

# 修平科技大學

## 資訊網路技術系

### BIOLOID 機器人製作

指導老師：陳松雄老師

組長：BN97080 莊宜樺

組員：BN97016 郭家瑋

BN97072 陳傾貽

BN97103 吳喬凱

BN97116 張家銘

中華民國 101 年 1 月 4 日

# 修平科技大學

## 資訊網路技術系

### BIOLOID 機器人製作

指導老師：陳松雄老師

組長：BN97080 莊宜樺

組員：BN97016 郭家瑋

BN97072 陳傾貽

BN97103 吳喬凱

BN97116 張家銘

指導老師：\_\_\_\_\_

口試老師：\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

中華民國 1 0 1 年 1 月 4 日

## 目錄

摘要	1
研究動機及目的	2
第一章 BIOLOID 機器人	
1.1 機器人心臟 - CM5 可程式控制器	3
1.2 機器人關節組件	7
1.3 機器人馬達 AX-12	8
1.4 機器人判段物件 AX-S1	10
第二章 編輯軟體介紹	
2.1 邏輯行為控制器	11
2.2 動作編輯器	21
第三章 機器人製作	
3.1 攻擊鴨(手動操作)	34
3.2 避障判斷車	40
3.3 障礙判斷排除車	49

第四章 機器狗賽跑	
4.1 競賽解說	55
4.2 機器人設計規格	57
4.3 程式流程圖	59
4.4 機器人程式碼	60
第五章 心得	67
第六章 未來展望	75
參考文獻	76

## 圖目錄

(圖1.1.1)CM5可程式控制器	3
(圖1.2.1)關節組件	7
(圖1.2.2)線材與輪子	7
(圖1.3.1)AX-12+伺服馬達背面	8
(圖1.3.2)AX-12+伺服馬達正面	8
(圖1.3.3)馬達刻度示意圖	9
(圖1.4.1)聲音感測裝置	10
(圖1.4.2)距離感測裝置	10
(圖2.1.1)邏輯行為控制器頁面	11
(圖2.1.2)CM5控制功能表	13
(圖2.1.3)燈號控制範例	13
(圖2.1.4)AX-12+控制功能表	14
(圖2.1.5)AX-12+角度控制示意圖	15
(圖2.1.6)AX-12+旋轉控制示意圖	16
(圖2.1.7)馬達順時針旋轉設置	17
(圖2.1.8)馬達逆時針旋轉設置	17
(圖2.1.9)馬達角度限制設置	17
(圖2.1.10)AX-S1控制功能表	18

(圖2.1.11)障礙物判斷準則示意圖	19
(圖2.2.1)動作編輯器頁面	21
(圖2.2.2)動作編輯欄位	22
(圖2.2.3)頁面資訊視窗	23
(圖2.2.4)馬達位置資訊視窗	26
(圖2.2.5)關節調整示意圖	26
(圖2.2.6)取得關節資訊	27
(圖2.2.7)軟體實際顯示狀況	27
(圖2.2.8)資料上傳示意圖<一>	29
(圖2.2.9)資料上傳示意圖<二>	29
(圖2.2.10)存檔示意圖	30
(圖2.2.11)資料下載示意圖<一>	31
(圖2.2.12)資料下載示意圖<二>	31
(圖2.2.13)資料下載示意圖<三>	32
(圖2.2.14)資料下載示意圖<四>	32
(圖2.2.15)資料下載成功圖示	32
(圖2.2.16)動作編輯頁面銜接	33
(圖2.2.17)動作編輯頁面結束	33
(圖3.1.1)攻擊鴨流程圖	34

(圖 3.1.2)攻擊鴨實體照-----	34
(圖 3.1.3)攻擊鴨判斷流程-----	35
(圖 3.1.4)攻擊鴨攻擊姿勢實體照-----	35
(圖 3.1.5)攻擊鴨程式段<一>-----	36
(圖 3.1.6)攻擊鴨程式段<二>-----	37
(圖 3.1.7)攻擊鴨程式段<三>-----	38
(圖 3.1.8)攻擊鴨程式段<四>-----	39
(圖 3.2.1)避障判斷車流程圖-----	40
(圖 3.2.2)避障判斷車實體照-----	40
(圖 3.2.3)避障判斷車程式段<一>-----	41
(圖 3.2.4)避障判斷車程式段<二>-----	42
(圖 3.2.5)避障判斷車程式段<三>-----	43
(圖 3.2.6)避障判斷車程式段<四>-----	44
(圖 3.2.7)避障判斷車程式段<五>-----	46
(圖 3.2.8)避障判斷車程式段<六>-----	48
(圖 3.3.1)障礙判斷排除車流程圖-----	49
(圖 3.3.2)障礙判斷排除車實體照-----	49
(圖 3.3.3)障礙判斷排除車動作示意<一>-----	50
(圖 3.3.4)障礙判斷排除車動作示意<二>-----	50

(圖 3.3.5)障礙判斷排除車動作示意<三>	50
(圖 3.3.6)障礙判斷排除程式段<一>	51
(圖 3.3.7)障礙判斷排除程式段<二>	52
(圖 3.3.8)障礙判斷排除程式段<三>	53
(圖 3.3.9)障礙判斷排除程式段<四>	54
(圖 4.1.1)機器狗比賽場地圖	56
(圖 4.2.1)機器狗規格圖<一>	57
(圖 4.2.2)機器狗規格圖<二>	57
(圖 4.2.3)機器狗規格圖<三>	58
(圖 4.2.4)機器狗規格圖<四>	58
(圖 4.3.1)機器狗流程圖	59
(圖 4.3.2)機器狗程式段<一>	60
(圖 4.3.3)機器狗程式段<二>	60
(圖 4.3.4)機器狗程式段<三>	61
(圖 4.3.5)機器狗程式段<四>	61
(圖 4.3.6)機器狗程式段<五>	62
(圖 4.3.7)機器狗動作頁面<一>	63
(圖 4.3.8)機器狗動作頁面<二>	64
(圖 4.3.9)機器狗動作頁面<三>	64



(圖 4.3.10)機器狗動作頁面<四>	65
(圖 4.3.11)機器狗動作頁面<五>	65
(圖 4.3.12)機器狗動作頁面<六>	66

## 摘要

「Bioloid」，是一種全方位的機器人套件，能夠依據我們的想法，來組裝出各種型態。Bioloid機器融合「積木」的優點（可以組裝出各種實體型態，也能夠無限擴充。），並提供程式設計介面、控制器、智慧伺服馬達與感測模組；透過整合應用，幫助我們設計出「機器人的行為」。

Bioloid機器人的運作，藉由各種接收到得資訊（像是感測模組與伺服馬達的資訊），來協調整體的行為。例如：

1. 我們拍一下掌聲，機器狗聽到後站起來。當聽到兩下拍掌聲，機器狗便坐下。
2. 當客人接近時，機器人能夠打招呼。
3. 機器人能夠避開障礙物
4. 機器人玩球

## 研究動機

最初我們這一組的專題是要做樂高機器人，但是因為學校課程的關係，樂高機器人要給一年級的學弟上課所使用，所以數量不夠，老師便改變了專題題目，讓我們做 Bioloid 機器人。

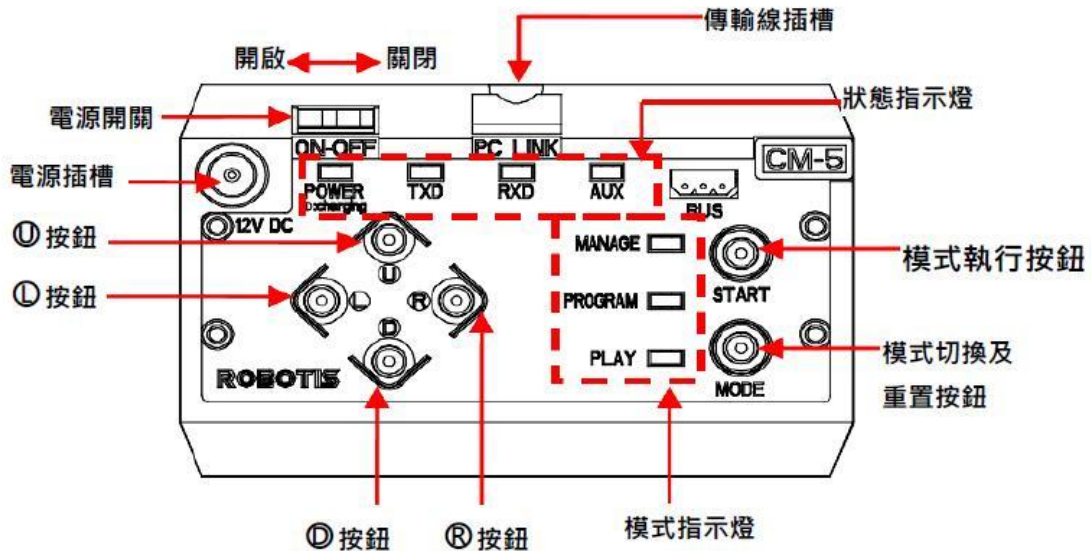
一開始我們還不知道是什麼，看到電視新聞上有機器人在跳舞，覺得很新奇，沒想到就是我們專題的機器人，這樣一來，更讓我們想要一探就近裡面的原理，為什麼它能跳舞？那還能做其他動作嗎？

## 研究目的

現在科技越發達，人們就越懶散，有網路的出現，要購買東西還是出去玩 . . . 等，只要動一動滑鼠就 OK 了，如此一來，人就越來越懶的動，身材、關節 . . . 等，都會退化。而機器人就等於代替人來行動，只要你有錢，也可以成為主人，而機器人就變成你專屬的傭人。機器人可以幫人類做非常多的事，所以就為了這個一直研發、一直在求進步。

# 第一章 BIOLOID 機器人

## 1.1 機器人心臟 - CM5 可程式控制器



(圖1.1.1)CM5可程式控制器

CM-5為Bioloid的核心，如同是Bioloid機器人的大腦。CM-5控制器最多可同時控制28個AX-12+伺服馬達和10個AX-S1感測模組；並內建按鈕可以當作輸入或是當作遙控功能的操作介面。(充電電池安裝於CM-5內部，也能自由拆換。)

### 1. 電源供應

CM-5 控制器內含一個鎳氫充電電池，提供Bioloid運作所需的電力。

當發現電力不足時，可透過外接電源線進行充電。充電步驟如下：

- (1). 安裝外接電源 (12V/5A) 到CM-5的「外接電源插座」，再打開電源開關。
- (2). 按下U鈕後，電源燈便會開始閃爍，代表充電進行中。

## 2. 通訊介面

電腦與CM-5控制器之間，透過RS232介面進行連線。如果電腦沒有RS232接頭，可以使用USB 轉RS232 轉換器。

## 3. 模式切換/重置按鈕 (Mode/Reset)

CM-5控制器具備三種模式來控制Bioid機器：管理模式、程式化模式、執行模式。當「模式指示燈」閃爍時，表示在待命模式（還沒決定要使用哪種模式）；按下模式切換鍵（MODE）鈕可以切換模式，按下模式執行鍵（START），CM-5控制器就會進入我們所選取的模式（燈號會持續發亮，而不會閃爍。）。

### <1>模式執行按鈕 (Start)

從「待命模式 (Standby)」進入到三大模式中的任一模式。除此之外，也能在「執行模式 (Play Mode)」中，給予額外的功能。

### <2>停止目前的運作

按下「模式切換鍵 (MODE)」，讓運作中的系統重置；或是關閉電源開關。

注意：模式切換鍵 (MODE) 本身也具備重置功能，所以是無法移作其他用途。

#### 4. 三大模式：

##### <1>管理模式 (Manage Mode)

用來驗證CM-5、AX-12+及AX-S1的狀態與測試機器人的運動。必須搭配機器人終端機來使用。使用此模式前，建議先完全了解CM-5、AX-12+及AX-S1，會比較容易上手。

##### <2>程式化模式 (Program Mode)

使用動作編輯器的過程中，CM-5會自動進入程式化模式。如想直接透過機器人終端機來編輯動作資料，請在機器人終端機的畫面下，將CM-5切入程式化模式即可。

##### <3>執行模式 (Play Mode)

執行行為控制程式所設計的程式。模式執行按鈕 (Start)

從「待命模式 (Standby)」進入到三大模式中的任一模式。除此之外，也能在「執行模式 (Play Mode)」中，賦予不同的功能。

## 其他部分

### <1>停止目前的運作

按下「模式切換鍵 (MODE)」，讓運作中的系統重置；或是關閉電源開關。

### <2>狀態指示燈 (Status Indicating LED)

#### i 電源 (POWER)

當電源啟動時，LED燈就會亮。當CM-5進入充電模式後，此燈會一直閃爍來表示充電狀態。

#### ii 資料傳送 (TXD)

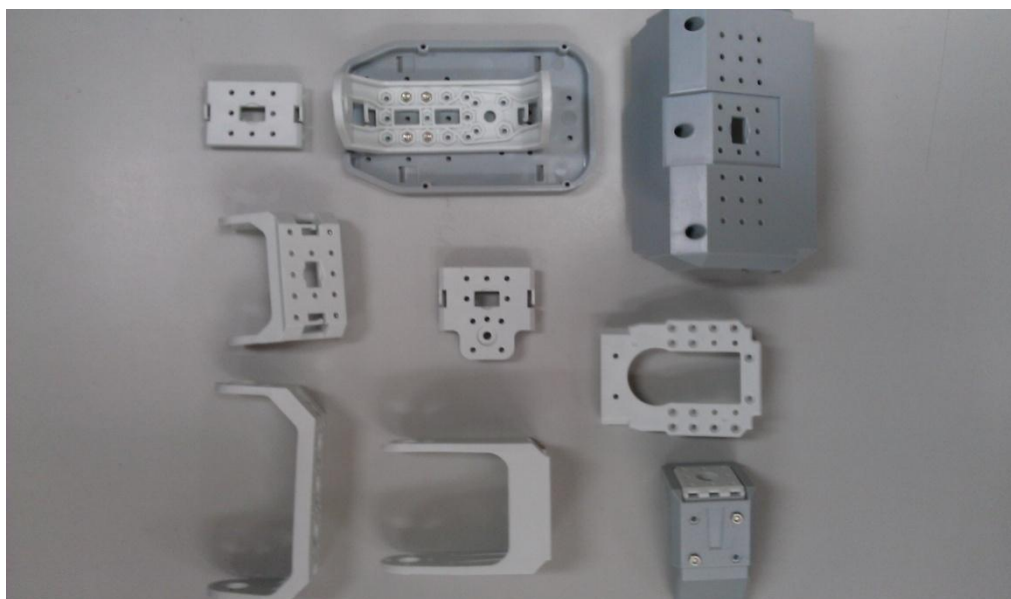
當CM-5傳送資料給外部裝置時，LED會亮。資料接收 (RXD) 當CM-5接收外部裝置資料時，LED 會亮。

#### iii 輔助 (AUX)

可透過「行為控制程式 (Behavior Control Programmer)」來決定功能。

<3>按鈕 (上、下、左、右按鈕) 可透過「行為控制程式 (Behavior Control Programmer)」來決定功能按鈕的功能。

## 1.2 機器人關節組件



(圖1.2.1)關節組件



(圖1.2.2)線材與輪子

框架、纜線、螺絲、螺帽、承軸和輪圈等等，能將CM-5及AX-12+、AX-S1銜接起來。



### 1.3 機器人馬達 AX-12+



(圖 1. 3. 1)AX-12+伺服馬達背面



(圖 1. 3. 2) AX-12+伺服馬達正面

AX-12+智慧型馬達，做為機器人的運動關節。內建了微控制器，可以控制速度、旋轉位置、感測溫度、讀取設定值...等。除此之外，當馬達旋轉位置超出我們預設的安全範圍（溫度過高、過載或是其他錯誤狀況），會自動關閉以避免損壞，並發出聲音警告使用者。

### <1>傳輸對象代號（ID）

每個AX-12+與AX-S1都有一個獨特的傳輸對象代號。當多個AX-12+與AX-S1連接在CM-5控制器時，可以藉由傳輸對象代號來區分彼此。因此，每個元件的傳輸對象代號不能重複，否則無法運作。

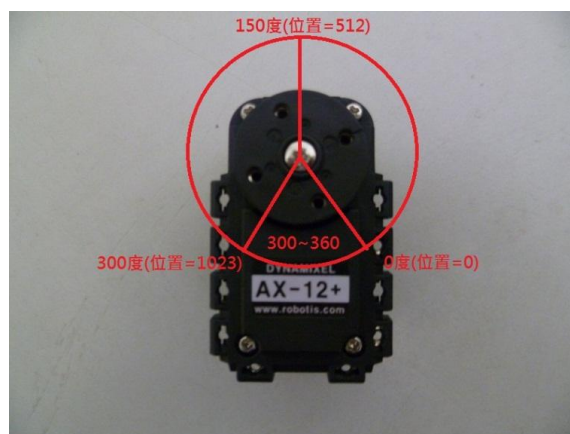
### <2>修改傳輸對象代號（ID）

AX-12+的傳輸對象代號，並非固定的。我們可以透過CM-5的管理模式或是行為控制來更改。

### <3>角度表示

AX-12+可控制的角度為 $0^{\circ} \sim 300^{\circ}$ 。

在系統中，以0~1023刻度來表示旋轉的位置。



(圖1.3.3)馬達刻度示意圖

## 1.4 機器人判段物件AX-S1

AX-S1外觀與AX-12+相當相似，但內容大不相同。AX-S1具備許多的感測器，如同機器人的五官。擁有「距離偵測」、「亮度偵測」、「溫度偵測」及「聲音偵測」的功能，並內建紅外線資料傳送的能力。不但如此，內建蜂鳴器還可以播放音效。至於外殼結構及通訊原理都和AX-12+幾乎相同。差別於使用的功能物件不相同、輸出的資料型態不同。(AX-S1的ID預設值是100，代號設定範圍100~109。)



(圖1.4.1)聲音感測裝置

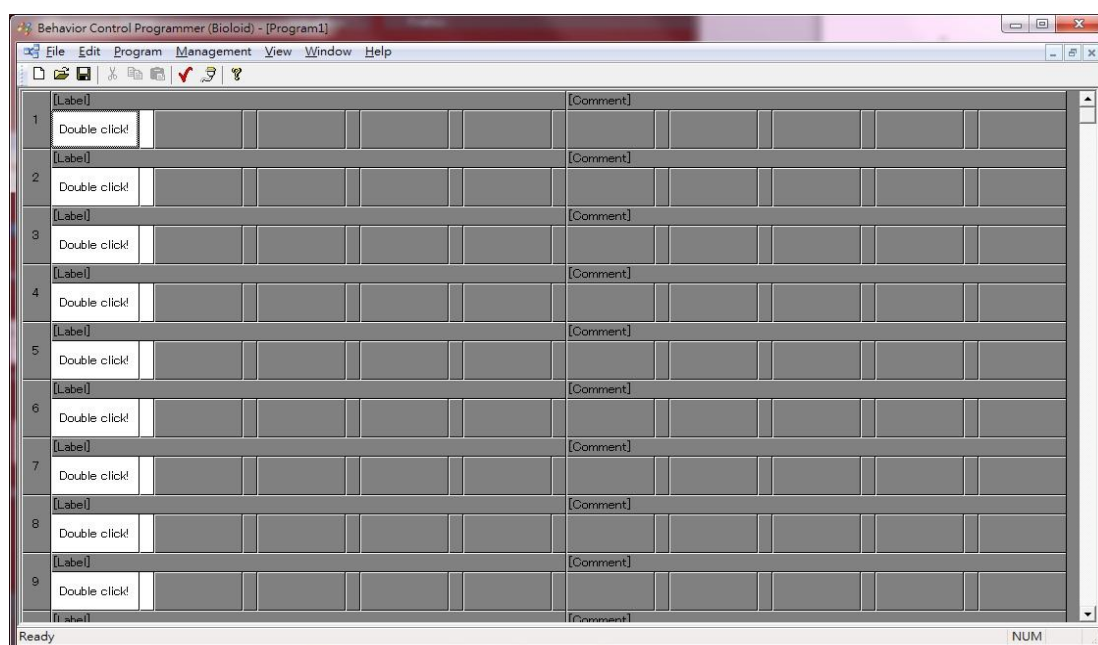


(圖1.4.2)距離感測裝置

## 第二章 編輯軟體介紹

BIOLOID軟體本身可分成三大部份，（行為控制（Behavior control）、動作編輯器（Motion Editor）、機器人終端機（Robot Terminal）），操作簡易，適合所有的使用者及機器人入門者。

### 2.1 邏輯行為控制器（Behavior Control Program）



（圖2.1.1）邏輯行為控制器頁面

機器人是一種能夠做各式各樣動作的裝置。然而，要讓機器人能夠依據環境的情形，做出配合的行為；機器人就需要有一個程式來指導機器人的動作。而「行為控制邏輯（Behavior Control Program）」就是在決定機器人行為的應用程式。

## 1. 輸入和輸出

銜接輸入與輸出之間的關係是行為控制邏輯（Behavior Control Program）的基本架構。「輸入」好比人類的五官，例如：耳朵、眼睛、觸覺等（AX-S1感測器）；而大腦（CM-5）接收到外在的資料，根據情況做出的合適反應；對機器人來說，這就是「輸出」。例如：要設計一支機器狗，拍一下聲音，它就起立，拍二下聲音便會坐下。由這個例子看來，我們的拍掌聲是一種「輸入」，起立和坐下是一種「輸出」。藉由行為控制邏輯，達到我們想想賦予機器人的功能。所以，我們需要認識輸入及輸出的功能物件來建立行為邏輯。其實說穿了，行為控制邏輯就是以邏輯的方式，來控制輸入及輸出的功能物件。







## 2. CM-5的控制功能

行為控制程式具備以下圖示，以物件來代表CM-5 的各式各樣的功能。

Icon (圖示)	Name (名稱)	Function (功能)
	動作播放頁面	播放 Motion Editor 中，所指定的動作頁面。
	動作播放的狀態	1 表示動作頁面正在播放中，0 表示結束或是沒有動作。
	傳遞無線資料	無線資料的傳送
	接收無線資料	接收無線的資料
	無線通訊旗標	1 表示接收到無線資料，0 表示沒有接收到無線資料。
	輔助燈	若輸入為 1，輔助燈亮起。若輸入為 0，輔助燈熄滅。
	CM-5 按鈕	CM-5 上有五個按鈕，用來作為輸入裝置。
	計時器	輸入的數值，每隔 0.125 秒會減 1。
	設定無線晶片的 傳輸對象位址	設定無線晶片的傳輸對象位址 (表示通訊的對象)
	顯示無線晶片自己 的通訊位址	顯示無線晶片自己的通訊位址 (表示自己的通訊代號)
	顯示	將輸入的數值，顯示在螢幕上。(重複顯示在同一個位置)
	換行顯示	將輸入的數值，顯示在螢幕上。(新的資料會開新的一行顯示)

(圖 2.1.2) CM5 控制功能表

EX: 按下按鈕U，AUX燈號亮起。按下按鈕D，AUX燈號熄滅。

1	[Label]	START								[Comment]	
2	[Label]	F	(	 Button	=	 U	)	THEN	LOAD	 AUX LE... <-	1
3	[Label]	F	(	 Button	=	 D	)	THEN	LOAD	 AUX LE... <-	0
4	[Label]	END								[Comment]	

(圖 2.1.3) 燈號控制範例

### 3. AX-12 的控制功能

Address (位址)	Icon (圖示)	Name (名稱)	Function (功能)
24		電源開關	1 代表開啟電源，0 代表關閉電源。
25		警示燈號	1 代表開啟燈號亮起，0 代表開啟燈號熄滅。
26		順時針位置誤差的極限	設定容許的位置誤差量 (0~255)。在這誤差容忍量中，馬達不會輸出扭力來校正位置。
27		逆時針位置誤差的極限	設定容許的位置誤差量 (0~255)。在這誤差容忍量中，馬達不會輸出扭力來校正位置。
28		順時針旋轉斜率	當馬達位置誤差超出容忍範圍，決定伺服馬達馬達要輸出多大的扭力 (0~255)，校正移動位置。
29		逆時針旋轉斜率	當馬達位置誤差超出容忍範圍，決定伺服馬達馬達要輸出多大的扭力 (0~255)，校正移動位置。
30		目標位置	馬達旋轉的角度範圍在 0-300 度 (數值範圍：0-1023)
32		移動速度	馬達移動速度 (數值範圍：0-1023)
34		最大扭力	決定伺服馬達最大輸出的扭力 (數值範圍：0-1023)
36		目前位置	馬達目前旋轉角度 (數值範圍：0-1023)
38		目前速度	馬達目前旋轉速度 (數值範圍：0-1023)
40		目前負載大小	感測馬達目前的承載 (數值範圍：0-1023)
42		輸入電壓	輸入電壓值 X10 (例如：12V，則是以 120 來表示。)
43		內部溫度	馬達內部溫度 (°C)
46		移動旗標	1 代表馬達正在移動中，0 代表停止狀態。

(圖 2.1.4)AX-12+控制功能表

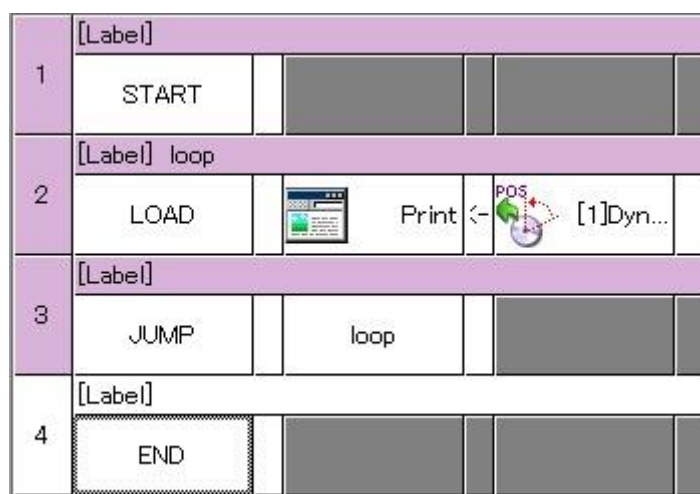
#### 讀取目前位置

AX-12+伺服馬達以0~1023的數值，來代表馬達旋轉角度(0~300 度)。

以下列範例可以幫助了解AX-12+位置控制功能。

EX：用手移動AX-12 馬達的轉盤，觀看移動中的位置變化。

- (1) 將CM-5控制器銜接AX-12+
- (2) 編寫行為控制程式
- (3) 將程式載入到CM-5控制器
- (4) 在程式下載畫面，點選「PLAY」執行程式。當我們轉動AX-12+ 伺服馬達的轉盤，便可以看到馬達目前位置的變化。



(圖2.1.5)AX-12+角度控制示意圖

<2>極限 (Margin，在此決定誤差的容忍程度。在這容忍程度中，馬達是不會輸出扭力來校正位置。)

當設定「順時針位置誤差的極限」與「逆時針位置誤差的極限」，則代表「目標位置」的容忍誤差。在「容忍誤差範圍」內，馬達是不會



輸出扭力來校正位置（除非超出容忍誤差的範圍）。合適的馬達力量控制，會比速度控制更來得重要。

#### <4>全轉模式

在全轉模式下 (Endless Turn)，AX-12+ 伺服馬達可以當作輪子使用。

#### 全轉設定

The diagram illustrates the configuration for full rotation control on an AX-12 servo motor. It consists of three parts:

- Parameter Table (Top):** A table with columns for Label, START, LOAD, and END. The 'LOAD' row is highlighted in purple and contains 'CM-5 Typing', 'Dynamixel', and 'Undefine'. A circled '1' points to the 'Dynamixel' cell, and a circled '2' points to the 'Undefine' cell.
- Set Parameter Dialog (Middle):** A window titled 'Set Parameter' with fields for 'ID' (set to 3) and 'Address' (set to 8). A list of motor types is shown, with 'Custom ID' selected and circled. A circled '4' points to the 'Address' field.
- Parameter Table (Bottom):** A table with columns for Label, START, LOAD, and END. The 'LOAD' row is highlighted in purple and contains '1:8', '0', and '0'. A circled '3' points to the '1:8' cell.

「1:8」表示「傳輸對象代號為1」的AX-12 伺服馬達，內部的「全轉控制功能」在位址「8」。我們將「逆時針角度限制」設定為零，馬達便可以360 度旋轉。

(圖2.1.6)AX-12+旋轉控制示意圖

#### <5.>旋轉控制（控制正轉與反轉。）

[逆時針旋轉] : 0 (速度最慢) ~ 1023 (速度最快)

2	[Label]	LOAD	[1]Dyn...	←	600	
---	---------	------	-----------	---	-----	--

(圖2.1.7)馬達順時針旋轉設置

逆時針以速度600 旋轉。

[順時針旋轉] : 1024 (速度最慢) ~ 2047 (速度最快)

[Label]					
2	LOAD		[1]Dyn...	<-	1624

(圖2.1.8)馬達逆時針旋轉設置

順時針以速度600 旋轉，則輸入數值為1624 (1024+600)。

<6>轉換為位置控制

「1:8」表示「傳輸對象代號為1」的AX-12 伺服馬達，內部的「全轉控制功能」在位址「8」。將「逆時針角度限制」設定為1023，馬達便可以回到原本的「位置控制模式」。

[Label]					
1	START				
[Label]					
2	LOAD	1:8	<-	1023	
[Label]					
3	END				

(圖2.1.9)馬達角度限制設置

4. AX-S1 的控制功能

Address (位址)	Icon (圖示)	Name (名稱)	Function (功能)
26		左側紅外線距離感測器	偵測左側障礙物距離 (數值越小, 距離越遠。)
27		中央紅外線距離感測器	偵測中央障礙物距離 (數值越小, 距離越遠。)
28		右側紅外線距離感測器	偵測右側障礙物距離 (數值越小, 距離越遠。)
29		左側紅外線亮度感測器	偵測左側光線的亮度 (數值越大, 亮度越高。)
30		中央紅外線亮度感測器	偵測中央光線的亮度 (數值越大, 亮度越高。)
31		右側紅外線亮度感測器	偵測右側光線的亮度 (數值越大, 亮度越高。)
32		障礙物感測	當距離感測器數值, 大於標準值時, 會產生通知訊息。 1 代表左側、2 代表中央, 4 代表右側。如果同時多個方向, 則是這些數值相加。例如: 3 = 左側與右側。
33		亮度感測	當亮度感測器數值, 大於標準值時, 會產生通知訊息。 1 代表左側、2 代表中央, 4 代表右側。如果同時多個方向, 則是這些數值相加。例如: 3 = 左側與右側。
35		聲音資料	感測器偵測到的音量大小
36		聲音資料的最大值	代表目前輸入聲音中的, 最大的音量。
37		聲音計數器	偵測拍掌聲音的次數
38		聲音時間偵測	用來偵測聲音的速度
40		蜂鳴器	0~52 音階 (從「La」開始, 上昇半個音階。)
41		蜂鳴器播放時間	每個音階播放的時間 (0~255); 每個數值代表 0.1 秒。
46		紅外線資料接收訊息	1 代表接收到紅外線資料, 0 代表沒有接收到紅外線資料。
48		紅外線資料接收	接收紅外線控制資料
50		發送紅外線資料	發送紅外線控制資料
52		障礙物偵測基準	設定距離多近, 才算是障礙物。同時也是決定偵測範圍。
53		亮度偵測基準	設定光線多亮, 才算是偵測到光線。

(圖 2.1.10)AX-S1 控制功能表

### i. 距離感測

#### AX-S1的控制功能

AX-S1有三顆紅外線感距離測器，當物體越接近感測器，數值會越高。

### ii. 判斷障礙物

機器人操作過程中，感測障礙物會遠比偵測距離還來得重要。「障礙物偵測基準」與「障礙物偵測」物件，就是用來偵測障礙的功能。機器人如何判斷障礙物是決定於「障礙物偵測基準」，距離多遠的物體，才能算是障礙物。而「障礙物偵測」，則是依據「障礙物偵測基準」，幫我們判斷是「否有障礙物的出現」與「障礙物出現的方向」。



(圖2.1.11)障礙物判斷準則示意圖

### iii. 短距離和長距離偵測

AX-S1感測模組有兩種距離感測模式短距離與長距離感測模式。我們可以在「短距離感測模式」取得較為精準的距離數值，或是採用「長距離感測模式」來拓展偵測距離。

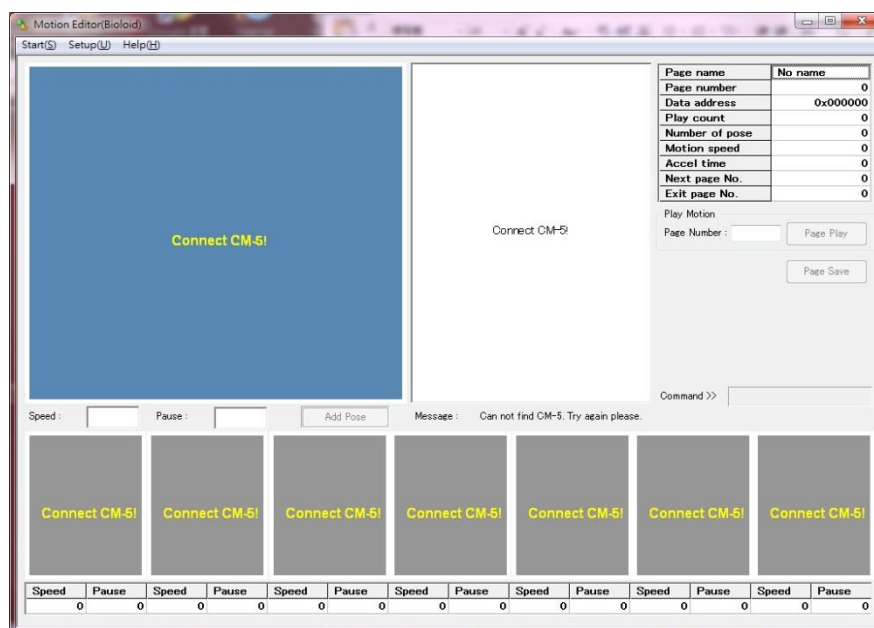
#### iv. 亮度感測

AX-S1具有三組亮度感測器，用來偵測周圍環境的亮度。數值越高，代表亮度越強。

#### v. 聲音感測器

AX-S1具有聲音感測器，用來偵測周圍環境的聲音；具備一些聲音相關的功能。當沒有聲音或聲音很小時，數值約為128，偵測到聲音時，聲音最大數值為255。感測資料約每秒3800次。在偵測到的聲音中，取最大值。這功能是用來測量一段時間內，我們拍了幾下聲音。當偵測到一次拍掌聲，計數器會暫停0.08 秒，來避免將單一拍掌聲誤視為連續拍掌聲。

## 2.2 動作編輯器 (Motion Editor)

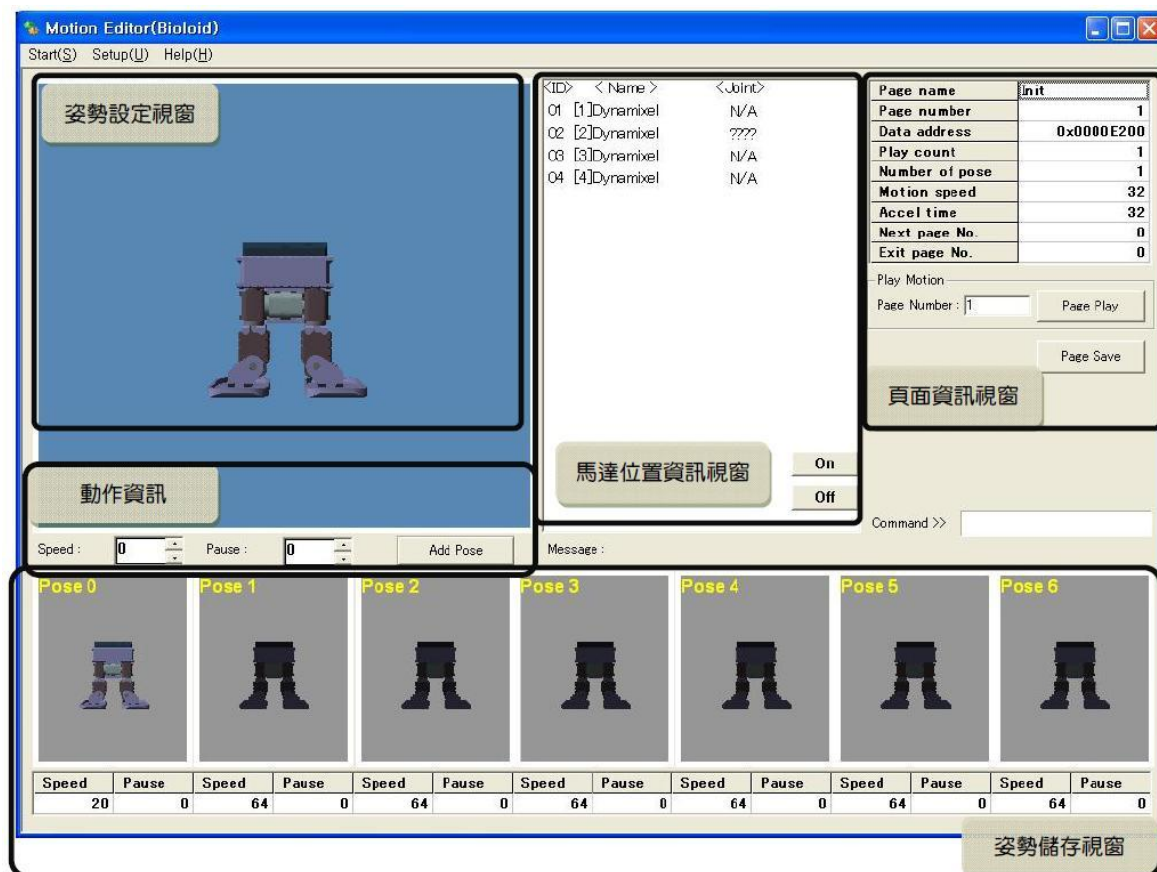


(圖 2.2.1) 動作編輯器頁面

動作編輯器像個動畫製造機，但是他的輸出不是動畫，而是活生生的機器人，設計完成的動作，我們再藉由行為控制，設計相關的邏輯。換句話說，「走路、吃飯、睡覺…」是一種動作，可藉由動作編輯器來設計。而決定「何時走路、何時吃飯、何時睡覺」是一種邏輯，便是使用行為控制來達成。行為控制能夠實現簡單的動作，但動作編輯器更能實現高度複雜的運動行為。將設計好的動作儲存在CM-5控制器，再透過行為控制來執行這些複雜動作。動作編輯器有兩種操作方式：第一種，使用「動作編輯器」軟體，方便且簡單。第二種，透過「機器人終端機」，讓CM-5進入「程式化模式」，來管理動作資料。前者，是圖形操作介面，較方便使用。而後者是純文字介面。

## 1. 開啟動作編輯器

動作編輯器的內容如下圖所示：



(圖 2. 2. 2) 動作編輯欄位

## 2. 動作編輯

動作頁面視窗資訊，也是動作頁面執行的內容。在每個項目用滑鼠左鍵點兩下，便可以修改頁面的內容。

Page name	Init
Page number	1
Data address	0x0000E200
Play count	1
Number of pose	1
Motion speed	32
Accel time	32
Next page No.	0
Exit page No.	0

Play Motion

Page Number :

(圖2.2.3) 頁面資訊視窗

i. Page name (動作頁面名稱)

動作頁面的名稱 (一種註解)，方便往後的使用。

ii. Page number (頁面編號)

動作頁面的編號 (1~127)。(每頁可以容納7種動作)當輸入不同的編號，就會讀取CM-5 控制器內部不同的動作頁面。

iii. Data address (資料位址)

顯示目前的動作頁面，所在的實際記憶體位址。

iv. Play count (播放次數)

動作頁面重複執行的次數 (預設是1，表示執行一次)。

v. Number of pose (動作數量)

顯示目前的動作頁面裡，有多少動作 (最多7個，也就是0~6)。動作播放是採循序的方式，從0 開始播放。



vi. Motion speed (動作頁面的速度)

表示整個動作頁面執行的速度 (預設是32)。

vii. Accel time (加速度時間)

當動作執行時，所有的馬達都是在「加速度→保持速度→減速度」的條件下重複運作。「accel time (加速度時間)」是代表馬達的「加速度與減速度時間」。當數值越小 (代表加速度時間越短)，瞬間加速度可能會造成機械損毀的機率上升。相反的，數值越高，反而有些部分會無法動作。

viii. Next Page No (銜接頁面)

一頁可以執行七個姿勢，若沒有設定銜接頁面 (Next Page No)，它會設為0 (不銜接動作)，只會執行目前頁面。如果要執行超過七個姿勢，就要設定銜接頁面 (Next Page No)。

ix. Exit Page No (中斷動作頁面)

在行為控制程式中，我們播放動作頁面。在動作播放的過程中，我們下達停止播放動作頁面 (「Motion play page (動作播放)」輸入0)，此時，執行「Exit page No (中斷頁面)」所指定的動作頁面。預設為零，代表停止目前的動作。如果不為零，則代表停止目前的動作，轉而執行指定的動作頁面。

x. Play Motion (播放動作)

當按下頁面播放 (Page Play)，機器人便會連續執行「姿勢儲存視窗」所設計的動作。頁面播放 (Page Play) 執行之後，原有的按鈕會變成停止按鈕 (Stop button)。當按下停止鈕 (Stop button)，便可以中斷機器人目前的動作。如果要指定播放的動作頁面，在Play Motion (播放動作頁面) 項目裡，輸入指定的動作頁面於Page number (動作頁面代號)，再按頁面播放 (Page Play) 即可。

xi. Page save (存放動作頁面)

編輯中的動作資料，是暫存存放在CM-5控制器的隨機記憶體 (RAM)。當我們切換頁面時，這些資料便會刪除。因此，我們必須按下Page save (存放動作頁面)，將設計好的動作資料存放在CM-5裡。

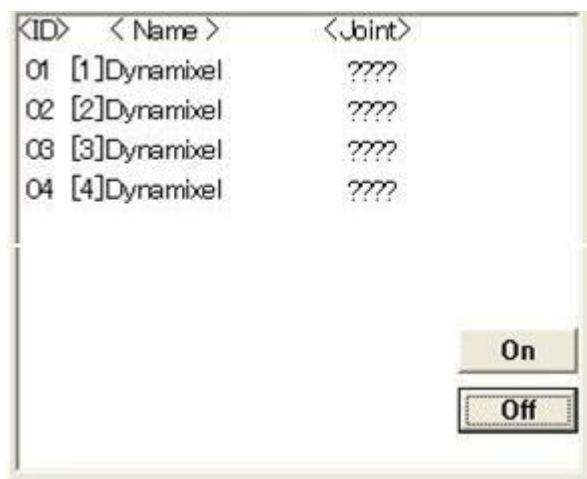
### 3. 製作動作

一個動作 (Motion)，是由許多的個別姿勢 (Pose) 所組成。例如：

「Walking droid」 機器人走路，就藉由許多的姿勢來組成；就好比動畫一般。

#### i. 設計姿勢

##### (1) 點選「OFF (關閉馬達扭力)」



(圖2.2.4)馬達位置資訊視窗

##### (2) 用手調整機器人的關節



(圖2.2.5)關節調整示意圖

(3) 點選「ON (開馬達扭力，並取得關節資訊。)」

<ID>	< Name >	< Joint >
01	[1]Dynamixel	0401
02	[2]Dynamixel	0396
03	[3]Dynamixel	0441
04	[4]Dynamixel	0327

On

Off

(圖2.2.6)取得關節資訊

(4) 「姿勢設定視窗」顯示實際狀況



(圖2.2.7)軟體實際顯示狀況

#### ii. 姿勢速度與暫停時間

可以各別設定每個姿勢的移動速度，也能夠決定他們的暫停時間。暫停時間單位為7.8ms；如果設定「Pause time」為50，則代表暫停390ms（ $50 \times 7.8\text{ms}$ ）。

#### iii. 儲存姿勢 (Save pose)

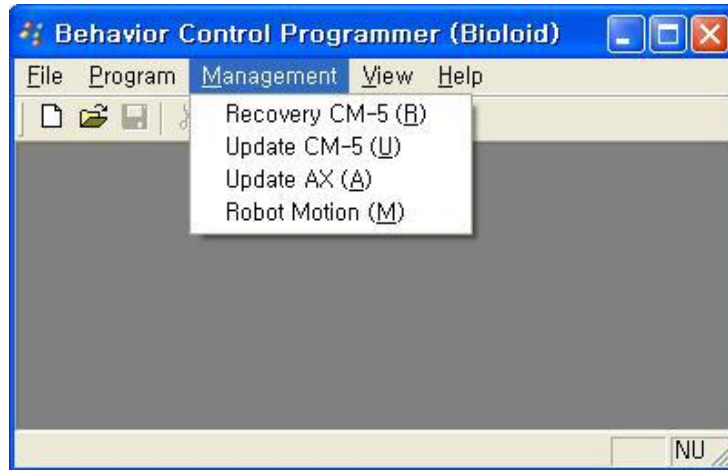
如果用上述的方式，來產生新的姿勢或是修改原有的姿勢；可以使用滑鼠拖曳的方式，來更新原有的姿勢。

#### iv. 上傳與下載動作資料

動作資料是藉由動作編輯器 (Motion Editor)，將編輯好的動作資料存放在機器人的CM-5 控制器。可以使用行為控制程式 (Behavior Control Programmer)，來管理上載 (CM-5→PC) 或下載 (PC→CM-5) 動作資料。這樣的好處，在於可以將設計好的動作，以檔案的型態存放在電腦。一方面是備份，另一方面是可以分享於不同的機器人。

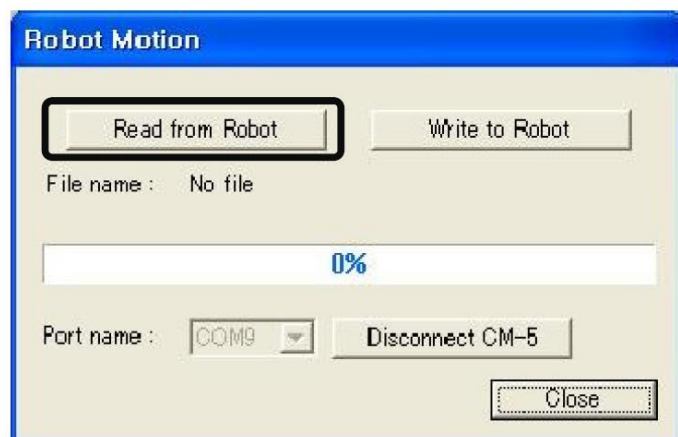
上傳動作資料

(1) 選擇「Robot Motion ( 機器人動作資料) 」



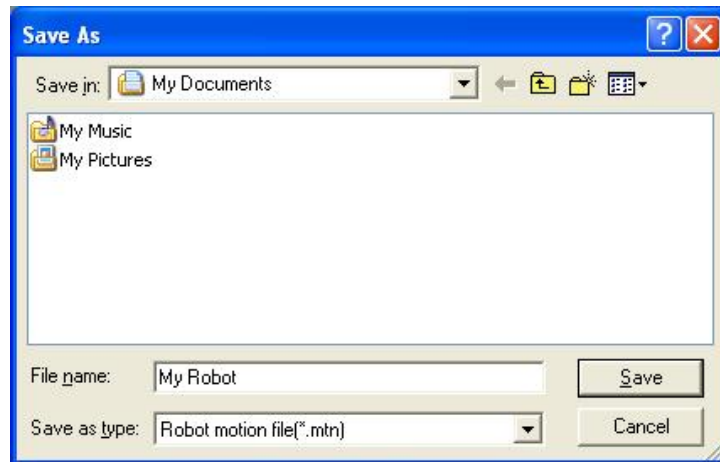
(圖2. 2. 8)資料上傳示意圖<一>

(2) 點選Read from robot ( 讀取機器人的動作資料)



(圖2. 2. 9)資料上傳示意圖<二>

(3) 選擇動作資料的存放位置與名稱



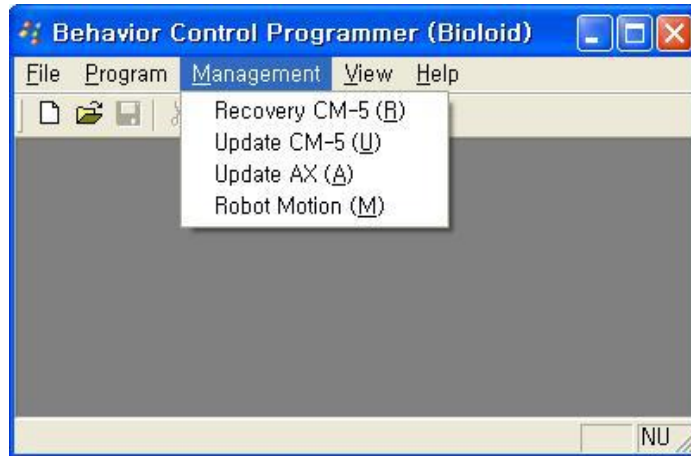
(圖 2.2.10)存檔示意圖

(4) 點選Read from robot (讀取機器人的動作資料)

(5) 完成後，請按下CM-5 的Mode (模式切換)，來重新啟動CM-5控制器。

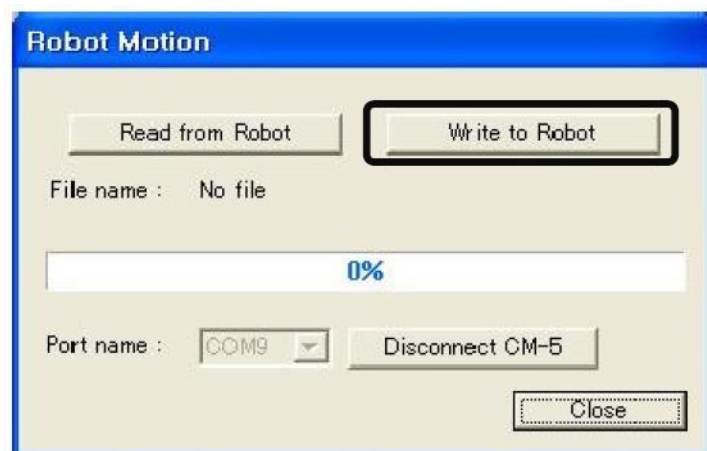
下載動作資料

(1) 選擇Robot Motion ( 機器人動作資料)



(圖2.2.11) 資料下載示意圖<一>

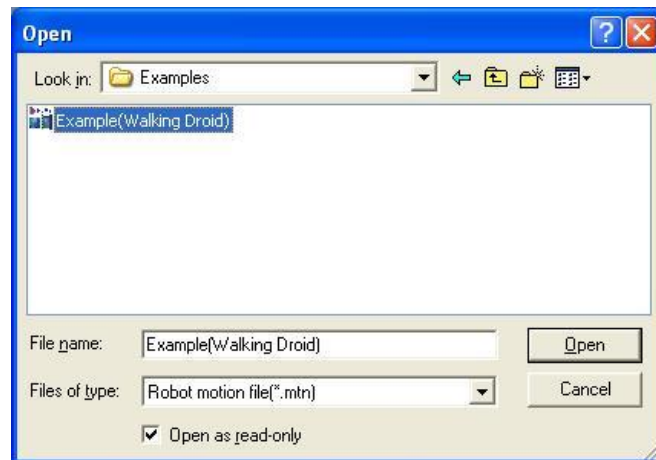
(2) 點選Write to robot ( 寫入機器人的動作資料)



(圖 2.2.12) 資料下載示意圖<二>

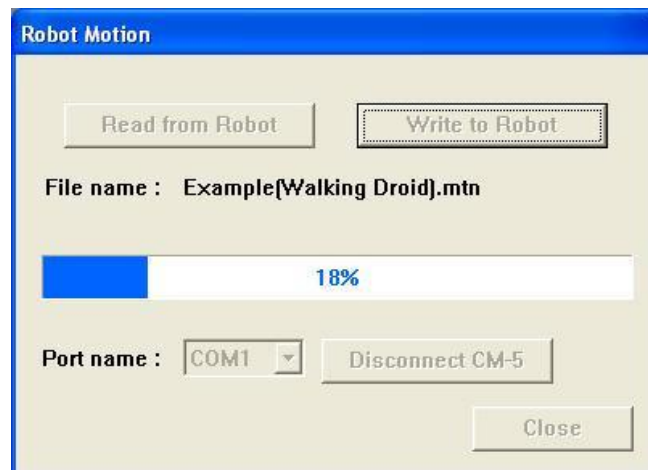


(3) 選擇欲載入的動作資料檔



(圖2.2.13) 資料下載示意圖<三>

(4) 點選Write to robot (寫入機器人的動作資料)



(圖2.2.14) 資料下載示意圖<四>

(5) 寫入成功



(圖2.2.15)資料下載成功圖示

### v. 銜接動作頁面與中斷動作頁面

為了讓原有規則的動作變得更有效率，我們可以充分的運用「銜接頁面 (Next Page No)」與「中斷頁面 (Exit pageNo)」。

Page name	Walk
Page number	5
Data address	0x00010A00
Play count	1
Number of pose	6
Motion speed	55
Accel time	32
Next page No.	5
Exit page No.	0

(圖2.2.16)動作編輯頁面銜接

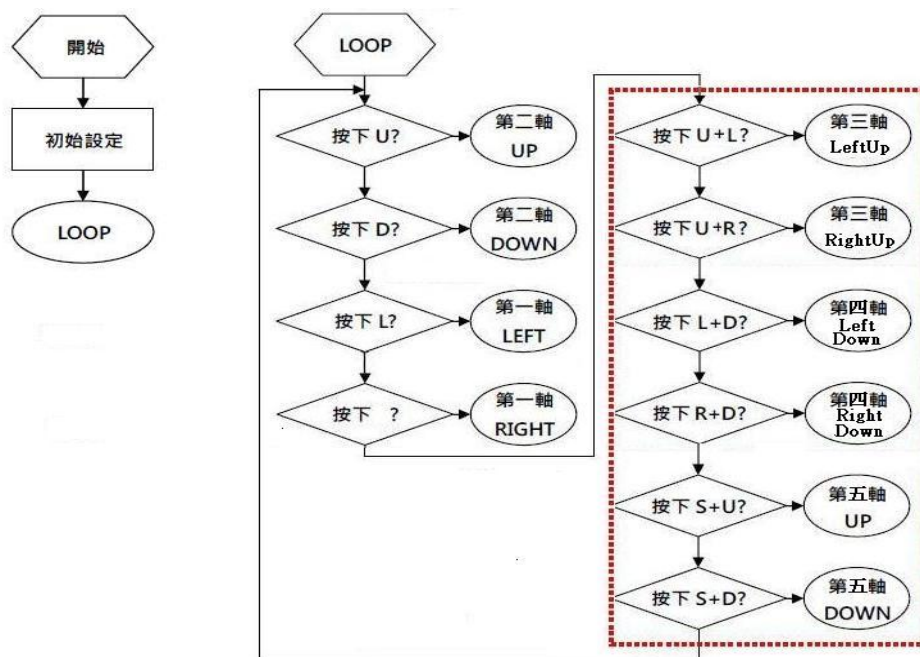
Page name	Walk
Page number	5
Data address	0x00010A00
Play count	1
Number of pose	6
Motion speed	55
Accel time	32
Next page No.	5
Exit page No.	44

(圖2.2.17)動作編輯頁面結束

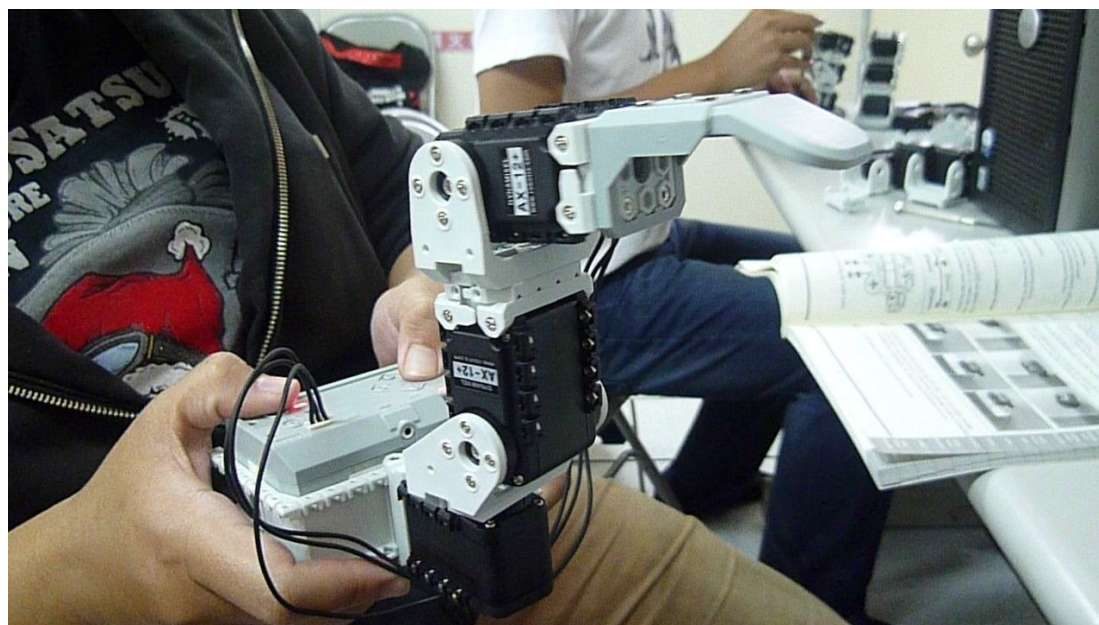
# 第三章 機器人製作

## 3.1 攻擊鴨

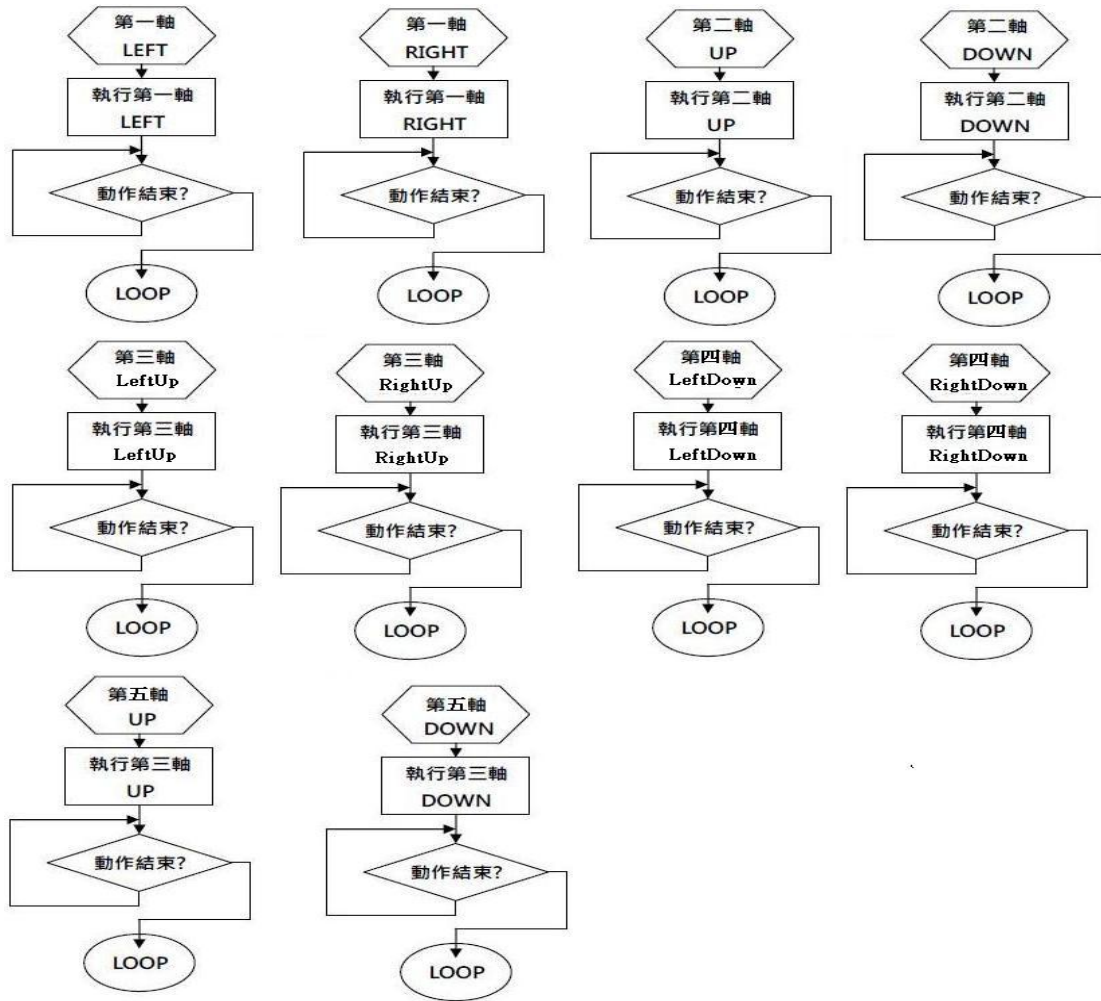
攻擊鴨-流程圖



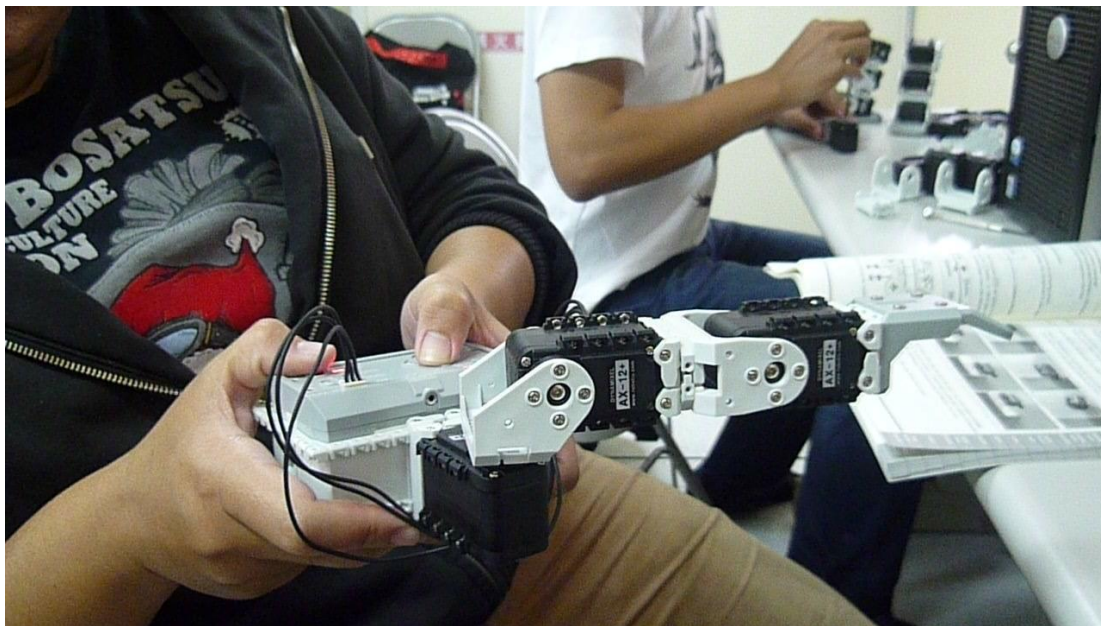
(圖 3.1.1) 攻擊鴨流程圖



(圖 3.1.2) 攻擊鴨實體照



(圖 3.1.3)攻擊鴨判斷流程



(圖 3.1.4)攻擊鴨攻擊姿勢實體照

## 攻擊鴨-程式碼

[Line]	[Label]									[Comment]
1	START									
2	LOAD	SPEED	All Dy...	{-	100					
3	CALL	Initia...	呼叫第16行							
4	IF	(	Button	=	U	)	THEN	CALL	Down a...	
5	ELSE IF	(	Button	=	D	)	THEN	CALL	Lift a...	
6	ELSE IF	(	Button	=	R	)	THEN	CALL	Turn r...	
7	ELSE IF	(	Button	=	L	)	THEN	CALL	Turn l...	
8	ELSE IF	(	Button	=	U+L	)	THEN	CALL	LeftUp	
9	ELSE IF	(	Button	=	U+R	)	THEN	CALL	RightU...	
10	ELSE IF	(	Button	=	L+D	)	THEN	CALL	LeftDo...	
11	ELSE IF	(	Button	=	R+D	)	THEN	CALL	RightD...	
12	ELSE IF	(	Button	=	U+S	)	THEN	CALL	Lift h...	
13	ELSE IF	(	Button	=	D+S	)	THEN	CALL	Down h...	
14	JUMP	Loop								

(圖 3.1.5)攻擊鴨程式段<一>



















程式碼第 2~3 行

設定機器一開啟的起步動作將所有的馬達設定轉速為 100

程式碼第 4~14 行是接收控制器(CM-5)給予的指令

其中 8~13 為需要同時按兩個按鍵的組合鍵

再由第 14 行迴圈來持續輸入指令運行

15	[Label]	Double click!												[Comment]				
16	[Label] Initial Action	CALL	Attack...	呼叫第60行										[Comment]				
17	[Label]	RETURN												[Comment]				
18	[Label]	CALL	Initia...	呼叫第54行										[Comment]				
19	[Label]	Double click!												[Comment]				
20	[Label] Lift arm	IF	(	 [2]Dyn...	)	>		312	)	THEN	COMPUTE	 [2]Dyn...	=	 [2]Dyn...	-		10	[Comment]
21	[Label]	RETURN												[Comment]				
22	[Label]	Double click!												[Comment]				
23	[Label] Down arm	IF	(	 [2]Dyn...	)	<		812	)	THEN	COMPUTE	 [2]Dyn...	=	 [2]Dyn...	+		10	[Comment]
24	[Label]	RETURN												[Comment]				
25	[Label]	Double click!												[Comment]				
26	[Label] Turn left	IF	(	 [1]Dyn...	)	<		812	)	THEN	COMPUTE	 [1]Dyn...	=	 [1]Dyn...	+		10	[Comment]
27	[Label]	RETURN												[Comment]				
28	[Label]	Double click!												[Comment]				
29	[Label] Turn right	IF	(	 [1]Dyn...	)	>		212	)	THEN	COMPUTE	 [1]Dyn...	=	 [1]Dyn...	-		10	[Comment]
30	[Label]	RETURN												[Comment]				
31	[Label]	Double click!												[Comment]				
32	[Label] LeftUp	IF	(	 [1]Dyn...	)	<		812	)	THEN	COMPUTE	 [1]Dyn...	=	 [1]Dyn...	+		10	[Comment]
33	[Label]	IF	(	 [2]Dyn...	)	>		312	)	THEN	COMPUTE	 [2]Dyn...	=	 [2]Dyn...	-		10	[Comment]
34	[Label]	RETURN												[Comment]				
35	[Label]	Double click!												[Comment]				

(圖 3.1.6)攻擊鴨程式段<二>

Initial Action 起步動作(16~8)































為第 3 行回歸初始位址後執行開啟機器時動作(手向前伸)

Turn left 左轉(26~27)、Turn right 右轉(29~30)

接收單一按鍵指令讓馬達-1 以正負 10 度左右移動

Lift arm 提升手臂(20~21)、Down arm 手臂向下(23~24).

## 接收單一按鍵指令讓馬達-2 以正負 10 度上下移動

32	[Label] LeftUp	IF	(		[1]Dyn...	<	812	)	THEN	COMPUTE		[1]Dyn...	=		[1]Dyn...	+	10	[Comment]
33	[Label]	IF	(		[2]Dyn...	>	312	)	THEN	COMPUTE		[2]Dyn...	=		[2]Dyn...	-	10	[Comment]
34	[Label]	RETURN																[Comment]
35	[Label]	Double click!																[Comment]
36	[Label] RightUp	IF	(		[1]Dyn...	>	212	)	THEN	COMPUTE		[1]Dyn...	=		[1]Dyn...	-	10	[Comment]
37	[Label]	IF	(		[2]Dyn...	>	312	)	THEN	COMPUTE		[2]Dyn...	=		[2]Dyn...	-	10	[Comment]
38	[Label]	RETURN																[Comment]
39	[Label]	Double click!																[Comment]
40	[Label] LeftDown	IF	(		[1]Dyn...	<	812	)	THEN	COMPUTE		[1]Dyn...	=		[1]Dyn...	+	10	[Comment]
41	[Label]	IF	(		[2]Dyn...	<	812	)	THEN	COMPUTE		[2]Dyn...	=		[2]Dyn...	+	10	[Comment]
42	[Label]	RETURN																[Comment]
43	[Label]	Double click!																[Comment]
44	[Label] RightDown	IF	(		[1]Dyn...	>	212	)	THEN	COMPUTE		[1]Dyn...	=		[1]Dyn...	-	10	[Comment]
45	[Label]	IF	(		[2]Dyn...	<	812	)	THEN	COMPUTE		[2]Dyn...	=		[2]Dyn...	+	10	[Comment]
46	[Label]	RETURN																[Comment]
47	[Label]	Double click!																[Comment]
48	[Label] Lift hand	IF	(		[3]Dyn...	>	512	)	THEN	COMPUTE		[3]Dyn...	=		[3]Dyn...	-	10	[Comment]
49	[Label]	RETURN																[Comment]
50	[Label]	Double click!																[Comment]
51	[Label] Down hand	IF	(		[3]Dyn...	<	812	)	THEN	COMPUTE		[3]Dyn...	=		[3]Dyn...	+	10	[Comment]
52	[Label]	RETURN																[Comment]

(圖 3.1.7)攻擊鴨程式段<三>

LeftUp 左上(32~34)、RightUp 右上(36~38)

使用 U+(L 或 R)組合鍵讓馬達 1 跟 2 以正負 10 度左上與右上移動

LeftDown 左下(40~42)、RightDown(44~46)

使用 D+(L 或 R)組合鍵讓馬達 1 跟 2 以正負 10 度左下與右下移動

Lift hand 抬手(48~49)、Down hand 手向下(51~52)

使用 S+(U 或 D)組合鍵讓馬達-3 以正負 10 度上下移動

53	[Label]	Double click!								[Comment]
54	[Label] Initial pose	LOAD	POS [1]Dyn...	-	512					[Comment]
55	[Label]	LOAD	POS [2]Dyn...	-	512					[Comment]
56	[Label]	LOAD	POS [3]Dyn...	-	812					[Comment]
57	[Label]	CALL	Moving...							呼叫第66行
58	[Label]	RETURN								[Comment]
59	[Label]	Double click!								[Comment]
60	[Label] Attack pose	LOAD	POS [1]Dyn...	-	512					[Comment]
61	[Label]	LOAD	POS [2]Dyn...	-	812					[Comment]
62	[Label]	LOAD	POS [3]Dyn...	-	512					[Comment]
63	[Label]	CALL	Moving...							呼叫第66行
64	[Label]	RETURN								[Comment]
65	[Label]	Double click!								[Comment]
66	[Label] Moving done	IF	MOVING [1]Dyn...	=	1	)	OR			[Comment]
67	[Label]	CONT IF	MOVING [2]Dyn...	=	1	)	OR			[Comment]
68	[Label]	CONT IF	MOVING [3]Dyn...	=	1	)	THEN	JUMP	Moving... (迴圈)	[Comment]
69	[Label]	RETURN								[Comment]
70	[Label]	Double click!								[Comment]
71	[Label]	Double click!								[Comment]
72	[Label]	END								[Comment]

(圖 3.1.8)攻擊鴨程式段<四>

Initial pose(54~58)

給予限定的馬達角度讓機器人手臂挺直

Attack pose(60~64)

給予角度讓機器人手臂向前伸

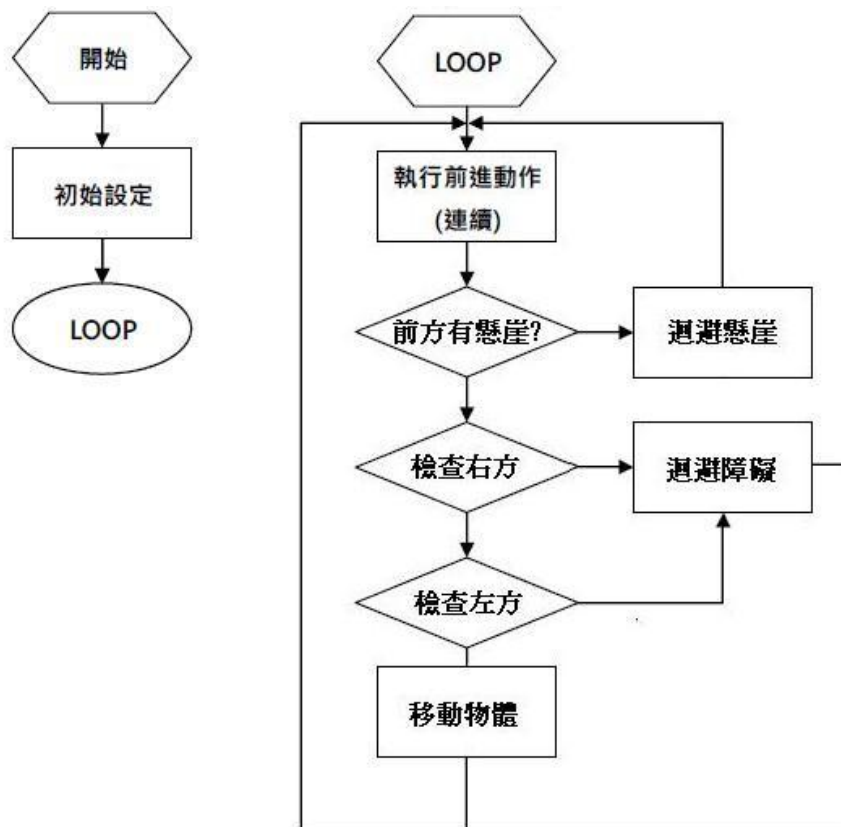
Moving done(66~69)

為旗標副程式移動旗標為 1 時為啟動，再用迴圈讓他持續執行

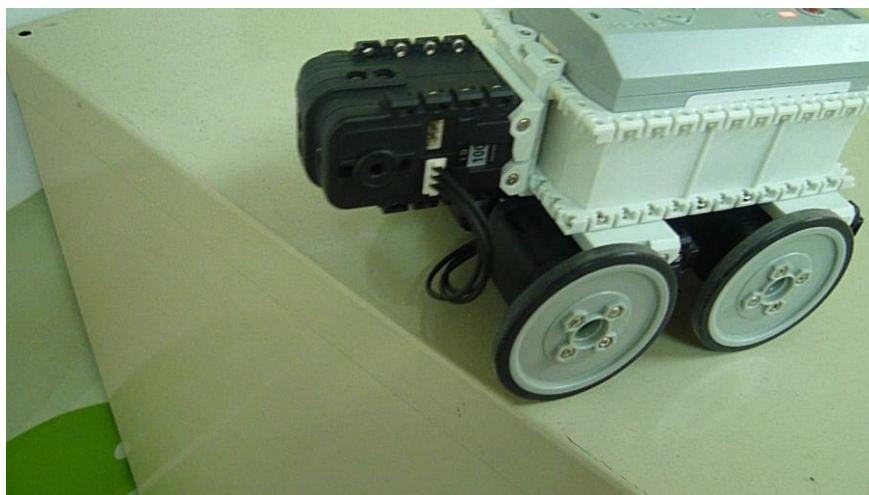


### 3.2 避障判斷車

避障判斷車-流程圖



(圖 3.2.1) 避障判斷車流程圖



(圖 3.2.2) 避障判斷車實體照

### 避障判斷車-程式碼

1	[Labe {}]	START									[Comment]
2	[Labe {}]	CALL	Initia...								[Comment]
3	[Labe {}]	LOAD	Speed	<-	400						[Comment]
4	[Labe {}]	LOAD	Refere...	<-	8						[Comment]
5	[Labe {}]	LOAD	[100]D...	<-	0						[Comment]
6	[Labe {}]	LOAD	[5]Dym...	<-	100						[Comment]
7	[Labe {}]	LOAD	[6]Dym...	<-	100						[Comment]
8	[Labe {}]	LOAD	[7]Dym...	<-	100						[Comment]
9	[Labe {}]	LOAD	[6]Dym...	<-	64						[Comment]
10	[Labe {}]	LOAD	[6]Dym...	<-	64						[Comment]
11	[Labe {}]	LOAD	[7]Dym...	<-	64						[Comment]
12	[Labe {}]	LOAD	[7]Dym...	<-	64						[Comment]
13	[Labe {}]	LOAD	[6]Dym...	<-	380						[Comment]
14	[Labe {}]	LOAD	[7]Dym...	<-	380						[Comment]
15	[Labe {}]	CALL	Initia...								[Comment]
16	[Labe {}]	Double click!									[Comment]
17	[Labe {}] Loop	IF	( [LEFT] [100]D... < 80 ) THEN						JUMP	Avoid ...	[Comment]
18	[Labe {}]	IF	( [100]D... < 150 ) THEN						JUMP	Loop	[Comment]
19	[Labe {}]	LOAD	Watchi...	<-	1						[Comment]
20	[Labe {}]	CALL	Turn r...								[Comment]

(圖 3.2.3) 避障判斷車程式段<一>

障礙物基準觀測、移動速度、馬達旋轉斜率、最大扭力參數與馬達初期跟全轉模式設定，然後再呼叫開機後的初步動作(雙手張開跟舉起雙手然後直線前進)，loop 為機器人的懸崖判斷以回圈方式持續運行，如果前方有懸崖將會右轉迴避。

21	[Label] Examine right	LOAD		Timer	=	1				[Comment]	
22	[Label]	CALL	Waitin...							[Comment]	
23	[Label]	IF	(		[100]D...	<	10	)	THEN	JUMP	Quit t...
24	[Label]	COMPUTE	Watchi...	=	Watchi...	+	1			[Comment]	
25	[Label]	IF	(	Watchi...	>	Refere...	)	THEN	JUMP	Avoid ...	
26	[Label]	JUMP	Examin...							[Comment]	
27	[Label] Quit to examine right	CALL	Stop							[Comment]	
28	[Label]	LOAD	Watchi...	=	3					[Comment]	
29	[Label]	CALL	Turn l...							[Comment]	
30	[Label]	LOAD		Timer	=	2				[Comment]	
31	[Label]	CALL	Waitin...							[Comment]	
32	[Label] Examine left	LOAD		Timer	=	1				[Comment]	
33	[Label]	CALL	Waitin...							[Comment]	
34	[Label]	IF	(		[100]D...	<	10	)	THEN	JUMP	Quit t...
35	[Label]	COMPUTE	Watchi...	=	Watchi...	+	1			[Comment]	
36	[Label]	IF	(	Watchi...	>	Watchi...	)	THEN	JUMP	Avoid ...	
37	[Label]	JUMP	Examin...							[Comment]	
38	[Label] Quit to examine left	CALL	Stop							[Comment]	
39	[Label]	CALL	Turn r...							[Comment]	
40	[Label]	COMPUTE		Timer	=	Watchi...	/	2		[Comment]	
41	[Label]	CALL	Waitin...							[Comment]	
42	[Label]	CALL	Stop							[Comment]	
43	[Label]	CALL	Moving...							[Comment]	
44	[Label]	CALL	Go for...							[Comment]	
45	[Label]	JUMP	Loop							[Comment]	

(圖 3.2.4) 避障判斷車程式段<二>

當偵測到前方有物體時會先停下右轉檢查物體大小(21行)，如果右邊符合範圍會左轉判斷左方(32行)是否符合，只要有一方判斷不符合(等待時間>判斷時間)就會跳躍到迴避障礙(47行)，兩邊都符合後呼叫移動物體(43行)的副程式(75行)進行搬移。



47	[Labe ] Avoid obstacle	[Comment]
	IF < [I00]D... >= 10 > THEN	JUMP Avoid...
48	[Labe ]	[Comment]
	LOAD Timer [- 8	
49	[Labe ]	[Comment]
	CALL Waitin...	
50	[Labe ]	[Comment]
	CALL Go for...	
51	[Labe ]	[Comment]
	JUMP Loop	
52	[Labe ]	[Comment]
	Double click!	
53	[Labe ] Avoid cliff	[Comment]
	CALL Go bac...	
54	[Labe ]	[Comment]
	LOAD Timer [- 14	
55	[Labe ]	[Comment]
	CALL Waitin...	
56	[Labe ]	[Comment]
	CALL Turn r...	
57	[Labe ]	[Comment]
	LOAD Timer [- 25	
58	[Labe ]	[Comment]
	CALL Waitin...	
59	[Labe ]	[Comment]
	CALL Go for...	
60	[Labe ]	[Comment]
	JUMP Loop	
61	[Labe ]	[Comment]
	Double click!	
62	[Labe ] Initialize Motor	[Comment]
	IF < 1:8 != 0 > THEN	LOAD 1:8 [- 0
63	[Labe ]	[Comment]
	IF < 2:8 != 0 > THEN	LOAD 2:8 [- 0
64	[Labe ]	[Comment]
	IF < 3:8 != 0 > THEN	LOAD 3:8 [- 0
65	[Labe ]	[Comment]
	IF < 4:8 != 0 > THEN	LOAD 4:8 [- 0
66	[Labe ]	[Comment]
	IF < 5:8 = 0 > THEN	LOAD 5:8 [- 1023
67	[Labe ]	[Comment]
	IF < 6:8 = 0 > THEN	LOAD 6:8 [- 1023
68	[Labe ]	[Comment]
	IF < 7:8 = 0 > THEN	LOAD 7:8 [- 1023
69	[Labe ]	[Comment]
	RETURN	

(圖 3.2.5) 避障判斷車程式段<三>

迴避障礙(47~51)在偵測前方物體超越範圍後呼叫的副程式，如果物體是右方抓取範圍不符合機器人就會往左前進以此類推。

避免懸崖(53~60)判斷前方為懸崖後會執行後退再進行右轉計時器給予 3 秒多的回轉時間(25\*0.125)。

Initialize Motor(62~69)為馬達初期跟全轉模式設定

70	[Labe]	Double click!													[Comment]
71	[Labe]	Initial Action	CALL	Liftin...											[Comment]
72	[Labe]	CALL	Go for...												[Comment]
73	[Labe]	RETURN													[Comment]
74	[Labe]	Double click!													[Comment]
75	[Labe]	Moving object	CALL	Liftin...											[Comment]
76	[Labe]	CALL	Turn r...												[Comment]
77	[Labe]	LOAD	 Timer	-	14										[Comment]
78	[Labe]	CALL	Waitin...												[Comment]
79	[Labe]	CALL	Stop												[Comment]
80	[Labe]	CALL	Put do...												[Comment]
81	[Labe]	CALL	Turn l.												[Comment]
82	[Labe]	LOAD	 Timer	-	14										[Comment]
83	[Labe]	CALL	Waitin...												[Comment]
84	[Labe]	CALL	Stop												[Comment]
85	[Labe]	RETURN													[Comment]
86	[Labe]	Double click!													[Comment]
87	[Labe]	Lifting object	CALL	Unfold...											[Comment]
88	[Labe]	CALL	Put do...												[Comment]
89	[Labe]	CALL	Grippi...												[Comment]
90	[Labe]	CALL	Liftin...												[Comment]
91	[Labe]	RETURN													[Comment]
92	[Labe]	Double click!													[Comment]
93	[Labe]	Put down object	CALL	Put do...											[Comment]
94	[Labe]	CALL	Unfold...												[Comment]
95	[Labe]	CALL	Liftin...												[Comment]
96	[Labe]	CALL	Grippi...												[Comment]
97	[Labe]	RETURN													[Comment]

(圖 3.2.6) 避障判斷車程式段〈四〉

## Initial Action 初步行動

機器人一開機時的初步動作雙手張開跟舉起雙手然後直線前進。

## Moving object 移動物體

先是呼叫舉起對象的副程式將物體舉起來右轉 2 秒左右( $14 \times 0.125$ )

後停止動作呼叫放下對象副程式機械手臂回歸到當初的位置機器人





















左轉同樣的時間回到一開始的位置。

## Lifting object 舉起對象(87~91)

展開機器手>放低雙手>抓住對象>舉起對象

## Put down object 放下對象(93~97)

放低雙手>展開雙手>舉起雙手>回歸未展開的雙手

99	[Label] Stop	[Comment]
	LOAD  [1]Dyn... [- 0	
100	[Label]	[Comment]
	LOAD  [2]Dyn... [- 0	
101	[Label]	[Comment]
	LOAD  [3]Dyn... [- 0	
102	[Label]	[Comment]
	LOAD  [4]Dyn... [- 0	
103	[Label]	[Comment]
	RETURN	
104	[Label]	[Comment]
	Double click!	
105	[Label] Go forward	[Comment]
	COMPUTE Variab... = 1024 + Speed	
106	[Label]	[Comment]
	LOAD  [1]Dyn... [- Variab...	
107	[Label]	[Comment]
	LOAD  [2]Dyn... [- Speed	
108	[Label]	[Comment]
	LOAD  [3]Dyn... [- Variab...	
109	[Label]	[Comment]
	LOAD  [4]Dyn... [- Speed	
110	[Label]	[Comment]
	RETURN	
111	[Label]	[Comment]
	Double click!	
112	[Label] Go backward	[Comment]
	COMPUTE Variab... = 1024 + Speed	
113	[Label]	[Comment]
	LOAD  [1]Dyn... [- Speed	
114	[Label]	[Comment]
	LOAD  [2]Dyn... [- Variab...	
115	[Label]	[Comment]
	LOAD  [3]Dyn... [- Speed	
116	[Label]	[Comment]
	LOAD  [4]Dyn... [- Variab...	
117	[Label]	[Comment]
	RETURN	
118	[Label]	[Comment]
	Double click!	
119	[Label] Turn right	[Comment]
	COMPUTE Variab... = 1024 + Speed	
120	[Label]	[Comment]
	LOAD  [1]Dyn... [- Speed	
121	[Label]	[Comment]
	LOAD  [2]Dyn... [- Speed	
122	[Label]	[Comment]
	LOAD  [3]Dyn... [- Speed	
123	[Label]	[Comment]
	LOAD  [4]Dyn... [- Speed	
124	[Label]	[Comment]
	RETURN	
125	[Label]	[Comment]
	Double click!	
126	[Label] Turn left	[Comment]
	COMPUTE Variab... = 1024 + Speed	
127	[Label]	[Comment]
	LOAD  [1]Dyn... [- Variab...	
128	[Label]	[Comment]
	LOAD  [2]Dyn... [- Variab...	
129	[Label]	[Comment]
	LOAD  [3]Dyn... [- Variab...	
130	[Label]	[Comment]
	LOAD  [4]Dyn... [- Variab...	
131	[Label]	[Comment]
	RETURN	

(圖 3.2.7) 避障判斷車程式段<五>

## Stop 停止模組程式

將四顆馬達的移動速度設定為 0 來停止行動

## Go forward 前進、Go backward 後退

給予馬達移動速度來行動但因為組裝關係左邊跟右邊的馬達是相反的，所以有一邊的馬達必須要順時針轉用 Variable 來計算馬達速度  
(逆時針 0(最慢)~1023(最快) 順時針 1024(慢)~2047(最快))

前進為 1、3 馬達順時針，2、4 馬達逆時針

後退為 1、3 馬達逆時針，2、4 馬達順時針

## Turn right 右轉

四顆馬達給予的值是 400 也就是四顆馬達都是屬於逆時針旋轉，但因組裝的關係右邊的 1 跟 3 馬達的必須要順時針才會前進所以逆時針會向後，進而讓車轉向右邊

## Turn left 左轉

四顆馬達給予的值為 1024+400 四顆馬達都是屬於順時針旋轉，原理同右轉左邊四顆馬達 2 跟 4 倒退走而左轉。



133	[Label] Gripping pose	LOAD	[6]Dyn...	[-	670					[Comment]
134	[Label]	LOAD	[7]Dyn...	[-	352					[Comment]
135	[Label]	CALL	Moving...							[Comment]
136	[Label]	RETURN								[Comment]
137	[Label]	Double click!								[Comment]
138	[Label] Unfolding pose	LOAD	[6]Dyn...	[-	492					[Comment]
139	[Label]	LOAD	[7]Dyn...	[-	532					[Comment]
140	[Label]	CALL	Moving...							[Comment]
141	[Label]	RETURN								[Comment]
142	[Label]	Double click!								[Comment]
143	[Label] Put down pose	LOAD	[5]Dyn...	[-	465					[Comment]
144	[Label]	CALL	Moving...							[Comment]
145	[Label]	RETURN								[Comment]
146	[Label]	Double click!								[Comment]
147	[Label] Lifting pose	LOAD	[5]Dyn...	[-	825					[Comment]
148	[Label]	CALL	Moving...							[Comment]
149	[Label]	RETURN								[Comment]
150	[Label]	Double click!								[Comment]
151	[Label] Moving done	IF	[5]Dyn...	!=	0	)	OR			[Comment]
152	[Label]	CONT IF	[6]Dyn...	!=	0	)	OR			[Comment]
153	[Label]	CONT IF	[7]Dyn...	!=	0	)	THEN	JUMP	Moving...	[Comment]
154	[Label]	RETURN								[Comment]
155	[Label]	Double click!								[Comment]
156	[Label] Waiting timer	IF	Timer	!=	0	)	THEN	JUMP	Waitin...	[Comment]
157	[Label]	RETURN								[Comment]
158	[Label]	Double click!								[Comment]
159	[Label]	END								[Comment]

(圖 3.2.8) 避障判斷車程式段<六>

Gripping pose 抓住動作 Unfolding pose 展開動作

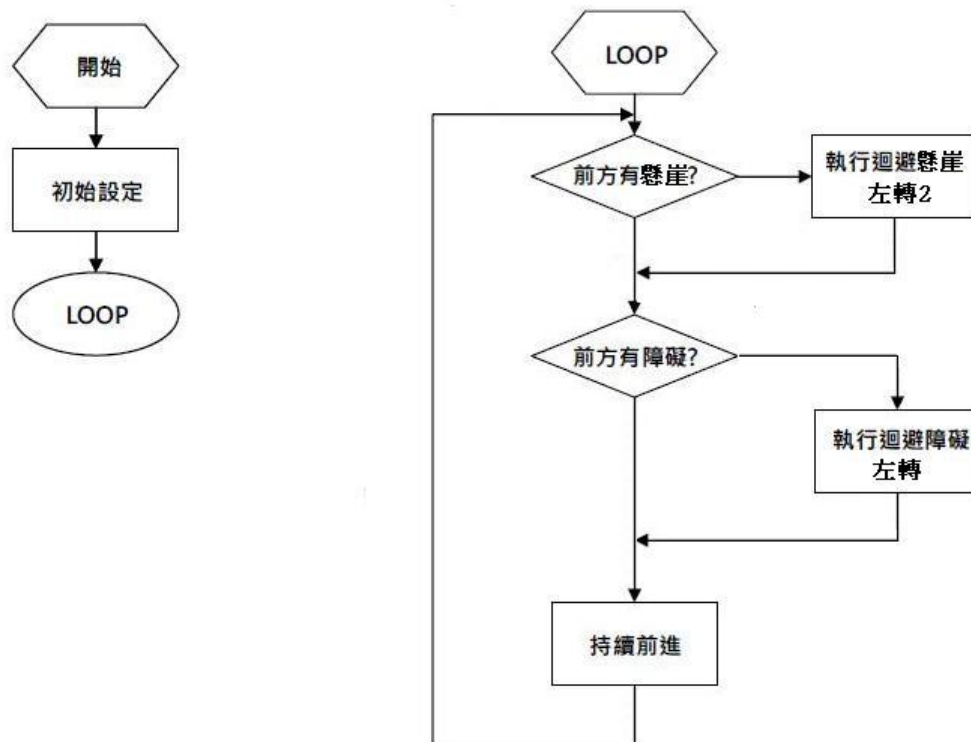
Put down pose 放低動作 Lifting pose 舉起動作

給予固定的角度移動馬達

在呼叫 Moving done 關閉運作中的馬達

### 3.3 障礙判斷排除車

障礙判斷排除車-流程圖



(圖 3.3.1)障礙判斷排除車流程圖



(圖 3.3.2)障礙判斷排除車實體照



(圖 3.3.3)



(圖 3.3.4)



(圖 3.3.5.)

### 障礙判斷排除車-程式碼

1	[Labe]	START								[Comment]
2	[Labe]	CALL	Initia...							[Comment]
3	[Labe]	LOAD	Forwar...	<-	500					[Comment]
4	[Labe]	LOAD	Backwa...	<-	500					[Comment]
5	[Labe]	LOAD	Turnin...	<-	500					[Comment]
6	[Labe]	CALL	Initia...							[Comment]
7	[Labe]	Double click!								[Comment]
8	[Labe] Loop	IF	< LEFT [100]D...	<	100	)	THEN	CALL	Avoidi...	[Comment]
9	[Labe]	IF	< RIGHT [100]D...	>=	200	)	THEN	CALL	Avoidi...	[Comment]
10	[Labe]	JUMP	Loop							[Comment]
11	[Labe]	Double click!								[Comment]
12	[Labe] Initialize motor	IF	< 1:8	!=	0	)	THEN	LOAD	1:8 <- 0	[Comment]
13	[Labe]	IF	< 2:8	!=	0	)	THEN	LOAD	2:8 <- 0	[Comment]
14	[Labe]	IF	< 3:8	!=	0	)	THEN	LOAD	3:8 <- 0	[Comment]
15	[Labe]	IF	< 4:8	!=	0	)	THEN	LOAD	4:8 <- 0	[Comment]
16	[Labe]	RETURN								[Comment]






(圖 3.3.6)障礙判斷排除程式段<一>

一開始給予前進、後退、轉彎速度設定與馬達初期跟全轉模式設定

第 8~9 行為機器人的懸崖與障礙物判斷以回圈方式讓他持續運行

此機器人無設定從控制器(CM-5)給予的指令機器人一啟動就呼叫第 6

行的前進模式。

21	[Label] Avoiding cliff	[Comment]
	CALL Go bac...	
22	[Label]	[Comment]
	LOAD  Timer [- 12	
23	[Label]	[Comment]
	CALL Waitin...	
24	[Label]	[Comment]
	CALL Turn r...	
25	[Label]	[Comment]
	LOAD  Timer [- 16	
26	[Label]	[Comment]
	CALL Waitin...	
27	[Label]	[Comment]
	CALL Go for...	
28	[Label]	[Comment]
	RETURN	
29	[Label]	[Comment]
	Double click!	
30	[Label] Avoiding obstacle	[Comment]
	CALL Go bac...	
31	[Label]	[Comment]
	LOAD  Timer [- 6	
32	[Label]	[Comment]
	CALL Waitin...	
33	[Label]	[Comment]
	CALL Turn r...	
34	[Label]	[Comment]
	LOAD  Timer [- 12	
35	[Label]	[Comment]
	CALL Waitin...	
36	[Label]	[Comment]
	CALL Go for...	
37	[Label]	[Comment]
	RETURN	
38	[Label]	[Comment]
	Double click!	
39	[Label] Waiting timer	[Comment]
	IF (  Timer > 0 ) THEN JUMP Waitin...	
40	[Label]	[Comment]
	RETURN	

(圖 3.3.7)障礙判斷排除程式段<二>

Avoiding cliff 避免懸崖、Avoiding obstacle 避免障礙物

當機器人偵測到前方有懸崖或障礙物時會先後退，等上一段判斷時間

(給予計時器一個數值每隔 0.125 秒減 1)然後再進行左轉後再等上一

段判斷時間後才進行直走。

42	[Label] Stop																		[Comment]
	LOAD	SPEED	[1]Dyn...	←	0														
43	[Label]																		[Comment]
	LOAD	SPEED	[2]Dyn...	←	0														
44	[Label]																		[Comment]
	LOAD	SPEED	[3]Dyn...	←	0														
45	[Label]																		[Comment]
	LOAD	SPEED	[4]Dyn...	←	0														
46	[Label]																		[Comment]
	RETURN																		
47	[Label]																		[Comment]
	Double click!																		
48	[Label] Go forward																		[Comment]
	COMPUTE		Variab...	=	1024		+		Forwar...										
49	[Label]																		[Comment]
	LOAD	SPEED	[1]Dyn...	←	Variab...														
50	[Label]																		[Comment]
	LOAD	SPEED	[2]Dyn...	←	Forwar...														
51	[Label]																		[Comment]
	LOAD	SPEED	[3]Dyn...	←	Variab...														
52	[Label]																		[Comment]
	LOAD	SPEED	[4]Dyn...	←	Forwar...														
53	[Label]																		[Comment]
	RETURN																		
54	[Label]																		[Comment]
	Double click!																		
55	[Label] Go backward																		[Comment]
	COMPUTE		Variab...	=	1024		+		Backwa...										
56	[Label]																		[Comment]
	LOAD	SPEED	[1]Dyn...	←	Backwa...														
57	[Label]																		[Comment]
	LOAD	SPEED	[2]Dyn...	←	Variab...														
58	[Label]																		[Comment]
	LOAD	SPEED	[3]Dyn...	←	Backwa...														
59	[Label]																		[Comment]
	LOAD	SPEED	[4]Dyn...	←	Variab...														
60	[Label]																		[Comment]
	RETURN																		

(圖 3.3.8)障礙判斷排除程式段<三>

Stop 停止模組程式(42~46)

將四顆馬達的移動速度設定為 0 來停止行動

Go forward 前進 (48~53)、Go backward 後退(55~60)

給予馬達移動速度來行動但因為組裝關係左邊跟右邊的馬達是相反

的，所以有一邊的馬達必須要順時針轉用 Variable 來計算馬達速度(

62	[Label] Turn right	COMPUTE	Variab...	=	1024	+	Turnin...	[Comment]
63	[Label]	LOAD	SPEED [1]Dyn...	--	0			[Comment]
64	[Label]	LOAD	SPEED [2]Dyn...	--	Turnin...			[Comment]
65	[Label]	LOAD	SPEED [3]Dyn...	--	0			[Comment]
66	[Label]	LOAD	SPEED [4]Dyn...	--	Turnin...			[Comment]
67	[Label]	RETURN						[Comment]
68	[Label]	Double click!						[Comment]
69	[Label] Turn left	COMPUTE	Variab...	=	1024	+	Turnin...	[Comment]
70	[Label]	LOAD	SPEED [1]Dyn...	--	Variab...			[Comment]
71	[Label]	LOAD	SPEED [2]Dyn...	--	0			[Comment]
72	[Label]	LOAD	SPEED [3]Dyn...	--	Variab...			[Comment]
73	[Label]	LOAD	SPEED [4]Dyn...	--	0			[Comment]
74	[Label]	RETURN						[Comment]
75	[Label]	Double click!						[Comment]
76	[Label] Turn right2	COMPUTE	Variab...	=	1024	+	Turnin...	[Comment]
77	[Label]	LOAD	SPEED [1]Dyn...	--	Turnin...			[Comment]
78	[Label]	LOAD	SPEED [2]Dyn...	--	Turnin...			[Comment]
79	[Label]	LOAD	SPEED [3]Dyn...	--	Turnin...			[Comment]
80	[Label]	LOAD	SPEED [4]Dyn...	--	Turnin...			[Comment]
81	[Label]	RETURN						[Comment]
82	[Label]	Double click!						[Comment]
83	[Label] Turn left2	COMPUTE	Variab...	=	1024	+	Turnin...	[Comment]
84	[Label]	LOAD	SPEED [1]Dyn...	--	Variab...			[Comment]
85	[Label]	LOAD	SPEED [2]Dyn...	--	Variab...			[Comment]
86	[Label]	LOAD	SPEED [3]Dyn...	--	Variab...			[Comment]
87	[Label]	LOAD	SPEED [4]Dyn...	--	Variab...			[Comment]
88	[Label]	RETURN						[Comment]
89	[Label]	Double click!						[Comment]
90	[Label]	END						[Comment]

(圖 3.3.9)障礙判斷排除程式段<四>

上圖中分別有左轉與左轉 2、右轉與右轉 2，這兩項差別在於左轉副  
程式中右邊馬達移動左邊完全不動左轉的角度較小。左轉 2 則是兩邊

馬達右邊前進、左邊後退讓左轉的角度變大

## 第四章 機器狗賽跑

### 4.1 競賽解說

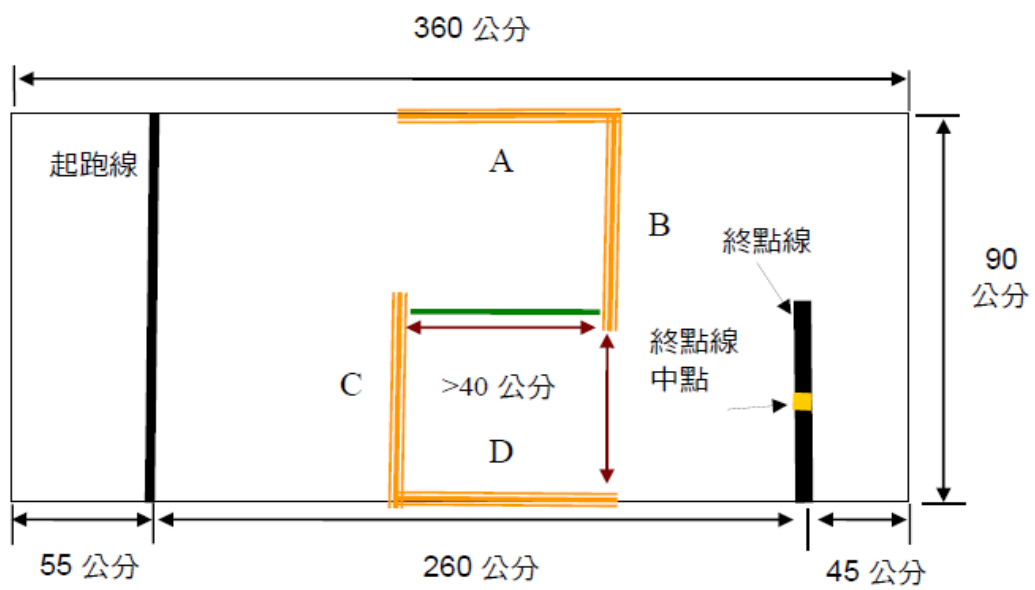
#### 一、機器狗的規定

1. 機器狗必須有四隻腳，在行走時的身高(地面到全身的最高點)及身長(身體的最前方到身體的最後方)不得超過40 公分，全身的重量不限制。
2. 機器狗於行動時必須以四隻腳行走，不得用以規則轉動的輪子著地移動自身(不得在腳底裝設輪子作為前進的裝置)。
3. 機器狗必須由其本身的程式自行控制行走，不得以遙控、電線或其他有形物體直接接觸控制。
4. 機器狗必須以電池作為電源，不得由外部的電源線供應電源。

#### 二、比賽規則

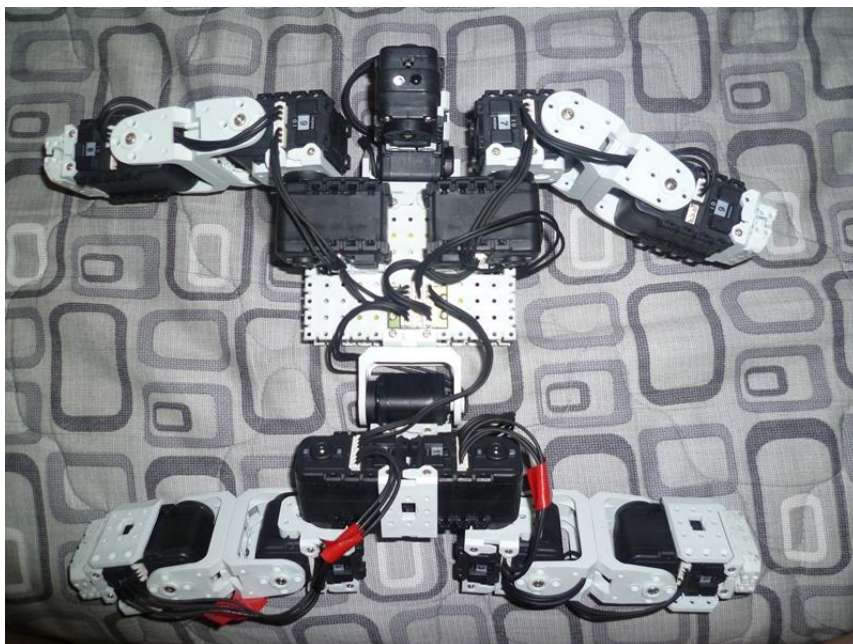
比賽時，機器狗站立於起跑線後方，任一隻腳不得碰觸到起跑線。當裁判發出哨聲後，操控手即可手動啟動機器狗跨越過起跑線，向終點線方向奔跑，當機器狗的任意2 隻腳(機器狗的機身必須在終點線的上方)跨越過終點線即完成比賽。





(圖4.1.1)機器狗比賽場地圖

## 4.2 機器狗設計規格

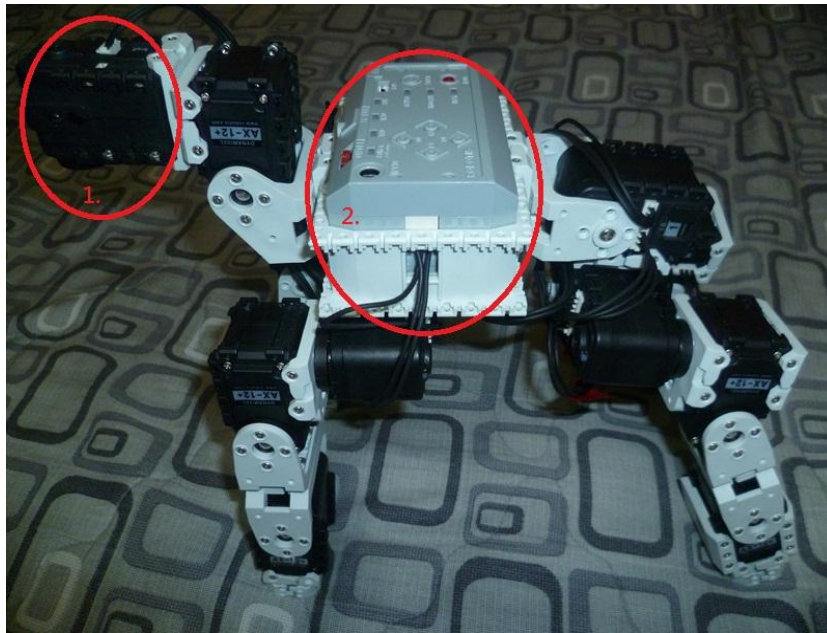


(圖 4.2.1)機器狗規格圖<一>

由 CM5 為主體加上 15 顆 AX-12+馬達、1 顆 AX-S1 感測模組、多種關節組件以及螺絲組合而成



(圖 4.2.2)機器狗規格圖<二>



(圖 4.2.3)機器狗規格圖<三>

身體的主體由 CM5 控制器為中心(如圖 2. 號圈圈)

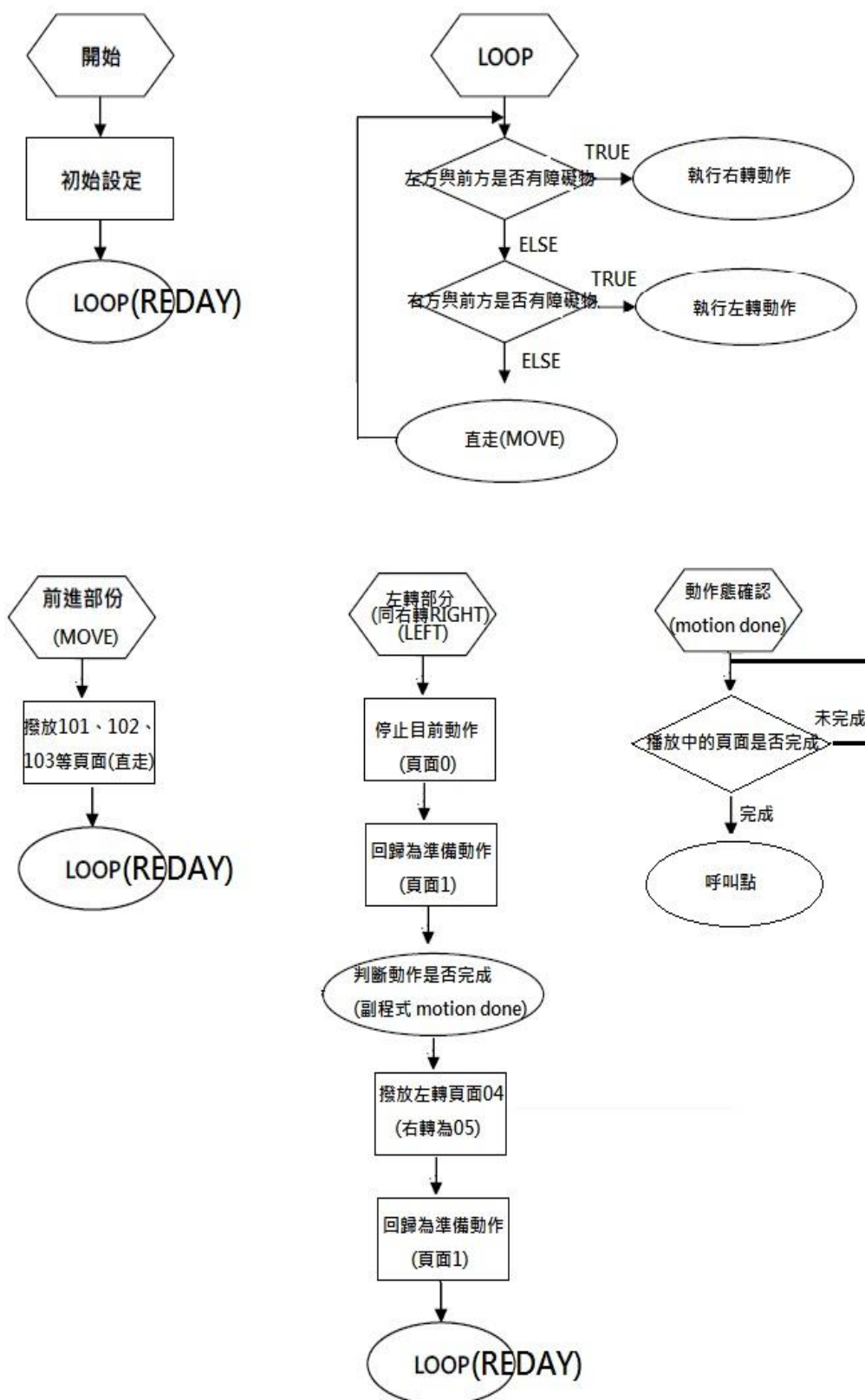
頭部為用作感測環境狀態的 AX-S1(如圖 1. 號圈圈)



(圖 4.2.4)機器狗規格圖<四>

主要動作用的 4 隻腳(如圖圈圈) 每隻以 3 顆 AX-12+組合(1 顆用於左右擺動 2 顆用於前後擺動)

### 4.3 程式流程圖



(圖 4.3.1)機器狗流程圖

## 4.4 機器人程式碼

### <1>主程式

[Label]						[Comment]
1	START	程式開始				
[Label]						[Comment]
2	LOAD	Motion... <-	1	進入準備動作		
[Label]						[Comment]
3	CALL	motion...		呼叫確認動作狀態的副程式		
[Label] ready						[Comment]
4	IF	( [100]D... >= 15 ) AND		如果左側或右側感應器偵測到障礙物並同		
[Label]						[Comment]
5	CONT IF	( [100]D... >= 35 ) THEN		時前方感測器偵測到障礙物則呼叫副程式		
[Label]						[Comment]
6	ELSE IF	( [100]D... >= 15 ) AND				
[Label]						[Comment]
7	CONT IF	( [100]D... >= 35 ) THEN			CALL	left
[Label]						[Comment]
8	ELSE	CALL	move	如果非以上狀況則呼叫副程式"行走"		
[Label]						[Comment]
9	JUMP	ready		跳躍到主程式		

(圖 4.3.2) 機器狗程式段<一>

### <2>副程式

直走

[Label] move						[Comment]
11	LOAD	Motion... <-	101	播放第101頁的動作		
[Label]						[Comment]
12	RETURN	返回呼叫點				

(圖 4.3.3) 機器狗程式段<二>

左轉

14	[Label] left	LOAD	 Motion... <-	0	停止目前的動作
15	[Label]	LOAD	 Motion... <-	1	回復為預備狀態
16	[Label]	CALL	motion...		呼叫確認動作狀態的副程式
17	[Label]	LOAD	 Motion... <-	4	播放第4頁的動作(左轉)
18	[Label]	CALL	motion...		呼叫確認動作狀態的副程式
19	[Label]	LOAD	 Motion... <-	1	再次回復為預備模式
20	[Label]	CALL	motion...		呼叫確認動作狀態的副程式
21	[Label]	RETURN			返回此副程式的呼叫點


(圖 4.3.4)機器狗程式段<三>

右轉

23	[Label] right	LOAD	 Motion... <-	0	停止目前的動作
24	[Label]	LOAD	 Motion... <-	1	回復為預備模式
25	[Label]	CALL	motion...		呼叫確認動作狀態的副程式
26	[Label]	LOAD	 Motion... <-	5	播放頁面5的動作(右轉)
27	[Label]	CALL	motion...		呼叫確認動作狀態的副程式
28	[Label]	LOAD	 Motion... <-	1	回復為預備模式
29	[Label]	CALL	motion...		呼叫確認動作狀態的副程式
30	[Label]	RETURN			回到此副程式的呼叫點

(圖 4.3.5)機器狗程式段<四>

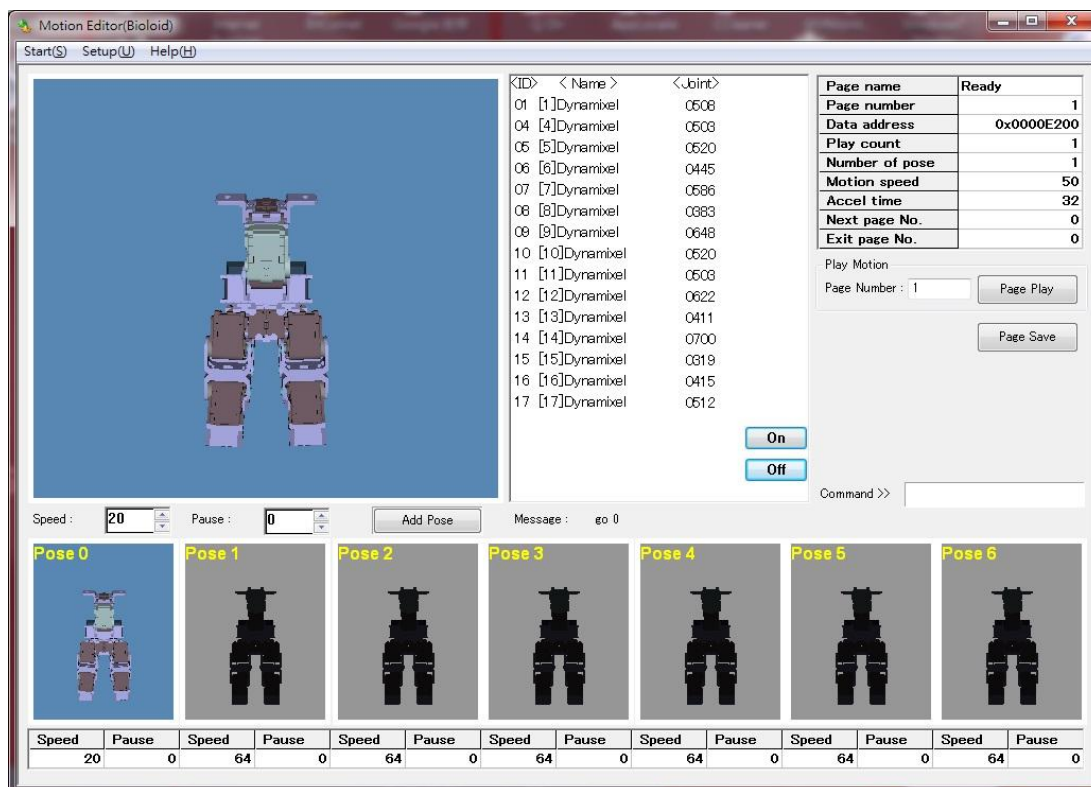
## 動作狀態確認

32	[Label] motion done						[Comment]	
	IF	(	 Motion...	=	1	)	THEN	JUMP
33	[Label]						[Comment]	
	RETURN	確認目前的動作頁面是否在播放中，若未完成則重複此迴圈直						
34	[Label]						[Comment]	
	Double click!	到完成，並執行下一行工作						
35	[Label]						[Comment]	
	END	程式結束點(雖有設立結束點，但此程式為無窮迴圈並不會停止)						

(圖 4.3.6)機器狗程式段<五>

# 動作編輯頁面

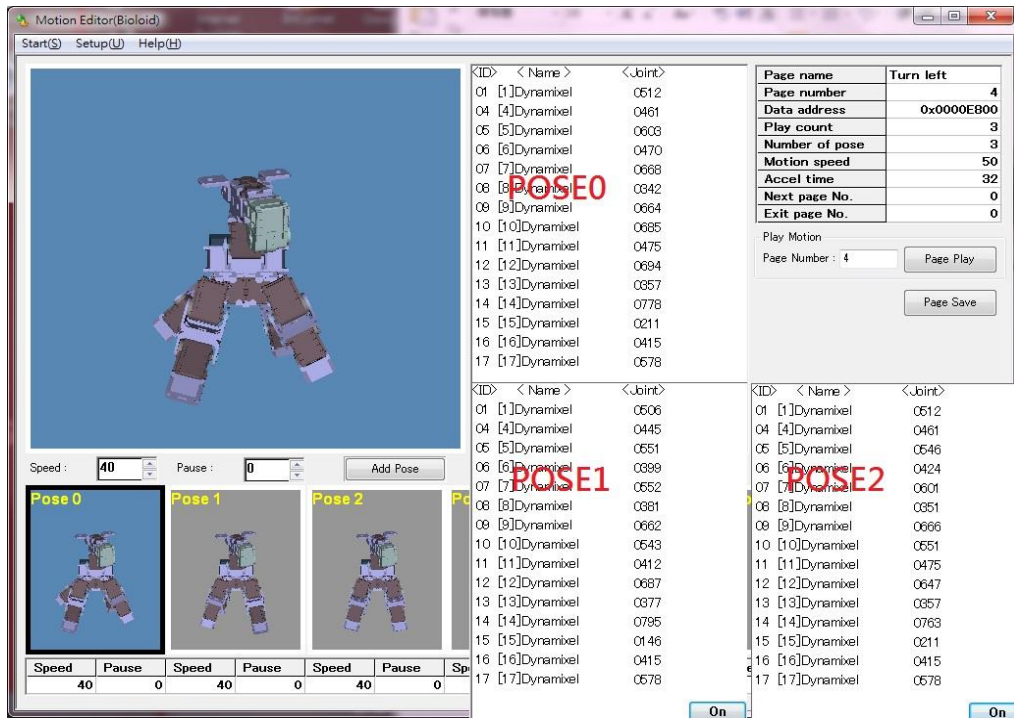
準備狀態



(圖 4.3.7)機器狗動作頁面<一>

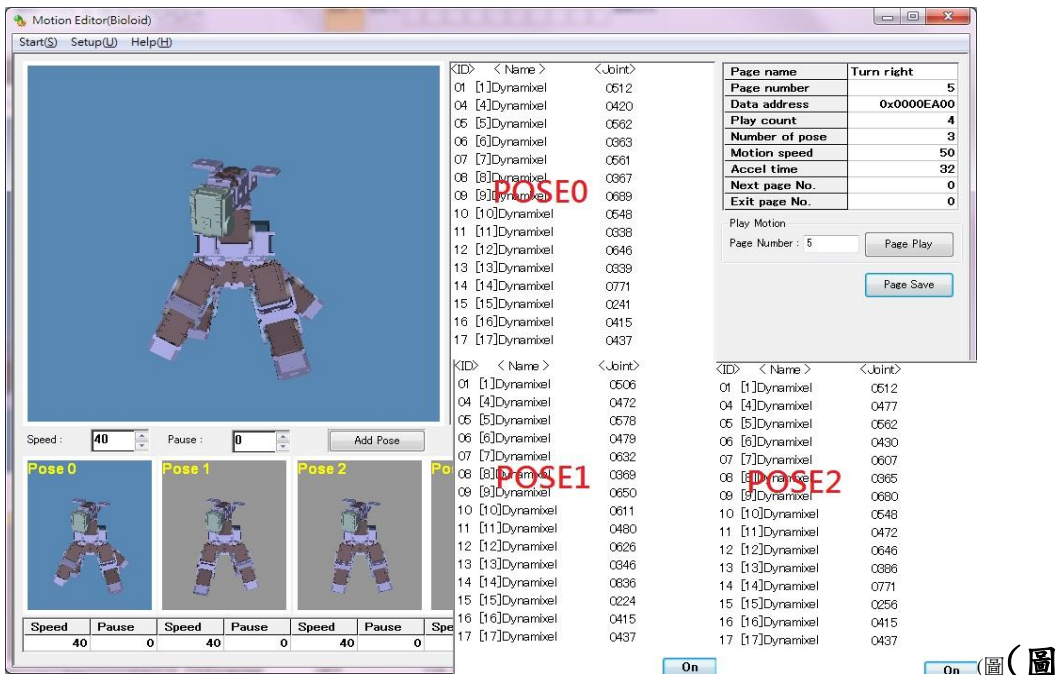


### 左轉頁面



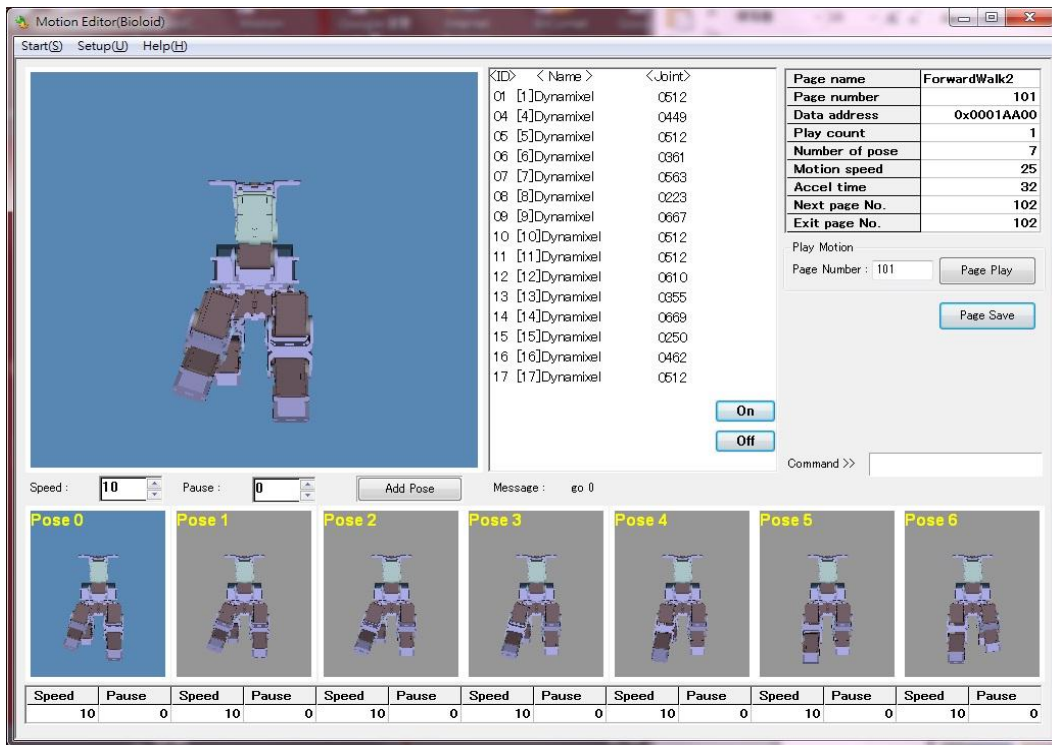
(圖 4.3.8) 機器狗動作頁面<二>

### 右轉頁面

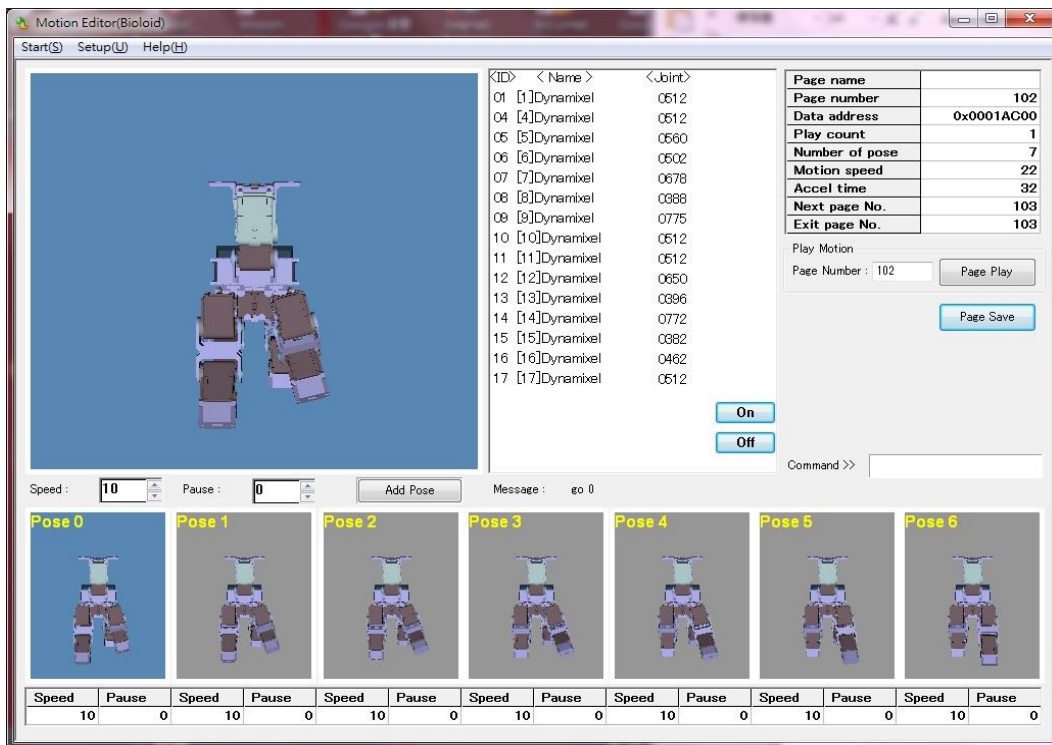


4.3.9) 機器狗動作頁面<三>

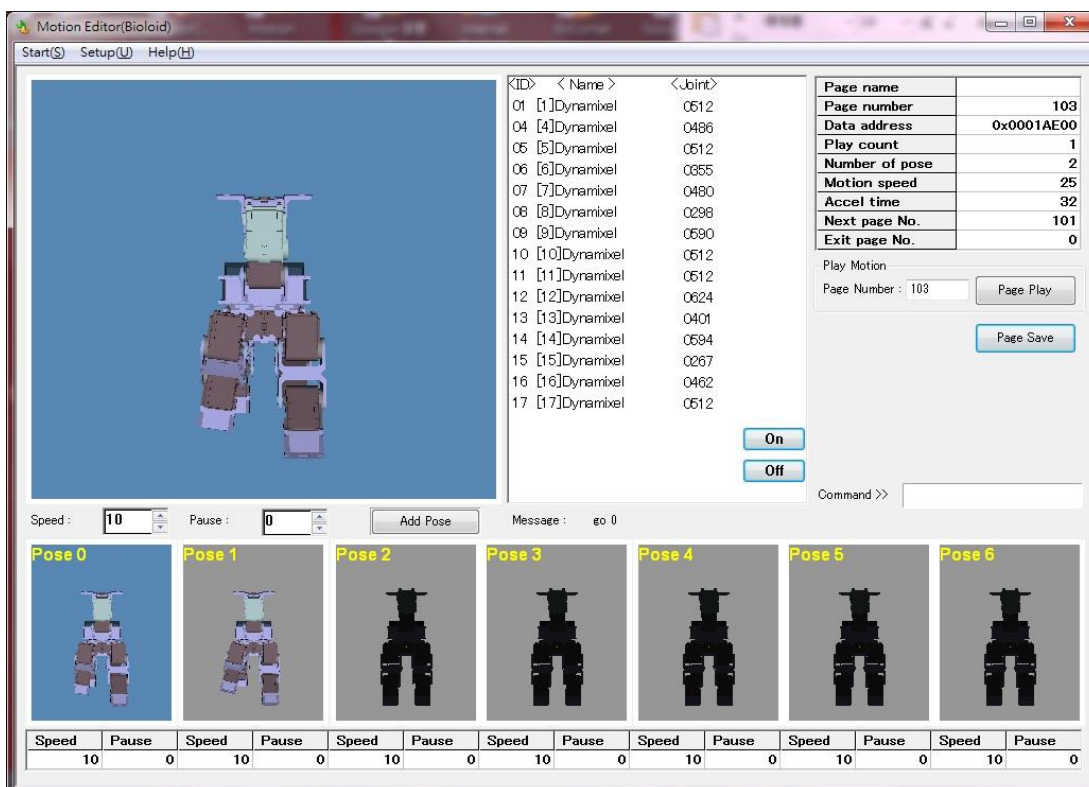
## 行走頁面



(圖 4.3.10) 機器狗動作頁面<四>



(圖 4.3.11) 機器狗動作頁面<五>



(圖 4.3.12)機器狗動作頁面<六>

## 第五章心得

BN97016

本來是要做樂高機器人，所以大二的暑假，便買了樂高機器人的書籍來看，不過因為學弟們上課需要用到樂高機器人，所以改做 BIOLID，這是一個我完全沒接觸過的機器人，跟樂高機器人的差異也滿大的，所以借到 BIOLID 我便回家看看說明書，看看 BIOLID 機器人的零件，它是一個靠螺絲，螺帽及關節馬達才能組成的機器人，因此我們就先買了需要的工具，螺絲起子，零件盒，及一些螺絲以便預防螺絲不夠的時候，再來就安排禮拜三下午及禮拜五下午，來做專題，我記得我們第一個做的是機汽車，當組成能動時，那種喜悅感至今我還記得，不過那台車子還是有點錯，所以我們就安排時間，除錯，實驗，再找更進階的範例機器人來組裝，研究程式，圖控程式….. 等等！直到現在，我們做了三個機器，機器恐龍，機器狗，及機器人，算是不錯的成績，專題讓我們學會自己研究，自己安排時間，甚至回家的時候，也會自動自發地拿起螺絲起子，看書研究，並且不懂時，會上網找資料，跟同學討論，專題讓我學會做機器人，還學會自己看書自己學習，這也算是另一種收穫跟成長。

一開始要開始做專題的時候，我們想了很久最後選定做樂高，我們也利用暑假的期間在研究樂高的書，也看了很多關於樂高的一些資訊、影片，我也有去問學長該怎麼開始做比較好，但最後因為機器不夠，所以松雄老師就建議我們改做 Bioloid，經過我們討論之後，於是我們就決定開始做 Bioloid。

在剛開始做的時候，我其實也不太清楚這是什麼，因為這是我第一次看到這個東西，一打開就看到很多零件，但都不知道該從哪裡著手，後來是慢慢的研究手冊、內建程式、圖控程式、上網查資料…等等，一開始慢慢的從範例開始模仿，到最後才慢慢的進入狀況，但在製作的過程當中我遇到了很嚴重的瓶頸，因為我看不懂程式，所以我要先學會怎麼看程式，這樣才能做之後的動作，於是我花更多的時間去研究它、去了解它，這個問題讓我的進度耽擱了，後來我也克服了這個困難，慢慢的調整我的腳步，讓我的腳步跟上預期的進度，一直到後來開始慢慢的想動作、想程式，慢慢修改的機器人。

後來老師說要去比賽，我們就上網去找資料，經過了大家的討論，我們決定去比機器狗賽跑，於是大家就項比賽邁進，開始作場地、程式，模擬比賽的情況，一直到比賽那天，去到現場真的有一點緊張，看了大家的機器狗好像都很厲害，過了好久，終於到我們了，經過了

一陣子，我們的機器狗出界了，雖然我們出界了，但我們是跑最快的，經過了這個比賽，或許我們失敗了，但我們也學習到很多其他組的經驗，這對我們來說也是一大經驗。

BN97080

在大三的寒假，買來一本樂高機器人的書來看，這是原本要預訂要做的專題，書的內容大多都是在教 C 語言的程式寫法，因此寒假算是複習以前程式語言學的，開學後，要開始做專題，因為學弟們，有新的課程，剛好需要用到大量的樂高機器人，因此我們便開始做這個

ROBBORT !

剛開始的我們，對這個真的很陌生，畢竟我們是第一屆拿這個當專題的學生，他的機器零件，比較沒辦法像樂高機器人，這麼多元，是由數個馬達及感應器組裝而成，而每個零件要組裝成一個機器，需要有不同的螺絲或是螺帽來組裝，因此剛開始的我們，先學會認識這個機器，便從最基礎的機器組起，操作，還記得我們第一個組裝的是四驅車，那是一台，會因前面的障礙，而會轉方向的車子，雖然組裝的過程，有好幾次的失敗，但組裝成功後的成就感，是不可言喻，接下來，我們就把預設的程式給燒錄進去，雖然是可以成功的動作，但仍是有小缺點，於是我們就規劃好，每個禮拜給自己不同的功課有些人認識硬體，有學人學會認識程式的設計，有的時候還會給彼此一些功課和目標呢！

此機器的缺點是外型無法太多元的改變，而且機器的組裝都是靠螺絲去組裝的，需要隔一段時間就檢查看看，螺絲跟羅帽是否鬆脫，以及傳輸的連結線，是否鬆落，這些細節，都是在長時間去做，去琢磨發現出來的，或許只是小小的發現，但卻給自己帶來說不出的成就跟喜悅！

而在專題當中我們學會了如何去分配工作，如何學會學習，從組裝機器到程式認識再到圖控程式的編輯，遇到難關的挫折成就了突破困難的喜悅，我喜歡這樣的努力，專題是要看我們是否有沒有辦法，將這四年所學會的東西融會貫通，更讓我們學會人與人之間要相處跟工作，以及如何面對問題時，用怎樣的態度去面對！



從原來的樂高轉變到 BIOLOID 機器人，其實原本得知要改變計畫的時候有點震驚，並也有點抗拒，但想想可以做更專業暨且更多樣性的機器人的時候，也多少興奮了了起來。

在一開始的過程中，從一個一個零件開始摸索，每個關節每個螺絲跟每個馬達，以及零件特性，都研究了一遍，照著範例做機器人並不難，難的是在過程中，會出現一些，跟預計所不一樣的事情，例如螺絲與鎖點的牢固度，連接線的長度與分配，雖然範例所給的是可行且完整的，但在逐漸摸索後，自己也能萌生一些特別的想法，從按部就班到自我創新，過程雖然不順遂，但也有一階段一階段的突破，像是在作同專題的別組來詢問時，我們便能完整並快速地給出建議，讓他們在研究上能更快的進入狀況，感覺起來這也代表我們真的有研究一番，機器人硬體不難，但在硬體背後的軟體工程，與軟工整合，才是最困難的部分，現階段組員們以瞭解機器人的硬體特性，與基本的軟體形式，接下來就要制訂主要的計畫目標，並去實現原創的軟硬體。

三年級下學期時我們都看著說明書，依上面的安裝過程，程式燒錄的步驟。剛開始我們就先照著說明書，組了一台障礙物偵測車，沒想到光看說明書組裝一台車子那麼困難，花了我們一個下午，還因為沒注意到連接線位置的問題，組裝時擋住了插孔位置，當中還拆拆合合了好幾次，終於組裝好了，興奮的把程式燒進去，結果就動了起來，超開心，然後測試看看，擋左邊向又轉、擋右邊向左轉、擋前面向後退，都 OK，不過就在我們前、左、右都擋時，照理說車子應該在原地停止，不過車子卻一直往前，讓我們百思不解，因為我們都是照著範例做，會有這種結果我們也很驚訝。後來我們就換看程式了，來看看光碟裡面附的程式有沒有問題，我們了幾個禮拜，問了老師跟機器人公司，結果是版本的問題。我們之後又研究動作編輯器，用了動作編輯器就可以讓機器人做你要讓它做的動作了。在這裡我們又發現了一個問題，在動作編輯器裡。我們機器本身的馬達不是鎖上去就好，而是要對準，不然你的機器人的腳就不能走了。之後我們都一直在組裝許多的機器人，也學著看行為控制邏輯的指令，跟修改動作編輯器裡的動作。

做 BioLoid 機器人，我們的重點是參加比賽，所以四年級上學期我們都在為比賽做準備，確定要參加比賽後，我們選了機器狗賽跑比賽，等到比賽規則出來後，我們就依著比賽的規則，組了符合比賽規格的機器狗，依比賽規則，機器狗本身須偵測出前方障礙物並轉彎，不得觸碰到障礙物，然後走完全程。

到比賽當天，我們抽到倒數第三組，所以我們很緊張的看了前面別的學校的機器狗，大家都緩慢得走完全程，到這裡我們組員安心了許多，想說有可能的明，但不知道是我們之前練習太多次導致馬達過載，上場比賽時，跑著跑著就跑出界了，雖然參加這次的比賽沒有得獎，不過我們在之中得到了許多寶貴的經驗。

## 第六章未來展望

正如前面簡介所介紹的，Bioloid 機器人是一種全方位的機器人套件，能夠依據我們的想法，來組裝出各種型態。「Bioloid」是由「bio」+「all」+「droid」三個單字所組成，意謂著「所有生物，都能藉著機器人，具體化的表現出來。」，可以依自己的創意組裝出各種生物，而我們在這次專題裡也組了許多機器人，有機器人、機器狗、機器恐龍…等。因為機器人體積不大，可以容易的進出我們人類進不去的地方，例如：探勘古蹟內部，裝上鏡頭我們就可以輕鬆看到裡面，而得到重要的考古資料，但是目前的機器人有些缺點，就是組裝步驟繁複，組裝時間過久。在未來我們希望能夠推出更容易組裝的機器人套件，讓人輕易的依自己所需要的功能，組裝出符合自己需求的機器人

## 參考文獻

(V1)Bioloid中文使用指南 / 採智科技

(V2)Bioloid中文使用指南 / 採智科技

機器人範例手冊 / 採智科技

快速組裝手冊 / 採智科技

採智科技

<http://www.idminer.com.tw/main.php>

機器人情報網

<http://www.robotworld.org.tw/>

BIOLOID 官方網站

[http://www.robotis.com/xen/bioloid\\_en](http://www.robotis.com/xen/bioloid_en)