



InnoSTAR™

Robotic kit

Inspire Your Imagination

EXPERIMENT GUIDE

華亨科技股份有限公司

感謝您選購華亨科技出品的“創意之星”模組化機器人套件。本書是“創意之星”機器人套件的實驗指導書。本實驗指導書詳細說明瞭如何使用“創意之星”的各種零件，詳細介紹了控制器的設計原理，並用十幾個實例對如何使用“創意之星”機器人套件進行說明。本書不包含機器人結構搭建部分的教學。關於如何搭建機器人結構，請自行查閱《創意之星機器人套件組裝指南》。

“創意之星”機器人套件具有多種版本，不同版本的零件、零件配置不一定相同。在標準配置之外還有眾多的可選配件。本實驗指導書根據“創意之星”機器人套件的最豐富配置進行編寫，如果發現您所選購的版本沒有完成某一個實驗所需的零件，可以選擇跳過本實驗或選購該零件。由於產品升級、配置更改可能導致本手冊內容不盡準確，敬請諒解！

產品光碟中包含有本實驗指導書所提及的所有實驗範例原始碼、電路圖等。

華亨科技　　編輯部 啓

目錄

目錄 3
I 本書使用的標注與格式說明 10
II 前言 10
III 各章節內容簡介 11
第一章 預備知識 13
1.1 完成本書實驗所需要的知識 13
1.1.1 先修課程及知識點 13
1.1.2 本書涉及到的其他知識和技能 14
1.2 典型機器人介紹 14
1.2.1“勇氣”號火星探測器 14
1.2.2 Remotec Andros F6A 排爆機器人 15
1.2.3 iROBOT Roomba 16
1.2.4 本田 ASIMO 仿人型機器人 18
1.2.5 KUKA Titan 工業機器人 19
1.2.6 全向移動機器人 20
1.2.7 蛇形機器人：SolidSnakeII 22
1.2.8 SONY 機器狗：AIBO 24
1.2.9 Segway 兩輪平衡車 25
第二章 “創意之星”機器人套件介紹 28
2.1 結構套件簡介 28
2.1.1 結構零件概述 28
2.1.2 ConnFLEX 連接結構 28
2.1.3 不同版本 29
2.1.4 使用零件 3D 模型 32
2.2 控制器及電源 33
2.2.1 MultiFLEX™2-AVR 控制器 34

2.2.2 PXA270 MultiFLEX™2-PXA270 控制器 36
2.2.3 電池和直流電源 39
2.3 感測器 40
2.3.1 感測器的信號類型及電氣規範 41
2.3.2 “創意之星”感測器介面 43
2.3.1 接近感測器 44
2.3.2 測距感測器 48
2.3.3 聲音感測器 53
2.3.4 碰撞感測器 54
2.3.5 傾覆感測器 54
2.3.6 溫度感測器 55
2.3.7 光強感測器 55
2.3.8 灰度感測器 56
2.3.9 PXA270 視覺和語音感測器 56
2.4 執行器 57
2.4.1 CDS5500 機器人伺服馬達 57
2.4.2 CDS5401 大扭矩 R/C 伺服馬達 62
2.4.3 BDMC1203 馬達驅動模組 64
2.4.4 Sabel 2x10 大功率馬達驅動模組 64
2.4.5 Faulhaber 大功率減速馬達 65
2.5 NorthSTAR 圖形化整合開發環境 66
2.5.1 安裝及使用介紹 67
2.5.2 使用流程圖開發 69
2.5.3 手寫程式碼開發 78
2.5.4 調試與線上監控 79
2.6 其他零件及光碟內容 86
2.6.1 UP-Debugger 下載調試器 86
2.6.2 PXA270 WiFi 無線網路卡或乙太網路線 88

2.6.3 ZigBee 無線通訊模組 90
2.6.4 光碟內容介紹 91
第三章 熟練使用“創意之星” 92
3.1 MultiFLEX™2 控制器 92
3.2 第一次使用 MultiFLEX™2 控制器 92
3.3 使用 MultiFLEX™2 控制器的 IO 介面 96
3.4 使用 MultiFLEX™2 控制器的 AD 介面 101
3.5 使用 MultiFLEX™2 控制器的伺服馬達介面 105
第四章 有趣的防碰撞小車 108
4.1 任務 1：目標分析與計畫 108
4.1.1 結構方案 109
4.1.2 感知方案 111
4.1.3 避障策略 112
4.2 任務 2：讓伺服馬達轉起來 113
4.2.1 搭建輪子結構、接線 113
4.2.2 編寫/下載程式 114
4.2.3 分析結果 119
4.3 任務 3：搭建完整的小車 119
4.4 任務 4：編寫/調試程式，檢查結果 120
4.5 任務 5：總結和作業 123
4.6 擴充任務：更智能的防碰撞小車 124
第五章 會追光的四足步行機器人 125
5.1 任務 1：目標分析與計畫 125
5.1.1 瞭解腿式機器人 125
5.1.2 步態規劃 127
5.1.3 任務規劃 128
5.2 任務 2：搭建機器人 129
5.3 任務 3：讓四足機器人行走 131
5.3.1 設計前進、後退步態 132
5.3.2 設計轉向步態 135

5.3.3 編寫步態 138
5.4 任務 4：讓伺服馬達跟蹤光源 141
5.5 任務 5：讓四足機器人跟蹤光源 152
第六章　擂臺賽機器人（無差別組） 157
6.1 任務 1：目標分析與計畫 157
6.1.1 熟悉比賽規則 157
6.1.2 我們的對策 158
6.1.3 任務規劃 160
6.2 任務 2：熟悉場地、標定感測器 160
6.2.1 比賽場地 160
6.2.2 標定灰度感測器 161
6.2.3 標定紅外測距感測器 164
6.3 任務 3：結構設計和感測器佈置 164
6.3.1 結構設計 164
6.3.2 感測器佈置 167
6.4 任務 5：推棋子練習 168
6.5 任務 6：實戰對抗練習 171
6.6 擴充任務：改進競賽能力 173
6.7 擴充任務：增強速度和力量 173
第七章　擂臺賽機器人（仿人組） 174
7.1 任務 1：目標分析與計畫 174
7.1.1 熟悉比賽規則 174
7.1.2 任務規劃 175
7.2 任務 2：熟悉場地，標定顏色 175
7.3 任務 3：搭建機器人 175
7.4 任務 4：爬坡練習 179
7.5 任務 5：尋找中心練習 180
7.6 任務 5：舉圓柱練習 181
7.7 任務 6：推棋子練習 183
7.8 任務 7：實戰對抗練習 185

7.9 擴充任務：改進競賽能力 187
7.10 擴充任務：增加速度和力量 187
第八章 我們自己的 segway 188
8.1 任務 1：目標分析與計畫 188
8.1.1 瞭解 segway 188
8.1.2 我們自己的方案 188
8.1.3 任務規劃 190
8.2 任務 2：搭建雙輪小車 190
8.2.1 搭建輪腿 191
8.2.2 搭建感測器連接臂 193
8.2.3 搭建擺錘 193
8.2.4 搭建支撐架 194
8.2.5 組裝完整的小車 195
8.3 任務 3：修改伺服馬達的參數 195
8.4 任務 4：紅外測距感測器的標定 196
8.5 任務 5：數位式 PID 演算法的實現 198
8.5.1 背景知識介紹 198
8.5.2 演算法的實現及程式編寫 199
8.6 任務 6：擴充任務 204
第九章 PXA270 語音問答機器人 204
9.1 任務 1：目標分析與計畫 204
9.1.1 語音識別技術 204
9.1.2 語音識別的分類 204
9.1.3 語音對話機器人 205
9.1.4 我們的任務和計畫 205
9.2 任務 2：結構設計和安裝 205
9.3 任務 3：設計對話 206
9.4 任務 4：劇本錄音 207
9.5 任務 5：程式編寫 209

9.5.1 新建專案 209
9.5.2 添加變數 211
9.5.3 初始化語音識別控制項 211
9.5.4 搭建邏輯程式框架 212
9.5.5 添加回答動作 214
9.6 任務 6：程式的執行 217
9.7 任務 7：思考與提高 217
第十章 PXA270 全向運動足球機器人 218
10.1 任務 1：目標分析與計畫 218
10.1.1 瞭解機器人足球賽 218
10.1.2 瞭解全向輪的特點 218
10.1.3 構建我們的全向底盤 219
10.1.4 視覺系統 220
10.1.5 任務規劃 220
10.2 任務 2：結構設計和攝影機安裝 221
10.3 任務 3：原理分析 223
10.4 任務 4：程式編寫 226
10.5 任務 5：顏色閾值標定 237
10.6 任務 6：程式的執行 238
10.7 任務 7：思考與提高 239
第十一章 拓展閱讀及技術資料 240
11.1 有用的拓展閱讀資料介紹 240
11.2 如何獲取相關技術資料 241
第十二章 附錄 242
12.1 MultiFLEX™2 介面示意圖 242
12.1.1 MultiFLEX™2-AVR 242
12.1.2 MultiFLEX™2-PXA270 243
12.2 MultiFLEX™2 原理方塊圖 244
12.2.1 MultiFLEX™2-AVR 244

12.2.2 MultiFLEX™2-PXA270 246
12.3 MultiFLEX™2 開放介面函數一覽 249
12.3.1 MultiFLEX™2-AVR 249
12.3.2 MultiFLEX™2-PXA270 251
12.4 常見問題解答 256
12.4.1 結構相關 256
12.4.2 控制器相關 257
12.4.3 CDS5500 伺服馬達相關 258

I 本書使用的標注與格式說明

本書是“創意之星”機器人套件的實驗指導書，我們試圖將內容講得足夠的詳細和準確。同時，您在閱讀本書時遇到以下三種記號請留神，不正當的操作可能會導致嚴重的後果。



警告。錯誤的操作或使用方式，可能導致設備故障、損毀等。



危險。錯誤的操作或使用方式，可能導致操作人員或附近人員遭受身心傷害。


注意。容易操作錯誤的部分，可能導致危險的操作。

以下三種標誌和上述的警示標誌用處不同，這些標誌分別用於表示：對您可能產生疑問的部分進行附加說明、指出補充資料在光碟的路徑、標誌本小節的內容是 MultiFLEX™ 2-PXA270 控制器專有或“創意之星”高級版專有。


釋疑。對當前所提及的術語、知識點或某段文字等進行詳細說明，起到釋疑解惑的作用。


光碟。指出某檔在光碟的路徑。



PXA270。本章節的內容只適用於 MultiFLEX™2-PXA270 控制器或高級版“創意之星”。

II 前言

本書我們不闡述機器人的學術定義，因爲機器人有太多的定義，而每一種書面的詮釋都顯得蒼白且不夠全面。如果您執著於諸如公式一般精確的定義的話，您可以嘗試使用“Yahoo”或者“Google”等搜索引擎。

我們可以大概瞭解一下機器人的進化歷程：電腦技術興起之前的機器人嘗試成果至多只能稱爲機械人、機械動物，不能稱之爲機器人。1978 年以前，尚無一台電腦或機器人能夠完成如人類一樣最基本的感知、推理，而所有朝這方面開展的研究與探索似乎都陷入了不可掙脫的泥潭。到了 1990 年，家用電腦的運算速度已經超過了 1978 年的大型電腦，具有特別編寫的軟體的機器人已經能夠聽懂少量的人類語言。2005 年，家用吸塵器機器人已經在很

多家庭進行服務。我們也可以看到，機器人技術的發展類似於拋物線，在當前資訊技術、製造技術等強有力推動下，機器人技術已經處於騰飛的時代。

相比而言，人類自身的進化就太過於漫長，人類花了三億年從原始脊椎動物演化爲類似蜥蜴般的動物，花了一億五千萬年進化到了類似老鼠般的生物，花了 7 千萬年達到了猿猴的程度，之後用去了約 5 千萬年才進化成現在的人類。

人類聰慧的大腦與勤勞的雙手在推動著機器人的進化，讓機器人極有可能在未來的 50 年內就完成人類幾億年的進化歷程，這個過程中您可能就是其中最有力的推進者之一，您能想像嗎？對機器人而言，或許您就是造物主！

當然，前提條件是您能夠把這本書的內容學好，這是起步，它將是您機器人技術旅程的一個月臺、一個支點。

知識的爆炸性增長考驗著學習者適應能力的極限，常規的學習手段和方式往往在學習過程結束前，就已經落伍了。所以，我們一直期望站在巨人的肩膀上，以期趕上技術前沿的腳步。“創意之星”機器人教學套件的出發點正是在這，我們不奢望它能成爲支撐您技術成長的巨人，它只是一塊墊腳石，您學習機器人的墊腳石。

機器人有很多種，比如工業機器人、特種機器人、水下機器人、仿生機器人、人形機器人等等。機器人涉及的技術也很多，比如電腦技術、軟體技術、電子資訊技術、機械設計技術等等。本書不專注於某種機器人形態，“創意之星”模組化、可重構的零件設計能夠讓您類比各種機器人形態。本書不深挖於各種機器人技術環節，我們將對這些相關知識進行融合，以最簡單的方式帶您進入技術殿堂，透過本書的學習，您會發現對上述提及的各種知識都已經有所瞭解，並且知道如何去深化學習。

III 各章節內容簡介

本書假設讀者是機械專案、自動化、機械電子專案、車輛專案、熱能或動力專案等相關工科專業的專科、本科學生，並且已經修完大學本科二年級課程（專科二年級），甚至已經修完大學本科三年級（專科三年級）課程，當然，如果您不在上述範圍內，同樣可以順利地閱讀本書，完成全部實驗。只是可能需要額外地學習一些相關的課程知識，才能充分理解本書所提及的知識點。本書的“推薦閱讀資料”列表中列出部分的此類書籍。

以下介紹本書各章節的內容，以便有經驗的讀者進行選擇性閱讀。

本書的第一章首先列出了要完成書中全部實驗所需要的基礎知識，以及相關的先修課程，之後簡明地介紹了本書涉及的其他技能，並在第十一章就如何獲得這些知識和技能給出了簡要的參考，這些內容僅作爲參考，並不一定對每個讀者都適用。第一章還給出了 9 個典型的著名機器人，包含輪式機器人、工業機器人、仿生機器人等各個類別。這部分的目的是讓您對目前世界上存在的各種機器人有個大致的瞭解。

第二章是“創意之星”機器人套件的詳細說明。本章對“創意之星”的結構零件、控制器、感測器、執行器及其它零件進行詳細介紹。從本章開始，您將真正開始“創意之星”的學習旅程。由於“創意之星”組成零件眾多，不可能將所有零件都在本書進行詳細講解，有些零件的詳細使用說明以電子版的形式提供，您可以用電腦查閱。

第三章是“創意之星”控制器的快速入門，用 5 個簡單的例子說明 MultiFLEX™2-AVR 控制器和 MultiFLEX™2-PXA270 控制器 IO、AD、R\C 伺服馬達、機器人伺服馬達等常用功能的使用方法。

從第四章開始，本書用 8 個大專案幾十個實驗任務來指導您使用“創意之星”，和如何藉由“創意之星”進行機器人技術的學習。我們將每一個項目進行細分，以任務爲單元進行講解。透過第四章、第五章，您就能學會 MultiFLEX™2-AVR 控制器的使用、CDS5500 機器人伺服馬達使用、NorthSTAR 圖形化整合開發環境使用、結構搭建等。透過第九章、第十章的學習，您可以掌握 MultiFLEX™2-PXA270 控制器的使用、語音及圖像程式的開發等。

第十二章附錄列出了“創意之星”機器人套件 MultiFLEX™2-AVR 控制器、MultiFLEX™ 2-PXA270 控制器介面示意圖、控制器程式介面函數、控制器電路圖以及常見問題解答等。

第一章 預備知識

1.1 完成本書實驗所需要的知識

1.1.1 先修課程及知識點

創意之星機器人套件可以作爲您進入專案學（尤其是控制學）殿堂的有效入門工具。雖然您可能需要具備大學本科二年級或者專科的知識基礎，才能順利理解全部專案的核心方法，但是只要具備以下知識基礎，您也可以透過各章節充滿樂趣和挑戰的項目實踐，逐漸瞭解專案學；並透過這些專案中的一些重要原則和實踐，發現專案學帶給您的樂趣和成就感：

- 初中畢業或者更高的知識水準；
- 一定的動手能力，包括結構組裝能力、讀圖能力和一定的空間想像能力；
- 初步的機械結構知識基礎，以及初步的力學、物理學常識；
- 熟悉個人電腦使用，具備 Windows XP 或更高版本環境下的一般操作能力。
- 初步的電腦程式設計思想，理解迴圈、條件轉移、變數等基本概念。

如果具備以下基礎，並且認真完成本書的各個項目，您將能夠深入理解現代專案學的幾個核心思想，並將能夠舉一反三，製造出更多具有反應、思考和自我行動能力的機器人（或自動化系統）：

- 大學工科專科的知識水準；
- 熟練掌握 Pro/E、SolidWorks 等至少一樣 3D CAD 軟體，具備建模和虛擬裝配能力；
- 要順利完成各專案中的擴充任務，需要讀者具備一定的工學基礎，包括電子電路、C 語言編程、自動控制基礎（如 PID 控制等）；
- 如果不使用圖形化整合開發環境而要自行編寫程式碼，必需具備 ANSI C 編程基礎。
- 如果要對“創意之星”的控制器進行二次開發，至少需要高中水準的英語基礎及一定的電子電路基礎。機器人上採用的大量感測器、電子元器件的資料手冊(Datasheet)大多是以英文撰寫的，因此一定的英語基礎是必要的。
- 如果用戶要在 MultiFLEX™2-PXA270 控制器上進行深入的二次開發，需要具備 ARM-Linux 方面的開發經驗。

“創意之星”機器人套件的設計原則是：

- 讓入門儘量簡單，無需具備高階語言編程知識，無需具備電子設計經驗，也無須具備結構設計的專業知識。
- 套件的電氣介面、軟體介面、底層程式碼盡可能開放，使進階使用者有深入二次開發的空間。

1.1.2 本書涉及到的其他知識和技能

- 機械設計與機械製造方面：
  - 機械製圖/機械設計；
  - 加工專案學；
  - 專案材料學；
  - 三維 CAD 設計軟體使用；
- 控制方面：
  - 類比/數位電路基礎；
  - 微機原理/AVR 單晶片原理及應用；
  - 自動控制原理；
  - GCC-AVR 單晶片編譯器、下載工具的使用；
  - ARM-Linux 開發知識。

1.2 典型機器人介紹

1.2.1“勇氣”號火星探測器

“勇氣”號（以及功能基本相同的“機遇”號）是迄今人類遣往其他行星上第一個大型、自動化的地面移動機器人。相比“索傑納”號火星探測機器人，“勇氣”號具備更強的功能，更可靠的移動以及更多的感測器。



圖 1.1 “勇氣”號火星車

“勇氣”號採用六輪驅動，透過特殊的懸掛機構可獲得很高的越障能力，使其能越過高於輪子的障礙物。控制方面，它的核心是一台每秒能執行約 2000 萬條指令的電腦，上部上伸出的一個桅杆式結構，其上裝有一對可拍攝火星表面彩色照片的全景攝影機以及各種感測器，距火星車輪子底部高度約爲 1.4 米。它搭載的機械手臂具有與人肩、肘和腕關節類似的結構，能夠靈活地伸展、彎曲和轉動。上面帶有多種工具，其結構簡介如圖 1.1 右圖所示。

“勇氣”號的工具之一是顯微鏡成像儀，以幾百微米的超近距離對火星岩石紋理進行觀察。另外還有穆斯鮑爾分光計和阿爾法粒子 X－射線分光計，可以用來進一步分析岩石構

成。還有一個相當於地質學家常用的小錘子的工具，能在火星岩石上打出直徑 45 毫米、深約 5 毫米的洞，用於研究岩石內部。

其裝載的主要感測器如下：

- 全景攝影機
  - 全景攝影機由兩台高解析度彩色立體攝影機組成，安裝在車上的攝影機架上，它可以做 360 度的水平掃描，攝影機架可以做 180 度的上下掃描，以攝取火星表面和天空的全景視圖。攝影機也用於形成著陸點附近的地形圖、搜索感興趣的岩石和土壤，以尋找火星遠古時期存在液態水的證據。
- 避危攝影機
  - 火星車上分別於車體前端與後端安裝兩組相同的避危攝影機，避危攝影機是由一組立體影像的黑白攝影機所構成的，所拍攝的影像除了用於障礙物偵測之外還用於探險車的路徑規劃上。而後端的避危攝影機與前端相同，並無特別的安排。
- 微型熱散射質譜儀
  - 微型熱散射質譜儀是一種紅外線質譜儀，可以用來遠距離探測岩石和土壤，幫助科學家識別火星的礦物，特別是識別由水的運動形成的碳酸鹽和粘土。
- α 質子 X－射線質譜儀
  - α 質子 X－射線質譜儀用於研究岩石和土壤放射出的 α 質子和 X－射線，以確定岩石和土壤的化學元素。
- 顯微成像儀
  - 顯微成像儀由顯微鏡和 CCD 相機組合而成，可提供火星的岩石和土壤的微觀情況，是對其他儀器的一種補充。
- 磁鐵陣列
  - 每個火星車裝有三個磁鐵陣列用來收集火星氣體、灰塵和岩石粉末中的磁性顆粒，供質譜儀進行分析研究。

所有這些感測器獲得的資訊均經過內置電腦處理，並發送回地面控制站供研究人員研究。

您可以在光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章” 裏找到迪士尼和 NASA 合作拍攝的火星車紀錄片《火星漫遊》。

我們自己的“勇氣號”！

利用“創意之星”套件，您也可以搭建出自己的火星車，可以模仿“勇氣”號的機械結構，實現強大的越野能力。

1.2.2 Remotec Andros F6A 排爆機器人

Remotec 公司是世界上著名的軍火生產廠家，現在是美國軍械製造商 Northrop Grumman 公司的成員。自上世紀六十年代開始製造各種用途的遙操作地面移動機器人產品，主要用於

排爆，核放射及生化場所的檢查和清理，處理有毒、有害物品，特警行動及機場保安。目前該公司爲世界各國的軍隊、員警部門製造的機器人被全球十餘個國家採用，大約有 1 萬台在服役。

Andros 系列機器人有多個型號，例如 Mark5-A1，MK8 等。Andros F6A 是 Mark 5-A1 型機器人的緊湊版。它具有許多獨特的設計功能,是安德魯斯系列產品中功能最多的機器人。

F6A 機器人最突出的特點是具有兩對可更換的輪胎，並且具有前後支臂形式的複合履帶結構。採用輪式行進時，行走速度快，操作人員可以迅速執行任務；採用履帶行進時透過能力強，能爬上普通樓梯，能過 40 釐米寬的壕溝，並且能上下臺階。F6A 機身較窄，尤其適合在狹窄的地方（例如飛機上）進行操作。F6A 採用了安德魯斯系列機器人獨有的活節式履帶，能夠跨越各種障礙，在複雜的地形上行走。

F6A 具有一支六自由度的機械手。機械手採用馬達驅動，馬達的轉動透過減速器減速後帶動手臂關節旋轉或伸縮。操縱該手臂可完成可疑物抓取、轉移等任務，上面配備有帶二自由度雲台的攝影機，電動手爪，以及可配備各種電子點火的武器。

機器人後側安裝有一台可伸縮的攝影機座，上面配置了另一台帶二自由度雲台的攝影機。機器人尾部設有光纖控制線纜卷盤，使得機器人除了可無線遙控之外也可透過光纜進行有線控制。



圖 1.2 F6A 排爆機器人

“勇氣”號火星探測機器人和 F6A 機器人均不具備全局的自主能力。由於這兩者均屬於未知、危險環境探測機器人，當前人工智慧還無法保證機器人在未知環境中自行決定並可靠地完成任務。因此，這類機器人主要依靠遠端操縱人員控制，並具有局部的、一定程度的自主能力，例如斷電前自動收起機械手，或可按預設軌跡行進而無需人工幹預等。

您可以在光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章\排爆機器人”裏找到總多排爆機器人資料。

“創意之星”搭建排爆機器人！

採用一套“創意之星”套件就完全可以製作出與 Andros Remotec F6A 類似的機器人，包括四個輪子、前後擺臂以及機械手臂、雲台。

1.2.3 iROBOT Roomba

以上介紹了兩種典型的特殊應用機器人。特殊、危險環境應用機器人也是機器人的各種應用中發展最早的方向，但機器人真正走入家庭依靠的卻是服務機器人、家政作業機器人等滿足人民日常需要的機器人產品。

自從 1939 年，西屋電氣公司在紐約舉行的世界博覽會上展示了該公司生產的機器人“Electro”以來，人類就開始幻想有朝一日，智慧型機器人能夠承擔起洗碗、洗衣等繁重的家務勞動，將人類從生活的負擔中解脫出來。但直到 iRobot 公司推出第一款全自動家用清潔機器人 Roomba，家用服務機器人才真正開始使得機器人技術進入社會生活。

美國的 iROBOT 公司曾開發了 Packbot 軍用機器人，該機器人曾在阿富汗和伊拉克執行偵察任務。2004 年 iROBOT 公司推出了智慧型機器人吸塵器：Roomba。iRobot Roomba 是一種智慧且高效的吸塵機器人，該機器人採用了許多感測器來感知 Roomba 的周邊環境，並且以每秒 67 次的速度來調整 Roomba 的執行狀態，來保證 Roomba 智慧、高效地、安全地進行清掃工作。如圖 1.3 所示。

iRobot Roomba 具有一個自動化的清潔系統。其中包括一個用來清掃牆邊的旋轉邊刷，兩個相對旋轉的用來捕獲較大碎片的垃圾刷子，還有一個真空吸附裝置用來收集灰塵和小碎垃圾。

研究顯示，在每次清掃任務中，平均一塊地方要被清掃 4 次。Roomba 能夠智慧地引導自己，調節清掃次數來保證足夠好的清掃覆蓋率。對普通人來說，家裏最髒的地方往往是很難清掃到的地方，但是 Roomba 緊湊的身材使它可以鑽入各種傢俱的底部空間，所以，相對於傳統的吸塵器來說，Roomba 可以把家裏打掃的更徹底。

Roomba 的內部結構如圖 2 所示。機器人採用“2 驅動輪差動+全向輪”平臺結構，內部安裝了一個真空吸塵器。機器人四周的光電開關陣列用於檢測邊緣障礙或樓梯邊緣等。其充電插座部分作了特殊設計，透過一組紅外感測器使機器人可以找到帶紅外發射源的充電基座，專門設計的機械結構使得機器人可以停泊在基座上充電。

機器人一側還裝有揚聲器，可以發出各種提示音。可以想像，隨著嵌入式處理器性能和語音處理、語音合成技術的進步，具有一定語音控制和交互能力的服務機器人很快也會出現。

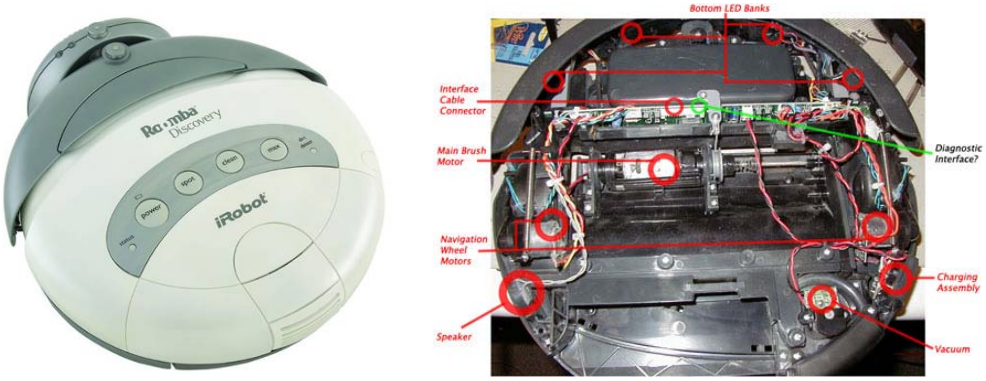


圖 1.3 Roomba 吸塵機器人

Roomba 機器人另外還配有多套“虛擬牆”，即紅外光對射裝置，其中一個發射一定波長的紅外線，另一個接收，用於感測紅外光路上是否有遮擋物，此功能可用於限制 Roomba 的清掃範圍，使得 Roomba 在您指定的地方執行清掃任務。這也是很多自主移動機器人用於限制和設置活動範圍的方法。

您可以在光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章

\Roomba”裏找到關於 Roomba 的更詳盡介紹。

“創意之星”模擬清掃機器人！

採用一套“創意之星”套件就完全可以製作出與 iROBOT　ROOMBA 類似的機器人（當然，沒有漂亮的外殼，也只能用馬達旋轉來模擬真空吸塵裝置）。您需要對控制器進行編程，以讓機器人能夠實現掃地等動作。

1.2.4 本田 ASIMO 仿人型機器人

2000 年，本田技研工業株式會社（以下簡稱本田）發佈了 ASIMO 仿人機器人，這是世界上第一台可遙控、有兩條腿、會行動的仿人機器人。它身高 1.2 米，體重 52 公斤，行走速度可達 1.6 公里／小時。它可以即時預測下一個動作並提前改變重心，因此可以完成“8”字形行走、上下臺階、彎腰等複雜動作。此外，ASIMO 還可以握手、揮手，甚至隨著音樂跳舞。ASIMO 原型機具有 26 個自由度（不包括手部）。本田從 20 世紀 80 年代開始一直致力於類人型機器人的研發。於 1986 年著手研究機器人行走的原理，1996 年研發出機器人原型 P2，1997 年公佈了第一個完全獨立，依靠兩條腿行走的類人型機器人 P3。

“ASIMO”是 P3 的進一步進化，具有與人類實際生活空間貼近的外形設計，以及更接近於人類的行走模式。另外，手臂的活動範圍也有所擴大，透過使用便攜控制器提高了操控性能。



圖 1.4 ASIMO 仿人機器人

“ASIMO”的主要參數如下表所示：

體重	43kg
身高	1,200㎜
行走速度	0～1.6km／h
動作自由度	頭部自由度　2　　腕部自由度　5×2＝10

	手部自由度　1×2＝2　腳部自由度　6×2＝12　合計　26 自由度	
	自由度：與人類關節相當。前後、上下、回轉，各 1 個自由度。	
作動器	伺服馬達＋諧波減速器＋驅動單元	
控制裝置	行走/操作控制單元，無線發送單元	
感測器	腳部：六軸向腳部方位感測器	
	軀體：陀螺儀和加速感測器	
電源部分	38.4V／10AH（Ni-MH）	
操作部分	操縱臺和便攜控制器	

2005 年 12 月 13 日，Honda 發佈的新一代 ASIMO 可以在辦公室等現實環境中作業，具有很高的運動能力。和以往相比，ASIMO 可以和人手拉手走路等，強化了與人配合的行動能力，而且增加了利用手推車搬運物品的功能。此外，新開發出的對這些功能進行統一控制的綜合控制系統，使 ASIMO 可以自行從事接待、嚮導、遞送等服務。並且極大地提高了移動能力，實現了時速 6km 的奔跑及迂回行走。 這是仿人形機器人的一個了不起的進步。

您可以在光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章\ASIMO”裏找到 ASIMO 視頻錄影。

“創意之星”仿人套件！

“創意之星”套件完全可以製作出結構優秀的仿人形機器人。但是需要自行設計一些結構零件，例如機器人胸部外殼等。仿人形機器人的控制比較複雜，推薦有興趣的讀者自行嘗試。

1.2.5 KUKA Titan 工業機器人

KUKA 推出的 Titan6 自由度（6 軸）工業機器人是目前世界上最大最強壯的 6 軸工業機器人。這款機器人的末端最大負載達到驚人的 1000kg，將應用於高負重的自動化場合中。目前 Titan 已經被吉尼斯世界紀錄正式記錄爲世界上最大負重的機器人。

Titan 是一個串聯式關節型機器人，包括腰部回轉、肩關節旋轉、肘關節旋轉、腕關節回轉、腕關節俯仰、末端回轉共六個自由度。



圖 1.5 工業機械手臂

首批 Titan 已經落戶德國的一家玻璃生產廠，主要用於搬運重達 700kg 的玻璃片。很多汽車生產商目前也對 Titan 表示了濃厚的興趣。由於傳統的汽車底盤需要 2 個或者多個機器人聯動才能提起，現在一台 Titan 就解決問題，不但節約了場地也提高了效率。由於 Titan 出色的載重能力，甚至遊樂場也對 Titan 感興趣。這台機器人的末端可以毫不費力地舉起 3-4 個成年人，並作出各種驚險刺激的空中飛舞動作。

您可以在光碟 “.....\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章\ 工業機器人”裏找到眾多工業機器人視頻資料。

“創意之星”的關節式機器人！

“創意之星”機器人套件可以方便地搭建出 3-6 自由度的關節式機器人模型，利用 NorthSTAR 軟體，可以方便地爲搭建出的機器人編程，並且離線執行。

1.2.6 全向移動機器人



圖 1.6 全向足球機器人

北京博創協北京資訊科技大學 Water 隊，參加 2008 年 Robocup 中國公開賽與 RoboCup2009 機器人足球世界盃比賽，獲得了優異的成績，目前在 Robocup 中型組研究中處於領先水準。

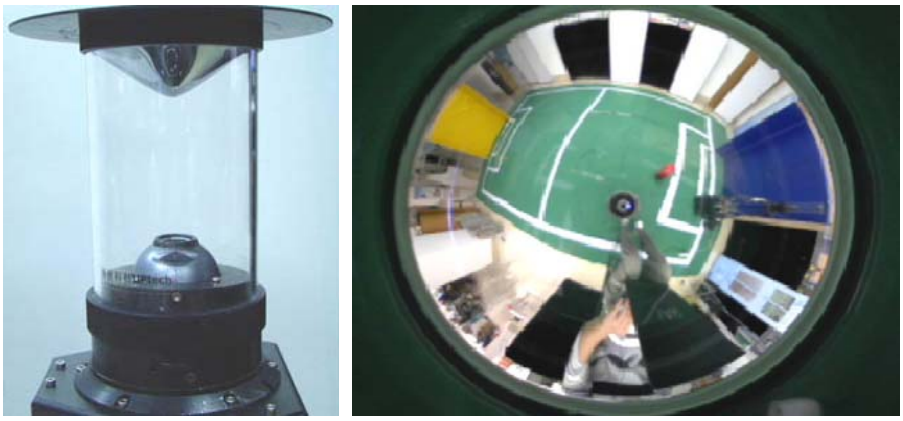


圖 1.7 全景攝影機

視覺感測器：Water 全向視覺系統由全向反射鏡面、USB 數位攝影機和三自由度安裝調節機構組成。全向反射鏡面是透過軟體擬合仿真繪製，特別對遠處的目標進行放大，使視覺的範圍覆蓋中型組整個比賽場地。三自由度安裝調節機構可以調節視覺系統的水平姿態。在影像處理軟體方面，首先對影像進行顏色分割，擷取場地上的白線、球門等標誌物，完成球、球門、其他機器人等目標的識別，進而使用基於全向視覺和里程計資訊的 Monte Carlo 定位方法完成機器人的自定位。全向視覺系統的最終輸出結果是機器人自身位置座標、方向角以及球、球門、其他機器人座標。上層決策系統根據這些資訊做出相應的決策，控制機器人按照一定的規則做出相應的動作。

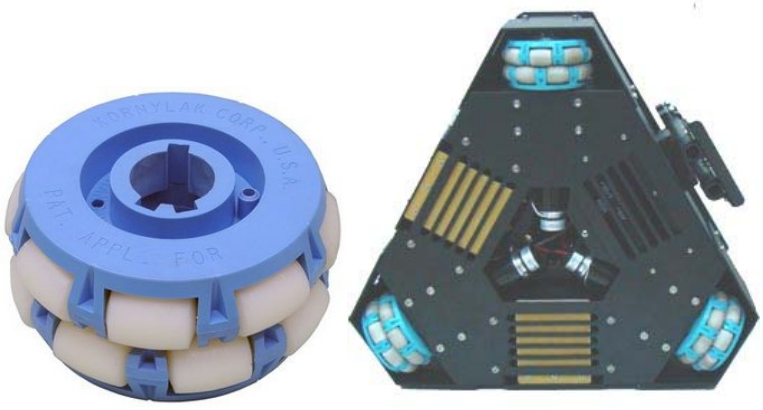


圖 1.8 全向輪

全向運動機構：全向運動機構包括全向輪、馬達、驅動軸系以及運動控制器幾個部分。全向輪是整個運動機構的基礎，由三個全向輪組成的輪系，在馬達驅動下，可以完成平面內 360°任意方向上的運動，運動過程中可以保持任意朝向，這對於比賽中更可靠的控球以及更迅速的運動有著非常重要的意義。馬達透過驅動軸系與全向輪相連接，驅動軸系除了傳導馬達驅動力的作用以外，還能保護馬達不受外力直接衝擊。運動控制器是根據控制系統的控制命令，同時控制多路馬達的關鍵零件，由 ARM 嵌入式處理器作爲主要控制晶片，並配合週邊電路組成。

決策控制系統：作爲整個機器人的“大腦”，決策控制系統要根據感測器提供的場上環境資訊，做出比賽形勢的判斷，選擇合適的比賽策略，協調機器人之間的動作，協調機器人

之間的通信與資訊交換等工作。決策控制系統涉及到多智慧體協同控制、網路通信、模式識別等多領域知識，是目前足球機器人領域研究的一個熱點，也是最有挑戰性的研究領域。參加 RoboCup2008 中國公開賽的 Water 機器人足球隊已經能夠實現比賽環境下的動態角色分配，並能夠完成多機器人編隊任務的實體試驗。可靠的決策系統是順利完成比賽任務的關鍵。

另外，足球機器人還包括射門機構、控球機構、電池組等其他部分。所有這些部分的協調工作，才能完成整個機器人的各種動作，實現控制思想。多機器人協同配合，才能完成比賽任務。

WATER 的主要技術參數如下：

外形尺寸：	45x45x78cm	工作電壓：	24V
質量：	23kg	成像方式：	全向反射鏡
驅動方式：	三輪全向	攝影機：	640x480
速度：	0~4m/s	採集速度：	30fps
加速度：	0~2.5m/s^2	射門方式：	電磁鐵
角速度：	540°/s	通信方式：	802.11b
馬達功率：	3x160W	主控機：	Notebook

您可以在光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章\ROBOCUP”裏找到足球機器人比賽的精彩片段。

“創意之星”的全向機器人！

“創意之星”機器人套件（高級版）可以方便地搭建出類似的全方位移動機器人，具備視覺功能，能夠識別、跟蹤色球。

1.2.7 蛇形機器人：SolidSnakeII

近幾年來，仿生機器人學在機器人領域佔有越來越重要的地位，蛇形機器人由於其結構的特殊性，已成爲仿生領域的研究熱點。蛇形機器人在戰場上的掃雷、爆破，礦井和廢墟中探測營救，管道維修以及外行星地表探測等條件惡劣、要求有高可靠性的領域有著廣闊的應用前景。模組化設計和高冗餘度設計等新思路的提出和逐步完善，使蛇形機器人成爲機器人研究的亮點。

SolidSnake II 結合了國內外蛇形機器人的發展現狀，充分考慮了蛇類生物的運動特點，從仿生學的角度，結合機器人動力學和摩擦學等的相關理論，建立了基於行爲控制理論的蛇類運動學模型，把蛇類生物的複雜運動形式化解爲局部的、簡單的行波狀態，並以固定的相位差沿蛇體進行傳播。採用中央處理機（即蛇的大腦）集中控制的方式把各種運動方式進行合成，實現了機器蛇的蠕動、遊動、側移、側滾、抬頭、翻越障礙物等運動形式。在對蛇類運動機理深入研究的基礎上，得出了利用杆狀結構的角度變化和運動時延，相位差去控制機器蛇運動的速率和運動方向的規律，並在實驗中驗證。

基於行爲控制設計的 solid snake – II 電路系統及程式構架：

電路系統構架：在電路設計上採用“分散式底層運動控制——高層中樞決策”的控制邏輯，透過 IIC 匯流排通訊。PC 端爲控制中樞系統，透過對匯流排的定時輪詢來實現隨時插拔關節。

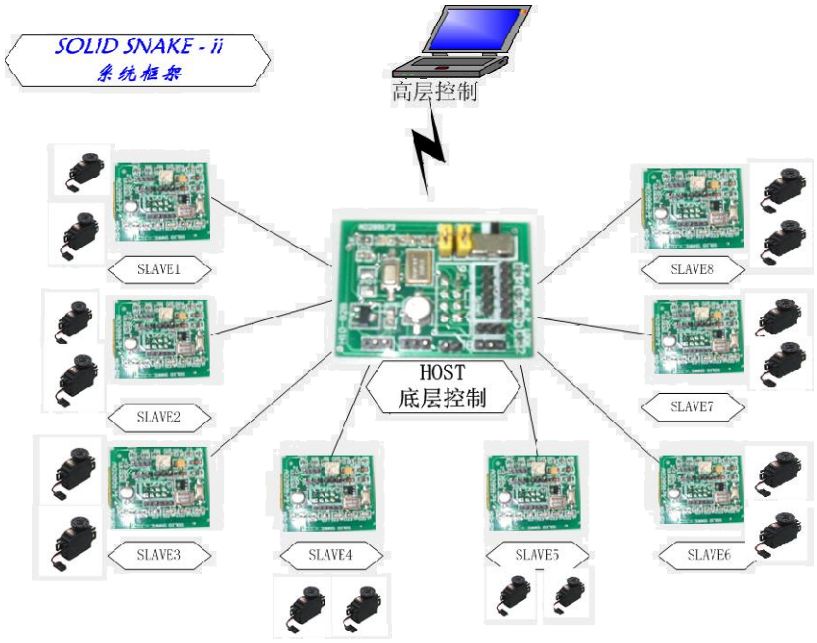


圖 1.9 機器蛇系統架構

蛇形機器人運動的動力學基礎：

蛇類生物的運動方式主要是透過蛇體扭矩的變化以及蛇體與地面的摩擦來共同提供動力。在運動形式上爲行波的傳播表現形式上主要有蠕動和遊動兩種基本方式，以及由這兩種方式相結合產生的各種複雜的運動方式。蠕動主要由垂直方向的扭矩產生角度變化，由於相鄰結構體的約束力和地面的摩擦反作用力的共同作用而產生的一種運動形式。經過力的合成作用產生的水平方向的力可以驅動蛇體向前移動。

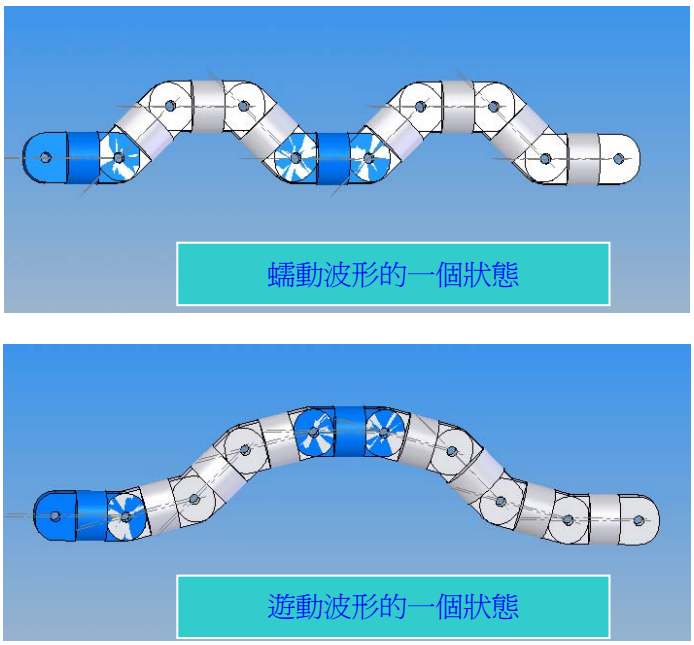


圖 1.10 機器蛇遊動方式

由於蛇形機器人可適應不同的環境要求，在防爆、反恐等領域有著廣闊的應用前景。因此，針對蛇形機器人的遠端控制和自主控制也是一項研究的熱點和難點。SolidSnake II –研究版裝配有攝影機和圖像傳輸系統，透過遠端控制電腦，可實現遠端的監視控制。同時，由於我們採用了按相位傳遞波形的方法，運動的方式可實現隨時調節，因此機器蛇在複雜的環境中有自主避障的運動能力，可穿過複雜的迷宮環境。

SOLID SNAKE – II 的主要技術參數：

項目	單元體長度	全長	直徑	淨重	最大速度	電池續航時間	平均功耗
數據	110mm	0.98m	50mm	1kg	18-20m/min	2 小時	25W

光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章\仿生機器人” 裏包含機器蛇在內的各種仿生機器人資料。

“創意之星”的全向機器人！

利用一套“創意之星”機器人套件，可以很容易地搭建出 8-16 個自由度的蛇形機器人，利用 NorthSTAR 軟體可以很方便地實現蛇形機器人的遊動、蠕動等運動形態，以及基於函數的運動或者加入對感測器資訊的反應，實現智慧化的運動。

1.2.8 SONY 機器狗：AIBO

日本 SONY 公司於 1996 年推出了劃時代的的娛樂機器寵物 AIBO——一隻非常有趣的機器狗。這是第一個正式銷售的寵物機器人，在日本與美國發售開始即被搶購一空。在隨後的

幾年裏，AIBO 發展了五代，外形越來越漂亮，功能越來越強大。AIBO 已經是機器人發展史上一個里程碑式的產品。

這種全新的機器動物寵物就象普通的有生命的寵物一樣，可以透過學習來增進自己的日常經驗，熟悉主人的指示，並且可以像有生命的寵物一樣有自己的情緒表述，會高興、憂愁、悲哀、生氣等等，所有的外界情報及資訊的獲取，都是從其頭部的感應器得來，並經過自己處理後成爲豐富其資訊庫的一部分而保存起來。

AIBO 內置一個生命養成程式，需要主人餵食、並陪它玩耍，AIBO 的各種功能才會逐漸發揮出來。其養成過程大約需要三個月左右，但是三個月後的 AIBO 脾氣稟性則根據其養成者的調教而各不相同了。



圖 1.11 AIBO 機器狗

AIBO 的一些基本資料：
體長：27.4CM
體重：約 1.6KG
動作：可獨立四足行走、發出叫聲、識別顏色及對其他物件的動作做出相應的反應。

光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章” 可以找到 AIBO 機器狗的視頻資料。

“創意之星”的機器狗！

“創意之星”套件可以製作出與 SONY AIBO 類似的機器狗（當然，沒有漂亮的外殼）。您也可以讓機器狗發出叫聲，識別物體，讓它對主人的語音控制做出反應。如果您只有標準版的創意之星機器人，也可以搭建出機器狗，但是無法實現語音控制、視覺識別等功能。

1.2.9 Segway 兩輪平衡車

賽格威（Segway ）是一種電力驅動、具有自我平衡能力的兩輪個人用運輸載具，由美國發明家狄恩·卡門與他的 DEKA 研發公司（DEKA Research and Development Corp.）團隊發明設計，並創立賽格威有限責任公司（Segway LLC.），自 2001 年 12 月起將賽格威商業化量產銷售。

Segway 的運作原理主要是建立在一種被稱爲“動態穩定”（Dynamic Stabilization）的基本原理上，也就是車輛本身的自動平衡能力。以內置的精密固態陀螺儀（Solid-State Gyroscopes）來判斷車身所處的姿勢狀態，透過高速的中央微處理器計算出適當的指令後，驅動馬達來做到平衡的效果。假設我們以站在車上的駕駛人與車輛的總體重心縱軸作爲參考線。當這條軸往前傾斜時，賽格威車身內的內置電動馬達會產生往前的力量，一方面平衡人與車往前傾倒的扭矩，一方面產生讓車輛前進的加速度，相反的，當陀螺儀發現駕駛人的重心往後傾時，也會產生向後的力量達到平衡效果。因此，駕駛人只要改變自己身體的角度往前或往後傾，賽格威就會根據傾斜的方向前進或後退，而速度則與駕駛人身體傾斜的程度呈正比。原則上，只要賽格威有正確打開電源且能保持足夠運作的電力，車上的人就不用擔心有傾倒跌落的可能，這與一般需要靠駕駛人自己進行平衡的滑板車等交通工具大大不同。



圖 1.12 segway 代步車

“創意之星”仿人套件！

“創意之星”套件完全可以製作出結構優秀的仿人形機器人。但是需要自行設計一些結構零件，例如機器人胸部外殼等。仿人形機器人的控制比較複雜，推薦有興趣的讀者自行嘗試。

車輛的能量來源是兩塊鎳氫（NiMH）充電電池，較後期的車款上也可以選配蓄電量更大的鋰充電電池。除了前後傾修正與前進後退外，賽格威的轉向可透過兩種不同的方式實現，其中一種是如同大部分的腳踏車類或摩托車類交通工具一般，駕駛人在車輛持續前進或者後退的狀態中將自己的身體重心往左右傾斜，利用自身重量所產生、與車身縱軸垂直的分量，作爲轉彎時的向心力而達到轉向的目的。除此之外駕駛人也可以扭轉賽格威的龍頭（把手）部份，使車輛左右兩個車輪產生轉速差，例如當龍頭向左轉時，右輪的轉速會比左輪快，達到向左轉的效果。必要時，賽格威甚至可以做出一輪向前一輪向後的動作，達到原地轉向的效果，因此大幅提升了這種交通工具的機動性。這種高度的機動性，再加上玻璃纖維材料製成的車輪，其踏面面積其實不比人類的雙腳大不了多少，因此理論上賽格威可以到達人類所能走到的大部分地方，甚至包括路邊的人行道或落差不會太大的階梯。

光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章\Segway” 可以找到 Segway 視頻資料。



用“創意之星”模擬 segway！

本書不討論 Segway 這種形式的交通工具是否會普及，但是作爲一種運用了高靈敏度感測器、高速和精密的控制演算法的案例，嘗試自己製作一個類似的自平衡機器人是一件很有樂趣、又充滿了挑戰的事情。創意之星完全可以做出類似 Segway 的兩輪自平衡小車模型。本書的第八章是一個很好的範例，您可以自己嘗試一下。

第二章　“創意之星”機器人套件介紹

打開“創意之星”的包裝箱，我們會看到很多零件，有塑膠零件、電路板、馬達、線等等。這些都是什麼東西？怎麼使用？有什麼注意事項？本章將爲您解答這些疑問，向您詳細介紹“創意之星”的所有部分。

我們將從機械結構、控制器、感測器、執行器、開發環境等方面來詳細介紹“創意之星”機器人套件。由於有些零件的詳細使用說明篇幅比較長，不能將之完全納入本書，我們給出了這些參考文件的光碟路徑，請您利用電腦進行查閱。

2.1 結構套件簡介

2.1.1 結構零件概述

“創意之星”是一種模組化機器人元件，其特點是組成機器人的各種零件都是通用、可重組的，各個零件之間有統一的連接方式，零件之間可以自由組合，從而構建出各種各樣的機器人結構。從光碟 “……InnSTAR 產品光碟\結構資料\結構圖片”，您可以看到各種各樣、不同形狀的、不同功能的機器人結構，這些都是專案師或者使用“創意之星”的使用者搭建的。

爲了保證“創意之星”機械結構的剛性、牢固，如下圖所示零件之間採用面接觸、鍵連接、螺釘緊固的方式進行組合。

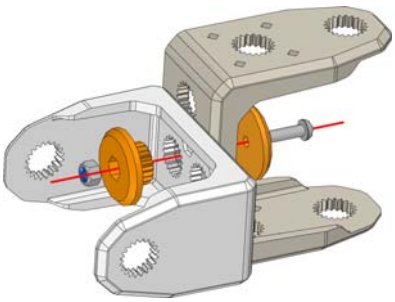


圖 2.1 典型連接方式

“創意之星”結構零件根據功能可以分爲 7 大類：LUI 型結構零件、連接零件、伺服馬達結構零件、機械手臂結構零件、履帶全向輪結構零件、仿人結構零件和其他結構零件，詳見圖 2.8。LUI 型零件和伺服馬達結構零件是機器人骨架的主體，透過連接零件將其組合在一起構成機器人主體；仿人結構零件、履帶全向輪結構零件、機械手臂結構零件都是比較專用的零件。

2.1.2 ConnFLEX 連接結構

在“創意之星”機器人套件裏，各種結構零件的連接遵循統一的標準。我們這種標準連

接方式稱爲“ConnFLEX 連接結構”。 ConnFLEX 連接結構主體是統一尺寸的花鍵和鍵槽，每個花鍵或鍵槽具有 20 個齒，每個花鍵可以轉動的角度是 18°的整數倍。因此，ConnFLEX 連接結構使得各個零件之間可以不同角度連接，例如平行連接、垂直連接，或者其他以 18°爲增量的角度來連接。爲了適用各種連接場合，還派生出兩並聯、三並聯、加長型、扁形等零件，如圖 2.2 所示。

ConnFLEX 連接結構連接精密、無間隙，單個連接可承受 5Nm 以上的彎矩和扭矩，如圖 2.3 所示。ConnFLEX 連接結構連接簡單，只需要一顆螺絲，就可實現牢固的連接效果，兩個或多個零件透過連接件可以實現各種角度的連接組合，如圖 2.4 所示。

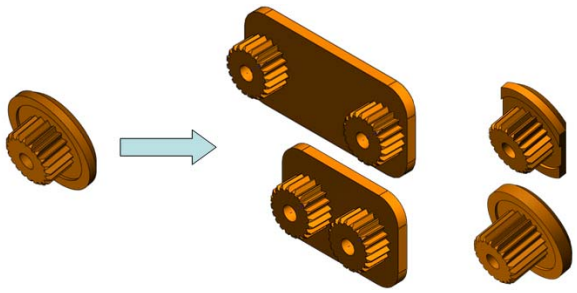


圖 2.2 花鍵的種類

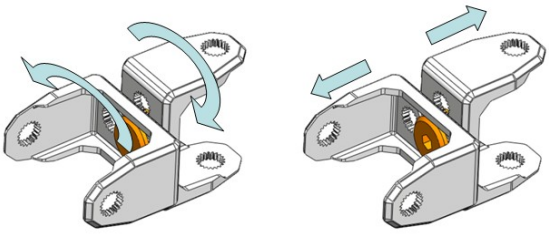


圖 2.3 ConnFLEX 連接結構的受力示意圖



圖 2.4 ConnFLEX 連接結構

2.1.3 不同版本

從“創意之星”的產品樣本上，您會發現“創意之星”有三個版本：入門版、標準版、高級版。我們利用圖 2.5、2.6、2.7 的 19 個典型結構作對比，說明這三個版本在零件、機器人伺服馬達和其他零件的差別。當然，這 19 個典型結構只是例子而已，並不是說“創意之星”只能搭建這麼些結構。



什麼是機器人結構？

我們在此將由特定零件、執行器、感測器等搭建組合而成的，能夠完成某些動作或行爲的機械、控制、感測的總和稱爲機器人結構。由於機器人種類繁多，機器人結構不一定像人，它可能是仿生爬蟲、輪式小車、履帶小車等等。

- **入門版**

入門版較爲簡單，伺服馬達、零件和感測器數量也較少，但價格較爲低廉，適用於機器人愛好者創作獨特的機器人；也可供初中、高中學生發展機器人創作和機器人相關實踐活動。

入門版使用 MultiFLEX™2-AVR 控制器，在 2.2.1 小節您可以看到這款控制器的所有功能。

入門版的零件數量、種類比標準版和高級版要少些，在《創意之星機器人套件組裝指南》P6 您可以看到詳細的零件清單。圖 2.5 沒打叉的部分是入門版的典型結構，從圖中我們可發現入門版的結構都相對比較簡單，沒有全向輪和履帶的結構。

入門版能做什麼呢？這個是您最關心的問題，也是我們在設計時重點考慮的問題之一。

1、 興趣小組、專案訓練中心的實踐器材。入門版價格低廉，零件數量對初級使用者而言已經非常充足，對學校的興趣小組、實踐中心而言，入門版是最合適的選擇。
2、 機器人比賽。使用“創意之星”入門版可以組織很多比賽，類似當前高職機器人比賽、創意機器人設計比賽、迷宮機器人比賽、循跡機器人比賽、機器人擂臺賽等等。
3、 單晶片、自動控制、C 語言、電子設計實驗平臺。區別於常規的單晶片實驗板，“創意之星”可以很生動地將您的程式碼以動作和行爲的方式展示出來，用它來輔助學習可以達到事半功倍的效果。



圖 2.5 入門版能夠搭建的結構

- **標準版**

標準版使用 MultiFLEX™2-AVR 控制器，結構零件種類和數量都很豐富，如圖 2.6 所示，除了入門版的結構之外還能夠搭建諸如 4 足爬蟲、機器蛇等複雜的機器人結構。

標準版主要適用於以下領域：

- 作爲大學工科學生的創新實訓課程教具和實驗器材。
- 作爲大學工科學生的課程設計或者畢業設計平臺。
- 供機器人研究者用於驗證理論演算法、驗證學術論文的結論。
- 供機器人研究者在開發新的機器人之前搭建快速原型，驗證原理和可行性。
- 供機器人發燒友用於創作獨特的、自己的機器人。
- 供初中、高中學生的機器人創作和機器人相關實踐活動。

在《創意之星機器人套件組裝指南》P7 您可以找到標準版詳細的零件清單，在 2.2.1 小節您可以看到 MultiFLEX™2-AVR 控制器所有功能。



圖 2.6 標準版能夠搭建的結構

如圖 2.6 所示，標準版能夠搭建仿生類的蛇形機器人、四足爬蟲、六足爬蟲；輪式機器人；簡單仿人形機器人；機械手臂等結構。當然，這些結構只是提供的範例，起到拋磚引玉的作用，相信透過這些結構熟悉“創意之星”機器人套件後，您可以搭建出更加有趣的結構來。有位同學使用 6 個伺服馬達和若干結構零件，搭建了一個並聯機器人，能夠控制毛筆的空間姿態，從而實現寫毛筆字的控制，您可以再光碟 “……\InnSTAR 產品光碟\結構資料\結構視頻”裏找到《小書童》的精彩視頻。

- **高級版**

高級版是“創意之星”功能最豐富的一個版本，使用 MultiFLEX™2-PXA270 控制器，擁有 50 餘種，760 餘個結構零件，12 種 25 個感測器，20 個機器人伺服馬達。足夠搭建出如圖 2.7 所示的每一種機器人結構。



圖 2.7 高級版能夠搭建的結構

從 2.2.2 小節，您可以詳細瞭解 MultiFLEX™2-PXA270 控制器的功能。MultiFLEX™2-PXA270 控制器除了 MultiFLEX™2-AVR 控制器的所有更能外，還擁有強大的圖像、語音處理能力，能夠識別顏色和人的語言，基於此您可以搭建很多有趣的結構，如：

1、 聽得懂主人的語言，會說話的小狗。您可以和它進行交流，指導它的行爲，塑造它的思想。

2、 追小球的小車。它能看見各種顏色的小球，會像小貓玩毛線團一樣淘氣地玩小球。

您可以在光碟 “……\InnSTAR 產品光碟\結構資料\結構視頻” 裏找到範例的實際視頻。

MultiFLEX™2-PXA270 控制器的圖像和語音處理能力，讓機器人真正擁有了視覺、聽覺和語言能力。透過本實驗指導書的學習，您充分掌握了這些方面的開發技巧後，您可以創造出令人不可思議的機器人來。

此外，創意之星的各種零件都可以單獨訂購，您也可以自行製造感測器、控制電路。以上三個版本只是爲了使用者選擇方便，由華亨科技按照不同要求而精心搭配的三種典型組合而已。

2.1.4 使用零件 3D 模型

如圖 2.8 所示，創意之星機器人套件總共有 50 幾種零件，這些零件可以任意組合，搭建出各種各樣的機器人結構。如圖 2.7 所示，仿生機器人、輪式機器人、履帶式機器人等等都是創意之星擅長搭建的機器人結構。

結構搭建是創意之星機器人套件使用的第一步，要充分瞭解各種零件的特性、組合特點和搭建技巧請仔細閱讀《創意之星機器人套件組裝指南》。

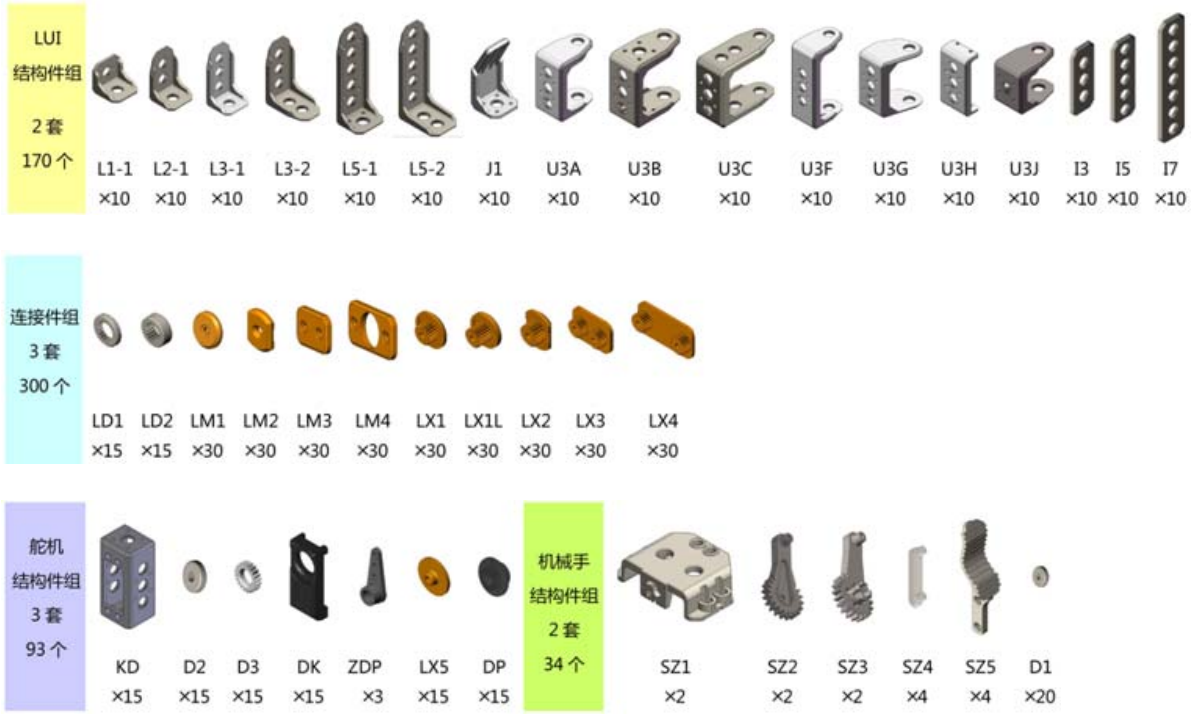




圖 2.8 “創意之星”零件匯總

配套光碟 “……\InnSTAR 產品光碟\結構資料\3D 模型”裏有各種零件的 3D 模型，使用者可以利用機械設計軟體載入這些 3D 模型，在電腦上搭建三維虛擬平台，在電腦上驗證透過了再動手搭建結構，這樣可以避免重複勞動，可以更好發揮創意。

2.2 控制器及電源

“創意之星”有兩個版本的控制器：以 ATmega128 爲核心的 MultiFLEX™2-AVR 控制器，以 PXA270 爲控制核心的 MultiFLEX™2-PXA270 控制器。

爲什麼我們要以核心處理器來命名和區分“創意之星”控制器呢？因爲處理器是這兩個控制器的本質區別。我們可以把機器人控制器當成一種“電腦系統”，它和家用電腦一樣都是以能夠進行高速資料運算的 CPU（中央處理器）爲核心和其他週邊設備組成，能夠運用事先儲存的程式，自動、高速地進行大量數值計算和各種資訊處理的現代化智慧電子設備，其性能強弱主要的衡量標準就是 CPU 的運算能力。

MultiFLEX™2-AVR 控制器的處理器是 ATMEL 公司出品的 AVR 系列 ATmega128 單晶片。ATmega128 能夠執行在 16MHz 的頻率下，對於“創意之星”這種羽量級的自動控制系統而言有足夠的資料處理能力。AVR 系列單晶片是當前最爲流行的單晶片種類之一，介面豐富、開發簡單是它最顯著的特點（可參閱“AVR 單晶片”：http://baike.baidu.com/view/942508.htm）。

MultiFLEX™2-PXA270 控制器的處理器是 Marvell Xscale PXA270，Xscale PXA270 處理器可工作在 533MHz 的主頻下，加入了 Wireless MMX、Intel Speed Step 等新技術，以其高性能、低功耗、多功能等特點在資訊家電、工業控制等領域得到了廣泛的應用。MultiFLEX™2-PXA270 控制器執行 Linux 作業系統，Linux 以其內核精煉、高效、原始碼開放且免費等優勢，嵌入式領域得到了廣泛的應用。MultiFLEX™2-PXA270 控制器和 MultiFLEX™2-AVR 控制器相比，除了能夠處理 IO、AD 和匯流排資料之外，還能夠處理語音、視頻，我們據此可以開發出功能更加豐富的機器人結構。

2.2.1 MultiFLEX™2-AVR 控制器

MultiFLEX™2-AVR 控制器是一款小型機器人通用控制器。MultiFLEX™2-AVR 控制器功能高度整合，具有眾多 IO、AD 介面，能夠控制 R/C 伺服馬達、機器人伺服馬達，具有 RS-232 介面和 RS-422 匯流排介面，能夠勝任常規機器人控制；MultiFLEX™2-AVR 控制器開發簡單，使用圖形化整合開發環境，只需編寫程式邏輯流程就能夠自動產生 C 語言，下載到控制器後就可實現機器人的各種功能控制。開放所有底層函數介面，如果您熟悉嵌入式 C 語言開發，您可以直接調用這些函數，從而專注於機器人上層演算法的研究和編寫（詳見附錄）。

“成吉思”六足機器人



圖2.9 “成吉思”

1988年MIT人工智慧實驗室設計了一個叫“成吉思”的六足機器人，看到左邊的圖片您可能會覺得它好簡陋啊，但是您可能沒想到它在機器人研究史上佔有重要的地位，在很多機器人文獻上都會引用成吉思的研究成果。在機器人研究史上，始終有兩個旗幟鮮明的流派存在。以史丹佛人工智慧實驗室爲代表的研究人員強調機器人透過強大的電腦、感知器等手段，在機器人內部建立機器人外部的客觀世界模型，讓機器人可以“縱觀全局”並且計算出最優化的行爲。這樣的思路和現在人類的行爲模式非常相似，但是機器人要構建外部模型需要計算非常龐大的資料，在1970年史丹佛人工智慧實驗室利用當時的巨型電腦作爲機器人的大腦，能夠讓機器人在1個小時裏移動不到半米。直到2004年NASA設計的最先進的機器人“勇氣”號火星車，在火星地表每秒只能走5釐米，每天只能行走100米。

MIT人工智慧實驗室主張機器人應該沿著生物進化的道路往前發展。成吉思就是典型的階段產物，它是一隻類似蟑螂的機器人。成吉思能察覺紅外線信號源，這是它捕食的物件，它不用知道自己生存的大片環境，只需要知道眼前幾公分距離的障礙物大致位置，只需知道獵物的大致方位，根據簡單的狀態機，它就能堅定不移的往獵物方向移動，快速地將獵物捕獲。

使用 MultiFLEX™2-AVR 控制器和“創意之星”的零件，您可以很容易的搭建出“成吉思”，用於驗證前輩們所提出的理論，說不定您會有新的發現呢。

2.2.1.1 功能概述

對照圖示意圖 2.10，MultiFLEX™2-AVR 控制器功能如下所示：

- ATmega128@16MHz
- 6 個機器人伺服馬達介面，完全相容 Robotis Dynamixel AX12+
- 8 個 R/C 伺服馬達介面
- 12 個 TTL 電位的雙向 I/O 介面，GND/SIG/VCC 三線制
- 8 個 AD 轉換器介面（0~5V）
- 2 個 RS-422 匯流排介面(可掛接 1-127 個 422 設備)

- 1 個無源蜂鳴器
- 透過 RS-232 與控制機通訊，可選配無線通訊模組
- 使用 USB 介面的 AVR-ISP 下載調試器

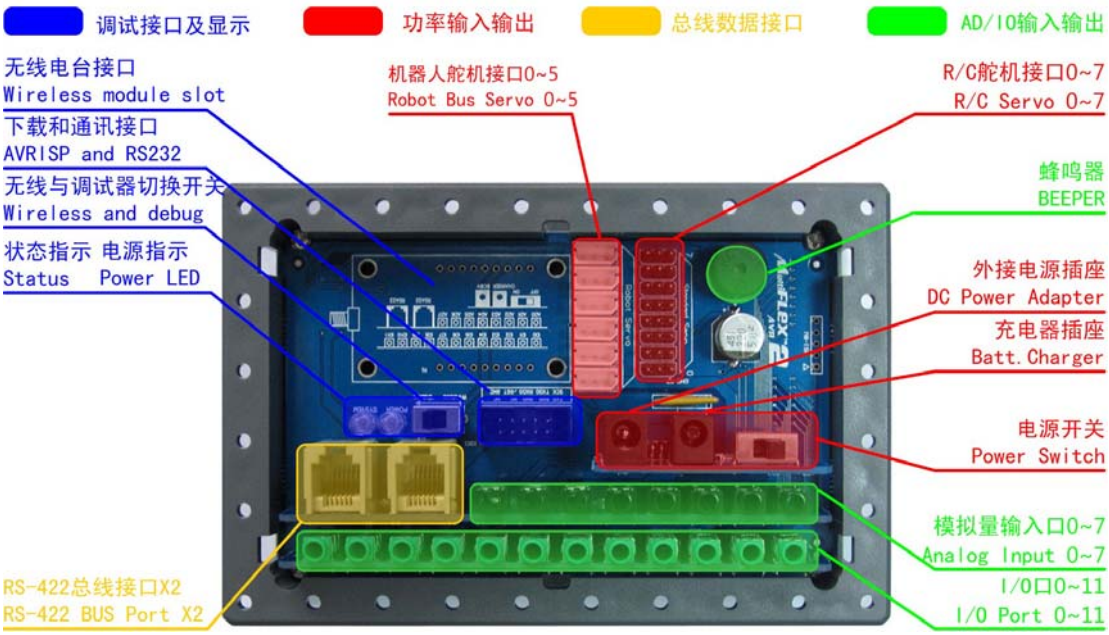


圖 2.10 MultiFLEX™2-AVR 控制器功能示意圖

2.2.1.2 外部介面及電氣規範

在“創意之星”使用過程中，您會頻繁操作 MultiFLEX™2-AVR 控制器，對其各種輸入輸出介面的電氣規範您需要心裏有數，錯誤的操作可能會導致危險。比如給充電介面接上 12V 的電壓可能會讓整個控制器冒煙。MultiFLEX™2-AVR 控制器的電氣規範如下表所示：

表 2.1 MultiFLEX™2-AVR 控制器的電氣規範

項目	數據	說明
電池電壓	6.5~8.4VDC	使用 2 節鋰聚合物電池，標準電壓 7.2V，使用過程電壓範圍爲 6.5V~8.4V。
充電電壓	——	智慧充電器充電，充電過程會自動調節電壓，典型值爲 8V~9V。
外接電源	8V	外接直流穩壓電源，電壓 8V，正常使用電流 0~5A。
保護	反接保護 過流保護	警告 長時間電源反接仍可能損壞控制器。 過流保護生效後，需要重新上電才能工作。
靜態功耗	0.5W	無外接設備下的靜態功耗。
保護電流	6~8A	超過此電流後，自動切斷。約 10 秒後才能再次工作。
I/O 電位		低電位 < GND + 1.5V

		高電位 > VCC – 1.5V
數位通訊介面	RS-232 介面	TX，RX，GND 三線制
數位輸入/輸出	12 個	GND/VCC/SIG 三線制（SIG 可以配置爲輸入或者輸出，在 NorthSTAR 軟體中配置或透過協定配置）。
類比輸入	8 個	GND/VCC/SIG 三線制，10 位精度。
機器人伺服馬達介面	6 個	1M 速率的半雙工非同步串列匯流排，理論可接 255 個機器人伺服馬達，由於供電能力限制，  建議同時使用時不超過 30 個。伺服馬達工作電壓等於控制器工作電壓。
R/C 伺服馬達介面	8 個	GND/VCC/SIG 三線制（SIG 爲信號輸出），工作電壓 5V。
ISP 功能	支援	GND/RST/MOSI/MISO/SCK 五線，配套提供 ISP 編程電纜。
無線通訊	支援	可選配 ZigBee 無線通訊模組，57.6KBps。

2.2.2 PXA270 MultiFLEX™2-PXA270 控制器

在機器人小說裏，最出名的莫過於《我，機器人》，作者阿西莫夫在書中提出了機器人三法則，這三法則在提出之後的幾十年時間裏被重複引用了無數次。

1. **機器人不應傷害人類，或坐視人類受到傷害而袖手旁觀；**
2. **機器人應遵守人類的命令，與第一條違背的命令除外；**
3. **機器人應能保護自己，與第一條相抵觸者除外。**

機器人小說和電影裏主要的故事情節一般都和三法則有關。機器人在遵守三法則的同時卻遇到了與邏輯判斷、使命相互矛盾的情況，機器人由此陷入了迷茫和掙紮之中，它們會努力去在這些矛盾中尋找平衡點，卻做出了很多出乎人們意料的事情來。

事實上，這樣的事情估計在未來 20 年內都很難出現，阿西莫夫在提出的這三法則看似平實可行甚至理所當然，但是只要您瞭解機器人技術，您會發現三法則給機器人負擔了多麼沉重的感知負荷。

到 2000 年爲止，所有的機器人都不能遵守機器人三法則，因爲機器人根本不能感知到人類的存在。在機器人“眼裏”，人類就是一個柱狀的障礙物，和木桶、箱子等沒有任何差別，更遑論讓它判斷當前的行爲是不是會傷害到人類，或者視野裏的人類是不是受到威脅呢？電腦技術的發展，讓機器人在處理視覺和語言上面有了長足的進度，如果機器人能夠識別人臉，能夠聽懂人的語言，那麼機器人就能初步將人從其他“障礙物”中區分出來，畢竟語言是人類區別其他任何生物與非生物的最重要條件，人臉是區別人與人的最重要手段。

“創意之星”的 MultiFLEX™2-PXA270 控制器具備實現上述兩點的硬體條件，當然，您需要寫點程式碼。

2.2.2.1 PXA270 功能概述

MultiFLEX™2-PXA270 控制器純粹就是爲智慧型機器人控制而存在的。它有以下特點：

1、 **高運算能力、低功耗、體積小。**MultiFLEX™2-PXA270 控制器具備 533MHz、32 位元的高性能嵌入式處理器和 Linux 作業系統，運算處理能力強大，而功耗不到 2W；體積小巧，可以直接放入仿人機器人體內。

上述特點有什麼用處呢？使用這個控制器可以讓您的機器人有眼睛（視覺），並且能夠把看到的圖像以 30 幀/s 的速度進行識別處理；這個控制器可以讓您的機器人能聽得懂您的語言；不到 2W 的靜態功耗，讓您可以不用半個小時就換塊電池；體型輕巧，可以用在小尺寸的機器人機體上。

2、 **控制介面豐富。**可以控制直流馬達（須配合 BDMC 系列伺服驅動器）；可以控制各種信號的伺服馬達（包括所有的傳統 R/C 伺服馬達、CDS55XX 系列機器人伺服馬達、韓國 Robotis 公司出品的 AX12+機器人伺服馬達等）；可以對機器人伺服馬達進行調速、位置控制、力矩控制；可以同時控制接近 40 路伺服馬達\馬達。

MultiFLEX™2-PXA270 控制器充分放開了對機器人執行機構的限制，您在設計機器人的時候只需要考慮使用什麼方案合適，而不用顧忌控制器能不能支援。您也不用耗費心力研究怎麼控制馬達和伺服馬達，MultiFLEX™2-PXA270 控制器上提供了您能使用到的所有函數介面，您直接調用即可。

3、 **資料介面豐富。**控制器具有 12 路雙向可設置通用 IO 介面，8 路 10 位元精度的 AD 介面；還有 RS-422 匯流排、RS-232 介面，足夠滿足絕大多數運用場合。

感知的豐富性是機器人區別於其他機器的重要因素之一，要讓您的機器人具有與眾不同的功能，對各種外界刺激有合理的反應，您需要給它添加各種各樣的感測器。溫度、光照、聲強、距離等感測器您可以透過 IO 和 AD 介面輸入，姿態、語音、視覺感測器可以透過 RS-422 匯流排、RS-232 介面、USB 介面輸入。

4、 **開發簡單。**配套 NorthSTAR 圖形化整合開發環境，您不用面對令人頭疼的 Linux 系統編程，嵌入式高手基本上都是高薪人士，原因是嵌入式編程難度高、人才少。您只是爲了設計一隻比較聽話的小狗，不用非得抱著厚厚的“磚頭”啃上半年到一年之後再開始動手設計，相信到時您對設計會說話的小狗的激情與創意都已灰飛煙滅了。

MultiFLEX™2-PXA270 控制器的詳細功能如下所示：

- MultiFLEX™2-PXA270 控制器：
  - 完全支持 MultiFLEX™2-AVR 控制器的全部功能
  - Marvell Xscale PXA270@533MHz，32 位元處理器，16M NOR-FLASH，128M NAND-FLASH ，64M SDRAM
  - Linux 作業系統
  - 4 個 USB Host，1 個 100bps 乙太網路埠、Wifi 模組插槽（Wifi 模組屬選購配件）
  - 1 個麥克風介面，1 個身歷聲音頻輸出介面
  - 支援攝影機作爲視覺感測器，麥克風作爲聽覺感測器

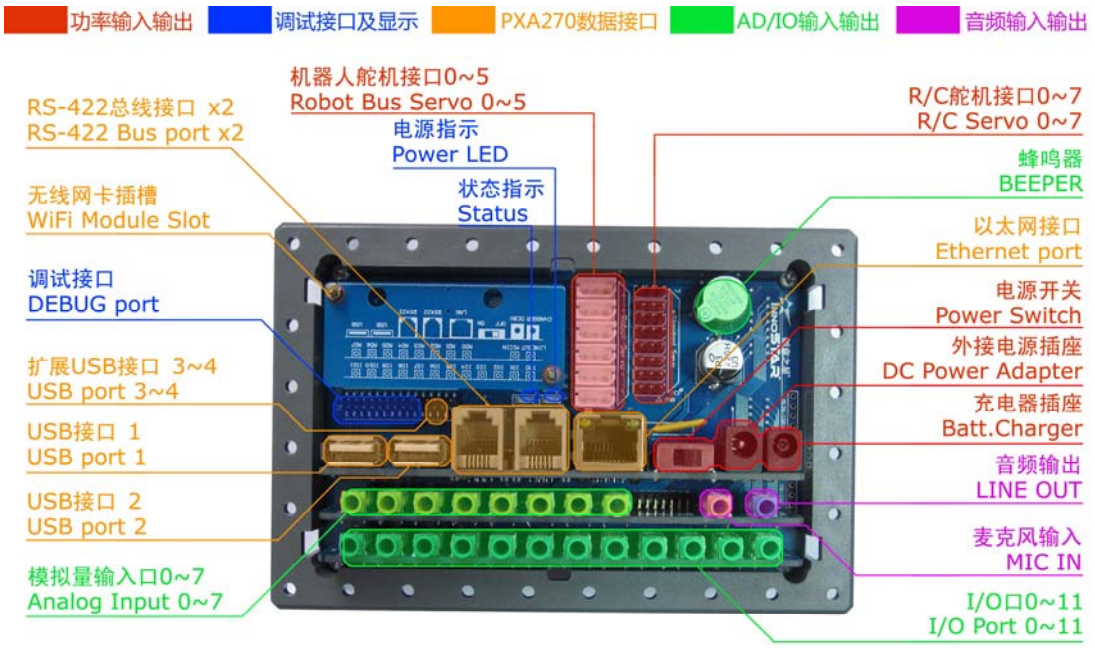


圖 2.11 MultiFLEX™2-PXA270 控制器

2.2.2.2 PXA270 外部介面及電氣規範

如圖 2.11 所示，MultiFLEX™2-PXA270 控制器相比 MultiFLEX™2-AVR 控制器多出了乙太網路介面、USB 介面、音頻介面。MultiFLEX™2-PXA270 控制器獨有介面電氣規範如下表所示：

表 2.2 MultiFLEX™2-PXA270 控制器的電氣規範

項目	數據	說明
電池電壓	6.5~8.4VDC	使用 2 節鋰聚合物電池，標準電壓 7.2V，使用過程電壓範圍爲 6.5V~8.4V。
充電電壓	——	智慧充電器充電，充電過程會自動調節電壓，典型值爲 8~9V。
外接電源	8V	外接直流穩壓電源，電壓 8V，正常使用電流 0~5A。
保護	反接保護 過流保護	警告 長時間電源反接仍可能損壞控制器。 過流保護生效後，需要重新上電才能工作。
靜態功耗	1.5W	無外接設備下的靜態功耗。
保護電流	6~8A	超過此電流後，自動切斷。約 10 秒後才能再次工作。
I/O 電位		低電位 < GND + 1.5V 高電位 > VCC – 1.5V
數位通訊介面	乙太網路	
數位輸入/輸出	12 個	GND/VCC/SIG 三線制（SIG 可以設爲輸入或者輸

		出，在 NorthSTAR 軟體中配置或透過協定配置）。
類比輸入	8 個	GND/VCC/SIG 三線制，10 位元精度
機器人伺服馬達介面	6 個	1M 速率的半雙工非同步串列匯流排，理論可接255 個機器人伺服馬達，由於供電能力限制，建議同時使用時不超過 30 個。伺服馬達工作電壓等於控制器工作電壓。
R/C 伺服馬達介面	8 個	GND/VCC/SIG 三線制（SIG 爲信號輸出），工作電壓 5V。
USB 介面	4 個 USB2.0	在介面板上有兩個對外 USB 介面，一個 USB 介面固定爲無線網卡專用埠，一個 USB 介面固定爲調試介面
乙太網介面	1 個	100M 自適應乙太網路介面
音頻介面	1 輸入 1 輸出	
無線通訊	支援	54M 無線乙太網路（選配零件）

2.2.3 電池和直流電源

MultiFLEX™2-PXA270 控制器和 MultiFLEX™2-AVR 控制器內置一組 7.2V 2.5Ah 的鋰聚合物電池，一般情況下使用電池供電。電池電量不足的情況下，您可以使用直流穩壓電源給控制器供電。

電池的種類很多，比如鎳氫電池、鉛酸電池、鋰離子電池、鋰聚合物電池等等。鋰聚合物電池具有重量輕、容量大、無記憶效應等優點，但對於日常維護要求較高，在使用中需要注意下面的注意事項。

- **電池**

如圖 2.12 所示，“創意之星”使用 7.2V 2.5Ah 的鋰聚合物電池，內置 8A 放電能力的充放電保護板。在電池的使用過程，您需要注意以下幾點：

1、 電池輸出電壓會隨著電池電量的下降而下降，充滿電時電池電壓約爲 8.4V，使用過程中可能會降低到 6.5V。

2、 電池電壓過低時（低於 6.5V），控制器電源指示燈會以 2s 的週期閃爍，蜂鳴器會發出短促的鳴叫。這時請您切斷電源，給電池充電，繼續使用會使電池過度放電，造成永久性損傷。您也可以選擇使用外接直流穩壓電源爲控制器供電。

3、 請使用“創意之星”原裝充電器給電池充電，非標準充電器可能會充不滿電量或者充壞電池。

4、 電池已經內置到控制器裏面，請不要自行拆卸。如果您需要更大容量的電池，請在確認電池符合要求（額定電壓 6.5~8.4V 之間）後，將電池直接連接到用於連接外接直流穩壓電源的插座上。

5、 注意：充電介面和直流穩壓電源介面是分開的，並且接頭不一樣，如果

不能順利地將插頭插入插座時請勿過度用力，先檢查介面是否正確。



圖 2.12 鋰聚合物電池

- **直流穩壓電源**

如圖 2.13 所示，“創意之星”配置了一個直流穩壓電源。電源的輸出爲 8V/5A，峰值電流可達 8A。直流穩壓電源使用過程中，您需要注意以下情況：

1、 電源的正常輸出能力是 8V/5A，峰值可達到 8A，但是長時間工作在 5A 以上會導致電源過熱，縮短電源使用壽命。

2、 您如果使用自己的外置電源時，請保證電源輸出電壓範圍是在 6.5V~10V 之間，低於 6.5V 時控制器會出現欠壓警報，高於 10.5V 控制器會出現過壓警報。

3、 欠壓警報症狀是蜂鳴器短促鳴叫，控制器電源指示燈以 2s 的時間間隔閃爍。過壓警報症狀是蜂鳴器短促鳴叫，控制器電源指示燈以 200ms 的時間間隔閃爍。



圖 2.13 直流穩壓電源

2.3 感測器

“創意之星”配置了大量的感測器，而什麼是感測器？在不同的運用場合如何選用合適的感測器呢？如果您沒有接觸過這些類型的機器人零件，您會很茫然。接下去我們將對常用的機器人感測器進行介紹。

結構零件構成了“創意之星”機器人的軀體，控制器是“創意之星”機器人的大腦，而

感測器就是“創意之星”機器人的感知器官：皮膚、眼睛、耳朵、鼻子和舌頭等等。

感測器能夠將機器人自身的電流、電壓、位置資訊，外部的距離、溫度、濕度、光線、聲音、圖像等資訊轉化爲電信號。感測器內部的電路對這些電信號經過預處理之後轉化爲能夠爲控制器所採集或讀取的電壓、脈衝、資料信號。感測器據此得到需要的內外部資訊。

控制器對這些信號解析提取之後，就能夠知道相應的資訊，比如機器人離前面的牆壁只有 10cm 了。您編寫程式的時候可以對此距離的不同數值採取不同的策略。比如機器人離牆壁 20cm 的時候減速，距離 10cm 的時候轉彎。

我們可以看到，“創意之星”配置了很多種類的感測器。在機器人領域還有其他種類、型號的感測器在使用。本小節不能做到面面俱到，您如果希望瞭解更多知識的話，您可以使用“Google”搜尋也是很好的選擇。

2.3.1 感測器的信號類型及電氣規範

首先我們需要先理解幾個概念：信號、數位信號、類比信號、開關信號、匯流排、通信協定。

- **信號**

信號是隨時間變化的物理量（光、電、文字、符號、圖像、資料等），可以認爲它是一種傳載資訊的函數。人們透過一定的手段獲取信號，透過適當的資訊分析和處理，才能取得所需的資訊。例如，我們用 MultiFLEX™2-AVR 控制器採集溫度感測器的輸出信號，透過解析之後就能夠知道當前溫度值，溫度值就是我們所需要提取的資訊。

根據信號的參數（多指時間或空間等參數）和信號函數的取值不同，可以分爲連續時間信號和離散時間信號兩類。根據函數取值是否連續可分爲類比信號、量化信號、抽樣信號、數位信號等。

- **類比信號**

在時間和數值上均是連續的物理量稱爲類比量，如電壓、電流、聲音、速度、壓力、溫度等，我們將表示這些物理量的信號稱爲類比信號，將工作在類比信號下的電子電路稱爲類比電路，同理將輸出信號爲類比信號的感測器稱爲類比量感測器。

- **數位信號**

有些物理量在時間和數值上均是離散的，它們的變化在時間上是不連續的，總是發生在一系列離散的瞬間。同時，它們的數值大小和每次增減變化都是某個最小單位的整數倍，小於這個最小數量單位的任何值都是沒有意義的。這類物理量稱爲數位量，其信號稱爲數位信號，工作在數位信號下的電子電路稱爲數位電路。

- **開關信號**

電腦的資料結構是由 0 和 1 構成的，所以 0 和 1 兩種狀態的資料是最容易處理的。數位信號裏，最簡單的信號就是只有高電位和低電位構成的 0 和 1。只有 0 和 1 的數位信號稱爲開關信號，同理只能輸出高電位或低電位的感測器電路稱爲開關量感測器。

- **資料匯流排**

所謂匯流排，就是指能爲多個功能零件服務的一組資訊傳輸線，它是電腦中系統與系統之間、或者各零件之間進行資訊傳送的公共通路。如 RS-422 匯流排、CAN 匯流排、IIC 匯流排、USB 匯流排等等。

- **RS-422 匯流排**

RS-422 匯流排只有四根信號線，其介面採用平衡驅動差動接收電路。因爲採用差動接收方法，起作用的是兩個輸入端的電位差，所以信號的電位幅度不必抬高，用 TTL 電位即可有很強的抗干擾能力，直接相連時距離可達 200 m。

這種介面電路形式還使得一個驅動器能同時接多個接收器，因此可利用 RS-422 實現多台 422 設備的互連，構成主、從式通信網路。主機（指在整個網路系統中起主導作用的機器）的 RS-422 發送端與所有從機（指在整個網路系統中處於從屬地位的機器或設備）的 RS-422 接收端相連；所有從機的 RS-422 的發送端連在一起，接到主機的 RS-422 接收端。這樣連接使得主機發送時從機都可以收到，而從機都可以向主機發送資訊。爲避免兩個或多個從機同時發送資訊而引起衝突，通常採用主機呼叫、從機應答方式，即只有被主機呼叫的從機（每一台從機都有自己的位址）才能發送資訊。

- **通信協議**

協議（protocol）一詞常用來表示“外交禮儀”、“條約”等。由此可見，通信協議也就是爲進行資料通信而事先約定的章程。

通信協定由表示資訊結構的格式和資訊交換的進程組成。格式規定了資料爲何種類型、如何排列，至於進程則規定了資料以怎樣的步驟和流向來實現資訊交流。資料通信時，爲了保證雙方能夠正確的收發資訊，應遵循相同的通信協議。我們最常見的通信協定時 TCP/IP 協定。

以上內容是類比電路、數位電路、電腦技術的基礎內容，更詳細的資料還請查閱相關教材。

“創意之星”配套的感測器有類比量感測器、數位量感測器、匯流排式感測器三種。具體如下表所示：

表 2.3 “創意之星”感測器類型及電氣規範

項目	資料類型	電氣規範
紅外接近感測器	開關量	低電位 0V 高電位 5V
碰撞感測器	開關量	同上
聲音感測器	開關量	同上
霍爾感測器	開關量	同上
姿態感測器	開關量	同上
光強感測器	類比量	0~5V
溫度感測器	類比量	同上
灰度感測器	類比量	同上
紅外測距感測器	類比量	同上

2.3.2 “創意之星”感測器介面

“創意之星”類比量和開關量感測器的介面是 3.5mm 耳機插頭，控制器上的 IO 介面和 AD 介面是 3.5mm 耳機插座。

如圖 2.14 所示，控制器上有兩排不同顏色的耳機插座。①第一排立板 12 個綠色的耳機插座爲控制器的 IO 輸入\輸出介面，可以接入開關量感測器。②第二排立板 8 個黃色耳機插座爲控制器的 AD 輸入介面，可以接入類比量感測器。③是控制器的 RS-422 匯流排介面。需要注意的是 MultiFLEX™2-PXA270 控制器在 8 個 AD 介面旁邊還有一黃一綠兩個耳機介面，這兩個介面是音頻輸入和輸出介面，不能和 AD 介面混淆。

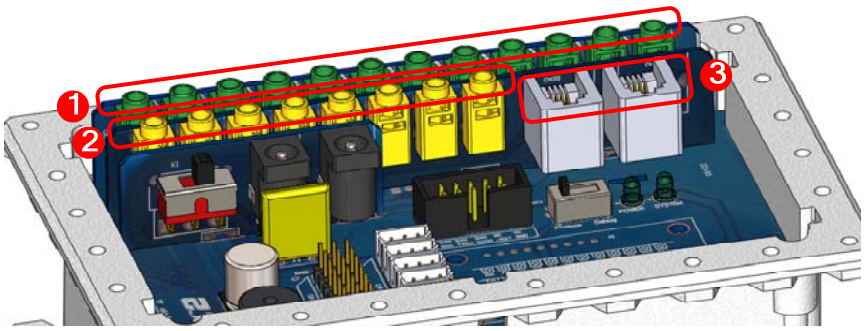


圖 2.14 “創意之星”控制器感測器介面

如圖 2.15 所示，感測器的 3.5mm 耳機接頭線序爲：VCC\SIG\GND（+5V\信號\地）。

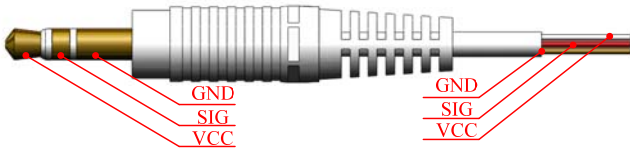


圖 2.15 3.5mm 耳機插頭

圖 2.16 爲 RS-422 匯流排使用的 6 針 RJ11 線材，線序如表格 2.4 所示。



圖 2.16 RS-422 使用的線材

表 2.4 RS-422 線序

水晶頭線序	定義	顏色

1、	VCC	白
2、	GND	黑
3、	422_A	紅
4、	422_B	綠
5、	422_Y	黃
6、	422_Z	藍

2.3.1 接近感測器

您最不希望看到的莫過於費盡苦心設計的機器人“啪”的一聲從桌上掉下來，摔了個稀爛。如何避免這種情況的出現呢？您可以在桌子上圍上一圈柵欄，不過估計老師或者家長不會同意這種做法。您在設計機器人的時候會遇到很多類似的機器人行爲或策略問題，這時候您可以回想一下人和動物是怎麼解決這些問題的，人類和動物進化了億萬年，在適應環境方面已經有了完美的解決方案，我們從中可以得到有效的借鑒。

人們透過眼睛看到臺階，並估計臺階的距離和高度。如果讓機器人用攝影機看到臺階，將之和其他物體區分開，並估算出距離，對機器人而言這還是一個世界難題，我們不打算鑽這個牛角尖。我們可以用接近感測器來判斷是不是到達臺階或桌子邊緣。紅外感測器和霍爾感測器是常見的兩種接近感測器。

2.3.1.1 紅外接近感測器

- **原理**

紅外接近感測器俗稱光電開關。它是利用被檢測物對光束的遮擋或反射，由同步回路選通電路，從而檢測物體有無的。光電開關將輸入電流在發射器上轉換爲光信號射出，接收器再根據接收到的光線的強弱或有無對目標物體進行探測。工作原理如下圖所示。多數光電開關選用的是波長接近可見光的紅外線光波型，因此也稱爲紅外開關。

由於紅外線是不可見光，紅外探頭體積小巧，隱蔽性非常高，所以各種規格的紅外開關、紅外測距感測器常常用於安防保衛領域。在好萊塢電影裏，我們常常看見金庫、博物館裏有一條條紅色光線，大盜們運用各種手段避開這些探測光線，最終盜得各種財寶。實際上，紅外線是不可見的，電影裏採用了普通可視光線冒充紅外線，以烘托緊張的故事情節，我們對此大可一笑而過，千萬不要受到誤導。

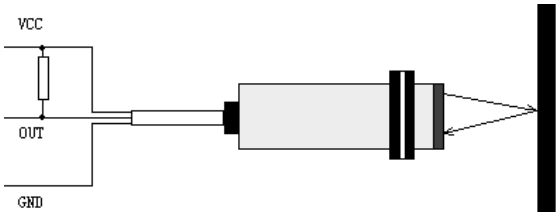


圖 2.17 紅外接近感測器工作原理

光電開關可以分類如下：

1. 漫反射式光電開關：它是一種集發射器和接收器於一體的感測器，當有被檢測物體經過時，物體將光電開關發射器發射的足夠量的光線反射到接收器，於是光電開關就產生了開關信號。當被檢測物體的表面光亮或其反光率極高時，漫反射式的光電開關是首選的檢測模式。

2. 鏡反射式光電開關：它集發射器與接收器於一體，光電開關發射器發出的光線經過反射鏡反射回接收器，當被檢測物體經過且完全阻斷光線時，光電開關就產生了檢測開關信號。

3. 對射式光電開關：它包含了在結構上相互分離且光軸相對放置的發射器和接收器，發射器發出的光線直接進入接收器，當被檢測物體經過發射器和接收器之間且阻斷光線時，光電開關就產生了開關信號。當檢測物體爲不透明時，對射式光電開關是最可靠的檢測裝置。

4. 槽式光電開關：它通常採用標準的 U 字型結構，其發射器和接收器分別位於 U 型槽的兩邊，並形成一光軸，當被檢測物體經過 U 型槽且阻斷光軸時，光電開關就產生了開關量信號。槽式光電開關比較適合檢測高速運動的物體，並且它能分辨透明與半透明物體，使用安全可靠。

5. 光纖式光電開關：它採用塑膠或玻璃光纖感測器來引導光線，可以對距離遠的被檢測物體進行檢測。通常光纖感測器分爲對射式和漫反射式。

“創意之星”機器人套件配套了 2 個漫反射式光電開關，其有效距離約爲 20cm。

提示：漫反射式光電開關發出的光線需要經檢測物表面才能反射回漫反射開關的接受器，所以檢測距離和被檢測物體的表面反射率將決定接受器接收到光線的強度。粗糙的表面反射回的光線強度必將小於光滑表面反射回的強度，材料的反射率是影響光電開關有效距離的重要參數。常用材料的反射率如下表所示。

表 2.5 常用材料的反射率

材料	反射率	材料	反射率
白畫紙	90%	不透明黑色塑膠	14%
報紙	55%	黑色橡膠	4%
餐巾紙	47%	黑色布料	3%
包裝箱硬紙板	68%	未拋光白色金屬表面	130%
潔淨松木	70%	光澤淺色金屬表面	150%
乾淨粗木板	20%	不銹鋼	200%

透明塑膠杯	40%	木塞	35%
半透明塑膠瓶	62%	啤酒泡沫	70%
不透明白色塑膠	87%	人手掌心	75%

紅外開關是紅外線光電開關的簡稱，利用被檢測物體對紅外光束的遮光或反射，由同步回路選通而檢測物體的有無，其物體不限於金屬，對所有能反射光線的物體均可檢測。現有的光電感測器優先使用的是波長 780nm——3um 的近紅外光，並已有比較穩定的整合化產品，與數位電路的介面也非常簡單。

- **使用**

如圖 2.18 “創意之星”機器人所使用的紅外光電開關型號爲 E18-B0，規格資料爲：

VCC：5V。
工作電流：小於 100mA。
輸出形式：NPN 三極管 OC 輸出。
封裝形式：專案塑料。



圖 2.18 紅外接近感測器

紅外接近感測器的是開關量感測器，接 IO0~IO11 的任意一個介面都可以透過 NorthSTAR 進行數值讀取和編程。由於輸出是開關量，只能判斷在測量距離內有無障礙物，不能給出障礙的實際距離。但是該感測器帶有一個靈敏度調節旋鈕，可以調節感測觸發的距離。“創意之星”套件在出廠前已經將感應觸發距離調整到約 20cm。

紅外接近感測器是機器人上最常用的感測器之一，用於躲避周圍障礙，或者在無需接觸的情況下檢測各種物體的存在。它的用途非常多，第四章將利用紅外接近感測器設計一個防碰撞小車，您可以根據這一章的指導，實際使用一下紅外接近感測器。

2.3.1.2 霍爾接近感測器

- **原理**

霍爾效應接近感測器是利用霍爾效應（Hall Effect）製成的接近開關，主要用於檢測磁性物體。市場上常見的霍爾接近感測器的檢測距離爲 10mm 左右。

以下是一些常見的接近開關的外形特點和安裝方式。可以看到，接近開關種類非常豐富，安裝方式也很多樣。不管是電感式、電容式還是霍爾效應式，其外觀都有些類似，介面也基本上都是三線制：信號輸出（通常爲 OC 輸出）、電源（通常 5-30V）、地。

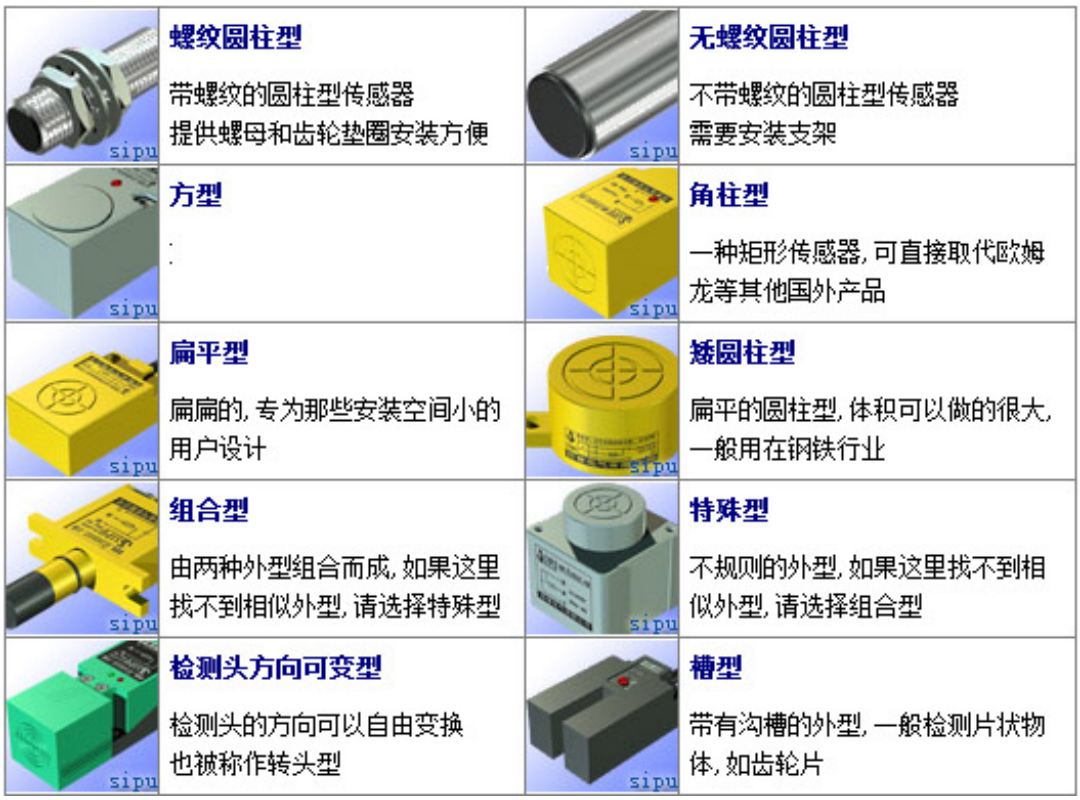


圖 2.19 接近開關的種類

- 使用

霍爾接近感測器是開關量感測器，接 IO0~IO11 的任意一個介面都可以透過 NorthSTAR 進行狀態讀取和編程，使用方法和紅外接近感測器一致。

霍爾接近感測器對磁場或磁性物體很敏感，工廠據此鋪設自主導引車（AGV）的行車軌跡，在水泥地面上鑲嵌磁化的鐵條，自主導引車可以檢測預鋪設的鐵條，確定行車方向。

下圖是“創意之星”使用的霍爾接近感測器，前方突起是感測器的探頭：

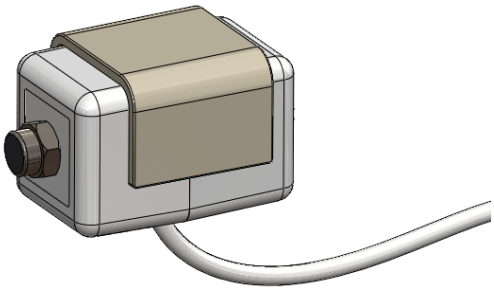


圖 2.20 霍爾接近感測器

2.3.2 測距感測器

在很多場合，我們需要準確知道機器人和目標物的距離，這樣的要求是不能透過接近感測器來實現的，因爲接近感測器是開關量，只能告訴我們“有無”，不能告訴我們“多少”。在實際生活中，我們可以用尺來測量距離，機器人不可能隨時隨地伸著把尺，它用於測量的尺規是聲波、紅外線或者鐳射。在生物界裏，蝙蝠是用超音波進行測距和定位的高手，超音波測距感測器、雷達等設備就是根據蝙蝠使用超音波的原理進行設計的。下面我們會介紹“創意之星”使用的紅外測距感測器和超音波測距感測器。雷射雷達比較昂貴，在“創意之星”套件裏面沒有這樣的零件。

- **原理**

日本 SHARP 公司推出了一系列的紅外測距感測器（Infrared Range Finder），用來測量前方物體和感測器探頭之間的距離。這些感測器體積小（手指大小）、重量輕（不到 10 克重），介面簡單，用於微型機器人的測距是非常理想的選擇。

GP2D12 是該系列感測器中的典型。它的輸出爲：0~2.5V 類比量（電壓值隨距離變化）；量程範圍 10~80 cm。這個型號的感測器作爲大多數微型移動機器人的防碰撞和漫遊測距用感測器都是足夠的。另外還可以用於檢測機器人各關節位置、姿態等。

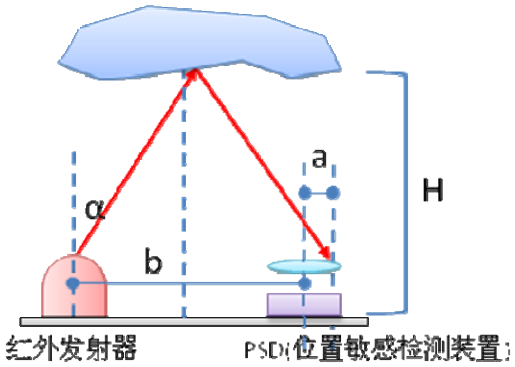


圖 2.21 GP2D12 及其原理

GP2D12 主要是由紅外發射器、PSD（位置敏感檢測裝置）及相關處理電路構成，紅外發射器發射一束紅外光線，紅外光線遇到障礙物被反射回來，透過透鏡投射到 PSD 上，投射點和 PSD 的中心位置存在偏差值 a，GP2D12 根據上圖所示的 a、b、α 三個值就可以計算出 H 的值，並輸出相應電位的類比電壓。

由上述原理可知紅外測距感測器的幾個重要的特性：

1、與障礙物的反射角度基本無關。下圖列出了 GP2D12 在以不同角度面對反射面時，實際距離與測得距離的偏差。實際距離均爲 40cm，反射物爲一塊 40cmx40cm 的白色木板。可以看到在反射物垂直於光路、20 度、40 度、60 度夾角時，輸出值誤差很小。

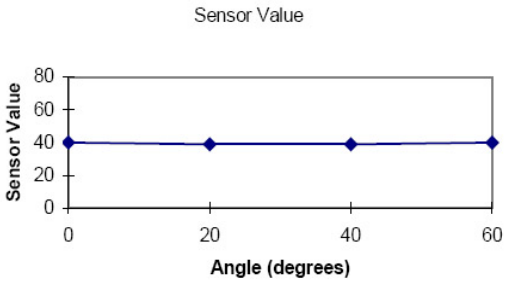


圖 2.22 GP2D12 測距結果與障礙物夾角的關係

2、與反射物體的顏色及材質基本無關。下圖列出了與感測器距離爲 50cm 的棕色卡片紙、藍色塑膠、黑色皮革、白紙四種不同材質的感測器輸出值。可以看到，該種感測器對反射物的材質並不敏感，實際輸出並不隨材質而變化。但是，有效測量距離是隨被測物體材質而不同的，例如對於白紙，最大有效測量距離可達到 80cm；但是對於黑色皮革，有效測量距離可能只能達到 60-70cm。這是由於不同材質的反射率不同所致。

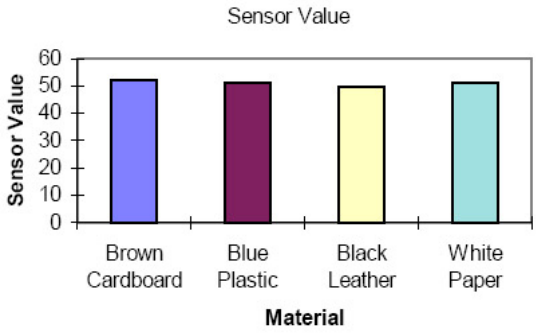


圖 2.23 GP2D12 測距結果與障礙物顏色的關係

下圖列出了不同距離下，採用一個 16 位元 A/D 轉換器對感測器的輸出信號進行 A/D 轉換後的結果。注意這種感測器的輸出不是線性的，也就是說，輸出值與實際反射物距離並非成反比或正比關係，在使用的時候，要對感測器的這一特性進行標定，多測量一些資料，並採用查表的方式來得到輸出資料與實際距離的對應關係。

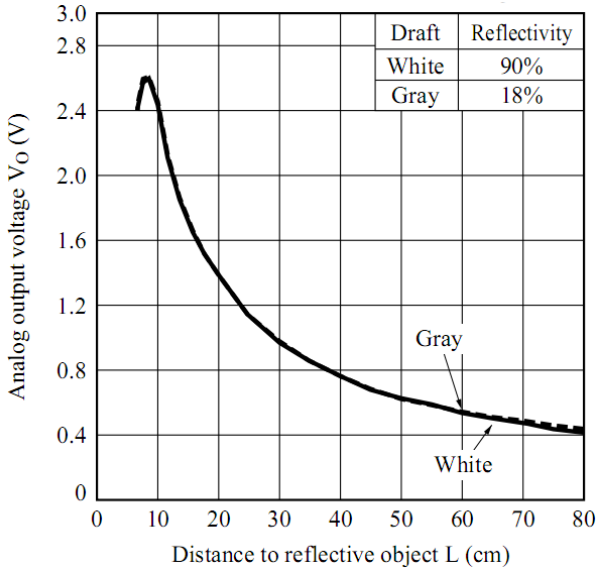


圖 2.24 GP2D12 測距結果與障礙物距離的關係

下表列出了 SHARP 出品的該系列感測器中，不同型號感測器的性能參數對比。

型号	输出形式	最小感测距离	最大感测距离	工作电流	待机电流
GP2D02	Serial	10cm	80cm	~22mA	~3mA
GP2D05	Digital	-	fixed at 24cm	~10mA	~3mA
GP2D12	Analog	10cm	80cm	~33mA	always on
GP2D15	Digital	-	fixed at 24cm±3cm	~33mA	always on
GP2D120	Analog	4cm	30cm	~33mA	always on
GP2Y0A02YK	Analog	20cm	150cm	~33mA	always on
GP2Y0D02YK	Digital	-	fixed at 80cm	~33mA	always on

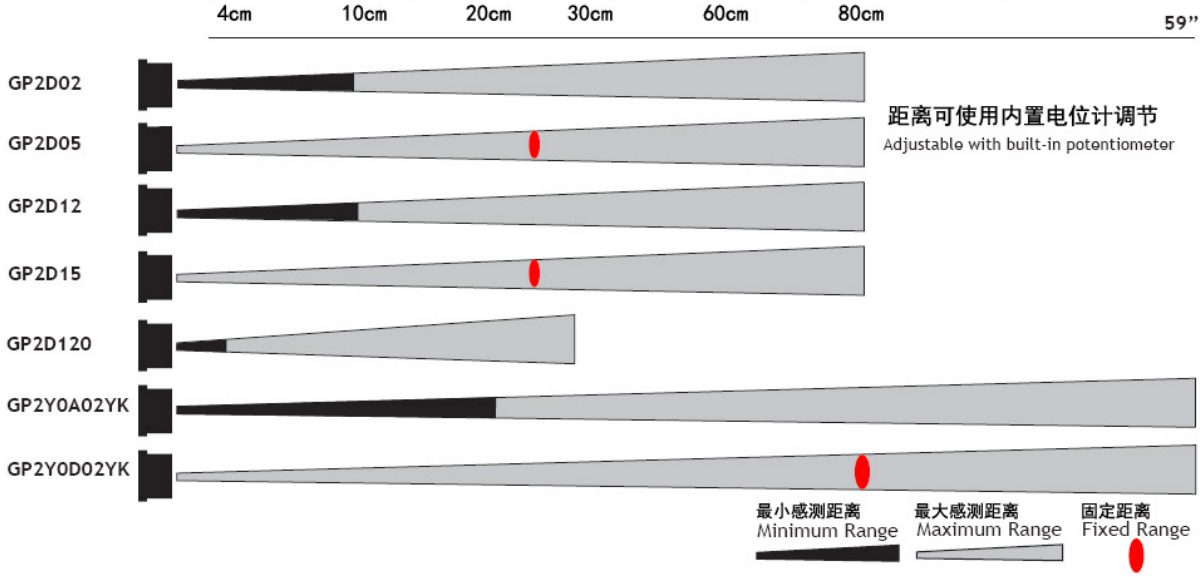


圖 2.25 SHARP 紅外測距系列產品

- **使用**

圖 2.26 爲“創意之星”使用的 GP2D12 紅外測距感測器，與常見 GP2D12 不同的是我們爲它設計了一個類似機器人頭部的外殼，可以方便地安裝到“創意之星”零件上。紅外測距

感測器爲類比量感測器，接 AD0~AD7 的任意一個介面都可以透過 NorthSTAR 進行數值讀取和編程。感測器的規格資料如下：

探測距離：10-80cm
工作電壓：4-5.5V
標準電流消耗：33-50mA
輸出量：類比量輸出，輸出電壓和探測距離非比例相關

MultiFLEX™2-AVR 控制器和 MultiFLEX™2-PXA270 控制器的 AD 精度爲 10 位元，測量電壓範圍 0~5V，對應輸出值 0~1023。您如果要得到真實的距離值需要做 2 次換算，假設您從 NorthSTAR 讀取的 AD 值爲 491，換算爲真實電壓值爲 5*(491/1023)=2.4V，從 2.24 的對照圖可以知道當前感測器探頭到障礙物的距離是 10cm。

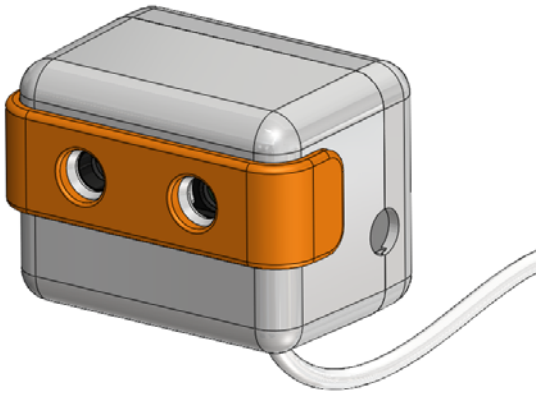


圖 2.26 紅外測距感測器

在第八章，我們使用紅外測距感測器測量機器人兩個懸臂與地面的距離，由此求得機器人的傾斜角度，根據這個角度控制機器人，讓兩輪小車也能平衡站立。

2.3.2.1 超音波測距感測器

- **原理**

超音波測距感測器工作原理是：探頭向前方發射一束超音波，超音波經前方障礙物反射返回，感測器再接收此反射波，透過計算聲波往返時間與音速相乘就可算出障礙物的距離。

由於超音波指向性強，能量消耗緩慢，在介質中傳播的距離較遠，因而超音波經常用於距離測量，如測距儀和物位測量儀等都可以透過超音波來實現。利用超音波檢測往往比較迅速、方便、計算簡單、易於做到即時控制，並且在測量精度方面能達到工業實用的要求，因此在移動機器人的研製上也得到了廣泛應用。

1、 超音波發生器

感測器需要有和蝙蝠一樣能夠產生超音波的零件，我們將之稱爲超音波發生器。一般而言，超音波發生器可以分爲兩大類：一類是用電氣方式產生超音波，一類是用機械方式產生超音波。電氣方式包括壓電型、磁致伸縮型和電動型等；機械方式有加爾統笛、液哨和氣流

旋笛等。它們所產生的超音波的頻率、功率和聲波特性各不相同，因而用途也各不相同。目前較爲常用的是壓電式超音波發生器。

2、壓電式超音波發生器原理

壓電式超音波發生器是利用壓電晶體的諧振效應來工作的。超音波發生器內部有兩個壓電晶片和一個共振板。當它的兩極外加脈衝信號，其頻率等於壓電晶片的固有振盪頻率時，壓電晶片將會發生共振，並帶動共振板振動，便產生超音波。反之，如果兩電極間未外加電壓，當共振板接收到超音波時，將壓迫壓電晶片作振動，將機械能轉換爲電信號，這時它就成爲超音波接收器了。

3、超音波測距原理

超音波發射器向某一方向發射超音波，在發射時刻的同時開始計時，超音波在空氣中傳播，途中碰到障礙物就立即返回來，超音波接收器收到反射波就立即停止計時。超音波在空氣中的傳播速度爲 340m/s，根據計時器記錄的時間 t，就可以計算出發射點距障礙物的距離，s=340t/2，這就是所謂的時間差測距法。

- **使用**

如圖 2.27 所示爲“創意之星”的超音波測距感測器模組 UP-Sonar5KN。超音波使用 RS-422 介面，將之接至控制卡的任何一個 RS-422 介面，都可以透過 NorthSTAR 進行數值讀取和編程。感測器的規格資料如下：

1. 工作電源：+5V
2. 工作電流：<20mA
3. 工作溫度範圍：－10℃～＋70℃
4. 超音波距離測量：
   最大測量距離—500cm
   最小測量距離—4cm
   解析度－1cm
   誤差－1%
5. 模組內帶溫度補償電路提高測量精度。
6. 模組尺寸：34mm×51mm
7. 模組重量：約 30g



圖 2.27 超音波測距感測器

2.3.3 聲音感測器

- **原理**

1）駐極體式電容麥克風構造與原理

機器人上最常用的聲音感測器就是麥克風。常見的麥克風包括動圈式麥克風、MEMS 麥克風和駐極體電容麥克風。其中，駐極體電容麥克風尺寸小，功耗低，價格低廉而性能不錯，是手機、電話機等常用的聲音感測器。大量具有聲音交互功能的機器人，例如 SONY AIBO，本田 ASIMO，均採用這類麥克風作爲聲音感測器。

駐極體式電容麥克風由一片很輕的振動膜及駐極電荷的背極板組成。構成駐極體式電容麥克風的內部零件相當精密，對外部的雜音很敏感，因此爲預防灰塵或異物質的侵蝕及電器雜音，要緊緊密封在只有音波可流入的圓形金屬殼中。

隨著音波的流入使金屬振動板振動時，振動板與電極板會隨音波的振動，產生距離上的變化，這種物理現象可解釋爲靜電容量變化。因駐極體式電容麥克風的靜電容量值很小，電器的耗電流量較大，故不可直接使用於一般的放大器(擴大器)上。爲符合放大器所要求的輸入信號耗電流量，必須要經由 JFET 使電流量轉換成放大器可接受的程度。

駐極體式電容麥克風隨振動板與背極板極化蓄電荷的類型及構造，可區分爲三大類:

1.背極式麥克風 Back Electret Type Condenser Microphone
2.薄膜式麥克風 Foil Electret Type Condenser Microphone
3.前極式麥克風 Front Electret Type Condenser Microphone

其內部構造如下圖所示：

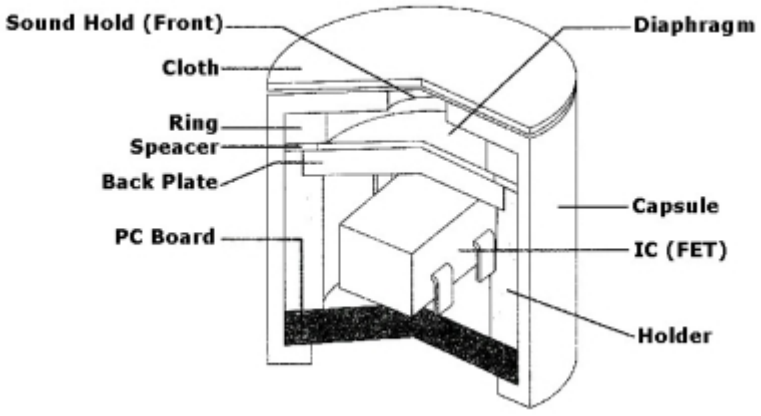


圖 2.28 麥克風內部結構圖

- **使用**

圖 2.29 是“創意之星”的聲音感測器，聲音感測器是開關量感測器，接 IO0~IO11 的任意一個介面都可以透過 NorthSTAR 進行數值讀取和編程。

聲音感測器是一種很好的人機交互設備，您可以借此讓機器人回應您的動作，比如拍下手掌——機器狗站起來，快速拍手——機器狗往前跑。高級版“創意之星”有麥克風，它和

這裏的聲音感測器不一樣，麥克風接在音頻輸入介面上，能夠採集人耳朵能夠聽到的所有聲音，輸出連續的電位信號。聲音感測器是 IO 量感測器，輸出只有 0 或 1 兩種電位狀態，比如聲音高過 60 分貝時輸出爲 1，低於 60 分貝輸出爲 0。

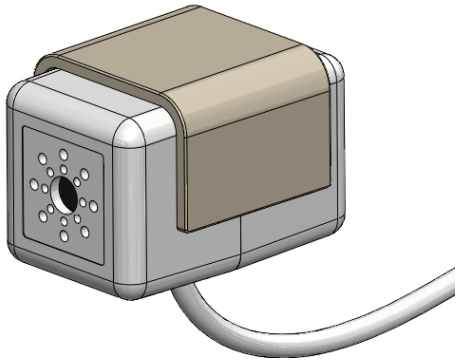


圖 2.29 聲音感測器

2.3.4 碰撞感測器

碰撞感測器是由一個按鈕開關和週邊電路構成，其輸出信號爲數位信號。當按鈕按下時，信號輸出端輸出低電位；按鈕被釋放時，信號輸出高電位。

圖 2.30 是“創意之星”的碰撞感測器，輸出開關量信號，接 IO0~IO11 的任意一個介面都可以透過 NorthSTAR 進行數值讀取和編程。碰撞感測器可以當成機器人的觸角，或某些位置的限制位元開關。

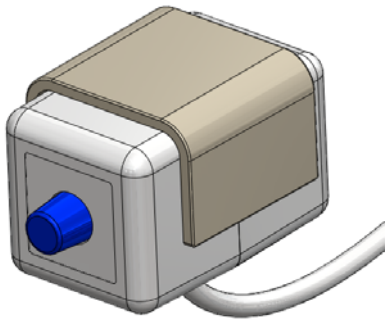


圖 2.30 碰撞感測器

2.3.5 傾覆感測器

姿態感測器採集的是機器人的姿態資訊，專業的姿態感測器（電子羅盤、陀螺儀等）價格昂貴，“創意之星”機器人套件提供了一個基本的姿態感測器——水銀開關（輸出開關量信號）。

水銀開關裏有一個位於玻璃管，可自由移動的水銀粒。水銀開關輸出不同的信號與水銀

粒在玻璃管的位置相關。比如水銀粒位於玻璃管 A 端時，水銀開關輸出高電位；當感測器由於某種原因姿態改變過大移動到玻璃管 B 端，水銀開關輸出低電位。根據水銀開關這種輸出信號與其本身姿態有關的特性，我們可以借此來檢測機器人的姿態。水銀開關在測量機器人是否傾覆上非常好用且成本低廉，所以我們也常將之稱爲傾覆感測器。

圖 2.31 是“創意之星”的傾覆感測器，輸出開關量信號，接 IO0~IO11 的任意一個介面都可以透過 NorthSTAR 進行數值讀取和編程。

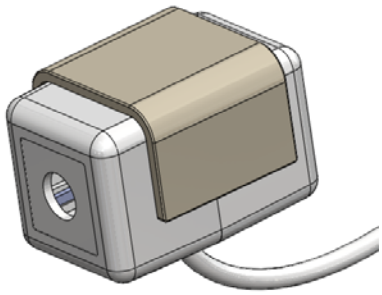


圖 2.31 傾覆感測器

2.3.6 溫度感測器

溫度感測器是一種檢測溫度的零件。其核心是美國 National Semiconductor 公司的 LM35 溫敏感測器。這款感測器的標準溫度檢測範圍是攝氏 0~70 度。爲了感測器的安全起見，不要將感測器放置在超過攝氏 80 度的環境中，該感測器並不具備防水能力，請不要用該感測器測量水溫。

圖 2.32 是“創意之星”的溫度感測器，輸出類比量信號，接 AD0~AD7 的任意一個介面都可以透過 NorthSTAR 進行數值讀取和編程。

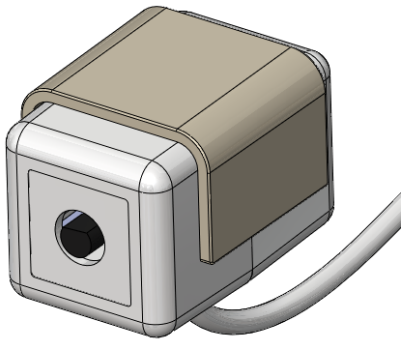


圖 2.32 傾覆感測器

2.3.7 光強感測器

光強感測器對可見光波長的光照強度（專業術語即“照度”）很敏感，其核心元件是一隻光敏電阻，其輸出信號爲與光強相關的類比信號。

圖 2.33 是“創意之星”的光強感測器，輸出類比量信號，接 AD0~AD7 的任意一個介面都可以透過 NorthSTAR 進行數值讀取和編程。

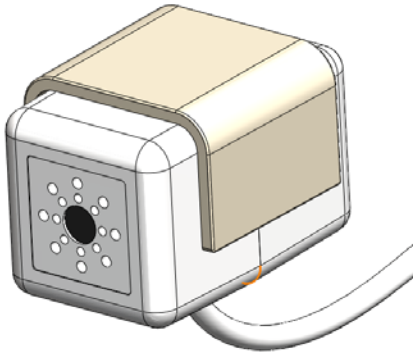


圖 2.33 光強感測器

2.3.8 灰度感測器

灰度感測器透過自身的高亮白色 LED 照亮被檢測物體，被檢測物體反射 LED 的白光。由於不同的顏色對白光的反射能力不一樣，同樣材質白色反射度最高，黑色反射度最低。灰度感測器前端有一個光敏電阻，用於檢測反射光的強弱，據此可以推斷出被檢測物體的灰度值。

圖 2.34 是“創意之星”的灰度感測器，輸出類比量信號，接 AD0~AD7 的任意一個介面都可以透過 NorthSTAR 進行數值讀取和編程。在機器人武術擂臺賽中或者足球機器人比賽中，使用多個灰度感測器組成陣列就可以判斷比賽場地的顏色梯度。在巡線機器人案例中，可以作爲區別白線與周圍地面的感測器。

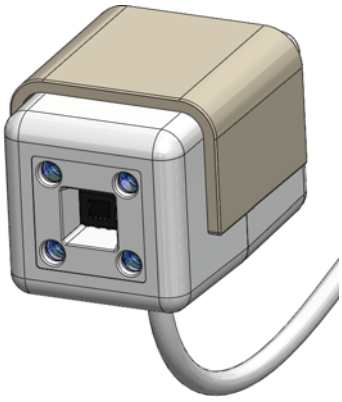


圖 2.34 灰度感測器

2.3.9 PXA270 視覺和語音感測器

在機器人運用中，真正意義上的視覺感測器就是攝影機，它能夠讓機器人和人一樣看到世界萬物。而真正意義上的聽覺感測器就是麥克風，它能讓機器人聽到各種聲音。不過，看到了不一定能夠分辨出來，聽到了不一定能夠聽懂，要讓已經具備眼睛和耳朵的機器人能看

懂，能聽懂，需要您設計相關的程式。

“創意之星”高級版使用的 MultiFLEX™2-PXA270 控制器有兩個 USB 介面，可以接入常見的 USB 攝影機，還有 1 個麥克風輸入介面。我們爲“創意之星”開發了圖像識別引擎，它能夠識別出各種顏色，並將其標示出來。利用這項技術，我們可以讓機器人很清楚的看到綠地毯上的紅球，並且知道紅球的位置。我們爲“創意之星”開發了語音識別引擎，您可以定義要讓“創意之星”識別的語句，當您對“創意之星”說這些語句的時候，“創意之星”能夠將它識別出來。

利用上述的兩種功能，我們可以設計很多有趣的機器人，比如聽話的小狗。第九章我們將指導您設計一個能夠和您語音對答的機器人。第十章我們將把 ROBOCUP 機器人足球比賽的全向機器人平臺小型化，用“創意之星”搭建出全向足球機器人，這種機器人平臺關鍵技術之一就是圖像識別技術，它需要利用攝影機識別場地、球門、足球、己方和對方機器人。

2.4 執行器

機器人之所以有“人”字，是因爲人們希望這種新的人造物能夠像人一樣，有聰慧的大腦、敏感的知覺、靈活的活動能力，能夠代替人完成人們所不期望親力親爲的事情。

在 2.1 我們瞭解了機器人的骨架，2.2 瞭解了機器人的大腦，2.3 瞭解了機器人的感知系統，本小節將爲大家介紹機器人的運動器官。

機器人常用的運動驅動零件有直流有刷馬達、直流無刷馬達、步進馬達、伺服馬達等。“創意之星”配置了直流有刷馬達、微型伺服馬達、機器人專用伺服馬達等。

2.4.1 CDS5500 機器人伺服馬達

proMOTION CDS系列機器人伺服馬達屬於一種集馬達、伺服驅動、匯流排式通訊介面爲一體的整合伺服單元，主要用於微型機器人的關節、輪子、履帶驅動，也可用於其他簡單位置控制場合。CDS 5500的特點如下所示：



圖 2.35 CDS5500 機器人伺服馬達

- 大扭矩：10Kgf.cm 的持續轉動輸出扭矩，大於 20Kgf.cm 位置保持扭矩
- 高轉速：最高 0.16s/60°輸出轉速
- DC 6.5V～8.4V 電壓範圍供電
- 0.32°位置解析度
- 位置伺服控制模式下轉動範圍 0-300°
- 在速度控制模式下可連續旋轉，調速
- 匯流排連接，理論可串聯 254 個單元
- 高達 1M 通訊串列傳輸速率

- ◆ 雙端輸出軸，適合安裝在機器人關節
- ◆ 高精度全金屬齒輪組，雙滾珠軸承
- ◆ 連接處 O 型環密封，防塵防濺水
- ◆ 0.25KHz 的伺服更新率
- ◆ 相容 Robotis Dynamixel 通訊協定
- ◆ 具備位置、溫度、電壓、速度回饋

2.4.1.1 匯流排通訊的特點

CDS5500 採用半雙工串列非同步匯流排通訊，控制器可以透過一個 UART 介面，控制多達 255 個的 CDS5500。這是什麼意思呢？我們借用現實中的一個例子來解釋。

某學校接到上級要來視察工作的通知，校長爲了給領導留個好印象，決定舉行一個隆重的歡迎儀式。他讓宣傳部編排了一些歡迎動作，這些歡迎動作需要 10 個方陣的學生共同表演，並且需要步驟協調，每個方陣由一個老師帶領，所有的方陣由宣傳部長指揮。帶隊老師和部長都發一個對講機，校長要求：

1、部長的對講機調在講話模式，帶隊老師的對講機調在接聽模式。

2、部長統一發佈方陣的動作指令，比如 A 方陣下蹲、B 方陣搖手、C 方陣站立等。每個帶隊老師都能聽到這些命令，A 方陣帶隊老師只關心與 A 方陣有關的命令，不理會和 B 方陣有關的命令，否則動作就亂套了。同樣，其他方陣的老師也只關心自己方陣的命令，除了要求所有方陣都執行的指令之外。

3、命令只能由部長發佈，任何帶隊老師不能給其他帶隊老師發佈動作命令。

4、帶隊老師不能隨便將對講機調爲講話模式，向部長諮詢事情，否則大家都調爲講話模式，就唧唧咋咋的成了一鍋粥了。只有在部長要求回答的時候才能回答，回答完畢後立刻將對講機再調回接聽模式。

某個風和日麗的日子，領導終於到了學校。部長就按照劇本對方陣老師下命令：所有方陣聽指令，我先發佈各個方陣的動作，各個方陣記住自己的動作之後先不執行，等我說執行，所有方陣一起執行。現在開始，A 方陣向前走 5 步，B 方陣搖手，C 方陣放氣球…….. ，現在所有方陣執行動作。

歡迎儀式動作整齊，配合默契，步驟協調，獲得了領導的大加讚賞。

匯流排是指能爲多個功能零件服務的一組資訊傳輸線，它是電腦中系統與系統之間、或者各零件之間進行資訊傳送的公共通路。它由一定的物理線路和介面、資料協定規範所構成。

1、部長和各個帶隊老師的對講機就是這樣的物理線路，它讓指揮者和執行者之間的通信成了可能。校長發佈的幾個規定就是匯流排協定，這個協定讓匯流排不會出現

衝突、紊亂，讓通信能夠正常完成。CDS5500 的半雙工串列非同步匯流排的物理線路是伺服馬達線、UART 介面等，CDS5500 也有一套匯流排協定。

2、部長發佈命令，其他帶隊老師只能接受命令，不能發佈命令，因此我們可以稱部長爲主機，其他的帶隊老師是從機，這個匯流排上只能有一個主機，否則多人發佈命令會導致帶隊老師（從機）不知道聽誰的命令好。從機可以有很多，現在有 10 個方陣，如果歡迎儀式的劇本複雜點，可能需要 255 個方陣也不一定。在 CDS5500 這個匯流排上，“創意之星”控制器就是主機，發佈 CDS5500 需要執行的命令，比如 ID 爲 1 的伺服馬達轉 50°，其他伺服馬達不動。CDS5500 就是從機，這個匯流排可以串聯很多 CDS5500，如圖 2.35 所示。

3、這個匯流排是半雙工匯流排，部長在說的時候，帶隊老師只能聽，帶隊老師要說話必須將接收機調爲講話模式，部長需要調成收聽模式，帶隊老師說完之後需要立刻切換爲收聽模式，否則會影響部長發佈命令。“創意之星”控制器作爲主機，發佈 CDS5500 的動作指令，伺服馬達在控制器要求應答的時候才發送應答資料，發送完成後立刻將自己的介面狀態調整爲接收狀態。

4、每個方陣都有自己的代號，這就是從機的 ID（位址），主機發佈的命令所有從機都可以接收到，但某從機只回應和自己 ID 匹配的命令。CDS5500 都有自己的 ID 號，出廠預設都是 1，您可以使用 RobotServo Terminal 軟體進行修改。

5、部長可以讓帶隊老師逐個執行動作，比如讓 A 方陣先轉向，再讓 B 方陣蹲下，也可以讓所有的方陣記住他們要執行的動作，然後再同時執行，這樣效果會比較整齊。“創意之星”控制器也可以讓 CDS5500 的工作在上述兩種模式，這是由匯流排協定規定的，您可以給 1 號 CDS5500 發轉 50°的命令，2 號伺服馬達轉 30°的命令，然後發一個命令讓所有匯流排上的 CDS5500 同時開始剛才發佈的命令，詳細協定可以查閱 CDS5500 資料手冊。

透過上述的例子您應該已經瞭解了 CDS5500 的半雙工非同步串列匯流排的意思及其特點。那麼匯流排式的機器人伺服馬達和 R/C 伺服馬達相比有什麼優點呢？我們再用上述例子來說明。

R/C 伺服馬達不是匯流排式的，只能直接接到控制器上，並且佔用一個專有埠。這就要求部長必須長 10 張嘴，拿著 10 個對講機才能給 10 個方陣下命令。我們如果要控制 20 個 R/C 伺服馬達，則您設計的控制器必須具備 20 個 R/C 伺服馬達的介面。“創意之星”控制器有 8 個 R/C 伺服馬達介面，能夠同時控制 8 個 R/C 伺服馬達。但匯流排式伺服馬達不一樣，只有一個 CDS5500 介面，理論上就能控制 255 個 CDS5500。爲什麼說是理論上呢？CDS5500 是需要消耗電能的，在工作的時候最高可能有 1500mA 的電流需求，如果在這個匯流排上接入 255 個伺服馬達的話，同時工作起來，匯流排上的電流將達到 200A 的可怕程度。並且實際上，我們也不可能搭建一個具有 255 個伺服馬達的結構。

2.4.1.2 CDS5500 的使用

如 2.4.1.1 小節的介紹，CDS5500 採用半雙工串列非同步匯流排進行控制，每個伺服馬達都有自己單獨的 ID 號，您在機器人結構搭建時需要對 ID 號進行配置，以免機器人的某些關節的 ID 號重疊。出廠的 CDS5500 預設 ID 爲 1，在“創意之星”包裝盒裏有 CDS5500 ID 編

號的貼紙，您設好 ID 後請將貼紙粘貼到伺服馬達後蓋上，避免遺忘。

CDS5500 有專用的調試環境 RobotServo Terminal，在這個環境下，您可以設置伺服馬達 ID、串列傳輸速率、工作模式、速度限制、角度限制、電壓限制等等。光碟 “……\InnSTAR 產品光碟\零件資料匯總\CDS5500 機器人伺服馬達” 裏有 RobotServo Terminal 詳細使用說明和 CDS5500 的資料手冊，這些檔案將指導您使用 CDS5500。

CDS5500 採用匯流排式通訊，您可以串聯使用。使用 R/C 伺服馬達時我們一直很苦惱，如果我們設計的結構有 20 個伺服馬達，我們需要接 20 根線都控制器，整個機器人被長短不一、顏色各異的線所包圍纏繞。CDS5500 讓我們擺脫這種窘境，我們可以將 6 足爬蟲每條腿上的 3 個 CDS5500 串聯在一起，然後再連到控制器。我們可以將機械手的六個關節串聯成一串，最後一根線接到控制器，整個走線非常簡單、清晰，可靠性也高很多。

需要注意的是：每個 CDS5500 需要使用不同 ID；每串 CDS5500 的數量不能太多，最好是 6 個以下。正常工作下，單個 CDS5500 的電流可能達到 500mA~1A，單組 6 個 CDS5500 的電流可能就會有 3-6A。這樣的電流會讓伺服馬達線發熱，並且產生比較大的電壓降，最後一個 CDS5500 可能因爲沿途伺服馬達線的分壓而導致工作電壓過低，CDS5500 電壓過低會出現重定、資料通訊不正常等等狀況。

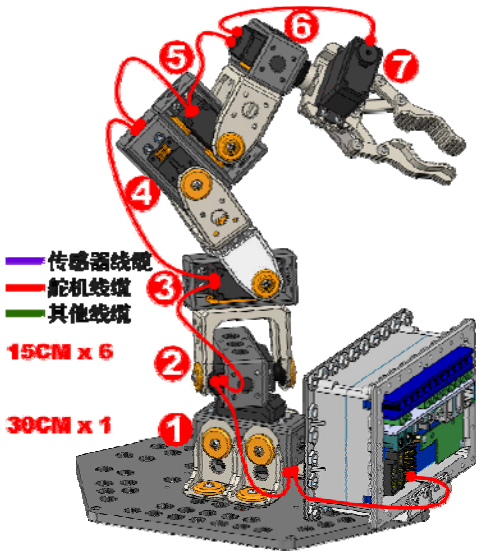


圖 2.36 CDS5500 機器人伺服馬達的串聯使用

第三章我們將有單獨一個小節專門介紹如何使用 “創意之星” 的控制器進行 CDS5500 控制，此處不對 CDS5500 的具體使用做過多描述。

2.4.1.3 CDS5500 的自我保護

CDS5500 還有其他的功能，諸如角度限制、扭矩限制等。在您搭建機器人過程中，這些功能很是重要，某些時候您需要根據結構進行重新配置。

在使用傳統 R/C 伺服馬達時，我們經常遇見伺服馬達長時間卡死，導致過熱燒毀的現

象，這是伺服馬達自我保護功能不完善的一種表現。在實際使用過程中，每一個伺服馬達的使用條件都是不一樣的，有些伺服馬達的使用環境非常良好，有些則是工作條件惡劣。比如機械手根部關節（圖 2.36 的①、②關節），負載遠遠高於機械手爪（圖 2.36 的⑦關節）。

CDS5500 會即時檢測工作電流、負載、溫度，借此判斷各種必要資訊。當出現超載、過熱、過壓、指令錯誤等就會發出警報資訊，甚至直接卸載以保護伺服馬達。如下表所示，CDS5500 定義了 5 種錯誤、危險的工作狀態，您可定義 CDS5500 對這 7 中工作狀態的應對措施，比如報警、指示燈閃亮、卸載等。

注意：上文提到的“卸載”這個概念，您可以將其理解爲“切斷扭矩輸出”。CDS5500 卸載進入保護模式，意思是 CDS5500 由於出現錯誤工作狀態，進入保護模式，切斷伺服馬達的扭矩輸出，伺服馬達處於沒有力量的狀態。卸載這個概念在接下去的章節裏還會重複出現。

表 2.6 CDS5500 保護狀態說明

名稱	詳細說明
0	---
指令錯誤	如果收到一個未定義的指令或收到ACTION前未收到REG WRITE 指令
超載	位置模式執行時最大輸出扭矩小於負載
校驗和錯	指令校驗和錯誤
指令超範圍	指令超過指定範圍
過熱	溫度超過指定範圍
角度超範圍	角度超過設定範圍
過壓欠壓	電壓超過指定範圍

您可以對以上的保護措施進行參數調節，以達到適合使用條件又能保護 CDS5500 的目的。主要有以下四個方面：

- 角度限制

CDS5500 在伺服馬達模式下，有效的角度控制範圍是 0-300°，對應控制量爲 0-1023。在某些運用場合，您可能需要限制伺服馬達的轉角，比如伺服馬達轉過 200°之後可能出現卡死，您需要將角度限制設置爲 0-682（682 對應 200°），當您的控制指令在 0-682 之間伺服馬達能夠按照您的指令運動，您的指令如果超過 682 伺服馬達將保持到 682 位置。

- 電壓限制

您可以設置 CDS5500 的工作電壓範圍，由於硬體設計的限制，CDS5500 的工作電壓是 6.5V-10.5V，低於 6.5V CDS5500 將不能正常工作，高於 10.5V 會燒毀 CDS5500。您設定好電壓限制之後，當 CDS5500 上的載入電壓高於或低於此範圍伺服馬達將會置位元錯誤標誌位元，軟體的“電壓超範圍”指示燈會點亮。

- 溫度限制

CDS5500 能夠檢測自身的溫度，溫度過高可能會燒毀伺服馬達。您可以設置正常工作溫度範圍，當 CDS5500 檢測到溫度過高後，伺服馬達將會強制卸載停止工作。

- 轉矩限制

“轉矩限制”限制了 CDS5500 的最大工作電流，起到限制 CDS5500 的最大輸出扭矩的

作用。在機器人手掌關節或需要長時間運轉的場合常常用到這個功能，預設爲最大值 1023，即爲最大扭矩輸出。透過限制扭矩大小，您可以讓機械手爪能夠抓起草莓，而又不會將草莓捏碎。

2.4.1.4 CDS5500 的調試設置軟體

CDS5500 的參數設置、保護參數調節、性能測試等都可以透過 RobotSevo Terminal 軟體來實現。您可以在光碟 “……\InnSTAR 產品光碟\零件資料匯總\CDS5500 機器人伺服馬達”裏找到軟體和軟體使用說明。

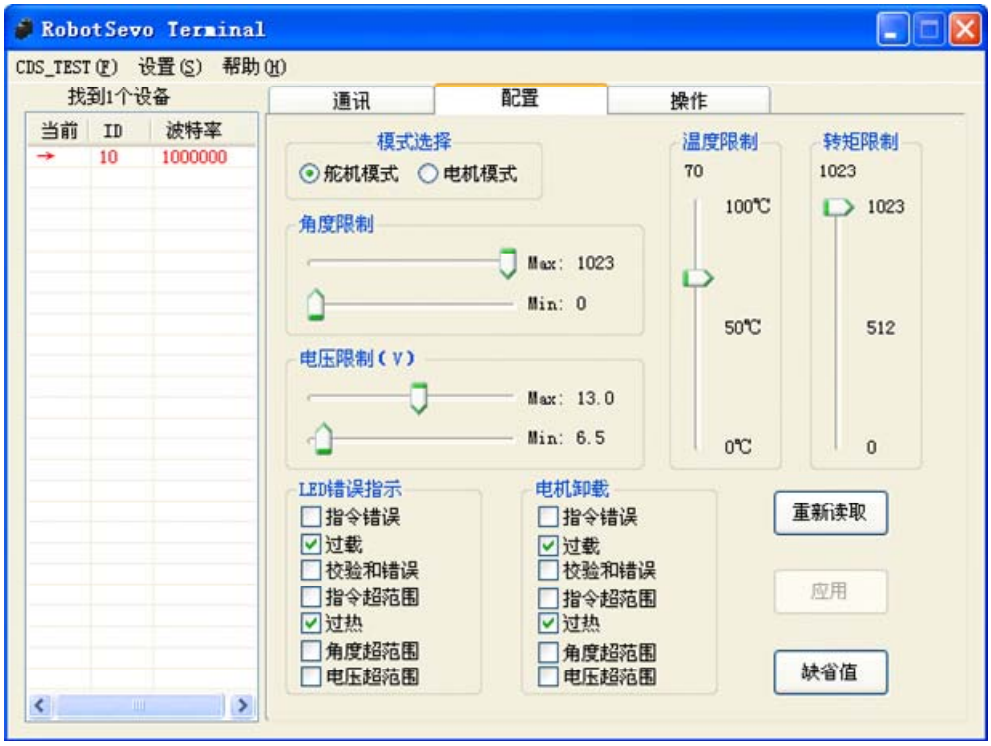


圖 2.37 RobotSevo Terminal

2.4.2 CDS5401 大扭矩 R/C 伺服馬達

“創意之星” 可以選用傳統 R/C 伺服馬達，MultiFLEX™2-AVR 控制器和 MultiFLEX™ 2-PXA270 控制器都有 8 個 R/C 伺服馬達介面。我們推薦使用的 R/C 伺服馬達型號是 SolidMOTION CDS5401。實物如下圖所示：



圖 2.38 CDS5401 伺服馬達

什麼是 R/C 伺服馬達？

伺服馬達應該稱爲微型伺服馬達系統，早期在模型上使用最多，主要用於控制模型飛機、飛艇、船隻等的伺服面，所以俗稱伺服馬達。早期的伺服馬達都是使用類比電路進行控制，之後陸續出現了數位電路控制的伺服馬達、機器人專用的匯流排式機器人伺服馬達等，爲了區分這些不同種類的伺服馬達，我們將使用類比電路控制的伺服馬達稱爲類比伺服馬達或者 R/C 伺服馬達。

表 2.7 CDS5401 的各項參數

項目	數據	說明
型號	SolidMotion CDS5401	數位化伺服馬達，雙端輸出軸
長/寬/高	40/19.8/38mm	標準伺服馬達。可相容大部分型號，包括 Futaba、JR、Hitec 等品牌
供電	4.8~7.4VDC	推薦 4.8~6V
最大扭矩@6V	13kgf.cm	即 1.2Nm
最大電流@6V	2.5A	運轉電流。持續不能超過 20s
空載速度	0.22sec/60degree	
重量	55g	
介面	3 線制	GND/VCC/SIG
輸出軸	銅合金	25 牙，花鍵
輸出軸軸承	滾珠軸承	雙滾珠軸承支撐
輸出線纜長度	30cm	隨機提供延長線

CDS5401 的連接電纜爲 3 條線，如下表所示：

編號	顏色	說明
1	棕色	GND
2	紅色	VCC，4.8~7.2VDC
3	黃色	伺服馬達 PPM 信號

“創意之星”套件提供的 CDS5401 伺服馬達屬於大功率零件,在使用原裝電源的情

況下請注意不要讓其運轉超過 20 秒。如果發現伺服馬達外殼發熱燙手，請立即停止使用，並等待其溫度降低之後再使用。長期運轉會造成伺服馬達溫度上升並燒毀內部的電路。

CDS5401 伺服馬達的輸出扭矩高達 13kg.cm，因此切勿在伺服馬達運轉時將手指或身體的任何部分伸入任何可能會卡住伺服馬達的位置。運轉的伺服馬達會造成手指受傷，甚至有骨折的危險。

我們對伺服馬達的基礎知識進行了總結，在光碟 “……\InnSTAR 產品光碟\零件資料匯總”的資料夾下，您可以找到《伺服馬達知識匯總》的文件。

2.4.3 BDMC1203 馬達驅動模組

您如果要搭建速度較高、推力較大的輪式或履帶式機器人的話，CDS5500 作爲直流馬達使用的方案可能不能滿足您的功率需求。這時候推薦您使用“馬達驅動模組+直流有刷馬達”的方案。

BDMC1203 使用 CDS5500 的標準控制協定，您可以將之當成 CDS5500 的馬達模式即可。

BDMC1203 工作電壓 6.5V~12V，峰值電流 3A，最高可以提供接近 36W 的功率輸出。實物如圖 2.39 所示，兩個白色三針接線端子和 CDS5500 相容，在電路板上焊接的紅黑兩個線接馬達的正負極。



圖 2.39 BDMC1203

如果我們要讓 BDMC1203 上的馬達轉動起來一般需要經過以下幾個步驟：

1、 將馬達的正負極接到 BDMC1203 的馬達介面上。

2、 使用 CDS5500 的伺服馬達線將 BDMC1203 接到調試器上，使用 RobotSevo Terminal 將其設爲馬達模式，配置合適的 ID，不能和匯流排上的 CDS5500 或其他的 BDMC1203 的 ID 衝突（出廠時預設是馬達模式）。

3、 使用 CDS5500 的伺服馬達線將 BDMC1203 接到控制器的數位伺服馬達介面上。

4、 使用伺服馬達控制項編程式控制，注意使用馬達模式。

2.4.4 Sabel 2x10 大功率馬達驅動模組

BDMC1203 能夠提供 36W 的功率輸出，Sabel2x10 可以提供 2 路 240w 的功率輸出，採

用這個模組加上 100W 以上的馬達，可以製作重達 50 公斤、移動速度達 2m/s 的輪式機器人。這兩款馬達驅動模組都可以配套 FAULHABER 2342 24CR 馬達在“創意之星”上使用，代替 CDS5500 作爲輪子和履帶的驅動馬達。



圖 2.40 Sabel2x10 直流驅動模組

Sabel2x10 有多種工作模式，透過指撥開關進行選擇，與 MultiFLEX™2-PXA270 控制器或 MultiFLEX™2-AVR 控制器配套使用時，我們選擇模式二：微控制器脈衝控制方式。利用 8 個 R/C 伺服馬達介面進行控制，方法如下：

1、 如圖 2.40 所示，左側 0V、5V、S1、S2 分別接控制信號的 GND、5V、PPM1、PPM2。右側 M1A、M1B、M2A、M2B 分別接兩個馬達的正負，B+、B-接電源的正負。S1 的 PPM 控制接在 M1A、M1B 的馬達，S2 的 PPM 控制接在 M2A、M2B 的馬達。

2、 在 NorthSTAR 編程時，將您所使用的 R/C 伺服馬達介面設置爲馬達模式。

3、 修改伺服馬達控制項的速度即可控制 Sabel2x10 的馬達驅動速度，速度範圍爲 0~1023。

注意：B+、B-的輸入電壓範圍是 6V~24V 之間，過低驅動模組不能正常工作，過高會導致驅動模組燒毀。

2.4.5 Faulhaber 大功率減速馬達

在進行機器人比賽時我們會發現，使用 CDS5500 驅動輪子會有點力不從心，速度比較低、力量比較小。

在 2.4.4 和 2.4.5 我們介紹了兩種馬達的驅動模組，這兩種驅動模組配合 FAULHABER 2342 24CR 馬達能夠提供接近 36W 的功率。

表 2.8 FAULHABER 2342 參數

項目	額定數據	說明
額定電壓	24VDC	
額定電流	1.5A	請在此電流以下使用，長期超載會燒毀馬達
額定功率	20W	
減速器	63：1	行星齒輪減速器
空載輸出轉速	120rpm	
空載電流	0.15A	
重量	140g	
尺寸	D32mm,L70mm	最大尺寸

該馬達功率大，輸出扭矩大，其結構爲爲空心杯直流馬達，運轉平穩。配套的減速器也具有良好的機械性能，非常適合作爲輪子驅動。

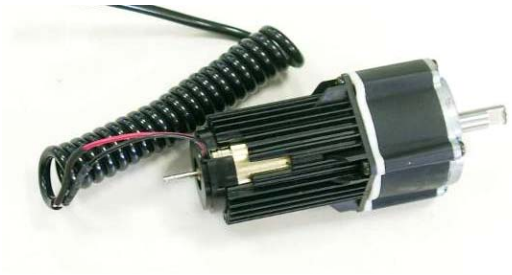


圖 2.41 2342 直流馬達

2.5 NorthSTAR 圖形化整合開發環境

NorthSTAR 是一個圖形化互動式機器人控制程式開發工具。在 NorthSTAR 中，透過滑鼠的拖動類似邏輯框的控制項和對控制項做簡單的屬性設置，就可以快捷的編寫機器人控制程式。程式編輯完後，可以編譯並下載到機器人中執行。NorthSTAR 編程環境具有操作簡便、功能強大特點，能讓您在圖示拖動中建立複雜的邏輯，讓您的機器人按照您的意願動作。

除了本章的內容之外，您可以在光碟“……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第二章\NorthSTAR 編程 10 例”資料夾下找到基於 NorthSTAR 進行程式開發的 10 個範例及其原始碼。

什麼是控制項？

控制項是圖形用戶介面(GUI)中螢幕上的一種物件，使用者可操作該物件來執行某一行爲，控制項是使用者可與之交互以輸入或運算元的物件。在 NorthSTAR 裏，您可以看到很多控制項，比如下圖的伺服馬達控制項，您操作這個控制項的屬性，輸入適當的資料，就可以完成對伺服馬達的控制。



2.5.1 安裝及使用介紹

NorthSTAR 安裝時，直接雙擊執行產品附帶光碟中的安裝檔（  “……\InnSTAR 產品光碟\NorthSTAR 圖形化整合開發環境”），按照提示操作即可。

具體安裝過程如圖 2.43——2.48。

需要注意的是圖 2.46 中可以選擇安裝元件。NorthSTAR 是主程序，Stk500_tool 是下載工具。如果已經安裝了 AVRStudio，就不需要安裝 Stk500_tool。

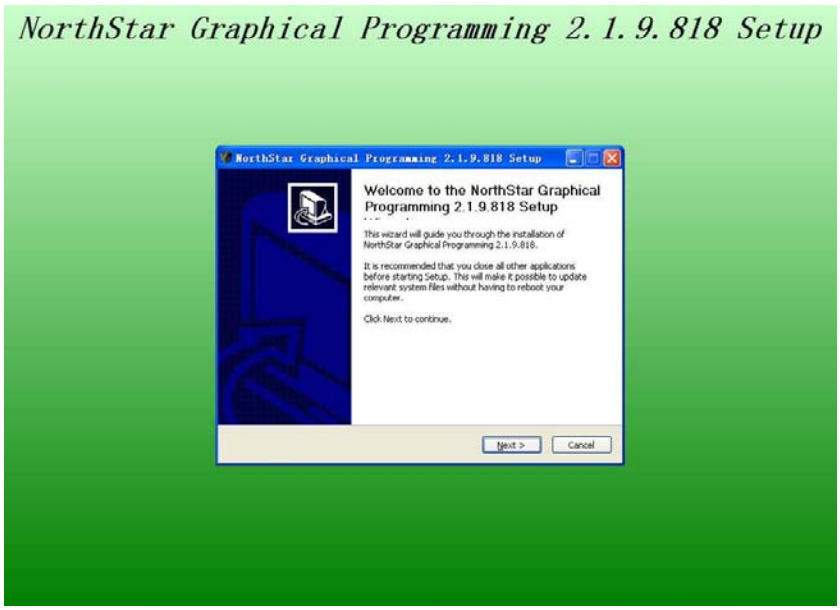


圖 2.43 歡迎介面

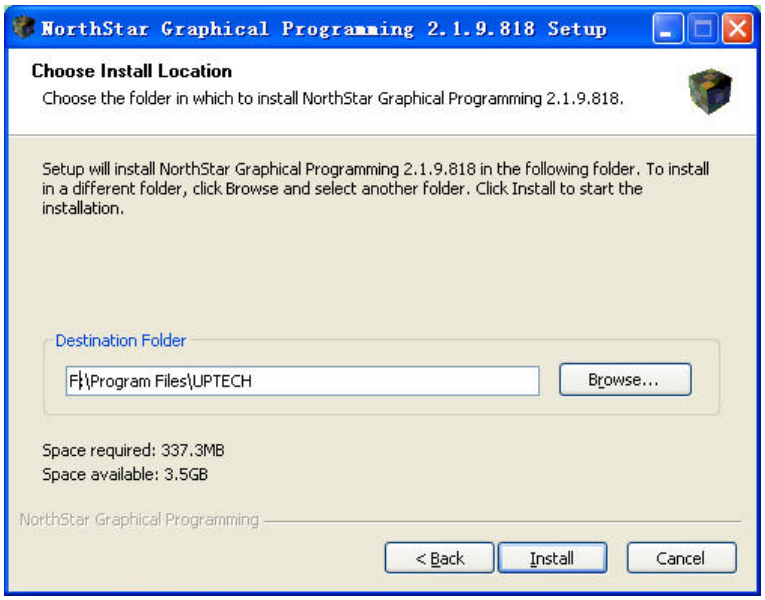


圖 2.44 安裝路徑選擇

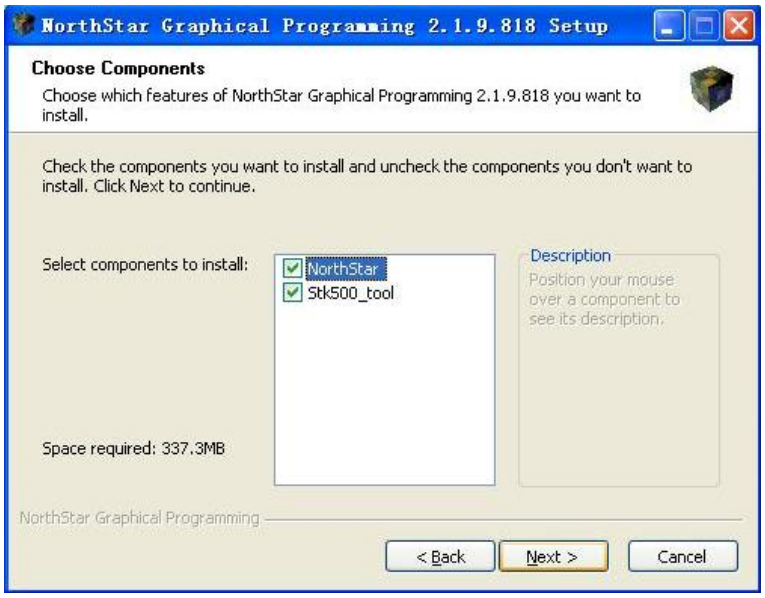


圖 2.45 安裝組件選擇

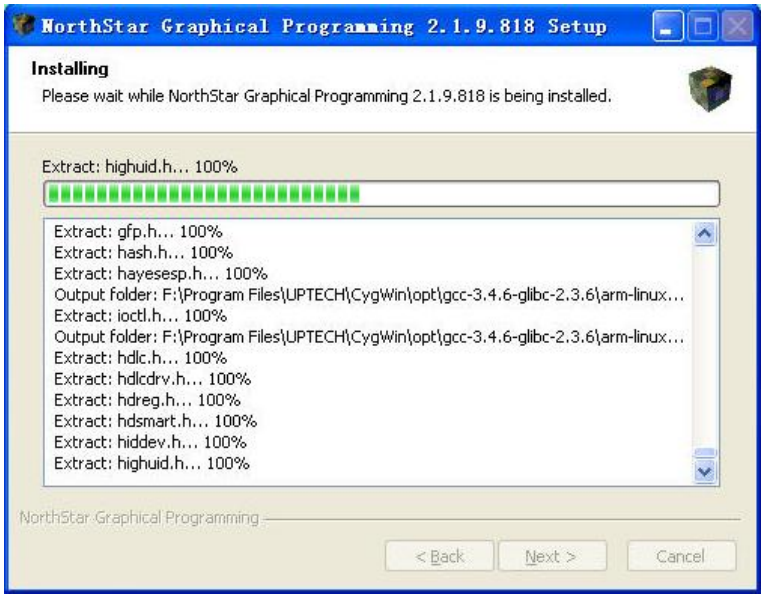


圖 2.46 安裝過程

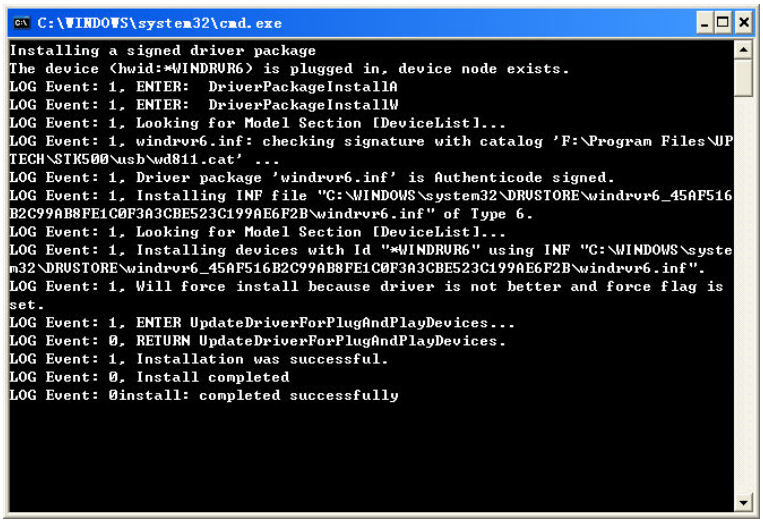


圖 2.47 安裝 Stk500_tool 下載工具

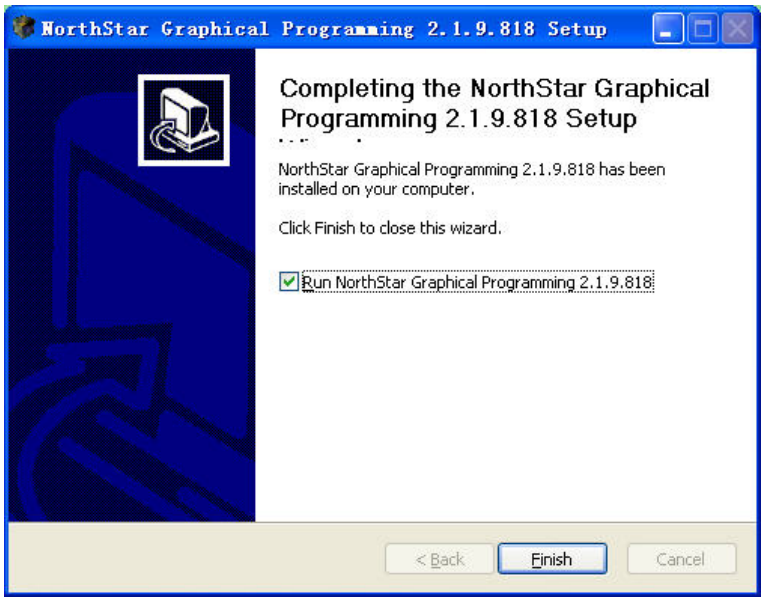


圖 2.48 完成安裝

2.5.2 使用流程圖開發

這裏用一個例子來說明它的使用。該例子的目的是用一個開關控制伺服馬達的正反轉。

1. 從開始→程式→NorthSTAR 目錄執行程式，從功能表或者工具欄選擇“新建”，跳出專案選項，選擇控制器爲 MultiFlex2—PXA270，選擇結構爲 Custom，如圖 2.50 所示；點擊下一步按鈕，設置伺服馬達個數爲 2，將 ID 爲 2 的伺服馬達設置爲馬達模式，如圖 2.51 所示；點擊下一步按鈕，不用設置 AD，直接點擊下一步，如圖 2.52 所示；設置 IO 個數爲 1，點擊完成，如圖 2.53 所示；

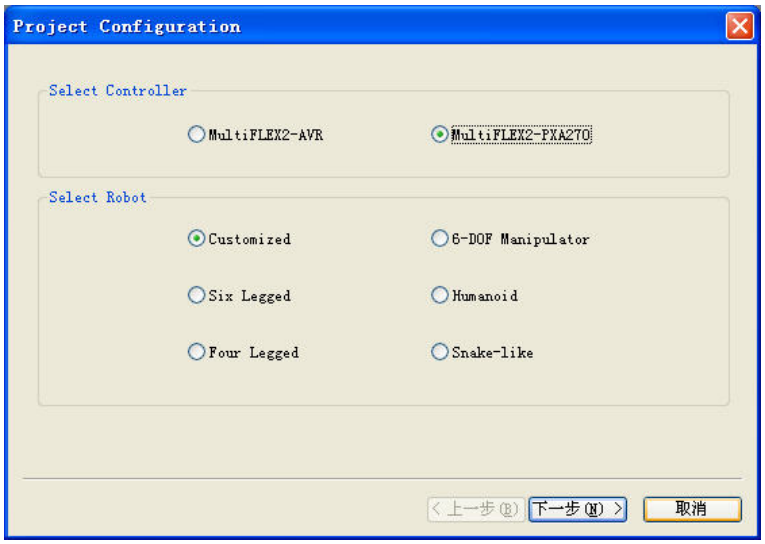


圖 2.50 設置控制器和結構

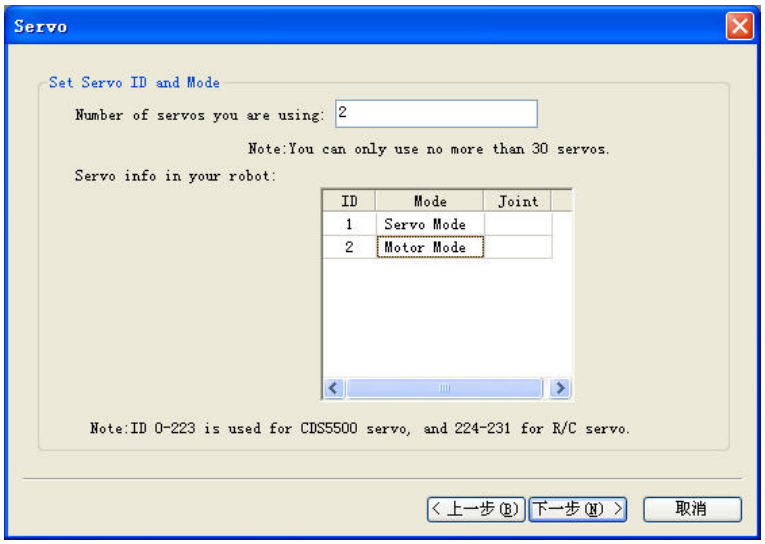


圖 2.51 設置伺服馬達

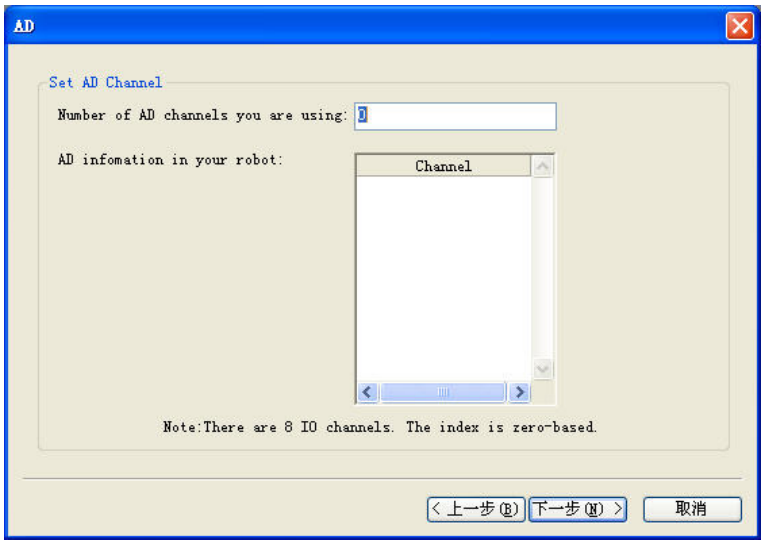


圖 2.52 設置 AD

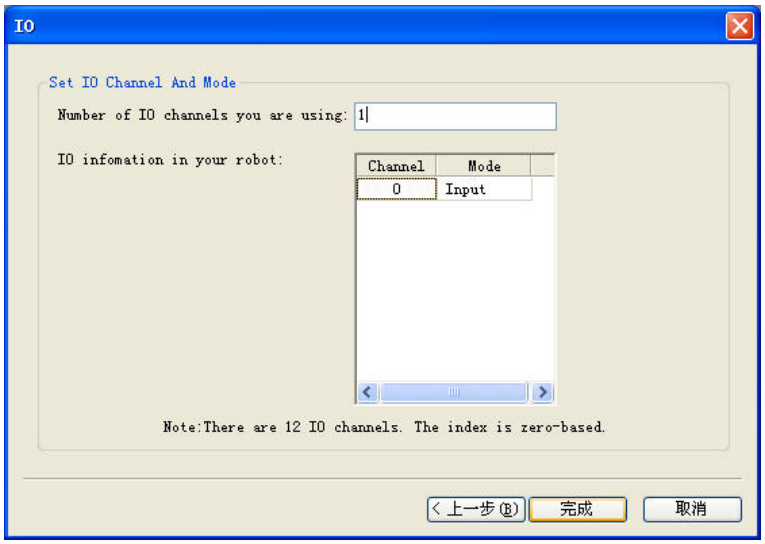


圖 2.53 設置 IO

2. 建立“while”和“while-end”模組，如圖 2.54 所示；



圖 2.54 建立“while”和“while-end”模組

3. 添加變數以保存感測器的值，設置屬性如圖 2.55 所示；

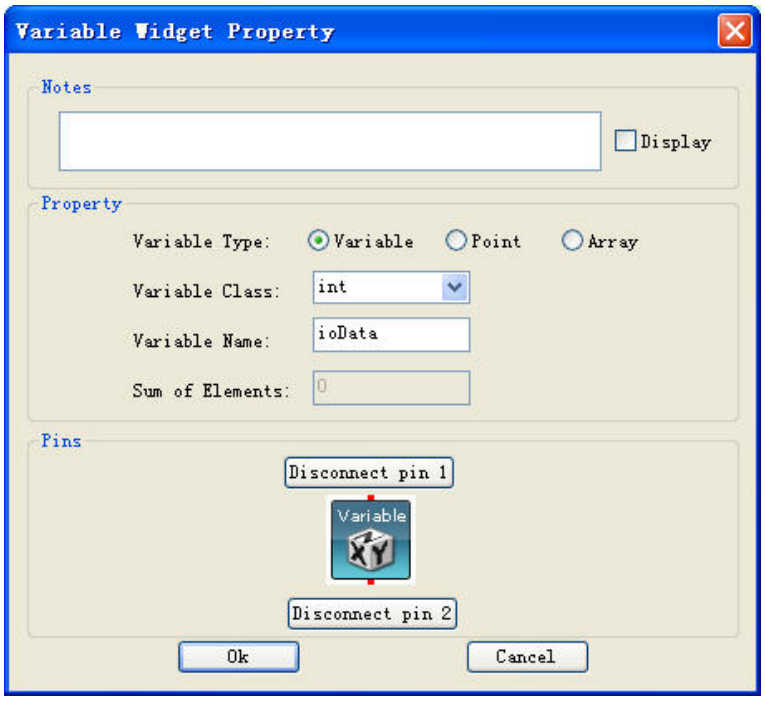


圖 2.55 設置變數模組屬性

4. 建立“IO Input”模組查詢開關的狀態，設置屬性如圖 2.56 所示；

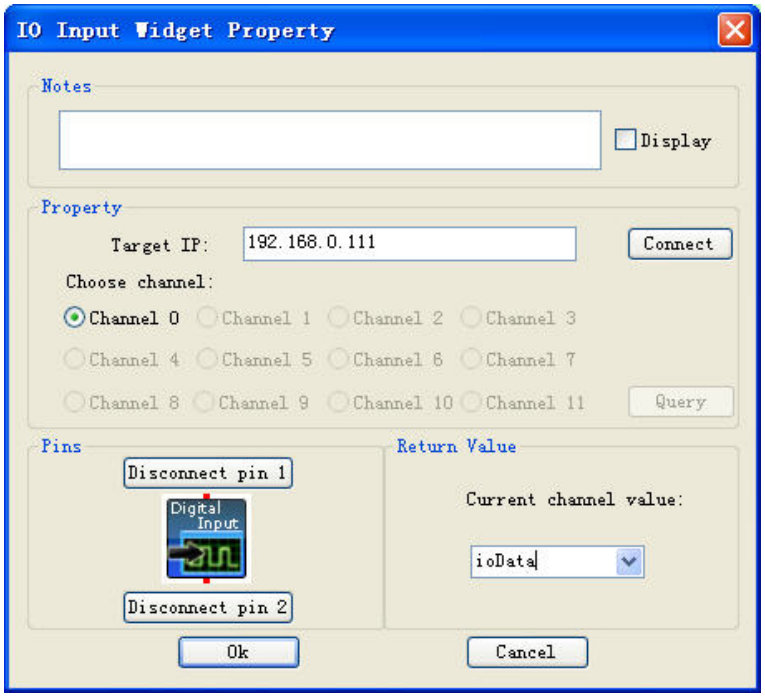


圖 2.56 設置“IO Input”模組屬性

5. 初步連接如圖 2.57 所示；



圖 2.57 初步連接各模組

6. 建立“if”和“if-end”模組，判斷開關狀態，設置屬性如圖 2.58 所示；

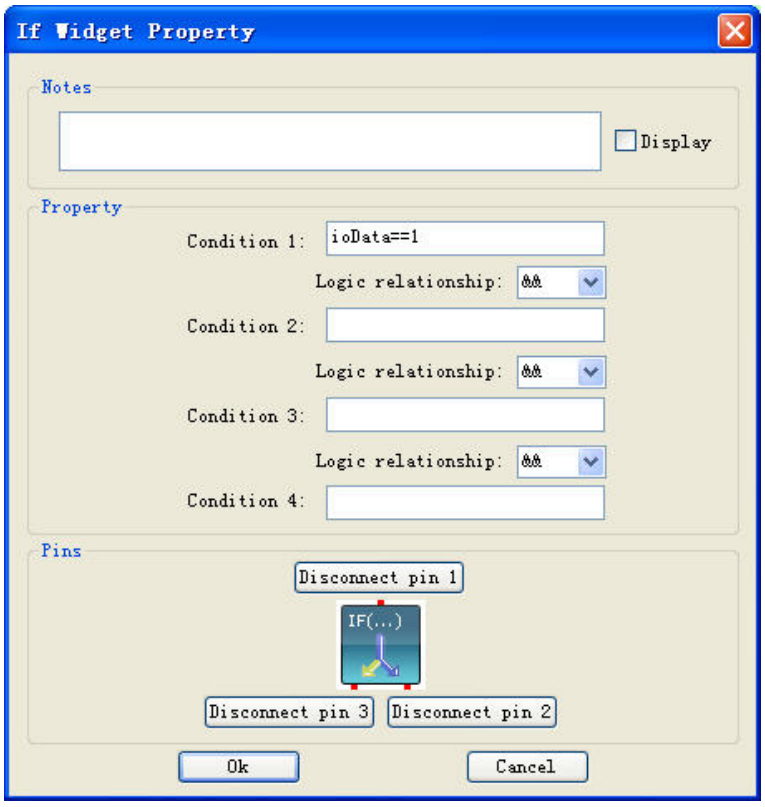


圖 2.58 設置“if”模組屬性

7. 建立“Servo”模組，根據開關狀態讓伺服馬達正轉，設置屬性如圖 2.59 所示；

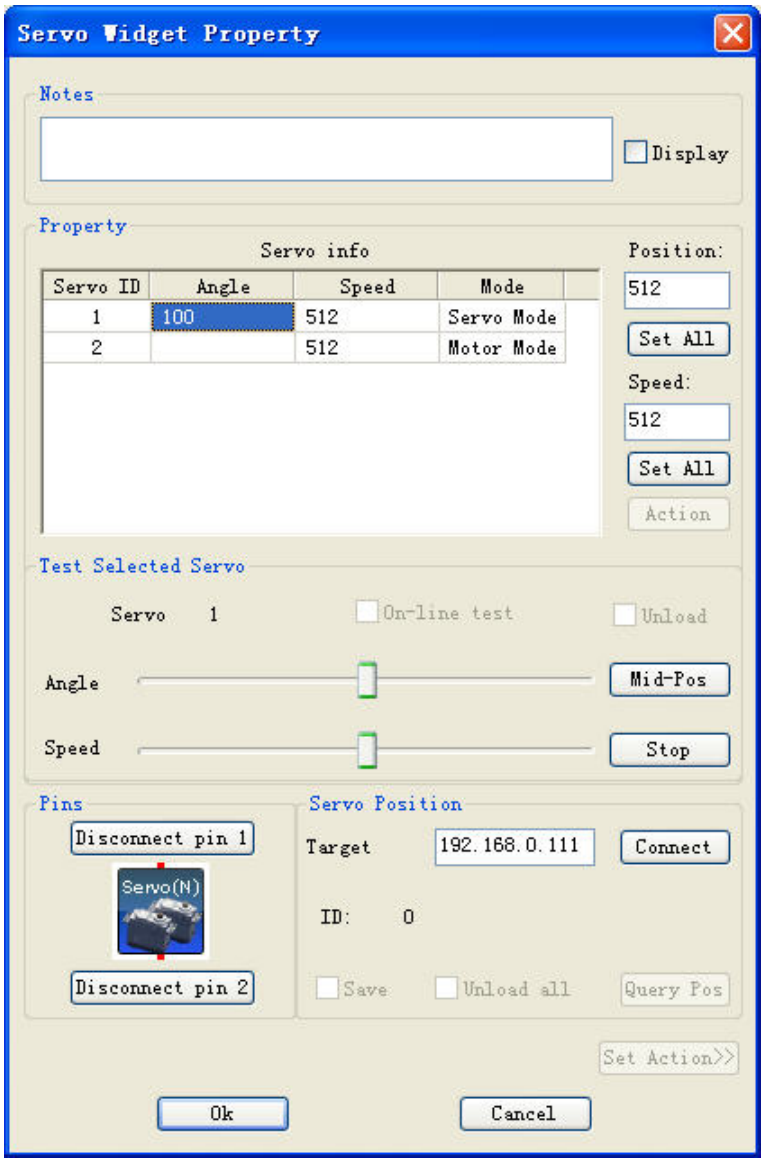


圖 2.59 設置“Servo”模組屬性

8. 連接模組如圖 2.60；



圖 2.60 連接模組

9. 建立“Servo”模組，讓伺服馬達反轉，設置屬性如圖 2.61 所示；

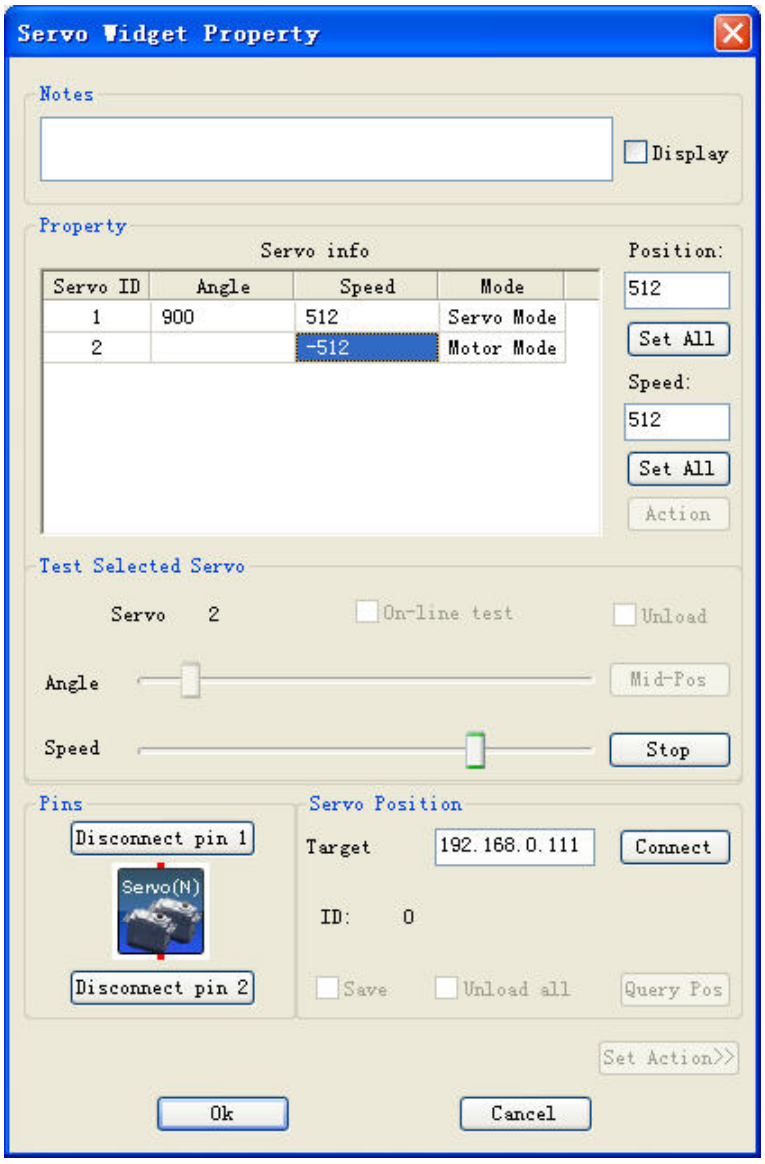


圖 2.61 設置“Servo”模組屬性

10. 建立“Delay”模組，讓伺服馬達有時間執行動作，設置屬性如圖 2.62；

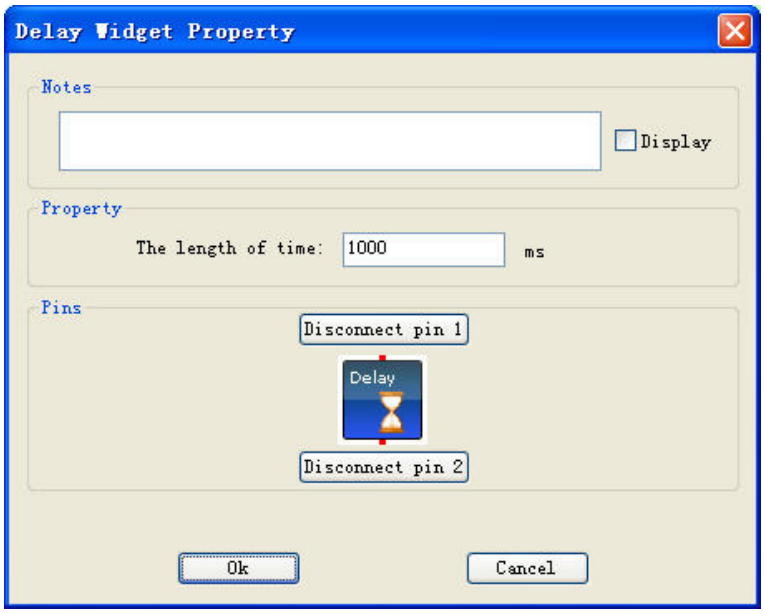


圖 2.62 設置“Delay”模組屬性

11. 連接所有模組如圖 2.63；



圖 2.63 連接所有模組後

12. 從功能表或者工具欄選擇“Compile”，結果如圖 2.64 所示；

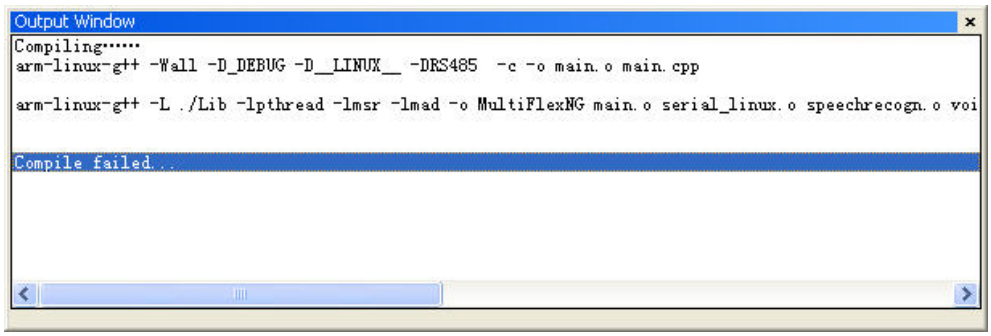


圖 2.64 編譯結果

13. 連接控制器，從功能表或者工具欄選擇“Download”，即可下載程式；把 ID 爲 1 和 2 的伺服馬達連接到控制器上，將一個開關插入 IO0 通道，按下開關和放開時，伺服馬達就會正反轉。

至此，我們已經完成了開關控制伺服馬達正反轉的例子。

2.5.3 手寫程式碼開發

在流程圖編輯過程中，如果需要手動輸入程式碼，從 Tools 功能表或者工具欄點擊 Edit Code，軟體就會切換到程式編輯模式，如圖 2.65 所示。此時手動輸入程式碼，然後編譯，下載，即可執行程式。圖中選擇的區域即爲手動輸入的程式碼。可以透過 File 功能表下的 Save Code 將代碼視窗的代碼保存成.c 或者.cpp 檔，或者透過 Load Code 來載入程式碼檔案到程式碼視窗。

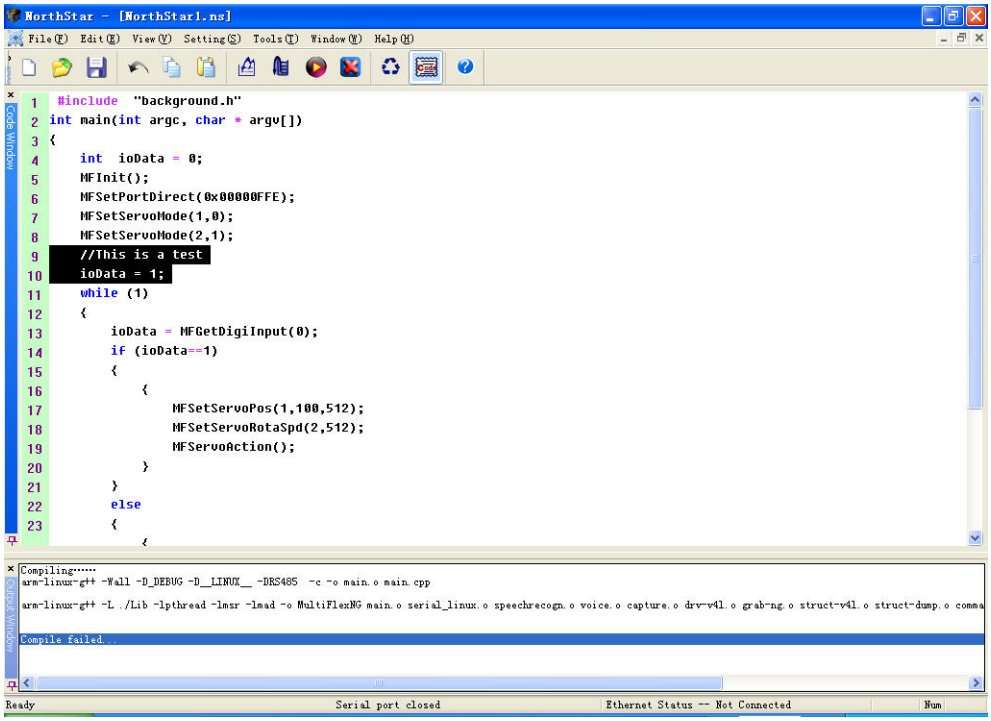


圖 2.65 程式碼編輯模式

2.5.4 調試與線上監控

NorthSTAR 具有調試和線上監控功能，可以即時查詢感測器的值、伺服馬達位置，或者設置伺服馬達速度和位置。NorthSTAR 中的調試功能分爲兩種：模組調試和流程圖調試。

模組調試有“IO Input”模組，“AD Input”模組和“Servo”模組調試三種。

當控制器選擇爲 MultiFlex2—PXA270 時，打開“IO Input”模組屬性，如圖 2.66 所示。如果沒有鏈結到控制器，“Query”按鈕不可用。用網路線連接 PC 和控制器，點擊“Connect”按鈕，稍等片刻建立連接後，“Query”啓用，此時選擇要查詢的通道，點擊“Query”即可進行查詢。如果控制器爲 MultiFlex2—AVR，模組屬性如圖 2.67 所示。把下載線插入到控制器上，點擊“Start service”啓動服務，下載完程式後，打開串列埠，此時“Query”啓用，點擊即可查詢當前中通道的值。

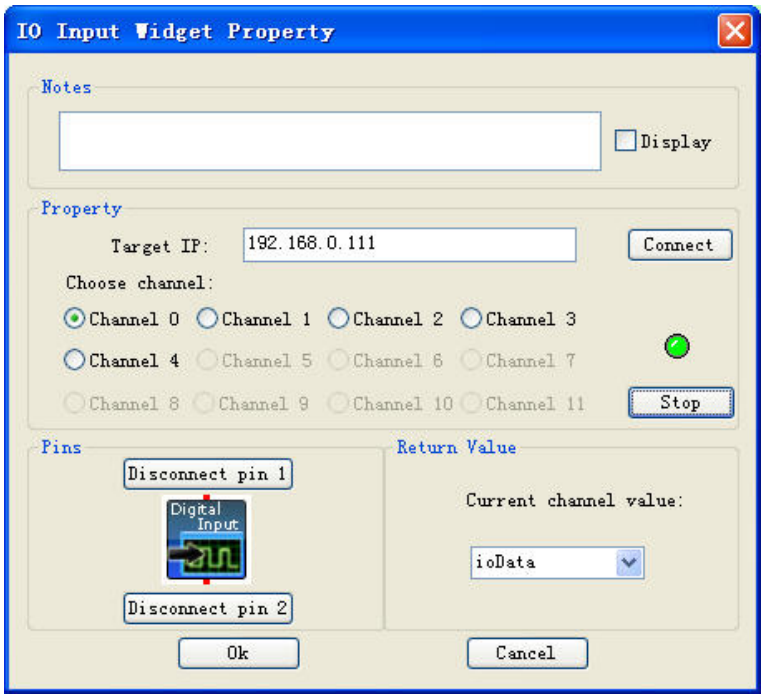


圖 2.66“IO Input”模組調試——MutliFlex2—PXA270

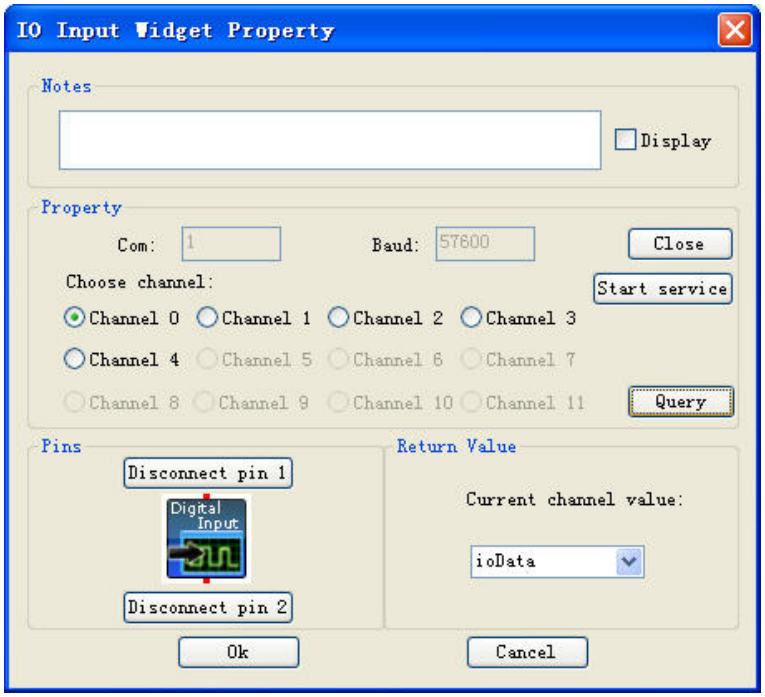


圖 2.67 “IO Input”模組調試——MutliFlex2—AVR

“AD Input”模組和“IO Input”模組的調試功能用法相同。如圖 2.68 和 2.69 所示。

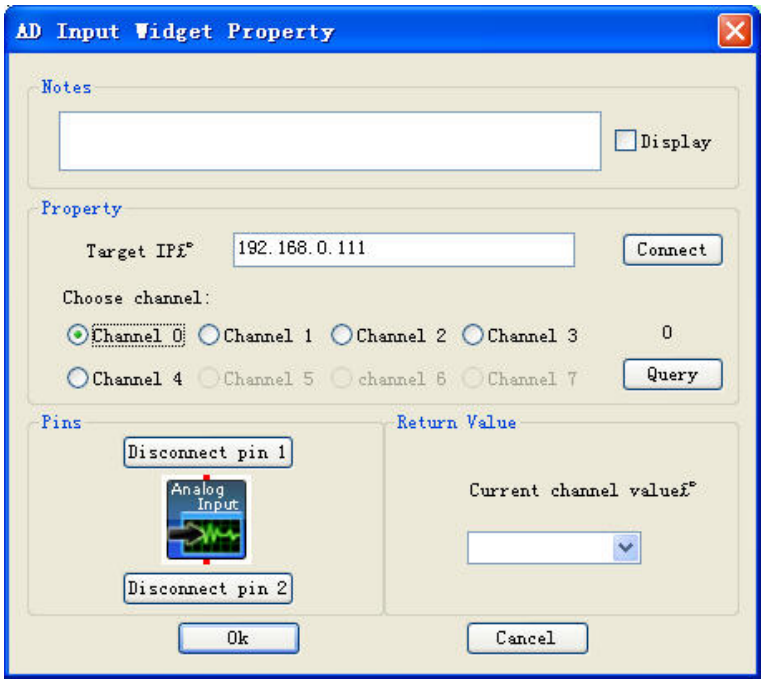


圖 2.68 “AD Input”模組調試——MutliFlex2—PXA270

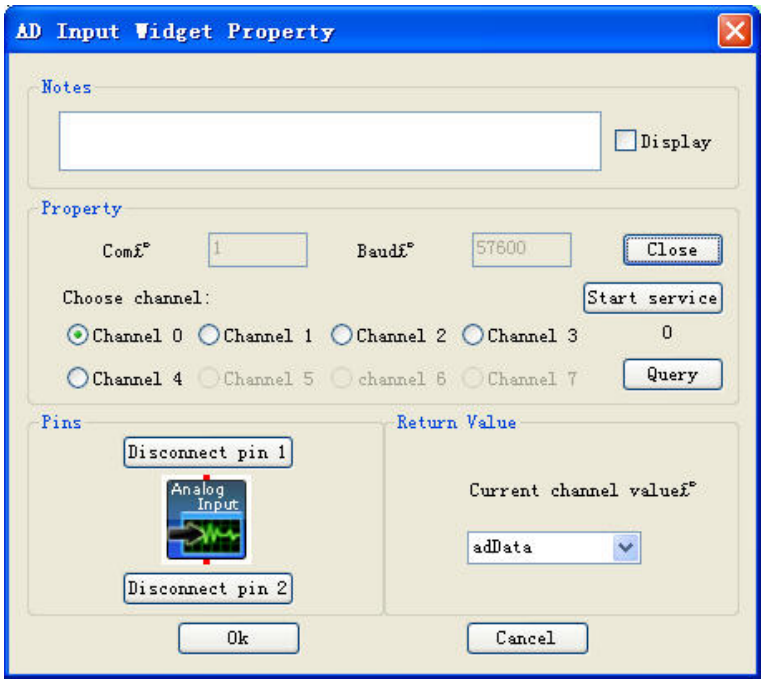


圖 2.69 “AD Input”模組調試——MutliFlex2—AVR

“Servo”模組的調試和“IO Input”調試模組相似。如圖 2.70 和 2.71 所示。當控制器爲 MultiFLEX™2-PXA270 控制器時，先和控制器建立網路連接，然後就可以查詢伺服馬達位置，保存伺服馬達位置，調試伺服馬達，卸載伺服馬達。

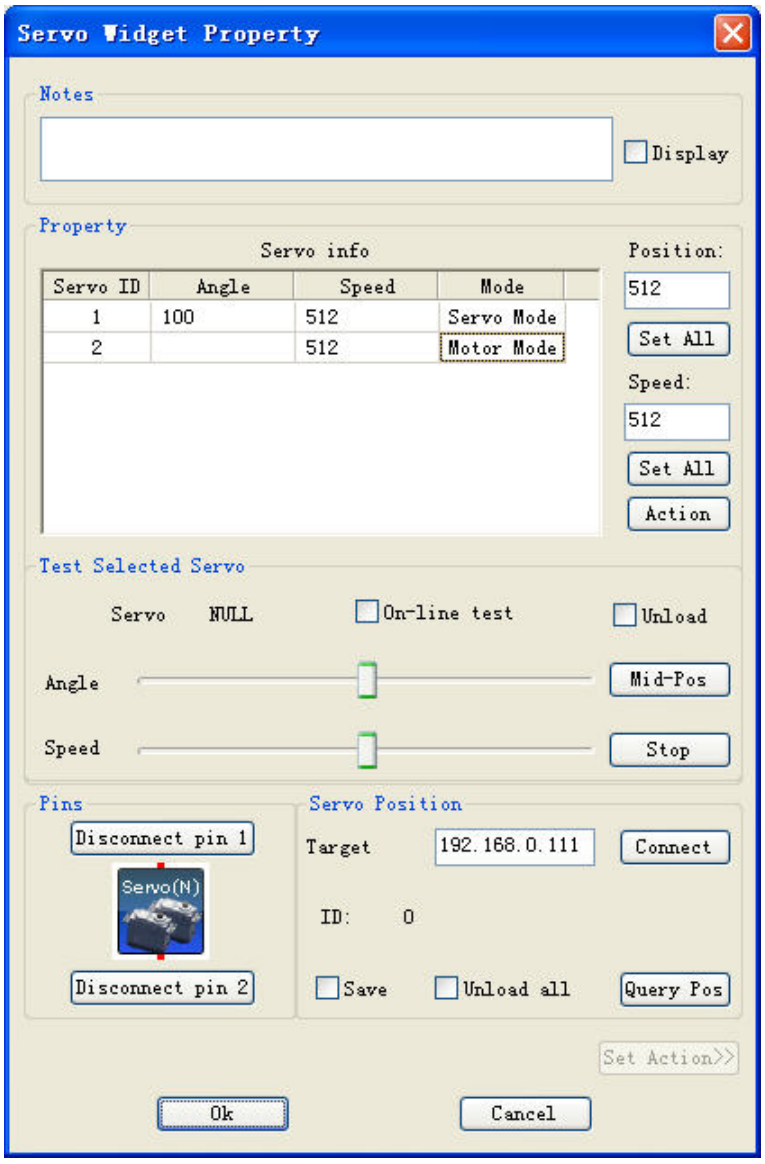


圖 2.70 “Servo”模組調試——MutliFlex2—PXA270

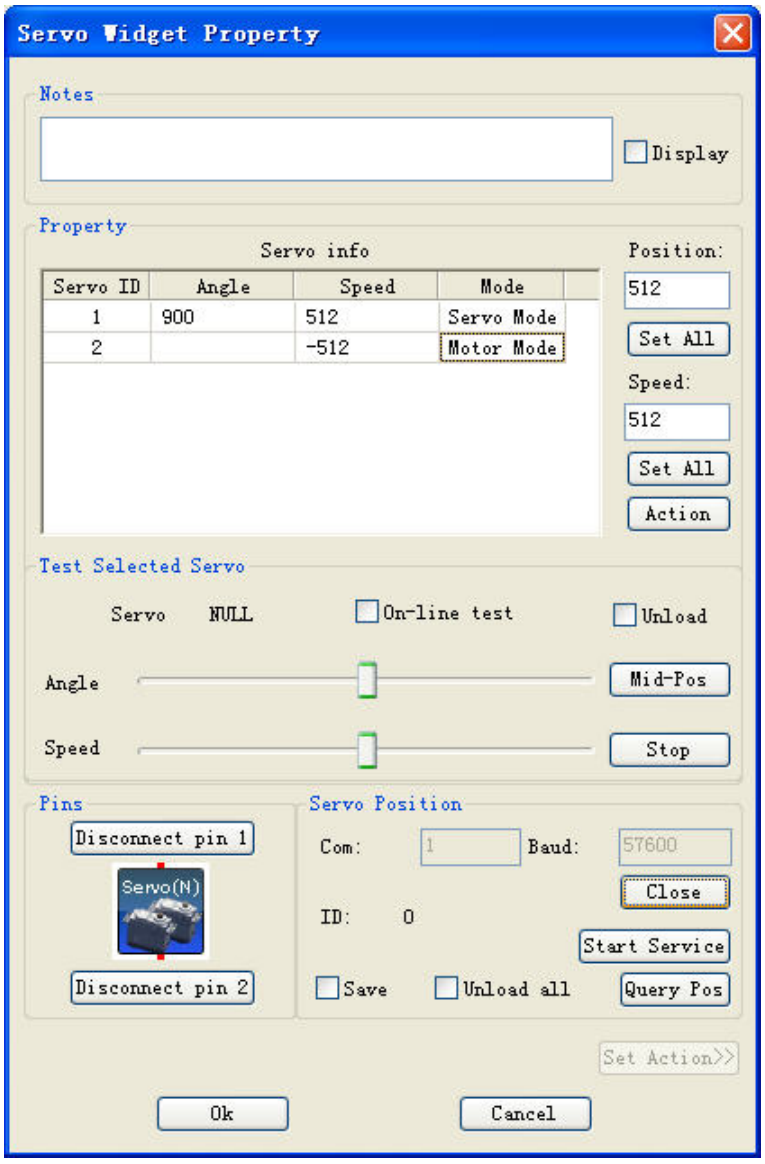


圖 2.71 “Servo”模組調試——MutliFlex2—AVR

此外可以透過“Query IO/AD”功能查詢當前 IO 或者 AD 通道的值，如圖 2.72 和 2.73 所示。還可以透過“Test Servo”功能單獨調試伺服馬達，如圖 2.74 和 2.75 所示。具體操作和模組調試類似。

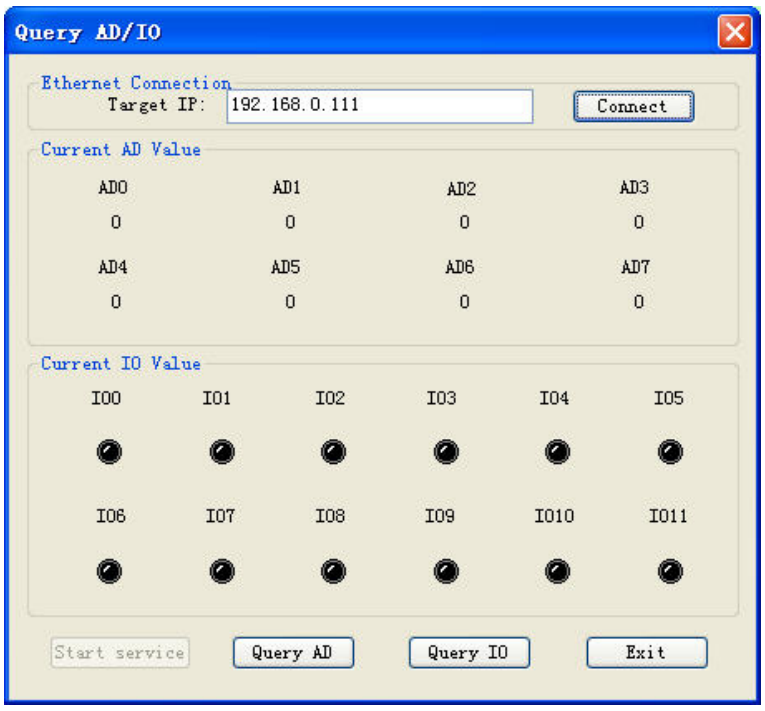


圖 2.72 “Query IO/AD” ——MultiFLEX™2-PXA270 控制器

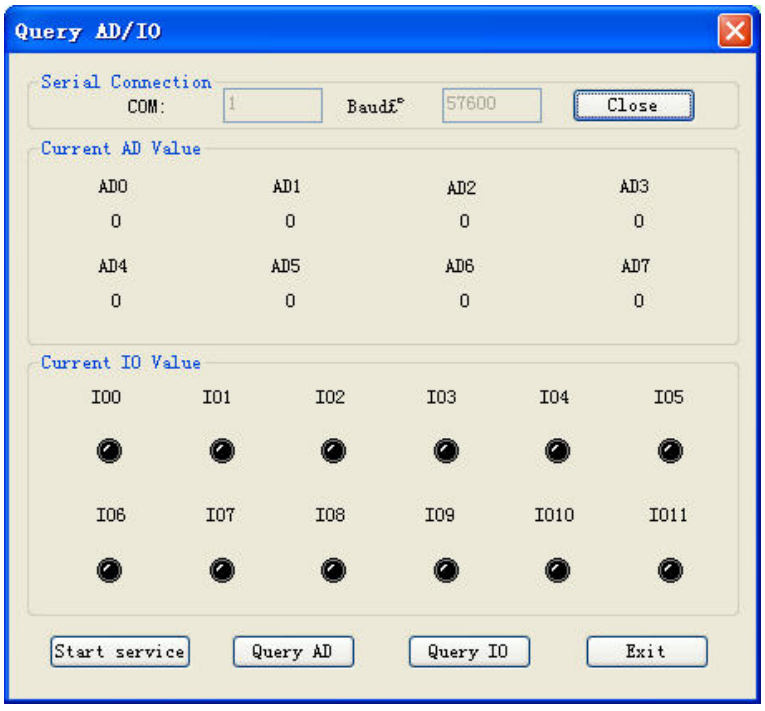


圖 2.73 “Query IO/AD” ——MultiFLEX™2-AVR 控制器



圖 2.74 “Test Servo”——MultiFLEX™2-PXA270 控制器

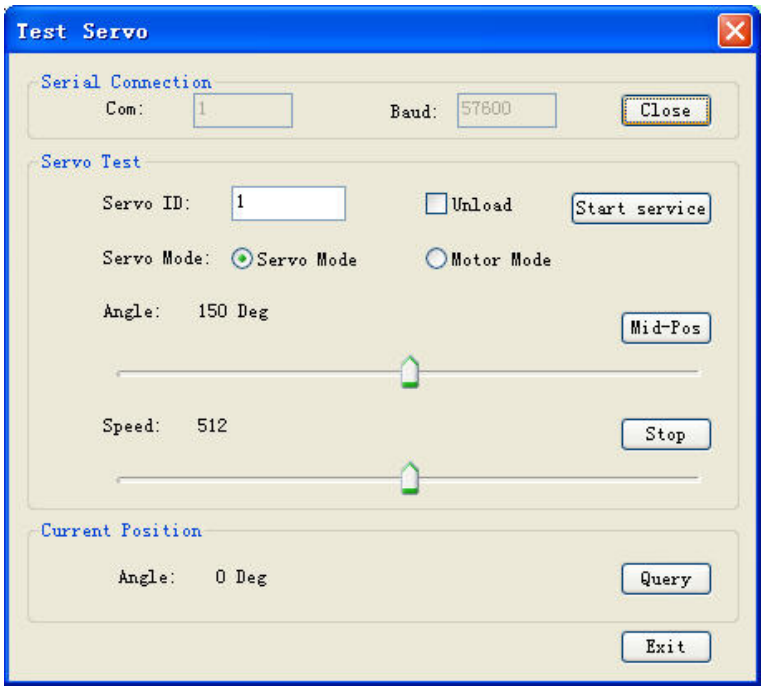


圖 2.75 “Test Servo”——MultiFLEX™2-AVR 控制器

流程圖調試時，從選單 Tools→Debug 調出“Debug”對話方塊，如圖 2.76 爲調試視窗。調試有兩種模式：單步（Step）和連續（Continue）。點擊開始，在“Step”模式下，可以不中斷點擊“Next”按鈕，軟體就會按照流程圖的邏輯從“start”模組一直執行到“end”模組後停止；在“Continue”模式下，“Next”按鈕處於禁用狀態，軟體會自動從“start”模組執行到“end”模組。在“Continue”模式下，可以點擊“Pause”來暫時停止調試，然後

可以點擊“Resume”恢復調試。點擊“Stop”可以終止調試。流程圖中當前執行的模組會處於灰白狀態。

在“Step”模式下，可以在點擊“Next”之前修改任意模組的屬性，然後點擊“Next”，將按照修改後的屬性執行；“Continue”模式下，流程圖不可編輯，只有停止調試後才可以編輯。

調試前，需要先和控制器建立連接——PXA270 需要建立網路連接，AVR 需要建立串列埠連接。

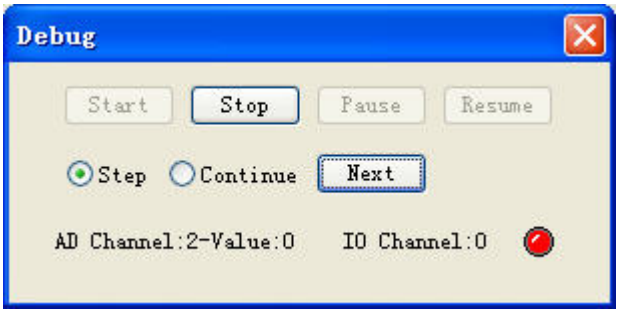


圖 2.76 調試流程圖

2.6 其他零件及光碟內容

2.6.1 UP-Debugger 下載調試器

UP-Debugger 多功能調試器整合了 USB 轉 232，半雙工非同步串列匯流排、AVRISP 下載器三種功能，在“創意之星”套件裏，我們可以用它來進行程式下載、串列埠通訊和 CDS5500 調試。

您可以從光碟 “……\InnSTAR 產品光碟\零件資料匯總\多功能調試器”找到《多功能調試器使用手冊》。調試器使用之前需要安裝驅動，使用過程需要注意選擇模式，這些在使用手冊裏都有詳細的說明。本節主要介紹如何用調試器進行控制器的程式燒錄和串列埠通訊。

如圖 2.77 所示調試器有兩種配線：

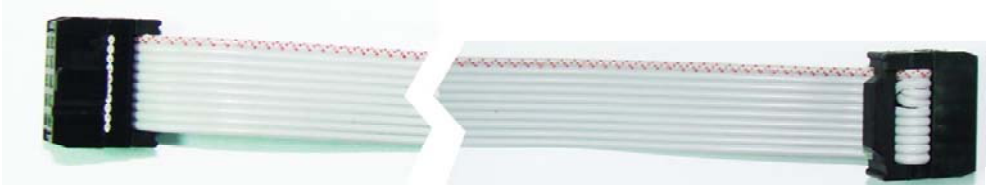




圖 2.77 調試器配線

第一條配線雙端都是 10 針 IDC 頭，是 MultiFLEX™2-AVR 控制器專用。第二條配線一端是 10 針 IDC 頭，另一端是 3 針和 5 針杜邦頭，是 MultiFLEX™2-PXA270 控制器專用。

- **給 MultiFLEX™2-AVR 控制器的 ATmega128 下載程式**

MultiFLEX™2-AVR 控制器和 MultiFLEX™2-PXA270 控制器的 AVR 控制卡有兩個處理器：主處理器 ATmega128 和輔助運算器 ATmega8。我們不建議您修改輔助運算器的程式，您可以透過 11 章熟悉控制器的具體架構。

1、 將調試器設爲 AVRISP 模式。

2、 把調試器的 USB 線接到 PC 端上，將 IDC 頭接到控制器的 IDC 座上，注意方向。

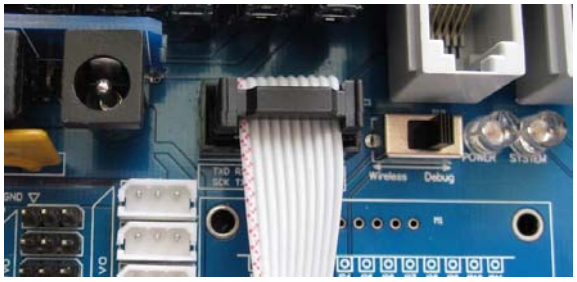


圖 2.78 ATmega128 下載接線

這時您可以使用 NorthSTAR 或者 AVR Studio 對控制器進行程式下載了。

- **給 MultiFLEX™2-AVR 控制器的 ATmega8 下載程式**

1、 將調試器設爲 AVRISP 模式。

2、 把調試器的 USB 線接到 PC 上，將 IDC 頭的 1、3、5、7、9 用插針接出（IDC 頭裏帶“△”的針腳爲 1 腳），如圖 2.79 ，9 爲 GND。

3、 控制器邊緣的 5 個插針孔是 ATmega8 的程式下載介面（側邊有 M8_ISP 標識）。將第 2 點接出的插針接入控制板邊緣的這 5 個插針孔，注意線的 GND 需和控制板上的 GND 對應，便可進行程式下載。



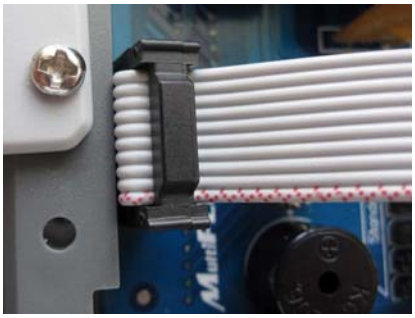


圖 2.79 ATmega8 下載接線

- **和 MultiFLEX™2-AVR 控制器進行串列埠通訊**

1、 將調試器設爲 RS-232 模式。

2、 把調試器的 USB 線接到 PC 上，將 IDC 頭接到控制器的 IDC 座上，注意方向。

3、 注意將 IDC 座旁邊的切換開關撥到 Debug 的位置上，這個切換開關用於切換有線或者無線通訊模式。

- **給 MultiFLEX™2-PXA270 控制器的 ATmega128 和 ATmega8 下載程式**

1、 將調試器設爲 AVRISP 模式。

2、 控制板上有 M128_ISP 和 M8_ISP，分別是 ATmega128 和 ATmega8 的程式下載介面。

3、 把調試器的 USB 線接到 PC 上，將 5 針杜邦頭接到控制器左側的 M128_ISP 或 M8_ISP 插針上，杜邦頭有“△”號的爲 GND 針腳，注意線的 GND 針腳和控制板的 GND 針腳對應。

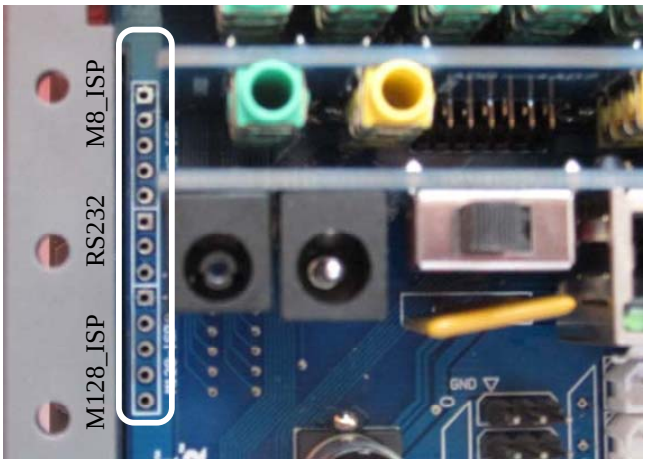


圖 2.80 下載和調試介面

- **和 MultiFLEX™2-PXA270 控制器進行串列埠通訊**

1、 將調試器設爲 RS-232 模式。

2、 把調試器的 USB 線接到 PC 上，將 3 針杜邦頭接到控制器左側的 RS-232 插針上，杜邦頭有“△”號的爲 GND 針腳，注意線的 GND 針腳和控制板的 GND 針腳對應。

2.6.2 PXA270 WiFi 無線網路卡或乙太網路線

“創意之星”高級版可以選配 WiFi 無線網路卡，借助無線網路卡您可以實現對機器人

的遠端操作。機器人之間可以進行通訊，完成群體協作的任務。光碟  “……\InnSTAR產品光碟\結構資料\結構視頻”裏的“主從機械手臂”就是無線網卡簡單運用的示範例子，主機械手臂可以人爲手動改變姿態，從機械手臂透過無線乙太網路和主機械手臂通訊，獲得主機械手臂的姿態資訊，並控制各個關節的伺服馬達，實現和主機械臂保持同步的姿態變化。這樣演示例子是遙控操作機器人的縮微版，我們甚至可以用此項技術進行排爆機器人的排爆機械手臂控制，實現良好的操作功能。

MultiFLEX™ 2-PXA270 控制器上使用的無線網卡模組型號是 VNT6656G6A40，VNT6656G6A40 是專門爲實現嵌入式系統的無線局域網通信應用而設計的一款 USB 介面的 WIFI 模組。使用該模組能夠讓使用者透過嵌入式硬體的 USB host 介面高效快捷的擴充出無線局域網路的通信功能。該模組的局域網路功能是由 VIA 的 VT6656 MAC/BB 零件實現，收發功能由 Aroha 實現，模組還內置高增益放大器，完全實現 IEEE802.11b/g 標準，最高通信速率可達 54Mbps。

使用方法如下：

1、 用網路線連接控制器和 PC，打開控制器，稍等片刻控制器啓動完畢。打開 NorthSTAR，新建一個基於 MultiFLEX™2-PXA270 控制器的程式。

2、 透過網路配置功能配置 MultiFLEX™2-PXA270 控制器的 IP，建立無線網路連接。選擇“設置”→“控制器網路配置”如圖 2.81 所示，控制器的預設 IP 爲 192.168.0.111（有線網 IP），點擊獲取狀態，稍等片刻，如果視窗中所有禁用控制項全部可用，就表示連接成功，此時可以透過“乙太網口 IP”設置控制器的有線網 IP，或者透過“無線網卡 IP”設置無線網 IP。

3、 如果要透過無線網路來控制控制器，需要先配置好 PC 的無線網路 IP。PC 的 IP 應設置爲 192.168.1.*，*應該在 0—255 之間，子網路遮罩 255.255.255.0，其他的可以不設置，注意 IP 不要和控制器重複。

4、 在控制器上新建一個網路連接。點擊圖 2.81 的“新建連接”，輸入網路名稱(SSID)後點擊確定。視窗會自動更新，列表中會出現剛才輸入的網路名稱，選擇後點擊連接（注意：這裏可以從 PC 的無線網路列表中選擇剛才輸入的網路名稱然後連接，同樣可以建立連接，某些時候透過 PC 連接更方便）。稍等片刻，列表中會提示已連接，此時 PC 狀態欄裏的無線網路圖示狀態會顯示連接成功。

5、 如果想透過無線網路訪問控制器，可以在通訊設置區域的“目標 IP”中輸入控制器無線網路 IP，然後點擊獲取狀態。如果無線網路連接成功，就可以獲取到和前面相同的資訊。此時可以拔掉網路線，透過無線網路調試、下載程式了。



圖 2.81 控制器網路配置介面

2.6.3 ZigBee 無線通訊模組

MultiFLEX™2-AVR 控制器可以使用無線通訊模組（不是無線網路卡），借此您可以實現對機器人的無線遠端遙控，也可以讓機器人組網實現多機器人協同任務。

RFM-ZGB1 是“創意之星”機器人套件的無線通訊模組，由一個可安裝到 MultiFLEX™ 2-AVR 控制器上的無線模組和一個接到 PC 上的無線接收器組成，基於 zigbee 通訊網路，通訊速率可以高達 57600。

一般而言，無線模組需要成對使用，相互通訊的無線設備必須保證串列傳輸速率、通道等參數一致。如圖 2.82 所示，控制器上的無線模組透過 2 組單排插針座安裝在介面板上，PC 端的無線接收器透過 USB 介面接到 PC 上。

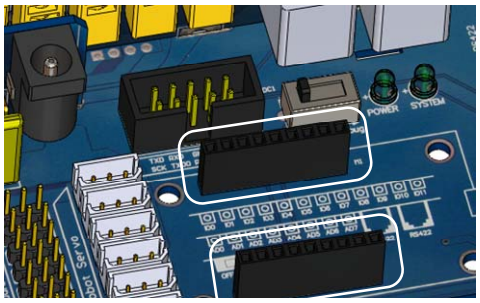


圖 2.81 控制器網路配置介面

在使用無線通訊模組時，注意將“無線與調試器”切換開關切換到 wireless 的位置上。

您可以透過以下方法測試無線通訊模組工作狀態：

打開 NorthSTAR，新建一個基於 MultiFLEX™2-AVR 控制器的空程式，選擇功能表“工具”→“感測器查詢”，將“埠號”改爲當前接收器使用的埠號，如 COM3。點擊“啓動服務”→“查詢 AD”/“查詢 IO”，您能夠得到當前控制器上的 AD、IO 值。

從光碟  “……\InnSTAR 產品光碟\零件資料匯總”資料夾下，您可找到《zigbee 無線通訊模組使用手冊》，根據本手冊您可以設置無線設備的串列傳輸速率和其他的配置參數。

2.6.4 光碟內容介紹

我們爲“創意之星”的教學準備了豐富的配套資料，您可以在配套光碟裏找到有關於機器人各種圖片、視頻、文字資料。

爲了您的查閱方便，光碟資料大致爲您介紹如下：

1、3D 模型：創意之星的所有零件的 3D 模型，爲 IGS 格式，您可以使用任意一種三維機械設計軟體導入這些模型，進行虛擬裝配和模擬。

2、零件資料匯總：包含 CDS5500、無線網卡、無線通訊模組、多功能調試器等各種零件的使用說明、驅動程式、配套軟體等。

3、結構資料：“創意之星”的典型結構的圖片、視頻資料。

4、控制器資料匯總：彙集了控制器的所有資料。如 PXA270 控制原始碼、AVR 控制原始碼、控制器的電路圖、編程環境的安裝檔、控制器相關資料手冊等。

5、實驗指導書：主要有實驗指導書的電子版、各章節範例的原始碼、各章節所涉及的視頻和圖片資料等。

6、NorthSTAR 圖形化整合開發環境：NorthSTAR 安裝檔。

第三章 熟練使用“創意之星”

3.1 MultiFLEX™2 控制器

“創意之星”標準版和入門版使用的是 MultiFLEX™2-AVR 控制器。

“創意之星”高級版使用的是 MultiFLEX™2-PXA270 控制器。

不同版本的控制器，使用的是不同的 CPU。因此，在進行開發時，我們推薦使用 NorthSTAR 軟體。NorthSTAR 具有面對不同控制器，僅在硬體連接及準備工作上存在差異，在軟體程式的開發階段，所有的操作都是一樣的。在完成程式流程的設計之後，僅需要透過指定目標控制器的型號，即可將程式編譯下載到該型號的控制器上執行。

NorthSTAR 圖形化整合開發環境同時還支援 C 語言編程方式，對於部分對控制精度有嚴格要求的用戶，我們推薦使用 NorthSTAR 的 C 語言模式進行開發。

3.2 第一次使用 MultiFLEX™2 控制器

1、準備工作（針對 MultiFLEX™2-AVR 控制器）

MultiFLEX™2-AVR 控制器可使用 NorthSTAR 圖形化整合開發環境進行編程，程式的下載需借助“創意之星”配套的多功能調試器。在進行開發之前，我們需要進行一些準備工作。

- 多功能調試器

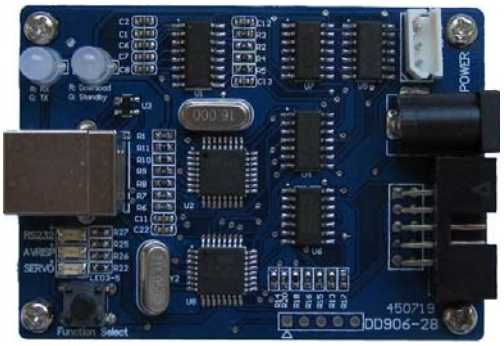


圖 3.1 多功能調試器

多功能調試器是創意之星 MultiFLEX™2-AVR 控制器的主要下載、調試工具。如果您還不熟悉多功能調試器的使用，請回顧 2.6.1 小節的內容。將多功能調試器的模式設爲 AVRISP 模式，至此，調試器準備工作完成。

- 電源

您可以使用控制器內置電池對控制器進行供電，如果控制器電池電壓不足，請準備配套

的直流穩壓電源。

- 準備控制器

取出控制器，將保護控制器開關的橡皮筋取下，注意保存好這個橡皮筋，在使用完控制器後請用橡皮筋將控制器開關原樣繫好。這個保護措施可以避免控制器的電源被誤打開，使電池過度放電而損壞。

將調試器的 IDC 頭接到控制器的調試通訊介面上。調試通訊介面爲 10 針 IDC 頭，具有防插反設計，如果插不進去，請檢查一下方向。

至此，如圖 3.2 準備工作完成。

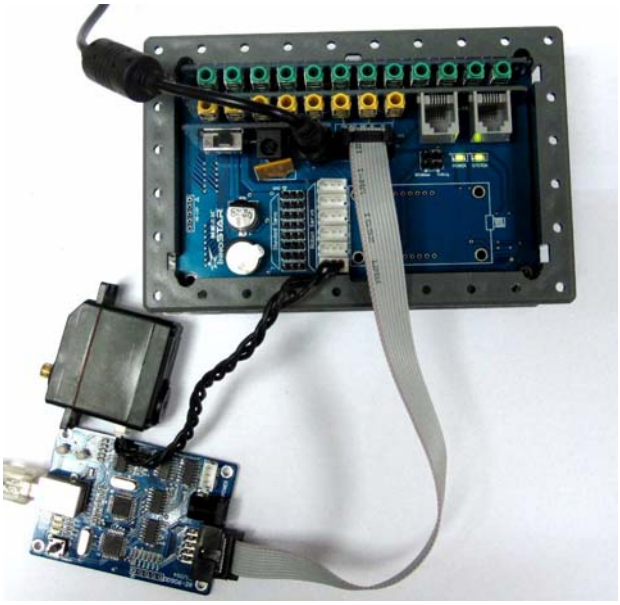


圖 3.2 控制器與多功能調試器的連接

2、準備工作（針對PXA270 MultiFLEX™2-PXA270 控制器）

- 乙太網線

乙太網路線是創意之星 MultiFLEX™2-PXA270 控制器的主要下載工具。在程式下載前，請將控制器與 PC 透過乙太網路線連接起來。並爲 PC 設置 IP，使其與控制器在一個 IP 段內以便進行通信，設置的方法：將“本地連接”的 IP 手動設置爲 192.168.0.xxx（xxx 爲 2-254 之間的任何值，111 除外，因爲 111 是 MultiFLEX™2-PXA270 控制器的 IP，強行設置會出現 IP 衝突），如圖 2.18 所示。

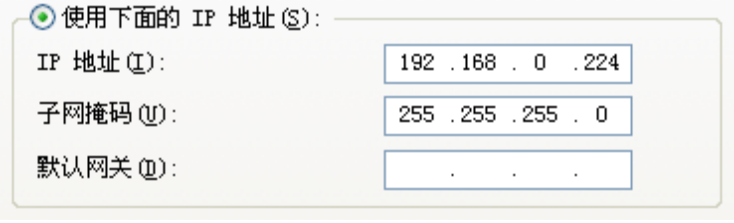


圖 3.3 本地連接的 IP 設置

- 準備控制器

確認控制器的網路線和電源連接好之後，打開控制器開關，等待大概 30 秒，讓控制器引導內置的 Linux 系統。

打開 NorthSTAR，透過功能表打開安裝目錄下的 DEMO 資料夾，隨便打開一個範例副程式，會出現原來隱藏的一些功能表。如圖 3.4，點擊“setting”（設置）→“Ethernet Connection”（網路連接）。出現網路測試視窗，點擊“connect”，稍等片刻如果連接正常，在視窗的下方會出現“Succeeded to connect”（網路連接成功）的提示。如果連接不正常會出現“connect to server failed”的提示框，這時請您檢查 PC 的 IP 設置是否正確、網路線是否接觸良好。

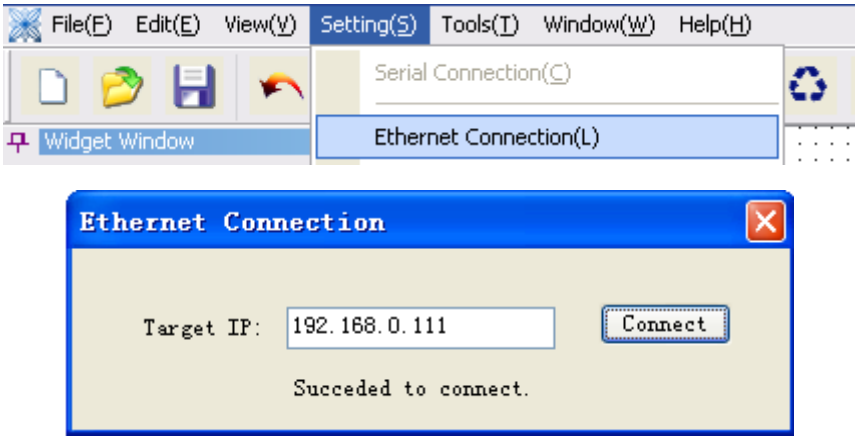


圖 3.4 測試網路通訊

3、建立專案

如果您的 PC 上還沒有安裝 NorthSTAR 圖形化整合開發環境，請從光碟 “……\InnSTAR 產品光碟\NorthSTAR 圖形化整合開發環境”裏找到安裝檔，並完成軟體的安裝（這部分您可以先查閱 2.5.1 小節）。

執行 NorthSTAR 圖形化整合開發環境，點擊“新建專案”圖示 ，在跳出專案配置頁面中根據需要進行程式開發的控制器型號配置。在“Select Controller”欄中，“創意之星”標準版和入門版請選擇“MultiFLEX2-AVR”，“創意之星”高級版請選擇“MultiFLEX2-PXA270”。因爲這裏我們還沒有具體對應的機器人結構，所以“Select Robot”中選擇“Customized”。點擊“Next”按鈕，進入下一頁面。

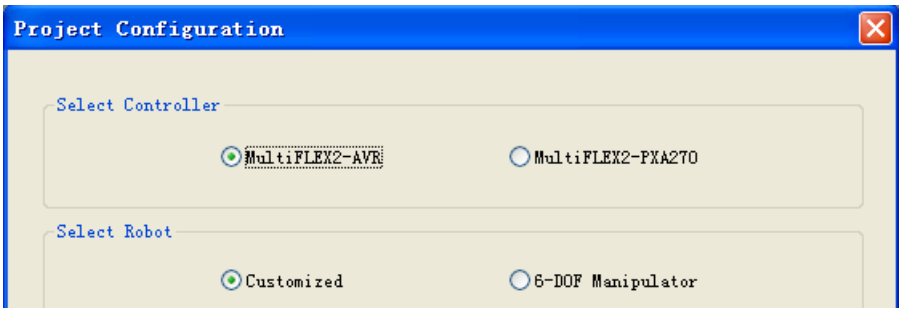


圖 3.5 專案配置

在這個簡單例子中，我們不使用任何感測器和伺服馬達，所以在所有配置頁中都使用預設值，不停點擊“下一步”按鈕，直至專案建立完成。

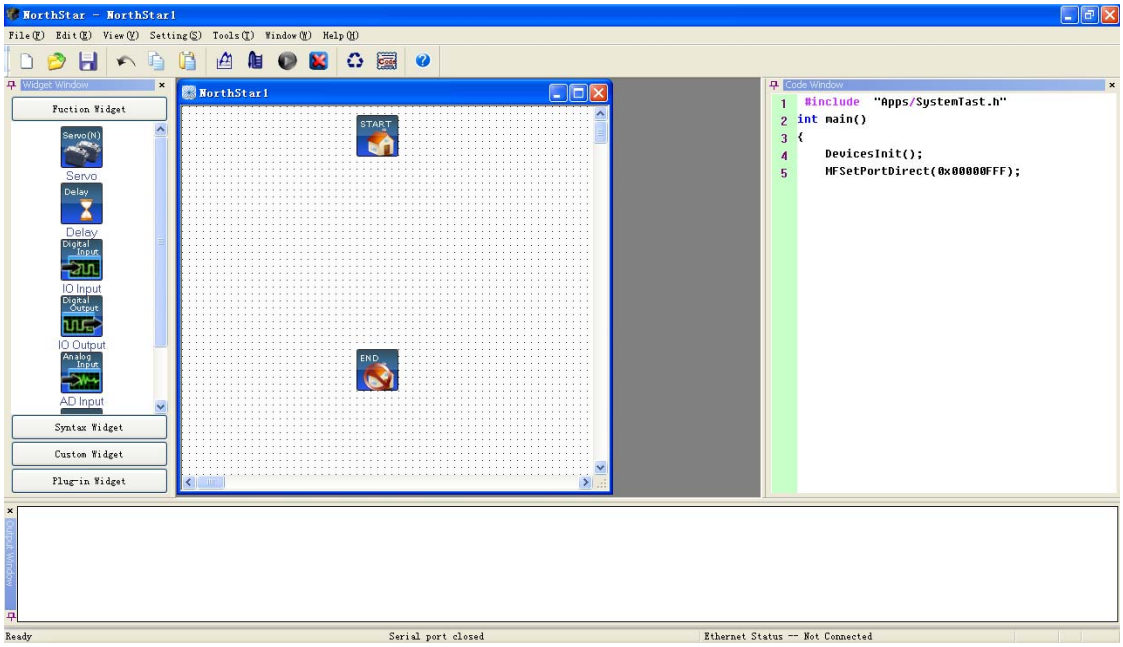


圖 3.6 專案建立完成

專案建立完成之後，我們可以在流程圖視圖中將“START”與“END”控制項連結，完成最簡單的一個程式，這就是我們的第一個創意之星程式。

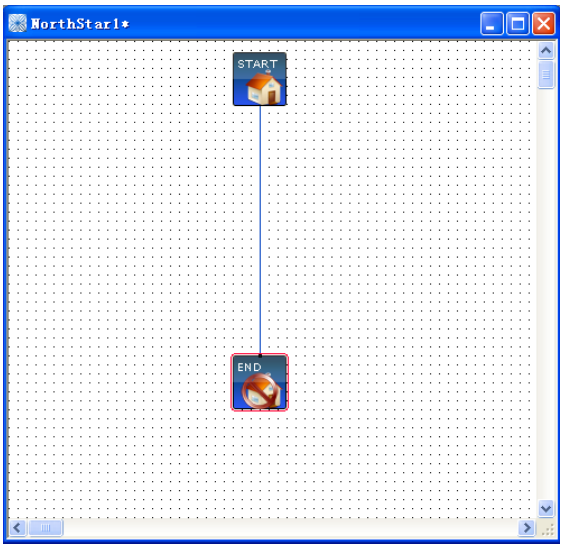


圖 3.7 簡單程式例子

4、編譯下載

點擊工具欄的“編譯”  ，等待程式編譯結束。若未出現錯誤提示，點擊“下載”  將編譯結束後的程式下載到控制器上。確認一切無誤之後，點擊“執行”  啓動程式。

因爲是個空程式，所以程式啓動之後沒有任何反應。

5、總結

透過這個例子，相信您已經瞭解了透過 NorthSTAR 對 MultiFLEX™2 控制器進行開發的過程。我們在 2.5 小節將會仔細講解 NorthSTAR 的使用方法，關於 NorthSTAR 的詳細使用您還可以查閱軟體配套的使用說明，您在 NorthSTAR 的幫助功能表裏可以找到它。

3.3 使用 MultiFLEX™2 控制器的 IO 介面

1、準備工作

我們準備了一個實驗來幫助大家學習控制器 IO 介面的操作。在這個實驗裏，我們將用一個觸碰開關去控制一個 LED 的亮滅，完成一個根據 IO 輸入來控制 IO 輸出的功能。

我們將一個 LED 連接到 IO0，再將一個觸碰開關連接到 IO1。控制器連接電源和下載設備。

2、建立專案

新建一個專案，在“Select Controller”欄中，根據目標控制器進行選擇：“創意之星”標準版和入門版請選擇“MultiFLEX2-AVR”，“創意之星”高級版請選擇“MultiFLEX2-PXA270”。進入 IO 屬性頁，IO 通道的數量改爲“2”，在列表框中 channel-0 的 Mode 改爲“Output”，而 channel-1 的 Mode 保持預設的“Input”。其餘配置選項均使用預設值。

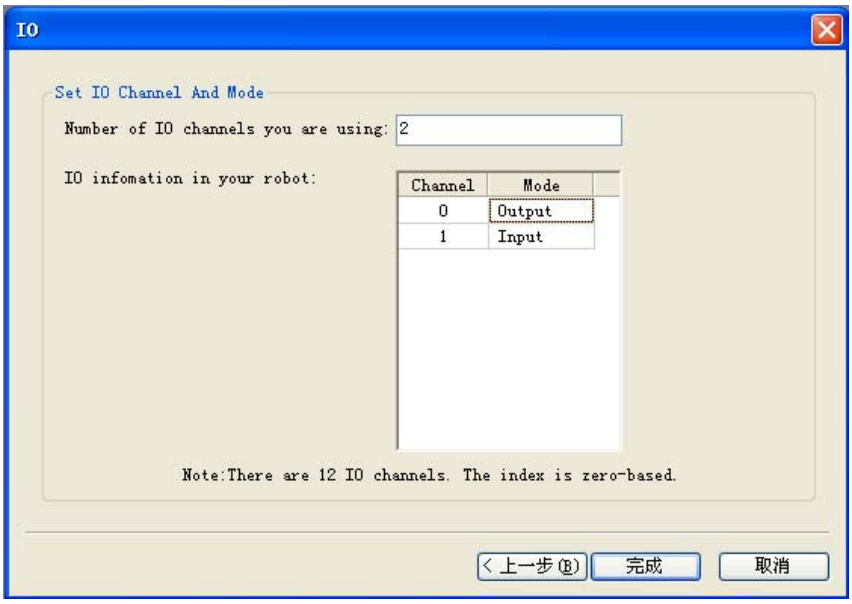


圖 3.8 IO 配置

專案建立完備之後，先進行初始化，從控制項列表 Function Widget 中拖曳出一個 IO Output 到流程圖，雙擊該控制項，跳出屬性頁，選擇 channel 0 ，Output 欄裏，將輸出值設爲 0，透過輸出低電壓點亮 LED，作爲初始化完成的信號。

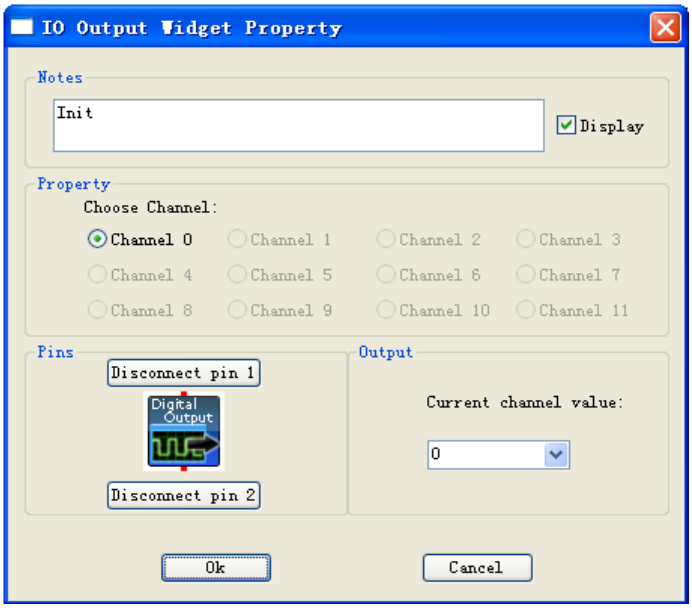


圖 3.9 IO 的屬性

在進行邏輯設計前，需要採取一些措施保證程式邏輯不停的執行，從控制項欄的 Syntax Widget 中拖曳出一個 While 控制項，會附帶產生一個 Loop-end 控制項。這兩個控制項之間的程式流程將會被不斷的迴圈執行，直到程式被強行中斷。

下面開始進行程式邏輯的設計，首先我們添加一個儲存 IO 輸入量的容器，先從控制項列表中拖曳出一個 Variable 控制項。雙擊該控制項跳出屬性頁，類型選擇“int”變數名稱定爲“SW_1”。

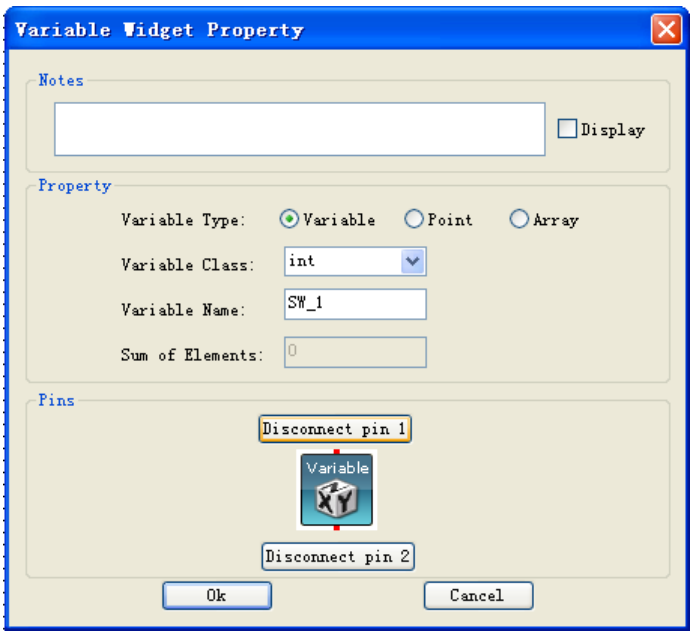


圖 3.10 變數屬性

然後，我們再拖入一個 IO Input 控制項，選擇“Channel 1”與觸碰開關建立對應關係，返回值選擇“SW_1”。

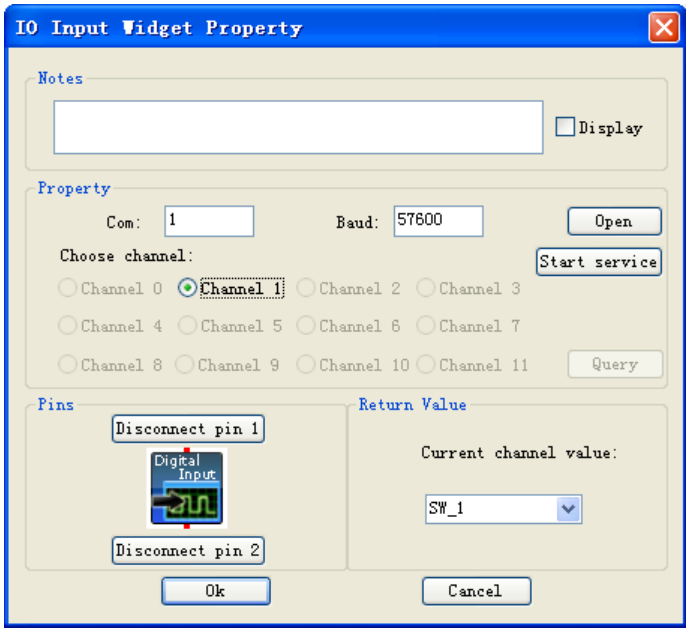


圖 3.11 IO 輸入屬性

接下來，我們需要對輸入值進行判斷，從控制項列表 Syntax Widget 中拖曳出一個 IF 控制項，建立一個判斷分支，雙擊拖曳出的 IF 控制項跳出屬性頁，Condition1 中填入判斷條件“SW_1 == 0”，即 IO 輸入爲低電壓，也就是觸碰開關被觸發的時候。

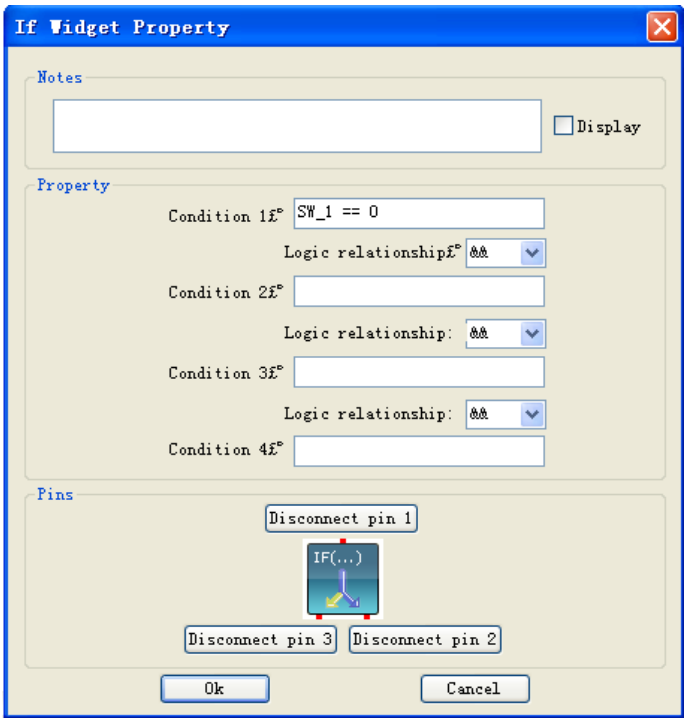


圖 3.12 條件判斷屬性

然後從控制項欄中拖曳出兩個 IO Output 控制項，分別連到 IF 控制項後面的兩個分支

中。If 控制項的左側接腳表示的是當 Condition 成立時的程式支流，右側則與左側相反。我們先雙擊左側分支的 IO Output 控制項，選擇“Channel 0”，Output 輸入“1”，即當觸碰開關被按下時，在 IO0 上輸出低電壓，滅掉 LED。接下來雙擊右側分支的 IO Output 控制項，選擇“Channel 0”，Output 輸入“0”，即當觸碰開關被放開時，再次點亮 LED。

用線將控制項按照我們的設計邏輯聯繫起來，一個簡單的 IO 控制程式就完成了。

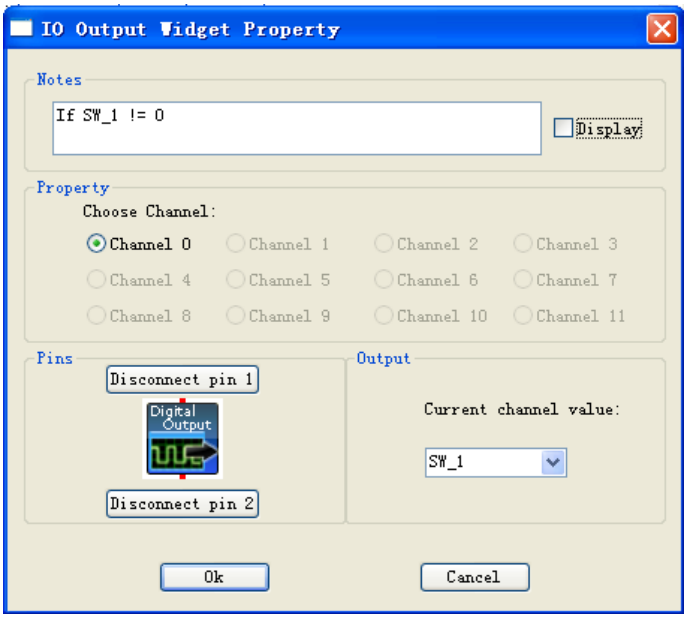


圖 3.13 IO 輸出屬性

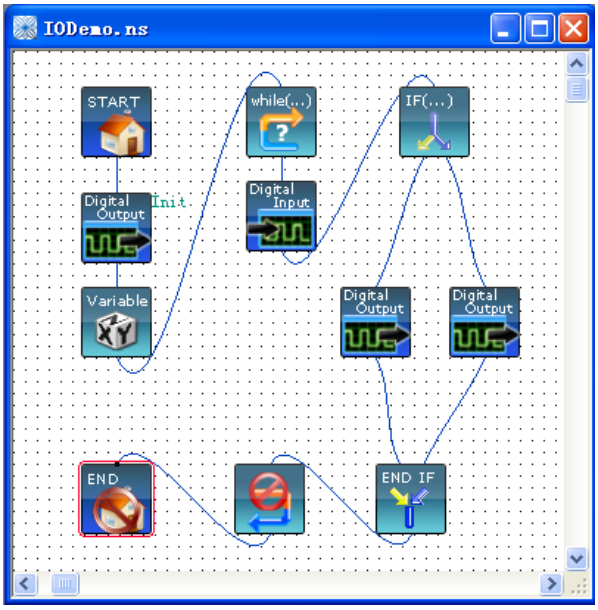


圖 3.14 控制程式

3、下載執行

點擊工具欄的“編譯”  ，等待程式編譯結束。若未出現錯誤提示，點擊“下載”  將編譯結束後的程式下載到控制器上。確認一切無誤之後，點擊“執行”  啓動程式。

我們會看到程式啓動後，連接在 IO-0 上的 LED 點亮了，然後我們按下 IO-1 上的觸碰開關，可以看到 LED 滅掉，鬆開觸碰開關，LED 再次亮起。

4、總結

在個實驗裏，我們透過 MultiFLEX2 控制器的 IO 埠獲取了一個開關量的輸入，同時輸出一個開關量控制 LED 的亮滅。本書此後的章節將會大量運用這種操作方式。

3.4 使用 MultiFLEX™2 控制器的 AD 介面

1、準備工作

我們準備了一個實驗來幫助大家學習控制器 AD 介面的操作。在這個實驗裏，我們將用一個紅外測距感測器控制一個數位伺服馬達的轉速。透過伺服馬達的轉動快慢來反映紅外探測器獲取的距離值大小。

這個實驗裏，我們需要從“創意之星”套裝中拿出一個紅外感測器連接到控制器的 AD0 介面，再拿出一個 CDS5500 伺服馬達，將其 ID 設置爲 1，連接到控制器的 Robot Servo 伺服馬達介面上。爲控制器連接上電源和多功能調試器。

2、建立專案

新建一個專案，在“Select Controller”欄中，根據目標控制器進行選擇：“創意之星”標準版和入門版請選擇“MultiFLEX2-AVR”，“創意之星”高級版請選擇“MultiFLEX2-PXA270”。Servo 數量改爲“1”，ID 爲“1”，Mode 爲“Servo Mode”。AD 數量改爲 1，Channel 設爲 0，其餘配置選項均使用預設值。

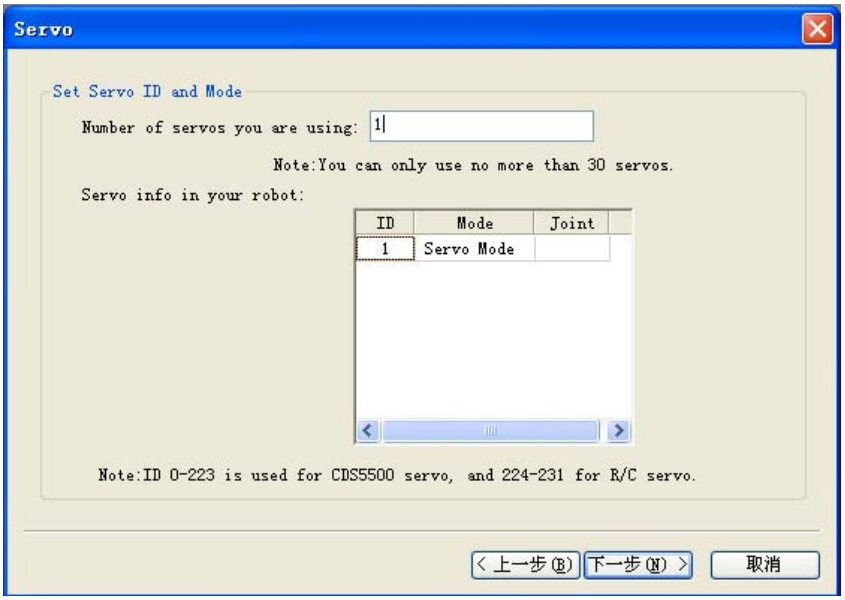


圖 3.15 Servo 屬性

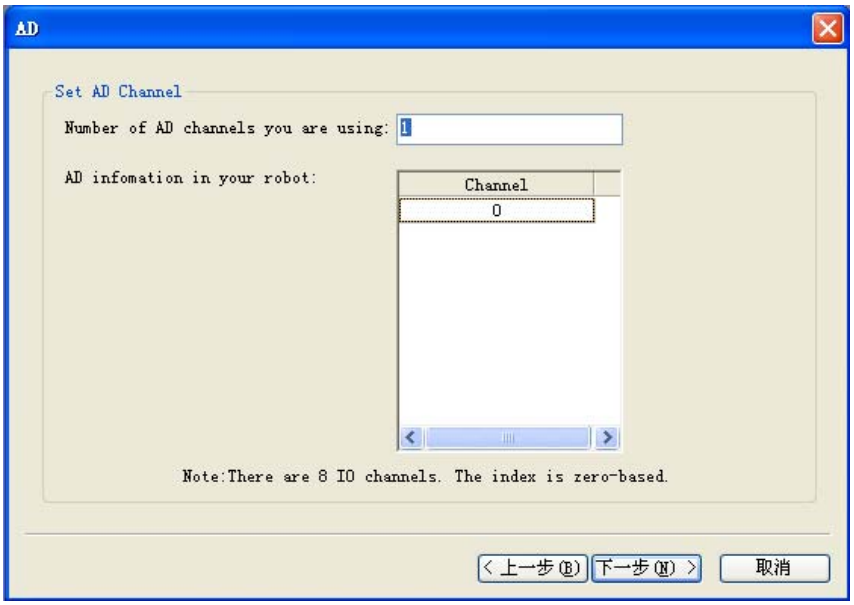


圖 3.16 AD 屬性

同上一小節實驗一樣，在進行邏輯設計前，需要採取一些措施保證程式邏輯不停的執行，從控制項欄的 Syntax Widget 中拖曳出一個 While 控制項，將其位置排列好，準備進行程式邏輯的搭建。

實驗中我們需要一個儲存 AD 輸入量的容器，從控制項列表中拖曳出一個 Variable 控制項。雙擊該控制項跳出屬性頁，類型選擇“int”變數名定爲“AD_0”。

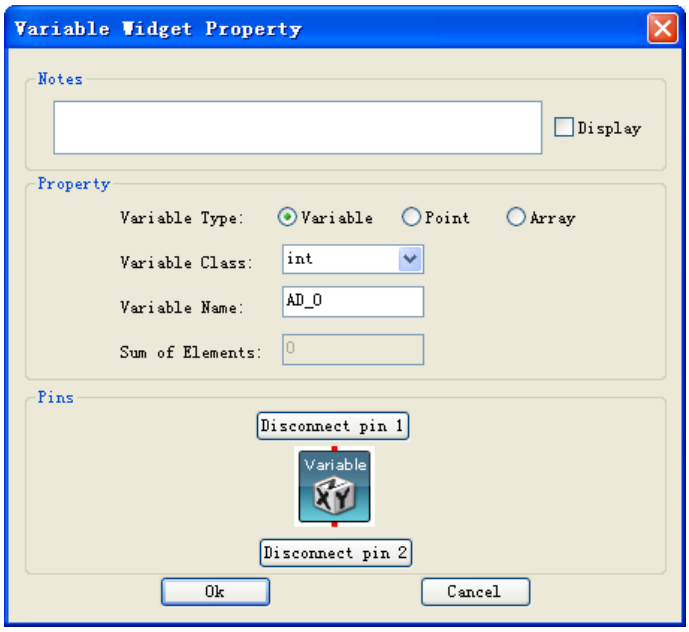


圖 3.17 添加變數

獲取 AD 輸入值，往流程圖裏拖入一個 AD 控制項，選擇“Channel 0”，返回值變數設爲“AD_0”。

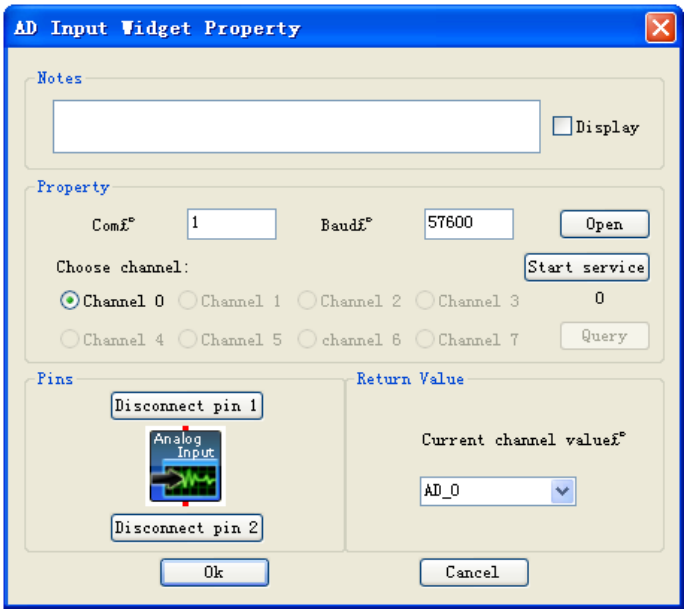


圖 3.18 AD 輸出屬性

爲了便於觀察，我們將一個伺服馬達的轉動角度與 AD 值建立對應關係，透過伺服馬達角度變化來反映 AD 值的增減。向流程圖拖入一個 Servo 控制項，雙擊控制項進入屬性頁，將 ID 爲 1 的伺服馬達 Angle 值設爲 AD_0。

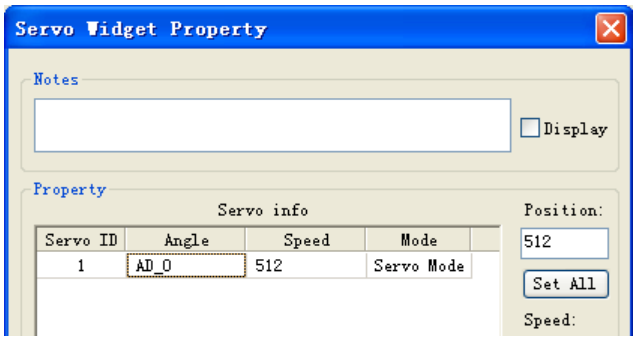


圖 3.19 Servo 設置

用線將控制項按照我們的設計邏輯聯繫起來，一個簡單的 AD 獲取程式就完成了。

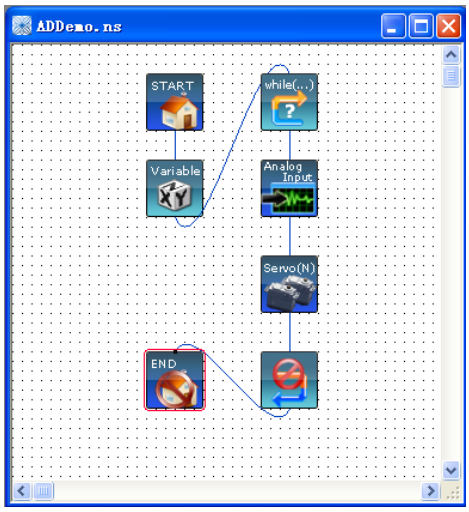


圖 3.20 程式流程

3、下載執行

點擊工具欄的“編譯”  ，等待程式編譯結束。若未出現錯誤提示，點擊“下載”  將編譯結束後的程式下載到控制器上。確認一切無誤之後，點擊“執行”  啓動程式。

程式啓動後，伺服馬達會輕微的轉動一個角度，如果未出現這個現象，可用手輕輕轉動伺服馬達的舵盤，如果舵盤角度位置鎖死，也是程式啓動的表現。接下來，將連接在 AD0 的紅外測距感測器對準一個平面，緩慢的拉遠或推近兩者的距離，觀察伺服馬達轉角是否跟隨變化。

4、總結

在這個實驗裏，我們透過 MultiFLEX2 控制器的 AD 埠獲取了一個類比量的輸入，同時輸出一個伺服馬達角度以觀察獲取值的大小變化。本書此後的章節將會大量使用到 AD 輸入的

感測器，請大家多熟悉本節內容。

3.5 使用 MultiFLEX™2 控制器的伺服馬達介面

1、準備工作

我們準備了一個實驗來幫助大家學習控制器對伺服馬達的控制。在這個實驗裏，我們需要用到兩個伺服馬達，讓其中一個伺服馬達工作在 Servo 模式，按一定的週期在兩個固定角度之間來回擺動；而另一個伺服馬達工作在 Motor 模式，按同樣的週期在正反兩個方向上來迴旋轉

從“創意之星”套裝中取出兩顆伺服馬達，ID 號分別設置爲 1 和 2，並將其連接到控制器面板的 Robot Servo 的任意插座上。控制器連接電源和下載設備。

2、建立專案

新建一個專案，在“Select Controller”欄中，根據目標控制器進行選擇，“創意之星”標準版和入門版請選擇“MultiFLEX2-AVR”，“創意之星”高級版請選擇“MultiFLEX2-PXA270”。Servo 數量改爲“2”，ID 1 的伺服馬達 Mode 設置爲“Servo Mode”，ID 2 的伺服馬達 Mode 設置爲“Motor Mode”。其餘配置選項均使用預設值。

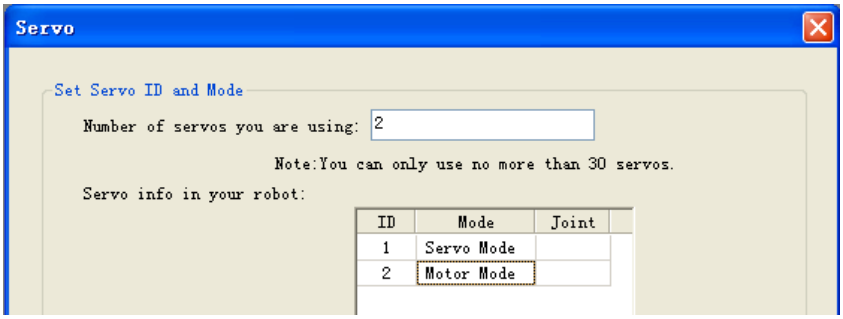


圖 3.21 Servo 屬性

在進行邏輯設計前，需要採取一些措施保證程式邏輯不停的執行，從控制項欄的 Syntax Widget 中拖曳出一個 While 控制項，將其位置排列好，準備進行程式邏輯的搭建。

向流程圖中拖曳一個 Servo 控制項，雙擊控制項進入屬性頁，將 ID 爲 1 的伺服馬達 Angle 值設爲 300；ID 爲 2 的伺服馬達 Speed 值改爲 300。可以看出，1 號伺服馬達我們控制的是角度，2 號伺服馬達我們控制的是轉動速度。

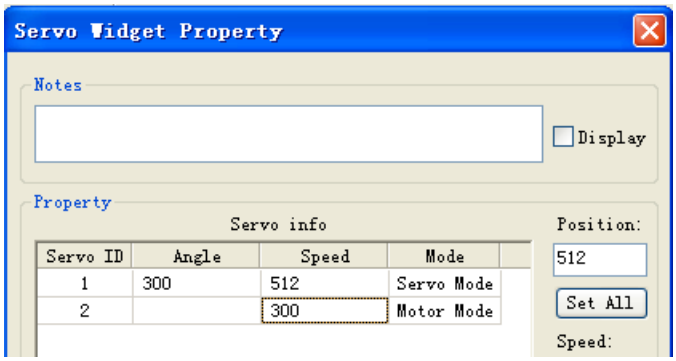


圖 3.22 第一個 Servo 動作

我們讓伺服馬達在 300 這個角度位置停留一段時間，所以需要加入一個延時停頓。向流程圖拖入一個 Delay 控制項，雙擊該控制項進入屬性頁，在延時長度中填入“3000”ms（毫秒）。

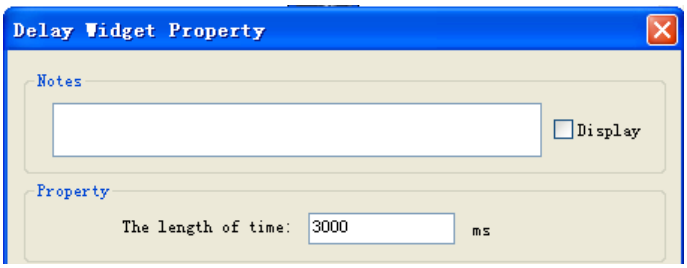


圖 3.23 停頓延時

下面我們讓伺服馬達轉到另一個角度，向流程圖中再拖入一個 Servo 控制項，雙擊進入屬性頁，將 ID 爲 1 的伺服馬達 Angle 值設爲 600；ID 爲 2 的伺服馬達 Speed 值設爲 600。

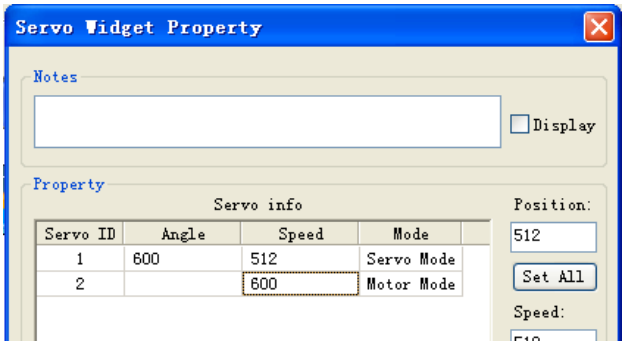


圖 3.24 第二個伺服馬達動作

在這之後再拖入一個 Delay 控制項，延遲時間設爲 3000ms。將這些控制項按照我們的設計邏輯連接起來，一個簡單的伺服馬達控制程式就完成了。

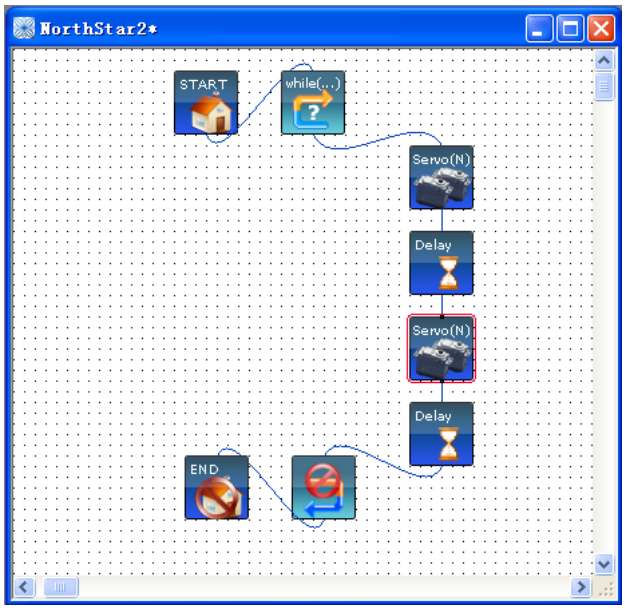


圖 3.25 程式流程

3、下載執行

點擊工具欄的“編譯”  ，等待程式編譯結束。若未出現錯誤提示，點擊“下載”  將編譯結束後的程式下載到控制器上。確認一切無誤之後，點擊“執行”  啓動程式。

程式啓動後，伺服馬達會先轉到一個角度（300），停留 3 秒之後轉到另一個角度（600），停留 3 秒之後再回到第一個角度位置，如此反復。

4、總結

在這個實驗裏，我們透過 MultiFLEX2 控制器的伺服馬達埠控制兩個伺服馬達工作在不同的模式之下，Servo 模式下可以控制伺服馬達轉動角度，Motor 模式下可以控制伺服馬達的旋轉速度和旋轉方向。Motor 模式與 Servo 模式的一點不同是：Motor 不能控制到位角度，但是能進行不限制的多周選擇，而 Servo 模式能控制旋轉角度但僅能旋轉 330 度左右。本書此後的章節將會大量運用這兩種模式。

第四章 有趣的防碰撞小車

4.1 任務 1：目標分析與計畫

機器人技術誕生是爲了服務於工業生產與人類的生活，所以一個機器人設計專案需要有其設計目標。在有了明確的功能需求以後，才能夠根據我們所掌握的技術尋求可行的設計方案。例如，如果我們要設計一台鐵路軌道狀態的巡檢機器人，這台機器人需要具有在鐵軌上平穩、快速行進的能力，所以其驅動方式就需要是輪式的。如果我們設計的是一台火山活動檢查機器人，這台機器人就需要具有能夠適應複雜地面環境、能夠承受高溫的能力，所以比較合適的驅動方式就是多足爬行。

我們在本章要設計一個簡單機器人系統——“防碰撞小車”。“防碰撞小車”是能夠在較平坦地面上以車輪驅動的方式行走的有自主避開前方障礙物能力的機器人。爲了實現這樣的功能，我們需要對“防碰撞小車”的組成結構進行規劃。

由於機器人系統是模仿人類等生物的結構、思維而構建的，所以我們也可以我們自身爲範本來設計機器人。人類躲避障礙物的一般流程是這樣的，用眼睛看到影像，用腦思考是否需要躲避障礙物，用腦控制肌肉做出運動，肌肉帶動骨骼完成回避動作。在這樣的流程中，人類用到的身體結構包括：眼、腦、肌肉、骨骼。用到的思維方式包括“是否需要躲避障礙物”和“控制肌肉做出運動”兩個部分，前者是邏輯判斷，後者是固定的行爲方法。

對於機器人，上述存在於我們自身的結構也可以在機械結構上近似的類比出來。“眼”替換爲“障礙感測器”，“腦”替換爲“控制器”，“肌肉”替換爲“伺服馬達”，“骨骼”替換爲“結構零件”。人類的思維在機器人上以軟體的形式類比，控制人類行爲方式的“邏輯思維”替換爲“邏輯判斷演算法”，人類對肌肉的協調控制，替換爲機器人對伺服馬達的協調控制。

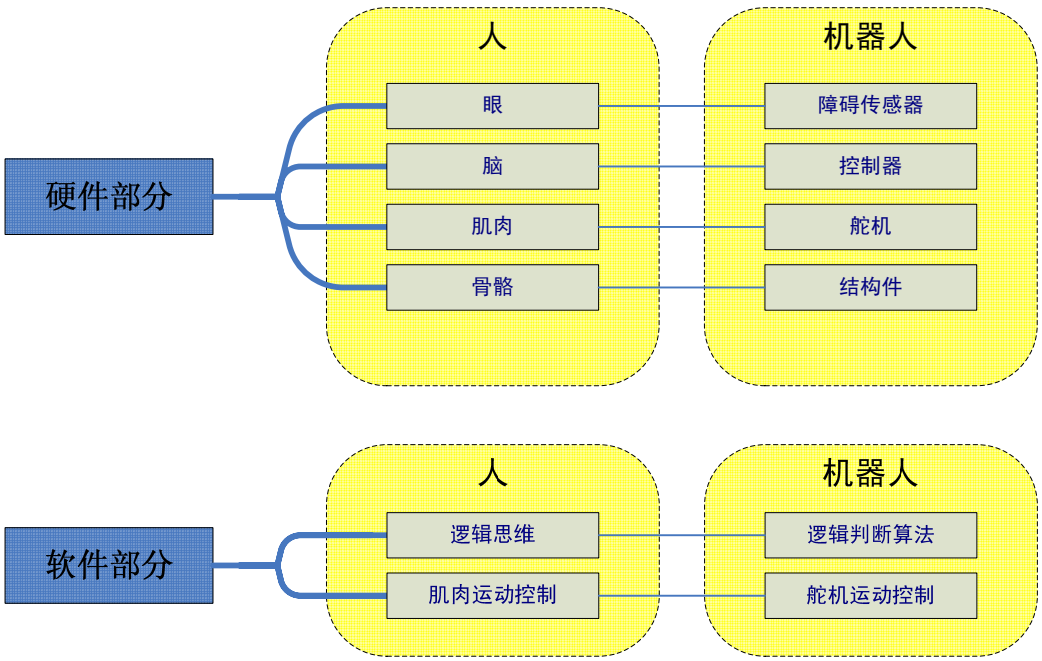


圖 4.1 機器人系統與人的比較方塊圖

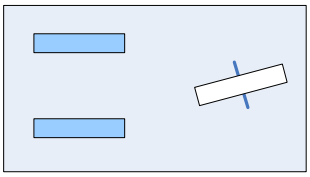
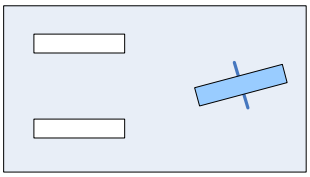
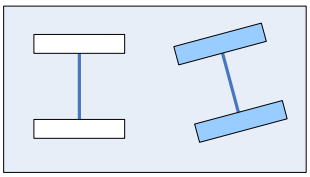
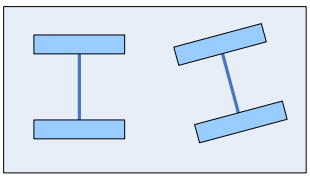
4.1.1 結構方案

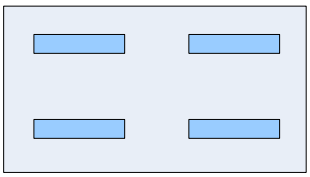
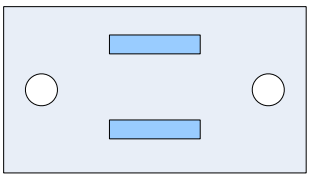
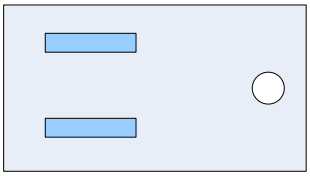
首先我們需要清點一下，我們能夠利用的物資有什麼，基於此制定的方案才有可行性。我們假定您是“創意之星”標準版的使用者（標準版的使用者最多），在標準版裏有：一個控制器、14 個 CDS5500、4 個輪子、眾多結構零件和感測器。

根據上面的清單，我們可以做一個輪式機器人。那麼，接下去我們需要上網研究一下，輪式機器人主要有哪些種類，各有什麼優缺點。當然，您也可以在學校的圖書館找到參考資料。

在我們現實生活中，我們能夠看到各種各樣的輪式車輛，他們是我們設計輪式機器人的參考。實際上，輪式機器人底盤就是把現實的輪式車輛進行縮微和簡化而已。除了摩托車、自行車，各種輪式車輛一般至少有 3 個輪子同時著地，這樣能夠讓車輛在任何時候都能保持平衡。同樣，在這裏我們也不應該選擇兩輪的結構，控制兩輪機器人保持平衡是很困難的，您在第八章能體會得到。

表 4.1 常見輪子底盤結構對照表

序號	結構類型	說明
1、		後面兩個獨立驅動的輪子，前面一個轉彎用無驅動輪子。
2、		後面兩個無動力輪子，前面一個有動力的轉向輪。
3、		前面兩輪驅動，後面兩輪從動，前後都帶差速器。
4		前後都是驅動輪，帶差速器。越野車的驅動方式。

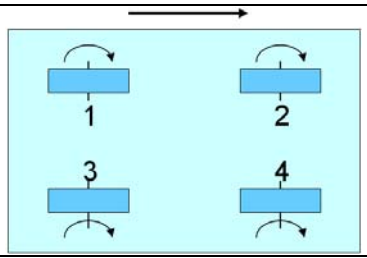
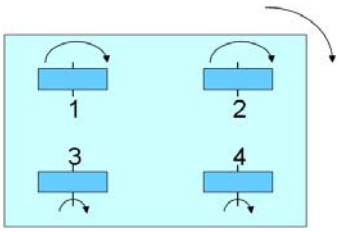
5		四輪獨立驅動，模型四驅車的驅動方式。
6		中間兩個獨立驅動的輪子，前後各有一個全向輪。
7		前面一個全向輪，後面兩個獨立驅動的輪子。

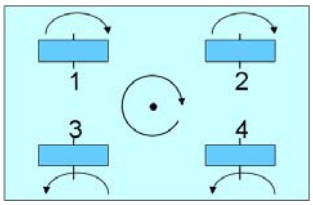
注：▭有動力的驅動輪，▭無動力的隨動輪，◯全向輪。

上表我們列出了 7 種常見的，輪式機器人底盤上經常用到的結構類型。由於“創意之星”標準版沒有全向輪和差速器，我們排除 3、4、6、7 結構。1、2 結構需要有比較大的轉彎半徑，5 結構的四個輪子可以單獨驅動控制，具有很強的機動能力，並且動力比較強勁，所以我們採用結構 5。

我們控制四個馬達，讓機器人能夠靈活運動，四個馬達轉動輸出和機器人平臺的方向對應關係如表 4.2 所示。之後我們需要根據機器人前方的障礙物情況，控制機器人的方向。

表 4.2 馬達與底盤的運動關係對照表

狀態	**驅動方式**	**說明**
前進和後退		給左右兩邊的輪子同樣向前或向後的馬達轉速，底盤就會前進或者後退。
有轉彎半徑的轉向		左邊的輪子給較高的轉速，右邊的輪子給較低的轉速，底盤就會向右前方轉向。

無轉彎半徑的轉向		左右兩個給一樣的速度，但不同方向的馬達轉動，底盤就會原地轉向。

至此，我們已經清楚我們需要什麼樣子的輪式小車來完成本次實驗。從《創意之星機器人套件組裝指南》的 5.3 小節，您可以看到這個輪式小車效果圖和搭建過程。

4.1.2 感知方案

避障小車顧名思義就是能夠知道前方障礙，並能夠自動繞開的小車。那麼它如何知道前方的障礙呢？透過前面的學習，您一定已經知道機器人是透過感測器來感知周圍環境這個基本知識點了。但是我們前面介紹了很多感測器，這些感測器在“避障小車”上如何運用，哪種最合適呢？我們透過表格 4.3 進行分析對比。

表 4.3 “創意之星”感測器特性對照表

感測器	特性	說明
紅外接近感測器	● 開關量感測器 ● 探測距離 20cm ● 可探測大多數介質表面	開關量感測器，接在控制器的 IO 輸入介面即可，使用方便。
霍爾接近感測器	● 開關量感測器 ● 探測距離 10mm ● 只對鐵磁性材料敏感	探測距離只有 10mm，沒有足夠的空間讓機器人轉向，避開障礙物，不可以使用。
紅外測距感測器	● 類比量感測器 ● 探測距離 10-80cm ● 可探測大多數介質表面	類比量感測器，測試距離合適，可以使用。
超音波測距感測器	● RS-422 感測器 ● 探測距離 4-500cm ● 可探測大多數介質表面	探測距離合適，可以使用。
碰撞感測器	● 開關量感測器 ● 觸碰才有反應	機器人需要碰到障礙物才有反應，容易碰壞機器人，不可以使用。

紅外接近感測器、紅外測距感測器、超音波測距感測器都是可以使用的。超音波測距感測器對與初學者比較有難度。紅外測距感測器的資料輸出頻率是 40ms，也就是我們每秒鐘只能得到 25 次距離資訊。紅外接近感測器的資料是開關量，只能知道 20cm 範圍之內有沒有障礙物，但是使用簡單，程式編寫簡單，感測器反應迅速。

從表格和上述分析，我們可以看出紅外接近感測器、紅外測距感測器、超音波測距感測器各有優缺點，都是可以選擇的。標準版“創意之星”沒有配備紅外測距感測器和超音波測距感測器，所以我們這裏採用紅外接近感測器作爲本實驗的機器人感知方案。

如下圖所示，我們在機器人的前方安裝兩個紅外接近感測器，如果左邊的感測器觸發，但右邊的沒有觸發，說明機器人左邊有障礙物，但右邊沒有，可以向右邊轉彎避開。同理可

以判斷前方障礙物、右邊障礙物等情況。

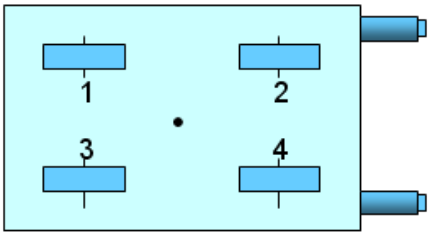


圖 4.2 感測器佈置簡圖

4.1.3 避障策略

對於這個機器人，我們有一個規劃目標：它能夠不斷的運動，並且不會碰到任何東西。我們已經搭建好了能夠靈活運動的機器人平臺，爲了滿足極限條件的要求，我們甚至要求這個平臺能夠零半徑轉彎。接下去，我們需要編程讓我們的機器人運動起來，沒有程式的機器人只是一塑膠、矽材料、金屬組合的死物，我們需要賦予它們靈魂。

這個程式應該是怎麼樣的呢？我們編寫的程式決定機器人的行爲模式，這個模式是不是有效呢？我們在編程的時候必須忘記自己是個程式師、是個學生或者是個老師，我們需要把自己當成我們現在搭建的機器人，以“第一視角”來考慮問題：

我，一個機器人。我只能看見前方 20cm 距離的東西，我們使命是在這片區域內遊蕩，這個過程中不能碰壞東西，我也不想碰壞自己。我先探尋我面前有沒有障礙物，如果沒有的話，我將筆直地往前走，直到遇見障礙物或者直到永遠。如果我前面有障礙物，我將判斷障礙物是在左邊還是在右邊，如果是左邊我就向右轉，然後繞開，如果是右邊就向左轉。如果是障礙物充滿了整個視野，擋住了整個前進的路線，我將向後退，然後左轉，重新探尋路線。

上述是我們以機器人的第一視角來考慮的避障策略，這不是一個完美的策略，我們最後一個任務會對此做分析。這個策略很簡單，但是個完整、可行的框架，更複雜的策略只是在這之上填寫更複雜的判斷而已。我們將這個策略用程式流程圖描述出來，作爲之後程式編寫的指導，如下圖所示：

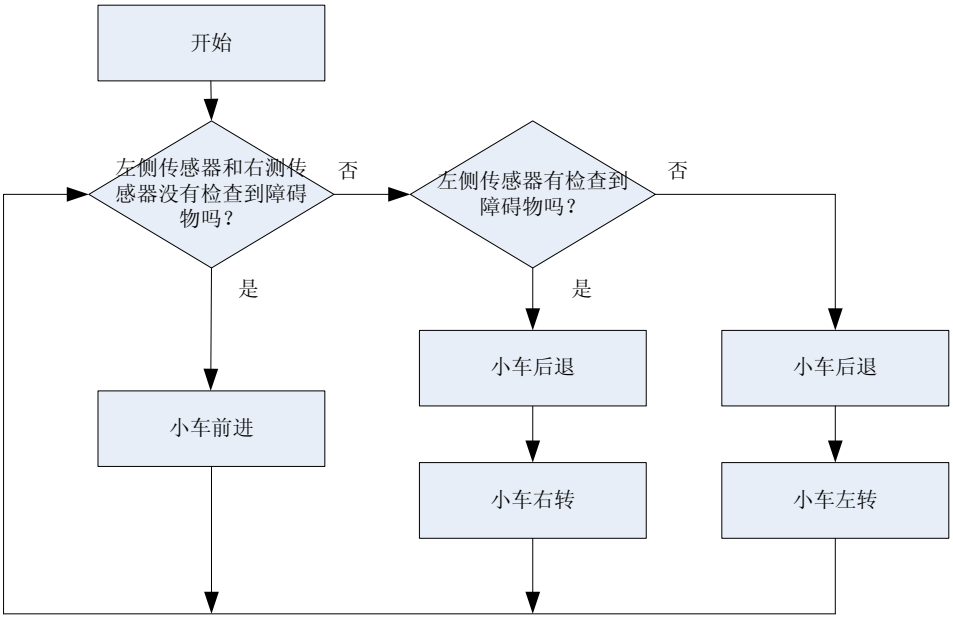


圖 4.3 避障策略程式流程圖

至此，我們已經完成了“避障小車”從結構搭建到程式控制的所有方案的擬定。我們接下去需要做一個計畫來完成這個項目：

1、學會控制 CDS5500，這是整個專案的關鍵環節。

2、搭建小車，這個工作是基礎。您可能需要花上 1 節課的時間。

3、編程，將控制策略的流程圖用真正的程式語言實現，並燒錄到控制器上。讓控制器開始履行大腦的角色。

4、總結和分析。每做一個專案都需要進行總結和思考，這樣才能有所沉澱、有所進步。

4.2 任務 2：讓伺服馬達轉起來

控制 CDS5500 是整個項目的關鍵環節，我們先搭建一個輪子，控制這個輪子轉起來，用以熟悉 CDS5500 的控制過程。

從《創意之星機器人套件組裝指南》的 5.3 小節所準備的實驗材料中取出以下零件：

- 結構若干；
- 伺服馬達連接線若干；
- 編好 ID 的機器人伺服馬達 CDS5500 一個；
- MultiFLEX™2-AVR 控制器一個；

4.2.1 搭建輪子結構、接線

- 用結構零件和 CDS5500 伺服馬達搭建出下面的結構；



注意！

要先將連接零件組裝好再安裝馬達，不要漏掉連接零件中的螺母。

圖 4.4 基本伺服馬達單元

- 將 L 型連接零件與伺服馬達單元相連接，形成一個基本的輪子單元；

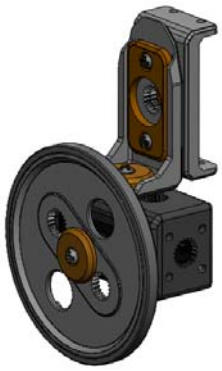


圖 4.5 輪子單元

最後，用伺服馬達連接線將伺服馬達與 MultiFLEX™2-AVR 控制器連接。

4.2.2 編寫/下載程式

- 打開 NorthSTAR 工作環境，在“Project Configuration”中選擇“MultiFLEX2-AVR”控制器、“Customized”自定義機器人。

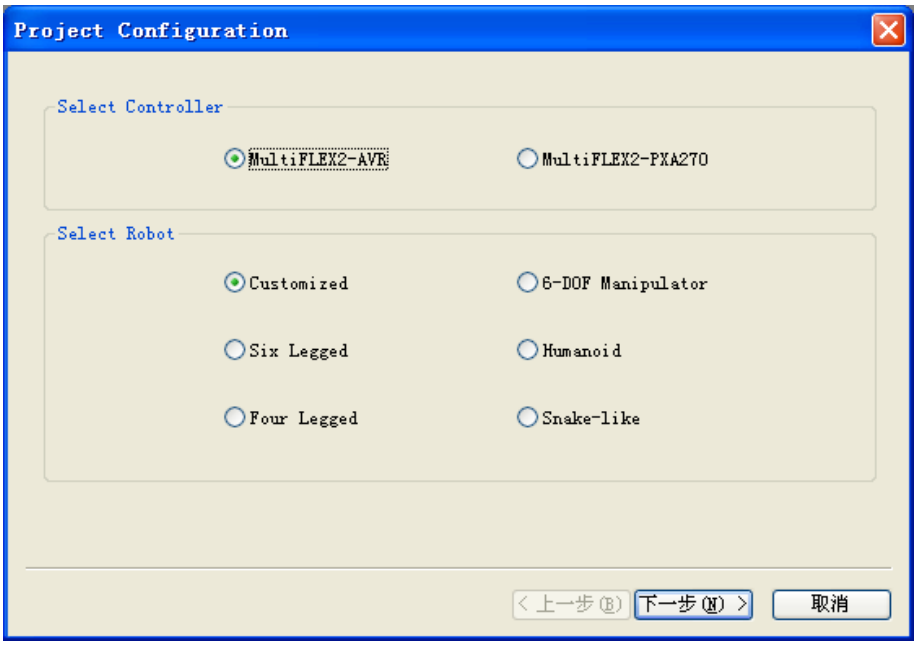


圖 4.6 專案設置視窗

- 單擊“下一步”設置伺服馬達，將“Number of servos you are using”設置爲“1”，表明我們使用了一個伺服馬達。按照選用的伺服馬達輸入“ID”值，將模式設爲“Moto Mode”。

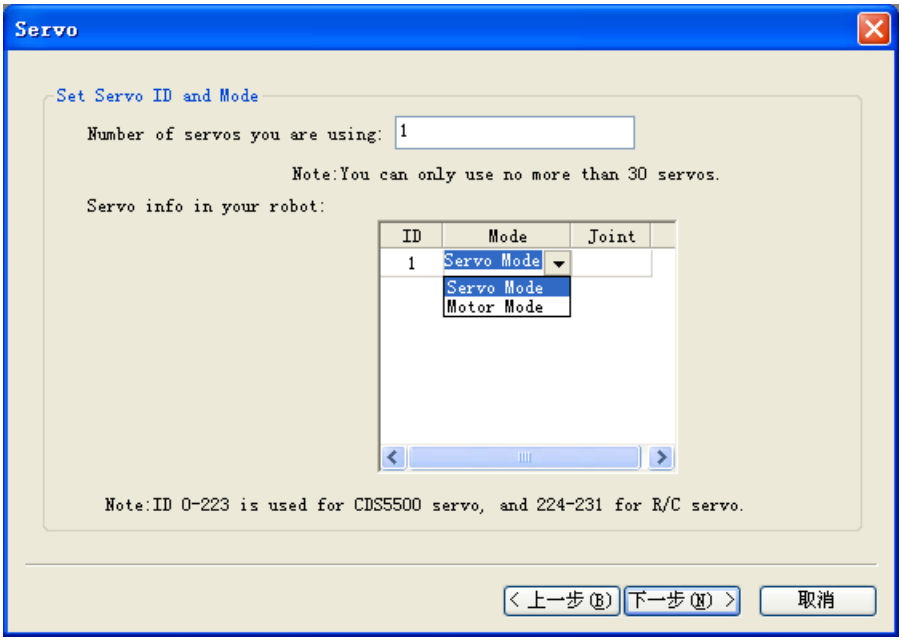


圖 4.7 伺服馬達設置視窗

- 連續單擊下一步直至設置完成。
- 從 view 功能表調出控制項視窗 Widget Window，選擇“Fuction Widget”，將其中的“Servo”控制項用滑鼠拖入工作區。

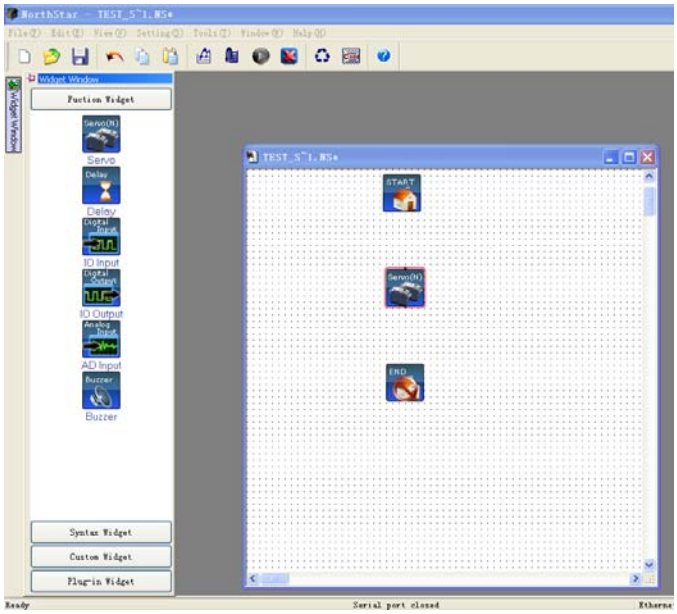


圖 4.8 編程視窗

- 雙擊工作區中“Servo”控制項，跳出如下視窗。

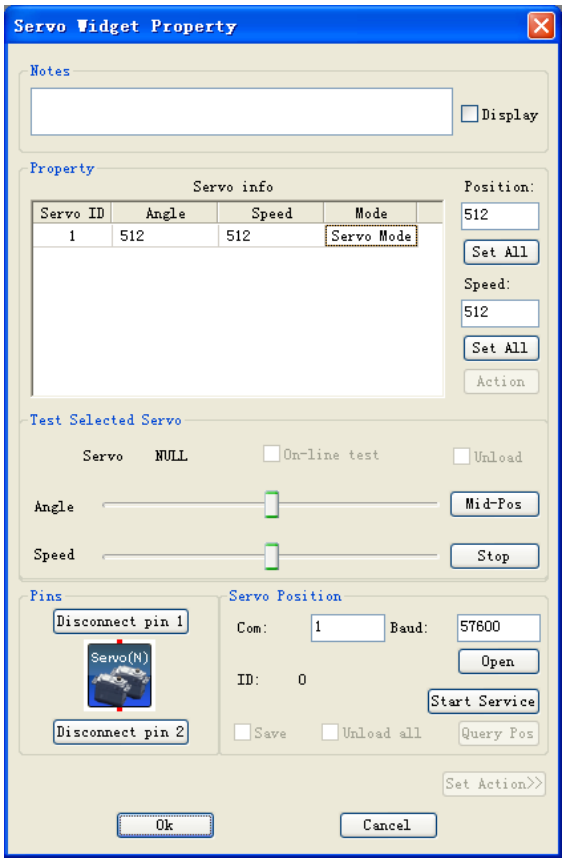


圖 4.9 伺服馬達屬性頁面

- 將“多功能調試器”與 MultiFLEX™2-AVR 控制器用線纜連接，讓“多功能調試器”

處於 ISP 下載模式，單擊“Start　Service”按鍵。

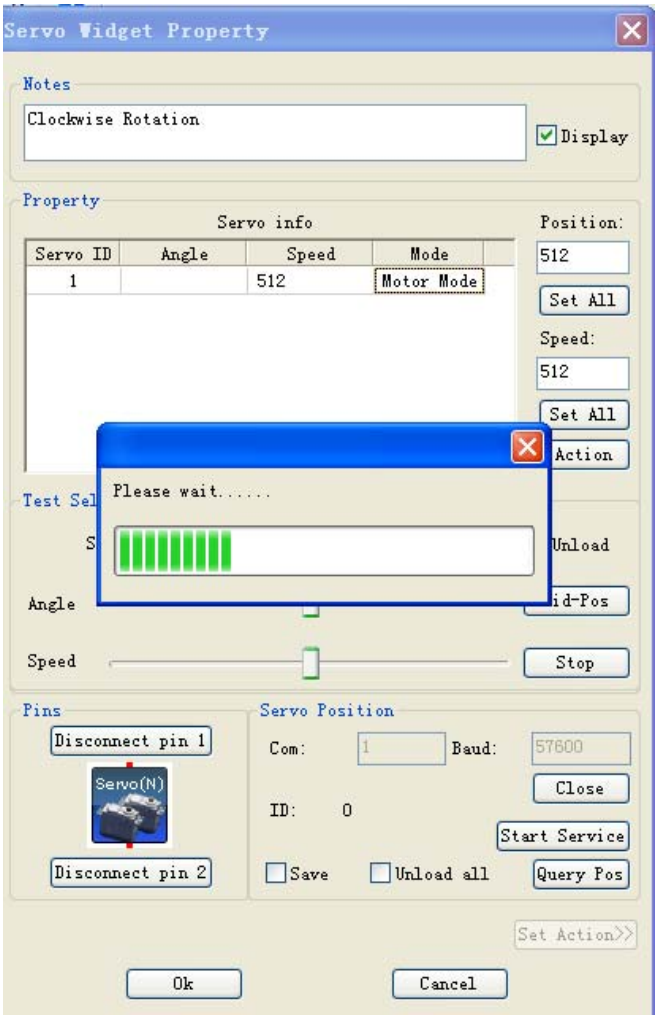


圖 4.10 線上調試

- 等待服務程式下載完成。確認“Com”控制項中的串列埠號碼正確，單擊“Open”。
- 單擊“Servo info”控制項中的“Servo ID”屬性中的“1”。勾選“On-line test”。

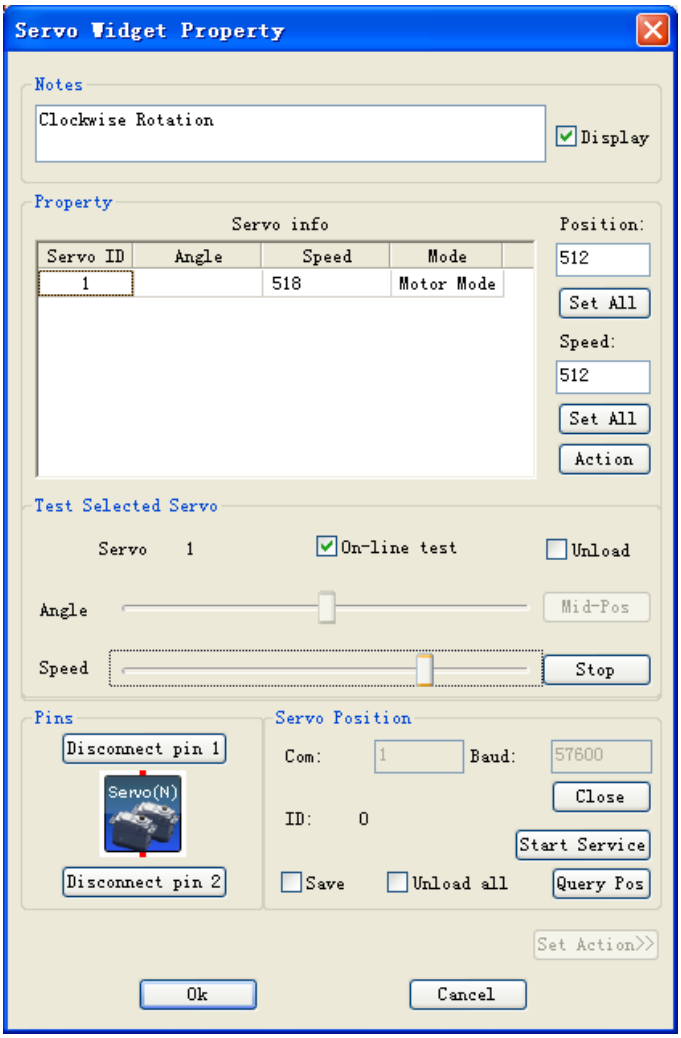


圖 4.11 線上調試

- 拖動“Speed”滑動條，伺服馬達的轉動速度隨著滑鼠拖動變化，同時“Servo info”中的“Speed”值，隨著滑動條的運動改變。
- 下面開始用流程圖進行編程。我們在這裏只是爲了讓伺服馬達按照我們的希望運動起來，所以在這裏，我們先讓伺服馬達正向旋轉 1 秒，再轉爲反向旋轉 1 秒。不斷重複。

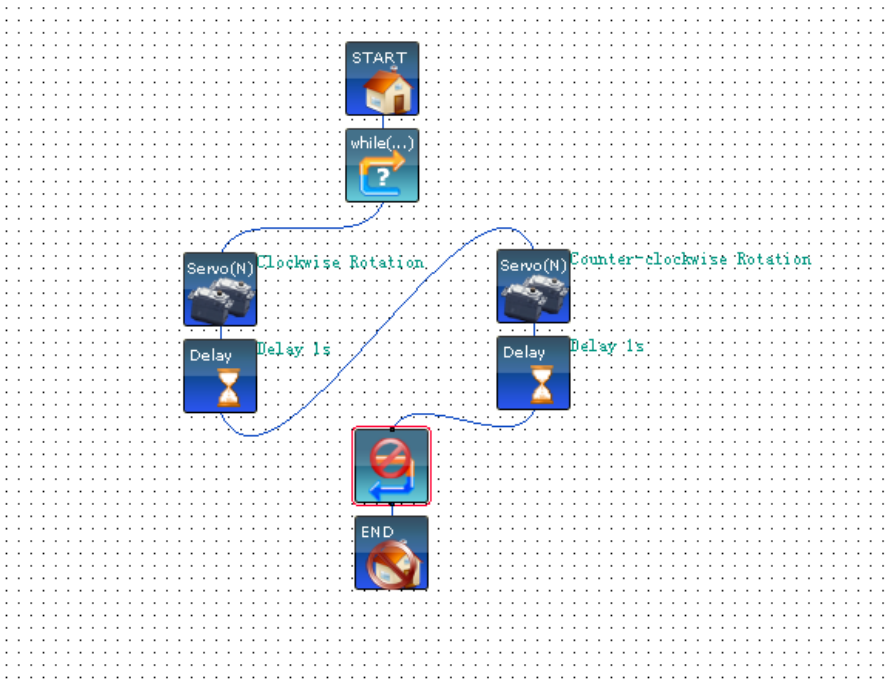


圖 4.12 伺服馬達控制流程圖

4.2.3 分析結果

編譯程式並下載到 MultiFLEX™2-AVR 控制器中，可以看到伺服馬達以 2 秒爲週期不斷往復運動。說明我們的程式可以對伺服馬達的運動速度和運動方向進行控制。

4.3 任務 3：搭建完整的小車

完成伺服馬達的控制實驗之後就可以開始搭建完整的避障小車了，具體的搭建過程可以參照《創意之星機器人套件組裝指南》5.3 小節，最終效果如下圖所示：

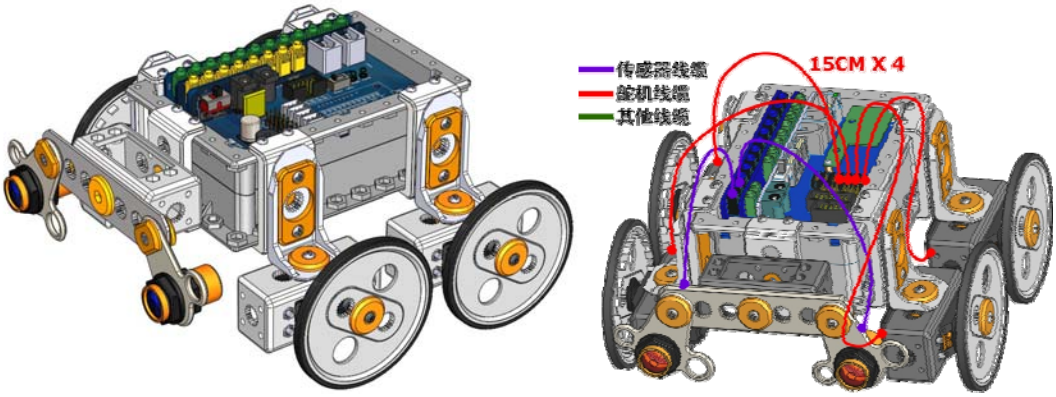


圖 4.13 避障小車

4.4 任務 4：編寫/調試程式，檢查結果

- 打開“North Star”編程環境，按照我們的硬體環境對專案進行設置；“Servo”中設置 4 個伺服馬達，並使它們處於“Moto Mode”。在“AD”中保持預設的“0”。在 IO 中設置 2 個 IO 埠，並使它們處於“Input”模式。

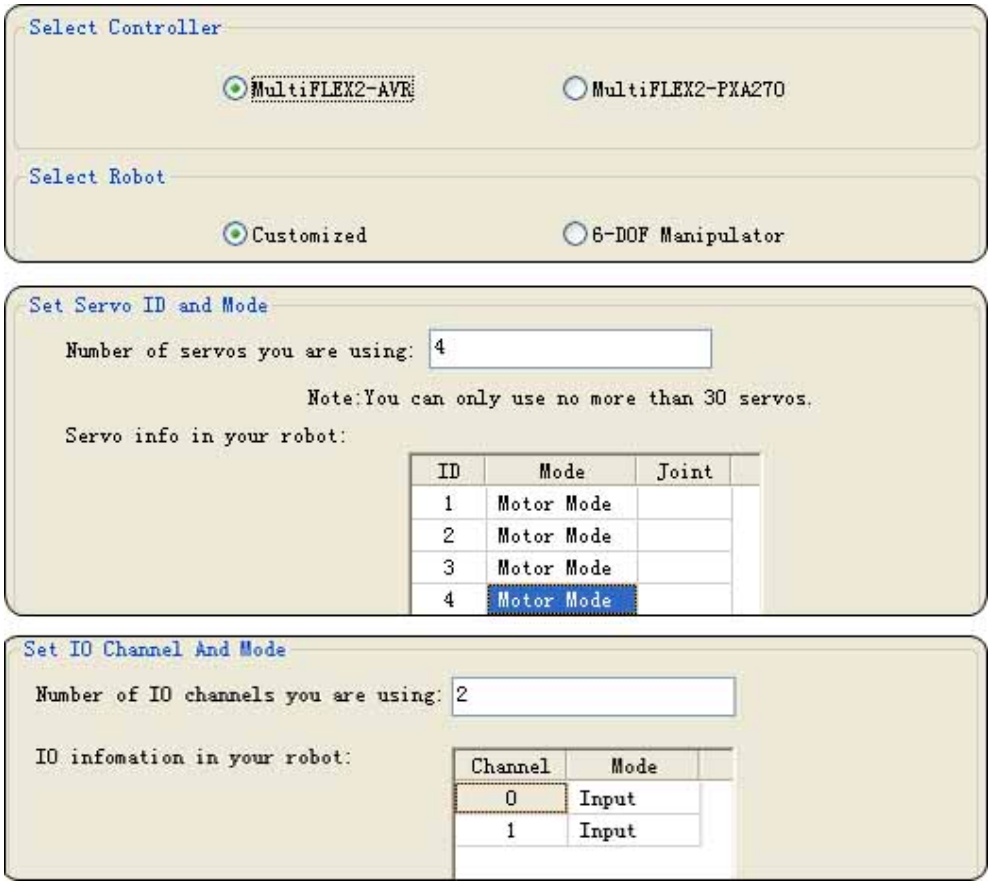


圖 4.14 流程圖專案初始化設置

- 在工作區內拖入 4 個“Servo”控制項，按照上一小節中調試伺服馬達的方法對小車上的 4 個伺服馬達進行調試，將 4 個“Servo”控制項分別調整成固定的動作，分別爲“前進”、“後退”、“左轉”、“右轉”。

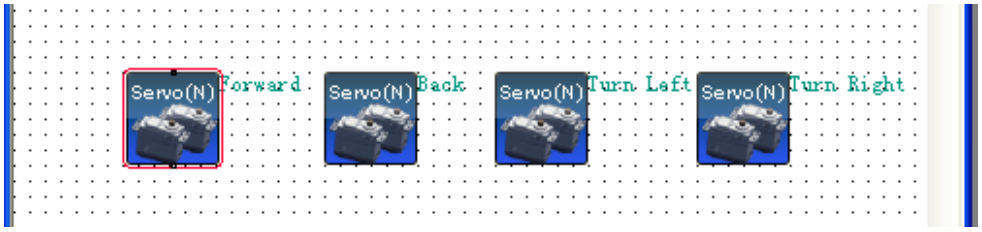


圖 4.15 機器人基本運動方式模組的建立

- 下面開始編寫調用這些運動方式的邏輯。我們可以打開草稿本將小車避碰的行爲流

程勾勒出來。

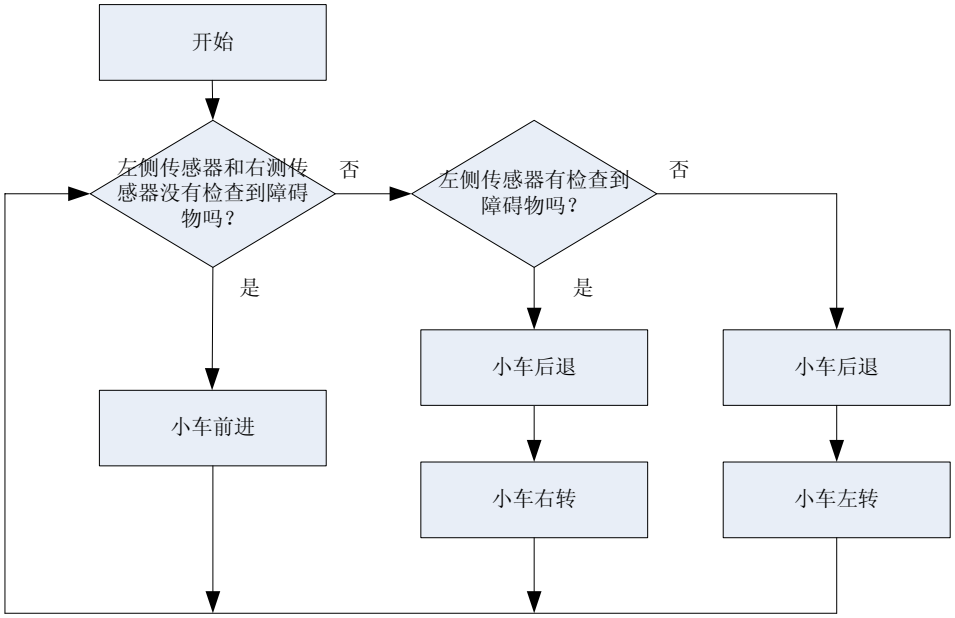


圖 4.16 機器人邏輯判斷流程

- 在勾勒出系統的行爲流程後，就可以以此爲依據設計出程式的流程了。

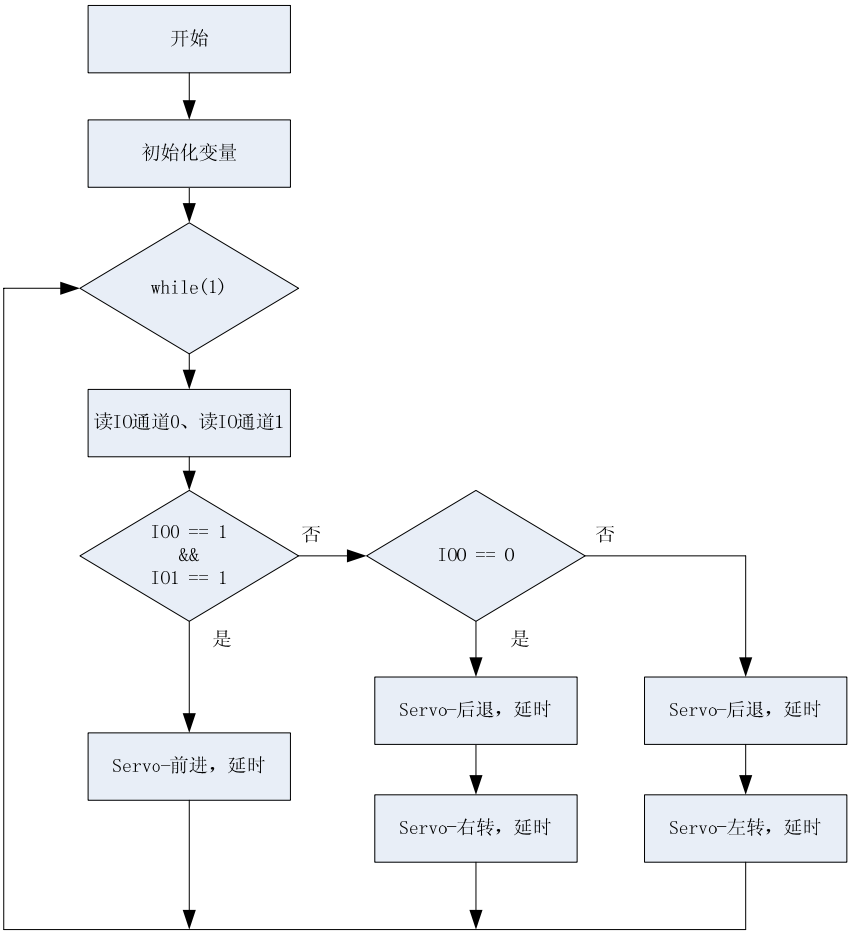


圖 4.17 控制程式流程圖

- 依據設計好的程式流程圖進行編程。

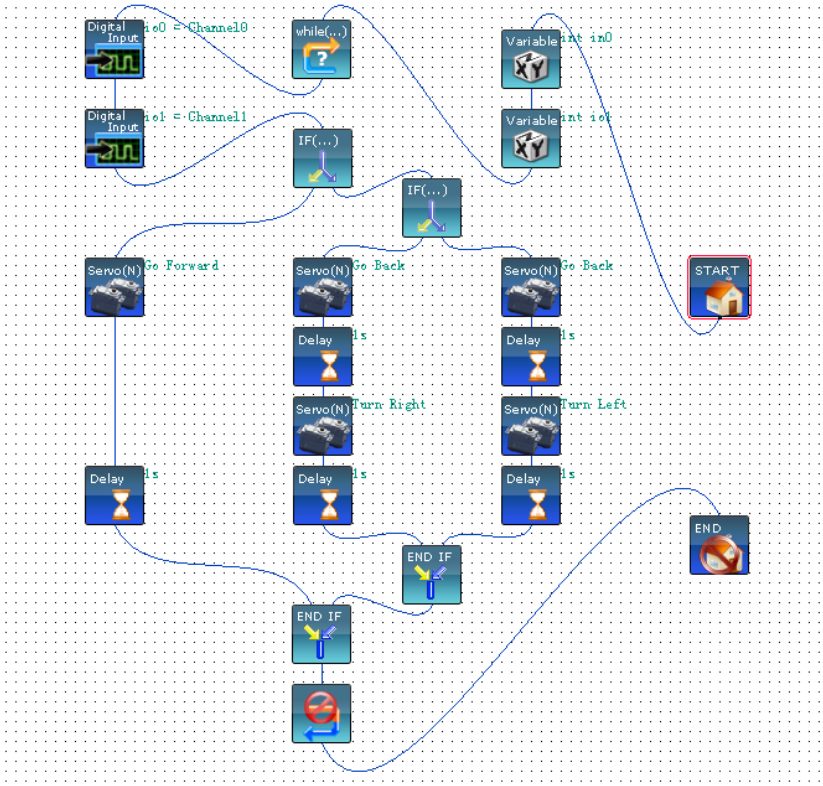


圖 4.18“North Star”程式

- 將程式編譯，並下載到“防碰撞小車”的控制器中，檢驗程式結果，如果與預期不符，則仔細檢查程式流程。

4.5 任務 5：總結和作業

至此，我們完成了機器人實驗中的第一個完整專案，“防碰撞小車”。

- 透過“防碰撞小車”的設計實驗，我們走過了進行一個專案設計的簡要流程，從需求分析到整體方案設計，再到設備選型和細節設計，最終完成樣機調試。這樣的方法在專案實踐中具有一定的通用性。
- 在對模組化機器人的組裝調整中，我們熟悉了各種結構零件的使用技巧，爲設計更爲複雜的結構打好了基礎。
- 我們進行了用控制器連機調試伺服馬達工作狀態的實踐，這是調整複雜機器人步態的基礎。
- 透過 NorthSTAR 的流程圖功能，我們實現簡單的邏輯控制，而各種複雜的邏輯演算法，都是建立在這樣的基本單元之上的。

作業：

參照附錄中的編程介面，使用“North Star”的程式碼編輯功能，將任務 4 中的程式以程式碼的形式寫出，並要求對每一行程式碼進行功能注釋。

4.6 擴充任務：更智能的防碰撞小車

在策略分析時我們說過這個方案不是完美的，原因是這個方案比較機械，不夠智慧，就像一個低級的單細胞生物，只能作出一些應有反應。

問題一、比如下面的情況，機器人走到一個由兩堵牆構成的夾角，初始狀態如左圖，它認爲左邊存在障礙，向右轉，到達右圖狀態，向左轉又回到左圖狀態。接下去它只能不斷的左轉然後右轉，陷入迴圈了。**怎麼辦？**

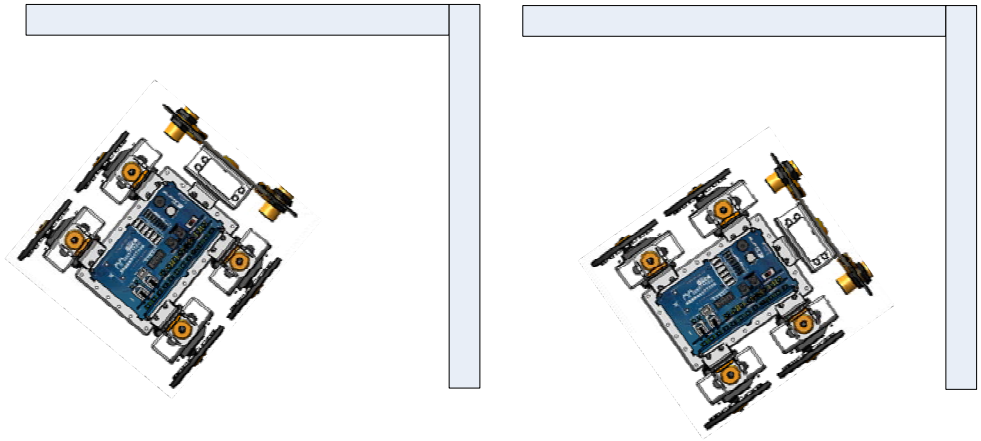


圖 4.19 不知如何應對的情況

問題二、它只能看到水平方向，高出地面一定高度的障礙物。如果障礙物不是高於地面的物體，而是一個樓梯的臺階怎麼辦？它是不是會直接滾落樓梯呢？

這個機器人只吃飯不幹活，成天在遊蕩，能不能給它安排一些任務呢？

如何實現一個鬥牛小車（推障礙物）？如何根據障礙物遠近來決定速度？

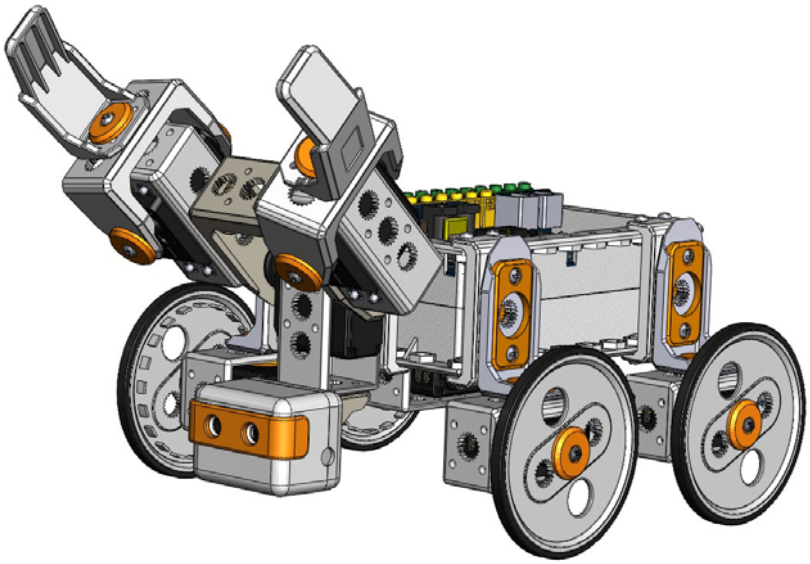


圖 3.17 智能防碰撞小車參考效果圖

第五章 會追光的四足步行機器人

5.1 任務 1：目標分析與計畫

在本章中我們將製作一個用機械足行走的機器人。與輪式機器人比較起來，足式機器人在外形上更接近於生物，具有適應各種地形的可能性。雖然目前足式機器人在實際應用中仍然不很常見，但其廣闊的應用前景仍然吸引了眾多學者。光碟 “……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第五章”的機器人大狗是當前最先進的仿生足式機器人，不久的將來，這種“bigdog”可以代替士兵在複雜的地形環境下運送戰略物資。

5.1.1 瞭解腿式機器人

我們對大自然的各種生物做下觀察，就能夠得出一個結論：大自然偏愛腿式運動。我們可以看到，除了水裏的魚和微生物，其他的生物一般都有或多或少的腿。因爲自然界中，生物的活動環境是非結構化、粗糙、複雜的，昆蟲需要翻越的障礙可以比它們的高度高出一個等級。如圖 5.1 所示，腿式機器人（足式機器人）顧名思義就是使用腿系統作爲主要進行方式的機器人。腿式運動以一系列機器人與地面之間的點接觸爲基本特徵，主要的優點包括在粗糙的地形上的自適應性和機動性。當然，腿式機器人的最主要缺點是動力、控制、結構的複雜性。

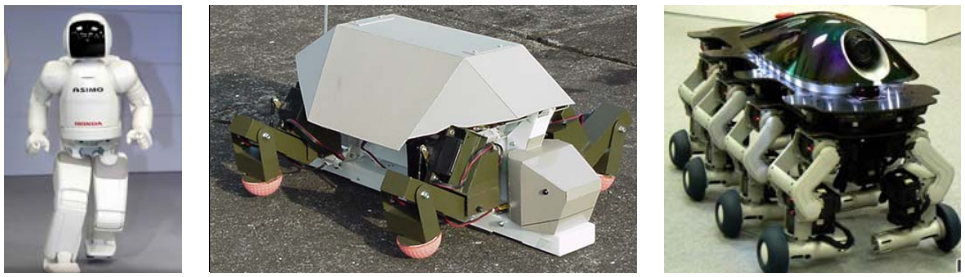


圖 5.1 各種腿式機器人

腿式機器人的構思來源於對腿式生物的模仿，所以，我們在設計腿式機器人時需要回歸自然，對自然屆的各種腿式系統進行初步的研究。在研究腿式機器人的特徵時，我們主要考慮以下幾個方面：

1、 腿的數目（和地面接觸點的數目）
2、 腿的自由度
3、 靜態和動態穩定性

- **腿的數目**

大型的哺乳動物都有 4 條腿，而昆蟲則更多，它們可能有 6 條、8 條甚至幾十上百條腿。人僅靠 2 條腿也可以完美的行走，甚至可以用單腿跳躍前進，不過這也是有代價的，維持平衡需要更複雜的主動控制。

三個點確定一個平面，機器人或動物只需要和地面有三個獨立的接觸點，就能夠保持靜態平衡。但是，機器人需要抬腿走路，所以 3 個點的平衡是不夠的，爲了能夠在行走中能夠實現靜平衡，需要至少 4 條腿，而 6 條腿的動物能在任何時刻都有 3 個穩定的支點。

我們從自然界中也可以得到印證，蜘蛛出生就能行走，平衡對它們而言不是問題。4 條腿的動物剛出生還不能立刻行走，需要用幾分鐘甚至幾個小時來嘗試。2 條腿的人類需要花上幾個月的時間才能學會站立、保持平衡，需要花上 1 年的時間才能行走，需要更長的時間才能跳躍、跑步、單腿站立。

在腿式機器人研究領域，世界各國已經展示了這種各樣的成功的雙腿機器人，最出名的是日本本田的 ASIMO，您在光碟 “\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第一章\ASIMO” 可以找到它的精彩視頻。四腿機器人站立不動的時候可以被動穩定住，但行走起來仍具有挑戰性，在本章的任務 3 您就能夠領會得到。在四腿機器人步行期間，需要主動偏移重心，借此控制姿態，實現移動。最成功的四腿機器人是美國軍方的 bigdog，資料路徑是 “\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第五章\BigDog Robot” 。六腿機器人行走期間的靜態穩定性特性，讓機器人的平衡控制不是問題，所以六腿機器人在移動機器人領域非常流行，您在出貨光碟的參考資料夾下可以找到眾多資料。

- **腿的自由度**

生物種類繁多，各種生物遵循著不同的演化過程，腿作爲生物軀體最重要的一部分之一，其構造也各式各樣。毛毛蟲的腿只有 1 個自由度，利用液壓透過構建體腔和增加壓力可以使得個腿伸展，透過釋放液壓可以使腿回收。而另一個極端方向，人的腿有 7 個以上的主自由度，15 個以上的肌肉群，如果算上腳趾頭的自由度和肌肉群，數量將更加恐怖。

機器人需要多少自由度呢？這個是沒有定論的，就像不同的生物在不同的生活環境和生活方式的刺激下，進化出了不同構造的腿一樣，由於機器人運用場合的不同，對自由度的要求也不一樣。

一般情況下，如圖 5.2 腿式機器人的每一條腿需要兩個關節，從而實現提起腿、擺動向前、著地後蹬的一系列動作。如果需要面對更複雜的任務要求，如圖 5.3 需要增加 1 個自由度，讓腿更加靈活。而仿人機器人的腿的自由度則更加複雜，ASIMO 每條腿都有 6 個自由度。

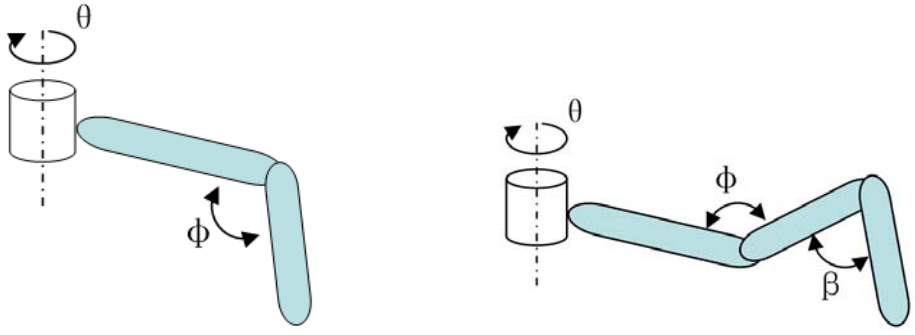


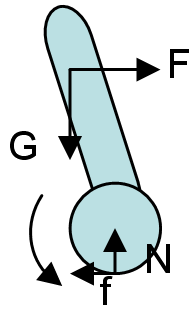
圖 5.2 2 個自由度的腿圖　　5.3 3 個自由度的腿

- **穩定性**

在此我們需要先明確兩個和穩定性相關的概念：靜平衡、動平衡。

在機器人研究中，我們將不需要依靠運動過程中產生的慣性力而實現的平衡叫做靜平衡。比如兩輪自平衡機器人就沒辦法實現靜平衡。機器人運動過程中，如果重力、慣性力、離心力等讓機器人處於一個可持續的穩定狀態，我們將這種穩定狀態稱爲動平衡狀態。

如圖 5.3 兩輪自平衡小車，輪子向前滾動，地面的摩擦力 f、支持力 N、重力 G、慣性力 F 的合力讓機器人保持向前傾斜一個小角度的狀態。當然這個過程輪子必須在不斷的加速，讓 F 慣性力保持不變。



5.4 自平衡小車的受力示意圖

根據上述的分析，我們可以知道，腿越多的機器人，它的穩定性越好，當腿的數量超過 6 條之後，機器人在穩定性上就有天然的優勢。

四足機器人是介於最困難的仿人機器人和相對容易的多足機器人之間，是最典型的腿式機器人研究平臺。四足機器人如果在任何時刻都是單腿邁動，可以保持靜平衡，如果有 2 條腿同時邁動，將不能保持靜平衡。

5.1.2 步態規劃

四足機器人是最典型的腿式機器人平臺，搭建四足機器人平臺能夠瞭解到多數腿式機器人結構搭建、步態規劃、控制系統設計等內容。所以本章我們將搭建一個四足機器人，並讓機器人走起來。

《創意之星機器人套件組裝指南》有四足機器人的搭建過程。完成四足機器人的結構搭建只是最基本的內容，如何讓機器人運動起來呢？這裏涉及到腿式機器人步態規劃這個方面的內容了。

腿式機器人在運動過程中，各腿交替呈現兩種不同的狀態：支援狀態和轉移狀態。腿處於支援狀態時，腿的末端與地面接觸，支援機器人的部分重量，並且能夠蹬腿使機器人的重心移動。腿處於轉移狀態時，腿懸空，不和地面接觸，向前或向後擺動，爲下一次邁步做準備。

步態的定義是：腿式機器人各條腿的支援狀態與轉移狀態隨著時間變化的順序集合。對於匀速前進的機器人，步態呈週期性變化，我們將這種步態稱之爲週期步態。更加智慧的機器人，能夠根據感測器獲取地面狀況和自身的姿態，據此生產即時的步態，我們將這種步態稱爲隨機步態或即時步態。

即時步態的設計過程非常複雜，需要參考專業的書籍。我們只爲四足機器人規劃週期步態。

週期步態中，所有腿支援狀態的時間之和與整個週期的比值，稱爲步態占空比。如果占空比是 0.75，說明不管任何時候，四足機器人一定有 3 條腿支撐軀體，機器人處於靜平衡狀態。如果機器人一直用 4 條腿站著不動，這步態占空比是 1，因爲支援狀態時間和與週期相等。所以，0.75≤步態占空比≤1 時，機器人處於靜平衡狀態，我們將這種步態稱爲靜平衡步態；反之如果步態占空比＜0.75，機器人處於非靜平衡狀態，需要借助運動時的慣性力，保持嚴格的時序，才能讓機器人不摔倒，我們將這種步態稱爲動平衡步態。

動平衡步態太過複雜，需要將各種受力進行配合，且要控制嚴格的機器人運動週期。所以，我們第一步要先實現動平衡步態，讓機器人能夠正常的行走。

根據上述的分析，我們得出幾個結論：

1、 如果機器人要在運動過程中保持靜態平衡，需要在任何時候都有 3 條腿支撐地面，並且重心在這三條腿構成的三角形之間。

2、 機器人需要透過腿部運動，主動移動重心，才能實現機器人的整體運動。

至此，我們已經分析清楚了四足機器人的步態設計要點，接下去的步態規劃遵照上述結論設計一個週期的步態即可，讓這個週期的步態迴圈執行，就能夠得到連續的週期步態。

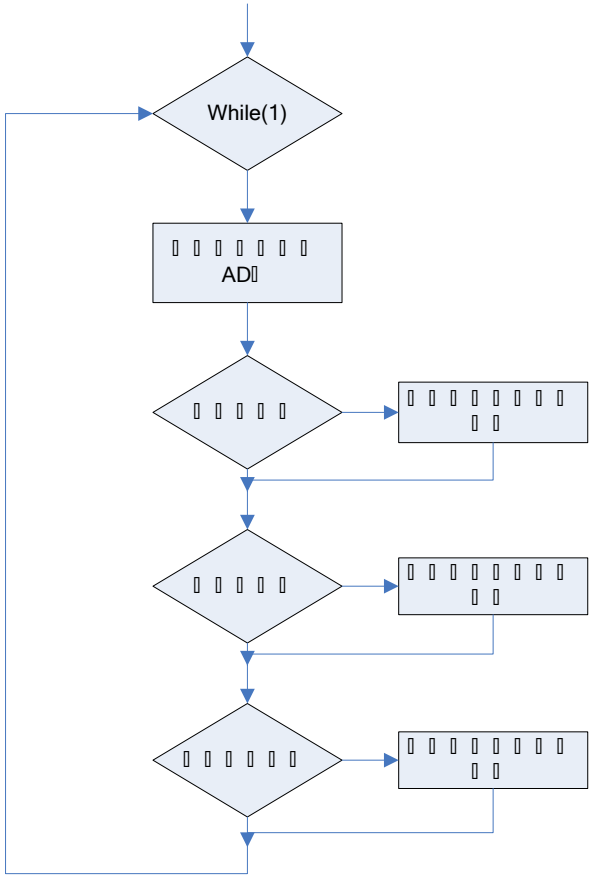
5.1.3 任務規劃

走起來是四足機器人最基礎的要求。仿生機器人需要研究的不只是動物的運動方式，還需要研究動物的行爲模式在機器人領域的運用。我們給四足機器人安裝上兩個能夠感覺光源的眼睛，爲它設計一些應有機制，看看它能夠有什麼反應。

我們期望這個四足機器人能夠利用我們賜予的眼睛去尋找光明，能夠像撲火的飛蛾一樣，英勇無畏的往光明的方向前進。爲此我們給它設計了能夠靈活運動的有四條腿的軀體，接下去將爲它設計帶眼睛的頭部。

在“創意之星”機器人套件所配的感測器裏，對光源敏感的感測器只有光敏感測器，我們可以用它作爲機器人的眼睛。參照昆蟲的樣子，我們可以將兩個感測器佈置在機器人的前方。

當機器人有了矯健的軀體，有了敏銳的眼睛之後，我們還需要考慮它如何控制軀體去接近光源。邏輯方塊圖如下所示，簡單的描述就是透過兩隻眼睛判斷光源的位置，如果光源在前方則往前移動，如果光源在左就往左移動，如果在右邊則往右邊移動。



5.5 程式邏輯方塊圖

至此，您對本項目的各個環節已經有了清楚的認識，我們總結如下：

1、根據《創意之星機器人套件組裝指南》搭建四足機器人平臺。

2、讓機器人行走起來，完成初步的步態規劃。

3、學習類比量感測器的使用，並用採集到的資料來控制一個伺服馬達的運動。

4、讓四足機器人能夠辨別光源的方向，並跟蹤光源。

5.2 任務 2：搭建機器人

參照《創意之星機器人套件組裝指南》6.2 小節搭建 10DOF-四足機器人的過程，搭建出機器人的四條腿，並將四條腿拼合到底板上，如圖 5.6 所示：

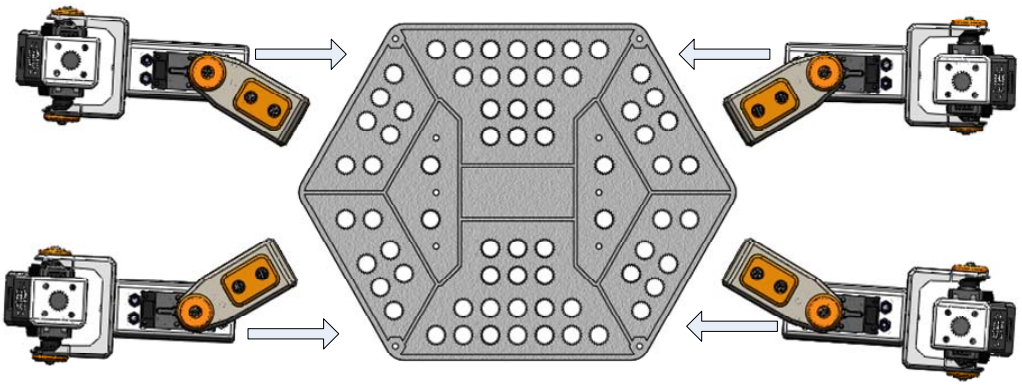


圖 5.6 搭建軀體

將腿部和底板組合到一起，機器人的軀體就完成了。不過看起來還是比較彆扭，我們再爲它組裝一個頭部和尾巴，尾巴只是裝飾作用，頭部安裝 2 個光敏感測器，能夠讓機器人看到光源。機器人對比左右兩隻眼睛的光照強度，就能夠判斷光源相對自己的位置，進而調整步態，往光源處走去。

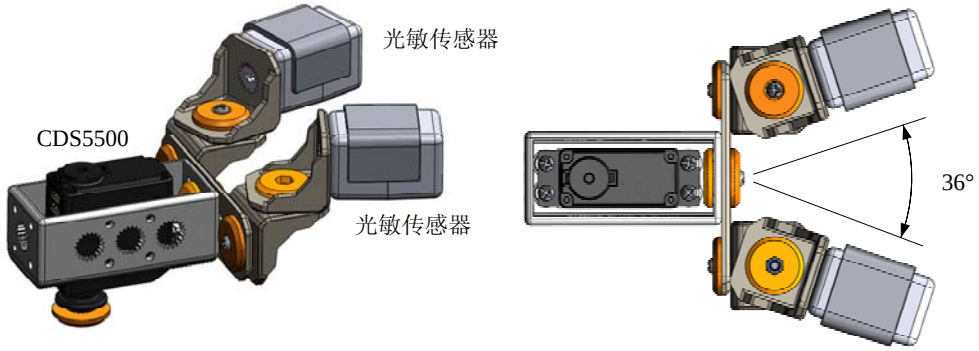


圖 5.7 搭建頭部

如圖 5.7 所示，我們給機器人頭部安裝了一個 CDS5500，作爲機器人的脖子，讓機器人可以左右擺動，尋找目標。前方的兩個光敏感測器有 36°的夾角，用於區分左右的光線。如圖 5.8 所示，當左前方的光源照射到機器人時，左邊的光敏感測器能夠大面積接受光照，右邊的光敏感測器由於 36°夾角的存在，光敏感測器探頭被外殼擋住，照射到傳感上的光線很少。因此兩邊的感測器輸出值就會有很大的差異，機器人據此判斷光源的大致方位。

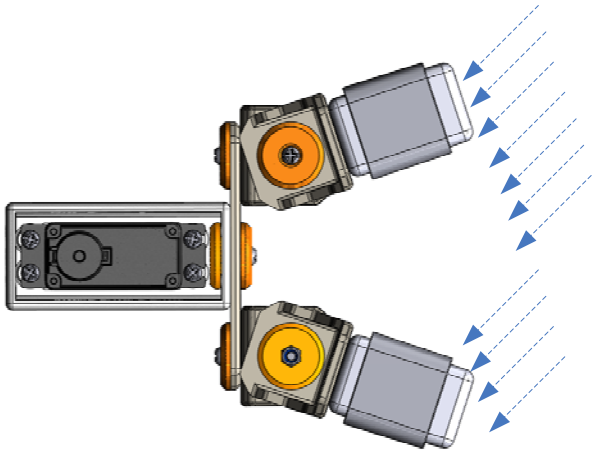


圖 5.8 光照示意圖

如圖 5.9 所示，我們將機器人軀體、頭部、尾巴、控制器組裝在一起，就完成了整個機器人的搭建過程。不過，現在的機器人只是有塑膠、零件、馬達等拼裝在一起的死物，我們接下去要透過編程賦予它生命。



圖 5.9 四足爬蟲

5.3 任務 3：讓四足機器人行走

多足機器人的行走方式按照平衡實現方式可以實現兩種，一種是接近動物行走方式的動態平衡方式，另一種是靜態平衡方式。

動態平衡的行走方式在 4 足動物的身上經常可以看到，如果您注意過烏龜的爬行就會發現它們在行走時，身體一側的前足和身體異側的後足總是同時抬起向前探出，而觸地的兩個支撐足則後蹬，使身體的重心向前移動，在身體傾覆之前，探出的兩個足已經觸地，成爲支

撐足。這樣不斷交替，烏龜就可以向前方移動了。在這個過程中，龜腹是不會觸碰地面的。我們選擇烏龜作爲研究範本，只是因爲它們的行動緩慢容易觀察，如果仔細觀察哺乳動物（小貓小狗一類的），就會發現其實 4 足動物的基本運動方式都是這樣的。

靜態平衡的行走方式在 4 足以上的節肢動物身上比較常見。可以捉一隻金龜子之類的甲蟲來來觀察它的運動，您會發現這些小蟲在前進中總有至少三個足接觸地面，而小蟲的重心的投影總在以這三個支撐足爲頂點的三角型區域內，這說明小蟲運動中的任何一個時刻，自身都是平衡的，即使完全停止它的動作，也不會傾覆，這與剛才探討的烏龜就不同了，如果把烏龜運動中的某些定格提出來，我們會發現那些狀態是會翻倒的。

對於我們的四足機械來說，靜態平衡的行走方式無疑具有明顯的優勢，就是我們在調試的時候不用擔心機械會翻覆，只要滿足了不翻覆這一點就說明步態的調試沒有出現大問題。所以我們選擇靜態平衡的行走方式。

5.3.1 設計前進、後退步態

在設計步態之前，我們先設置好約束條件：

- 機器人在任一個時刻必須有不少於 3 個著地支撐足。
- 機器人的重心的投影必須在 3 或 4 個支持點所圍成的多邊形內。

我們先來看看機器人上電靜止時的狀態：

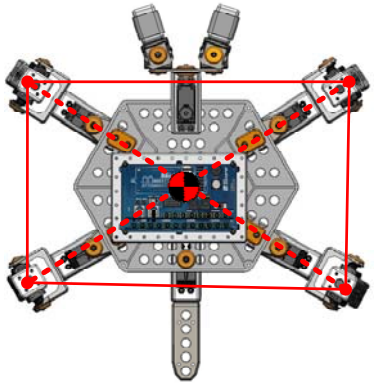


圖 5.10 靜止狀態

可以看到，重心處於 4 個支點的正中心，這意味著失去任何一個支點，結構的重心都將處於其他三個支點所構成的三角形的一條邊上，結構會有向一側傾覆的危險。所以爲了讓結構能夠平穩的抬起一支腳需要先讓重心移動。當然，我們無法改變結構的配重狀態，所以我們先控制所有的足向後方平移，在摩擦力的作用下，軀幹將向前探出。

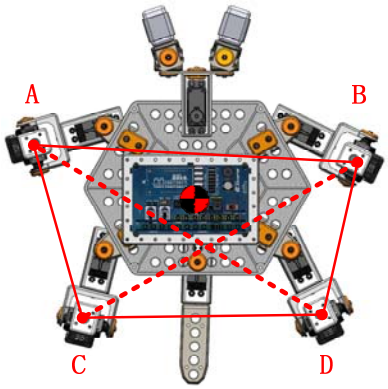


圖 5.11 前移重心

前移後的重心位於 ABC 區域和 ABD 區域中，所以現在可以移動 C 或 D 兩個支撐足中的一個了。我們選擇移動Ｄ。如圖 5.11，落地後的 D 足與 A、C 足形成的 ACD 區域包含重心，B 足被解放出來，可以開始移動了。

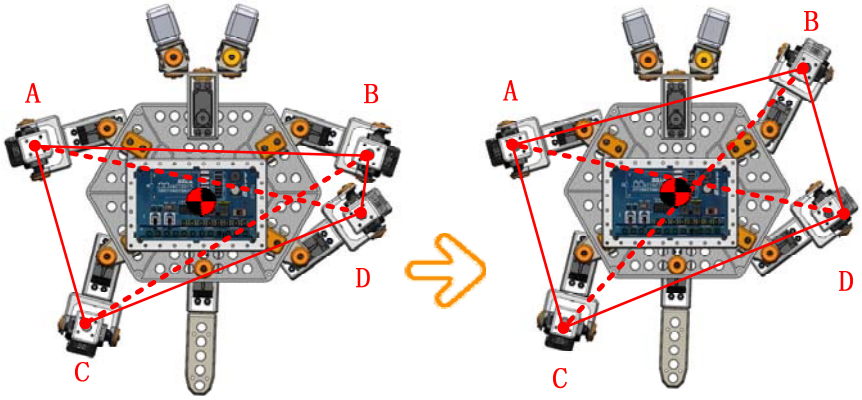


圖 5.12 右前進步態

右側的 B、D 足向前邁出的動作做完後，開始向後蹬的動作，使重心向左前方移動。

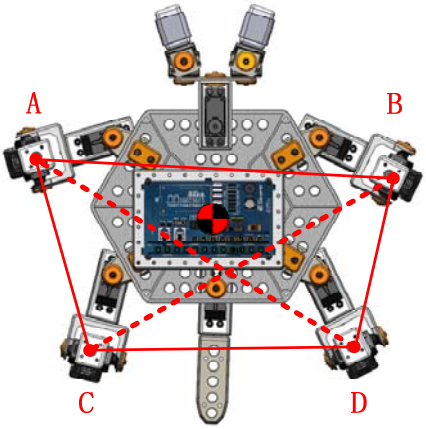


圖 5.13 恢復初始姿態

機器人回到剛開始重心前移後的狀態，但是頭的朝向偏向了左邊。爲了彌補方向的偏

移，如圖 5.13，左側的 C 足開始向前邁出。C 足落地後，重心在 BCD 區域內。A 足得到解放，向前邁出。

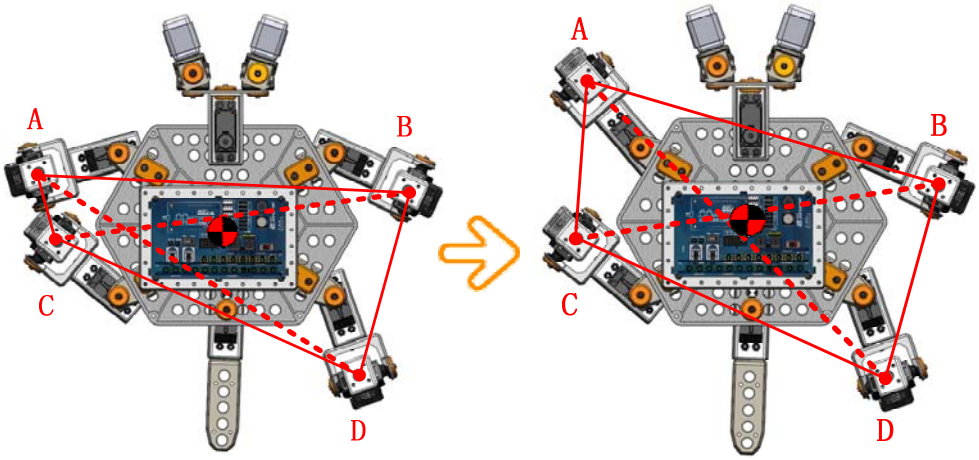


圖 5.14 左前進步態

左側的 A、C 足向前邁出完畢，開始向後蹬的動作，使重心向右前方移動。機器人的頭部回正。

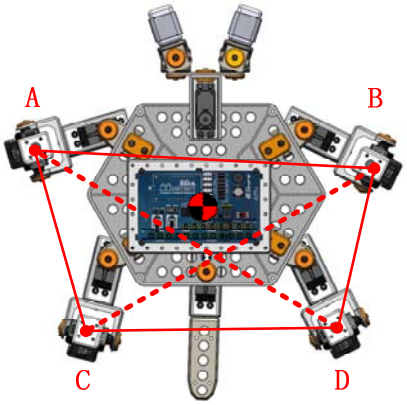


圖 5.15 恢復初始姿態

至此一個前進運動週期就完成了，按照這樣的方法迴圈下去，機器人就會向前一左一右地前進了。

以上我們所介紹的是一個最簡單的讓四足機器人前進的步態，但是這種步態並不是最理想的，因爲在重心前移的過程中總是 4 個支持足同時著地，這樣足的位置變化會在地面上發生滑動，而滑動的結果是使大量的能量浪費掉了。所以，如果能夠在 3 個足著地的狀態下進行機械足的後蹬運動就會節省更多的能量。

讓我們在來觀察我們的機器人，把重心前一步的步態做一下調整，讓 A 向前伸出，C 不要直接運動到最後方。這是爲了我們的前進步態週期能夠完美閉合而進行的修訂。這時重心在 ABC 區域內，讓 D 足向前邁出，同時，A、C 足向後蹬。D 足向著地後，B 足向前邁出，在 B 足著地之前機器人有 3 個足著地，所以這時讓 D 開始後蹬。

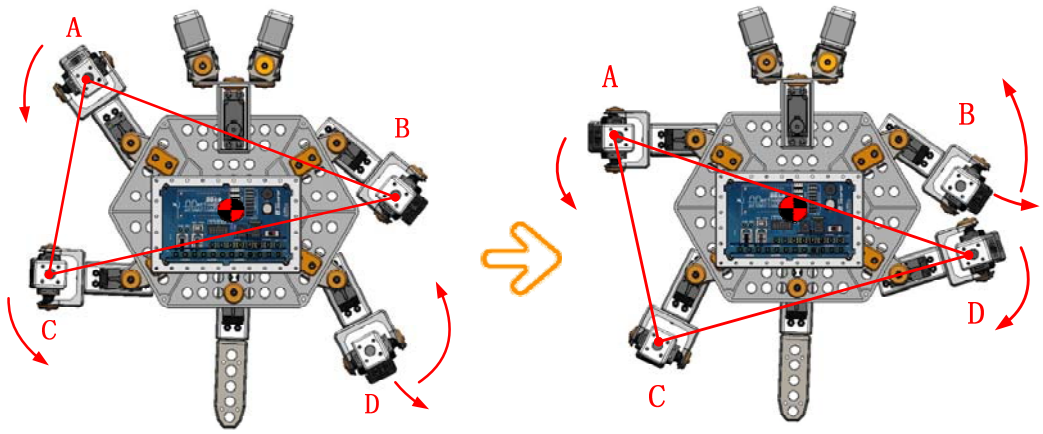


圖 5.16 調整步態

在重心脫離 ACD 區域前讓 B 足著地，與 D 足一起繼續後蹬，一旦重心落入 ABD 區域，馬上讓 C 足向前邁出。

如圖 5.17，在 C 足著地前，D 足已經後蹬到了規定位置。C 落地後開始後蹬，並向前邁出 A 足，這時 B 足後蹬結束。A 足著地後後蹬，這時向前邁出 D 足，完成一個前向運動流程。

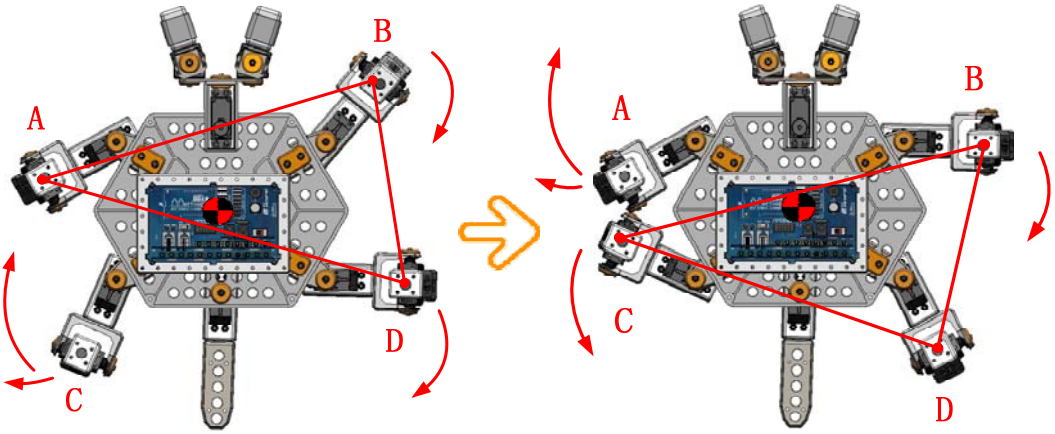


圖 5.17 左前進

如果您曾經仔細觀察過昆蟲綱或蜘蛛綱動物的運動，會發現他們就是按照類似上述流程行動的。只要是直線運動，都可以用這樣的方法實現出來，因爲後退只是將頭尾的方向對調而已。值得注意的是，通常我們思考後退步態的時候，習慣的人爲安裝“倒放錄影”的方式將整體流程顛倒過來就可以了，實際上這樣的方式是不理想的，因爲用這樣的方式時忽略了重心位置的影響。

5. 3. 2 設計轉向步態

下面，我們仍然按照由簡到繁的方式來分析轉向流程。第一步仍然是移動重心。

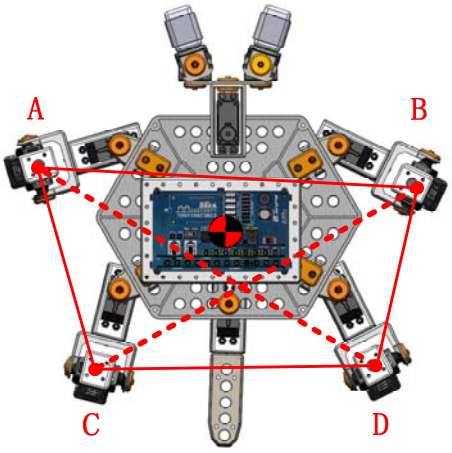


圖 5.18 初始姿態

然後依次邁出 D 足、B 足。

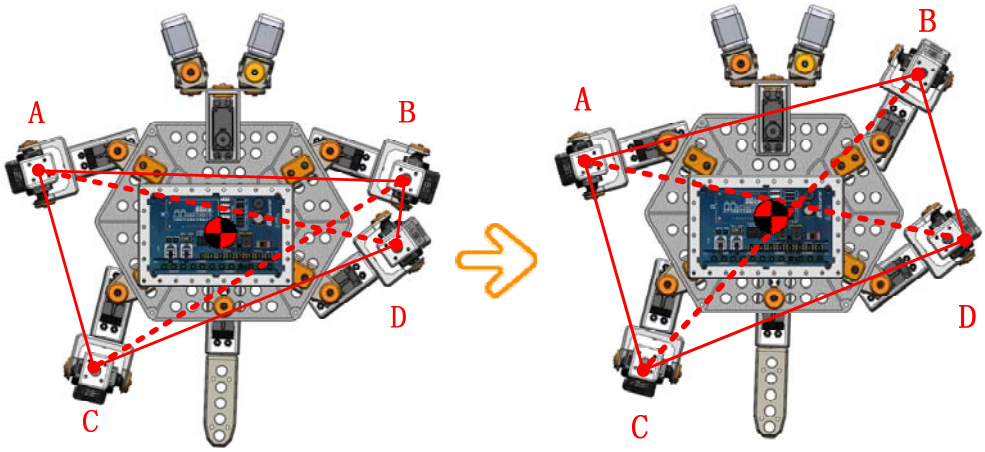


圖 5.19 轉向姿態

這時，讓 A、B、C、D 四個足同時蹬地，B、D 後蹬，A、C 前蹬。機器人軀幹向左扭轉一個角度。

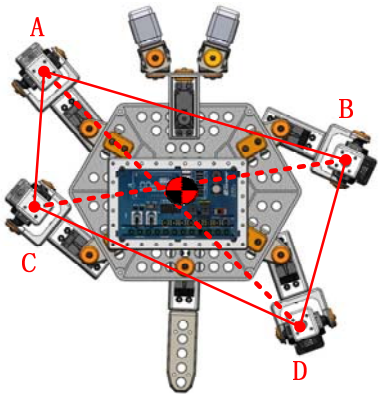


圖 5.20 轉向過程

這個狀態，重心在 CBD 區域中，所以 A 足可以移動。讓 A 向後邁出後，重心在 ABD 區域中，C 足也可以移動了。讓 C 足向後邁出。

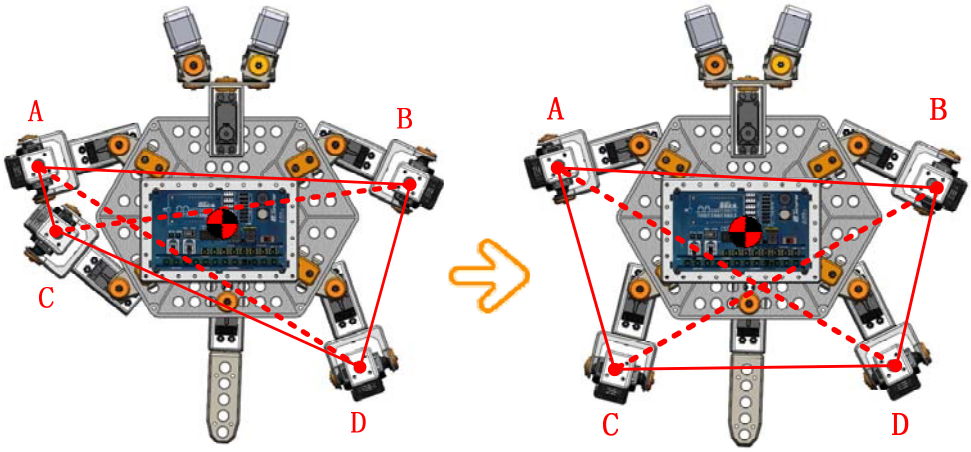


圖 5.21 轉向完成

至此，一個完成的原地轉向週期就完成。在這裏，我們遇到了在前面已經提及過的問題，就是 4 足同時蹬地時的效率低下。於是，我們採用與剛才相同的思路修正步態。首先是重心移動後的狀態，修正爲如圖 5.22 所示。讓 A、C 向前蹬地，重心在 ABC 區域，向前邁進 D 足。

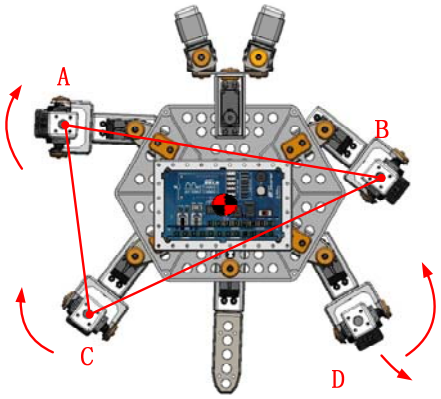


圖 5.22 修正步態

D 足著地後，重心在 ACD 區域，C 足繼續前蹬，D 足後蹬，B 足向前邁出。當 B 足著地後，C 足移動到了限制位置，重心在 BCD 區域，A 足向後邁出。

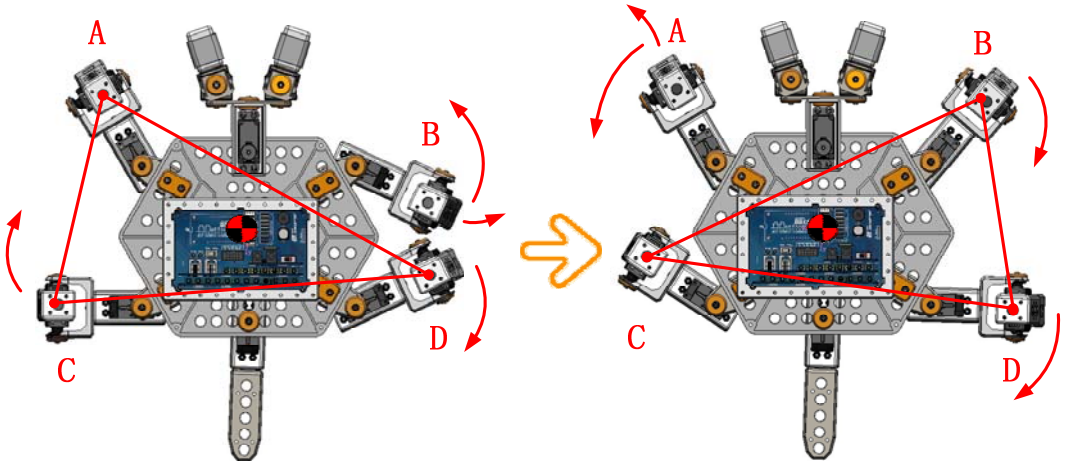


圖 5.23 轉向過程

A 足著地後，D 足移動到了限制位置，重心在 ABD 區域，C 足向後邁出。C 足著地後，機器人回到了初始狀態，完成了一個左轉週期。

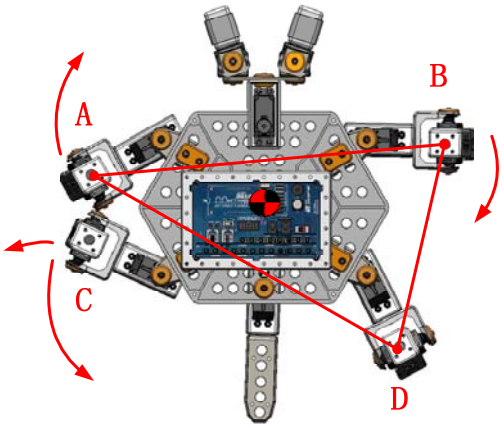


圖 5.24 轉向完成

右轉的步態也可以用這樣的方法實現，只是邁向前方的是左側的足。

5. 3. 3 編寫步態

以上步態的理論分析就結束了。我們怎樣才能用程式實現這些步態呢？現在打開“North Star”開始建立專案。在選擇機器人結構的單選框裏選擇“four legged”（四足機器人），這些預定義的結構帶有三維虛擬場景。

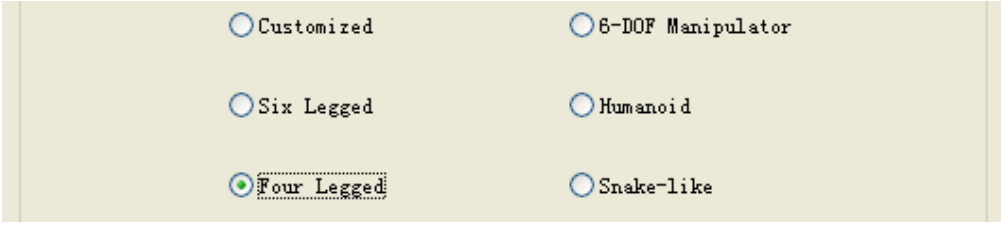


圖 5.25 預定義結構

在“Servo”視窗中，透過修改“ID”欄內的伺服馬達 ID，就可以改變伺服馬達 ID 號，參照視窗右邊圖示，設置 ID 與機器人關節之間的對應關係。

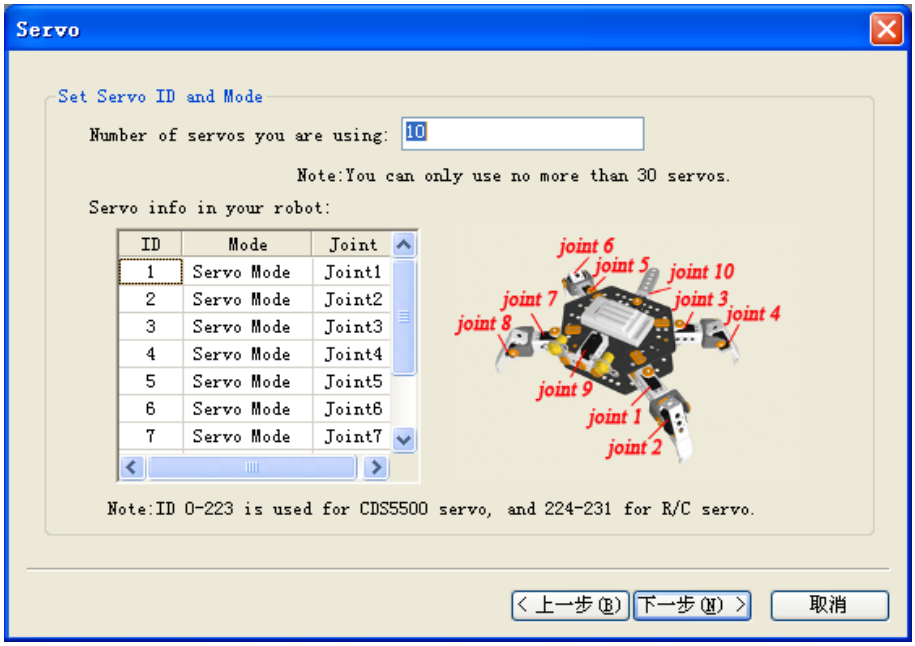


圖 5.26 設置伺服馬達

在工作區視窗中拖入“Servo”控制項，雙擊打開屬性視窗。

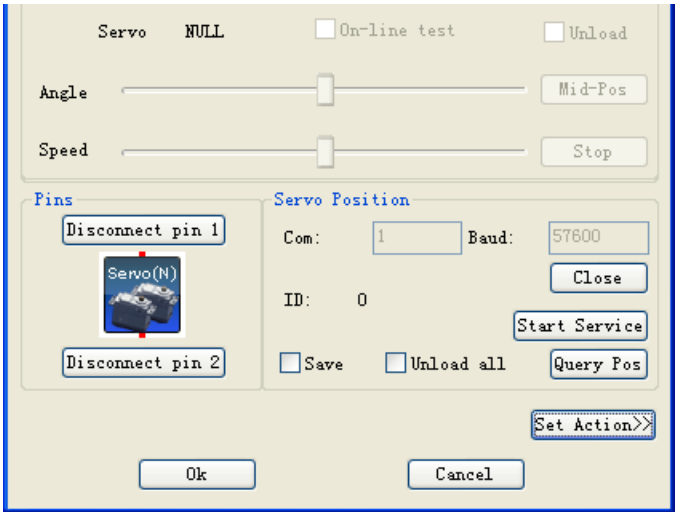


圖 5.27 伺服馬達屬性視窗

單擊“Set Action”展開 3D 虛擬環境。

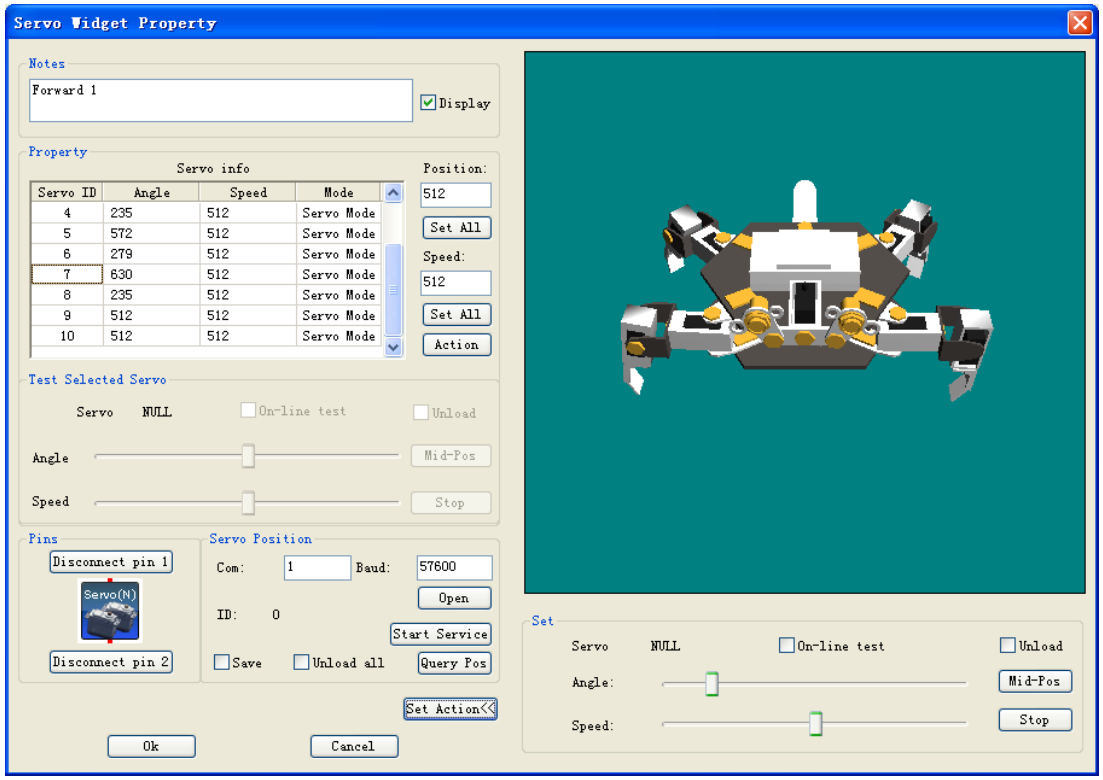


圖 5.28 三維虛擬場景

用滑鼠拖動三維場景中的結構就可以生成相應的伺服馬達位置序列，將 3D 模型擺放出自己需要的狀態，單擊“OK”就可以使“Servo”控制項記錄下當前的狀態，程式執行該控制項的時候，機器人就會呈現出和三維場景一樣姿態。如果將機器人透過多功能調試器與電腦相連，還可以在“On-line Test”控制項上打勾，這樣，拖動 3D 結構的時候，機器人也會跟著動起來。

還有另一種調試方式，單擊“Open”打開串列埠，單擊“Start Service”下載服務程式。選擇“unload all”，將所有伺服馬達卸載，這時可以用手直接扭動結構的關節，把結構調整到我們需要的姿態，比如初始狀態，點擊“quary pos”，North Star 會查詢伺服馬達的當前位置，“Property”中的“Angle”值會隨著我們對機器人的扭動而改變。如果在“Save”上進行勾選，則電腦讀取的伺服馬達狀態將被記錄在列表框中，點擊“OK”即完成一個步態的編寫。

在調試步態的過程中，常常會遇到寫一部分，調試一部分的情況，因爲伺服馬達運動的速度參數很難在實驗前計算到位，所以調試過一次之後，機器人的各個關節都處於位置鎖死的狀態，所以我們還會用到另一個功能“Unload all”勾選該選擇框後，所有的伺服馬達上的力矩將解除，這時我們可以輕鬆扭動機器人的每個關節了。

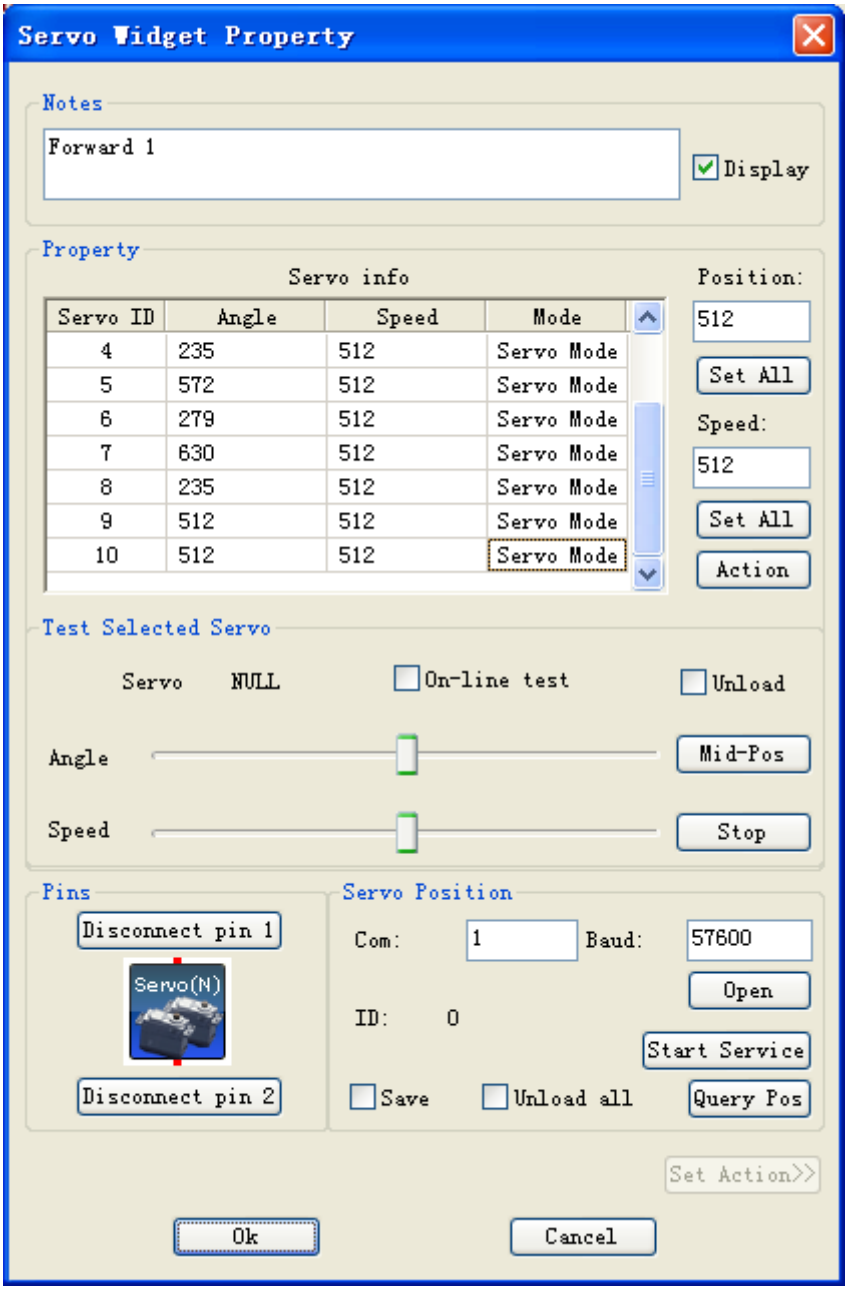


圖 5.29 伺服馬達屬性視窗

用上述方法將機器人的動作分成若干狀態點，一步步進行伺服馬達運動序列調試，這樣將一個完整的行走流程調試完後，我們就實現了 4 足機器人的“前進”、“後退”、“左轉”、“右轉”等基本運動方式控制模組了。下面將這些運動方式控制模組與邏輯判斷模組組合起來就可以實現機器人的“追光”行爲了。

5.4 任務 4：讓伺服馬達跟蹤光源

在進行正式的機器人動作編程前，我們先來熟悉一下將要用到的光強感測器。這裏我們設計一個小實驗，用機器人頭部的旋轉伺服馬達來跟蹤光源。

- **建立專案**

在 NorthSTAR 軟體中新建一個 Project，跳出 Project Configuration 頁面。“Controller”欄選擇我們的編程物件：MultiFLEX2-AVR；因爲我們搭建的是標準結構，“Robot”欄選擇“Four Legged”。點擊“next”，跳出 Servo 頁面。

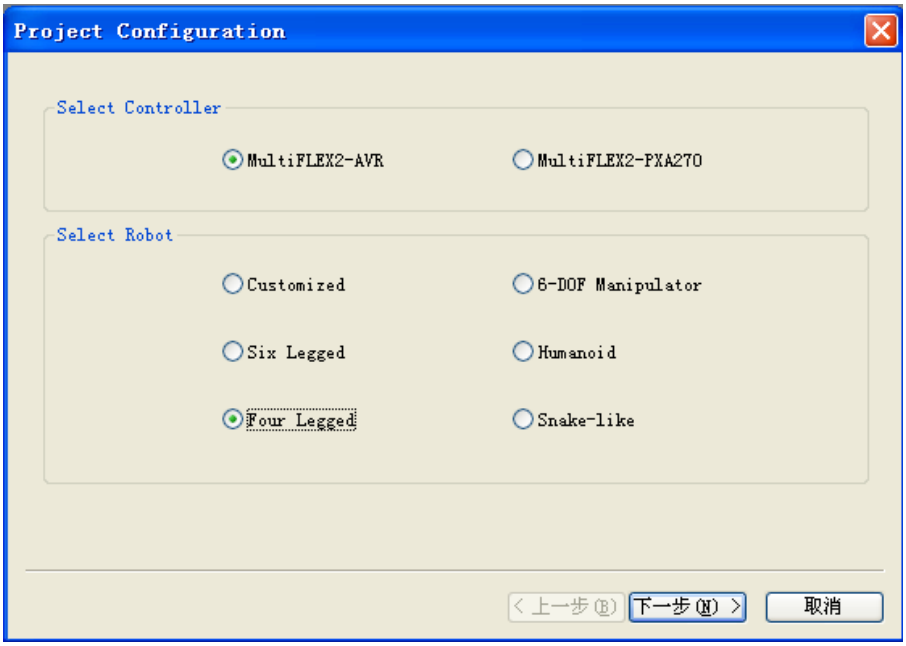


圖 5.30 新建專案配置

在 Servo 頁面裏，我們可以看到機器人結構上所有關節對應的伺服馬達 ID，無需修改，直接點擊“next”按鈕，進入 AD 頁面。

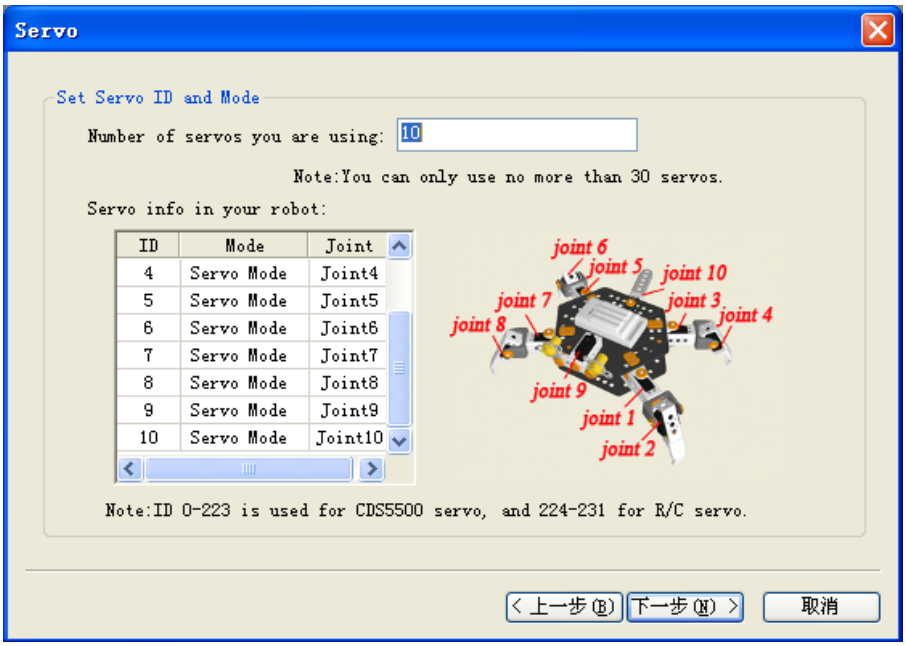


圖 5.31 伺服馬達配置

實驗中用到的兩隻光強感測器分別對應兩路 AD 信號，所以在 AD 頁面中我們在“Number of AD channels you are using”中填入數字 2。點擊“next”按鈕，進入 IO 頁面。

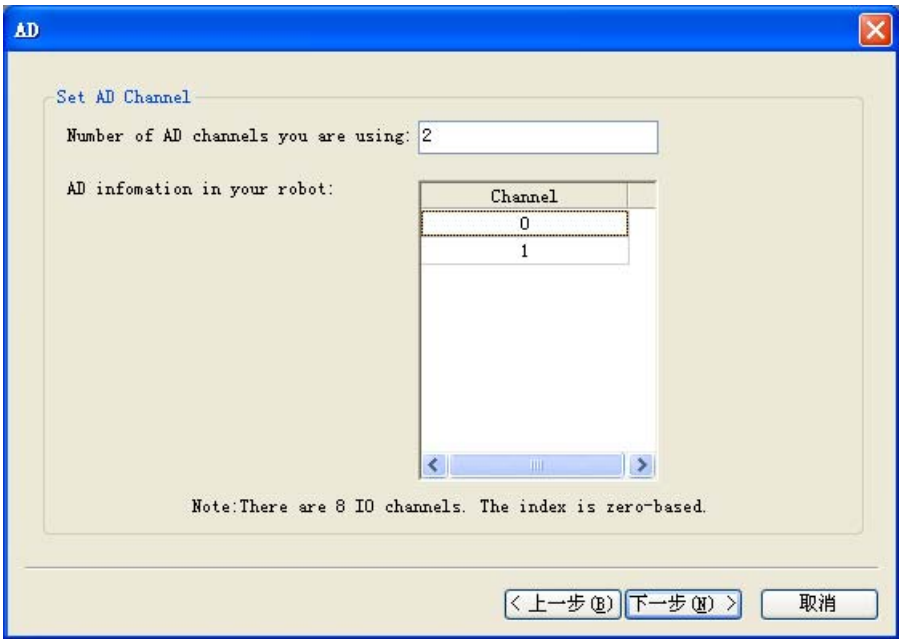


圖 5.32 AD 感測器配置

實驗中未使用 IO 感測器，所以 IO 感測器數量設爲 0。

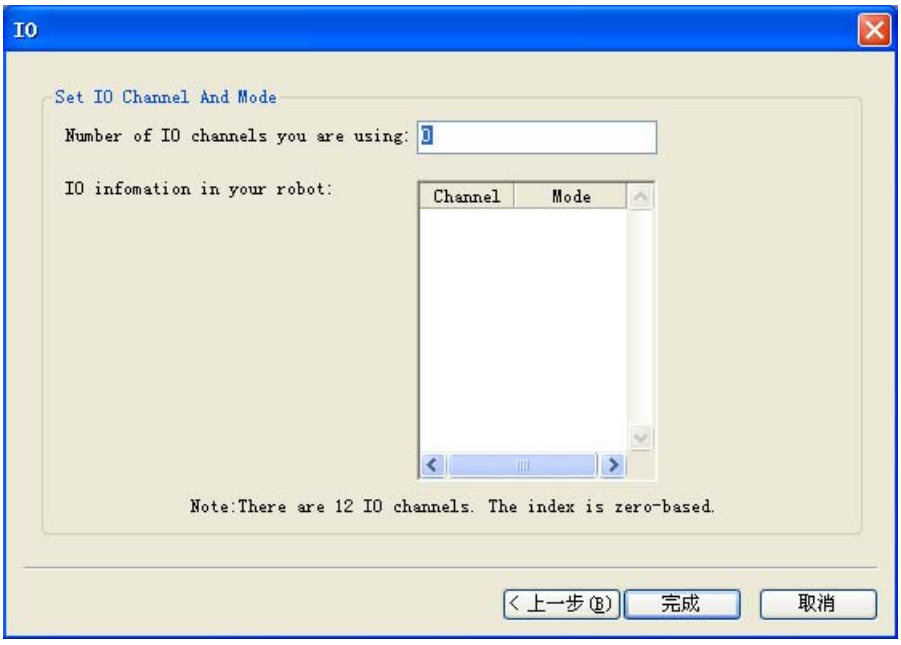


圖 5.33 IO 感測器配置

- **添加變數**

實驗中我們需要添加兩個變數來儲存光強感測器獲得的光強數值。滑鼠右擊流程圖空白處，在跳出的功能表裏選擇“Add variable”，生成一個 Variable 控制項，雙擊該控制項進入其屬性頁，選擇變數類型，輸入變數名，點擊“OK”按鈕。

本實驗需要添加的變數：

變數名	類型	說明
Left	Int	儲存左側光強感測器數值
Right	Int	儲存右側光強感測器數值
Diff	Int	儲存左右兩側光強數值的差值



圖 5.34 添加變數

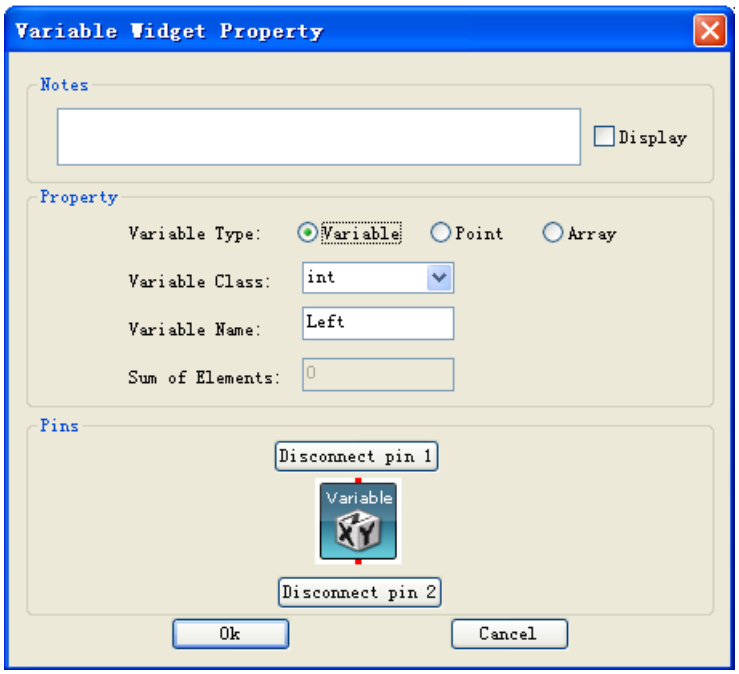


圖 5.35 變數屬性

變數添加完畢之後，需要將其與相應的感測器建立連接。從介面左側的控制項列表“Function Widget”裏拖曳一個 AD Input 控制項至流程圖裏，雙擊該控制項進入屬性頁，在“Return Value”裏設置保存感測器數值的變數。這裏我們將左側光強感測器連到控制器的 AD0，將右側光強感測器連到控制器的 AD1；設置 Channel-0 的 AD 值儲存在 Left，Channel-1 的 AD 值儲存在 Right。

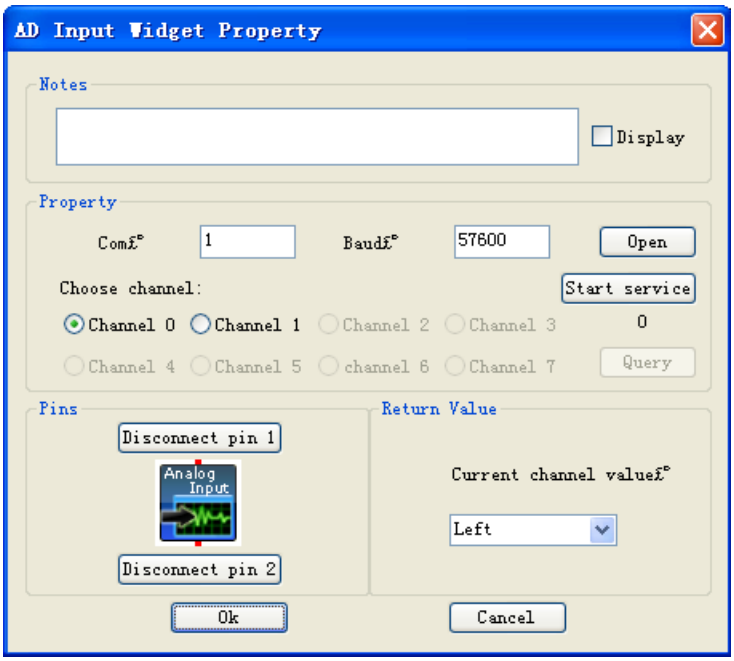


圖 5.36 設置光強返回值

- **程式邏輯設計**

在獲取了兩測的光強數值之後，我們可以開始設計程式的邏輯框架。我們將情況分爲三種：狀況一，左側光比右側光強；狀況二，左側光比右側光弱；狀況三，左右光強近似相等。這三種情況在邏輯上較好區分，且覆蓋了可能出現的狀態。在程式的主迴圈裏，我們用三個條件分支來區分這三種狀況。

作爲左右光強比較的中間變數，我們對兩側光強的差值 Diff 進行計算：

Diff = Left – Right

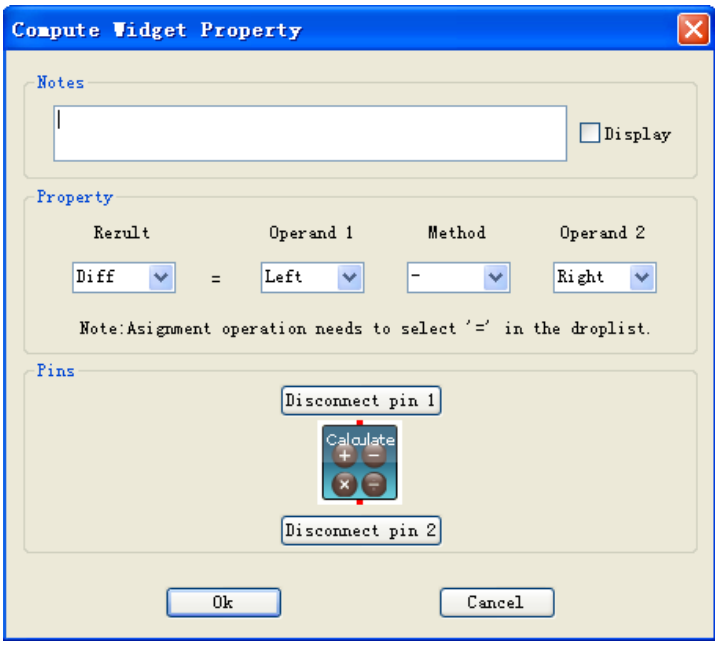


圖 5.37 計算左右光強差值

三種狀況的區分，將圍繞這個差值 Diff 進行：

狀況一，左側光比右側光要強：Diff >50
狀況二，左側光比右側光要弱：Diff <-50
狀況三，左側光強和右側光強近似相等：Diff>=-50 且 Diff<=50

在判斷條件裏，使用的 50 這個數值，是考慮到光強感測器採集資料可能存在著差異，在後面的實驗中，可以根據光強感測器的具體表現進行調整。

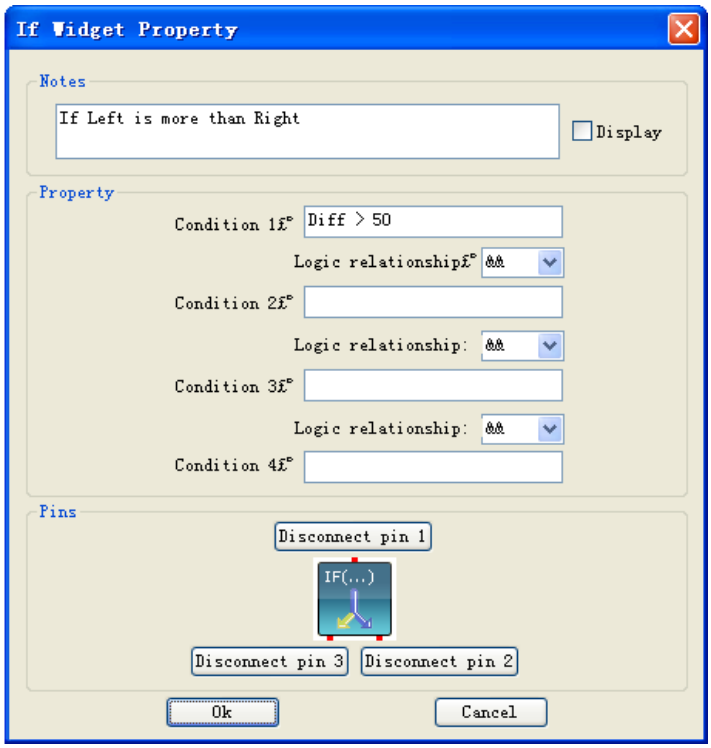


圖 5.38 左側光較強的判斷條件

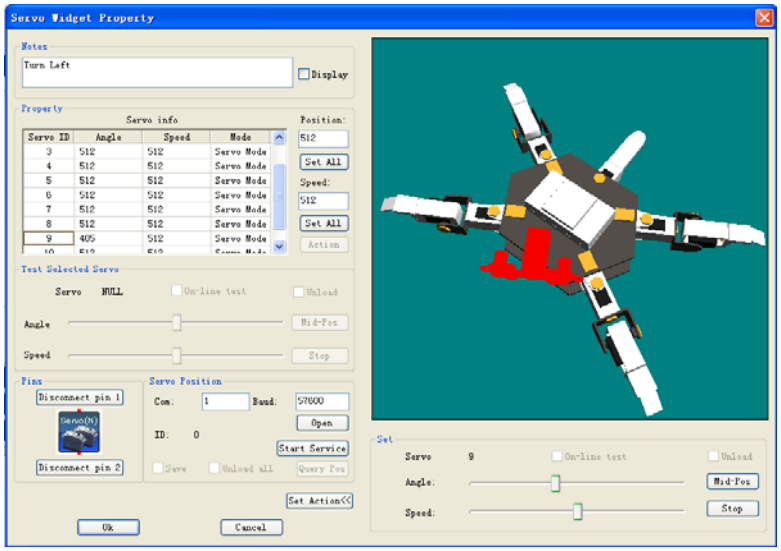


圖 5.39 頭部伺服馬達轉向左側

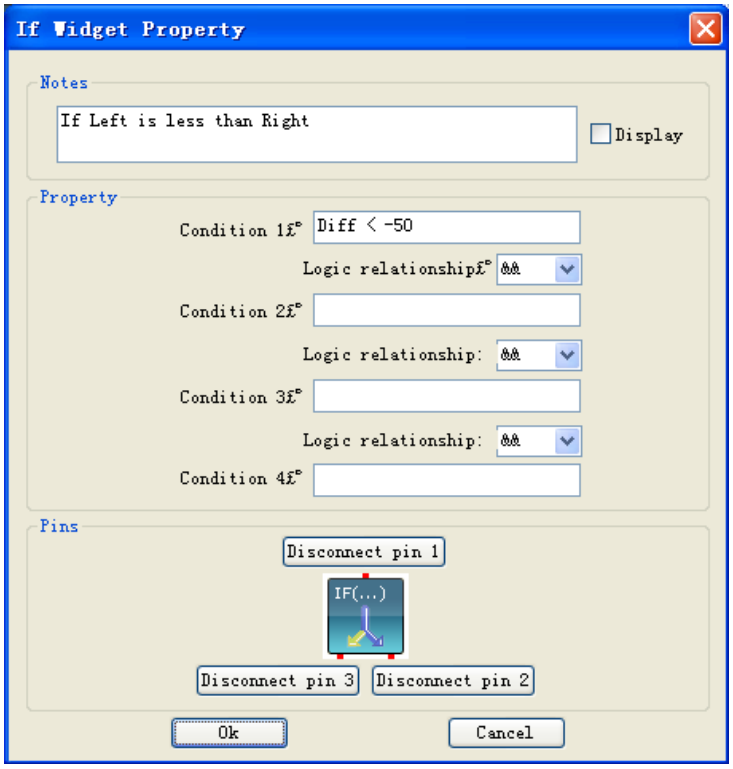


圖 5.40 右側光較強的判斷條件

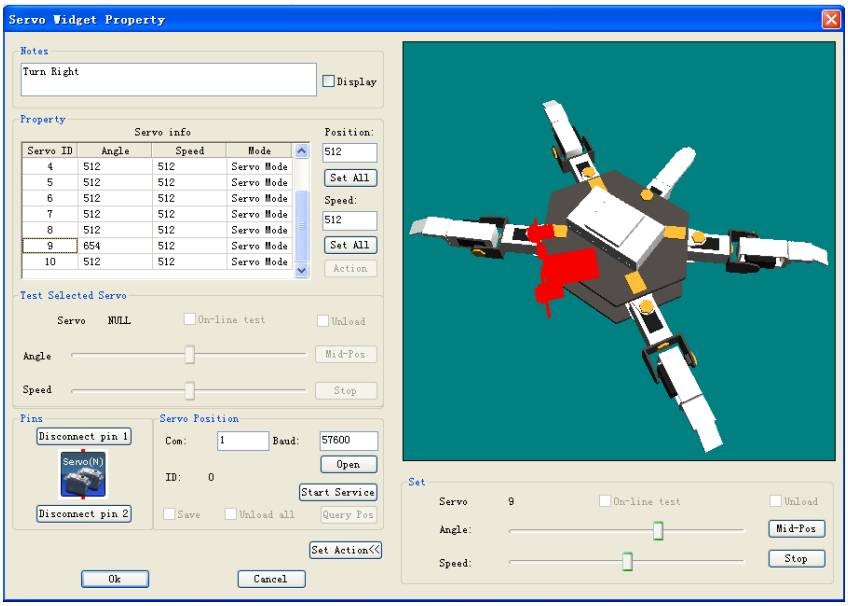


圖 5.41 頭部伺服馬達轉向右側

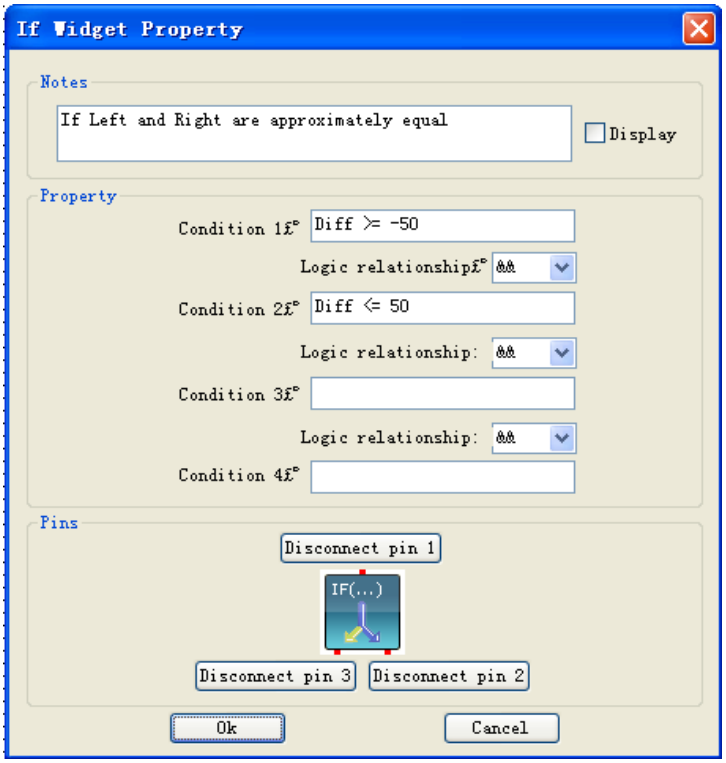


圖 5.42 左右光強大致相等的判斷條件

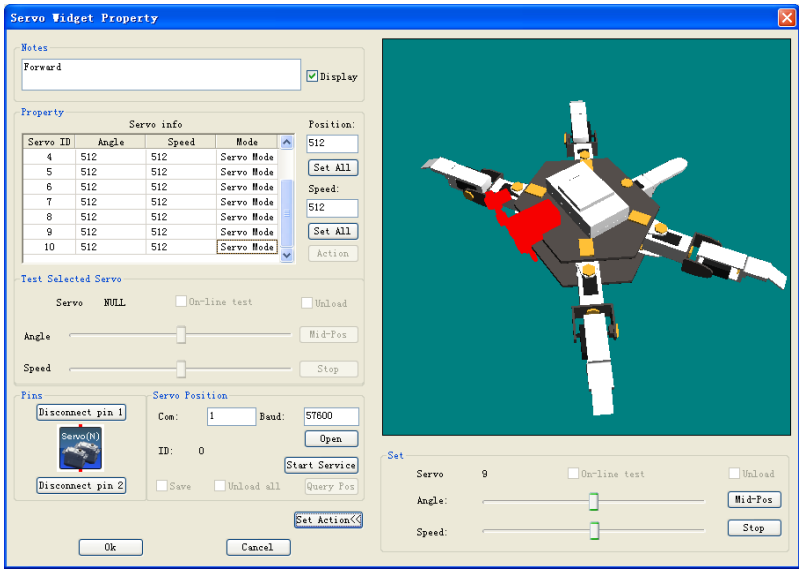


圖 5.43 頭部伺服馬達轉向中間

- **等待延遲**

在具體的實驗中，如果在短時間內向多個伺服馬達發生大量控制指令，會導致伺服馬達匯流排的擁堵，造成伺服馬達收到新指令的時間延後，最直觀的表現就是伺服馬達轉動與指令發送直接存在一個延遲，有一種“欲速而不達”的感覺。爲了保持伺服馬達資料匯流排的通暢，我們通常透過在伺服馬達指令發送間隙進行一定量的延時的操作。

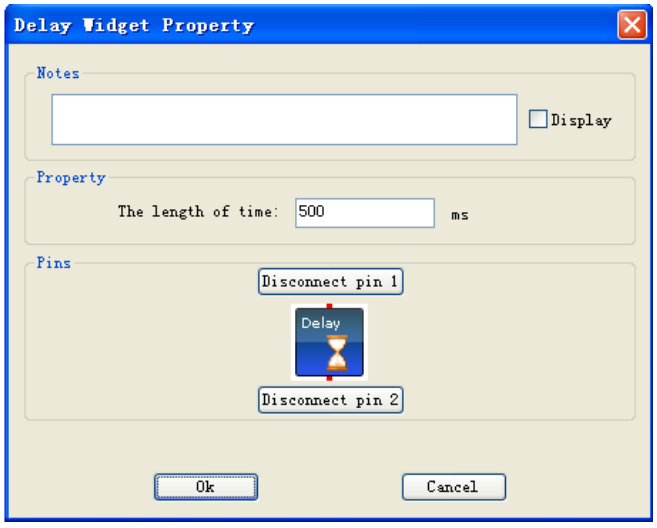


圖 5.44 等待延時

- **最終的程式流程及對應原始碼**

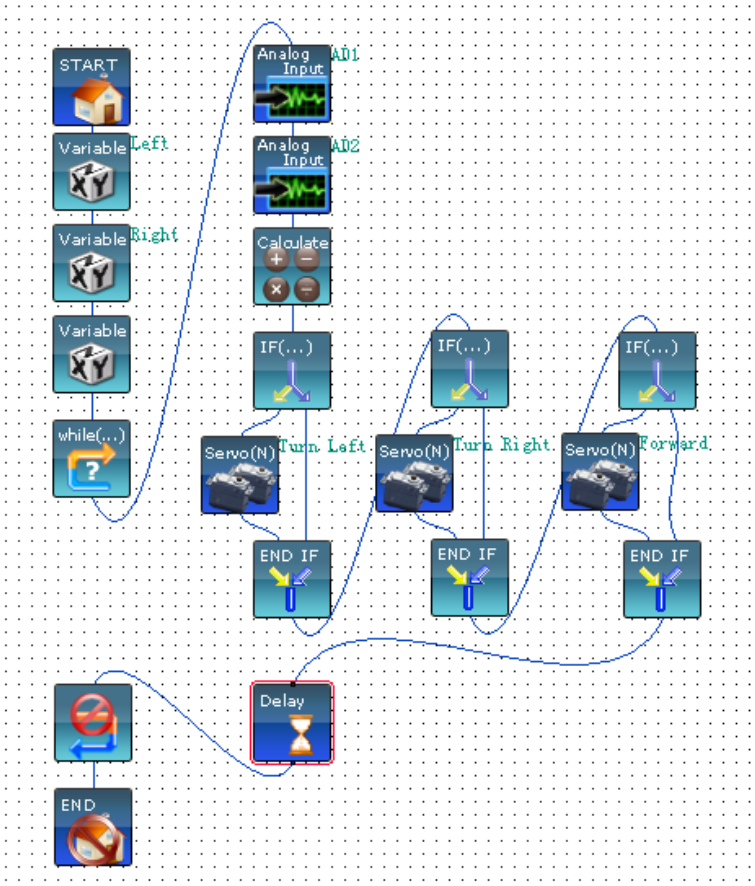


圖 5.45 程式流程

```
#include    "Apps/SystemTast.h"
int main()
{
      int    Right = 0;
      int    Left = 0;
      int    Diff = 0;
      DevicesInit();
      MFSetPortDirect(0x00000FFF);
      MFSetServoMode(1,0);
      MFSetServoMode(2,0);
      MFSetServoMode(3,0);
      MFSetServoMode(4,0);
      MFSetServoMode(5,0);
```

```
MFSetServoMode(6,0);
MFSetServoMode(7,0);
MFSetServoMode(8,0);
MFSetServoMode(9,0);
MFSetServoMode(10,0);
//Left
//Right
//Diff
while (1)
{
	//AD1
	Left = MFGetAD(0);
	//AD2
	Right = MFGetAD(1);
	Diff = Left - Right;
	//If Left is more than Right
	if (Diff > 50)
	{
		//Turn Left
		MFSetServoPos(1,512,512);
		MFSetServoPos(2,512,512);
		MFSetServoPos(3,512,512);
		MFSetServoPos(4,512,512);
		MFSetServoPos(5,512,512);
		MFSetServoPos(6,512,512);
		MFSetServoPos(7,512,512);
		MFSetServoPos(8,512,512);
		MFSetServoPos(9,405,512);
		MFSetServoPos(10,512,512);
		MFServoAction();
	}
	//If Left is less than Right
	if (Diff < -50)
	{
		//Turn Right
		MFSetServoPos(1,512,512);
		MFSetServoPos(2,512,512);
		MFSetServoPos(3,512,512);
		MFSetServoPos(4,512,512);
		MFSetServoPos(5,512,512);
		MFSetServoPos(6,512,512);
		MFSetServoPos(7,512,512);
		MFSetServoPos(8,512,512);
		MFSetServoPos(9,654,512);
```

```
            MFSetServoPos(10,512,512);
            MFServoAction();
        }
        //If Left and Right are approximately equal
        if ((Diff >= -50)&&(Diff <= 50))
        {
            //Forward
            MFSetServoPos(1,512,512);
            MFSetServoPos(2,512,512);
            MFSetServoPos(3,512,512);
            MFSetServoPos(4,512,512);
            MFSetServoPos(5,512,512);
            MFSetServoPos(6,512,512);
            MFSetServoPos(7,512,512);
            MFSetServoPos(8,512,512);
            MFSetServoPos(9,512,512);
            MFSetServoPos(10,512,512);
            MFServoAction();
        }
        DelayMS(500);
    }
}
```

5.5 任務 5：讓四足機器人跟蹤光源

在前面的任務裏，我們完成了四足機器人行走步態的設計，也完成了對光強感測器的處理實驗，下面我們將這兩個部分整合在一起，完成本章的最終目標。

我們的新程式可以在任務 3 的程式基礎上進行修改得來。在任務 3 中，僅僅是讓頭部伺服馬達對光源方向作出了回應，只需要將相應的轉頭動作替換成任務 4 中設計的行走和轉向步態即可。

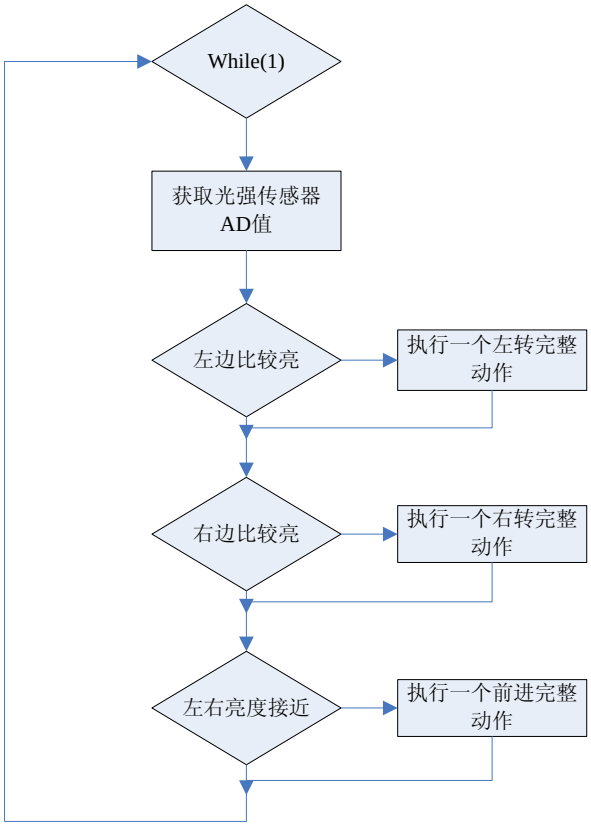


圖 5.46 程式邏輯

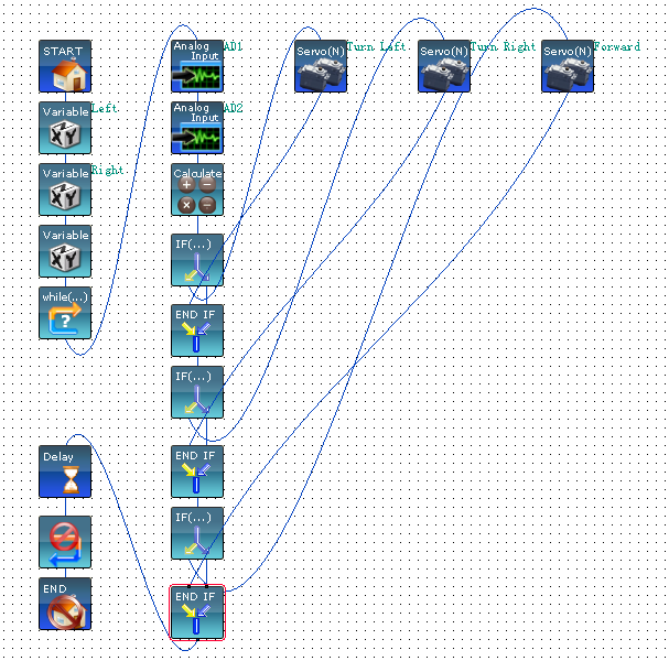


圖 5.47 重新排列圖示

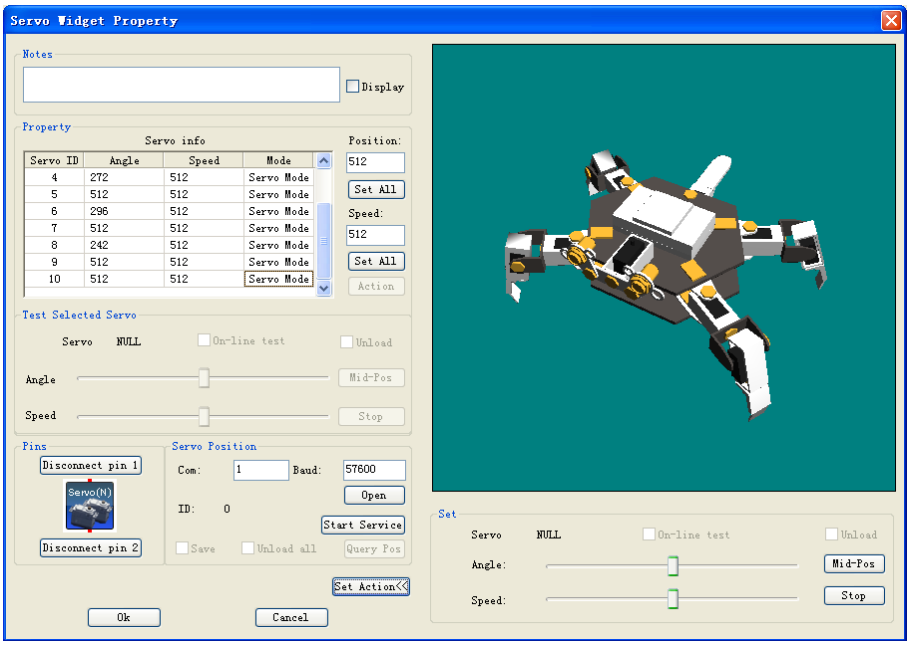


圖 5.48 行爲編輯

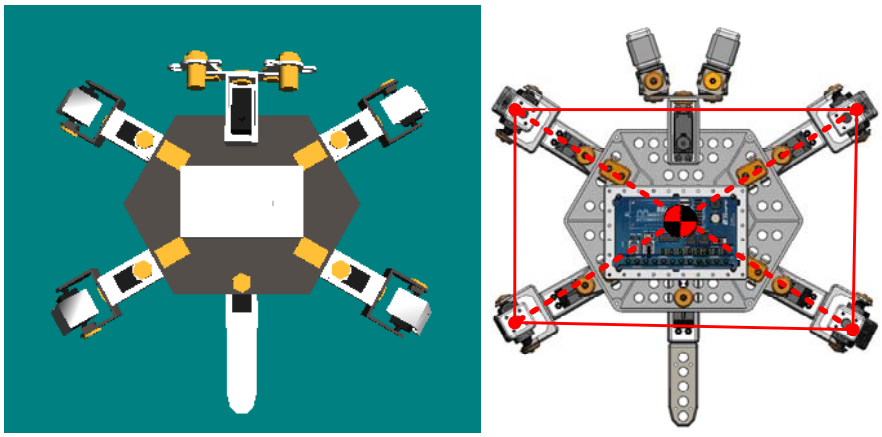


圖 5.49 初始步態

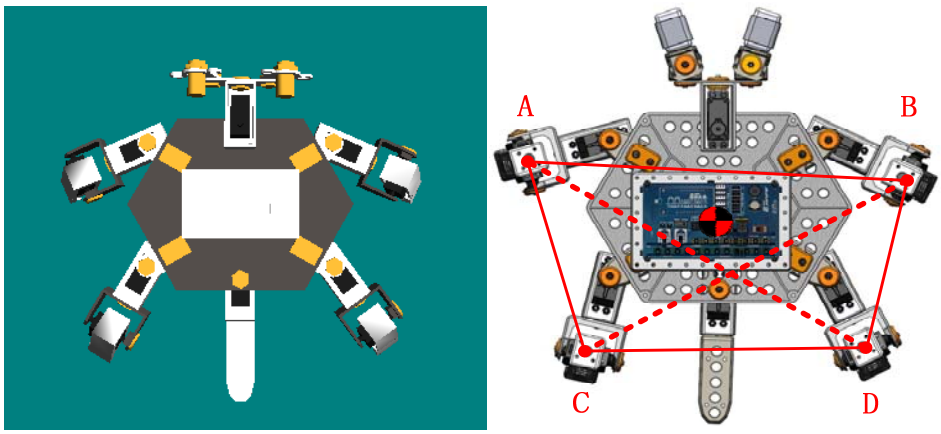


圖 5.50 步態一

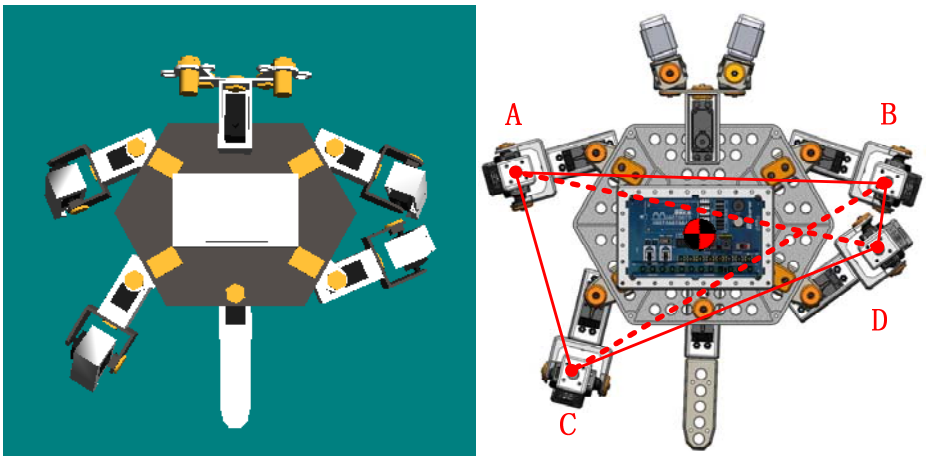


圖 5.51 步態二

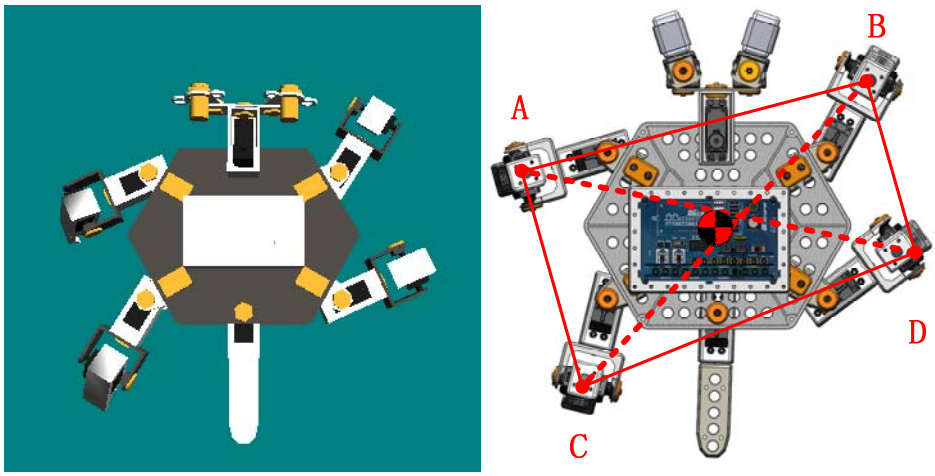


圖 5.52 步態三

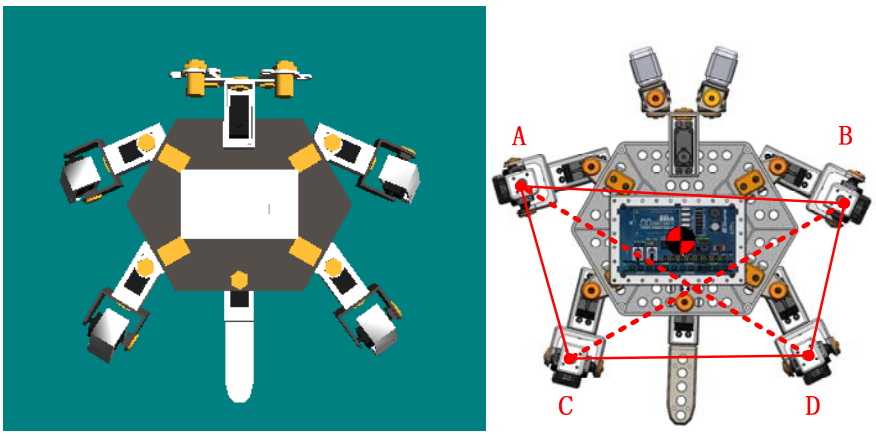


圖 5.53 步態四



圖 5.54 增加步態後的流程圖

第六章 擂臺賽機器人（無差別組）

6.1 任務 1：目標分析與計畫

本章的主要目的是帶您搭建出一台能夠參加無差別組的擂臺賽機器人，您可以用它來參加機器人擂臺賽，同時掌握這個環節所展現出來的機器人技術。

6.1.1 熟悉比賽規則

- **競賽目的**

機器人擂臺賽的目的在於促進智慧型機器人技術（尤其是自主識別、自主決策技術）的普及。參賽隊需要在規則範圍內以各自組裝或者自製的自主機器人互相搏擊，並爭取在比賽中獲勝，以對抗性競技的形式來推動相關機器人技術在大學生、青少年中的普及與發展。

機器人擂臺賽未來的發展目標是：比賽中，兩個使用雙腿自主行走的仿人形機器人互相搏擊並將對方打倒或者打下擂臺。

- **競賽內容概述**

在指定的擂臺上有雙方機器人和 5 個中國象棋棋子。 雙方機器人模擬中國古代擂臺搏擊的規則，互相擊打或者推擠。如果一方機器人整體離開擂臺區域或者不能再繼續行動，則另一方獲勝。如果雙方均未離開擂臺且都能自主移動，則在比賽時間結束後，推下擂臺上象棋棋子數量多的一方獲勝，否則雙方判爲平局。

比賽場地（即擂臺，如圖 6.1 大小爲長、寬分別爲是 2400 mm，高 150mm 的正方形矮台，臺上表面即爲擂臺場地。底色從外側四角到中心分別爲純黑到純白漸變的灰度。 場地的兩個角落設有坡道，機器人從出發區啓動後，沿著該坡道走上擂臺。場地四周圍 0.5 米處有高 400mm 的方形白色圍欄。比賽開始後， 圍欄內區域不得有任何障礙物或人。

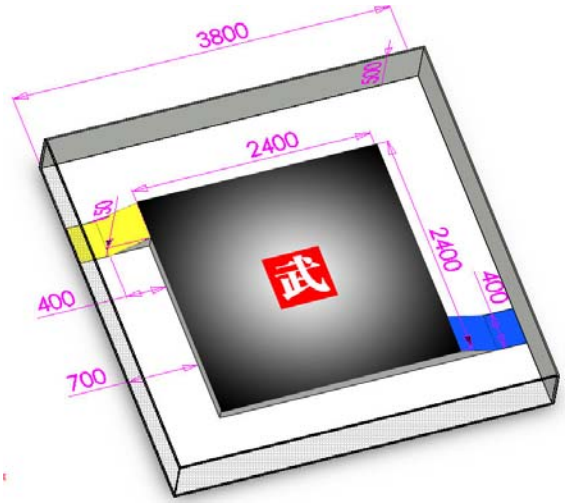


圖 6.1 擂臺賽場地示意圖

- **規則分析**

我們需要熟悉比賽規則，然後才能在比賽規則允許的範圍內，儘量讓我們的機器人具有別人不具有的優勢。對上述的比賽規則分析得到以下幾個重點：

1、分爲無差別組和仿人組，本章只討論無差別組。
2、需要從擂臺外的出發區上坡進入擂臺。
3、需要確保自己不掉下擂臺。
4、需要能夠找到棋子或者敵方。
5、需要能夠推動敵方，將敵方推下擂臺。
6、需要能夠將棋子推下擂臺。
7、需要避免自己被敵方推下擂臺。

6.1.2 我們的對策

根據比賽規則提出來的幾個要點是我們進行比賽平臺設計的依據。

1、自動上坡，到達比賽場地。

可以採用兩種方式實現爬坡行爲：

- 定時方式：根據場地調試，設置合適的延時，讓機器人進行爬坡；
- 傳感方式：安裝感測器或者攝影機，透過區分斜坡和場地的顏色來實現。

其中定時方式較爲簡單，傳感方式較難。對於無差別組機器人來說，採用這種方式即可。

2、需要確保自己不掉下擂臺。

需要有感測器進行擂臺邊沿的檢測，當發現機器人已經靠近邊沿立刻轉彎或者掉頭。擂臺和地面存在比較大的高度差，我們透過測距感測器很容易發現這個高度落差，從而判斷出擂臺的邊沿。如圖 6.2 所示，在機器人上安裝一個測距感測器，斜向下測量地面和機器人的距離，機器人到達擂臺邊沿時，感測器的測量值會突然間變得很大。由於紅外測距感測器使用方便，並且“創意之星”控制器最多可以接入 8 個紅外測距感測器，我們可以將它作爲首選方案。

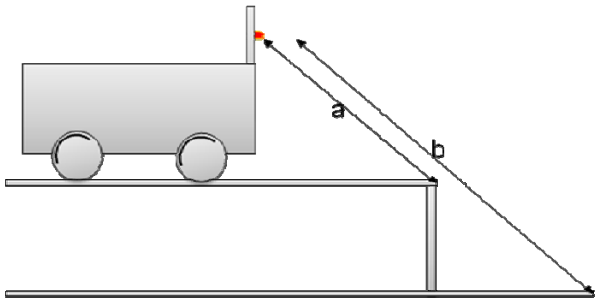


圖 6.2 擂臺邊沿檢測

擂臺地面是有灰度變化的，我們可以在機器人腹部安裝一些灰度感測器，來判讀機器人覆蓋區域的灰度變化，從而判讀機器人相對場地的方向。可以透過整體灰度值來判讀機器人的位置是不是靠近邊沿，如果機器人靠近邊沿就轉彎或者後退。

3、需要能夠找到棋子或者敵方。

根據比賽規則，我們可以知道棋子材質爲松木，重約 50~100g，直徑 70mm，高 44mm 的圓桶狀(兩個棋子疊放)，顏色爲松木原色，字體顏色爲紅色。根據第四章有趣的避障小車，我們可以得到一點啓發，避障小車可以發現障礙物，並且繞開。對於擂臺賽而言，棋子和障礙物沒有本質的區別，只是我們這回不是要繞開而是將之推下擂臺而已。我們同樣可以透過紅外接近感測器來發現障礙物（棋子）。敵方機器人，對於我們的機器人而言，就是一個外形比棋子大一點的障礙物，差別就在於棋子不會動，而敵方機器人會動而已。

不過，這裏使用紅外接近開關作爲尋找障礙物（敵方或者棋子）的方案並不算優秀，紅外接近開關的有效測量範圍是 20cm，20cm 之外的物體是察覺不到的。我們可以改成紅外測距感測器，它的有效測量範圍是 10-80cm，比較適合我們當前的使用場合。

我們可以佈置高低不同的兩個紅外測距感測器，用於區分棋子或者敵方機器人，如下圖所示：

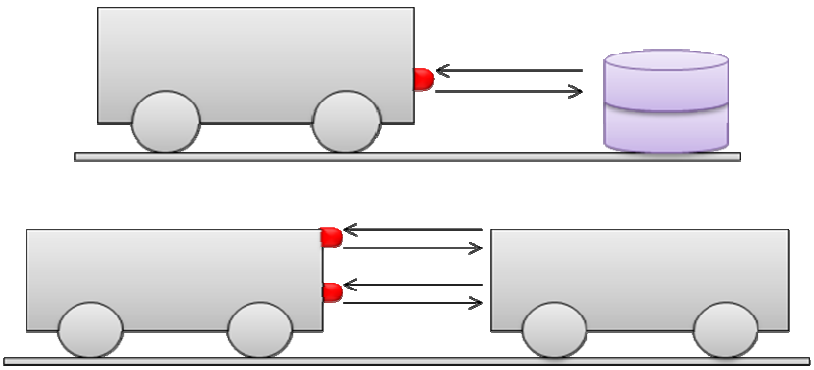


圖 6.3 前方感測器安裝示意圖

4、需要能夠將棋子推下擂臺。

推棋子時，機器人需要做三件事情：尋找棋子，推動棋子，自身定位。尋找棋子的實現主要依賴於灰度感測器和測距感測器。感測器的佈局至關重要，機器人必須能夠透過灰度感測器定位自己的位置，透過測距感測器確定棋子的位置。因此，結構設計時必須合理佈置感測器位置。比賽中輸贏的關鍵是把敵方推出場外，所以可以採用被動式策略尋找棋子，即機器人按照某種策略漫遊，如果檢測到棋子就把棋子推出場外，否則機器人繼續漫遊。當把棋子推出場外時，機器人會接近擂臺邊沿，此時機器人需要適當調整自己狀態(速度，位置)以防止掉出擂臺。這個過程中機器人的自身定位非常重要。

我們找到棋子之後，如何將它們推下擂臺呢？兩種方法：1、往前直走，不管怎麼都不轉彎，這樣一定能走到擂臺的邊沿，也能夠將棋子推下擂臺。2、透過識別場地的灰度，判斷出機器人的位置和方向，對準最近的邊沿前進。

方案一勝在簡單，程式實現難度小。但是可能出現機器人需要推動棋子走過超過半個場地才能到達邊沿的情況。這個過程可能出現敵方干擾、推擠，失敗的機率很高。

方案二實現的難度比較高一些，但是效率高。

5、需要能夠推動敵方，將敵方推下擂臺。

我們可以想像，兩隻鬥牛相互推擠，贏的一定是力氣比較大的一方。

我們需要先考慮，什麼因素會影響小車的推力。在現實生活中我們會看到兩種現象：1、一輛汽車在爬坡，但是動力不足，反而慢慢從坡上滑了下來。2、汽車在泥地上面行走，但走不快，因爲輪子總是在打滑。

上面兩個例子可以得到影響汽車行進效果的因素：1、動力不足。2、摩擦力不夠。如果動力不足，我們小車可能會被敵方的小車推得輪子倒著轉。如果摩擦力不夠，小車在推擠的時候輪子會打滑，輪子一直在轉，但小車就是不能往前走。

推動敵方和推動棋子要做的事情是一樣的，即：尋找敵方→推動敵方→自身定位。不同的是，推動敵方需要更大的動力，而且自身定位更加重要。如果己方被敵方推動時，己方就必須能夠及時擺脫或者進行抵抗。

所以，在比賽規則允許的條件下，儘量增加小車的重量、動力輸出、輪子的接地面積。我們可以使用四輪驅動方案，兩輪驅動方案重量和推力都不夠。六輪驅動方案不好轉彎，機器人不夠靈活。

6、需要避免自己被敵方推下擂臺。

如果我方機器人在前進的時候被敵方從後面推擠，並且我方的機器人沒有察覺，會出現什麼情況呢？

我方機器人的動力方向和敵方機器人的動力方向剛好一致，敵方不費吹灰之力就可以將我方機器人推下擂臺。所以我們需要能夠察覺這在被推擠的狀態，並且能夠透過掉頭、轉彎、後退等手段避開機器人的推擠。

6.1.3 任務規劃

1、熟悉本比賽結構所需要零件的使用方式。比如灰度感測器、紅外測距感測器等。

2、搭建機器人、佈置感測器。需要在規則允許的條件下搭建機器人，透過上述的分析，我們將機器人的結構定義爲一個四輪驅動的小車。機器人的腹部應該安裝灰度感測器陣列，用於判讀場地灰度變化。前方應該安裝紅外測距感測器，用於判讀棋子、敵方機器人和擂臺邊沿。

3、寫程式，逐個實現上述 7 個實現要點，並作練習。

4、模擬比賽練習。

6.2 任務 2：熟悉場地、標定感測器

6.2.1 比賽場地

比賽場地（即擂臺，如圖 6.4 所示）大小爲長 2400 mm，高 150mm 的矩形矮台，臺面即爲擂臺場地。底色從外側四角到中心分別爲純黑到純白漸變的灰度。場地的材質爲木質，場地表面最大承重能力 50kg。場地的兩個角落設有坡道，坡道入口處的區域爲出發區。機器人從出發區啓動後，沿著該坡道走上擂臺。

場地四周圍 0.5 米處有高 400mm 的方形白色圍欄。出發區及坡道用正藍色和正黃色顏色塗敷。出發區平地尺寸爲 300x400mm。出發坡道水準長度爲 400mm，寬度爲 400mm，坡道頂端高度與擂臺平齊，即 150mm。

象棋棋子的材質爲松木，重約 50~100g，直徑 70mm,高 44mm 的圓桶狀(兩個棋子疊放)，顏色爲松木原色，字體顏色爲紅色，如圖 5.2 所示。

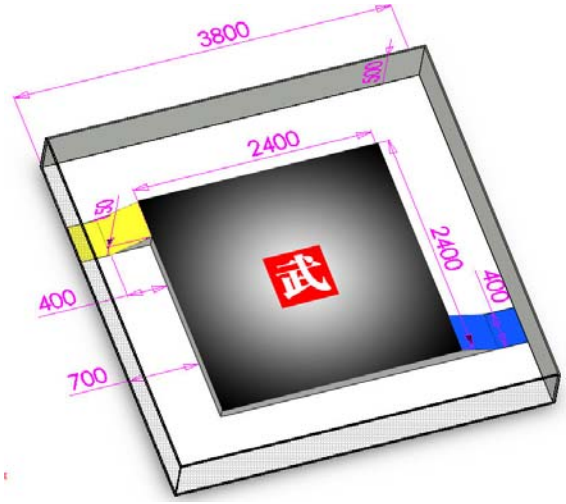


圖 6.4 擂臺賽場地



圖 6.5 象棋棋子示意圖

場地最重要的部分是擂臺上黑白漸變的區域，這是比賽得以進行的關鍵。簡單的說，機器人要想贏得比賽，就得在這塊區域裏行動自如，隨心所欲。如何做到“行動自如，隨心所欲”呢？機器人必須能隨時確定自己所處的位置。要想確定自己的位置，機器人就必須有雪亮的眼睛。在擂臺賽中，機器人的眼睛就是灰度和測距感測器。接下來我們介紹感測器的標定。

6.2.2 標定灰度感測器

我們在結構搭建的時候會用到灰度感測器和紅外測距感測器。您如果對這兩個感測器的工作原理還不清楚的話，請重新學習第二章的相關部分。

受製造技術、材料的限制，很難保證兩個感測器有完全一致的性能。在實際使用過程中會發現，即使以同樣的標準使用兩個不同的灰度感測器測量灰度，得到的資料也會有一定的落差。

爲了消除這種差異性，需要對感測器進行標定。就像一個兩眼視力不一樣的人，需要佩戴兩個鏡片度數不一樣的眼鏡一樣。我們這裏的標定，目的是建立灰度感測器的值和場地區域的對照表，以方便後續編寫控制程式時使用。

我們的結構上使用 4 個灰度感測器進行場地灰度測量，如圖 6.6 所示，1、2、3、4 號感測器採集到的灰度值是不一樣的，我們對比這幾個感測器的值就能夠知道當前機器人的方向。場地的灰度是梯度變化的，爲了讓這幾個感測器的值有儘量大的差值，感測器的佈置應該儘量拉開距離，感測器菱形覆蓋的面積應該儘量大。

灰度感測器的標定需要同時標定一組即將使用的感測器——即同時標定結構中的四個灰度感測器。此時可以先搭建一個簡易的機器人底盤，固定好感測器。我們這裏採用了菱形佈局，如圖 6.6。

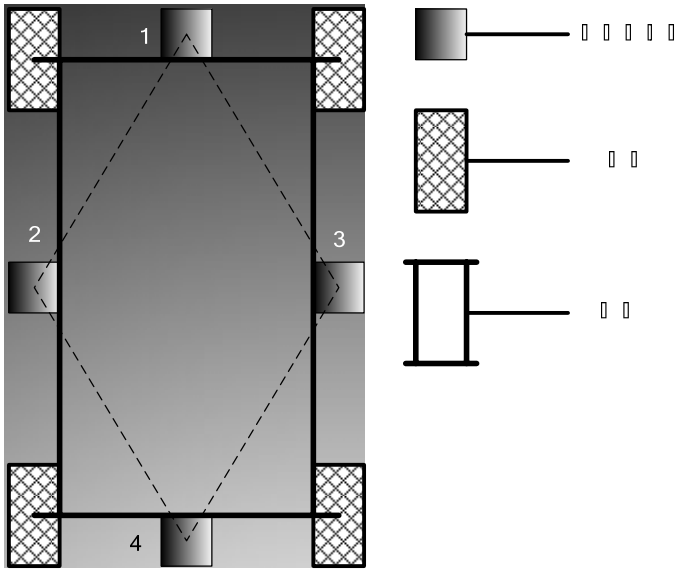


圖 6.6 灰度感測器佈局示意圖

灰度感測器的標定可以採用三種方法：平行邊沿標定，灰度梯度方向標定，隨機位置標定。

平行邊沿標定方法如下：

1) 按 300mm 等分場地邊沿（如圖 6.7，圖中白色方塊表示機器人放置位置），然後從一邊開始，依次將機器人放置(和邊緣平行)在等分點上，用 NorthSTAR 查詢灰度感測器的值並做相應記錄；

2) 將機器人掉頭後，從另一邊開始，依次在每個等分點上查詢灰度值並記錄。

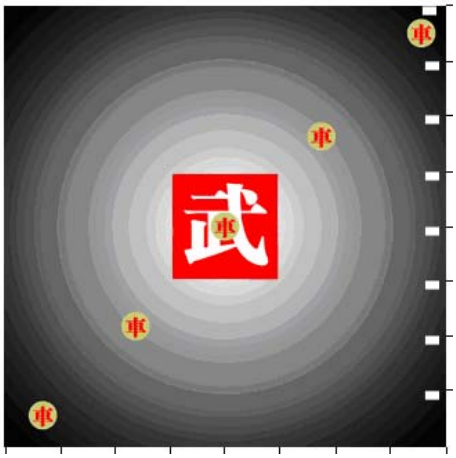


圖 6.7 等分場地邊沿

灰度梯度標定法：

1) 將場地對角線的一半等分爲 6 份（如圖 6.8，圖中白色圓塊表示機器人放置位置）；
2) 將機器人面向場地中心放置在圖中 6.7 中的最外側的等分點上，記錄灰度值；
3) 依次將機器人放置在其他的等分點記錄灰度值。

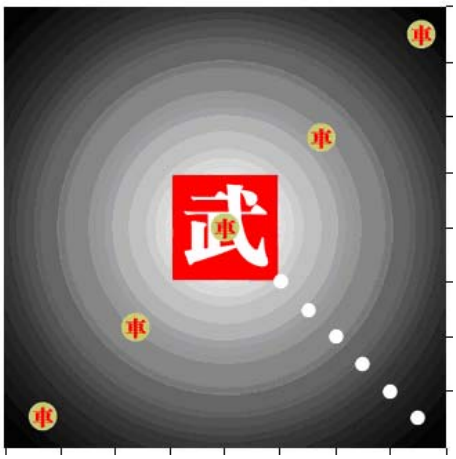


圖 6.8 等分場地對角線

隨機標定法：將機器人以非規則的方式（和邊沿不平行，沒有正好面向中心）放置在場地的不同位置，採集幾組灰度值，記錄並觀察四個灰度感測器的值和機器人的位置關係。

記錄資料後，計算機器人處在據中心不同距離處的灰度均值，感測器的最大最小值，前後灰度之間的差值，障礙物和機器人之間距離與感測器值之間的關係等資料，以備程式設計時使用。表一、表二爲一組場地實測資料。灰度數值受灰度感測器的一致性，安裝位置和環境光的影響較大，表中資料僅供參考。

表一（等分場地邊沿）

機器人側邊和邊緣平行時灰度數據									
前	後	右	左	前-右	前-後	後-右	前-左	右-左	均值
235	207	231	265	4	28	-24	-30	-34	234.5
367	348	319	412	48	19	29	-45	-93	361.5
410	404	349	453	61	6	55	-43	-104	404
405	424	355	457	50	-19	69	-52	-102	410.25
356	394	324	425	32	-38	70	-69	-101	374.75
275	304	267	336	8	-29	37	-61	-69	295.5
231	233	232	274	-1	-2	1	-43	-42	242.5

表二（等分場地對角線）

機器人面向中心灰度數據									
前	後	右	左	前-右	前-後	右-後	前-左	右-左	均值
244	207	234	261	10	37	27	-17	-27	236.5
339	283	305	355	34	56	22	-16	-50	320.5
493	453	425	505	68	40	-28	-12	-80	469
551	537	482	567	69	14	-55	-16	-85	534.25
612	614	537	621	75	-2	-77	-9	-84	596

注：表中“前”表示安裝在前面的灰度感測器，“前-後”表示前面感測器減去後面感測

器的差值。

根據表中資料可以看出，除左邊的灰度感測器，其他三個灰度感測器的值一致性較好。可以根據這三個感測器的值來判斷機器人當前的方向。另外可以根據 4 個灰度感測器的平均值來判斷機器人在場地上的位置。接近場地中心時，平均值較小，接近場地邊沿時，平均值較大。得到這些邏輯，就可以在編寫控制程式時使用。詳細邏輯見後續章節。

6.2.3 標定紅外測距感測器

在開始使用 GP2D12 之前，您需要詳細閱讀 2.3.1 小節有關於紅外測距的部分。

紅外測距感測器數值和實際距離不是簡單的線性關係，GP2D12 在 10cm 位置輸出值最大，10cm 以下的距離是感測器的盲區，不建議使用。10cm-20cm 之間是最靈敏的區域，所以，我們需要保證結構中的 GP2D12 和地板的距離保持在 10cm-15cm 之間。

從 2.3.1 小節，可以看到 GP2D12 的一般性測距使用需要考慮到比較多的細節。在這個比賽平臺上，我們只需要判斷機器人是不是到達邊緣即可。比賽結構搭建完成、感測器安裝位置確定後，將機器人放置於擂臺的不同位置，讀取 GP2D12 的 AD 值，將這些 AD 值取平均值。將機器人移至擂臺邊緣，當感測器紅外射線和地面的交點 A 離開擂臺邊緣時，感測器 AD 值會大幅度變化，測量幾次取均值即爲機器人檢測到擂臺邊緣的 AD 值。

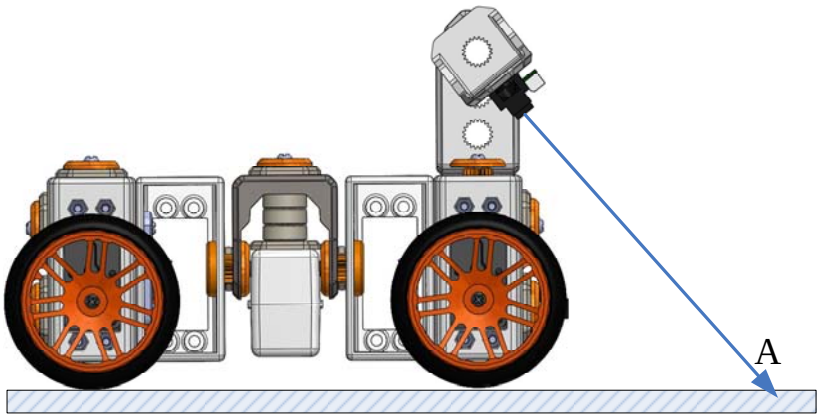


圖 6.9 紅外測距感測器的安裝示意圖

6.3 任務 3：結構設計和感測器佈置

6.3.1 結構設計

我們在 5.1 節已經得出了機器人結構的設計要求：重心低，品質大，動力強勁，行動靈活，感測器合理佈置。我們將遵照以上要求設計比賽結構。

本節的範例結構僅作參考。該結構能夠符合比賽規則的要求，能夠正常完成比賽，但一定不是最優化的方案。我們期望您能夠爲廣大機器人擂臺賽參與者提供更完美的解決方案。

在開始結構搭建前，您需要先閱讀《創意之星機器人套件組裝指南》，熟悉“創意之星”的零件種類和連接方式。您只有做到對“創意之星”的零件使用心裏有數之後，搭建過程才會順利；如果盲目搭建，可能會出現搭建步驟錯誤，反復拆裝的情況發生。

我們設計的結構如圖 6.10 所示。

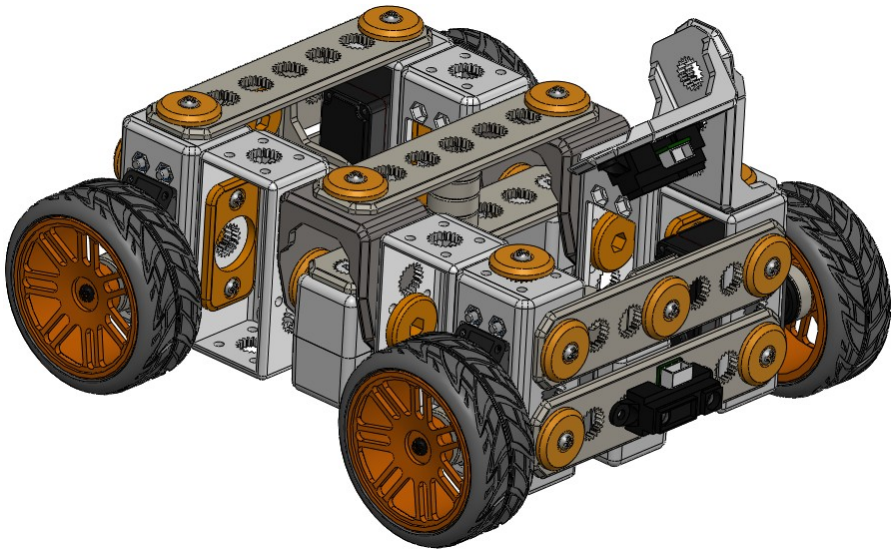


圖 6.10 無差別組輪式底盤

整個比賽結構搭建可以分爲以下幾個重要步驟：

（一）、搭建框架

先按照圖 6.11 所示，搭出兩個側框架。將 A、B、C、D、E、F 的幾個連接花鍵裝到框架上，注意需要將 M3 螺母裝到花鍵裏面，如圖 6.12 所示。

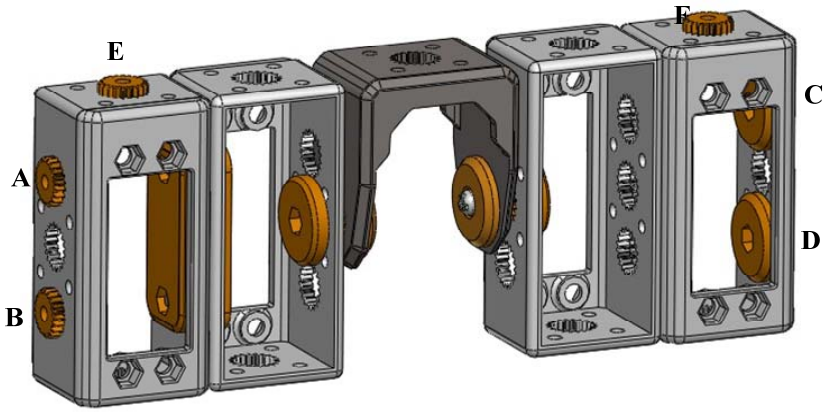


圖 6.11 結構框架

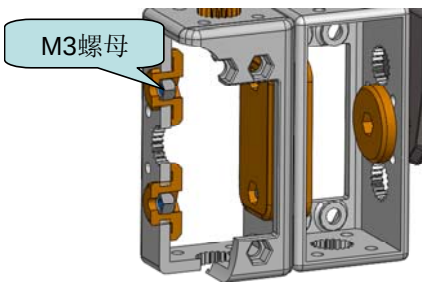


圖 6.12 螺母安裝

（二）、安裝驅動伺服馬達和輪子

將驅動伺服馬達安裝到側框架上，這個過程中您需要注意伺服馬達不能裝反，上一步預

先安裝上去的花鍵裏的螺母不能掉出來。

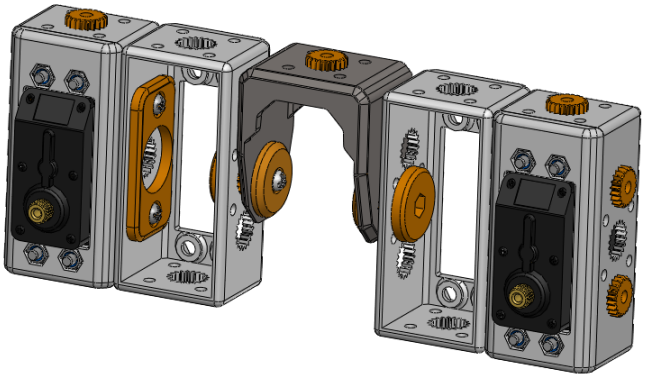


圖 6.13 安裝伺服馬達

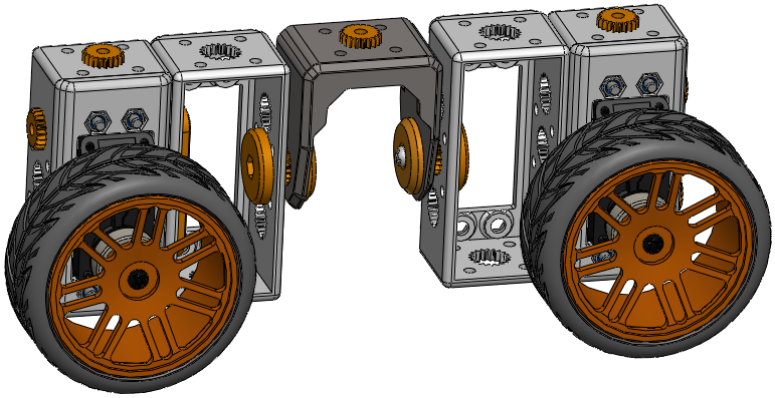


圖 6.14 安裝輪子

（三）、整體組裝

將兩個側框架透過 I7 零件連接起來，構成一個完整的底盤框架，如圖 6.15 所示。

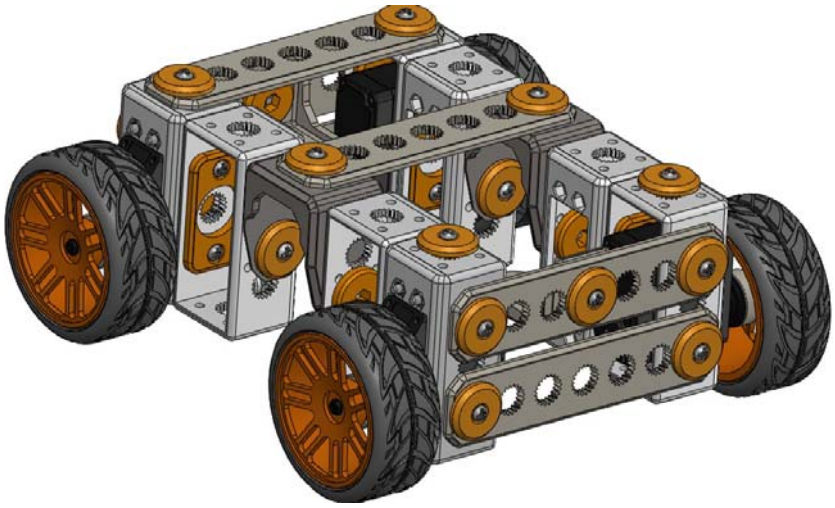


圖 6.15 底盤組裝

至此，底盤搭建完成，接下去需要設計灰度感測器和紅外測距感測器的安裝方式。

6.3.2 感測器佈置

如圖 6.16 所示，我們利用一個向前方傾斜的感測器來進行擂臺邊緣的檢測，避免機器人掉到擂臺底下。這個測距紅外探測點必須和機器人主體之間有一定的距離，讓機器人有反應的時間。

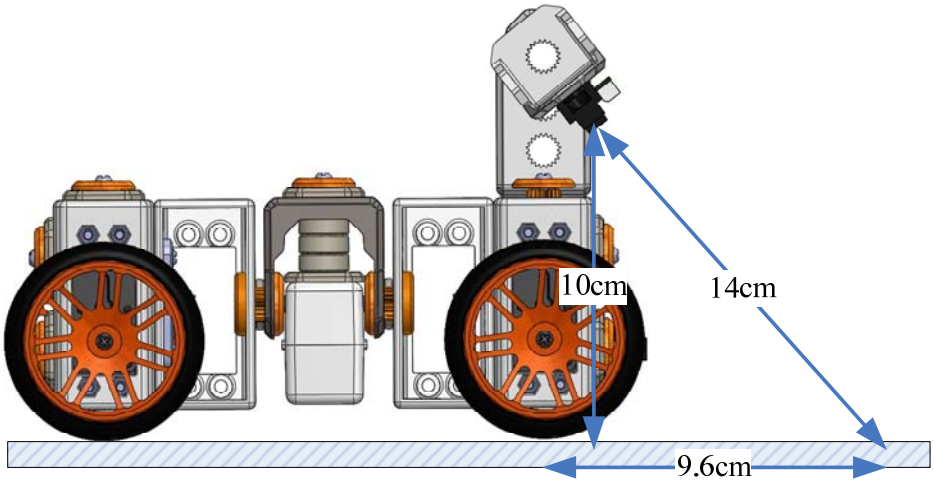


圖 6.16 紅外測距感測器的安裝示意圖一

如圖 6.17 所示，我們可以在機器人的正前方安裝兩個紅外測距用於敵方和障礙物的探測。棋子的高度是 44mm，機器人必然會比棋子高，因此將兩個感測器安裝在不同的高度。我們在 28mm 和 59mm 的位置分別安裝一個紅外測距感測器，定爲 A 和 B 感測器。在比賽過程中，A 探測到障礙物而 B 沒有，則說明障礙物是棋子。如果 A 探測到障礙物 B 也同時探測到障礙物，且兩者探測到的障礙物距離差距不大，說明障礙物可能是敵方機器人。當然，這個過程可能會有其他因素的干擾，您可以在實際調試過程中設法解決這些問題。

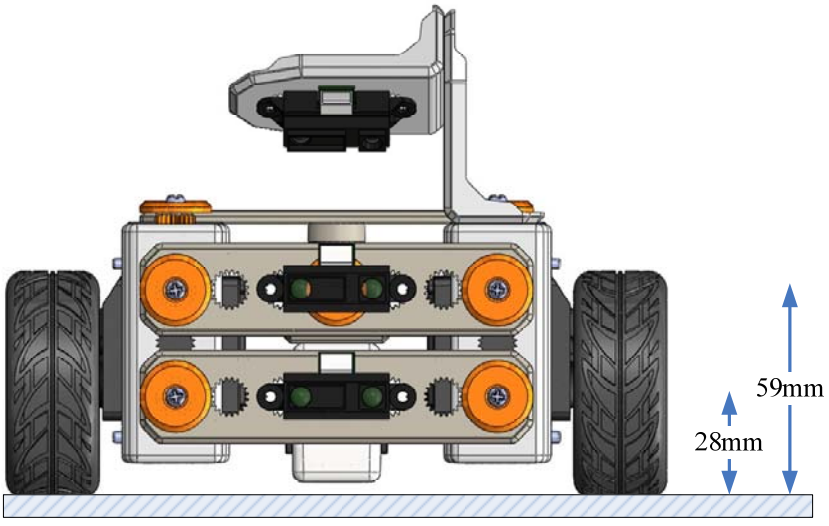


圖 6.17 紅外測距感測器的安裝示意圖二

灰度感測器根據本節開始的分析，採用菱形分佈，如圖 6.18 所示。

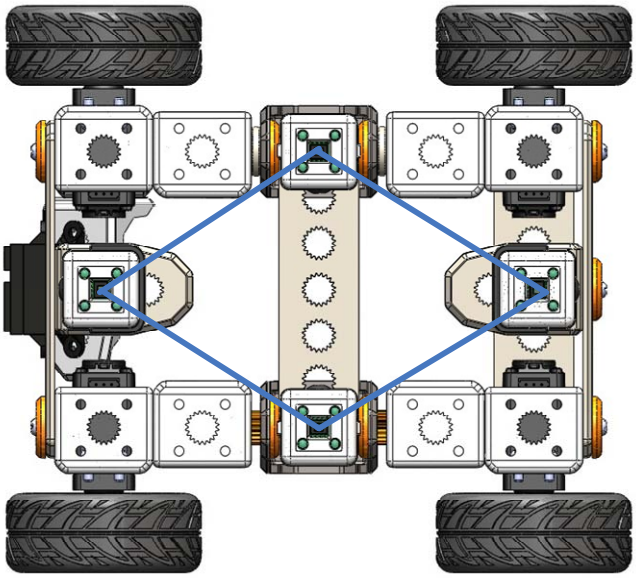


圖 6.18 灰度感測器佈置示意圖

上述感測器佈置方案是我們提供的一種可行，並且透過驗證的方案，希望可以作爲您搭建結構的參考。“創意之星”的結構特點是“無所不能”，您可以盡情發揮您的創意，用“創意之星”設計出更加合理的方案。

6.4 任務 5：推棋子練習

棋子品質較輕，推動比較容易。功能實現關鍵是棋子的檢測和邊緣檢測。這裏實現的邏輯以保證機器人不會掉落擂臺爲目的，即機器人不主動去尋找棋子，而是在場內漫遊，發現棋子後向前推動，推出場地後後退、左轉，然後繼續漫遊，實現邏輯如圖 6.19 所示。由於推動棋子過程中需要根據自身位置做相應的處理，這裏把推動棋子的過程單獨封裝，具體的實現邏輯如圖 6.20 所示。圖 6.19 和 6.20 中的延時需要根據場地實際調試獲取。

如果要實現主動尋找棋子的策略，就需要改變圖 6.19 中的邏輯，讓機器人在沒發現棋子時，主動尋找棋子。無論採用主動模式還是被動模式，機器人檢測擂臺邊沿並做出相應動作的行爲必須擁有最高優先順序。

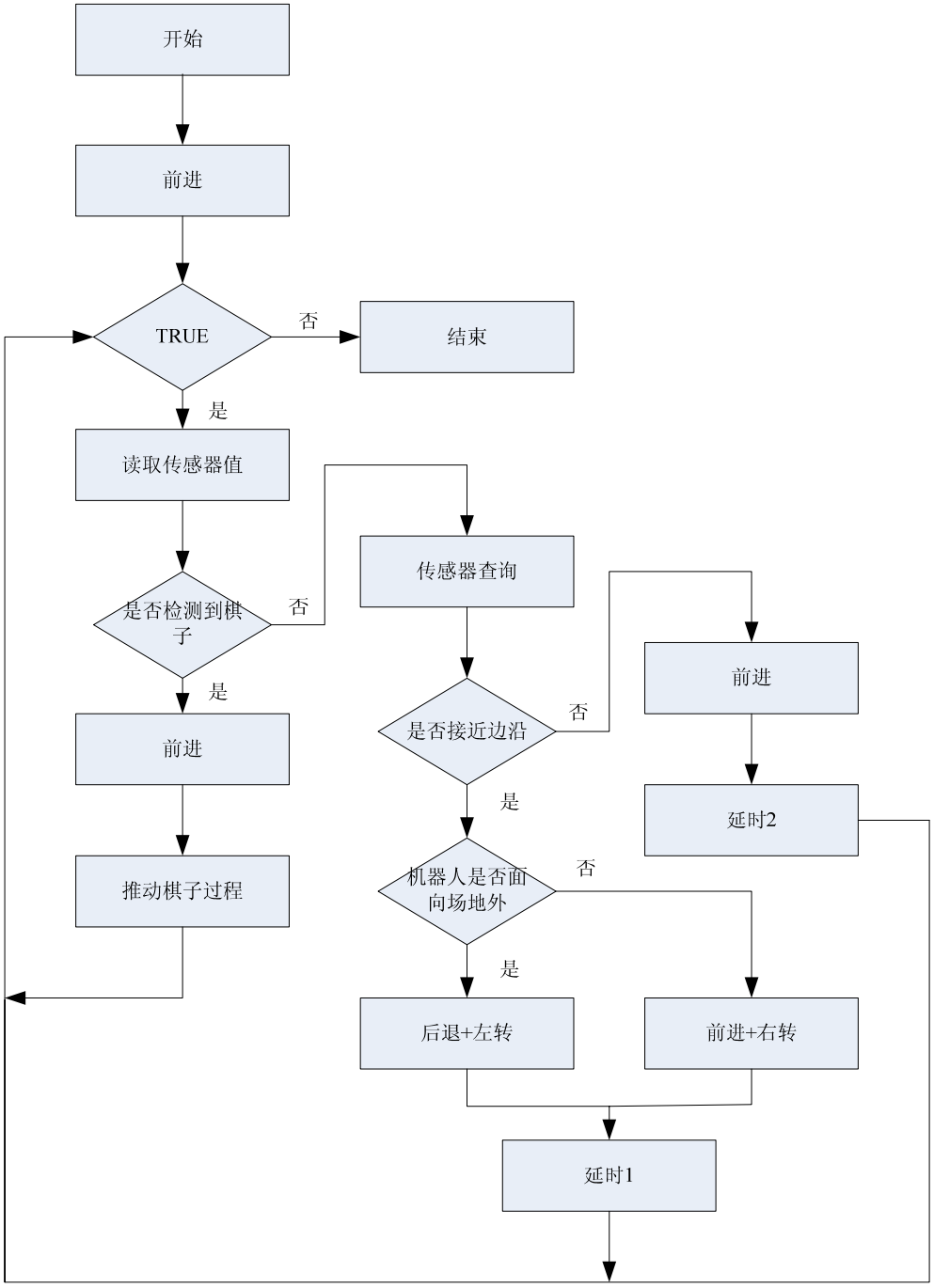


圖 6.19 推棋子實現邏輯

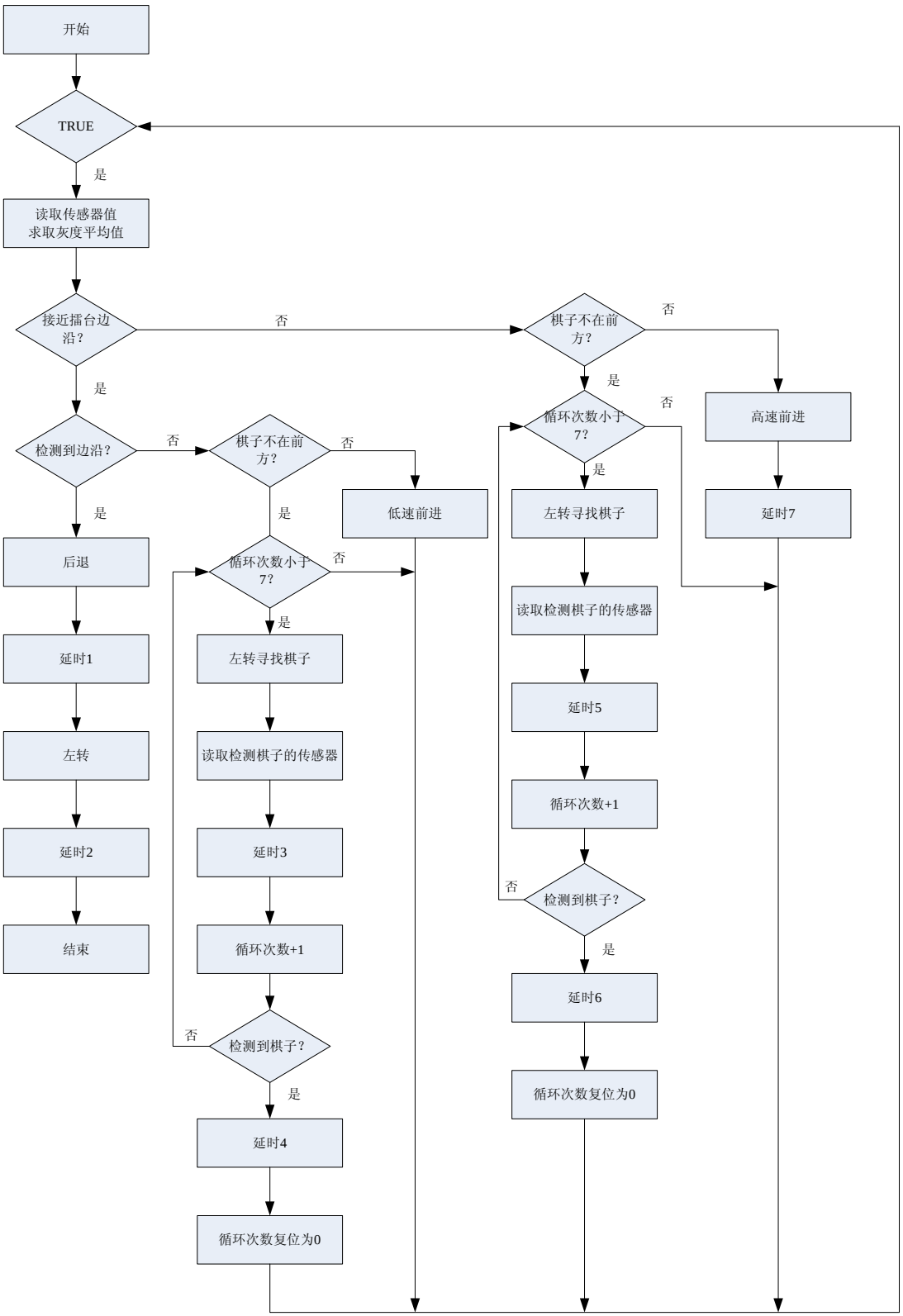


圖 6.20 推動棋子過程邏輯實現

6.5 任務 6：實戰對抗練習

擂臺賽的關鍵，就是在己方不出擂臺的情況下，將敵方推出場外。所以推動敵方的策略是最爲關鍵的。由於敵方機器人同樣擁有動力裝置，推動時必然遭受巨大阻力。因此，在檢測到敵方機器人時，必須以最大的動力迅速將敵方推出場外。考慮到不同機器人動力的差異性，需要同時具備應對己方機器人動力不足，被敵方推動的情況。如圖 6.21 爲實戰對抗的實現邏輯，圖 6.22 爲推動敵方機器人過程的實現邏輯。

需要注意的是，圖 6.21 中的邏輯中加入了爬坡環節。爬坡作爲比賽中最先完成的行爲，在編程實現實戰對抗策略時，需要和實戰對抗行爲銜接起來。此外，這裏對機器人即將把敵方推出場外的情況做了特殊處理——高速前進+延時+高速後退+延時。這是爲了快速將敵方推出場外，而且保證自己不會摔到場外。

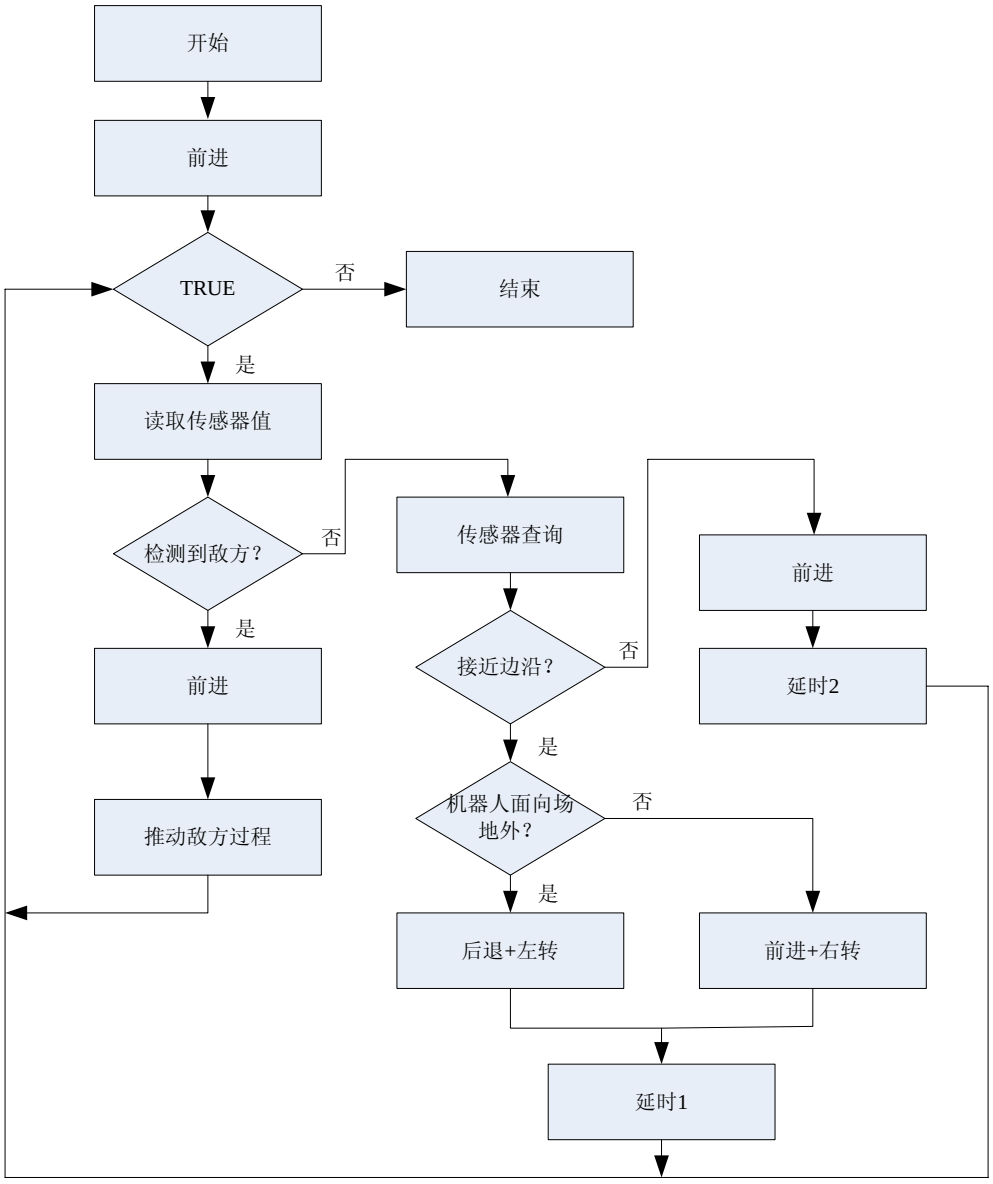


圖 6.21 實戰對抗實現邏輯

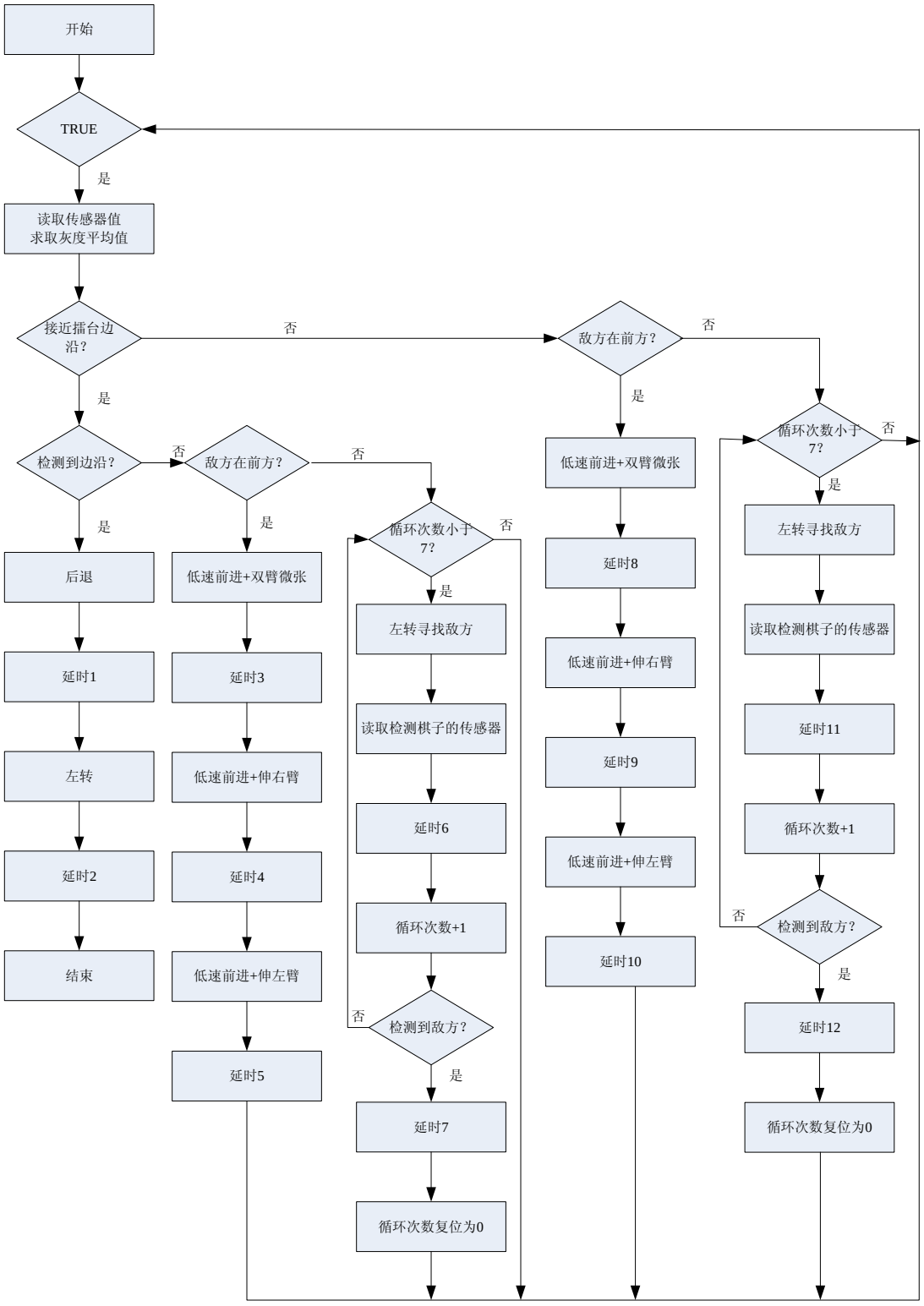


圖 6.22 推動敵方過程邏輯

6.6 擴充任務：改進競賽能力

調試中發現，機器人的重量對機器人行爲的影響較大。動力性能相同的情況下，品質輕的機器人很難和品質大的機器人對抗。另外，邊緣檢測感測器的位置需要保證機器人不會摔倒在場外，灰度感測器的安裝高度對其性能影響較大。所以我們來進行以下改進：

1) 增加機器人重量。比賽規定機器人重量不超過 3kg，在此條件下，給機器人增加配重。配重可以選用電池或者其他物體。安裝配重時要儘量保持機器人的重心最低。
2) 調整邊緣檢測感測器位置。比賽規定機器人在場地上的投影尺寸不超過 300x300mm，在此條件下，將邊緣感測器安裝位置盡可能的伸出底盤外側。或者架高感測器，讓感測器探測紅外線與地面交點儘量遠離機器人本體。
3) 調整灰度感測器距地面高度。灰度感測器距離地面太遠，反射光線太弱；距地面太近，反射光線太強。反射光過強或者過弱都會導致機器人自身定位不準確。調整時，可以設定幾個位置，調試並找出沿著場地灰度梯度方向變化範圍最大的位置。

6.7 擴充任務：增強速度和力量

使用 CDS5500 機器人伺服馬達的馬達模式來驅動機器人，機器人動力較小，在推動敵方機器人時會比較吃力。此時，我們需要增強機器人的速度和力量。實現方式有兩種：

1) 採用 2.4.3 提到的馬達驅動模組+直流有刷馬達方案，即使用 BDMC1203 驅動模組和 FAULHABER 2342 24CR 直流馬達。
2) 採用 Sabel2x10 驅動器+直流有刷馬達的方案。

這兩種方式都可以增強機器人的動力。具體的使用方法請參考 2.4.3，2.4.4，2.4.5 章節。

第七章 擂臺賽機器人（仿人組）

7.1 任務 1：目標分析與計畫

7.1.1 熟悉比賽規則

- **比賽規則摘要**

a) 對於仿人機器人組競賽，機器人可明顯地分爲兩部分，即機器人身體部分和移動底盤，比賽時機器人身體部分和移動底盤部分在賽前需要分開稱重，比賽時必須透過雙腿和底盤固連在一起；各項限制如下：

i. 機器人身體部分需具備頭部、軀幹、四肢幾個基本的人體特徵，必須具備兩個手臂（每個手臂不少於 3 個動力關節）；一個具備俯仰或旋轉關節的腰部（旋轉極限角度不能超過 180 度），以及明顯的雙腿特徵，並且重量不低於 1.5kg；

ii. 移動底盤可以使用輪式或履帶式驅動，其在場地上的投影尺寸不得超過 190x180mm 的長方形，高度不超過 100mm；如果移動底盤有外側壁的話，所有外側壁面必需與地面垂直，即禁止使用能夠插入對方底盤下的鏟形裝置；

iii. 完整的機器人整體高度不低於 400mm，重量不超過 3kg，機器人的兩條手臂肩關節頂部距地面的高度不低於 300mm。

iv. 機器人身體必須穿著裝飾性的服裝。鼓勵自行製作有特色的服裝。

v. 本組別機器人在參賽前必須透過資格認定。資格認定的方法是：機器人從出發區出發，運動到場地中心，將中心的白色粗糙圓柱體（如圖 5 所示）舉起（整體離開地面），並能將其放下。每個機器人有三次嘗試機會，如果三次嘗試均失敗，則取消該機器人的參賽資格。

- **規則分析**

和無差別組機器人相比，仿人組除了要完成爬坡、推棋子和推敵方三個任務外，還需要進行資格認定。資格認定是指：機器人從出發區出發，運動到場地中心，將放置在場地中心的白色粗糙圓柱體舉起（整體離開地面），並能將其放下。只有完成此任務，機器人才有參賽資格。要完成資格認定，機器人需要完成兩個子任務：尋找中心區域，舉圓柱。

和無差別組相比，比賽規則對仿人組機器人的要求更加嚴格，例如：身體部分需具備頭部、軀幹、四肢幾個基本的人體特徵，完整的機器人整體高度不低於 400mm，兩條手臂肩關節頂部距地面的高度不低於 300mm。按照這種要求搭建出來的機器人，重心必然會比無差別組機器人高很多。在這種情況下，機器人完成任務時就需要調整重心來防止摔倒。

爬坡時，機器人要盡可能的降低重心，提高動力；爬上坡頂時，機器人需要提高重心，防止向前傾倒。調整重心的時機要根據場地情況實際測定。爬坡行爲的實現方式同樣可以採用定時方式和傳感方式，具體方法見 6.1 節。

推動棋子、敵方時，機器人可以採用和無差別組一樣的策略。不同的是，推敵方機器人時，可以考慮加入手臂動作。

總的來說，仿人組機器人的要求比無差別更高。仿人組需要比無差別組更高的動力性能，更靈活、魯棒性更高的控制策略。

有了第六章的基礎，我們在輪式底盤基礎上搭建一台輪式仿人機器人來進行比賽。

7.1.2 任務規劃

1、結構搭建。我們直接利用第六章設計的底盤，在這之上設計感測器佈置，仿人結構。

2、爬坡練習。機器人重心較高，必須透過一定的方式調節重心，才能順利爬上坡道，登上擂臺。

3、尋找中心練習。圓柱在場地的中心位置，機器人需要透過灰度梯度的識別，知道如何到達場地的中心。

4、舉起圓柱。我們搭建的機械手臂和仿人軀體有可能並不適合舉起圓柱體的要求，這個過程需要不斷調整，尋找最合適的結構。

5、推棋子和對抗練習。這是第六章的內容，由於機器人的整體結構出現了變化，我們還是需要重複做下測試，以確定上一章所寫的演算法的有效性。

6、對抗練習。設計兩套機器人，嚴格依照比賽規則進行對抗練習，這才能真正檢驗控制演算法、結構的合理性，發現問題進行進一步改進。

7、提高比賽效果的改進設計。根據上面的聯繫和實際對抗訓練，我們能夠發現很多不足的地方，爲了彌補這些不足，我們可能需要用到一些更加強有力的馬達，更高精度的感測器、更加強健的結構等等。

7.2 任務 2：熟悉場地，標定顏色

仿人組比賽使用和無差別組相同的場地，如圖 6.4 所示。圖 6.4 中場地中心的紅色區域是爲了突出比賽宗旨而設計的。仿人組的資格認定環節，就是讓機器人舉起放置在紅色區域中心的圓柱體。需要注意的是，中心紅色區域的灰度不遵循週邊的灰度漸變規律。所以編程時一定不要對紅色區域和週邊區域採用相同的處理方式。

由於我們直接使用第六章搭建的機器人底盤，感測器位置和安裝方式已經確定，無需再次標定。直接使用第六章的標定結果即可。需要注意的是，如果對第六章搭建的機器人進行了改進，改變了感測器的位置和安裝方式，就需要重新標定。另外，由於仿人機器人的高度較高，用來檢測敵方機器人的感測器位置可能需要根據實際情況調節，調節後要重新標定。

7.3 任務 3：搭建機器人

根據比賽規則的限定，我們設計的機器人結構如圖 7.1 所示，我們將分步完成這個結構的搭建。

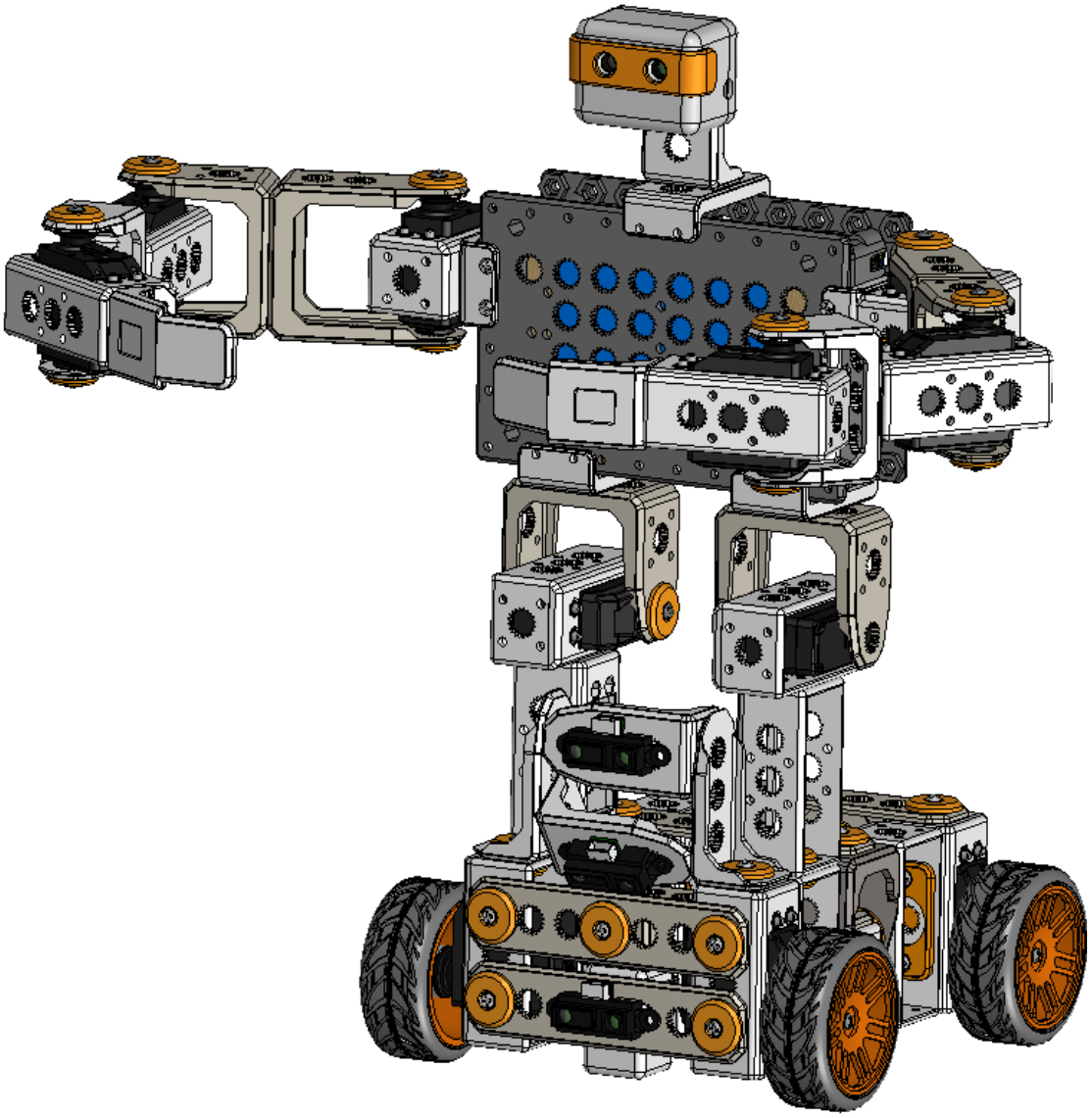


圖 7.1 仿人組比賽結構

第一步：設計機械臂

爲了讓機器人能夠抱起圓柱，我們設計的機械手臂必須有足夠的長度。根據實際測試發現如圖 7.2 所示的 3 自由度機械臂比較符合實際需要。

您可以將左右機械臂先搭建好，然後透過 4 個螺釘固定到機器人控制器外殼上，控制器的外殼是機器人的胸殼。

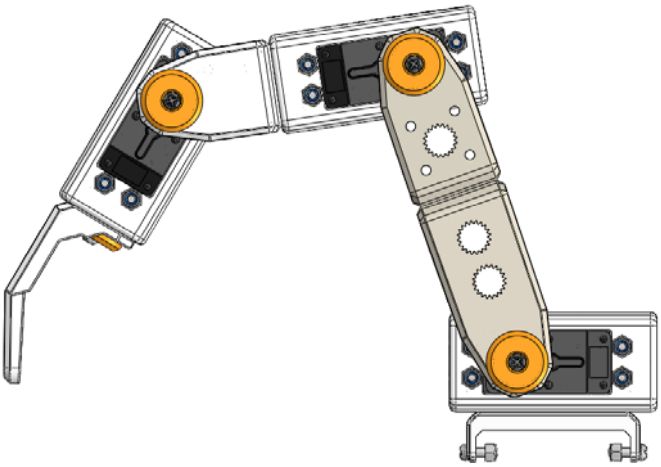


圖 7.2 3 自由度機械臂

第二步：設計腿部和腰部

我們以無差別組的結構爲基礎底盤，在這個底盤之上添加仿人部分。根據比賽要求，機器人必須能夠彎腰，腿部和腰部至少需要一個自由度，我們設計如圖 7.3 所示的安裝方式：

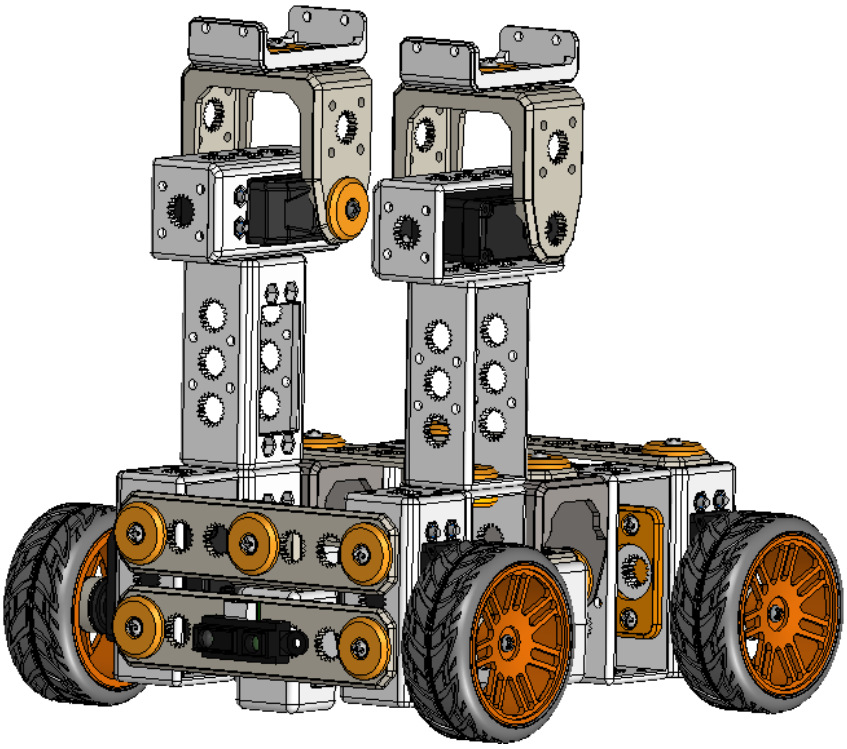


圖 7.3 腿部和腰部

第三步：修改感測器位置

根據比賽規則的規定，仿人機器人會有比較高的高度，所以無差別組的感測器佈置已經不符合仿人組的要求。我們佈置了 3 個感測器：A 用於探測棋子，當 A 觸發，但其他感測器不觸發的時候，可以判斷碰到的障礙物是棋子；B 感測器用於檢測邊緣，紅外射線與地面接觸點到探頭的距離是 12cm，是 GP2D12 的最佳使用距離；C 感測器用於探測敵方或者圓柱，

如圖 7.4 所示。

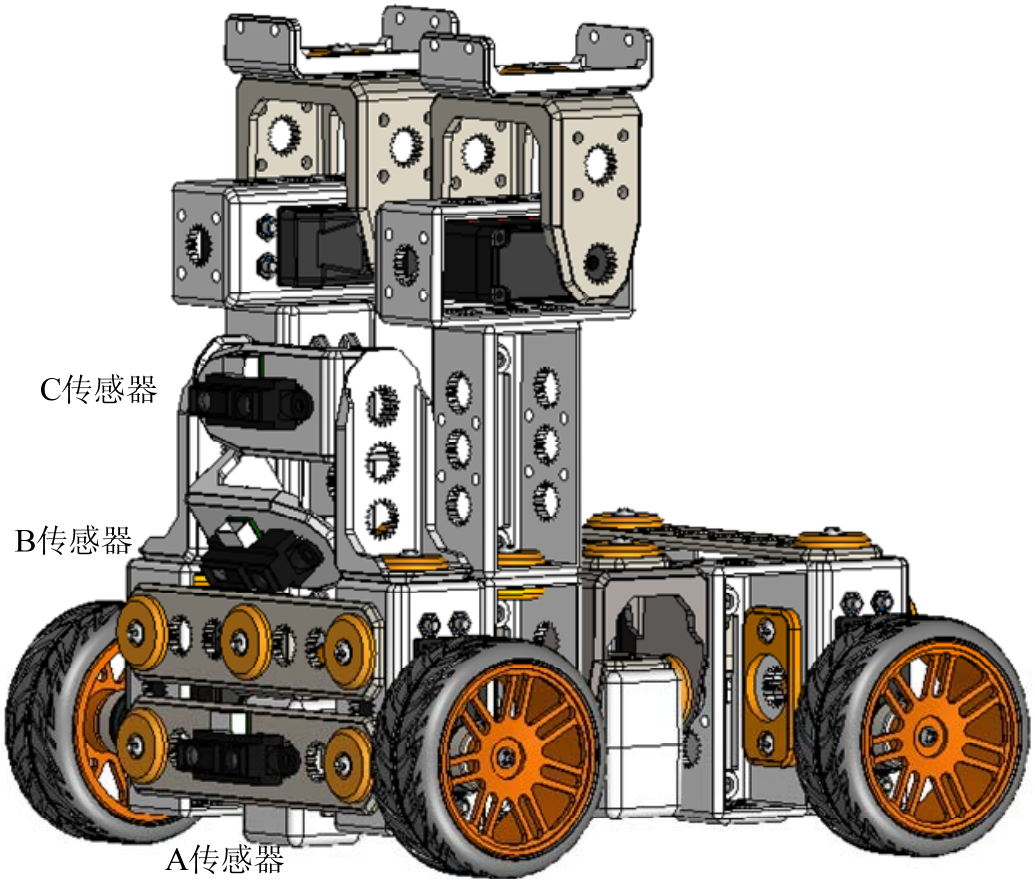


圖 7.4 感測器佈置

第四步：整體搭建

如圖 7.5 所示，我們利用控制器作爲機器人的軀幹。前面所搭建的兩個機械臂固定到控制器外殼的側邊，同時給機器人安裝上一個頭部，整個機器人即搭建完成。

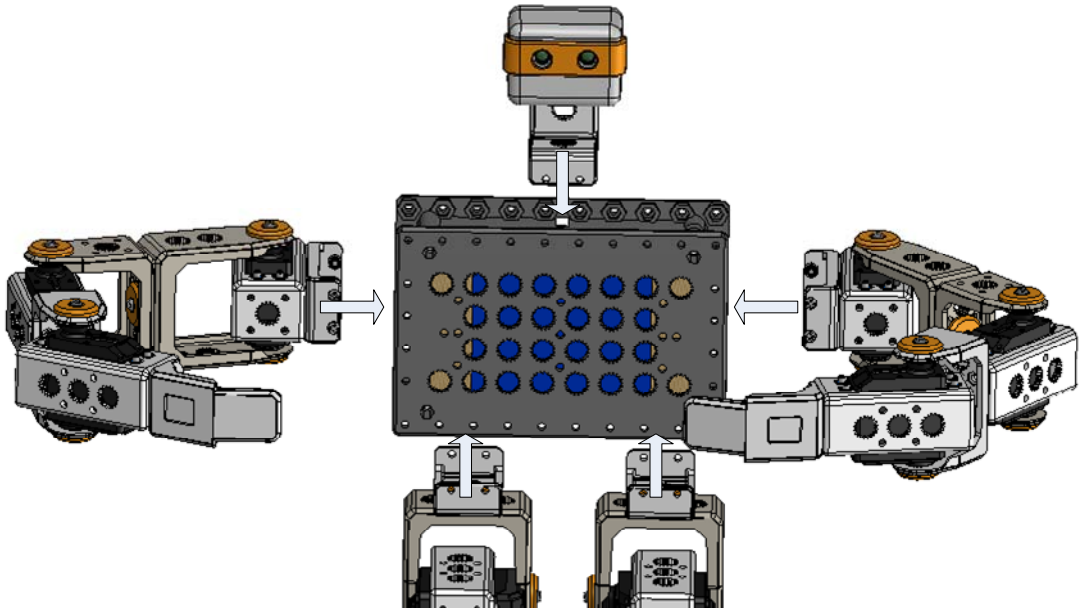


圖 7.5 整體組裝

7.4 任務 4：爬坡練習

這裏我們仍然採用簡單的定時方式實現仿人組機器人的爬坡行爲。需要注意的是，仿人機器人需要加入身體俯仰動作，透過調整重心來輔助機器人完成爬坡行爲。爬坡行爲的實現邏輯如圖 7.6 所示。圖中延時 2，延時 3 需要根據場地情況嚴格測定。另外，機器人身體俯仰的幅度也需要實地測定，過大過小都會影響機器人的爬坡行爲。

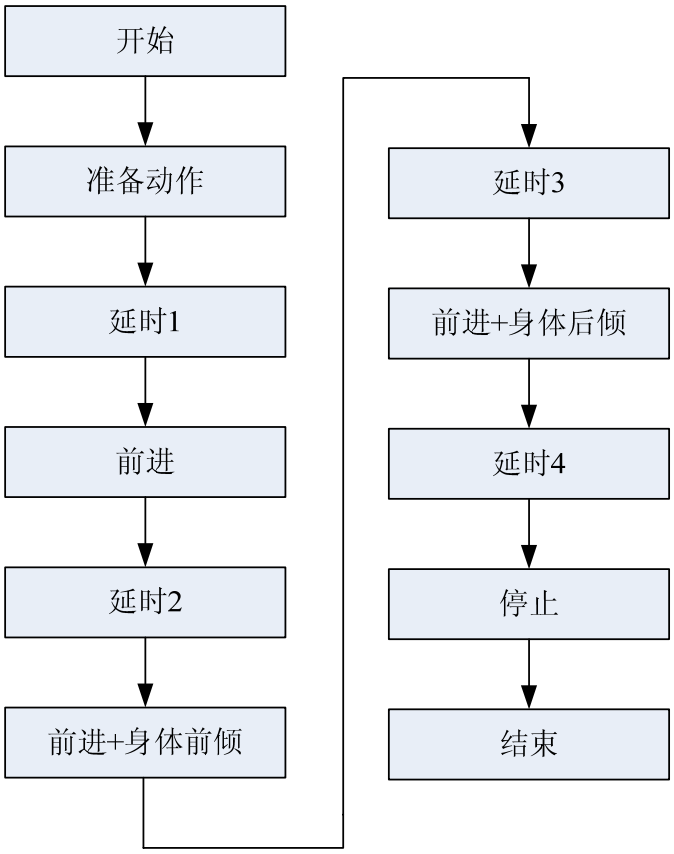


圖 7.6 爬坡實現邏輯

7.5 任務 5：尋找中心練習

資格認定環節要求機器人能走到中心區域，並舉起放置在中心區域的圓柱。機器人走到中心區域同樣有兩種實現方式：定時方式和感測方式。

定時方式是指，讓機器人按預定的軌跡走到場地中心。這種方式優點是邏輯簡單，缺點是靈活性差，如果場地條件改變，就可能導致行爲失敗。感測方式是指機器人根據灰度感測器和測距感測器的值，自主尋找場地中心並接近圓柱。這種方式靈活性較高，但是實現邏輯較爲複雜。此外，場地中心的紅色區域不遵循灰度漸變。這裏我們採用感測和定時結合的方式來實現。首先採用感測方式走到中心區域邊沿，然後借助測距感測器調整機器人方向，最後採用定時方式接近圓柱體。實現邏輯如圖 7.7 所示。邏輯中的延時同樣需要根據場地情況實地測定。

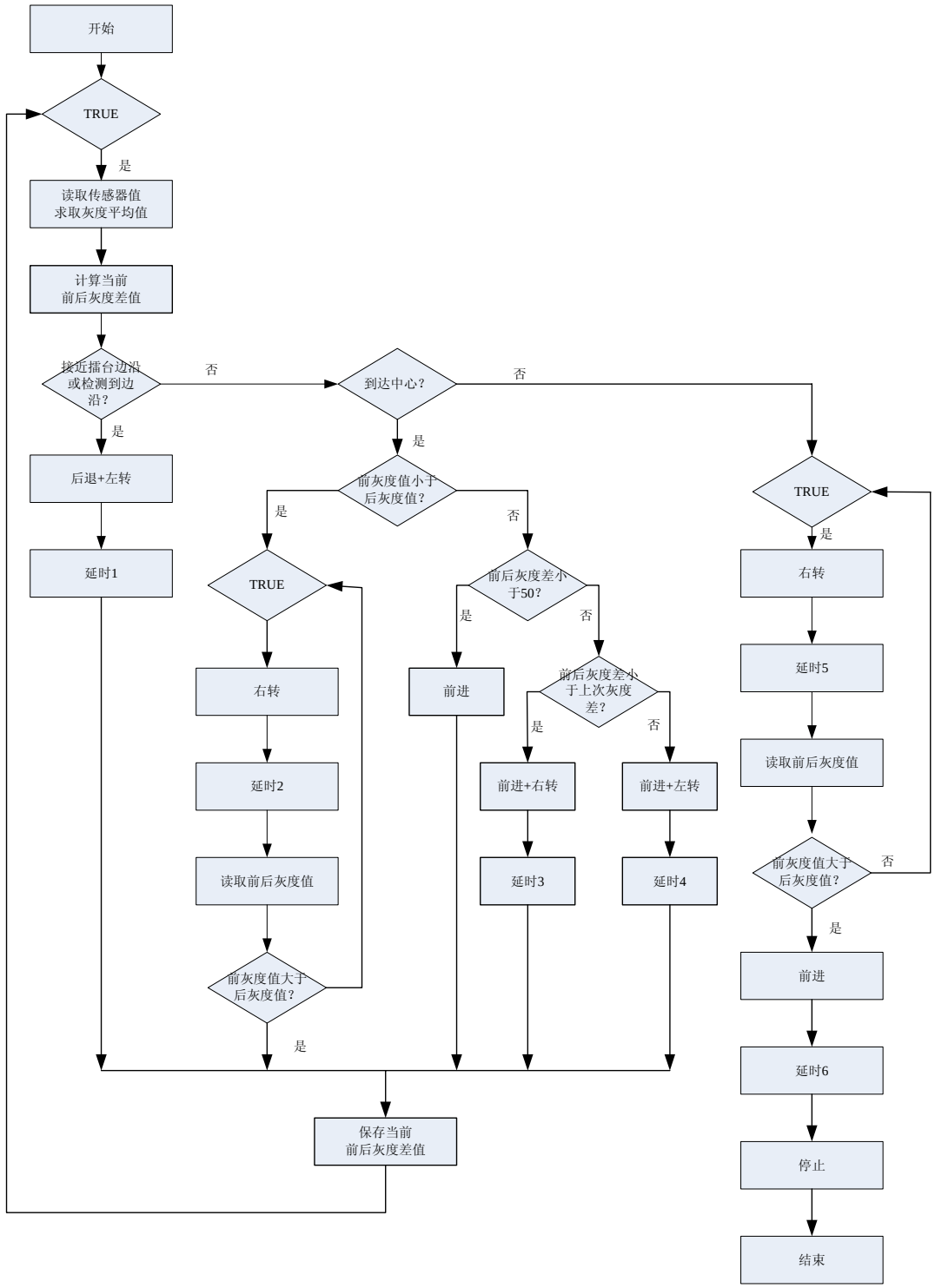


圖 7.7 尋找中心實現邏輯

7.6 任務 5：舉圓柱練習

舉圓柱時，機器人和圓柱之間的距離要合適，舉起圓柱的幅度、速度要適當，才能完成任務。舉圓柱行爲的邏輯非常簡單，只需順序執行一系列動作即可。但是舉圓柱的動作必須

根據場地和圓柱體實際情況逐個動作進行調試。將尋找中心和舉起圓柱兩個行爲融合到一起，機器人就可以完成資格認定。這裏給出一個機器人按照定時模式走到場地中心，舉起圓柱的邏輯，如圖 7.8 所示。

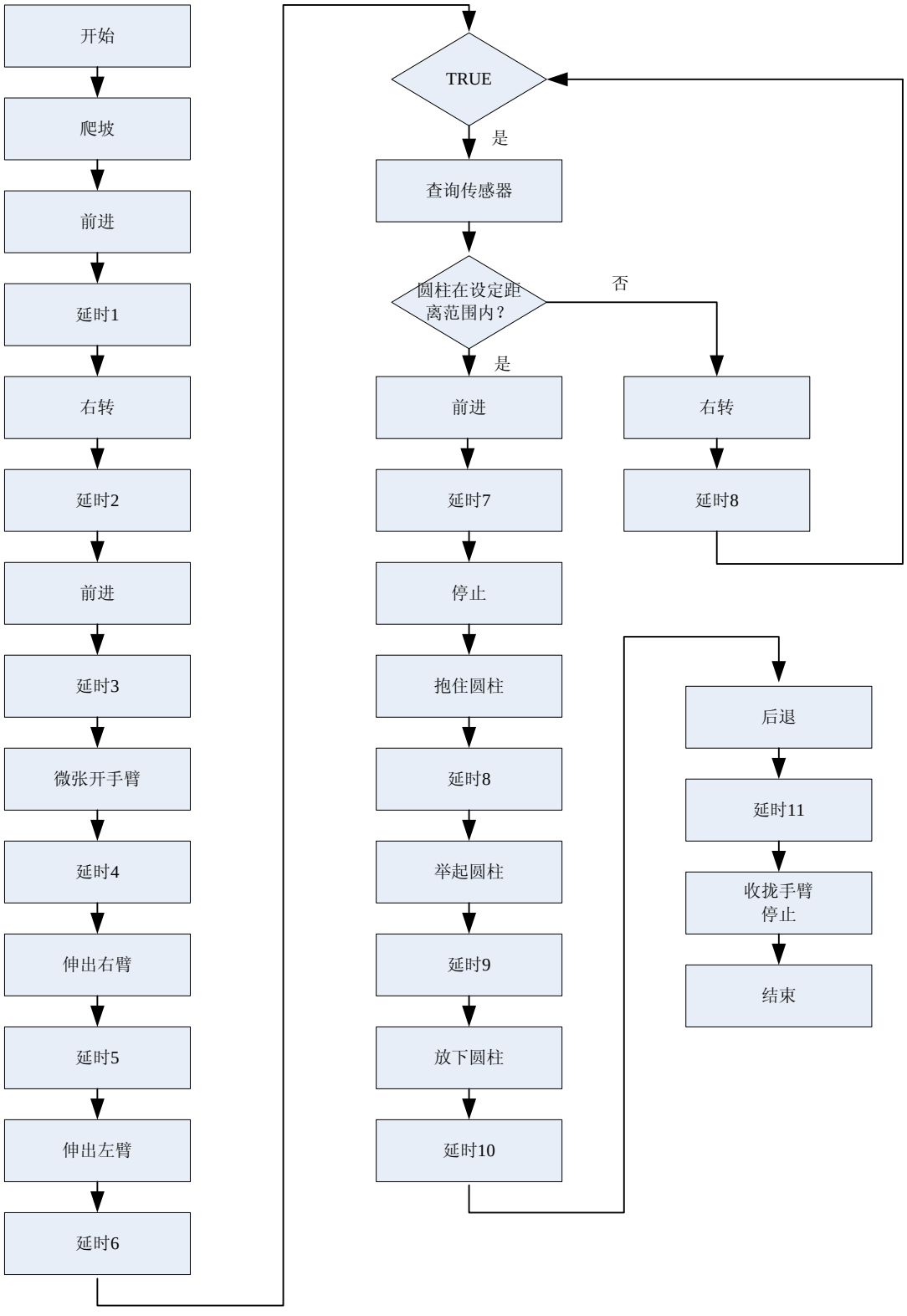


圖 7.8 資格認定實現邏輯

7.7 任務 6：推棋子練習

對仿人機器人來說，推棋子時只需採用和無差別組機器人相同的邏輯即可。需要注意的是，機器人前進時，最好保證手臂處於收攏狀態。具體邏輯如圖 7.9 和 7.10 所示。

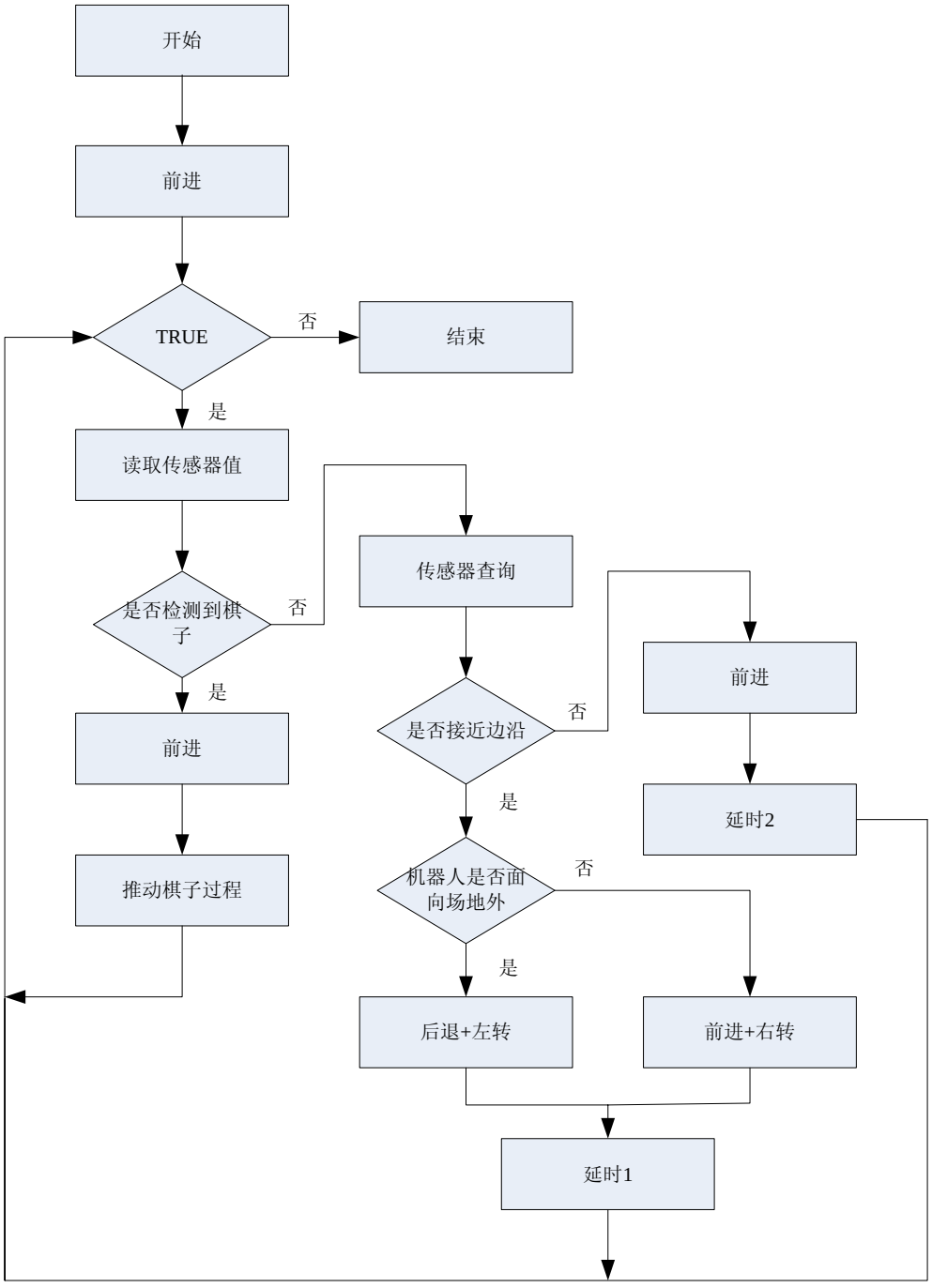


圖 7.9 推棋子實現邏輯

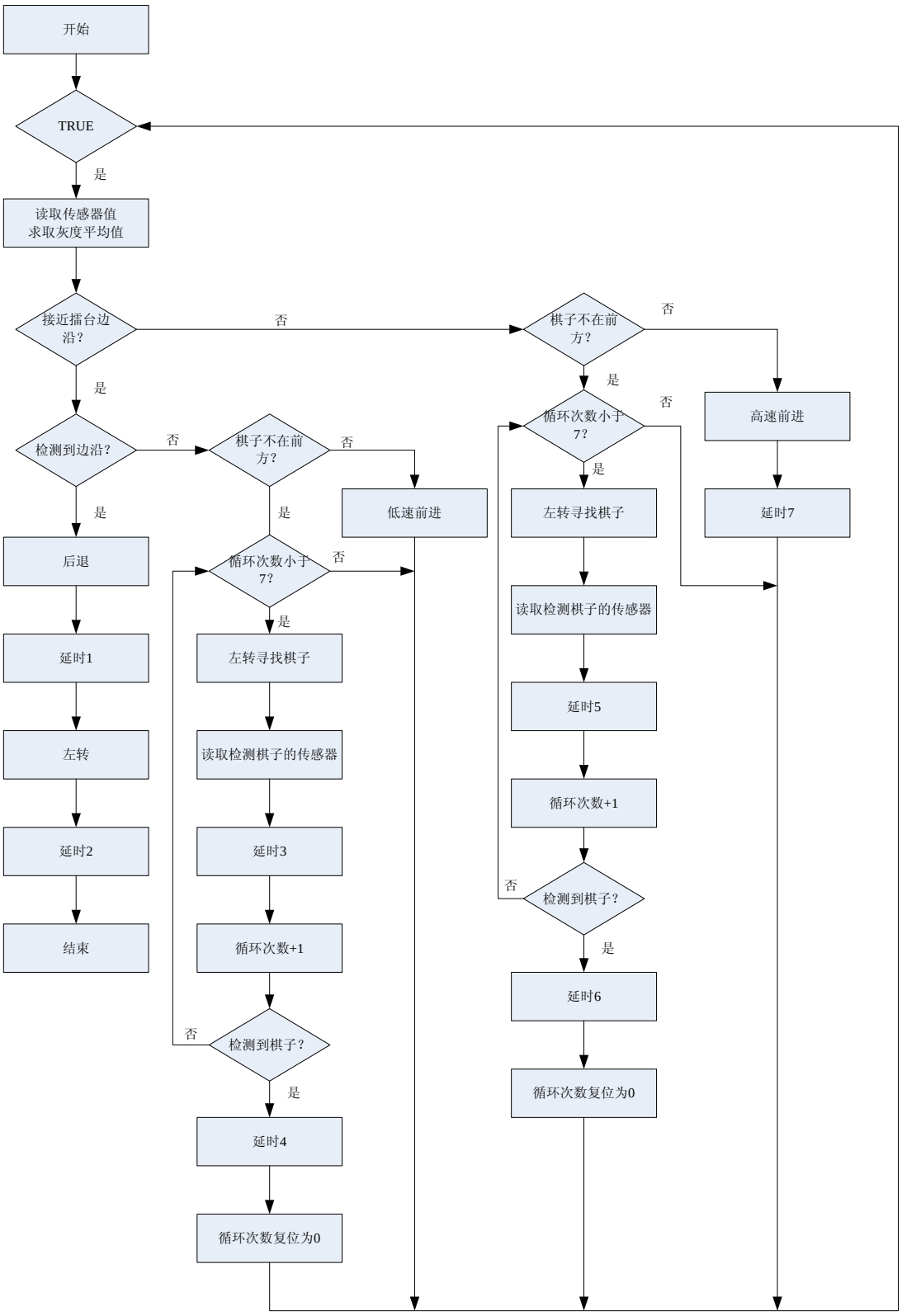


圖 7.10 推動棋子過程實現邏輯

7.8 任務 7：實戰對抗練習

仿人機器人推敵方機器人時，除了可以採用無差別組機器人類似的方法外，還可以利用雙臂來進行進攻。特別是當把敵方機器人推到場地邊沿時，可以利用雙臂把敵方機器人推倒在場外，從而防止因速度太大把自己摔出場外的情況發生。圖 7.11 和 7.12 爲實現邏輯。

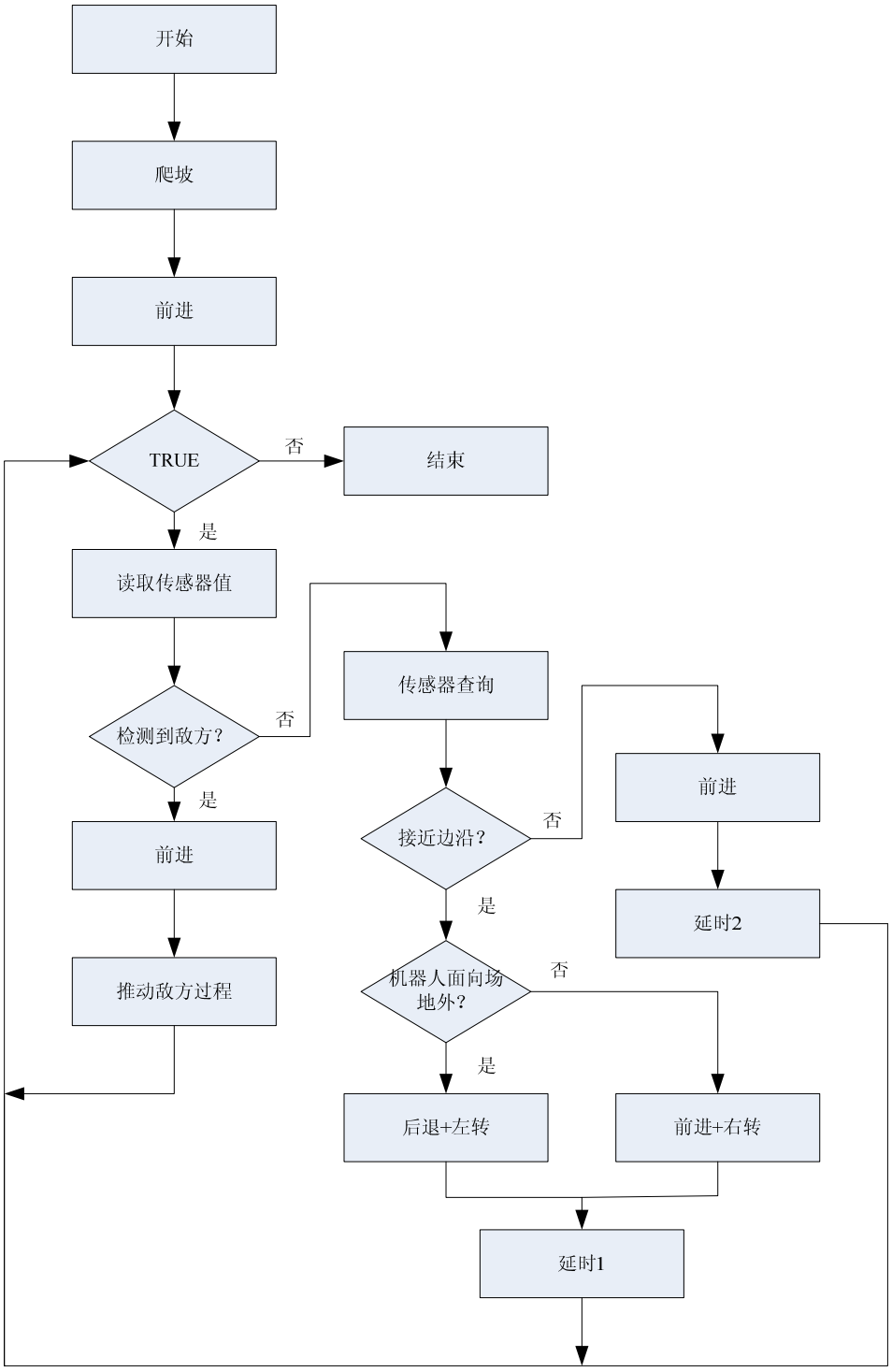


圖 7.11 實戰對抗實現邏輯

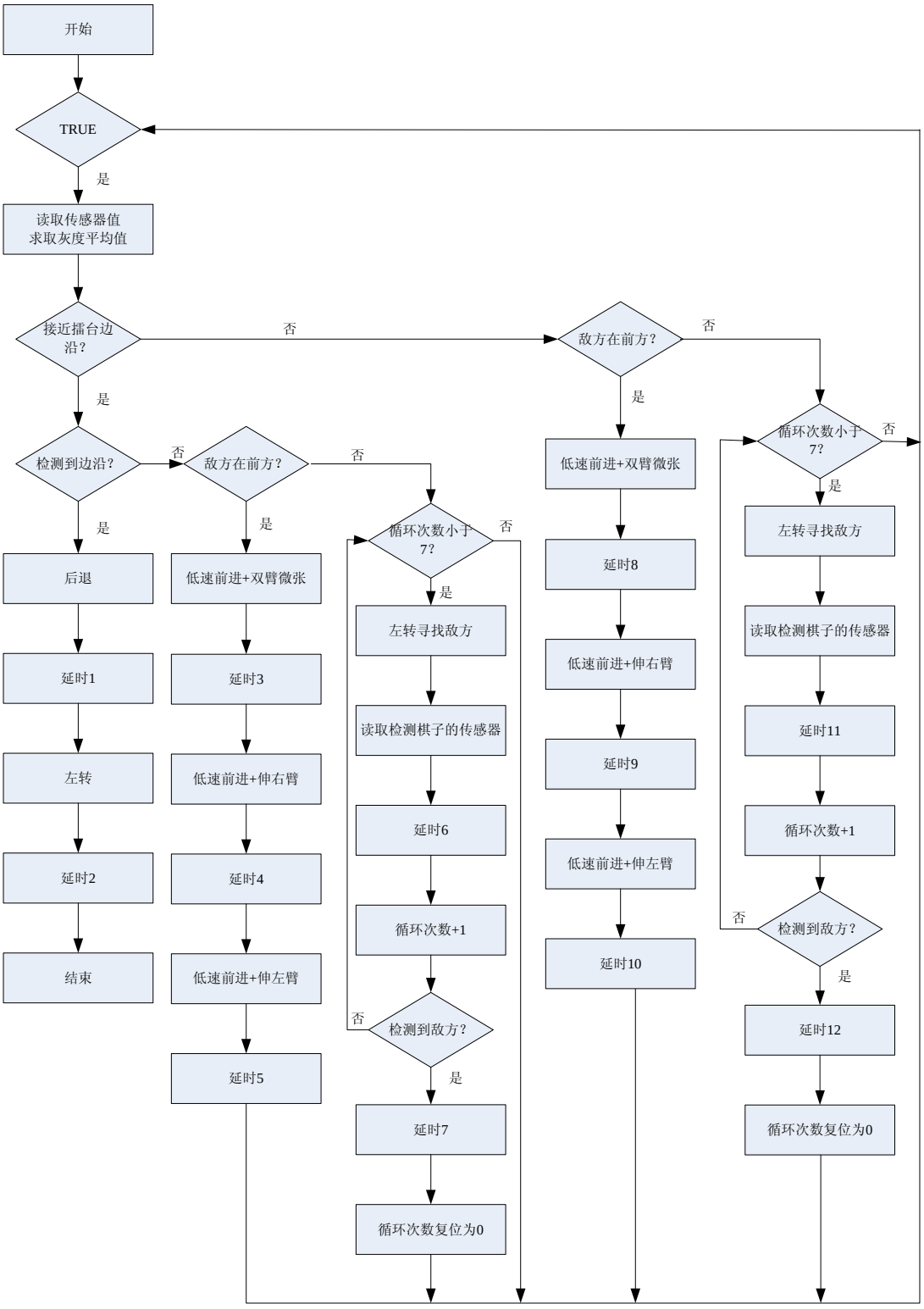


圖 7.12 推動敵方過程實現邏輯

7.9 擴充任務：改進競賽能力

仿人機器人的改進可以採用 6.6 節的方法 1 和 3 來進行。由於比賽要求仿人組機器人在場地上的投影尺寸不超過 190x180mm，所以方法 2 中將邊緣檢測感測器伸出底盤外側的改進方法這裏不可行。

7.10 擴充任務：增加速度和力量

和無差別組機器人相比，仿人組機器人的重量大大增加，提高動力的要求更加迫切。實現方式我們仍然可以採用 6.7 節介紹的方式。

第八章 我們自己的 segway

8.1 任務 1：目標分析與計畫

本章將使用“創意之星”套件搭建一個雙輪機器人（小車），使其能夠只使用兩個輪子便能保持平衡！兩輪自平衡小車類似於一階倒立擺，由於其不穩定的動態特性，兩輪自平衡小車成爲驗證各種控制演算法的理想平臺，具有重要的理論意義。到現在爲止，全世界範圍內出現了非常多的兩輪自平衡小車的成功例子，有些已經在運用到現實生活領域。Segway 代步車是最優秀的例子。

8.1.1 瞭解 segway

從 1.2.9 我們已經熟悉了 segway 是什麼樣的一種設備。那麼，它如何實現自平衡的呢？我們需要分析它的實現機理，指導我們自己的設計。

Segway 的姿態控制思路是：檢測車體的傾斜角度，車體轉動角速度，車輛的行進速度等資訊，透過調節輪子的轉動，提供回轉力矩抵消傾斜的趨勢，讓車體保持平衡。

Segway 主要由三個部分構成：

1、感測器系統

Segway 使用了 5 個陀螺儀、2 個傾斜感測器用於檢測車體的姿態，配備磁性碼盤用於檢測馬達輸出速度，還有其他感測器用於檢測是不是有人站在車輛上等。

2、控制系統

控制系統透過處理各種感測器的信號，知道車體當前的姿態、速度等資訊，控制馬達的輸出速度和扭矩，讓車體保持平衡的前提下，能夠按照操縱者的意圖前進、後退或者轉彎。

3、推進系統

推進系統包含離合器、變速箱、車輪、馬達等。這個系統將控制系統的馬達控制指令轉化成實際的輪子轉動輸出。

8.1.2 我們自己的方案

Segway 的系統是很龐大的，如果照著它的設計方案，我們可能需要幾十個專案師才能設計完成本章的項目。我們需要搭建的系統就是一個能夠自動平衡的小車，並不需要能夠載人，segway 的系統架構對我們而言有參考價值，但方案的細節我們需要重新考慮。

1、感測器系統

我們需要測量機器人的姿態，當然，不需要如 segway 那樣測量那麼多的姿態資訊，我們只需要能夠得到機器人前後俯仰的角度資訊即可。我們知道機器人的前後傾斜角度就可以透過控制馬達的轉向，反向補償這個傾斜角，合理的補償方式能夠讓機器人保持平衡。

使用什麼感測器呢？最容易想到的當然是本書 2.3.5 節介紹的“傾覆感測器”，實際上這個感測器並不合適，這種感測器輸出時開關量，並且對於倒立擺這樣要求比較高控制頻率和控制精度的系統而言，傾覆感測器的回應速度、精度都是遠遠不夠的。我們還有其他的感測器嗎？沒有了！

我們需要測量機器人傾斜角度，如果直接能夠得到這個角度是最好的，如果不能我們可以透過間接的手段來獲得。比如對角加速度進行積分，比如測量機器人前後方向兩個點對地面的距離，透過幾何關係也可以算出傾斜角度。

如圖 8.1，我們用兩個紅外測距來實現機器人傾斜角度的測量。紅外測距可以得到 a、b 的值和 c 進行幾何求解就可以得到 φ 角度。

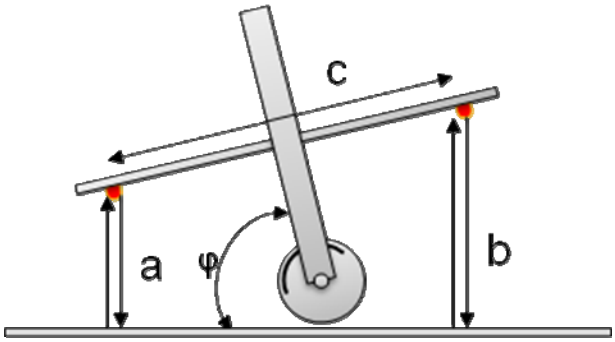


圖 8.1 角度測量示意圖

2、控制系統和推進系統

我們使用“創意之星”的 MultiFLEX™2-AVR 控制器或者 MultiFLEX™2-PXA270 控制器與 CDS5500 機器人伺服馬達就能夠完成這個控制系統和推進系統的搭建，“創意之星”控制器可以完成感測器資料獲取，CDS5500 轉動控制。

3、機器人結構

我們需要搭建一個雙輪小車，如圖 8.1 所示，機器人還需要有兩個懸臂用於安裝紅外測距感測器進行機器人傾斜角度的測量。

4、控制方法

整個結構方案、感測器方案都確定好了，那麼在我們能夠得到機器人的傾斜角度，能夠有效控制機器人輪子的轉動時，我們該如何讓機器人保持平衡呢？

如圖 8.2 所示，a-b=△h，如果△h＞0 說明機器人右傾斜，這時我們需要給一個順時針的輪子轉動ω（設定順時針ω＞0），如果△h＜0 說明機器人左傾斜，這時我們需要給一個逆時針的輪子轉動ω（設定順時針ω＞0）。因此我們可以得到一個控制關係式：ω=k*△h，k 是一個比例因數。我們只需要調整控制的週期和 k 的參數就能夠讓機器人平衡下來。

如果您學過自動控制就會發現，我們用到了比例控制方式，實際上還有其他更好的控制演算法，例如模糊控制、PID 控制等等。這些控制方式的效果都會比單純的比例調節要來的好，不過控制方法比較複雜。

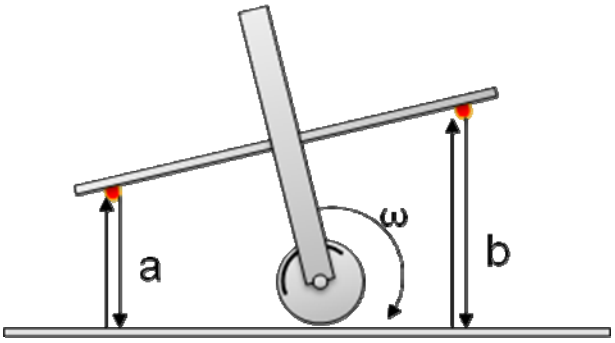


圖 8.2 控制方式簡圖

8.1.3 任務規劃

8.1.2 小節已經對自平衡小車如何設計做了詳細的介紹。接下去是具體實施環節：

1、我們需要先搭建出實驗的平臺出來，這是必須是符合上述理論分析的兩輪小車。

2、我們先熟悉我們所需要用到的零件，比如 CDS5500 和紅外測距感測器，可能還需要做標定工作。

3、在實驗平臺上設計控制演算法，對控制策略進行驗證。這是最關鍵的一個環節，您可能會遇到很多次失敗，但是最終一定會成功的，因爲這個方案已經有很多人驗證過。

8.2 任務 2：搭建雙輪小車

我們需要搭建的小車如圖 8.3 所示。

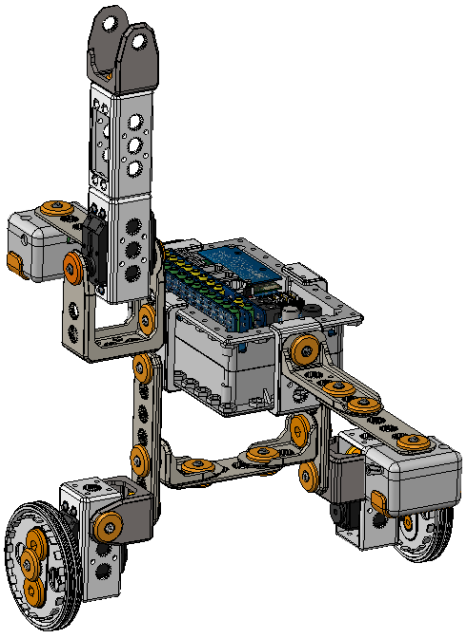


圖 8.3 雙輪自平衡小車

上圖中，雙輪自平衡小車主要由兩個“輪腿”、兩個連接紅外感測器的連接臂、一個擺錘、控制器和一個連接兩輪腿的“支撐架”組成。

支撐小車的輪子主要是由 CDS5500 和輪子（零件號 LG）構成，CDS5500 工作在可以整周轉的馬達模式。輪子透過 U3J 和 I7 結構零件連接到控制器外殼上，組成兩個“輪腿”。

您可能會問，爲什麼要把控制器架這麼高，直接把輪子安裝在控制器外殼上不行嗎？之所以這樣做，其原因有二：第一，根據 2.3.2 節中的圖 2.30 “GP2D12 測距結果與障礙物距離的關係”我們可以知道，紅外測距感測器的最佳工作範圍是在 10~20cm 之間（此時其線性度和靈敏度都比較高），因此，我們需要把感測器抬高到這個距離之內；第二，是爲了抬高整個結構的重心，使小車更容易平衡。至於爲什麼要抬高重心，大家可以想像用自己的手掌立一根筷子與立一根竹竿相比那個更容易些，有興趣的讀者可以參考相關的文獻資料或者自己做一下運動學分析，在此本書不展開討論。

兩個紅外感測器分別安裝在控制器外殼的另外兩邊，透過測量與地面之間的距離來感知小車的傾斜程度，並根據傾斜程度的大小來調節兩個輪子的速度，使小車保持平衡。實際上只使用一個紅外感測器也可以（您可以思考一下原因），但是爲了保持小車的重力平衡以及使用下文將介紹的“差分”式 PID 演算法，我們使用兩個紅外測距感測器。

由第三個伺服馬達和相應結構零件組成的“擺錘”，其主要作用是爲了更進一步地調節小車重心的平衡，當小車往一邊傾倒的時候，使擺錘向相反的方向傾倒，減小整個結構重心的偏移量。

下面，我們就雙輪自平衡小車的搭建方法做詳細介紹。

8.2.1 搭建輪腿

第一步，如圖 8.4 搭建伺服馬達組合零件。這裏需要注意的是，在安裝伺服馬達之前需要在伺服馬達架內事先放入兩個 LX2 連接件和螺母，因爲在後面的組裝中要在此處安裝一個 U3J 連接件。

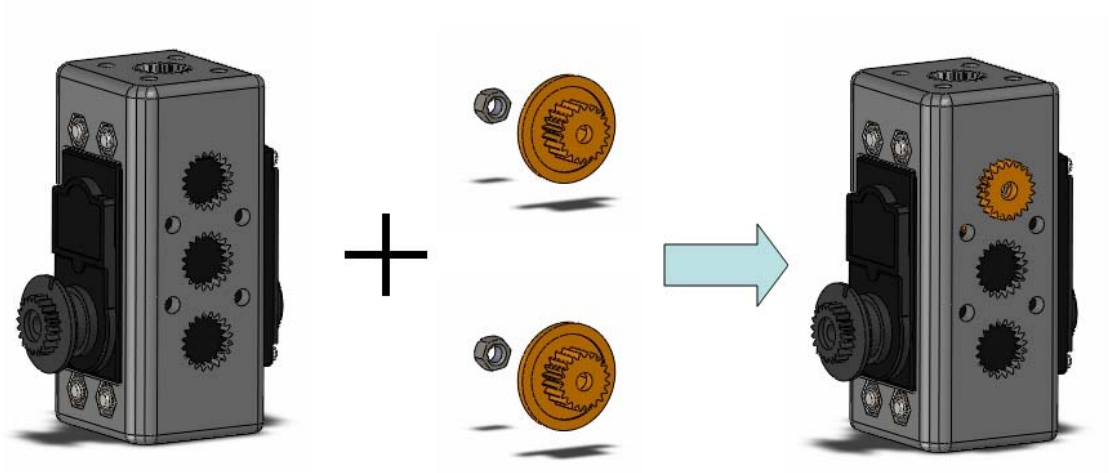


圖 8.4 搭建伺服馬達組合件

第二步，搭建輪子，如圖 8.5 所示。所使用的零件爲 2 個 LG 輪子、2 個 LX1L 連接件、一個 LM1 連接件，並且爲了增加輪子與地面的摩擦力，可以在兩個輪子上套上兩個 LT 橡膠圈。

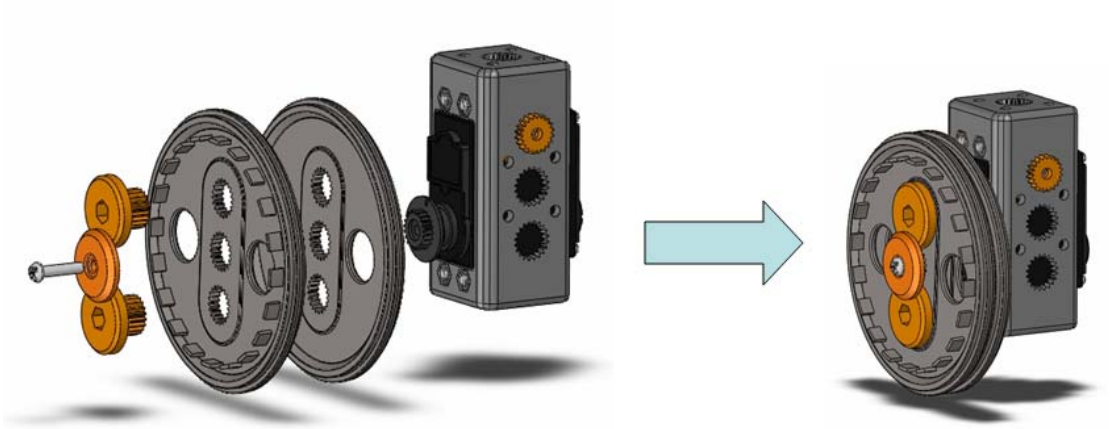


圖 8.5 搭建輪子

第三步，搭建“腿部”，如圖 8.6 所示。爲了增加結構剛度，腿的支撐零件使用兩個 I7 累加組成，所使用的 U 型件爲 U3J。

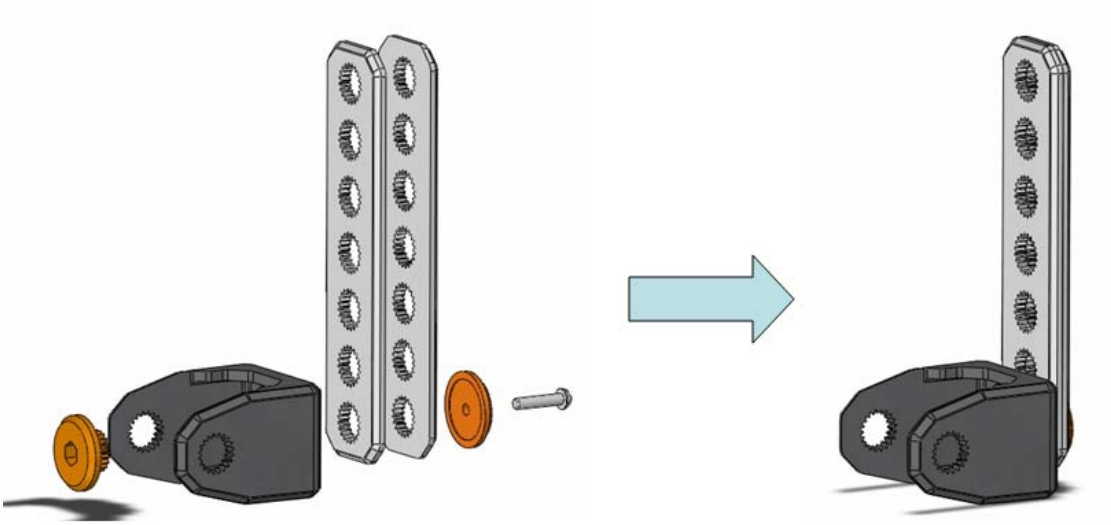


圖 8.6 搭建“腿部”

第四步，搭建完整“輪腿”，如圖 8.7 所示。

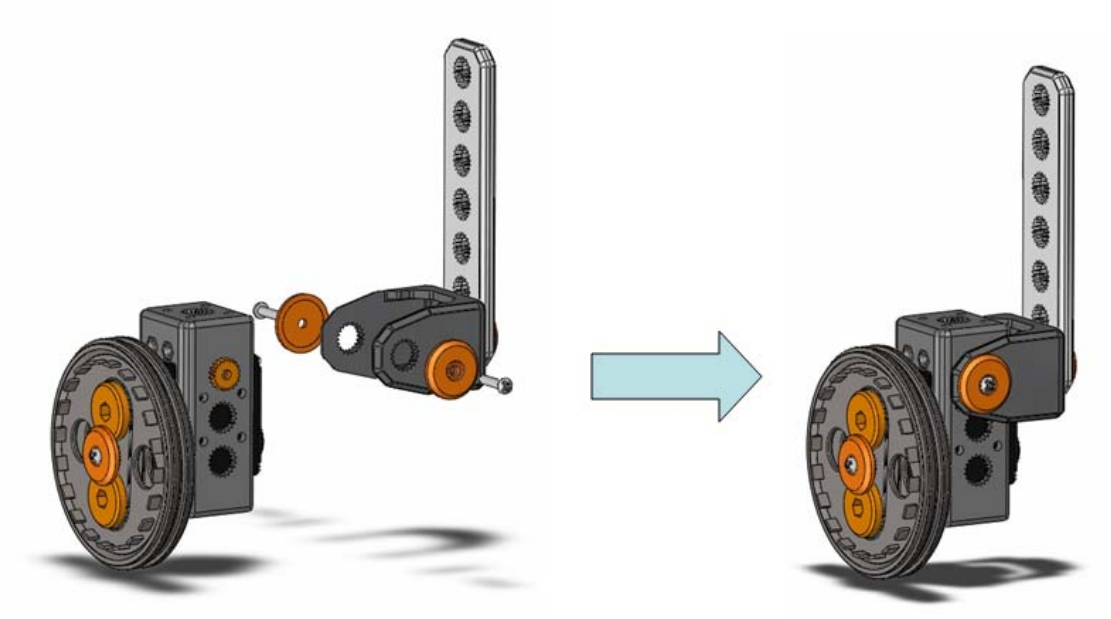


圖 8.7 搭建“輪腿”

8.2.2 搭建感測器連接臂

感測器連接臂的搭建方法比較簡單，如圖 8.8 所示，所使用的零件爲：L5-1 和 I7 結構零件、LX1 和 LM1 連接件以及紅外感測器模組。

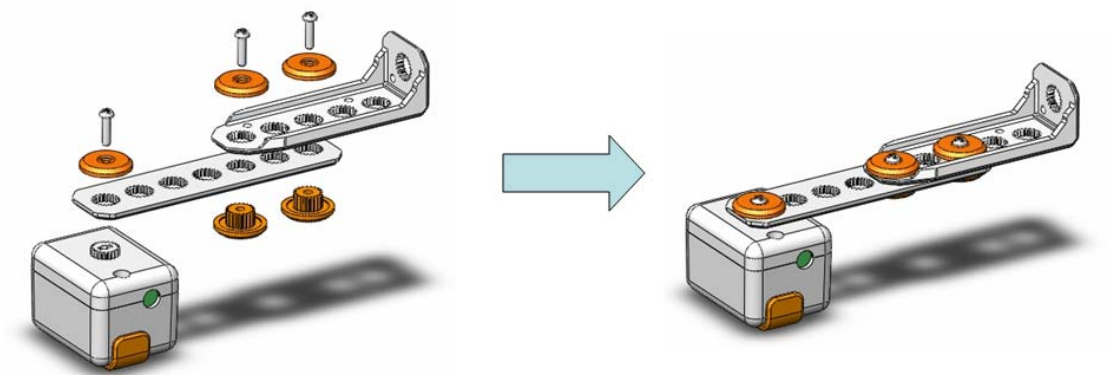


圖 8.8 搭建感測器連接臂

8.2.3 搭建擺錘

擺錘的搭建方法也比較簡單，如圖 8.9 所示，所使用的零件爲：伺服馬達組合零件、KD 伺服馬達架、U3J 結構零件以及相應連接件。

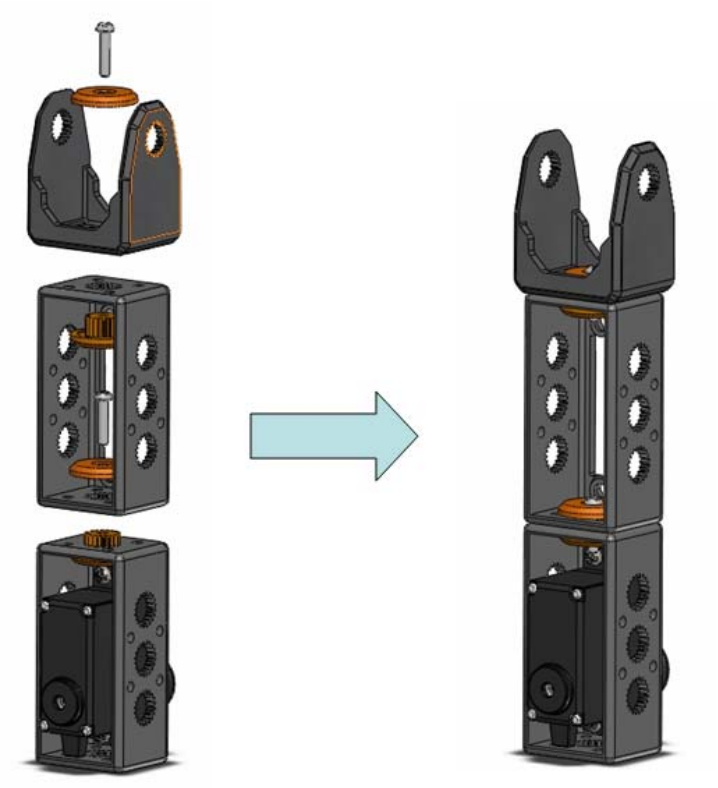


圖 8.9 搭建擺錘

8.2.4 搭建支撐架

支撐架的作用是連接兩個輪腿，增強結構剛度，搭建方法如圖 8.10 所示，所使用的零件爲：2 個 L2-1 和 1 個 I5 結構零件、連接件中的 LX1 和 LX1L 以及 LM1。需要注意的是，兩個 L2-1 結構零件之間只相距兩個 I5 構架的孔的距離，而不是 3 個。

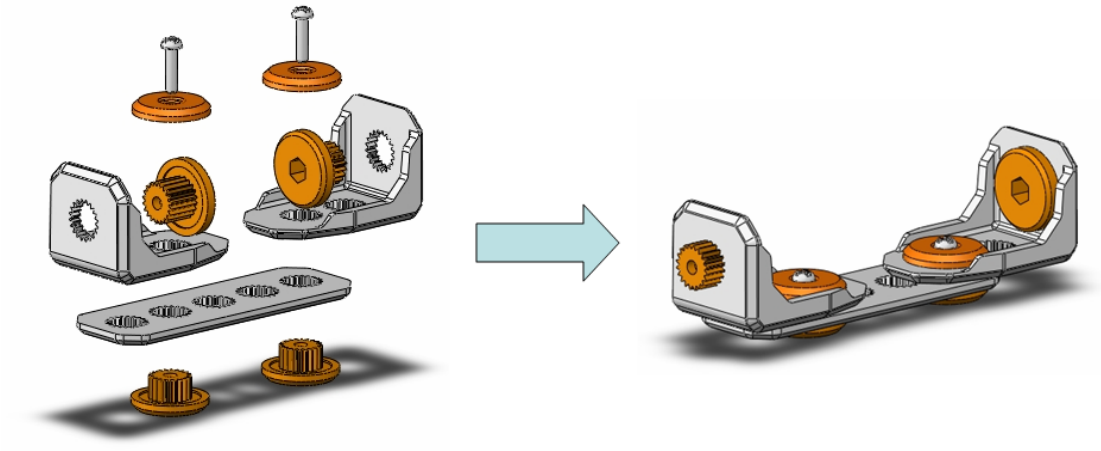


圖 8.10 搭建支撐架

8.2.5 組裝完整的小車

如圖 8.11 所示，首先使用 KB 和 KC 結構零件搭建好控制器的外殼，並在其四邊的中心位置安裝 U3H 結構零件以及相關連接件。兩個輪腿安裝在長邊的兩個 U3H 結構零件上，輪腿之間安裝支撐梁，需注意的是由於支撐梁的長度不足，需要在梁端加上一個 LD1 和一個 LD2 連接件作爲匹配。

兩個感測器和連接臂安裝在短邊的兩個 U3H 結構零件上，而擺錘則透過一個 U3C 連接件與控制器外殼的一個長邊相連接。

需要注意的是，擺錘在安裝時最好先使用調試器使伺服馬達回位到 150°的位置再豎直安裝。

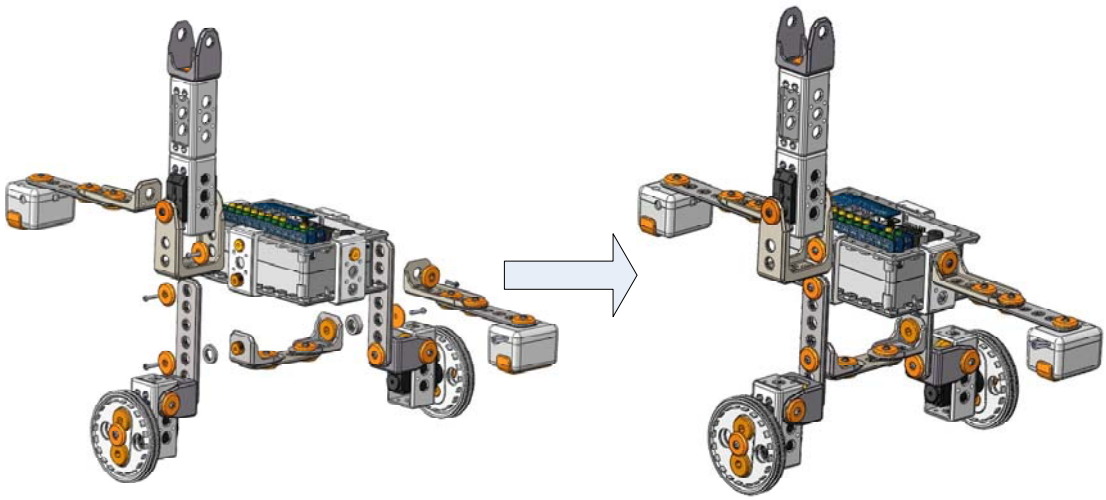


圖 8.11 搭建完整小車

至此，我們的雙輪自平衡小車就搭建完成了，大家在搭建的時候一定要有耐心並且注意思考安裝的順序，避免重複勞動，另外，需要加螺釘螺母的地方一定要加上，不要圖省事而誤了大事。

8.3 任務 3：修改伺服馬達的參數

使用多功能調試器和 RobotSevo Terminal 伺服馬達調試軟體，將 3 個伺服馬達的資料返回方式，由預設的“所有返回”改成“讀取返回”，如圖 8.12 所示。關於多功能調試器和 RobotSevo Terminal 軟體的使用請參考光碟“……\InnSTAR 產品光碟\零件資料匯總\多功能調試器”資料夾下的使用說明。

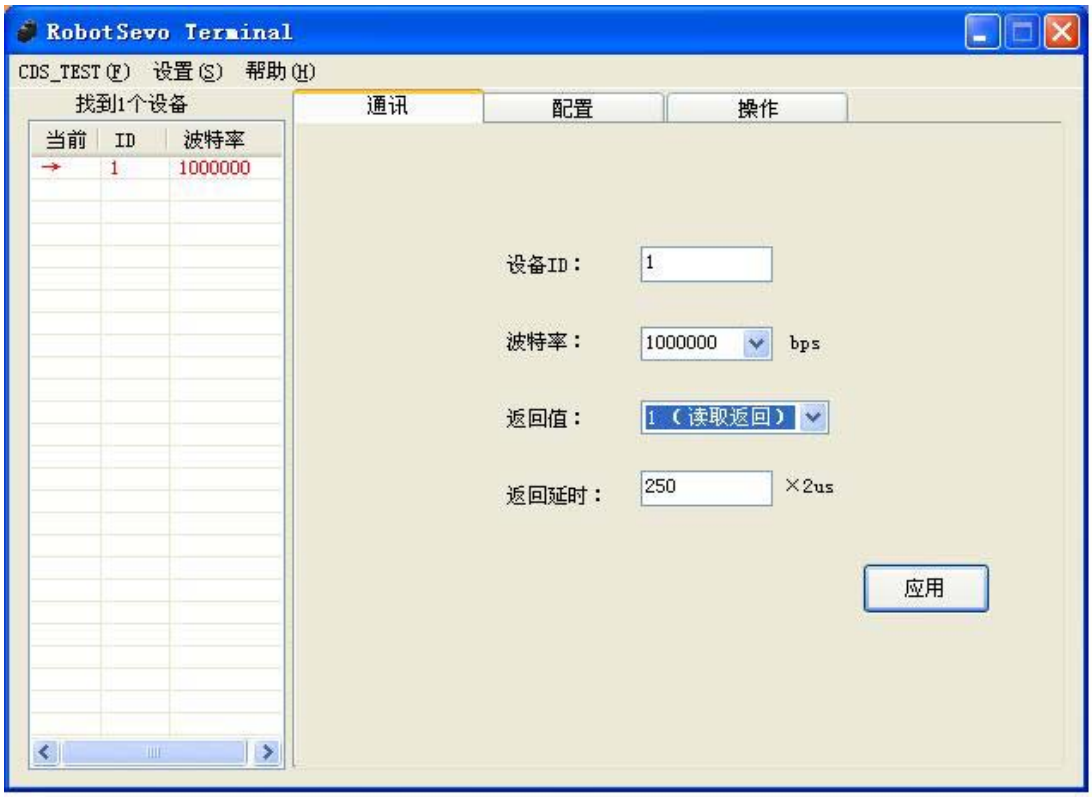


圖 8.12 將伺服馬達的資料返回方式改爲“讀取返回”

上述修改的原因是：在預設的情況下，伺服馬達資料返回的方式爲“所有返回”，也就是說，當伺服馬達接收到任意一條指令後，都會將該指令原樣返回發送給控制器，以便控制器確認伺服馬達已經接受到了指令。但是這樣的“問答”通訊方式會佔用較多控制器的通訊時間，並且使伺服馬達的回應速度變慢。在本章的應用中，控制器需要做高負荷的 PID 運算，伺服馬達的回應速度也是越快越好。因此這裏需要修改一下 3 個伺服馬達的資料返回方式，將預設的“所有返回”改成“讀取返回”，然後點擊“應用”按鈕，這樣一來伺服馬達只有在接收到讀取資料的指令後才會返回相應資料，其他情況下不返回。

8.4 任務 4：紅外測距感測器的標定

在我們寫控制演算法之前，需要事先知道小車在平衡位置時兩個感測器所採集的信號值，根據與平衡時的值相比較才能知道小車在某一時刻傾斜的方向和傾斜的程度，因此，需要對感測器進行標定，也就是記錄下在平衡位置時感測器的讀數。

標定的方法如下：

1、 兩個紅外測距感測器分別插到控制器的 AD0 埠和 AD7 埠。

2、 透過調試器將控制器連接上 PC，打開 NorthSTAR 軟體，單擊功能表 Tools|QueryAD/IO（圖 8.13），跳出查詢 AD/IO 數值的對話方塊（圖 7.12）。

3、 將調試器設置爲 AVRISP 模式，單擊“Start Service”按鈕，下載服務程式，然後

單擊“QueryAD”按鈕，便可看到當前各個 AD 埠的採樣數值。

4、 將調試器設置爲 RS232 模式，在 COM 文字方塊輸入當前調試器所在埠號，將串列傳輸速率改爲 57600，點擊“open”打開相應串列埠。

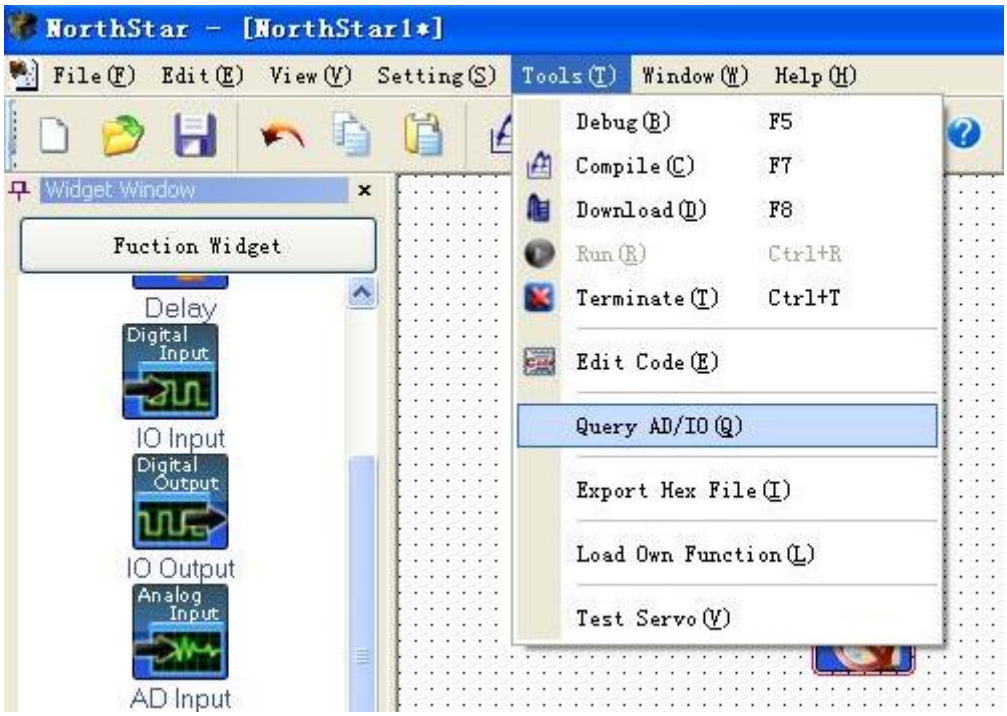


圖 8.13 選擇功能表 Tools|QueryAD/IO

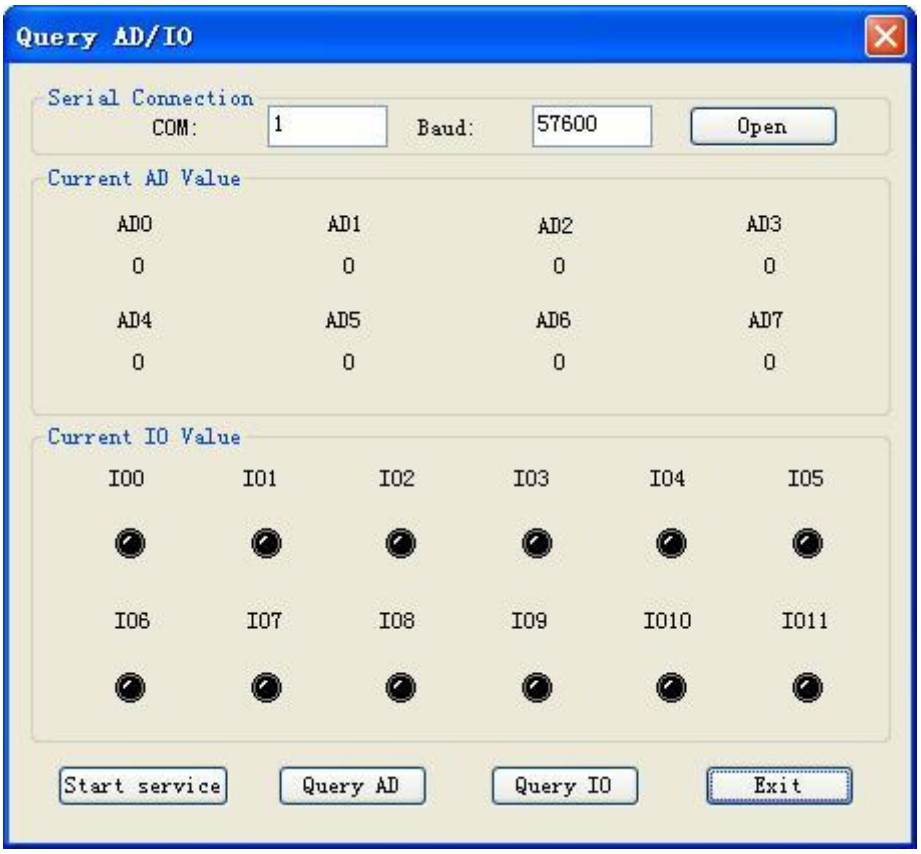


圖 8.14 線上查看 AD 採樣值

5、 用手扶住雙輪小車，使其處於平衡位置，分別記錄下此時兩個感測器（AD0 和 AD7）的讀數值，可以採用多次讀數取平均值的方法使讀數更精確些。

至此，紅外測距感測器的標定工作完成。

8.5 任務 5：數位式 PID 演算法的實現

PID 調節，亦即“比例微分積分調節”，是自動控制領域技術比較成熟而且應用最爲廣泛的一種調節方式。本節將使用數位式 PID 演算法來實現自平衡小車的控制程式。

8.5.1 背景知識介紹

由於本節所使用的 PID 演算法是非常成熟的理論，因此這裏只介紹所需要用的結論性知識，而有關 PID 演算法的基本原理和推導過程感興趣的讀者請參考光碟“……\InnSTAR 產品光碟\實驗指導書\各章節參考資料\第八章”裏的《馬達伺服控制和 PID 演算法簡介》，這裏不展開論述。

數位式 PID 演算法公式如下：

$$\Delta u(k) = u(k) - u(k-1) = Kp\left[e_k - e_{k-1} + \frac{T}{T_I}e_k + \frac{T_D}{T}(e_k - 2e_{k-1} + e_{k-2})\right]$$

$$= Kp\left(\Delta e_k + \frac{T}{T_I}e_k + \frac{T_D}{T}\Delta^2 e_k\right) = Kp\,\Delta e_k + Kie_k + Kd\,\Delta^2 e_k$$

式中 $\Delta e_k = e_k - e_{k-1}$;

$$\Delta^2 e_k = e_k - 2e_{k-1} + e_{k-2} = \Delta e_k - \Delta e_{k-1};$$

k----------採樣序號，k=0, 1, 2……

u_k ---------第 k 次採樣時刻的電腦輸出值；

e_k ---------第 k 次採樣時刻輸入的偏差值；

e_{k-1} ---------第 k-1 次採樣時刻輸入的偏差值；

Kp ---------比例係數；

Ki ---------積分係數，Ki=KpT/T_I ；

Kd ---------微分係數，Kd=KpT_d /T；

u_0 --------開始進行 PID 控制時的原始初值；

上式的輸出只是控制量的增量△u_k ，因此又稱爲“增量式 PID”演算法。當控制系統採用恒定的採樣週期時，第 k 次採樣時刻的輸出值u_k 可以由下式求得：

$$u_k = u_{k-1} + \Delta u_k$$

在本章的應用中，u_k 對應的就是伺服馬達的速度，e_k 對應的就是兩個紅外感測器讀數與平衡位置讀數的偏差值，這裏使用“差分”的概念，其基本思想就是利用兩個紅外感測器讀數的“差值”作爲偏差值，也就是 PID 的輸入值。PID 演算法中最重要的三個係數Kp 、Ki 和Kd 需要使用試驗的方法來確定。

8.5.2 演算法的實現及程式編寫

由於數位元式 PID 演算法需要高負荷的運算和恒定的採樣週期，因此，我們需要直接修改 MultiFLEX™2-AVR 控制器的硬體程式來實現。

爲了使單晶片能夠騰出足夠的運算時間來進行 PID 運算，本章所使用的 MultiFLEX™

註解掉了，因此請讀者使用附帶光碟“……\InnSTAR 產品光碟\實驗指導書\各章節原始碼\第八章”的硬體程式來編寫演算法。

使用 AVR Studio 軟體打開專案 FractalMaster（注意：專案所在目錄不能含有中文字元），整個專案比較複雜，不過我們不需要瞭解所有程式碼的含義，只需要知道在哪里加入我們的控制演算法就可以了。

在專案左邊的檔案瀏覽視窗的 Source 資料夾中，我們所用到的只有 Debug.c 文件和 Isr.c 文件，如圖 8.15 所示。

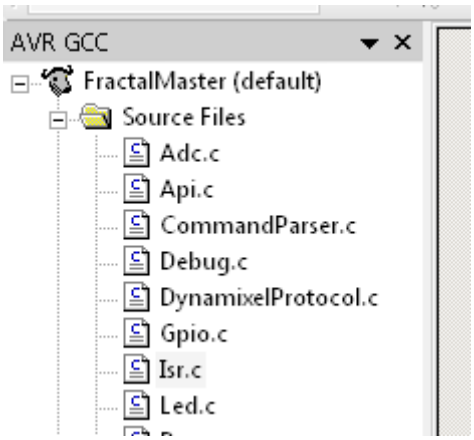


圖 8.15 專案原始檔案

其中，Debug.c 是入口函數 Main 函數的所在地，我們需要在 Main 函數中加入對伺服馬達模式的設置，將作爲輪子使用的兩個伺服馬達設置成“馬達模式”，將作爲擺錘使用的伺服馬達設置成“伺服馬達模式”。

在 Isr.c 檔中有一個 SIGNAL()函數，該函數是 2.5ms 計時器中斷函數，這就是我們控制演算法中固定週期的來源，我們只需用在這個函數中加入我們的演算法實現即可。

所需要注意的是，Main 函數中的 DevicesInit()語句是系統初始化，不能刪除！而 SIGNAL 函數中的 sei();是中斷服務語句，也不能刪除！

對所需要用到的幾個函數說明如下：

void adc_control.pStartAdc(uint8 channel)

—— 開始一次 AD 轉換，參數 channel 爲通道號 0~7。

uint8 adc_control.pGetAdcRegValue(uint16* item)；

—— 讀取上次 AD 轉換的數值，16 位元無符號整形變數指標 item 是所要存放資料的變數的位址，成功則返回 1，否則返回 0。

uint8 servo_control.pSetServoMode(uint8 id, uint8 mode)；

—— 設置伺服馬達工作模式，id 爲伺服馬達號，mode 可選擇的數值爲 SERVO_MOTO_MODE

（馬達模式）和 SERVO_SERVO_MODE（伺服馬達模式）。

uint8 servo_control.pWriteWord(uint8 id, uint8 adress, uint16 word)；

—— 向伺服馬達暫存器表中寫入兩個位元組的資料（可用於控制伺服馬達轉角和轉速），id 爲伺服馬達號碼，adress 爲要寫入資料的暫存器位址（伺服馬達暫存器位址的定義和說明請參考《CDS5500 使用手冊》），word 爲要寫入的 2 位元組資料。

以上的函數在 Debug.c 和 Isr.c 檔案中均已聲明，直接調用即可。

本章中所編寫的參考程式流程圖如圖 8.16 所示：

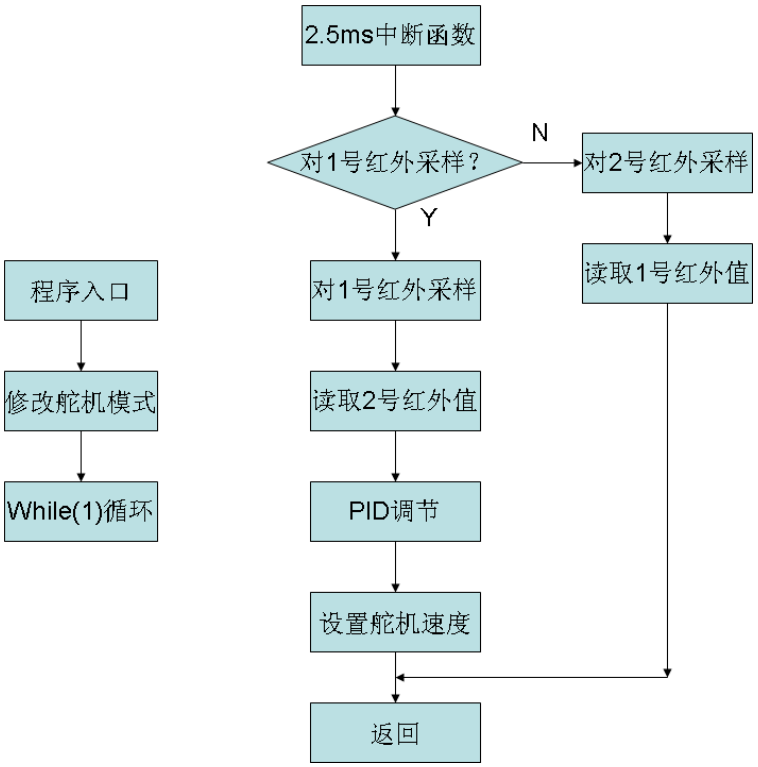


圖 8.16 程式流程圖

下面是相關的程式碼，僅供參考。

在 Debug.c 裏的程式碼：

```
int main(void)
{
    DevicesInit();  //系統初始化

    servo_control.pSetServoMode(1,SERVO_MOTO_MODE);   //設置 1 號伺服馬達爲馬達模式
    _delay_ms(100);                                   //延時 100ms
    servo_control.pSetServoMode(2,SERVO_MOTO_MODE);   //設置 2 號伺服馬達爲馬達模式
    _delay_ms(100);                                   //
    servo_control.pSetServoMode(3,SERVO_SERVO_MODE);  //設置 3 號伺服馬達爲伺服馬達
模式
```

```
    _delay_ms(100);                                          //

    while(1){}      //閉環
}
```

這裏作爲輪子使用的伺服馬達 ID 爲 1 號和 2 號，作爲擺錘使用的伺服馬達爲 3 號。

Isr.c 檔中的相關代碼如下：

```
int16 Ek = 0;                       //當前的偏差值
int16 Ek_1 = 0;                     //上一次的偏差值
int16 Ek_2 = 0;                     //上上次的偏差值
int16 Uk = 0;                       //目標增量
int16 Ukk = 0;                      //上一次目標增量
int16 Kp = 17;                      //比例係數
fp32 Ki = 0.01;                     //積分係數
fp32 Kd = 0.1;                      //微分係數
uint16 Adc0 = 0;                    //1 號紅外值（對應 0 號 AD 埠）
uint16 Adc7 = 0;                    //2 號紅外值（對應 7 號 AD 埠）
int16 Adc0_balance = 355;           //平衡時 1 號紅外值
int16 Adc7_balance = 360;           //平衡時 2 號紅外值

SIGNAL(SYSTEM_TIMER_INT)
{
    sei();      //中斷服務函數

    static uint8 sta_bStartADC0 = 0;        //是否對 1 號紅外採樣標誌
    uint16 speed1 = 0;                      //1 號伺服馬達速度計算值
    uint16 speed2 = 0;                      //2 號伺服馬達速度計算值
    uint16 angle3 = 0;                      //3 號伺服馬達（擺錘）角度計算值

    if(sta_bStartADC0 == 0)                 //如果不是對 1 號伺服馬達採樣
    {
        adc_control.pGetAdcRegValue(&Adc7); //讀取 2 號紅外值
        adc_control.pStartAdc(0);           //開始檢測 1 號紅外
        sta_bStartADC0 = 1;                 //改變標誌
    }
    else if(sta_bStartADC0 == 1)            //如果是對 1 號伺服馬達採樣
    {
        adc_control.pGetAdcRegValue(&Adc0); //讀取 1 號紅外
        adc_control.pStartAdc(7);           //開始檢測 2 號紅外
        sta_bStartADC0 = 0;                 //改變標誌

        //開始 PID 演算法
        Ek = (Adc7-Adc0)-(Adc7_balance-Adc0_balance);       //計算偏差值
        Uk = Kp*(Ek-Ek_1+Ki*Ek+Kd*(Ek-2*Ek_1+Ek_2))+Ukk;    //PID 調節
        Ukk = Uk;                                           //保存輸出值
```

```
        Ek_2 = Ek_1;                                        //保存上次偏差
        Ek_1 = Ek;                                          //保存這次偏差

        //設置馬達轉向、速度，設置擺錘角度
        if(Uk >= -1 && Uk <= 1)       //如果調節值在閾值範圍內，不需要調節
        {
            speed1 = speed2 = 0;      //馬達速度設爲 0
            angle3 = 512;             //擺錘角度設爲中位
        }
        else
        {
            if(Uk > 1023)             //限制值大小
                Uk = 1023;
            else if(Uk < -1023)
                Uk = -1023;
            if(Uk > 1)                          //如果是逆時針傾倒
            {
                speed1 = Uk;                    //1 號伺服馬達正轉
                speed2 = Uk|(0x01<<10);         //2 號伺服馬達反轉
                angle3 = 512-(Uk>>2);           //擺錘順時針擺動
            }
            else                                //如果是順時針傾倒
            {
                Uk = -Uk;                       //值變正
                speed1 = Uk|(0x01<<10);         //1 號伺服馬達反轉
                speed2 = Uk;                    //2 號伺服馬達正轉
                angle3 = 512+(Uk>>2);           //擺錘逆時針擺動
            }
        }
        servo_control.pWriteWord(1,0x20,speed1); //設置 1 號伺服馬達轉速
        _delay_us(200);                          //延時 200us
        servo_control.pWriteWord(2,0x20,speed2); //設置 2 號伺服馬達轉速
        _delay_us(200);                          //
        servo_control.pWriteWord(3,0x1E,angle3); //設置擺錘角度
    }
}
```

請讀者參考以上程式碼，自己測定平衡時 2 個紅外感測器的讀數，並透過不斷試驗的方法求得合適的 PID 參數，編寫控制程式。

程式編譯完成後，使用 MultiFLEX™2-AVR 控制器的讀者可直接使用多功能調試器將編譯好的檔案下載到 AVR 單晶片上去。如果您使用的是 MultiFLEX™2-PXA270 控制器也可以完成本實驗，您可以直接修改 AVR 控制卡上的程式，將 MultiFLEX™2-PXA270 控制器當成 MultiFLEX™2-AVR 控制器使用，下載程式的方法可參考 2.6.1 小節。

至此，數位式 PID 演算法的實現過程介紹完畢，希望讀者能夠順利實現小車的自平衡站

8.6 任務 6：擴充任務

如果讀者已經順利實現了雙輪小車穩定站立，那麼接下來可以嘗試讓小車前進和轉彎！

不過作爲擴充任務，本節僅給出一點提示，作爲參考：

適當地修改一下 1 號紅外和 2 號紅外平衡時的讀值，小車會有什麼反應？

當調節兩個輪子的轉速的時候，讓一個輪子的速度大一點，另一個輪子的速度小一點，小車又會有什麼樣的反應？

第九章 PXA270語音問答機器人

9.1 任務 1：目標分析與計畫

9.1.1 語音識別技術

與機器進行語音交流，讓機器明白您說什麼，這是人們長期以來夢寐以求的事情。語音識別技術就是讓機器透過識別和理解過程把語音信號轉變爲相應的文本或命令的高技術。語音識別是一門交叉學科。近二十年來，語音識別技術取得顯著進步，開始從實驗室走向市場。人們預計，未來 10 年內，語音識別技術將進入工業、家電、通信、汽車電子、醫療、家庭服務、消費電子產品等各個領域。

語音識別技術所涉及的領域包括：信號處理、模式識別、機率論和資訊理論、發聲機理和聽覺機理、人工智慧等等。

9.1.2 語音識別的分類

根據識別的物件不同，語音識別任務大體可分爲 3 類，即孤立詞識別（isolated word recognition)，關鍵字識別（或稱關鍵字檢出，keyword spotting)和連續語音識別。其中，孤立詞識別 的任務是識別事先已知的孤立的詞，如“開機”、“關機”等；連續語音識別的任務則是識別任意的連續語音，如一個句子或一段話；連續語音流中的關鍵字檢測針對的是連續語音，但它並不識別全部文字，而只是檢測已知的若干關鍵字在何處出現，如在一段話中檢測“電腦”、“世界”這兩個詞。

根據針對的發音人，可以把語音識別技術分爲特定人語音識別和非特定人語音識別，前者只能識別一個或幾個人的語音，而後者則可以被任何人使用。顯然，非特定人語音識別系統更符合實際需要，但它要比針對特定人的識別困難得多。

9.1.3 語音對話機器人

語音識別技術被認爲是 2000 年至 2010 年間資訊技術領域十大重要的科技發展技術之一，具有廣闊的技術前景和商業前景。近年來，世界各國多個研究機構紛紛投入人機對話形態的機器人研究，誕生了很多學術成果和商業產品。如日本大阪大學的“若九”機器人，能與一名女演員同台表演時長爲 20 分鐘的話劇；日本名古屋的 IT 企業夢想提供公司的“IFBOT”機器人具有 5 歲兒童的說話能力，能與人進行簡單的交談，幫助老年人擺脫孤獨的煩惱。



圖 9.1 “若九”機器人

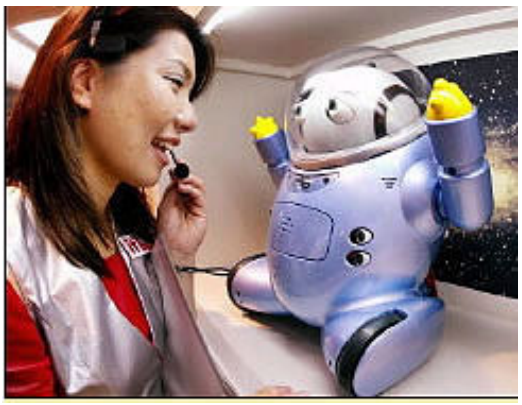


圖 9.2 “IFBOT”機器人

9.1.4 我們的任務和計畫

在這個實驗裏，我們需要設計一個能聽懂人類的話語的機器人，還要能發出聲音，與人類進行交流。“創意之星”的語音識別引擎是孤立詞識別（isolated word recognition)，所以使用起來比較簡單，但是要實現比較智慧的功能還需要經過精心設計。爲了達到理想的效果，我們可以預先設計好對話內容，透過一個對話劇本讓機器人看起來更聰明一些。機器人的發音，這裏可以使用”創意之星”的 mp3 播放功能來實現，我們將機器人說的臺詞錄成 mp3 格式的檔案，讓機器人播放這些檔案來進行發音，有興趣的朋友，還可以使用一些音效軟體對音調進行處理，讓其變得更爲俏皮一些。

根據上面的思路，我們的實現計畫分爲四步進行：第一步，機器人結構搭建；第二步，設計對話的劇本；第三步，進行對話臺詞機器人部分的語音錄製；第四步，編程實現對話邏輯。

9.2 任務 2：結構設計和安裝

由於《創意之星機器人套件組裝指南》有詳細的搭建說明，本節不再重複。

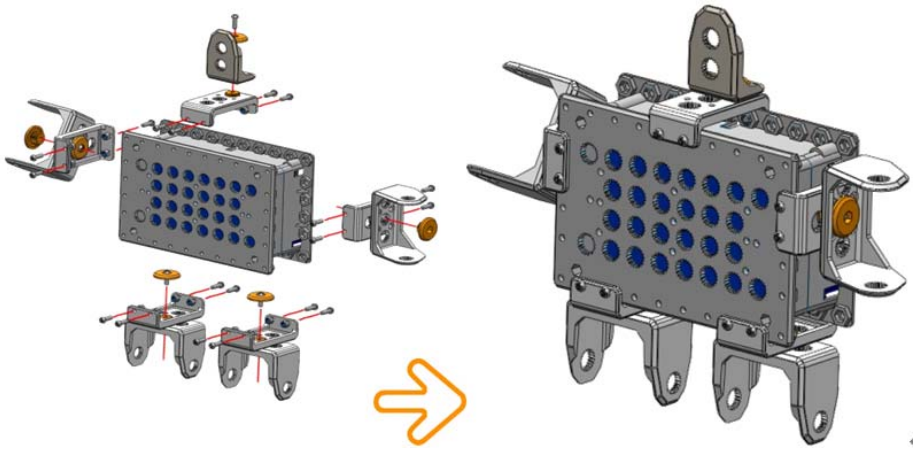


圖 9.3 軀體的搭建

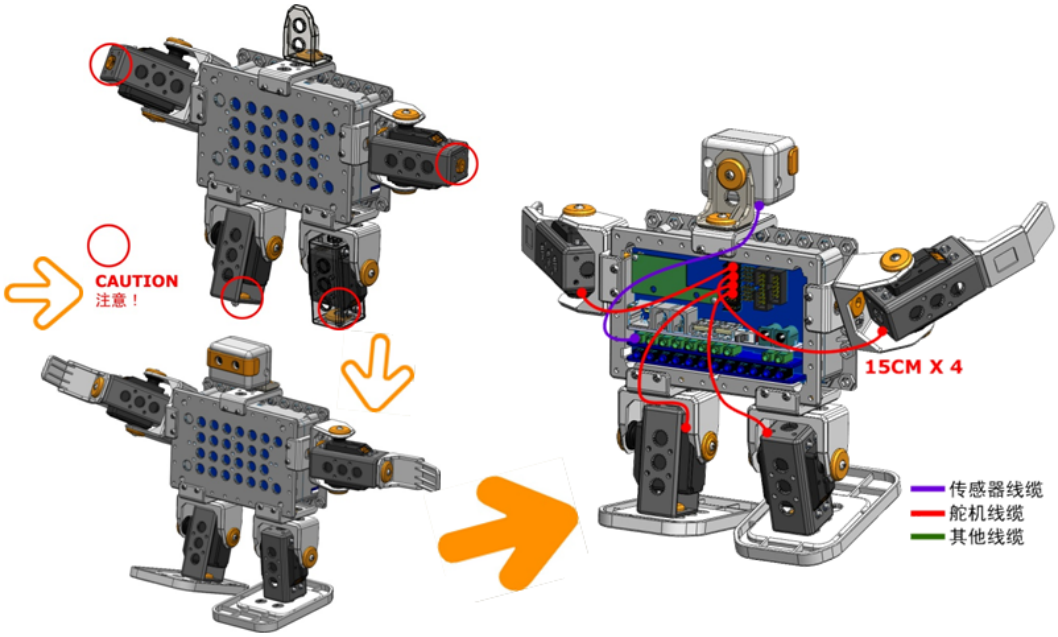


圖 9.4 四肢的搭建

9.3 任務 3：設計對話

我們的機器人應該是禮貌的，所以日常問候用語是必需的。再加入一些簡單的臺詞，讓對話稍微豐滿一些。下面是一段簡單的例子：

People：你好
Robot：你好
People：你叫什麼名字？
Robot：我叫創意之星
People：你從哪來？
Robot：我來自華亨

People：你也是機器人嗎？
Robot：是的
People：真聰明
Robot：謝謝
People：再見
Robot：Bye
一個簡單的對話劇本完成了。馬上動手將它實現吧！

9.4 任務 4：劇本錄音

爲了讓我們的機器人說話語調能更加有親和力，我們可以事先將機器人的臺詞用錄音軟體錄製下來，後期可以透過一些軟體轉換語調，讓我們的機器人更具親和力。

使用 Windows 自帶的“錄音機”程式即可進行臺詞的錄製，該程式的位置在系統功能表“開始”“所有程式”“附屬應用程式”“娛樂”“錄音機”，具體使用方法請查閱 Windows 的說明主題。

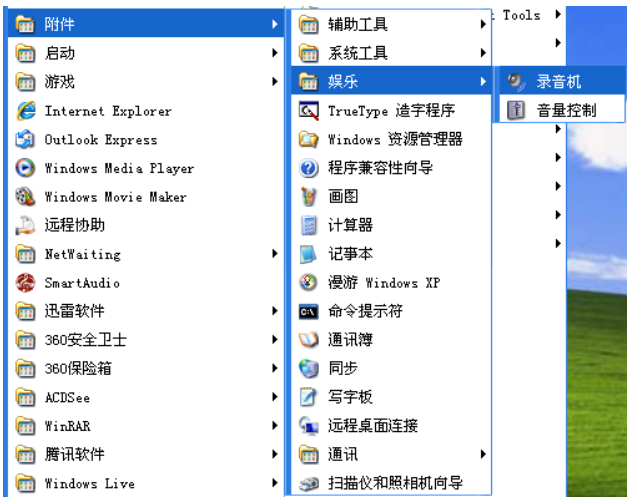


圖 9.5 錄音機



圖 9.6 錄音機程式介面

使用“錄音機”程式錄製的音效檔案是 WAV 格式，通常體積都比較大，我們可以將其壓

縮成 MP3 格式，減小儲存控制項的佔用，同時也縮短程式下載時間。MP3 壓縮工具有很多，這裏我們使用一款免費的 MP3 編碼引擎—LAME，在 Windows 下是一個檔案名爲 lame.exe 的命令行程式，各大下載網站也可以找到這個程式。



圖 9.7 Lame 工具包

這個工具的命令格式爲：

lame [options] inputfile [outputfile]

例如，我們錄製了一句臺詞，保存成爲檔“hello.wav”，則可以透過下面的指令將其轉化爲 mp3 格式。

lame hello.wav hello.mp3

具體的示範如下：

首先，我們將下載的 lame 程式解壓，這裏我們解壓到 D：根目錄下，目錄明爲“lame”，然後把我們需要進行編碼的“hello.wav”檔案也拷貝到該目錄下。



圖 9.8 拷貝檔案至 Lame 目錄

下面，需要進入 DOS 命令提示字元，點擊 Windows 的系統功能表“開始”“執行”。輸入“cmd”，按下 Enter，跳出命令提示字元。

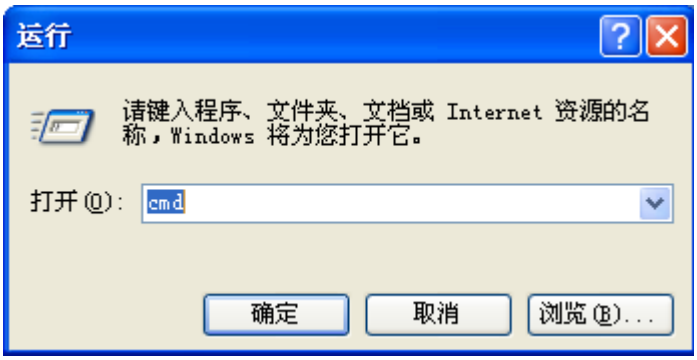


圖 9.9 進入 DOS 命令提示字元

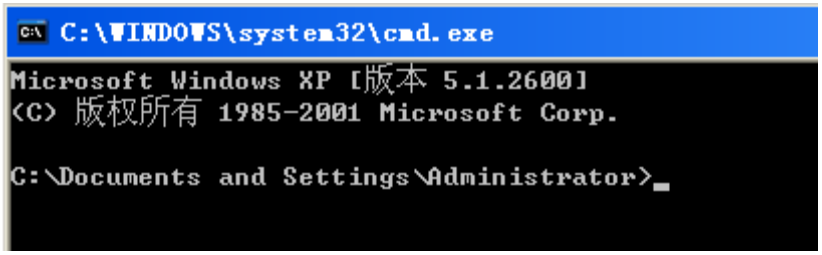


圖 9.10 DOS 命令行

進入存放著 lame.exe 及 hello.wav 的目錄，進行如下操作：

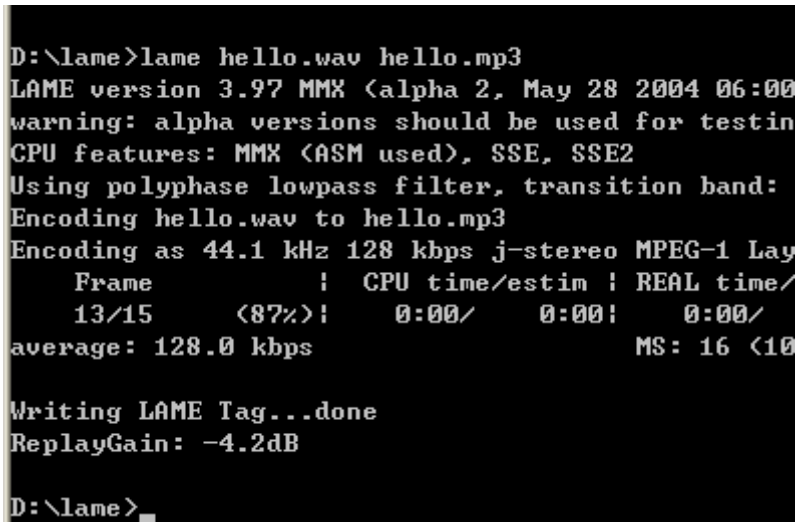


圖 9.11 轉換成功

至此，“hello.mp3”檔案誕生。

按照上面的方法，將劇本中機器人每一句臺詞都單獨錄製成 wav 檔，並轉換成 mp3 格式，以上工作完成後，此項任務便宣告結束。

9.5 任務 5：程式編寫

所有材料準備齊全，正式的編程任務開始。借助“創意之星”配套的 NorthSTAR 軟體，我們可以完成所有的程式編寫工作。

9.5.1 新建專案

在 NorthSTAR 軟體中新建一個 Project，跳出 Project Configuration 頁面。“Controller”欄選擇我們使用的控制器：MultiFLEX™2-PXA270 控制器；因爲我們搭建的是非標準結構，“Robot”欄選擇“Customized”。點擊“next”，跳出 Servo 頁面。

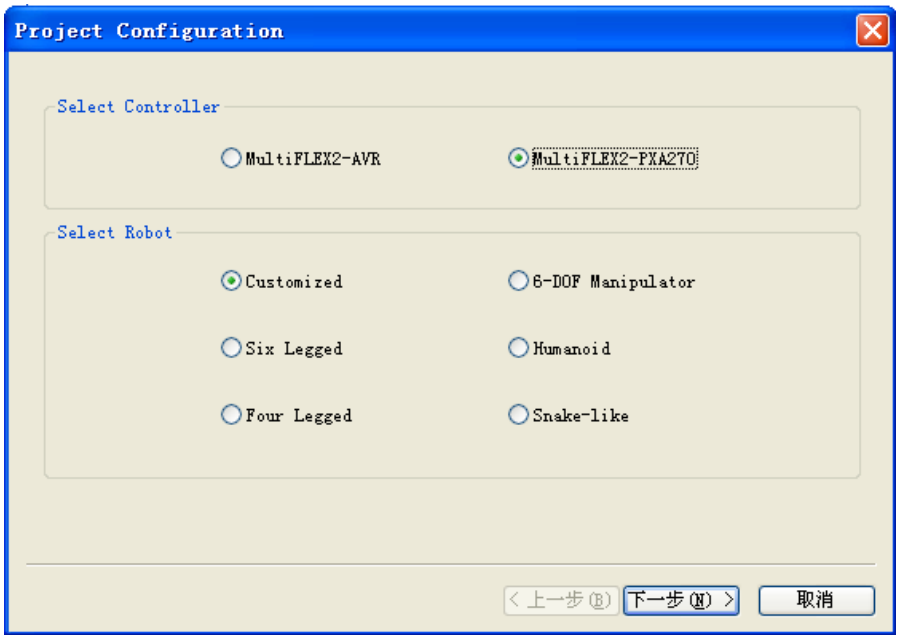


圖 9.12 新建專案

在 Servo 頁面中，我們爲機器人設置四個伺服馬達，分別對應雙手和雙足四個關節。四個伺服馬達均配置成 Servo Mode。

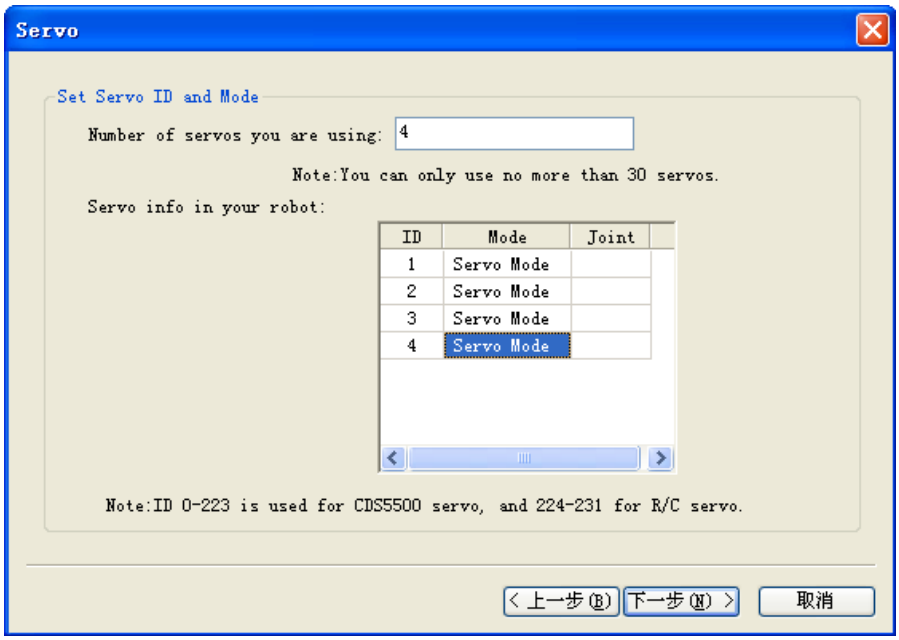


圖 9.13 伺服馬達設置

繼續點擊“next”按鈕，AD 和 IO 感測器都使用預設設置。

9.5.2 添加變數

在這個程式中，需要一個變數來儲存識別結果，以供後面進行判斷應該用哪句話來回應。這裏我們添加一個 int 型的變數 SR_Res。

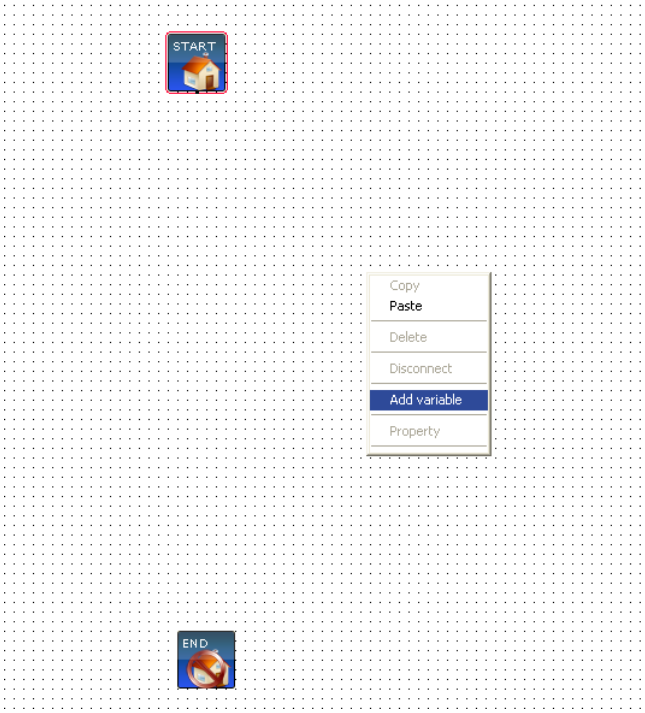


圖 9.14 添加變數

9.5.3 初始化語音識別控制項

從左邊控制項欄的“Plug-in Widget”分類中選擇 Speech 控制項，並按住滑鼠左鍵拖曳入流程圖。雙擊該控制項，在 Property 裏依次添加需要識別的短句。在這個程式裏，需要識別的短句依次是“你好”、“你叫什麼名字”、“你從哪來”、“你也是機器人嗎”、“真聰明”、“再見”。

在“Recognition Result”裏，將語音識別的結果保存變數設置爲 SR_Res，這樣每次識別結果都會被賦值到 SR_Res 中，供程式邏輯判斷用。

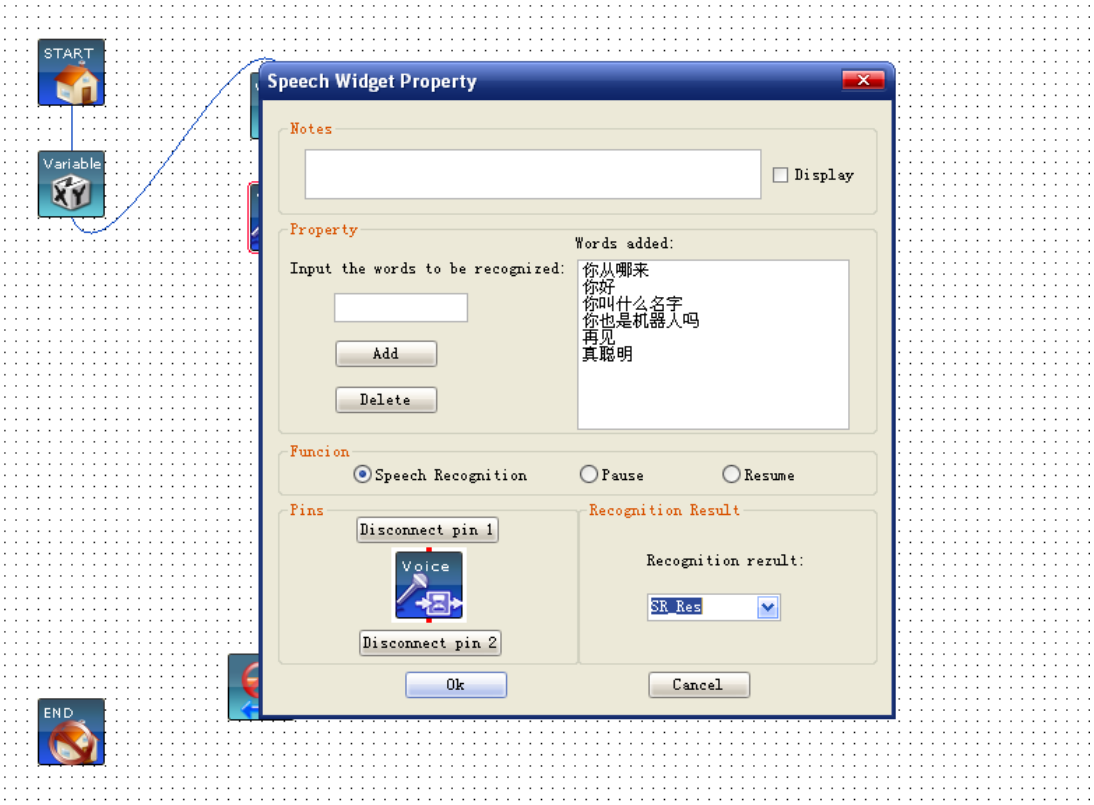


圖 9.15 識別詞條

9.5.4 搭建邏輯程式框架

得到了識別的結果，下面我們可以開始進行成判斷邏輯的框架搭建。識別的結果有六種，所以這裏我們使用兩個 Switch 控制項來構建判斷邏輯。



圖 9.16 Switch 框架

邏輯框架搭建好後，流程圖生成的臨時代碼如下：

Code Window

```
#include  "background.h"
int main(int argc, char * argv[])
{
    int  SR_Res = 0;
    MFSRInsertText(0,(char*)&("你从哪来"),9);
    MFSRInsertText(1,(char*)&("你好"),5);
    MFSRInsertText(2,(char*)&("你叫什么名字"),13);
    MFSRInsertText(3,(char*)&("你也是机器人吗"),15);
    MFSRInsertText(4,(char*)&("再见"),5);
    MFSRInsertText(5,(char*)&("真聪明"),7);
    MFSRStart();
    MFInit();
    MFSetPortDirect(0x00000FFF);
    //语音识别结果
    while (1)
    {
        MFSRGetResIndex();
        //Switch_Part1
        switch (SR_Res)
        {
            default :
            //Switch_Part2
            switch (SR_Res)
            {
            }
        }
```

圖 9.17 Switch 框架程式碼

在上面的程式碼中，可以看到插入識別詞條的函數 MFSRInsertText()第一個參數是一個唯一的數位，這個數位就是識別結果對應的返回值。如：

```
MFSRInsertText(0,(char*)&("你從哪來"),9);
```

則當控制器識別到“你從哪來”詞條的時候，會返回結果“0”。在我們這個程式裏，因爲在 Speech 控制項裏進行了設置，所以這個結果會賦值到 SR_Res，後面的程式中只要對這個變數進行判斷即可。

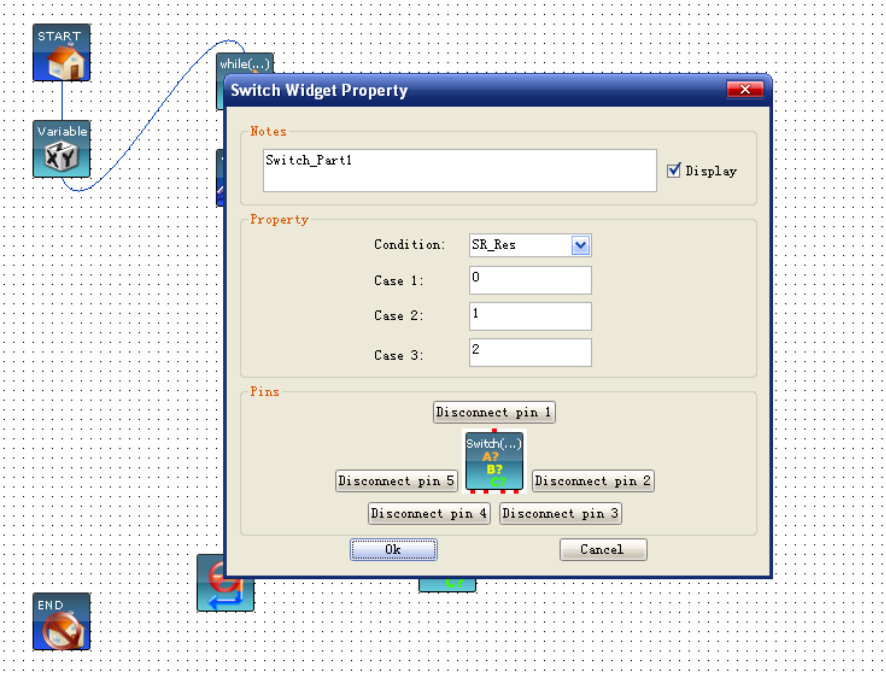


圖 9.18 Switch 控制項一

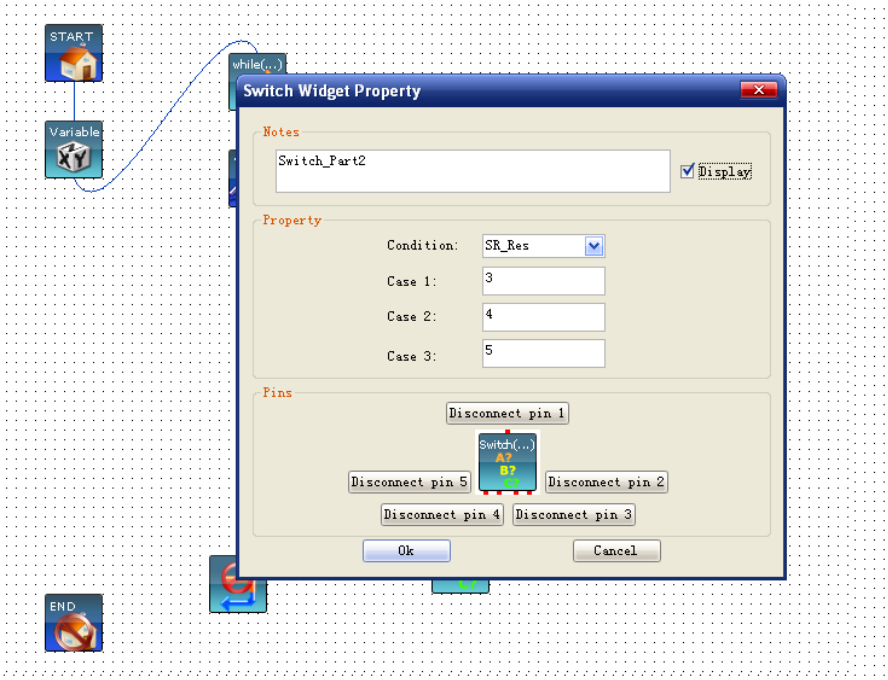


圖 9.19 Switch 控制項二

9.5.5 添加回答動作

在前面的程式中，我們用 Switch 控制項將識別結果做出了 6 個分支，每個分支對應一

條語句的識別處理。下面，我們從“Plug-in Widget”分類中選擇 MP3 控制項，並按住滑鼠左鍵拖曳入流程圖。雙擊該控制項，跳出該控制項的屬性框，在 Property 中選擇我們之前錄製的 mp3 檔案，需要注意識別結果的分支與回答語句的對應關係。

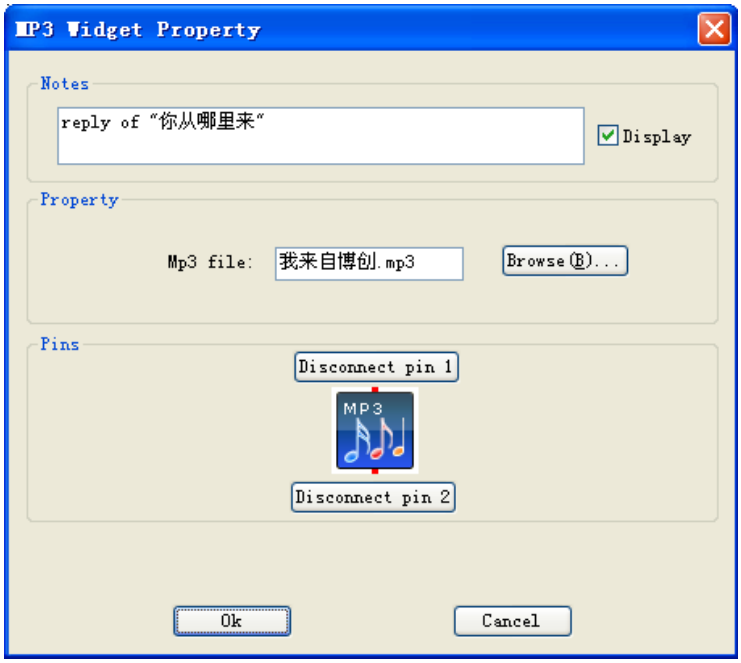


圖 9.20 軀體的搭建

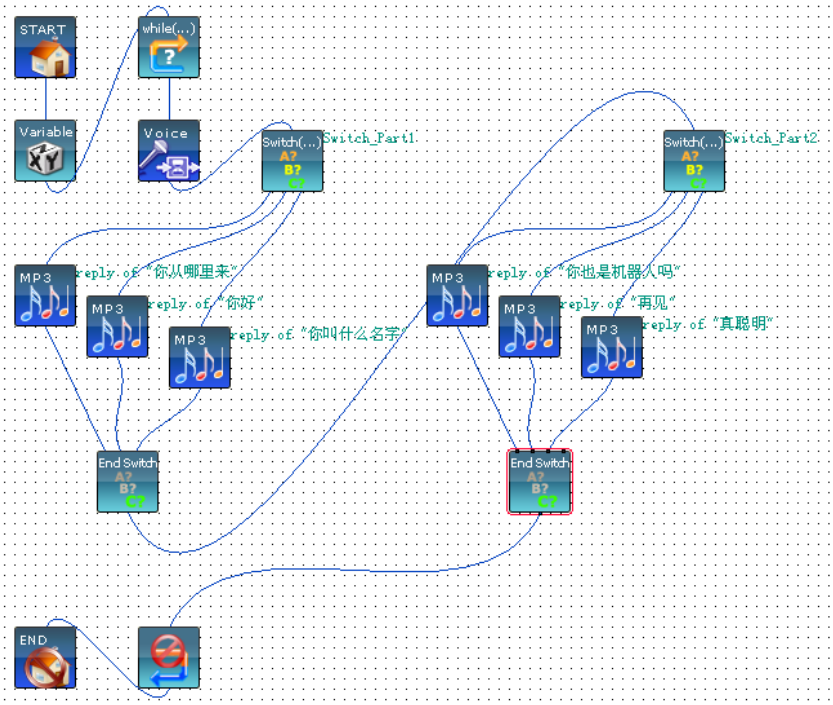


圖 9.21 軀體的搭建

流程圖完成後生成的 C 語言程式如下：

```
#include    "background.h"
int main(int argc, char * argv[])
{
      int    SR_Res = 0;
      MFSRInsertText(0,(char*)&("你從哪來"),9);
      MFSRInsertText(1,(char*)&("你好"),5);
      MFSRInsertText(2,(char*)&("你叫什麼名字"),13);
      MFSRInsertText(3,(char*)&("你也是機器人嗎"),15);
      MFSRInsertText(4,(char*)&("再見"),5);
      MFSRInsertText(5,(char*)&("真聰明"),7);
      MFSRStart();
      MFInit();
      MFSetPortDirect(0x00000FFF);
      //語音識別結果
      while (1)
      {
            MFSRGetResIndex();
            //Switch_Part1
            switch (SR_Res)
            {
                  case 0:
                  //reply of "你從哪里來"
                  MFMp3Play((char*)&("我來自華亨.mp3"),14);
                  break ;
                  case 1:
                  //reply of "你好"
                  MFMp3Play((char*)&("你好.mp3"),8);
                  break ;
                  case 2:
                  //reply of "你叫什麼名字"
                  MFMp3Play((char*)&("我是創意之星.mp3"),16);
                  break ;
            }
            //Switch_Part2
            switch (SR_Res)
            {
                  case 3:
                  //reply of "你也是機器人嗎"
                  MFMp3Play((char*)&("是的.mp3"),8);
                  break ;
                  case 4:
                  //reply of "再見"
                  MFMp3Play((char*)&("bye.mp3"),7);
                  break ;
```

```
                case 5:
                //reply of "真聰明"
                MFMp3Play((char*)&("謝謝.mp3"),8);
                break ;
            }
        }
}
```

9.6 任務 6：程式的執行

所有工作結束後，就可以編譯並下載到機器人實體進行驗證了。點擊 NorthSTAR 工具欄上的編譯按鈕，查看 Output Window 裏面的編譯資訊，提示“Compile succeded…”說明編譯成功。

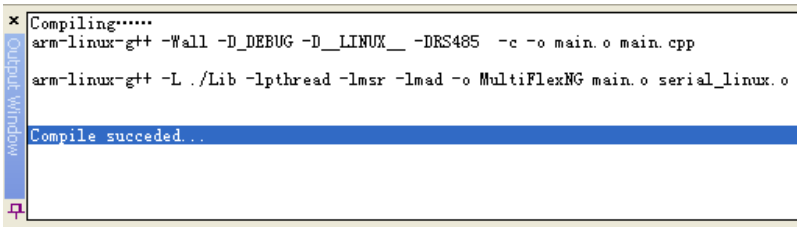


圖 9.22 編譯成功

點擊工具欄上的下載按鈕，查看 Output Window 裏面的編譯資訊，提示“downloading succeded…”說明下載成功。

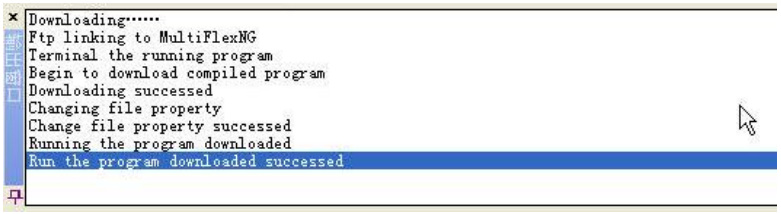


圖 9.23 下載成功

將機器人放在桌上，爲控制器接上外置喇叭和麥克風。點擊工具欄上的執行按鈕，程式便啓動執行，可以開始按照我們設計好的劇本與他對話了。

9.7 任務 7：思考與提高

在這章實驗裏，我們借助 NorthSTAR 的 Speech 控制項在創意之星 II 高級版控制器上實現了語音識別功能。並和我們設計好的對話劇本結合，實現了一個語音對話問答機器人。功能的主體實現了，但是是否還有提高的餘地呢？

任務 2 裏搭建機器人時安裝的伺服馬達和感測器都未用上；機器人會說的話還比較少；能否讓感測器也參與到交互中來；發揮想像力，一定會出現屬於您自己的機器人。

第十章 PXA270全向運動足球機器人

10.1 任務 1：目標分析與計畫

本章的目的是讓您透過本章的學習，瞭解和掌握 MultiFLEX™2-PXA270 控制器視覺處理的使用方式，掌握全向輪的使用特點。本章的教學道具是一套帶視覺的全向運動機器人，和前面的章節一樣，您需要根據本章的指引將之一步步實現出來。

10.1.1 瞭解機器人足球賽

RoboCup 機器人足球世界盃賽及學術大會是國際上級別最高、規模最大、影響最廣泛的機器人足球賽事和學術會議，每年舉辦一次。機器人足球賽的最初想法是由加拿大不列顛哥倫比亞大學的艾倫·馬克沃斯教授于 1992 年提出的。日本學者迅速對此進行了系統的調研和可行性分析。1993 年 6 月，淺田塾和北野宏明等著名學者決定創辦日本機器人足球賽，並暫命名爲 RoboCup J 聯賽（J 聯賽是日本足球職業聯賽的名稱）。一個月之內，他們就收到了來自國外的熱烈反響，要求將其擴展爲一個國際性的比賽，並將其定名爲機器人足球世界盃賽（the Robot World Cup Soccer Games），簡稱 RoboCup。

RobuCup 迄今爲止已經辦了 12 屆，比賽規則難度不斷提高，參加比賽的機器人隊伍的技術也有了突飛猛進的提高。參賽機器人經過 12 年的進化，和剛開始幾年的機器人相比有了明顯的不同。剛開始幾年的機器人一般使用兩輪差動式機構，隨著全向輪的發明，全向機器人迅速取代常規的兩輪差動式機器人。全向機器人的特點是可以在任意一點上，都可以往任何方向前進，沒有規定的前方，不用轉彎，在賽場非常靈活。

我們在 1.2.6 小節對全向機器人的特點進行了詳細的介紹，您可以重新翻閱這部分的內容。

10.1.2 瞭解全向輪的特點

全向機器人的根本特徵是使用全向輪，可以任意方向行進。那麼我們需要瞭解一下什麼事全向輪，與常規的輪子有何區別。



圖 10.1 各種全向輪

上圖是當前比較常見的全向輪，根據載重的不同，全向輪的大小、面積各有不同。

如圖 10.2 所示，我們可以看到全向輪整體是一個圓形，可以和常規輪子一樣由馬達帶

動，實現轉動。全向輪四周鑲嵌著可以自由滾動的輥子，這些輥子可以在垂直於輪軸的方向自由轉動。根據上述的構造特點，全向輪可以當成普通輪子，實現輪軸方向轉動，也可以在地面自由滑動。

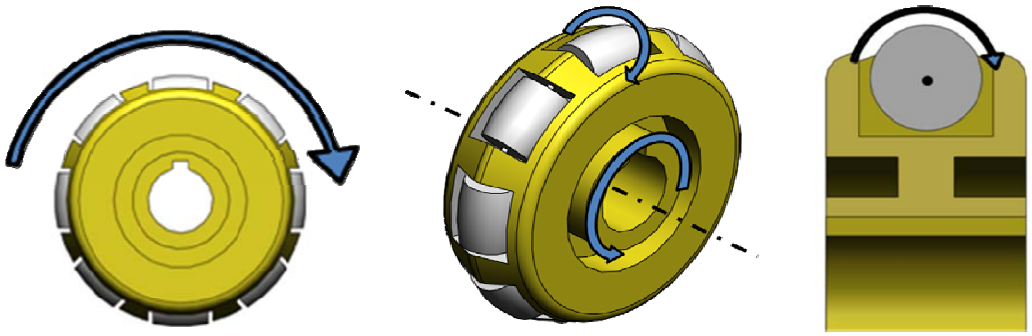


圖 10.2 全向輪轉動特點

10.1.3 構建我們的全向底盤

“創意之星”高級版有 20 個 CDS5500、四個全向輪和眾多結構零件，用於搭建一套全向機器人是足夠的。

透過第四章任務一的分析，我們對機器人底盤的搭建有了一定的思路。首先，我們確定使用幾個驅動輪子，這個控制方式密切相關。

1、2 輪式：它需要配套全向輪用於保持底盤平衡，我們不採用這個方案。

2、3 輪式：3 點可以支撐起一個平面，這個方案是可行的。如下圖所示，我們採用圓周佈局，將三個全向輪呈 120°夾角排列。1 號、2 號輪子如箭頭方向轉動，3 號輪子作爲從動輪，機器人即可向兩個轉動方向的向量累加方向前進。同理可以設計出機器人水準側移、後退、原地旋轉等等方向的運動。

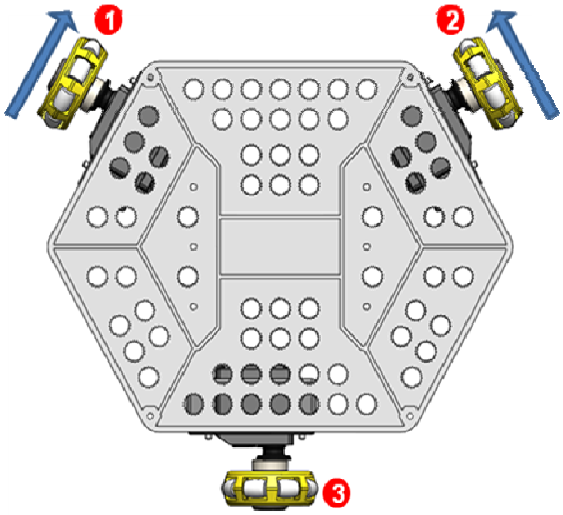


圖 10.3 3 輪全向平臺

3、四輪：如下圖所示，四個全向輪圓周排列，夾角 90°，由四個輪子的轉動方向進行向量累加可以方便地得到前進、後退、轉向、側移等各種動作。

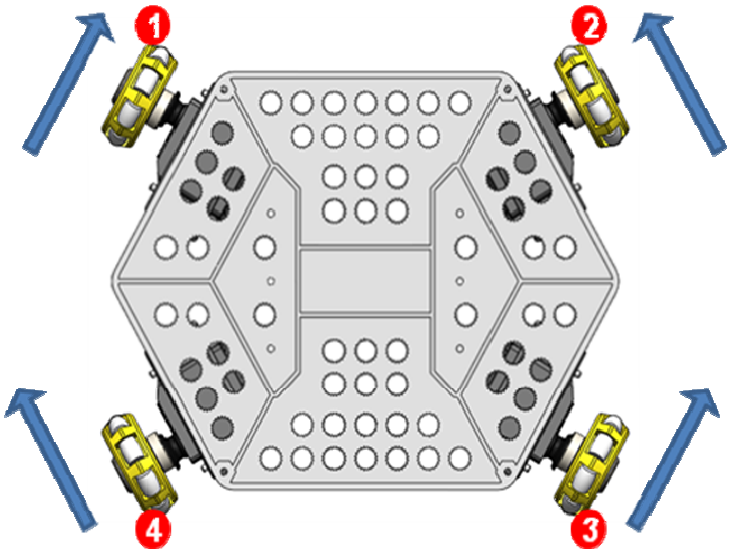


圖 10.4　4 輪全向平臺

根據上述分析可知，3 輪和 4 輪都是可行的方案，根據圖 10.3 和 10.4 的對比可以發現，3 輪的動力要遠小於 4 輪結構，比如在前進的時候只有 2 個輪子起作用。我們最常用的方向是前進、後退、側移、旋轉，3 輪全向的側移很不好控制（您可以透過繪製三個輪子的轉動向量圖進行計算）。所以，我們選擇 4 輪全向結構。

10.1.4 視覺系統

全向運動足球機器人由兩大系統構成：全向運動平臺、視覺系統，上一小節我們已經選定了全向運動平臺的結構。視覺系統我們的選擇是唯一的：攝影機和 MultiFLEX™2-PXA270 控制器。

在“創意之星”高級版套裝中，包含了一個 USB 攝影機，借助 MultiFLEX™2-PXA270 控制器強勁的性能，我們能夠在這個羽量級的平臺上也能實現大型機器人才具備的顏色識別。

MultiFLEX™2-PXA270 控制器使用 NorthSTAR 開發環境，您不用編寫圖像識別引擎，只需要設置 NorthSTAR 視覺空間的屬性，就能夠得到我們目標物（小足球）在圖像範圍內的座標。根據這個座標資訊就可以判定機器人和足球的相對位置，進而控制機器人的運動，去接近足球。

10.1.5 任務規劃

1、搭建全向平臺和視覺系統，這是接下去實驗的基礎。

2、熟悉圖像識別的基礎知識。

3、編寫程式，進行圖像採集和機器人基本運動控制。

4、對顏色進行標定，去除背景干擾，實現比較好的識別效果，

5、完成整個項目的程式設計。

10.2 任務 2：結構設計和攝影機安裝

如圖 10.5 所示，使用創意之星 II 的結構零件，我們可以輕易的構造一個全向移動的機器人底盤。以下是搭建的步驟：

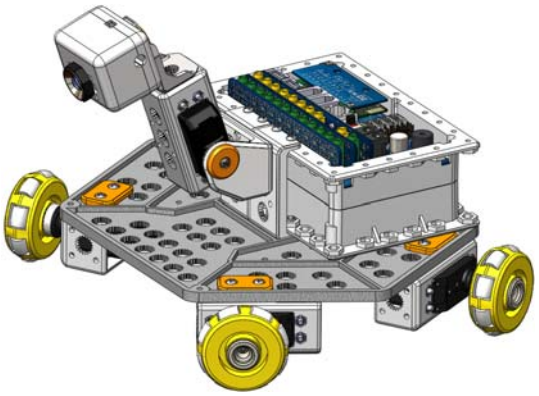


圖 10.5 全向運動足球機器人

第一步 搭建運動平臺

《創意之星機器人套件組裝指南》5.4 小節以示意圖的方式對全向運動平臺的搭建過程進行了詳細的說明。您可以依照該指南進行平臺搭建，完成到圖 10.6 的程度即可。



圖 10.6 全向運動平臺

第二步 搭建視覺部分

如圖 10.7 我們給機器人設計了一個帶攝影機的頭部，圖示的攝影機外殼是用“創意之星”頭部的外殼手工改裝的。您大可不必如此麻煩，直接將攝影機固定在 L2-1 零件上即可。

我們爲機器人視覺部分做了一個可以俯仰的關節，這樣機器人就可以像人一樣俯仰頭部，尋找目標物。

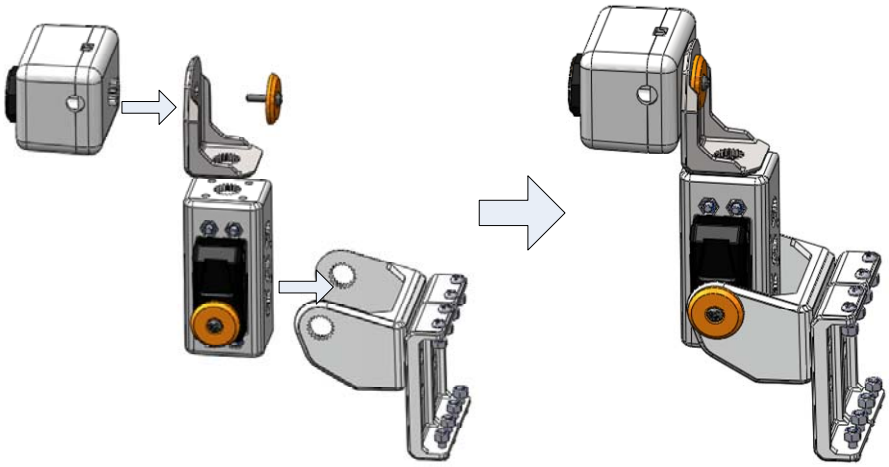


圖 10.7 搭建視覺部分

第三步 總裝

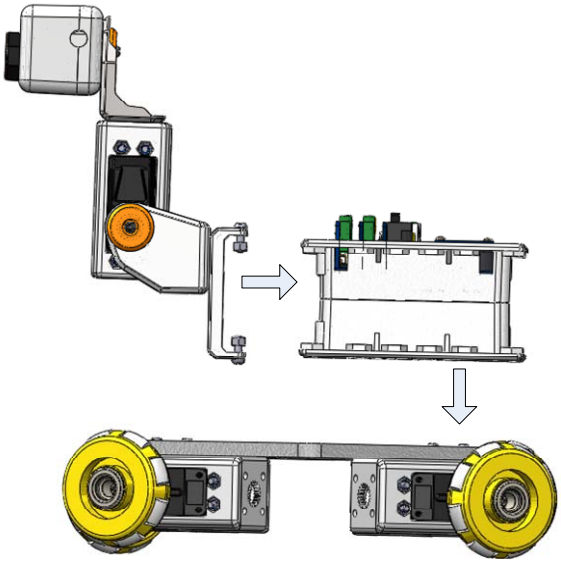


圖 10.8 總裝

如圖 10.8，將視覺部分透過 U3H 固定到控制器上，將控制器固定到底板上，就完成了整台機器人的搭建。

第四步 接線：

如圖 10.9 所示，將 1、2、3、4 號伺服馬達作爲底盤驅動伺服馬達，串聯起來接到控制器的機器人伺服馬達介面。將 5 號伺服馬達接到其他的機器人伺服馬達介面。將攝影機的 USB 接到控制器的任意 USB 介面，接線即告完畢。

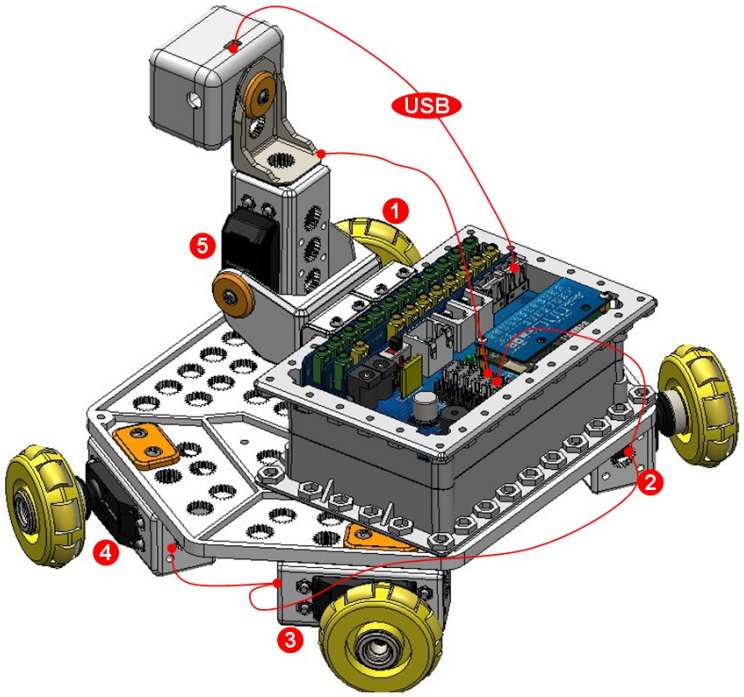


圖 10.9 全向足球機器人效果圖

10.3 任務 3：原理分析

在程式中，獲得的圖像資料是一個一維的陣列，原始的圖像資料通常是 RGB24 格式，即每三個位元組一個圖元，三個位元組分別是該圖元紅色、綠色、藍色的顏色分量值。

在 RGB 顏色空間下一般很難排除光照的影響，因爲在亮度突變的環境裏，三個顏色分量都會劇烈變化，所以不適合用來做顏色區分。常規的做法是將這個圖像陣列轉換到 HSI 空間去。HSI，即色調（Hue）、色飽和度（Saturation）和亮度（Intensity），其中前兩項受亮度的影響很小，各種顏色在色度（Hue）分量的方向上是連續的均勻分佈的，是極好的顏色標定坐標系。

對畫面中所有圖元進行轉換後，依次對各點的 H 值進行排查，符合範圍要求的就作爲有效點，否則爲無效點。遍曆完整幅圖像的圖元之後，就將有效點的 x 和 y 分別求平均值，則可以得到該顏色區域在畫面中的顏色質心的座標。在顏色干擾不是很嚴重的情況下，我們可以認爲桔色區域的質心就是桔色球的中心。

透過桔色球的中心位置，可以判斷球相對於機器人的大概方位。在視頻圖像平面中，橫向爲 X 軸，縱向爲 Y 軸，透過球中心的 X 值可以判斷球相對機器人正前方是偏左還是偏右，若球中心 X 小於畫面中心點 X 值說明球在機器人左前方，否則反之。透過球中心的 Y 值就可以判斷球與機器人的距離，雖然距離值和球中心 Y 值不是嚴格的線性關係，但是變化趨勢是一致的，球離機器人越遠，球中心在畫面中的 Y 值就越大。

知道了球相對機器人的方向和距離，要做追球動作就很簡單了，偏左了就左轉、偏右了就右轉，中間設個無效區。球遠了就前進，近了就後退。這兩種運動可以透過兩個馬達速度進行合成。

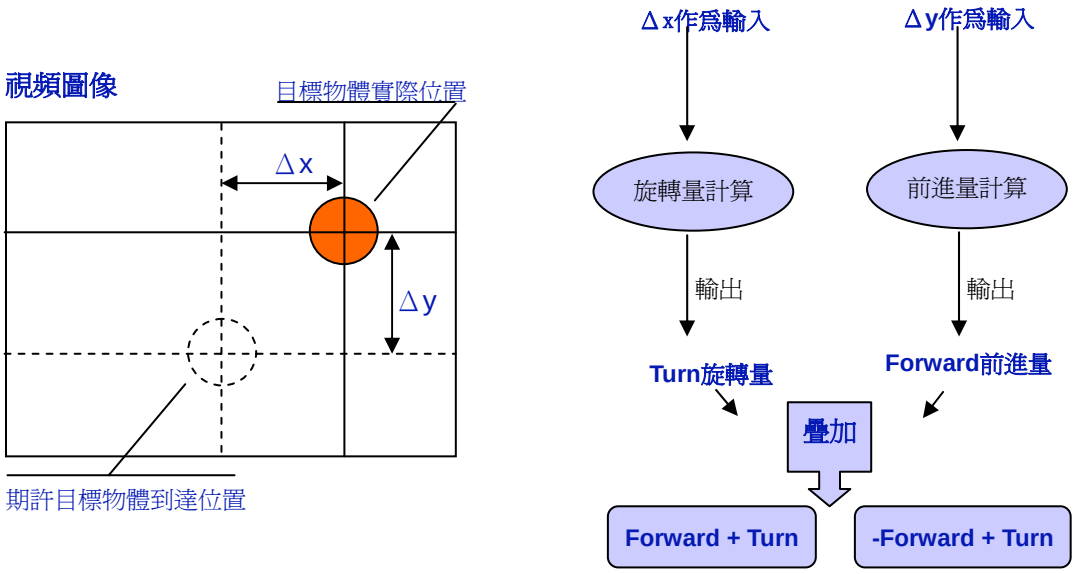


圖 10.10 視覺原理

- **標定速度**

理論上的數值計算已經弄清楚了，下面需要跟實際的硬體平臺結合起來。首先我們需要對伺服馬達轉動方向與底盤的運動方向進行一下標定測試。首先，將底盤右側兩個伺服馬達與多功能調試器連接，在調試器連接的 PC 上執行 CDS5500 的調試程式。查找設備，可以看到 1 號和 2 號兩個伺服馬達，正是我們所連接的兩個伺服馬達。停止查找，點擊“確定”按鈕進入操作介面。

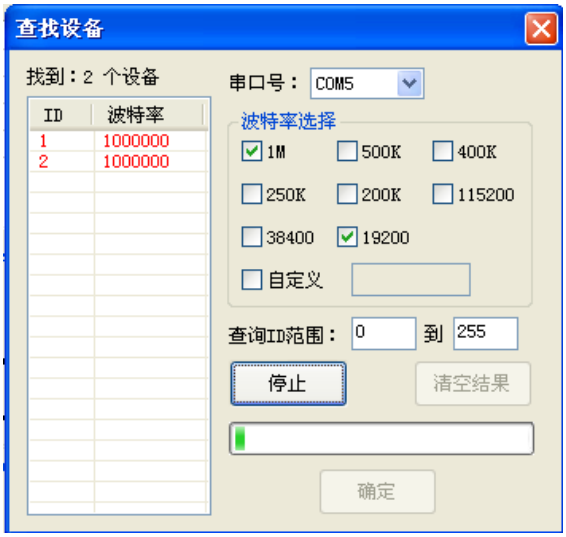


圖 10.11 搜索匯流排上的伺服馬達

進入操作介面後，在左邊設備列表裏選擇 1 號伺服馬達，在控制欄裏選擇力矩輸出，然後拖動速度滑動條，在正負速度之間測試。可以看出，負速度時該伺服馬達造成的效果是底盤右側往前運動，正速度則使右側底盤往後運動。

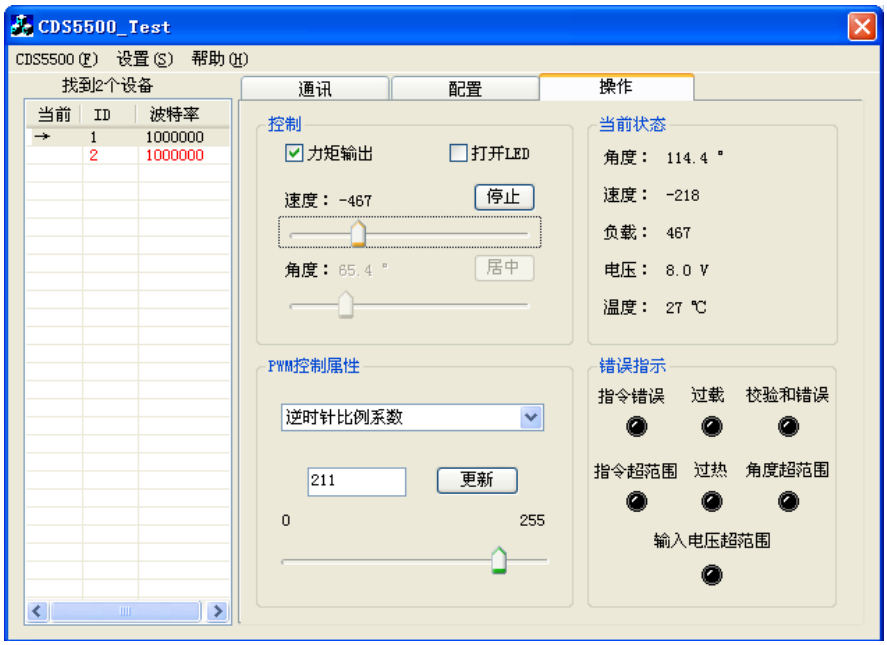


圖 10.12 標定 1 號伺服馬達方向

標定完 1 號伺服馬達方向，切換到 2 號伺服馬達。同樣的在正負速度之間切換，可以看到，2 號伺服馬達對底盤的運動影響和 1 號伺服馬達一致。得出結論：底盤上右側的伺服馬達，當他們的速度符號相同時，對底盤運動的影響是一致的。對右側的伺服馬達來說，正速度使底盤右側向後運動，負速度則使底盤右側向前運動。

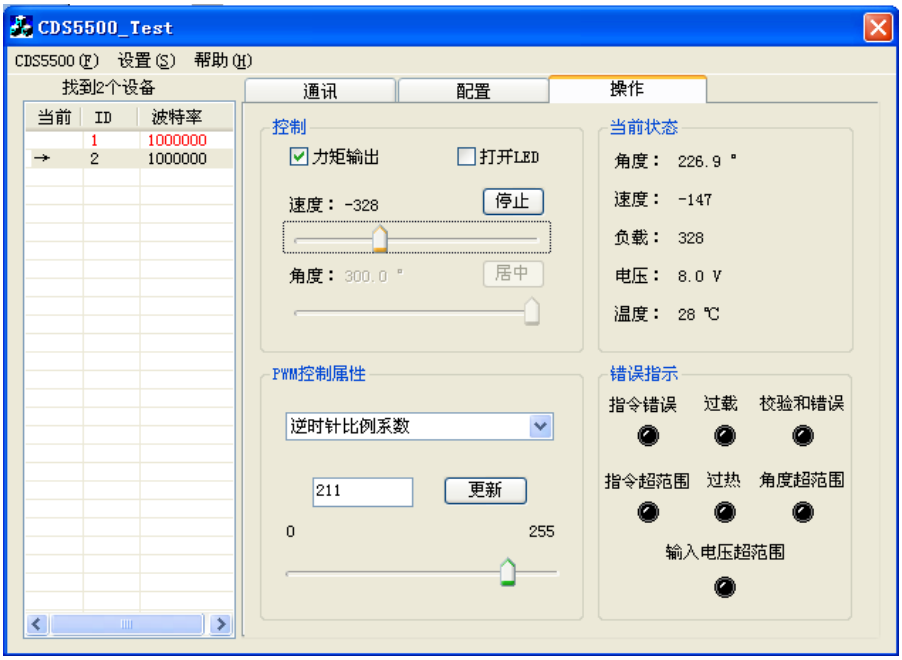


圖 10.13 標定 2 號伺服馬達方向

使用相同的方法標定左側的 3 號 4 號伺服馬達，將會得出結論：底盤上左側的伺服馬達，當他們的速度符號相同時，對底盤運動的影響是一致的。對左側的伺服馬達來說，正速度使底盤左側向前運動，負速度則使底盤左側向後運動。

- **速度合成**

機器人運動過程，主要是前進和自身旋轉兩種運動。

前進：根據上面的標定結果，左側伺服馬達速度爲正數，右側伺服馬達速度爲負數時，機器人將向前運動。反之則向後退。

旋轉：當兩側所有伺服馬達速度爲正數時，機器人將向右順時針自轉；當兩側所有伺服馬達速度爲負數時，機器人將向左逆時針自轉。

在實際的運動過程中，這兩種運動是可以累加的，只需要將各動作的速度值在每個運動伺服馬達上進行簡單求和即可實現。具體的運動學分析過程，可以自行進行推算，在此不再詳述。

10.4 任務 4：程式編寫

新建專案

在 NorthSTAR 軟體中新建一個 Project，跳出 Project Configuration 頁面。“Controller”欄選擇我們的編程物件：MultiFLEX2-PXA270；因爲我們搭建的是非標準結構，“Robot”欄選擇“Customized”。點擊“next”，跳出 Servo 頁面。

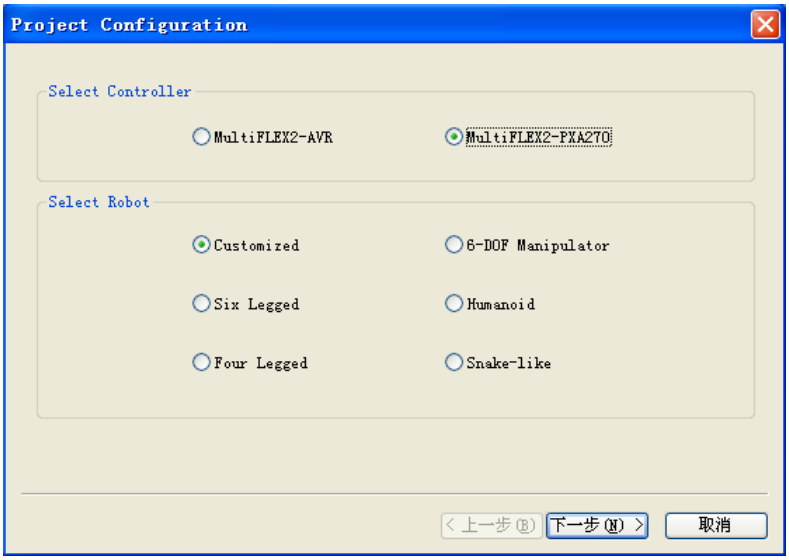


圖 10.14 新建專案

在 Servo 頁面中，我們爲機器人設置五個伺服馬達，1～4 號伺服馬達爲機器人底盤馬達，配置成 Motor Mode，5 號伺服馬達爲攝影機俯仰控制關節，配置成 Servo Mode。

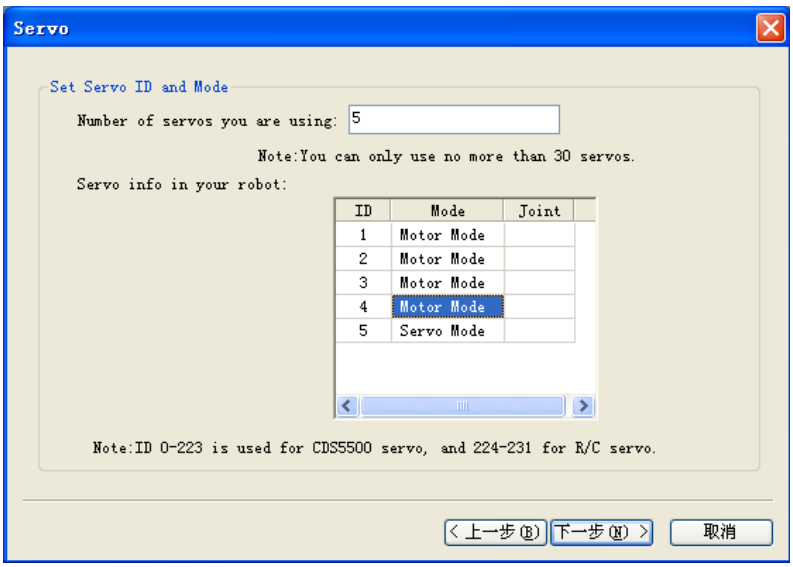


圖 10.15 伺服馬達配置

由於本實驗中未使用其他感測器，所以 AD 和 IO 的數量都設置爲 0。

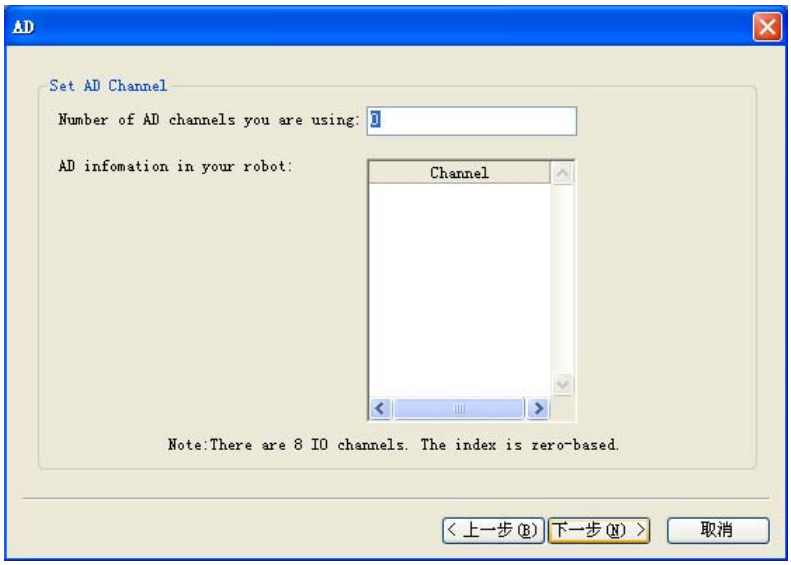


圖 10.16 AD 配置

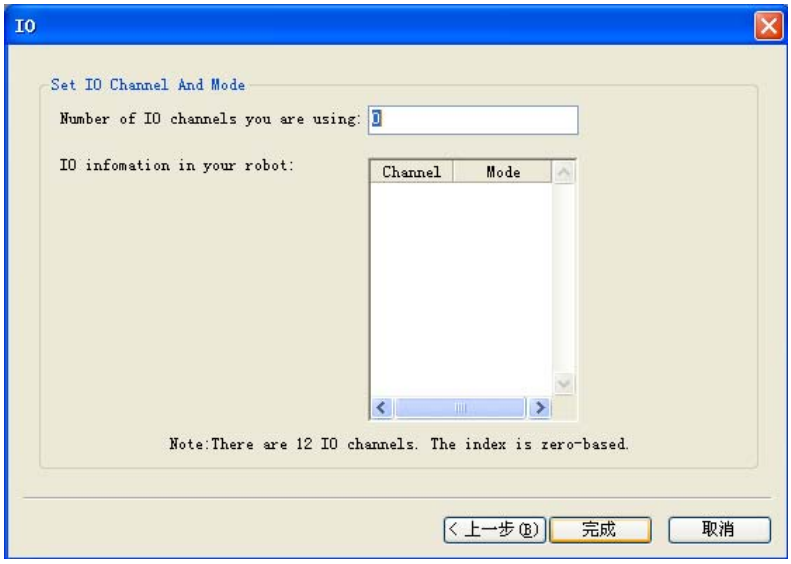


圖 10.17 IO 配置

添加變數

本實驗中將會用到 8 個變數，用於計算過程中的臨時儲存。

變數名	類型	說明
Ball_x	Int	儲存目標球的顏色質心在畫面中的橫坐標
Ball_y	Int	儲存目標球的顏色質心在畫面中的縱坐標
Dx	Int	目標球的顏色質心與期許座標點的 X 軸向差值
Dy	Int	目標球的顏色質心與期許座標點的 Y 軸向差值
Forward	Int	前進分量
Turn	Int	旋轉分量
Left	Int	底盤左側伺服馬達的速度值
Right	Int	底盤右側伺服馬達的速度值

添加變數的方式有兩種：一是在流程圖空白處右擊滑鼠，在跳出的功能表中選擇“Add variable”；二是從左側控制項欄“Syntax Widget”裏拖曳一個 Variable 控制項到流程圖裏。兩者的效果是一樣的。

在 Variable 控制項的屬性裏，輸入變數名及變數類型。在本實驗中，共需要八個變數，每個變數對應一個控制項。

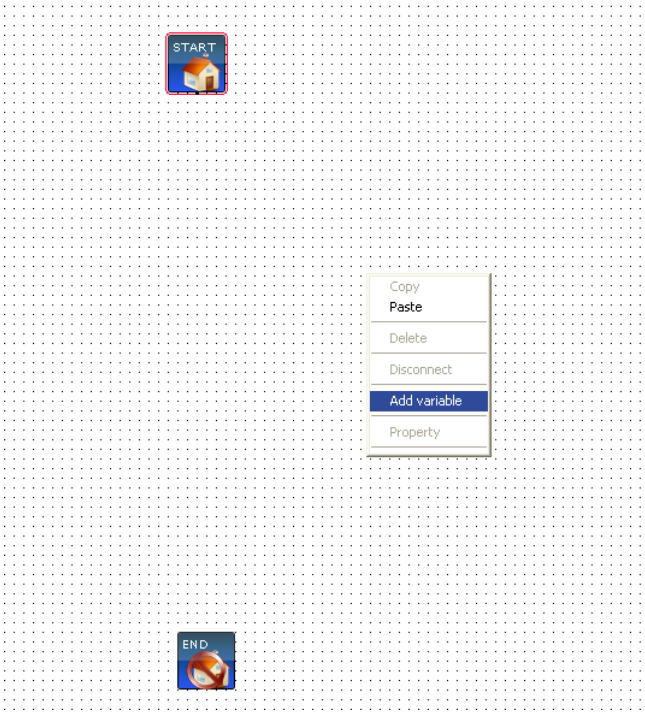


圖 10.18 添加變數

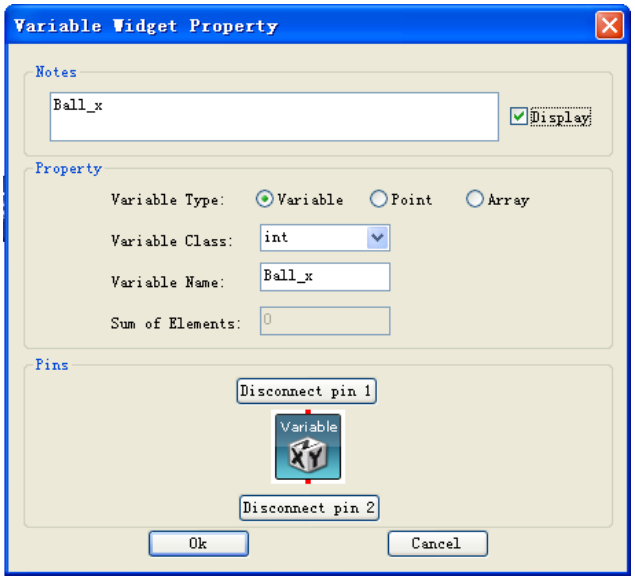


圖 10.19 變數控制項

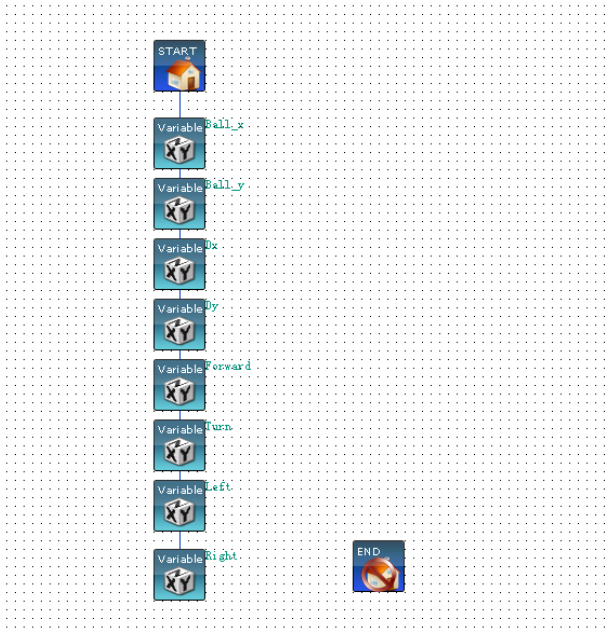


圖 10.20 變數添加完成

初始化

從左側控制項列表“Function Widget”裏拖曳一個 Servo 控制項至流程圖中。雙擊該控制項，在跳出的屬性框裏進行伺服馬達的初始化。將 1～4 號伺服馬達的 Speed 設置爲 0，將 5 號伺服馬達的 Angle 設置爲 512，Speed 爲 256，這是一個停止的狀態，初始化完畢。

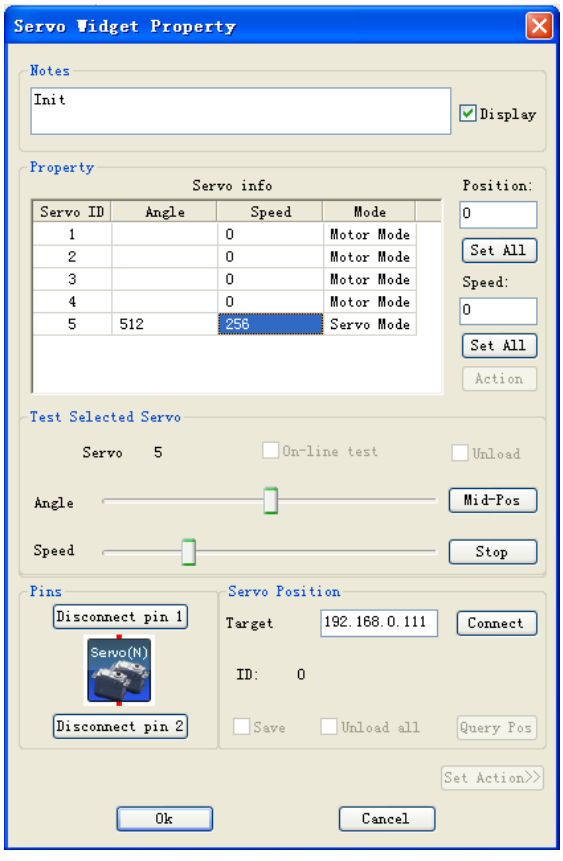


圖 10.21 初始化伺服馬達

獲取識別結果

從左側控制項列表“Plug-in Widget”裏拖曳一個 Vision 控制項至流程圖中。雙擊該控制項，在跳出的屬性框裏設置 Return Value 的賦值變數，如圖中設置了 Ball_x 和 Ball_y 變數，則當視覺系統識別出目標球的顏色質心時，將會將該質心在畫面中的 x、y 座標值分別賦到 Ball_x 和 Ball_y 變數中。

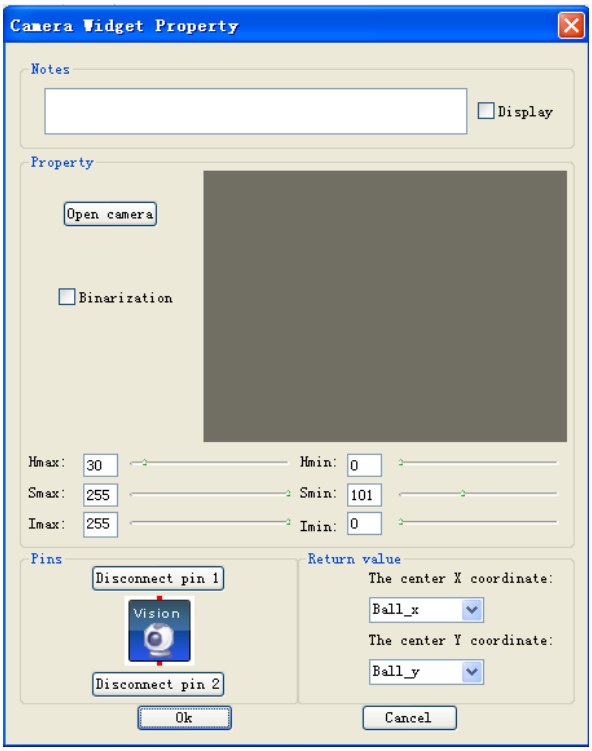


圖 10.22 設置 Return Value

速度計算

有了目標球的座標，我們就可以開始進行控制量的計算了。要使機器人能跟上球，一般我們期許球在圖像中能處於（160,60）這個座標點，在具體實驗時，可以根據情況上下浮動。

首先我們計算目標球當前位置在 X 軸向上與預期目標點的差值：

$$Dx = Ball\_x - 160$$

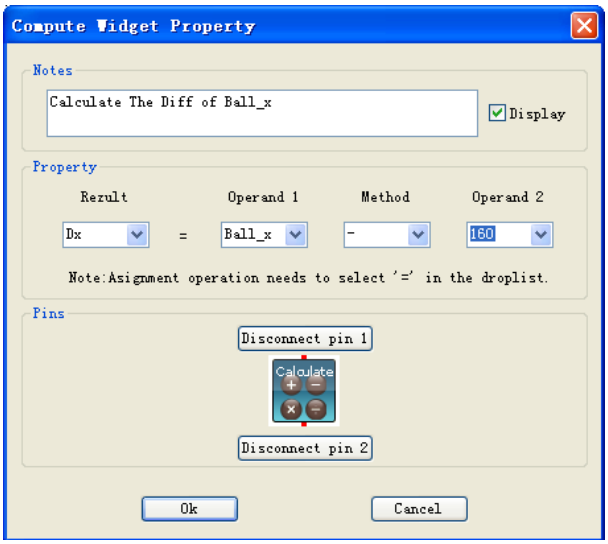


圖 10.23 計算 Dx

同樣的，還需要計算目標球當前位置在 Y 軸向上與預期目標點的差值：

Dy = Ball_y – 60

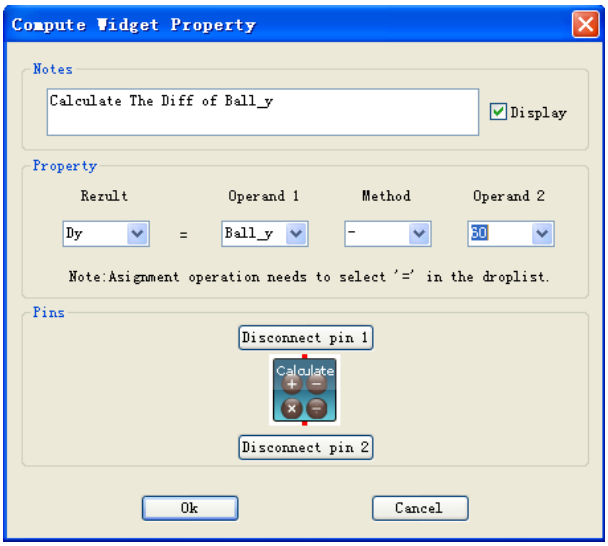


圖 10.24 計算 Dy

根據這兩個差值，我們可以計算出一個這種差距的速度值。這裏我們使用一種叫做“比例控制”的控制方法，將誤差的輸入量進行一個比例放大後直接得到輸出量。

我們將 Y 軸向上的差值放大 10 倍後，作爲前進速度分量的速度值：

Forward = Dy × 10

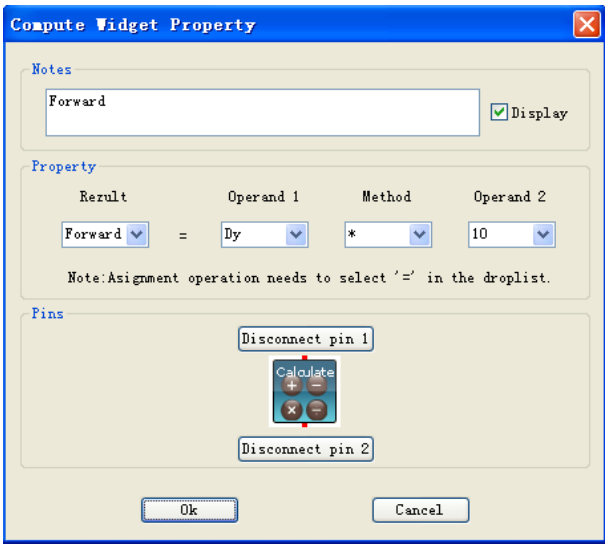


圖 10.25 計算 Forward

然後將 X 軸向上的差值放大兩倍後，作爲旋轉速度分量的速度值：

Turn = Dx × 2

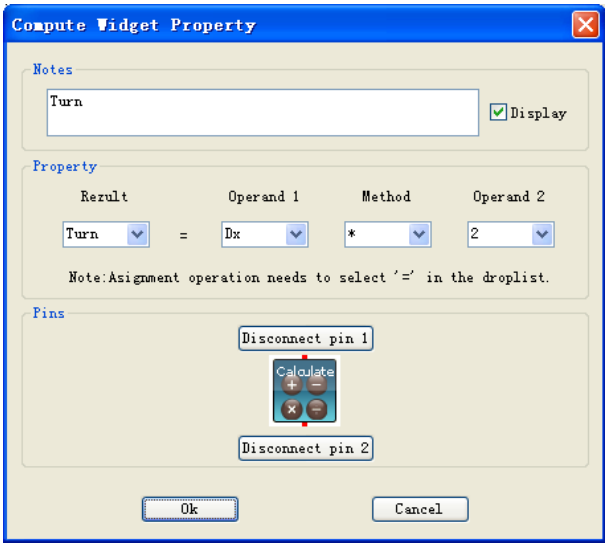


圖 10.26 計算 Turn

有了前進速度和旋轉速度，我們就可以將其進行累加了。根據原理分析階段我們標定出的速度與實際方向的關係，累加的結果應該是：

Left = Forward + Turn

Right = Turn - Forward

其中，Left 是底盤左側所有伺服馬達的速度值，Right 是底盤右側所有伺服馬達的速度值。

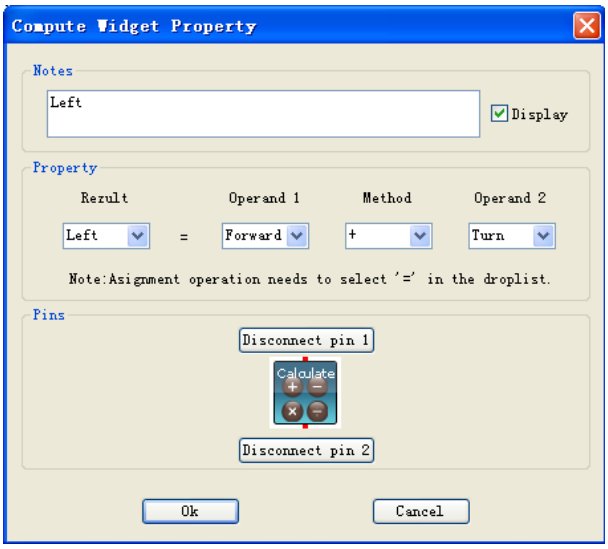


圖 10.27 計算 Left

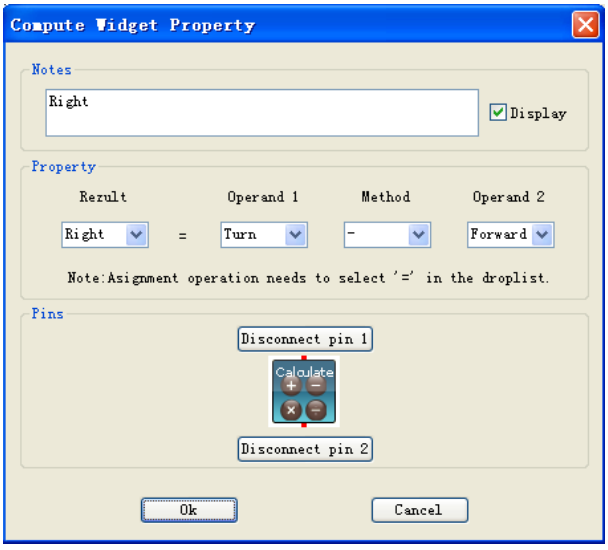


圖 10.28 計算 Right

執行

運算的過程結束了，下面將運算結果輸出，變成機器人移動的速度。從左側控制項列表“Function Widget”裏拖曳一個 Servo 控制項至流程圖中。雙擊該控制項，在跳出的屬性框裏進行伺服馬達速度的賦值。1 號和 2 號伺服馬達在底盤的右側，所以我們將 Right 值作爲他們的速度值。3 號和 4 號伺服馬達在底盤的左側，所以其速度值爲 Left。控制俯仰的 5 號伺服馬達，我們爲其賦值 600，讓攝影機俯下一些，這樣可以跟地面成一定的角度，方便進行距離控制。至此，一個閉環的控制週期完成，我們添加一個 while 控制項將這個過程周而復始的迴圈起來，就能讓機器人動態的跟蹤一個帶顏色的目標球了。

爲了讓控制過程更有效率，通常在這個 while 迴圈裏面，我們會加入一個 Delay 控制項，進行 250 毫秒左右的延時，這麼做的目的是爲了等待新的圖像幀的到來，保證每個週期開始執行時，視頻圖像已經更新完畢。避免出現相同的結果，重複的發送伺服馬達速度指令，減少過多的指令在匯流排緩衝過程中可能出現堵塞或丟失的風險。

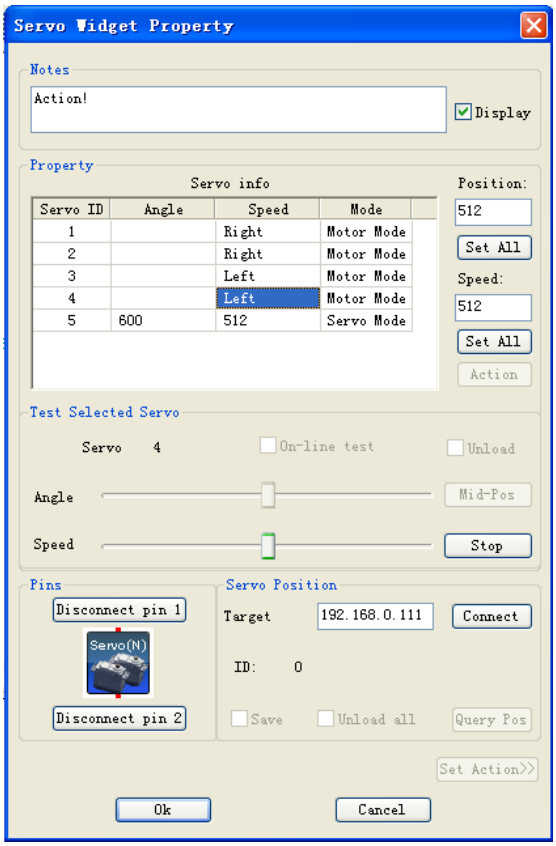


圖 10.29 伺服馬達初始化

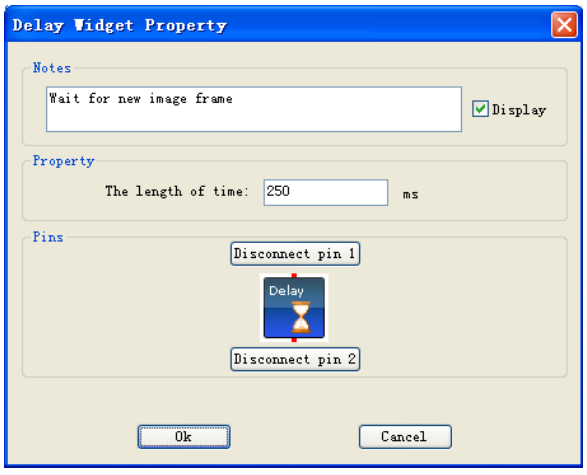


圖 10.30 等待延遲



圖 10.31 程式流程

流程圖生成的 C 語言程式碼如下：

```
#include    "background.h"
int main(int argc, char * argv[])
{
      int    Right = 0;
      int    Left = 0;
      int    Turn = 0;
      int    Forward = 0;
      int    Dy = 0;
      int    Dx = 0;
      int    Ball_y = 0;
      int    Ball_x = 0;
      MFCapOpen();
      MFCapSetH(30,0);
      MFCapSetS(255,101);
      MFInit();
      MFSetPortDirect(0x00000FFF);
      MFSetServoMode(1,1);
      MFSetServoMode(2,1);
      MFSetServoMode(3,1);
      MFSetServoMode(4,1);
      MFSetServoMode(5,0);
      {
            MFSetServoRotaSpd(1,0);
```

```
        MFSetServoRotaSpd(2,0);
        MFSetServoRotaSpd(3,0);
        MFSetServoRotaSpd(4,0);
        MFSetServoPos(5,512,256);
        MFServoAction();
    }
    while (1)
    {
        Ball_x = MFCapGetCenterX();
        Ball_y = MFCapGetCenterY();
        //Calculate The Diff of Ball_x
        Dx = Ball_x - 160;
        //Calculate The Diff of Ball_y
        Dy = Ball_y - 60;
        //Forward
        Forward = Dy * 10;
        //Turn
        Turn = Dx * 2;
        //left
        Left = Forward + Turn;
        //Right
        Right = Turn - Forward;
        //Action!
        {
            MFSetServoRotaSpd(1,Right);
            MFSetServoRotaSpd(2,Right);
            MFSetServoRotaSpd(3,Left);
            MFSetServoRotaSpd(4,Left);
            MFSetServoPos(5,600,512);
            MFServoAction();
        }
        //Wait for new image frame
        DelayMS(250);
    }
}
```

10.5 任務 5：顏色閾值標定

程式設計完成，在執行之前，我們需要對目標顏色閾值範圍進行手動標定。顏色標定的目的是爲了將球的顏色標記出來，而將背景中相近顏色去除，幫助機器人識別目標物。如果這一步沒做好，則執行時機器人將可能出現誤識別的情況。

將機器人與 PC 用網路線連接起來，打開機器人開關，稍等 5 秒鐘。待機器人控制器的

系統啓動完畢之後，雙擊流程圖裏的 Vision 控制項，在跳出的屬性框裏點擊“Open camera”按鈕，介面上會顯示機器人看到的圖像。將 Binarization 項勾選致能，圖像中的有效圖元將會用紅色標記出來，拖動畫面下方的閾值範圍滑杆，畫面中的標記也會即時的變動。改變 H 值範圍，將改變標定的顏色；改變 S 值範圍，將改變對色飽和度的要求，通常用於排除背景中相近顏色的干擾。

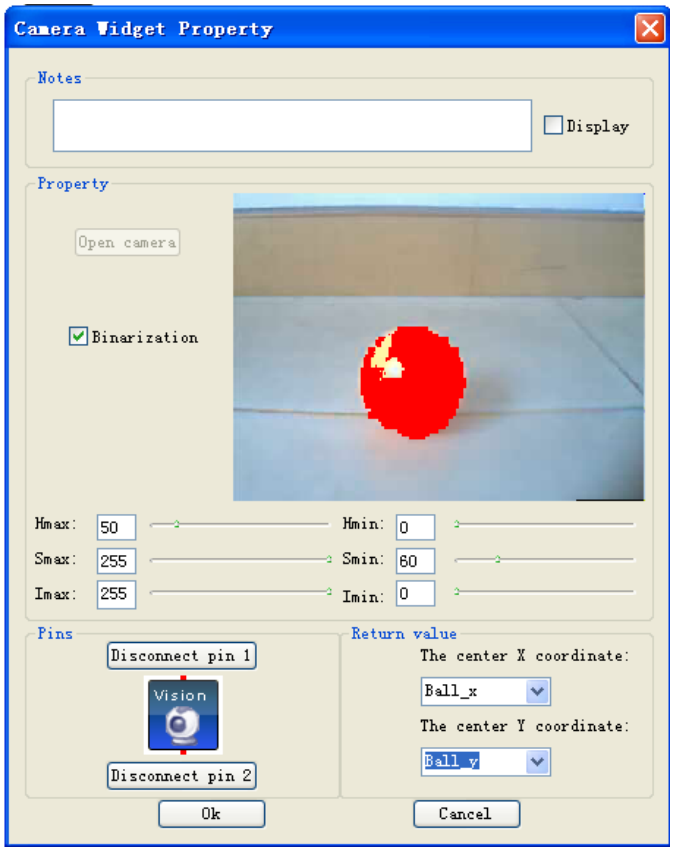


圖 10.32 標定目標顏色閾值

10.6 任務 6：程式的執行

所有工作結束後，就可以編譯並下載到機器人實體進行驗證了。點擊 NorthSTAR 工具欄上的編譯按鈕，查看 Output Window 裏面的編譯資訊，提示“Compile succeded...”說明編譯成功。

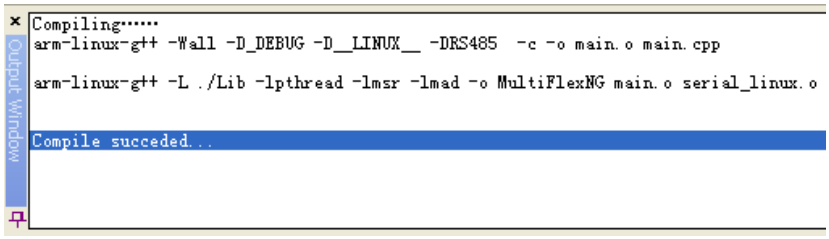


圖 10.33 編譯成功

點擊工具欄上的下載按鈕，查看 Output Window 裏面的編譯資訊，提示“downloading succeded…”說明下載成功。

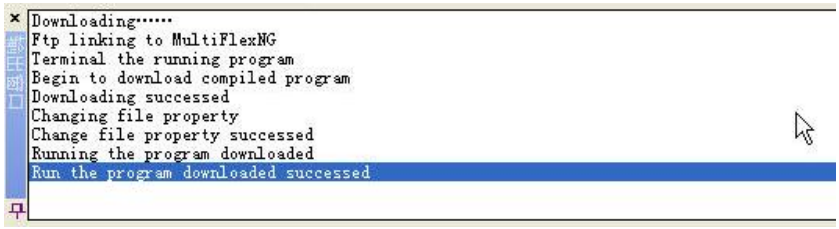


圖 10.34 下載成功

將機器人放在地上，將目標球放在機器人面前 10 釐米處，點擊工具欄上的執行按鈕，機器人便開始按照我們設計的邏輯執行起來了。試著將球拉遠或移近，查看機器人是否像預計的那樣運動。

10.7 任務 7：思考與提高

在這章內容裏，我們借助 NorthSTAR 的 Vision 控制項在“創意之星”高級版平臺上實現了一個追逐圓球的全向機器人，它會隨著面前目標球的移動而前進後退，時刻保持與球的距離以及自己與球的朝向角度。功能的主體實現了，但是是否還有提高的餘地呢？

這個機器人的運動是旋轉和前進的累加，全向移動的特點並未發揮出來，如何實現任意方向的移動？請大家開動腦筋。

我們的程式邏輯是建立在球在機器人可視範圍之內來設計的，如果球離開了視野，跑到機器人背後會怎麼樣呢？

發揮想像力，一定會出現屬於您自己的機器人。

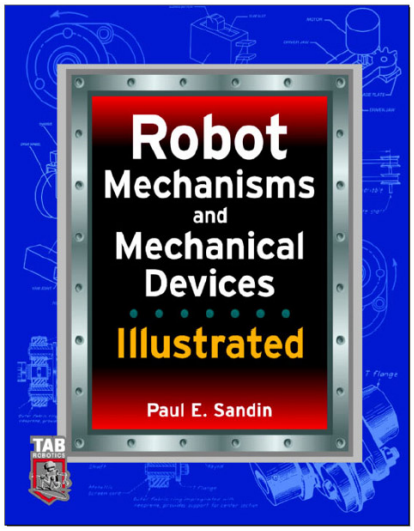
第十一章　拓展閱讀及技術資料

11.1 有用的拓展閱讀資料介紹

鑒於機器人技術如此廣泛而複雜，本書只能針對“創意之星”機器人套件所涉及到的領域做一個介紹，並且不可能很系統性地講述每一方面的知識和技巧。如果讀者希望製作機器人、並且以此爲自己的興趣方向，以下幾本書是我們推薦閱讀的：

1）《Robot Mechanisms And Mechanical Devices》

這本書介紹了機器人製作所需要的大量機構知識。圖文並茂，將原理講的深入透徹，並有大量的實例說明。內容非常全面豐富、系統性強。包括了齒輪、鏈條、蝸輪蝸杆、螺紋、電動機、氣缸、移動機器人原理、減速機等各方面的內容。



2）《Systems and Methods for Mobile Robot Positioning》

這本書的重點是詳細介紹了移動機器人導航所需要的感測器和演算法。內容很全面，介紹了最新的導航演算法進展，以及世界範圍內的各種優秀感測器。



3）《現代機器人學》、《仿生機械學》

有別於傳統的機器人學書籍（主要介紹空間機構學和動力學），這兩本書介紹了大量的仿生學與機器人相關的知識。本書所使用的蛇形機器人的運動規劃、四足機器昆蟲的步態，均受益於這兩本書中的內容。

4）《Build Your Own Combat Robot》

這是一本很有趣的書，一步一步教您如何製作一個戰鬥機器人。其中有大量值得一看的內容，涉及到機器人的機構可靠性和魯棒性方面的技巧。戰鬥機器人(Combat Robot) 是來自北美的一項很有挑戰性的機器人對戰比賽，從這本書中，讀者可以獲取很多結構設計和攻擊方式的靈感，可以有效地用於“機器人武術擂臺賽”的機器人設計和實踐中。

5）《Anatomy of a Robot》

這本書重點介紹了自主機器人的一些理論基礎和實踐。

6）《機器人編程——基於行爲的機器人實戰指南》

iRobot 公司和 MIT 的著名人物編寫。以深入淺出的方式介紹了機器人的包容架構的特點、原理和應用。一個表現好的、健壯機器人應該是什麼樣的？全書一直在思考這個問題，並試圖給出這個問題的答案。

11.2 如何獲取相關技術資料

智慧型機器人技術是一個比較新的技術領域，出現不過十多年。這方面的書籍、文獻相對較少，大量的機器人出版物、專著都是關於機器人機構學、運動學等工業機器人相關領域的，較少有智慧型機器人製作、設計、專案實踐方面的中文出版物。

但是在很多情況下，並沒有相關的技術資料，怎麼辦？

好在網路時代，已經有幾個相當優秀的中文機器人技術網站，可供大家參考：

- 機器人愛好者網站：http://www.roboticfan.com/
  - 包括新聞資訊、技術文章、博客、論壇、視頻資料，內容相當豐富；
- 機器人天空網站：http://www.robotsky.com/
  - 類似機器人愛好者網站；
- 機器人發燒友網站：http://www.robotdiy.net/
  - 國內最早的機器人製作方面的網站；
- Google 搜索引擎: http://www.google.com
  - 英文的機器人資料和網站遠遠超過中文資料。要獲得更豐富的資料，善用 Google 是個較好的辦法。
- Servo Magazine：http://www.servomagazine.com/
  - Servo Magazine 雜誌是一本美國出版的機器人技術刊物。這本雜誌可讀性非常強，包括技術專欄、問題答疑、新品介紹、測評報告等欄目，也有大量機器人相關零件、軟體的廣告，網站上提供每期雜誌的電子版(PDF 檔)下載。推薦所有讀者閱讀。

第十二章 附錄

12.1 MultiFLEX™2 介面示意圖

12.1.1 MultiFLEX™2-AVR

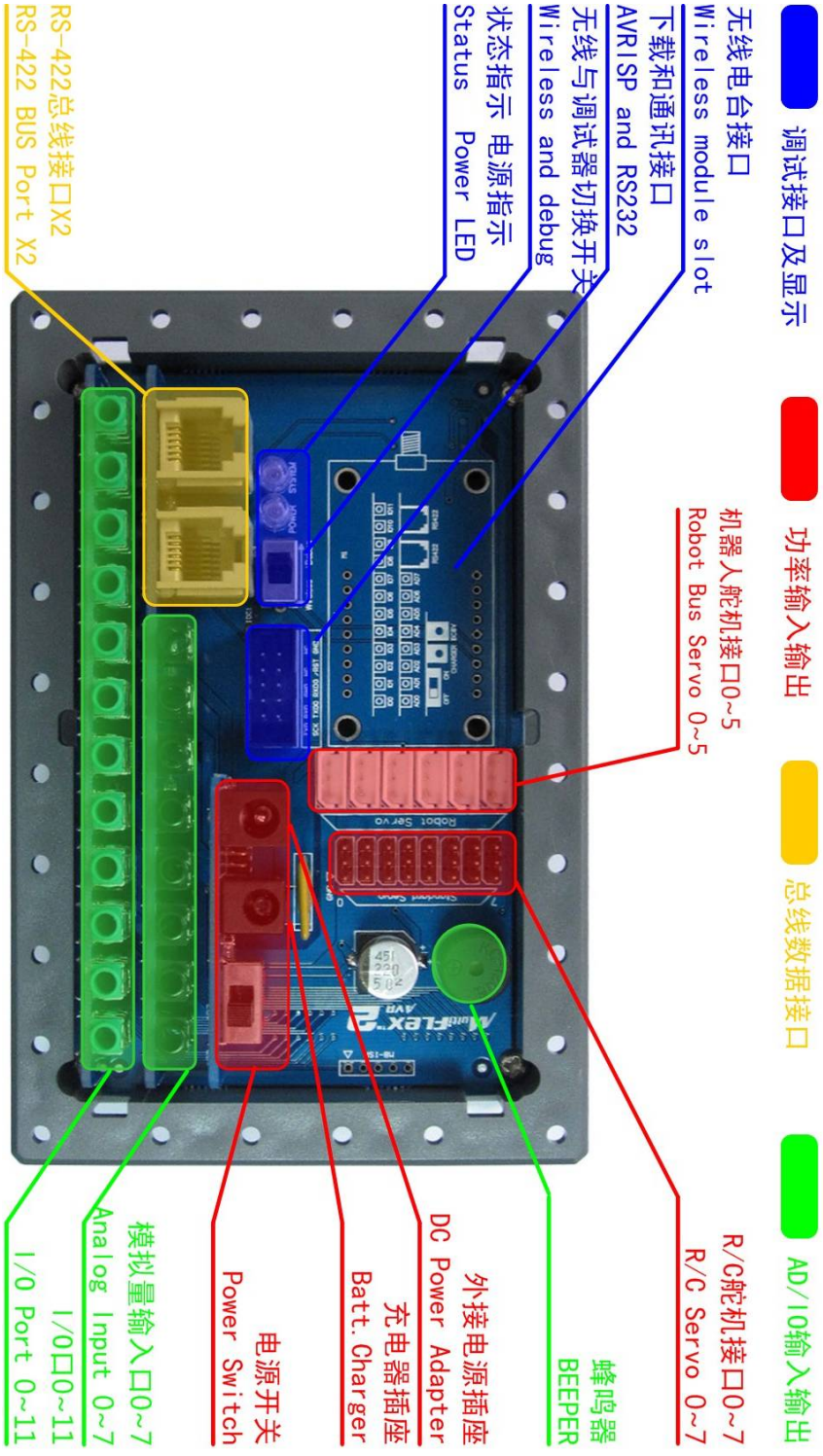


圖 12.1 MultiFLEX™2-AVR 控制器介面示意圖

12.1.2 MultiFLEX™2-PXA270

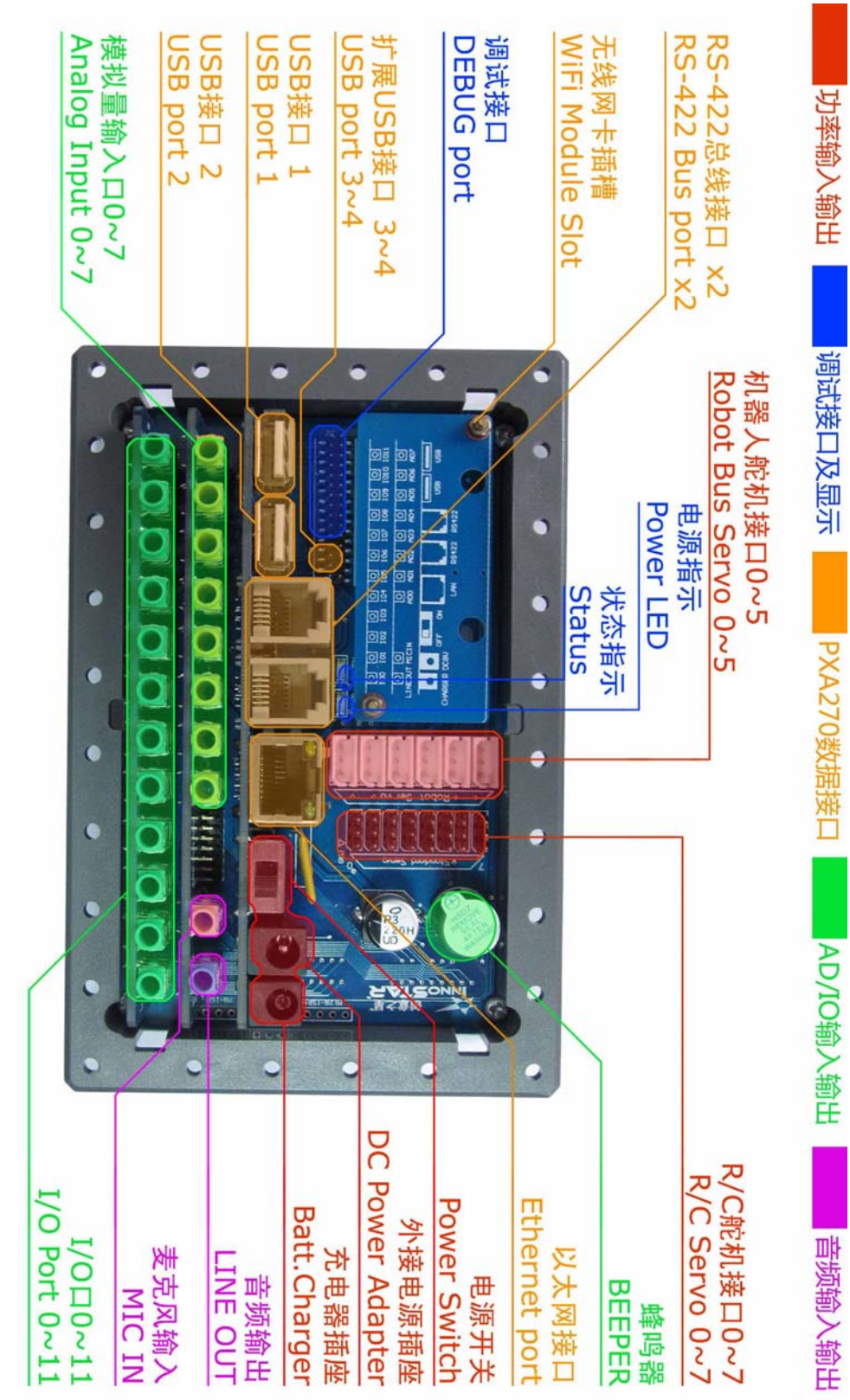


圖 12.2 MultiFLEX™2-PXA270 介面示意圖

12.2 MultiFLEX™2 原理方塊圖

12.2.1 MultiFLEX™2-AVR

如圖 12.3 所示 MultiFLEX™2-AVR 控制器由兩部分組成，上層爲介面模組，下層以 ATmega128 爲核心的 AVR 控制卡。

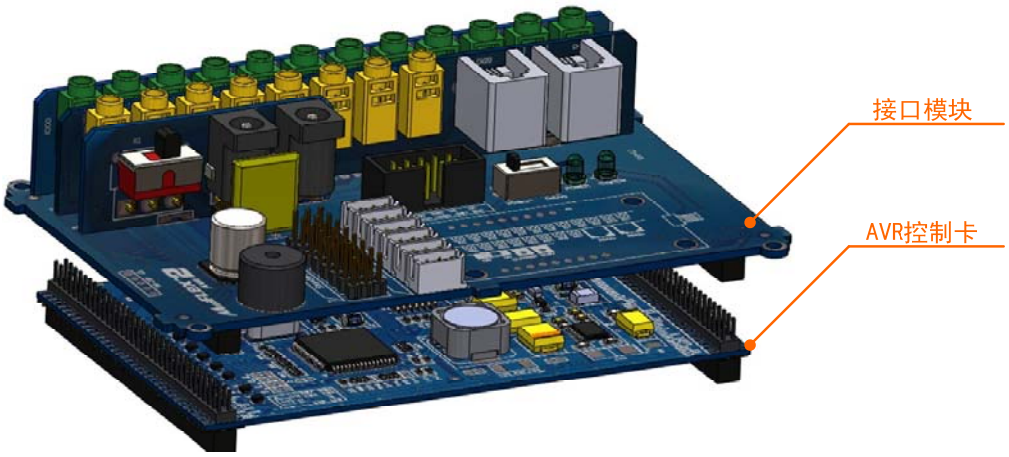


圖 12.3 MultiFLEX™2-AVR 控制結構示意圖

下圖爲 MultiFLEX™2-AVR 控制器 AVR 控制卡部分的架構簡圖：

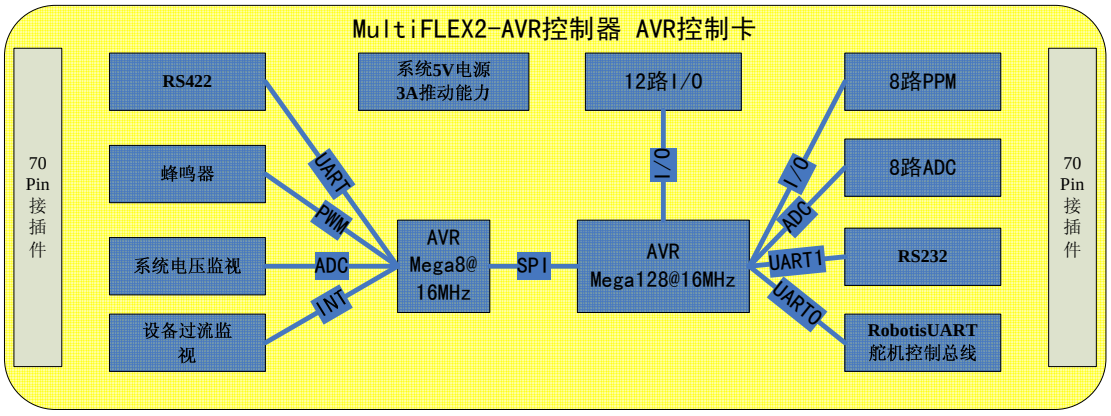


圖 12.4 AVR 控制卡架構簡圖

AVR 控制卡是 MultiFLEX™2-AVR 控制器的核心，內部包括 2 塊 ATMEL 公司生產的 AVR-8bit 單晶片。

主控制器爲 ATmega128，頻率爲 16MHz，其直接控制的功能模組如下：

- 非同步串列介面 UART1 與 MAX202 構成 RS-232 介面，作爲控制器的主通訊介面，用於與控制機連接。

- 非同步串列介面 UART0 與三態門 74HC126 構成半雙工 UART，作爲 RototisUART 伺服馬達控制匯流排介面，用於 CDS 系列機器人伺服馬達的控制。
- 類比數位轉換器 ADC 採用 TL431 提供 2.5V 電壓基準，輸入的類比信號透過 10K 電阻網路分壓、濾波。
- 定時/計數器 TIMER3，產生定時中斷控制 I/O，產生 PPM 信號，用於控制 R/C 伺服馬達。同時 TIMER3 的終端爲系統的 GPIO 採樣、數位濾波和 ADC 採樣提供了固定的節拍。
- 同步串列介面 SPI 與 74HC126 構成主處理器與協控制器 Atmega8 的通訊匯流排。

協控制器爲 ATmega8，頻率爲 16MHz，其直接控制的功能模組如下：

- 非同步串列介面 UART0 與 MAX491 構成 RS-422 匯流排介面，用於連接匯流排設備或進行系統級連接。
- 定時/計數器 TIMER1 工作在 PWM 控制器模式，產生音頻脈衝推動蜂鳴器。
- 類比數位轉換器 ADC，定時採集系統電壓，在系統過壓、欠壓時警報。
- 外部中斷 INT1 與 MIC2505 電源管理晶片構成設備電源過流保護器，在感測器電源總電流超過 2A 時切斷設備供電。

下圖爲 MultiFLEX™2-AVR 控制器的電路結構方塊圖：

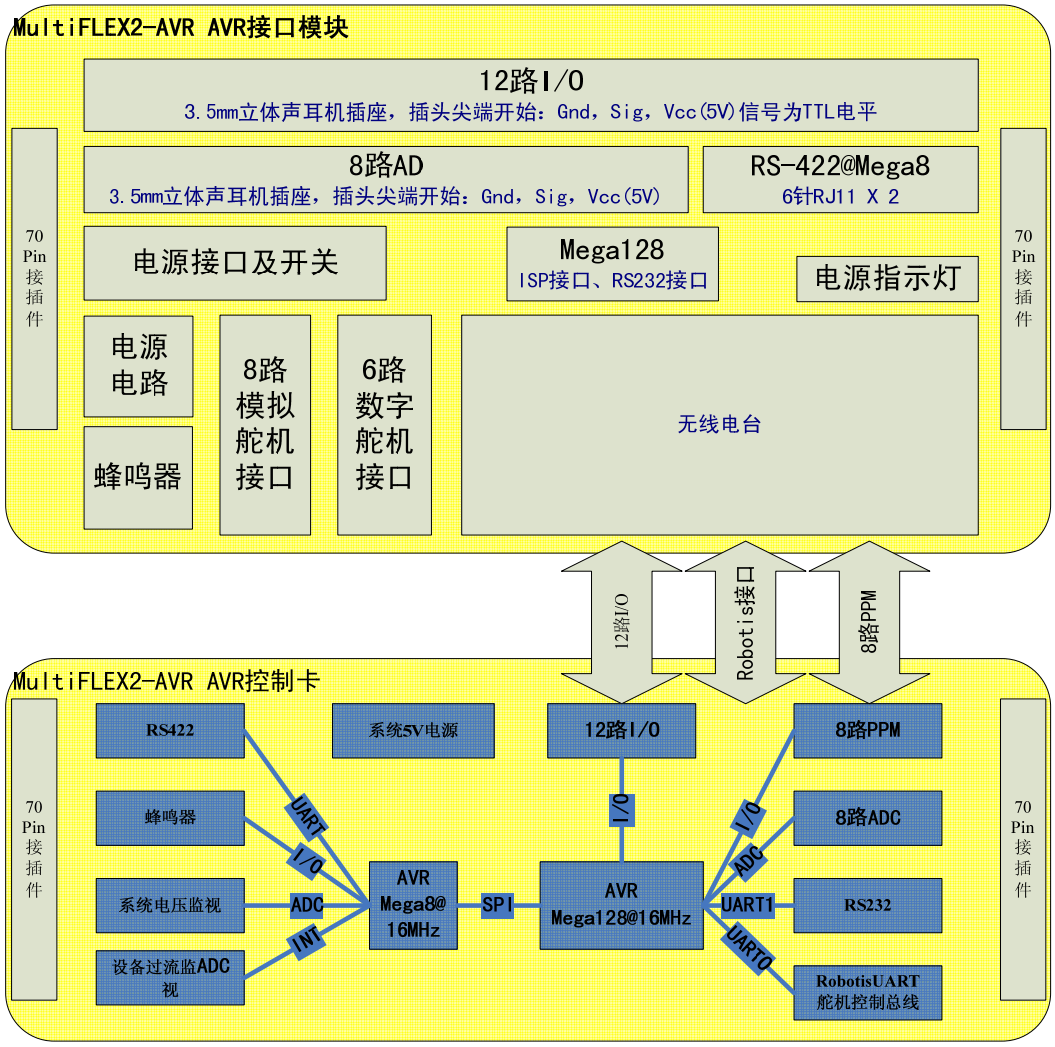


圖 12.5 MultiFLEX™2-AVR 控制器的電路結構方塊圖

12.2.2 MultiFLEX™2-PXA270

MultiFLEX™2-PXA270 控制器由三部分組成，上層爲“MultiFLEX™2-PXA270 介面模組”，中層爲 AVR 控制卡，下層爲 PXA270 控制卡。

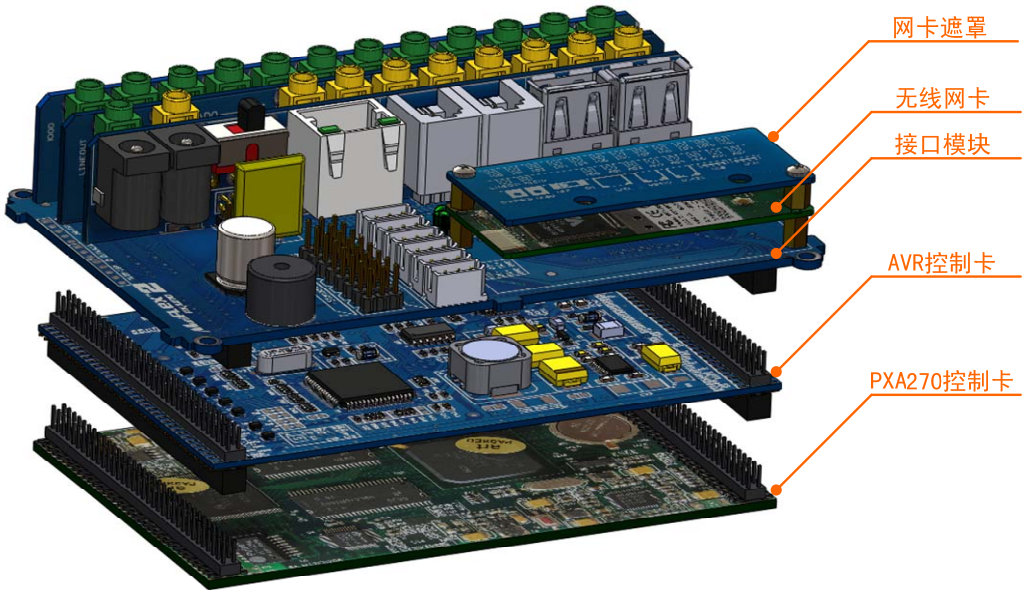


圖 12.6 MultiFLEX™2-PXA270 結構示意圖

下圖爲 MultiFLEX™2-PXA270 控制器的 PXA270 控制卡架構簡圖：

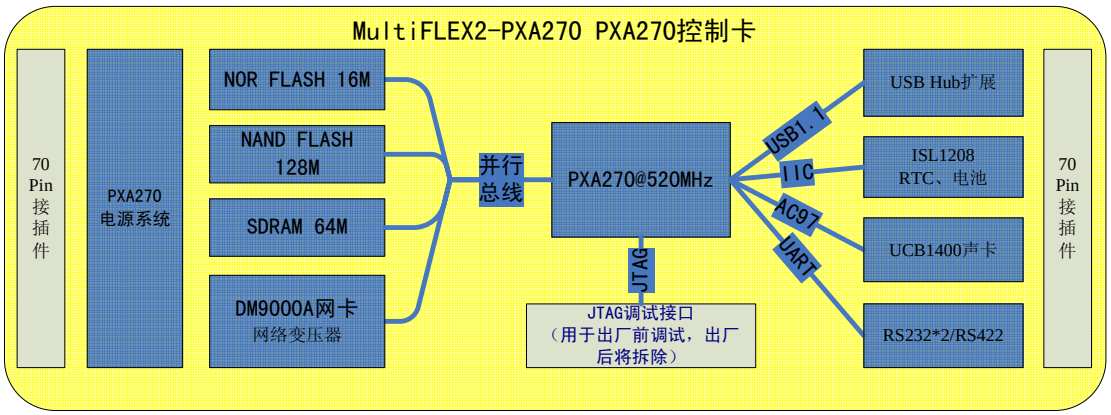


圖 12.7 PXA270 控制卡架構簡圖

PXA270 控制卡是 MultiFLEX™2-PXA270 控制器的核心，擔負了“大腦”這樣的角色，是微型化的電腦，可以進行對視覺、語音的處理，並可以直接接入乙太網。

處理器爲 PXA270-533M，主頻率爲 533MHz，PXA270 控制板的結構模組如下：

- 16M NOR-FLASH 記憶體，用於儲存引導程式和 Linux 內核程式。
- 128M NAND-FLASH 記憶體，用於儲存使用者檔案或程式。
- 64M SD-RAM 記憶體，系統的記憶體。
- 10/100M 乙太網路卡 DM9000A，可以相容直連線與交叉線。
- USB-HUB AU9254，擴充出 4 個 USB1.1 標準的 USB 介面，用於連接攝影機、藍牙模組、無線網卡等設備。
- AC97 音效卡 UCB1400，用於語音輸入與音效播放。
- 1 個調試 RS-232 介面，用於內核調試，不建議初級使用者使用。
- 1 個 RS-232 介面，用於與 AVR 控制卡連接，傳遞控制資訊。
- 1 個 RS-422 介面，用於連接匯流排設備。
- RTC 時鐘 ISL1208，爲系統提供即時日期、時間。當前尚未啓用這項功能。

MultiFLEX™2-PXA270 控制器的 AVR 控制卡與 MultiFLEX™2-AVR 控制器的 AVR 控制卡是完全相同的，它擔負了“小腦”這樣的角色，主要用於對機器人的運動、感覺進行協調控制，使作爲“大腦”的 PXA270 控制卡進行底層控制時更加方便。AVR 控制卡的結構圖見 12.2.1。

下圖爲 MultiFLEX™2-PXA270 控制器的電路結構方塊圖：

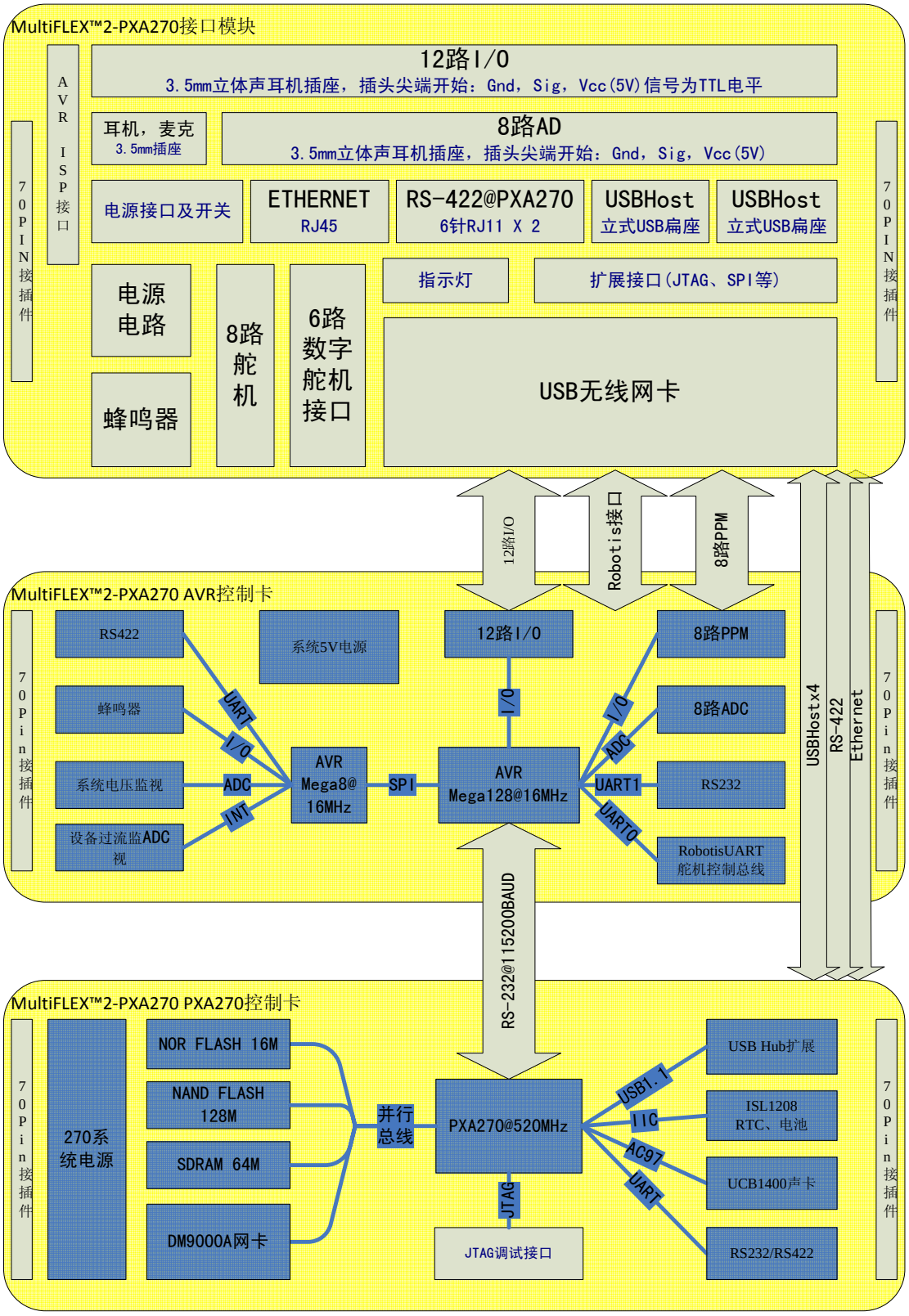


圖 12.8 MultiFLEX™2-PXA270 控制器的電路結構方塊圖

12.3 MultiFLEX™2 開放介面函數一覽

12.3.1 MultiFLEX™2-AVR

MultiFLEX™2-AVR標準編程介面，編譯器爲GCC，編程標準爲GNU99：

1、系統控制函數：

MFInit

功　　能：初始化系統；在主函數的開始部分調用；
函數原型：void MFInit(void)；
輸　　入：無；
輸　　出：無；
示　　例：MFInit(); // 系統初始化

DelayMS

功　　能：毫秒延時函數；
函數原型：void DelayMS(signed long int inMS)；
輸　　入：延遲時間，單位爲毫秒，不得大於32767；
輸　　出：無；
示　　例：DelayMS(1000); // 延遲1000ms

2、GPIO控制函數：

MFSetPortDirect

功　　能：設置GPIO埠的方向，當方向標誌置位元時，對應GPIO通道爲輸出；當方向標誌清零時，對應GPIO通道爲輸入；通道爲輸入時，預設狀態爲高電位。
函數原型：void MFSetPortDirect(unsigned long int inData)；
輸　　入：無符號長整形數的低12位元爲GPIO的方向標誌，依次對應GPIO的0-11通道；
輸　　出：無；
示　　例：MFSetPortDirect(0x00FF); // 設置GPIO的0-7通道爲輸出，8-11通道爲輸入。

MFGetDigiInput

功　　能：讀取GPIO輸入值；
函數原型：signed long int MFGetDigiInput(singed long int inID)；
輸　　入：要獲取的GPIO通道的通道編號；
輸　　出：輸出的長整形值爲邏輯真，則輸入值爲高電位；爲邏輯假，則輸入值爲低電位；
示　　例：signed long int temp_value = 0; // 定義變數temp_value
temp_value = MFGetDigiInput(0); // 讀取0通道的值，儲存在
// temp_value變數中

MFDigiOutput

功　　能：設置GPIO輸出值；
函數原型：void MFDigiOutput(singed long int inID, singed long int inVal)；
輸　　入：inID爲選擇的GPIO輸出埠編號；inVal爲輸出的值，只有邏輯真/邏輯假兩種值；
輸　　出：無；
示　　例：MFDigiOutput (5, TRUE);　　// 設置5通道輸出高電位

3、ADC控制函數：

MFGetAD

功　　能：讀取指定通道的AD轉換值；
函數原型：signed long int MFGetAD(signed long int inID)；
輸　　入：長整形數的低3位元爲ADC的通道編號，0-7有效；
輸　　出：長整形數的低10位元爲讀取到的AD轉換值；
示　　例：signed long int temp_value = 0;　　// 定義變數temp_value
temp_value = MFGetAD (7);　　// 讀取7通道的AD值，並儲存進
// temp_value變數中

4、伺服馬達控制函數：

R/C伺服馬達ID範圍：0xE0-0xE7；對於PPM0-PPM7。

MFSetServoMode

功　　能：設置伺服馬達工作模式，有伺服馬達模式和馬達模式兩種；
函數原型：void MFSetServoMode (signed long int inID, signed long int inMode)；
輸　　入：inID爲伺服馬達ID號，inMode爲伺服馬達模式，模式的關鍵字已經在ServoUnit.h中定義，SERVO_MODE_SERVO爲伺服馬達模式，值爲0；SERVO_MODE_MOTO爲馬達模式，值爲1；
輸　　出：無；
示　　例：MFSetServoMode(1, SERVO_MODE_MOTO); // 將1號伺服馬達設置爲
// 馬達模式
MFDelayMS(200);　　// 延時200ms，使伺服馬達
// 有足夠的時間切換模式

MFSetServoPos

功　　能：設置伺服馬達的目標位置；本方法在伺服馬達處於伺服馬達模式時使用；
函數原型：void MFSetServoPos(signed long int inID, signed long int inPos, signed long int inSpeed)；
輸　　入：inID爲伺服馬達ID號；inPos爲伺服馬達目標位置，取值範圍爲0-1023；inSpeed爲伺服馬達從當前位置運動到指定位置過程中的運動速度，取值範圍爲0-1023；
輸　　出：無
示　　例：MFSetServoPos(224, 512, 100);　　// 讓PPM0伺服馬達以100的速度運
// 動到512位置

MFSetServoRotaSpd

功　　能：設置伺服馬達的運動速度；本方法在伺服馬達處於馬達模式時使用；

函數原型：void MFSetServoRotaSpd(signed long int inID, signed long int inSpeed)；

輸　　入：inID爲伺服馬達的ID號；inSpeed爲伺服馬達的轉動速度，當inSpeed爲正時伺服馬達正傳，當inSpeed爲負時伺服馬達反轉；inSpeed的取值範圍爲-1023-1023；

輸　　出：無；

示　　例：MFSetServoRotaSpd(2, -1000);　　// 讓2號伺服馬達以1000的速度反向轉動

MFGetServoPos

功　　能：讀取指定ID的伺服馬達的位置；

函數原型：signed long int MFGetServoPos(signed long int inID)；

輸　　入：希望讀取位置的伺服馬達的ID號；

輸　　出：長整形數的低10位元指定ID號的伺服馬達的位置值；

示　　例：signed long int temp_value = 0;　　// 定義變數temp_value
temp_value = MFGetServoPos (4);　// 讀取4號伺服馬達的位置，寫入
// temp_value變數中

MFServoAction

功　　能：使伺服馬達執行非同步寫入的命令；

函數原型：void MFServoAction(void)；

輸　　入：無；

輸　　出：無；

示　　例：MFServoAction ();　// 對一組伺服馬達的運動設置完成後，這個函數
// 會讓它們開始運動

12.3.2 MultiFLEX™2-PXA270

MultiFLEX™2-PXA270標準編程介面，編譯器爲G++：

1、系統控制函數：

MFInit

功　　能：初始化系統；在主函數的開始部分調用；

函數原型：void MFInit(void)；

輸　　入：無；

輸　　出：無；

示　　例：MFInit();　// 系統初始化

DelayMS

功　　能：毫秒延時函數；

函數原型：void DelayMS(signed long int inMS)；

輸　　入：延遲時間，單位爲毫秒，不得大於32767；

輸　　出：無；
示　　例：DelayMS(1000);　　// 延遲1000ms

2、GPIO控制函數：

MFSetPortDirect

功　　能：設置GPIO埠的方向，當方向標誌置位元時，對應GPIO通道爲輸出；當方向標誌清零時，對應GPIO通道爲輸入；通道爲輸入時，預設狀態爲高電位。
函數原型：void MFSetPortDirect(unsigned long int inData)；
輸　　入：無符號長整形數的低12位元爲GPIO的方向標誌，依次對應GPIO的0-11通道；
輸　　出：無；
示　　例：MFSetPortDirect(0x00FF);　　// 設置GPIO的0-7通道爲輸出，8-11通道爲輸入。

MFGetDigiInput

功　　能：讀取GPIO輸入值；
函數原型：signed long int MFGetDigiInput(singed long int inID)；
輸　　入：要獲取的GPIO通道的通道編號；
輸　　出：輸出的長整形值爲邏輯真，則輸入值爲高電位；爲邏輯假，則輸入值爲低電位；
示　　例：signed long int temp_value = 0;　　// 定義變數temp_value
temp_value = MFGetDigiInput(0);　　// 讀取0通道的值，儲存在
// temp_value變數中

MFDigiOutput

功　　能：設置GPIO輸出值；
函數原型：void MFDigiOutput(singed long int inID, singed long int inVal)；
輸　　入：inID爲選擇的GPIO輸出埠編號；inVal爲輸出的值，只有邏輯真/邏輯假兩種值；
輸　　出：無；
示　　例：MFDigiOutput (5, TRUE);　　// 設置5通道輸出高電位

3、ADC控制函數：

MFGetAD

功　　能：讀取指定通道的AD轉換值；
函數原型：signed long int MFGetAD(signed long int inID)；
輸　　入：長整形數的低3位爲ADC的通道編號，0-7有效；
輸　　出：長整形數的低10位爲讀取到的AD轉換值；
示　　例：signed long int temp_value = 0;　　// 定義變數temp_value
temp_value = MFGetAD (7);　　// 讀取7通道的AD值，並儲存進
// temp_value變數中

4、伺服馬達控制函數：

R/C伺服馬達ID範圍：0xE0-0xE7；對於PPM0-PPM7。

MFSetServoMode

功　　能：設置伺服馬達工作模式，有伺服馬達模式和馬達模式兩種；

函數原型：void MFSetServoMode (signed long int inID, signed long int inMode)；

輸　　入：inID爲伺服馬達ID號，inMode爲伺服馬達模式，模式的關鍵字已經在ServoUnit.h中定義，SERVO_MODE_SERVO爲伺服馬達模式，值爲0；SERVO_MODE_MOTO爲馬達模式，值爲1；

輸　　出：無；

示　　例：
```
MFSetServoMode(1, SERVO_MODE_MOTO); // 將1號伺服馬達設置爲
                                    // 馬達模式
MFDelayMS(200);                     // 延時200ms，使伺服馬達
                                    // 有足夠的時間切換模式
```

MFSetServoPos

功　　能：設置伺服馬達的目標位置；本方法在伺服馬達處於伺服馬達模式時使用；

函數原型：void MFSetServoPos(signed long int inID, signed long int inPos, signed long int inSpeed)；

輸　　入：inID爲伺服馬達ID號；inPos爲伺服馬達目標位置，取值範圍爲0-1023；inSpeed爲伺服馬達從當前位置運動到指定位置過程中的運動速度，取值範圍爲0-1023；

輸　　出：無

示　　例：
```
MFSetServoPos(224, 512, 100);   // 讓PPM0伺服馬達以100的速度運
                                // 動到512位置
```

MFSetServoRotaSpd

功　　能：設置伺服馬達的運動速度；本方法在伺服馬達處於馬達模式時使用；

函數原型：void MFSetServoRotaSpd(signed long int inID, signed long int inSpeed)；

輸　　入：inID爲伺服馬達的ID號；inSpeed爲伺服馬達的轉動速度，當inSpeed爲正時伺服馬達正傳，當inSpeed爲負時伺服馬達反轉；inSpeed的取值範圍爲-1023-1023；

輸　　出：無；

示　　例：
```
MFSetServoRotaSpd(2, -1000);    // 讓2號伺服馬達以1000的速度反
                                // 向轉動
```

MFGetServoPos

功　　能：讀取指定ID的伺服馬達的位置；

函數原型：signed long int MFGetServoPos(signed long int inID)；

輸　　入：希望讀取位置的伺服馬達的ID號；

輸　　出：長整形數的低10位元指定ID號的伺服馬達的位置值；

示　　例：
```
signed long int temp_value = 0;     // 定義變數temp_value
temp_value = MFGetServoPos (4);     // 讀取4號伺服馬達的位置，寫入
                                    // temp_value變數中
```

MFServoAction

功　　能：使伺服馬達執行非同步寫入的命令；

函數原型：void MFServoAction(void)；

輸　　入：無；

輸　　出：無；

示　　例：MFServoAction ();　// 對一組伺服馬達的運動設置完成後，這個函數
// 會讓它們開始運動

5、LCD顯示函數

MF2X4LCD

功　　能：在RS422匯流排設備，InnoStarDisplay上顯示資訊；

函數原型：void MF2X4LCD(const char*item,int insize)；

輸　　入：item爲儲存待顯示的字串的陣列首位址指標；insize爲儲存待顯示的字串的陣列長度，注意，字串的陣列長度爲字元個數加1。

輸　　出：無；

示　　例：MF2X4LCD(“華亨科技”, 9); // “華亨科技”爲4個全形字元，佔用8個位
// 元組，所以字串的陣列長度爲9

6、視覺函數

MFCapOpen

功　　能：開啓攝影機視頻捕捉；

函數原型：bool MFCapOpen()；

輸　　入：無；

輸　　出：開啓操作是否成功，成功爲true，失敗爲false；

示　　例：　MFCapOpen();　　// 打開攝影機

MFCapSetH

功　　能：設置色調閾值；

函數原型：void MFCapSetH(int inHmax,int inHmin)；

輸　　入：inHmax色調閾值的上限值，inHmin色調閾值的下限值；
輸入值範圍爲0-255；

輸　　出：無；

示　　例：MFCapSetH(150, 100);　// 設置色調閾值爲100-150

MFCapSetS

功　　能：設置飽和度閾值；

函數原型：void MFCapSetS(int inSmax,int inSmin)；

輸　　入：inSmax色飽和度閾值的上限值，inSmin色飽和度閾值的下限值；
輸入值範圍爲0-255；

輸　　出：無；

示　　例：MFCapSetS(120, 50);　　// 設置飽和度閾值爲50-120

MFCapSetI

功　　能：設置亮度閾值；
函數原型：void MFCapSetI(int inImax,int inImin)；
輸　　入：inImax亮度閾值的上限值，inImin亮度閾值的下限值；
　　輸入值範圍爲0-255；
輸　　出：無；
示　　例：MFCapSetI(70, 20);　　// 設置亮度閾值爲20-70

MFCapGetCenterX

功　　能：獲取目標區域中心x座標；
函數原型：int MFCapGetCenterX()；
輸　　入：無；
輸　　出：目標區域中心x座標；輸出值範圍爲0-320；
示　　例：int center_x = 0;　　// 定義x座標變數
　　center_x = MFCapGetCenterX();　// center_x變數中儲存了目標區域中
　　// 心x座標

MFCapGetCenterY

功　　能：獲取目標區域中心y座標；
函數原型：int MFCapGetCenterY()；
輸　　入：無；
輸　　出：目標區域中心y座標；輸出值範圍爲0-240；
示　　例：int center_y = 0;　　// 定義y座標變數
　　center_y = MFCapGetCenterY();　// center_y變數中儲存了目標區域
　　// 中心y座標

7、語音識別函數

MFSRStart

功　　能：啓動語音引擎；
函數原型：void MFSRStart()；
輸　　入：無；
輸　　出：無；
示　　例：MFSRStart();　　// 啓動語音引擎

MFSRInsertText

功　　能：插入語音識別條目；
函數原型：bool MFSRInsertText(int index,const char*item,int insize)；
輸　　入：index爲所插入詞條的索引號，不同詞條的索引號必須唯一，否則後插入的詞條將會覆蓋之前插入的相同索引號詞條；item爲儲存著待插入詞條的陣列首位址指標；insize爲儲存著待插入詞條的陣列長度
輸　　出：詞條插入是否成功，成功爲true，識別爲false；
示　　例：bool temp_state = false;　　// 定義狀態變數
　　temp_state = MFSRInsertText(0, “華亨科技”, 9);　// 語音識別條目代號
　　// 爲0，內容爲“華亨
　　// 科技”，詞條的陣

// 列長度爲9

MFSRGetResIndex

功　　能：獲取語音識別結果；
函數原型：int MFSRGetResIndex()；
輸　　入：無；
輸　　出：識別詞條的索引號，與插入詞條時提供的索引號一致；
示　　例：int temp_value = 0;　　// 定義變數temp_value
　　temp_value = MFSRGetResIndex();　　// 將匹配出的詞條編號儲存在
　　// 變數中

8、MP3播放函數

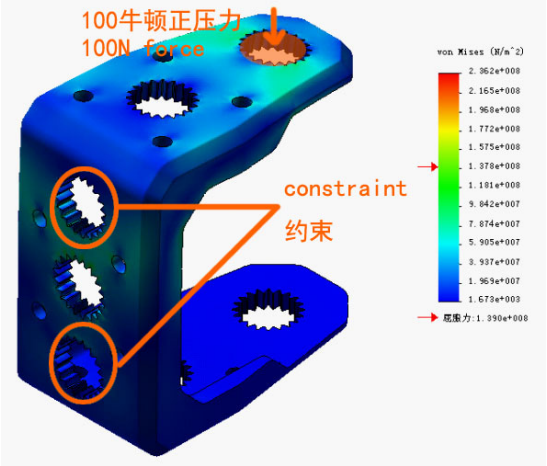
MFMp3Play

功　　能：播放MP3檔；
函數原型：void MFMp3Play(const char*item,int insize)；
輸　　入：item爲儲存mp3檔案名的陣列首位址；insize爲儲存mp3檔案名的陣列長度
輸　　出：無；
示　　例：MFMp3Play(“夕陽蕭鼓.mp3”, 13);　　// 播放名爲“夕陽蕭鼓.mp3”
　　// 的MP3檔，mp3檔案名的字
　　// 元長度爲 13

12.4 常見問題解答

12.4.1 結構相關

- “創意之星”機器人套件是否提供 3D 模型檔？
  - 是的，我們提供所有結構和連接零件的 3D 模型，檔案格式爲 IGES 或者 STEP。
- 結構零件的材料是什麼？
  - 所有深灰色、淺灰色、白色、桔黃色結構零件都使用增強型專案塑料（ABS）製成。
- 各個結構零件是否可以任意角度連接？
  - 不能。連接的相對角度只能以 18 度增量。例如 18 度，36 度，90 度，等等。
- 如何組裝這些零件？
  - 組裝之前您最好先對自己要做的結構有一個清晰的構想圖。建議畫出草圖。或者使用我們提供的 3D 模型檔，在電腦中搭建虛擬圖片。IGES 格式的檔能夠被大多數 3D CAD 軟體讀取。
- 結構零件的強度如何？
  - 結構零件均考慮了使用中的強度。您可以使用光碟中提供的 3D 模型，利用 ANSYS 或者 UGS、SolidWorks 等軟體中提供的 CosmosWorks 有限元素工具進行分析。參考下圖：



- 各個連接零件連接的強度如何？
  - 各個連接零件均採用專利的 ConnFLEX 技術連接。通常一個連接包括 2 個結構零件、2 個連接零件，一個螺絲和一個螺母。單個連接的抗扭轉強度爲：失效彎矩 25Nm。
  - 如果需要更高的連接強度，可以採用多點連接。典型的雙點連接（連接件使用 LM3 和 LX3）的失效彎矩可達 80Nm。在這麼大的彎矩下，結構零件會變形、斷裂，而不是連接元件。

12.4.2 控制器相關

- MultiFLEX™2-AVR 控制器和 MultiFLEX™2-PXA270 控制器兩種控制卡有何區別？我應該選擇哪一種？
  - MultiFLEX™2-AVR 控制器採用 AVR 單晶片作爲處理器，其規格爲 8 位元 RISC CPU，主頻率 16MHz，採用 AVR 指令集。MultiFLEX™2-PXA270 控制器採用 Marvell PXA270 作爲處理器，其規格爲 32 位元 Xscale 架構 CPU（大致相當於 ARM10 架構），主頻率爲 533MHz，採用增強的 ARM 指令集。
  - MultiFLEX™2-AVR 控制器價格較低，功耗低（<1W），主頻率低，無法勝任圖像、語音、高強度計算等要求。但是 MultiFLEX™2-AVR 控制器沒有採用作業系統，由 NorthSTAR 編寫的程式編譯爲可執行程式後，直接透過 ISP（In-System Programming）下載到控制器內執行，其即時性高，時序嚴格可控，並可以嵌入彙編，對於馬達控制、PWM 輸出及其他高即時性要求的任務比較理想。
  - MultiFLEX™2-PXA270 控制器價格較高，主頻高，功耗稍高（約 2W），PXA270 處理器上執行 Linux 作業系統，負責處理網路通訊、圖像、語音、USB 設備等高運算量的任務，NorthSTAR 編譯的程式是執行在 Linux 系統中。由於非即時的 Linux 作業系統的存在，MultiFLEX™2-PXA270 控制器的性能強大很多，但是即時性較 MultiFLEX™2-AVR 控制器要差。
- 如何開發基於 MultiFLEX™2 控制器的程式？
  - 大致有兩種使用方式：不熟悉 C 語言的用戶，可以使用 NorthSTAR，利用流程圖編程，並在 NorthSTAR 環境中調試、編譯、下載；熟悉 C 語言和相關處理器開發的使用者，可以直接使用 Source Insight、ICCAVR、GCC、ADS（Xscale）等編輯器、編譯器和開發工具直接進行開發。
- MultiFLEX2 控制器是否提供原始碼？
  - 是的，我們提供大部分原始碼，一些商業的函式庫檔案除外（非本公司開發的）。

這些代碼採用 GPL 授權，請您在使用的時候嚴格遵守 GPL 的規定。

- 我不小心刷掉了控制器中的韌體，怎麼辦？
  - MultiFLEX™2-AVR 控制器中沒有不可以修改的韌體，您每次下載 NorthSTAR 的程式都會更新 MultiFLEX™2-AVR 控制器 Atmega128 的韌體。
  - 對於 MultiFLEX™2-PXA270 控制器，控制器的韌體包括檔案系統、Blob、Linux 內核和一些驅動程式。如果開發過程中這些東西遺失或損壞，請聯繫博創科技售後服務支持。
- NorthSTAR 軟體提供的圖像識別控制項如何使用？
  - MultiFLEX™2-PXA270 控制器才能支援該功能。
  - 該控制項是一個簡單的圖像識別元件，使用它可以獲取 USB 攝影機的圖像中的特定顏色區域。顏色區域和閾值可以由使用者設定。顏色區域的中心座標可以返回到一組變數中。您可以使用這個功能來實現簡單的圖像識別，例如，利用返回的座標值用於跟蹤一個紅色乒乓球，或者控制一個小車沿著黃顏色的軌跡走，而忽略綠色的軌跡。
  - 我們將持續推出更多種類的圖像識別控制項，例如形狀識別、邊沿檢測等。
  - 您如果熟悉 Linux 嵌入式開發，也可以自己編寫圖像識別的代碼。
- NorthSTAR 軟體提供的語音識別控制項如何使用？
  - MultiFLEX™2-PXA270 控制器才能支援該功能。
  - 該控制項中可以輸入任意中文語句，並根據輸入的文字來識別對應的普通話語音。
  - 例如，輸入“您好”，控制器程式執行時，將能夠識別“您好”的普通話發音，並返回一個值。使用者可以把這個值保存到變數中，用於程式控制的過程。
  - 本功能爲非特定語音識別。使用前不需要訓練；因此識別率有一定限制，在安靜環境下可以達到 90%以上。
- R/C 伺服馬達的 ID 是什麼意思？
  - 使用 NorthSTAR 編程時，我們可以像控制 CDS5500 一樣控制 8 個 R/C 伺服馬達介面，只不過沒有馬達模式而已。這時同樣需要 ID 號來辨別您控制的是哪個伺服馬達，CDS5500 的 ID 號是儲存在伺服馬達內部的，R/C 伺服馬達不能儲存外部資料，所以對應的 ID 號儲存在控制器上，PPM0 對應 E0（224）、PPM1 對應 E1（225）、PPM2 對應 E2（226）、PPM3 對應 E3（227）、PPM4 對應 E4（228）、PPM5 對應 E5（229）、PPM6 對應 E6（230）、PPM7 對應 E7（231）。

12.4.3 CDS5500 伺服馬達相關

- 爲什麼 CDS5500 伺服馬達的舵盤（Horn）似乎比較容易損壞？
  - 作爲執行器的輸出軸與結構零件連接的關鍵零件，舵盤被設計爲一個相對薄弱的環節，它可以承受的最高扭矩爲 25Kg.cm，這一數字已經相當於 CDS5500 伺服馬達最大輸出扭矩的 150%左右。舵盤損壞通常意味著您的伺服馬達已經嚴重超載，舵盤損壞可以保護伺服馬達，避免伺服馬達損壞。
- CDS5500 能夠返回位置、速度、負載等，這些資訊有什麼用？
  - 一個關節型機器人的各個關節的 CDS5500 伺服馬達能夠返回位置，則可以用於機器人示教。例如：
  - 用手把機器人各關節擺放到需要的角度和位置，打開 NorthSTAR 軟體，即可記錄機器人的當前姿態；
  - 把機器人各關節扳動到下一個位置，再在 NorthSTAR 軟體中記錄一個姿態；

- 重複以上動作，並使用 NorthSTAR 軟體播放這些記錄的姿態。
- 機器人將會順序地執行剛才記錄的各個姿態。
- 除此之外，這些回饋功能還能用於對關節進行限位、限制輸出扭矩、以保護伺服馬達不受損傷，避免意外操作導致危險。

- CDS5500 伺服馬達的最大電流是多少？如何安全使用？
  - 正常工作時，工作電流約 400~700mA；
  - 啓動和堵轉時電流會達到 2000~2500mA。因此堵轉時間超過 4 秒鐘時，馬達將會自動保護，停止輸出扭矩。
  - 雖然有保護功能，但如果短時間內多次短暫堵轉，保護功能可能無法生效，很可能會因爲過熱損壞伺服馬達。
  - CDS5500 配套的舵盤被設計爲在超載 50%的情況下會損壞。如果舵盤經常性地損壞，說明負載過大。請注意設計是否不合理，相關伺服馬達是否承受了比額定值更大的負載。另外，安裝舵盤時角度不正確也會造成舵盤容易損壞。
  - 儘量不要用手快速猛烈地轉動伺服馬達輸出軸。轉動過程應該平穩、輕柔。突然猛扳伺服馬達輸出軸會造成齒輪損壞。