

# 無線跳舞機器人

## 一、軟、硬體需求:

在此使用 **RF24G** 無線傳輸模組做傳輸控制，**RF24G** 的優點有可一對多發送資料，且可透過程式切換發送/接收模式做雙向傳輸

### 1、軟體:

InnoBASIC Workshop 或 InnoBASIC Workshop2

### 2、硬體:

硬體方面分為機器人及控制端兩部分

#### i. 機器人端:

Robotinno 機器人，RF 模組一個。

RF 模組可裝置在機器人背後並以 CommandBus 線與 ServoCommander16 連接。



#### ii. 控制端: 以下列出 a.b 兩種方案

(a) RfCommander 一個，電腦一台。

此方案可透過電腦 Debugin 指令直接對 RfCommander 下命令，RfCommander 具有兩個 RF 模組，可以同時做發送與接收動作。



(b) CommandBoard 一個，BC1 一個，RF 模組一個，KeyPad 一個，LCD2 X 16 一個，鎳氫電池 6V/2200mAH 一個。

此方案使用多個模組整合成一個控制平台，將程式存入 BC1 中，用 KeyPad 鍵盤模組下命令，LCD 顯示指令狀態，優點為方便攜帶。



## 二、動作編寫:

在設計動作前必須先把音樂決定好，定好音樂後就可以搭配音樂進行編舞的動作，舞步只要跟音樂搭配的好整體感必定會精采許多，所以在選取音樂上也是很很重要的一環。

音樂建議可以挑選節奏清楚明快的音樂，節奏強烈的音樂若機器人動作能精確的對在各重節拍上，動作不需要非常困難，表演起來也是相當精彩的。

軟硬體都準備好，音樂也選定後就準備進入動作編輯的部分，動作主要是由Frame所構成，透過程式去將每個Frame讀出做出一連串的動作；使用 innoBASIC Workshop或innoBASIC Workshop2 軟體中的動作編輯器便能輕鬆編輯每個動作。若還在使用innoBASIC Workshop的玩家們建議可到利基科技網站(<http://www.innovati.com.tw/>) 檔案下載區內下載innoBASIC Workshop2 軟體使用，以下就以innoBASIC Workshop2 軟體為主做說明。

動作不一定要一次寫到底，可以根據音樂段落將動作分成數段，寫成副程式，一般音樂大多可以分成前曲、主歌、副歌，就可以將動作寫成三段甚至更多段副程式，好處是不會因為要修改某一段動作而需要動到全部程式，只需要修改某一段副程式即可，並不會影響到其他副程式。

準備開始編寫動作，用 USB 線將電腦與機器人連接好(調整動作時注意勿拉扯到 USB 線，USB 接頭較脆弱容易被扯掉)，將機器人接上電，點選工作-->動作編輯器，進入後點選十六軸全身兩足機器人，初始設定會將模組 0 所有伺服機定位在 1500，開始編輯動作前必須先將微調值調好，點選右上角設定微調值按鈕，利用微調將機器人調正，在微調機器人的時候不要讓機器人伺服機有太大施力狀態，所以建議將機器人直立懸空微調，微調好後可以儲存一份檔案，方便以後進動作編輯器時不需再重新做微調，只需把微調檔呼叫出來即可。

接下來就要開始編及機器人動作了，下面就依照各步驟做詳加說明。

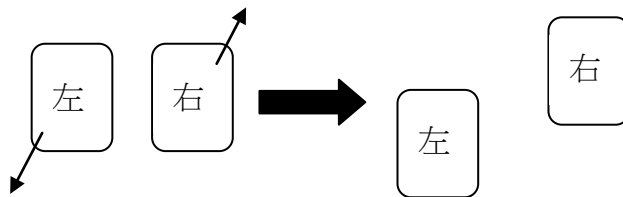
### 1、動作編輯:

進入動作編輯器並且微調值都調好後，回到動作編輯畫面，可以拉動對應的伺服機編號旁的拖曳方塊或直接輸入數值來調整伺服機位置，動作編輯器頁面如

下圖所示，設定伺服機動作速度，點選 Speed 或 Time 並輸入數值，在此建議使

用 Time，在寫程式時執行 Frame 要給的 Pause(程式暫停)時間會比較容易算，要剛好執行完這個動作後立刻接下一個動作只要將 Pause 的時間跟 Time 的時間設定一樣便可以，動作調好後點選模組方塊內之儲存便會將動作儲存在動作編號相對應之位置，動作編號為 0 便會儲存在 Frame0 之位置，按讀取後輸入動作編號就可以把該 Frame 叫出來。

動作設計上手部動作會比較容易，主要注意動作時手不會卡到身體；在設計腳部動作就需要注意重心平衡以及與地面摩擦問題，平衡沒做好就容易倒地，很容易讓機體摔壞或變形，快速的朝某個方向移動也容易造成重心偏移，設計上也需要注意，腳底若貼著地面移動會與地面產生摩擦力，摩擦力會形成機器人的位移，如下圖腳底貼著地面張開便會有右轉效果。在動作設計上面可以上各網



站搜尋有關機器人跳舞的影片做參考。

## 2、程式編寫:

在編動作時可以配合程式編寫，編好一段動作就可以用程式去跑過一遍，檢視動作有無要修改處，以下介紹程式流程。

在程式開頭必須設定模組

Peripheral Myser As servorunnerA @ 0

Servocommander16 模組編號設定為 0。Servocommander16 模組編號是內設不可更改的

我們需要對伺服機做微調，使用 `Myser.SetPosOffset`(伺服機編號，微調值)，程式如下

```
Sub Offset()  
    Myser.SetPosOffset(0,0)  
    Myser.SetPosOffset(1,0)  
    Myser.SetPosOffset(2,0)  
    Myser.SetPosOffset(3,0)  
    Myser.SetPosOffset(4,0)  
    Myser.SetPosOffset(5,0)  
    Myser.SetPosOffset(6,0)  
    Myser.SetPosOffset(7,0)  
    Myser.SetPosOffset(8,0)  
    Myser.SetPosOffset(9,0)  
    Myser.SetPosOffset(10,0)  
    Myser.SetPosOffset(11,0)  
    Myser.SetPosOffset(12,0)  
    Myser.SetPosOffset(13,0)  
    Myser.SetPosOffset(14,0)  
    Myser.SetPosOffset(15,0)  
End Sub
```

我們可以把呼叫 `Frame` 寫成一段副程式

```
Sub Runframe(Byval ID As Byte)  
    Myser.LoadFrame(ID)  
    Myser.runallservo()  
End sub
```

這段副程式可以將 `Frame` 讀出並且啟動所有伺服機，之後需要執行 `Frame` 時只需要呼叫 `Runframe(ID)`，其中 `ID` 代表要呼叫的 `Frame` 編號。

將存好的 `Frame` 讀出編成連串動作，我們可以將每段動作寫成一段副程式。

```
Sub Dance1()  
    Runframe(0)  
    Pause 500  
    Runframe(1)
```

```
        Pause 1000
        Runframe(2)
        Pause 700
    End sub
```

Pause 的時間可以設定與 Frame 之 Time 的時間相同，若 Pause 設定比 Time 還小會造成動作還沒到位就去執行下一個 Frame，若值大於 Pause 可能就會造成動作做完後停住等 Pause 時間到才繼續下一個動作，Pause 單位為 ms，所以其實差 50 左右(0.05 秒)是感覺不太出來的。

編寫主程式

```
Sub main()
    Offset()
    Dance1()
    Dance2()
End sub
```

將先前寫好的副程式呼叫出來，一個簡單的程式就這樣完成了

### 3、進階變化應用:

大部分的初學使用者可能都會把一個 Frame 中所有伺服機的 Time 都設定一樣，這樣動作會比較好寫沒錯，但是變化會比較少，其實我們可以把手跟腳拆開來編，甚至手部還可以拆成左手跟右手；例如 Frame1 的腳部是做一個蹲下的動作需要花 1000ms，Frame1、2、3、4 的手部做一個轉圈圈的動作 Time 都設為 250，透過程式就可以做到一邊蹲下手一邊轉圈圈的動作。程式碼可照下面方式：設定呼叫左、右手 Frame 的副程式，呼叫方式 Runhand(左手,右手)

```
Sub Runhand(Byval Lhand As Byte , Byval Rhand As Byte)
    Myser.LoadFrame(Lhand)
    Myser.run3servo(4,5,6)
    Myser.LoadFrame(Rhand)
    Myser.run3servo(8,9,10)
End Sub
```

設定呼叫腳部 Frame 的副程式

```
Sub Runleg(Byval Leg As Byte)
    Myser.LoadFrame(Leg)
    Myser.run10servo(0,1,2,3,7,11,12,13,14,15)
End Sub
```

## 設定動作

```
Sub main()
    Runleg(1)
    Runhand(1,1)
    Pause 250
    Runhand(2,2)
    Pause 250
    Runhand(3,3)
    Pause 250
    Runhand(4,4)
    Pause 250
End Sub
```

程式一進入執行一個需要 1000ms 的腳步動作，後面做呼叫手部的 Frame，手部動作一個為 250ms，執行 4 個花費時間 1000ms，剛好讓腳的伺服機能夠跑到定位，所以這段程式就可以做到一邊蹲下手一邊轉圈圈的動作。透過這種方式就可以使動作變得靈活順暢，也可以讓動作有更多的組合方式。

節拍對法，可以用 Pause 時間計算，如果 Frame0 要對應到音樂第 10 秒，那麼只要計算前面的 Pause 時間，將 Frame0 放在 Pause 時間加起來第 10000 的地方，編好一段動作就可以搭配音樂試跳一遍，檢察有沒有地方需要再做調整等；以此方法一段一段編排，最後再組合起來就好。

### 三、RF24G 程式編寫:

動作音樂都完成後，最後就要使用 RF 無線遙控了，下面就介紹簡單的 RF 程式使用。

#### 1、機器人端:

程式使用:將之前寫好的程式套用進來，再將寫好的動作副程式放至各 case 中，case 數量可依需求自由增加減少，若有使用多隻機器人，可在程式第 11 行的地方可修改機器人的編號，將 1 改成相對應之編號便可，後面的 255 在此設定為機器人共同編號，以便同時呼叫所有機器人。

```
Peripheral RF As RF24G @ 31 '設定 RF 模組 ID 碼為 31
Dim Rdatacheck As Byte '宣告變數 Rdatacheck 檢查資料編號用
Dim Rdata(2) As Byte '宣告陣列 Rdata 儲存接收之資料,Rdata(0)為機器人編號,Rdata(1)為動
做編號,Rdata(2)為資料編號
Sub main()
    RFRinital() '載入 RF 初始值
    RF.EnRxReadyEvent() '開啟接收完成事件
    Rdatacheck = 0 'Rdatacheck 設為 0
    Do
        If Rdata(2) <> Rdatacheck Then '檢查資料編號是否為 0
            Rdata(2) = Rdatacheck '將資料編號設為 0
            If Rdata(0) = 1 Or Rdata(0) = 255 Then '此行為設定機器人編號，將 1 改掉便可更換編號，255
                設為機器人共同編號
                Select Case Rdata(1) '選擇動作編號執行對應動作
                    Case 1
                    Case 2
                    Case 3
                    Case 4
                    Case 5
                End Select
```

```

        RF.ClrBuffer          '清除暫存資料
    End If
End If
Loop
End Sub
Sub RFRinitial()
    RF.SetMode(1)           '設定 RF 模式,0 為發送,1 為接收
    RF.SetCh(0)            '設定頻道為 0
    RF.SetRegCode(0)       '設定註冊碼為 0
    RF.SetRFID(0)         '設定識別碼為 0
    RF.Config()           '將設定的模式、頻道、註冊碼、識別碼下載至模組
End Sub
Event RF.RxReadyEvent()  '設定接收完成事件
    RF.GetArray(Rdata)    '將接收之資料存於陣列 Rdata 中
End Event

```

## 2、控制端:

此為 RCommander 之程式，程式可直接使用，程式有 2 處會因使用者設定需做調整，程式第 10 及 11 行之 RoboNum 及 MotionNum 為設定機器人數量及動作數量，此程式機器人數量預設 3 隻，動作預設 5 個，

```

Peripheral RFT As RF24G @ 2
Peripheral RFR As RF24G @ 3

Dim Tdata(2) As Byte      '宣告陣列 Tdata,Tdata(0)為機器人編號,Tdata(1)為動作編號,Tdata(2)為
資料編號
Dim RoboNum,MotionNum As Byte      '宣告變數設定機器人數量與動作數量
Dim TxReady As Byte
Dim x As Byte
Sub main()
    RFTinitial()          '載入 RF 模組發送端初始設定
    Tdata(2) = 0          '資料編號從 0 開始
    RoboNum = 3           '設定機器人有 3 隻
    MotionNum = 5        '設定動作有 5 個
Do
    Debug CLS            '清除終端視窗
    Debug CSRXY(1,1), "請輸入欲控制之機器人編號: "
Do
    Debugin Tdata(0)

```



```

If Tdata(0) > RoboNum Then '檢查輸入機器人編號是否在範圍內
    Debug CSRXY(1,1), "無此編號請重新輸入機器人編號: "
Else
    If Tdata(0) = 0 Then '輸入 0 則 Tdata(0) 為 255, 啟動所有機器人
        Tdata(0) = 255
        Debug "ALL"
    Else
        Debug Tdata(0) , " 號"
    End If
End If

Loop Until Tdata(0) <= RoboNum Or Tdata(0) = 255 '機器人編號在輸入範圍內離開迴圈
Debug CSRXY(1,4), "請輸入欲執行之動作: "
Do
    Debugin Tdata(1)
    If Tdata(1) > MotionNum Or Tdata(1) = 0 Then '檢查動作編號是否在範圍內
        Debug CSRXY(1,4), "無此動作! 請重新輸入動作: "
    Else
        Tdata(2) = Tdata(2) + 1 '每送出一筆資料則資料編號加 1
        RFT.SendArray(Tdata) '送出陣列 Tdata
        Do
            Loop Until TxReady = 1 '在迴圈內等待資料傳送完成
            Debug Tdata(1)
        End If
    Loop Until Tdata(1) <= MotionNum And Tdata(1) > 0 '動作編號在輸入範圍內離開迴圈
    Do
        Debug CSRXY(1,7), "繼續使用請輸入 1 "
        Debugin x
        Loop Until x = 1
    Loop
End Sub

Sub RFTinital()
    RFT.EnTxReadyEvent() '啟動發送完成事件
    RFT.SetMode(0) '設定為發送模式
    RFT.SetCh(0) '設定頻道為 0
    RFT.SetRegCode(0) '設定註冊碼為 0
    RFT.SetRFID(0) '設定識別碼為 0
    RFT.Config() '將設定的模式、頻道、註冊碼、識別碼下載至模組
End Sub

```

Event RFT.TxReadyEvent() '設定發送完成事件

```
TxReady = 1
```

End Event

使用方法：

只要在終端視窗輸入機器人及動作編號即可，輸入機器人編號時給予 0 為控制所有機器人，輸入的編號超出設定範圍訊號不會發射出去並且會要求再輸入一次。下圖為執行結果。

