計劃所設計之感測模組也可應用在各種需要伺服控制及磁場感應之場合，對於未來的相關產品之感測模組設計，有很大的幫助。此外，本計劃之運動控制器規劃及設計皆有考慮未來的功能擴充及發展，因此，該硬體可以依需要適當地調整，不管在硬體腳位上還是新增功能皆可輕易配合。尤其是，該控制器架構可做為各種應用場合之核心控制器，對於未來的相關產品之控制器設計，也奠定了穩固的基礎。該計劃之完成也衍生許多想法及技術的專利，目前該合作廠商正進行該項之工作以取得專利權，同時確保產品銷售之順暢。在服務型機器人的領域中，本計劃之執行不僅為廠商帶來商業的契機，同時也建立該領域之機器人的核心技术，相信對於未來一系列的機器人開發計劃，將可順利的進行及完成。

參考文獻

1. Scott Mackenzie, The 8051 Microcontroller, Prentice Hall, 1998.
2. 溫植燁編著·8051 單晶片實務與應用 - 使用Keil C·文魁資訊股份有限公司·2004。
3. DE2 Development and Education Board User Manual, Altera Corporation, 2006.
4. DC Motors, Speed Controls, Servo Systems, including Optical Encoders, An Engineering Handbook by Electro-Craft Corporation, Hopkins, MN, Fifth Edition, 1980.
5. 林永裁、鄭美珠·微電腦I/O 控制界面設計實務·第3 版·全欣資訊圖書股份有限公司·1995。
6. M. K. Gupta, "Implementation of Digital PID Controller on a Single FPGA Chip Using VHDL" , A Deemed University.
7. 林柏辰、陳慶逸·VHDL 數位電路實習與專題設計·文魁資訊·2003。
8. KiltSteve, Advanced FPGA Design - Architecture, Implementation, and Optimization, WILEY, 2007
9. 張雄偉、陳亮、徐光輝編著·DSP 晶片的原理與應用·五南圖書出版公司·2004。
10. 范逸之、江文賢、陳立元·C++ Builder 與RS 232串列通訊控制·文魁圖書·2002。
11. Developer's Kit for Texas Instruments DSP User's Guide, MathWorks Inc. February 2001
12. Yasuhiko Dote, Servo Motor and Motion Control Using Digital Signal Processors, Prentice Hall, New Jersey, 1990.
13. 江昭暄、李俊奇、吳偉賢、范丙林 譯者·感測電路·高立圖書有限公司·1997。