

高雄縣高英高級工商職業學校
Kao Ying Industrial Commercial Vocational High School

專題製作報告



微電腦自動遊戲機

陳大元

中華民國 105 年 01 月

微電腦自動遊戲機

摘 要

隨著科技的快速發展，時間的流逝，自從觀太陽、擺鐘到現在的高科技，人類不斷研究，不斷創新紀錄。我將整合電子、電機、機械等方面的知識，製作一組完整的遊戲機系統，包含了感測電路、馬達電路、CPU電路、蜂鳴器等各電路部分，配合電源、模型、直流馬達等硬體以及作為可自動感應之邏輯判斷的程式設計軟體，因此本專題的製作可說是涉及了多方面的知識領域。

其中的核心-CPU，選擇了較為熟悉的8051單晶片，因為它的硬體架構及周邊設備完整、指令集功能強大、程式可複寫功能等種種優勢，正符合設計的需求，而利用程式的模組化，副程式的應用，使程式如堆積木般的組合起來，更容易閱讀及進行修改。

遊戲機的動作原理是藉由無段開關的偵測是否按下按鍵，再經由 8051 單晶片，由組合語言程式進行邏輯判斷後，對直流馬達電路下指令做轉動動作。

而在這次的專題便朝人類方便性的方向而去研究，在章節架構編排上，除了摘要、目錄、圖表目錄外，介紹了做此次專題的動機及目的，再進行 8051 的相關資料探討，呈現出此次專題實習測驗的一切過程，並展現成果。

關鍵詞：單晶片、組合語言

目 錄

摘要	I
目錄	II
表目錄	III
圖目錄	IV
壹、前言	1
一、研究動機	1
二、研究目的	1
三、製作架構	2
四、研究方法	4
貳、理論探討	5
參、專題製作	16
一、設備及器材	16
二、製作方法與步驟	17
三、專題製作	19
肆、製作成果	23
參考文獻	25
附錄一 微電腦自動遊戲機之程式碼	26
附錄二 單晶片 89C51 特性介紹	42

表目錄

表 2-1-1 MCS-51 族微處理器的晶片種.....	5
表 2-1-2 MCS-51 各接腳用途.....	6
表 3-1-1 專題製作使用儀器（軟體）設備一覽表	16
表 3-3-2 專題製作計畫書	18
表 3-3-3 自動遊戲機之材料表	22

圖目錄

圖 1-3-1 專題製作流程圖	3
圖 2-1-1 MCS-51 接腳圖	7
圖 2-1-2 MCS-51 記憶體結構圖	8
圖 2-2-3 單晶片 8051 的接腳圖	10
圖 3-2-1 製作方法與步驟	12
圖 3-3-1 8051 主電路	19
圖 3-3-2 馬達控制電路	19
圖 3-3-3 電源電路	20
圖 3-3-4 蜂鳴器電路	20
圖 3-3-5 LED 電路.....	21
圖 3-3-6 按鍵電路	21
圖 4-1-1 遊戲機電路板製作過程(一).....	23
圖 4-1-2 遊戲機電路板製作過程(二).....	23
圖 4-1-3 遊戲機電路板製作過程(三).....	23
圖 4-1-4 遊戲機電路板製作過程(四).....	23
圖 4-1-5 微電腦自動遊戲機成品	24
圖 4-1-6 測量按鈕(一).....	24
圖 4-1-7 測量音量(二).....	24
圖 4-1-8 外觀檢測(三).....	24

壹、前言

一、研究動機

讓現代的孩子瞭解遊戲，不僅僅只是在智慧型手機或平板電腦上，生活週遭也有相當多的樂趣，如早期的遊戲機，增進了朋友與家人之間的感情，不要只是成天低著頭滑著手機，而缺少了人與人之間應有的實際互動關係。

透過全新的方式，重新呈現當初在大家眼中平凡無奇的玩具，以電子產業的方式，讓它重新活潑生動的出現在眾人眼前，用以拉近人與人之間的互動與交流。

之所以會選擇做出這個專題的主要原因是因為學校裡所教授的知識，都是希望學生未來可以用來解決日常生活中所遇到的問題，正因如此希望學生可以動動腦發揮創意，同時也希望學生可以從閱讀中蒐集、整理與資料分析，並依循線索，解決問題的能力。藉由學校之專業實習課程所學，學習電路製作與程式輸寫透過微電腦自動遊戲機的製作過程，並去了解 89C51 單晶片是如何將電路及程式相結合去進行運作。

二、研究目的

大多數人玩的玩具中，夾手機遊戲是一個大家都覺得有趣且刺激的，所以希望利用微電腦控制來製作遊戲機，加上聲光效果，來增添生活上的樂趣。

利用單晶片 89C51 和組合語言程式，來製造微電腦自動遊戲機。

三、製作架構

(一)專題製作流程

1. 畫出電路圖與 Layout 電路圖
2. 利用麵包板、電源供應器、電表，測試電路是否能運作
3. 熟悉 89C51 的使用方法及撰寫程式
4. 熟悉 89C51 燒錄方法
5. 將電路移到一般的電路板上，測試是否正常運作
6. 聲光功能是否正常
7. 最後校正及外部包裝設計

(二)專題製作流程圖

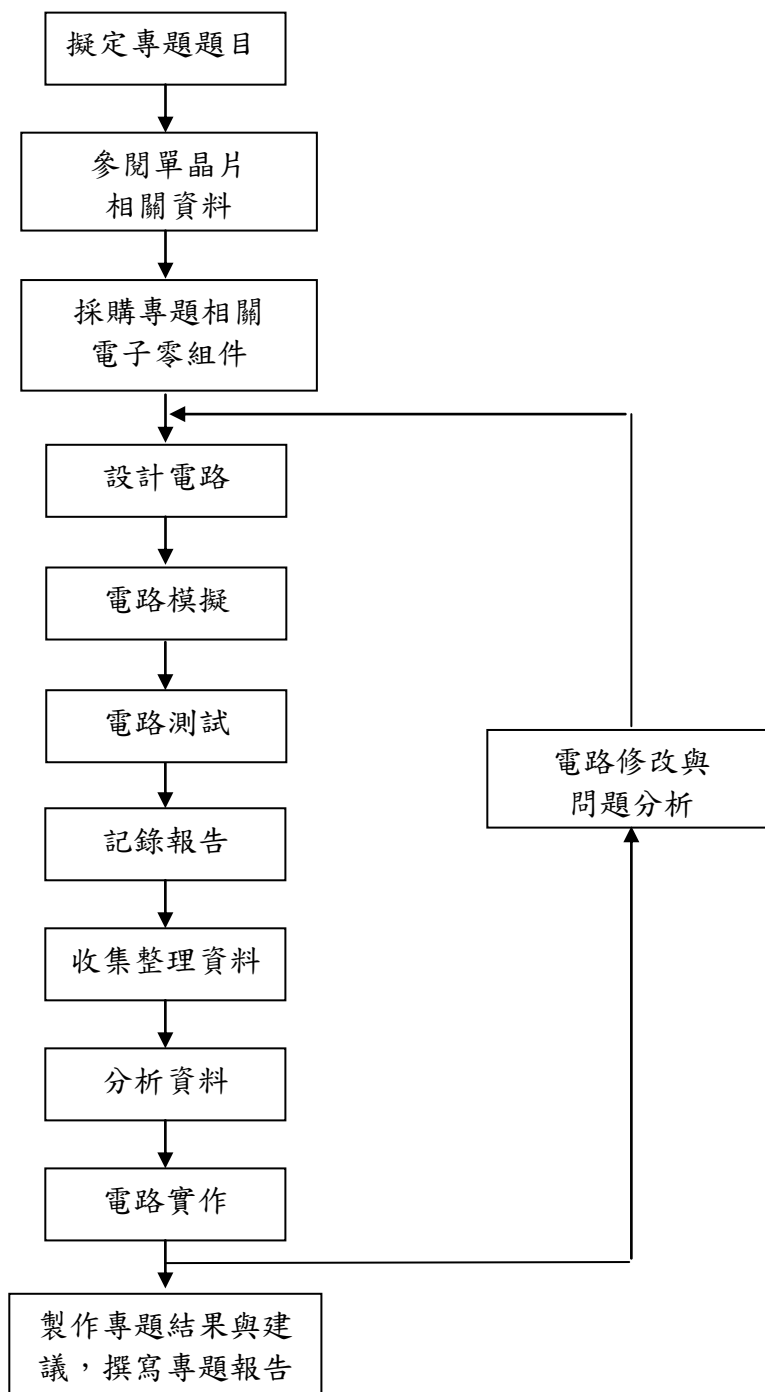


圖 1-3-1 專題製作流程圖

四、研究方法

- (一) 先利用麵包板、電源供應器、電表，測試馬達電路是否能運作。
- (二) 熟悉 89C51 的使用方法、組合語言使用方法，熟悉燒錄方法。
- (三) 程式撰寫。
- (四) 將程式燒入 89C51，測試是否能依據設計正常運作。
- (五) 將電路移植到一般的電路板上，並測試是否正確。
- (六) 聲光功能是否正常
- (七) 最後校正及外部包裝設計

貳、理論探討

本章將綜覽電子實習及單晶片相關的理論與實務研究，共分為二節來進行相關的理論分析及探討。第一節介紹電子相關零組件的理論與原理；第二節說明單晶片的內部架構、特性、理論基礎及功能，以及組合語言程式設計原則。

一、8051 背景知識

(1) 單晶片 8051 介紹

所謂單晶片微電腦 (Single Chip Microcomputer)，簡單的說就是：本身除了具備中央處理單元的功能外，將記憶體單元、輸入/輸出單元組合在同一顆晶片內，只須一些支援電路就能獨立作業。

(2) MCS-51 單晶片特性介紹

MCS-51 為八位元微電腦，內有4K 位元組 (Byte) 唯讀記憶體 (ROM) 及128 位元組 (Byte) 隨機存取記憶體 (RAM)，外部RAM 和ROM 則可個別擴充到64K 位元組，提供5 個中斷源並可執行雙層中斷優先規劃功能，四組並列式平行I/O 埠 (32 位元)，一個全雙工串列埠，並能位元定址及執行布林 (Boolean) 運算。

表 2-1-1 MCS-51 族微處理器的晶片種

名稱	EEPROM	EPROM	ROM(位元組)	RAM(位元組)	16 位元計時器	電路型式
8051		(8751)	4K	128	2	HMOS
8051AH		8751H	4K	128	2	HMOS
8052AH		8751BH	8K	256	3	HMOS
80C51BH		87C51	2K	128	2	CMOS
89C51	89C51		4K	128	2	CMOS
89C52	89C52		8K	256	3	CMOS

(3) MCS-51 接腳說明

表2-1-2 MCS-51 各接腳用途

pin	代號	用途
40	Vcc +	5V 電源供應
20	Vss	接地
39-32	P0. 0-P0. 7	I/O Port 0 or A/D Bus
1-8	P1. 0-P1. 7	I/O Port 1
21-28	P2. 0-P2. 7	I/O Port 2 or A/D Bus
10-17	P3. 0-P3. 7	I/O Port 3
9	RST	Reset
18 , 19	XTAL1, XTAL2	石英震盪器輸入
29	PSEN	Program Store Enable
30	ALE/PROG	Address Latch Enable
31	EA/VDD External	Access Enable

(4) 8051 記憶體結構

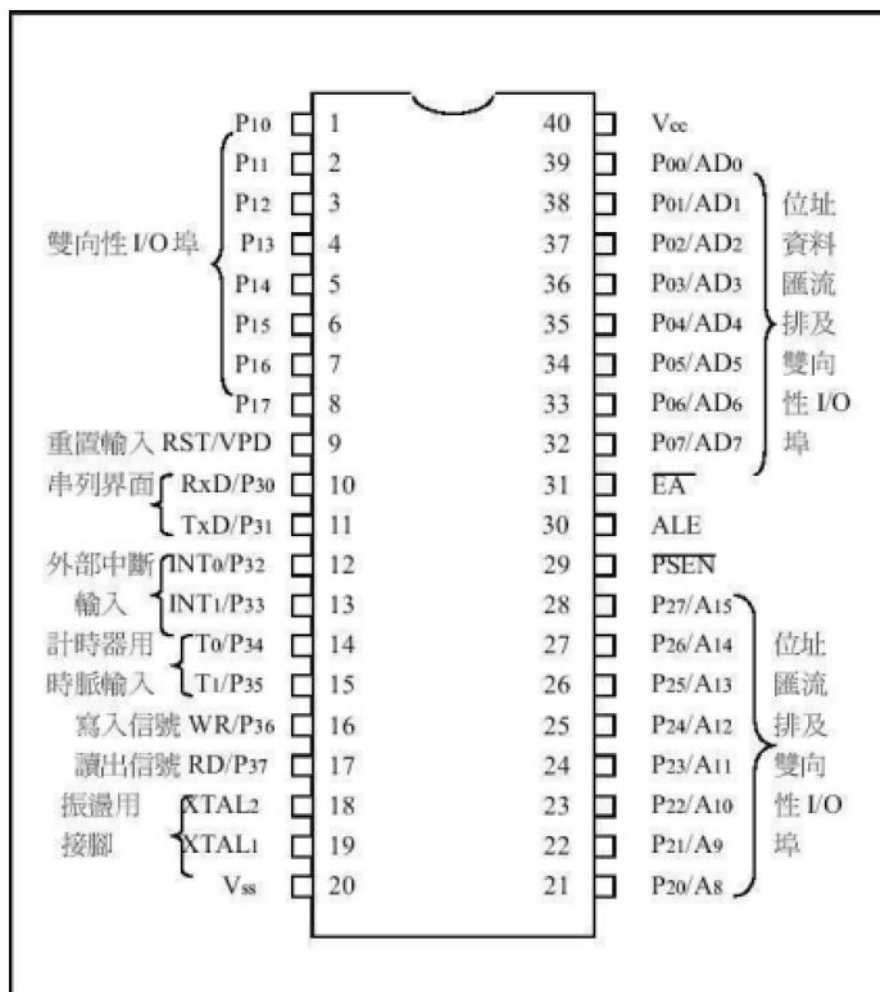


圖2-1-1 MCS-51 接腳圖

1. 內部加上外部的程式記憶體 (ROM) 共64K byte
2. 可在外部擴充64K byte 資料記憶體 (RAM)
3. 內部資料記憶體空間256 byte

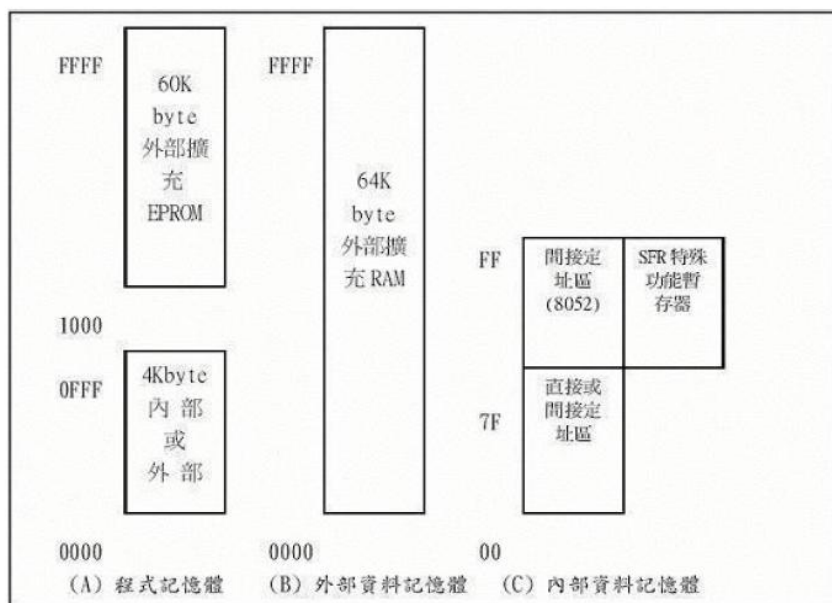


圖2-1-2 MCS-51 記憶體結構圖

二、單晶片微處理機

(一) 單晶片微處理機的簡介

一個微電腦需包含微處理器(CPU)，存放程式指令(ROM)及存取的資料的RAM，輸入/輸出埠(I/O埠)及時脈、計數器、中斷系統等。它們經由位址匯流排(Address Bus)、資料匯流排(Data Bus)和控制匯流排(Control Bus)的連接，及透過輸入/輸出埠與週邊裝置連線，構成為電腦系統。由於單晶片微處理機是把為電腦的主要元件製造在一塊晶片上，所以可以把單晶片微處理機看成是一個不帶週邊裝置的微電腦。

單晶片微處理機具有以下特點：

1. 受密度限制：晶片內記憶容量較小，ROM小於64K，RAM小於1K。
2. 可靠性良好：單晶片是依工業控制的要求所設計的，其抗工業雜訊干擾優於一般的CPU，程式指令及常數資料燒錄在ROM內，因其許多訊號通道均在同一個晶片內，故可靠性高。
3. 易擴充：單晶片具有一般微電腦所需的組件，如三態雙向匯流排，平行及串列的輸入/輸出接腳，可以擴充為各種規模的微電腦系統。
4. 控制功能強：為了滿足工業控制的要求，單晶片的指令，除了輸入/輸出控制指令，邏輯判斷指令外，更具有極豐富的條件分歧跳躍指令。

(二) 單晶片微處理機的應用範圍

1. 智慧型產品：單晶片與傳統的機械產品相結合，使傳統機械產品結構簡化、控制智慧化、構成新一代的機電整合的產品。例如電打字機採用單晶片，取代近千個機械組件，縫紉機採用單晶片作控制，可執行多功能自動操作、自動調速、控制縫補花樣的選擇。
2. 智慧型儀表：用單晶片改良原有的測量，控制儀表，能使儀表數位化、智慧化、多功能化、綜合化。而測量儀器中的誤差修正，線性化等問題也可迎刃而解。
3. 測控系統：用單晶片可以設計各種工業控制系統、環境控制系統、資料控制系統。例如溫室控制、水閘自動控制、電鍍生產線自動控制，及汽輪機電液調節系統。
4. 數值控制機：在目前數位控制系統的簡易控制機中，採用單晶片可提高其可靠性及增強功能，降低控制機成本。
5. 智慧型介面：用單晶片進行介面的控制與管理，單晶片與主機平行工作，可大大地提高了系統的執行速度。如在大型資料讀取系統中，用單晶片對A/D轉換進行控制不僅可提高讀取速度，還可對資料進行預先處理，如數位濾波、線性化處理與誤差修正等。在通訊界面中使用單晶片可對資料進行編碼解碼、分配管理、接收/發送控制等。

(三) 微電腦硬體結構

微電腦硬體結構包含中央處理單元、記憶體單元、輸入單元與輸出單元等四個主要單元，其結構關係則如下圖所示。

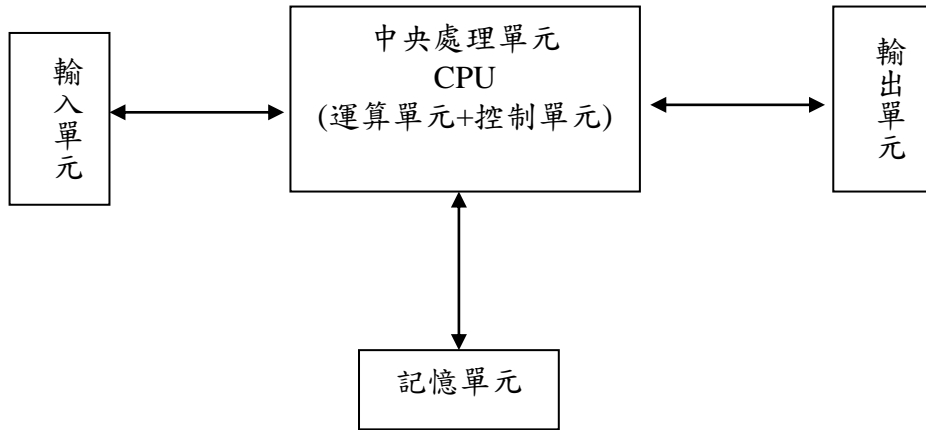


圖 2-2-1 微電腦硬體介面結構圖

其中，中央處理單元則是由運算單元與控制單元兩部分所組成的單元，即是一般所通稱的 CPU(Center Processing Unit)，此為微電腦最重要的部分。以下就微電腦中各單元的功能做簡單介紹：

1、運算單元(Arithmetic Logic Unit，簡稱 ALU)

運算單元又稱為算數邏輯單元，在中央處理單元中可用於執行算數運算，(如：加、減、乘、除等)，以及邏輯運算(如：AND、OR、NOT 等)，能將記憶體單元或輸入單元送至中央處理單元的資料執行各種運算。當運算完成後再由控制單元將結果資料送至記憶體單元或輸出單元。

2、控制單元(Control Unit，簡稱 CU)

此單元在中央處理單元中，負責協調與指揮各單元間的資料傳送與運作，使得微電腦可依照指令的要求完成工作。在執行一個指令時，控制單元先予以解碼(Decode)，瞭解指令的動作意義後再執行(Execute)該指令，因此控制單元將指令逐一執行，直到做完整個程式的所有指令為止。

3、輸入單元(Input Unit，簡稱 IU)

此單元是用以將外部的資訊傳送到 CPU 做運算處理或存入記憶體單元，一般在為電腦的輸入單元有鍵盤、磁碟機、光碟機、滑鼠、光筆、掃描器或讀卡機等週邊設備。

4、輸出單元(Output Unit，簡稱 OU)

此單元是用以將 CPU 處理過的資料輸出或儲存傳送外部週

邊設備，一般在為電腦的輸出單元有顯示器、印表機、繪圖機、燒錄機或磁碟機等週邊設備。

5、記憶單元(Memory Unit，簡稱 MU)

記憶體單元是用來儲存輸入單元傳送來的資料，或儲存經過中央處理單元處理完成的資料。記憶體單元之記憶體可分為主記憶體(Main Memory)與輔助記憶體(Auxiliary Memory)兩種，而主記憶體依存取方式不同，又可分為唯讀記憶體(Read Only Memory，簡稱 ROM)與隨機存取記憶體(Random Access Memory，簡稱 RAM)。ROM 所儲存的資料，在微電腦中只能被讀出但不能被寫入，也不會因為關機斷電而使資料流失；至於 RAM 在微電腦中，則可被讀出或寫入資料，但在關機斷電後儲存於 RAM 中的資料將會流失。輔助記憶體則是指磁片、硬碟或磁帶等週邊硬體，一般亦為輸出入單元，主要用來彌補主記憶體的不足，其容量可無限制擴充。

(四) AT89S51/P89C51 特性

微電腦系統包括 CPU、記憶體(RAM/ROM)及 I/O 介面三大部分，而單晶片微處理器就是把 CPU、記憶體及 I/O 等製作在同一個晶片上，作成體積小、成本低、硬體接線容易及擴充性佳的微電腦控制系統。8051 於 1980 年由 Intel 公司所開發的，迄今已將近三十年，各公司開發相關家族的晶片也很多，不過目前以 ATMEL 的 AT89S51/52 及 PHILIPS 的 P89C51 等兩種族系晶片為主流。其特性如下：

- 8位元的CPU。
- 32條雙向I/O。
- 128/256bytes資料記憶體RAM，可擴充至64K。(AT89S51/52)
512/1Kbytes資料記憶體RAM，可擴充至64K。(P89C51)
- 2/3個16位元計時器。
- 具全雙工串列埠UART。
- 6/8個中斷源。

AT89S51：EX1、TF0、TF1、RI、TI 6個中斷源。

AT89S52：EX0、EX1、TF0、TF1、RI、TI、TF2、EXF2 8個中斷源。

- 晶片內具有時脈振盪電路。
- 雙指標暫存器(DPTR)。
- 14位元看門口計數器(WDT)。
- PCA計數器陣列(P89C51)。
- 可線上燒錄(ISP, In-System Programmable) 的快閃記憶體(Flash Memory)，只要5V電壓，即可燒錄與清除。

(五) 單晶片的接腳

89C51 與 8051 皆為 40 支接腳之單晶片，其接腳圖與功能說明如下：

P1.0	1		40	Vcc
P1.1	2		39	P0.0/AD0
P1.2	3		38	P0.1/AD1
P1.3	4		37	P0.2/AD2
P1.4	5		36	P0.3/AD3
P1.5	6	8	35	P0.4/AD4
P1.6	7	0	34	P0.5/AD5
P1.7	8	5	33	P0.6/AD6
R5T	9	1	32	P0.7/AD7
RXD/p3.0	10		31	\overline{EA}
TXD/P3.1	11	單	30	ALE
$\overline{INT0}$ /P3.2	12		29	\overline{PSEN}
$\overline{INT1}$ /P3.3	13	晶	28	P2.7/A15
T0/P3.4	14		27	P2.6/A14
T1/P3.5	15	片	26	P2.5/A13
\overline{WR} /P3.6	16		25	P2.4/A12
\overline{RD} /P3.7	17		24	P2.3/A11
XTAL2	18		23	P2.2/A10
XTAL1	19		22	P2.1/A9
GND	20		21	P2.0/A8

圖 2-2-2 單晶片 8051 的接腳圖

- 1.Vcc：+5 電源供應接腳。
- 2.GND：接地接腳。
- 3.P0.0~P0.7：埠 0，為開洩極(OpenDrain)雙向 I/O 埠。在做為外部擴充記憶體時，可低八位元位址線(A0~A7addressline)與資料匯流排(databus)雙重功能。在做為一般 I/O 埠時必須加上如下之外部提升電路。

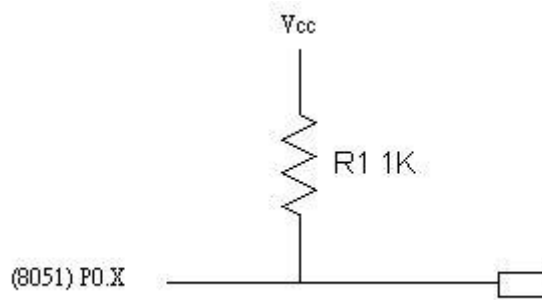


圖 2-2-4 單晶片埠 0 應用於 I/O 時的提升電路圖

- 4.P1.0~P1.7：埠 1，為具有內部提升電路的雙向 I/O 埠。
- 5.P2.0~P2.7：埠 2，為具有內部提升電路的雙向 I/O 埠。在做為外部擴充記憶體時，可為高八位元位址線(A8~A15addressline)。
- 6.P3.0~P3.7：埠 3，為具有內部提升電路的雙向 I/O 埠。此外，埠 3 的每支接腳都具有另一特殊功能，其功能如下：
 - RXD(P3.0)：串列傳輸的接收端。
 - TXD(P3.1)：串列傳輸的輸出端。
 - $\overline{INT0}$ (P3.2)：外部中斷輸入端。
 - $\overline{INT1}$ (P3.3)：外部中斷輸入端。
 - T0(P3.4)：計時/計數器外部輸入端。
 - T1(P3.5)：計時/計數器外部輸入端。
 - \overline{WR} (P3.6)：外部資料記憶體寫入激發信號(Strobe)。
 - \overline{RD} (P3.7)：外部資料記憶體讀取激發信號(Strobe)。
- 7.RST：重置信號(Reset)輸入端。在單晶片工作時，將此腳保持在“Hi”兩個機械週期，CPU 將重置。
- 8.ALE：位址鎖住致能(Address Latch Enable)，在每個機械週期都會出現，可做為外部電路的時脈源。
9. \overline{PSEN} ：程式激發致能(Program Strobe Enable)，可輸入外部程式記憶體的讀取信號。
10. \overline{EA} ：外部存取致能(External Access Enable)，當 EA 接腳為“L0”時，則讀取外部程式記憶體執行。
- 11.XTAL1：反相振盪放大器的輸入端。

12.XTAL2：反相振盪放大器的輸出端。

(六) 單晶片程式指令介紹

組合語言程式其定址法可分為直接定址法、間接定址法、暫存器定址法、立即定址法、索引定址法、位元定址法等六種，其意義如下：

1.直接定址法

所謂直接定址法，就是在指令中，直接定運算元所在的位址。僅適用於內部資料記憶體(RAM)及特殊功能暫存器(SFR)。

如下：

MOV A, 3FH；把位址 3FH 的內容存入累加器 A

ADD A, 30H；把位址 30H 的內容加到累加器 A

2.間接定址法

間接定址法是把運算元的位址存放在一個暫存器，這個暫存器就是運算元位址的指標。

3.暫存器定址法

8051 內部 RAM 的每個暫存器庫均含有 8 個暫存器，稱為 RO-R7，若運算元是使用 RO-R7 的位址都稱為暫存器定址法。

如下：

MOV A, R7；把暫存器 R7 的內容存入累加器 A

MOV A, R6；把暫存器 R3 的內容加到累加器 A

4.立即定址法

立即定址法是把運算元直接放在運算碼的後面。若運算元是常數資料，則必須以“#”號當作立即值的前置符號。

如下：

MOV A, #30H；把一個常數 30H 存入累加器 A

MOV R5, #05H；把一個常數 05H 存入 R5 暫存器

5.索引定址法

8051 的索引定址法僅適用於 ROM(程式記憶體)，而且只能讀出，不能寫入。所謂索引定址法就是以一個基底暫存器的內容，再加上一個索引暫存器的內容，所得的值即是運算元所在的位址。採索引定址法時，當基底暫存器的是 DPTR(資料指標暫存器)或 PC(程式記數器)，當

索引暫存器的則是累加器 A。

如下：

```
MOV    A , #30H
```

```
MOV    DPTR , #300H
```

```
MOVC   A , @A+DPTR
```

；將程式記憶體位址 330H(30H+300H)的內容存入
累加器 A

6.位元定址法

位元定址法是指對內部資料記憶體(RAM)及特殊功能暫存器(SFR)的某個位元直接設定或清除。就因為 8051 具有位元定址法，所以我們可以輕易的控制功能強大的特殊功能暫存器(SFR)，讓 8051 發揮最大效用，這是 8051 很重要的角色。但是位元定址法，只能使用於可位元址的暫存器。

如下：

```
SETB C ;設定進位旗標 C 為 1。
```

```
SETB P1、0；設定埠 1(P1)的第 0 位元為 1。
```

```
MOV C , ACC、2
```

；把累積器 ACC 的第 2 位元的值存入進位旗標。

參、專題製作

此章共分為三節依序說明本專題所應用到之設備及器材、製作方法與步驟及專題製作等。

一、設備及器材

表 3-1-1 專題製作使用儀器（軟體）設備一覽表

儀器（軟體） 設備名稱	應用說明
個人電腦	專題報告、電路圖製作及進行專題成品電路測試
數位相機	紀錄整個專題製作流程
雷射印表機	列印專題資料、圖片及專題報告成果
三用電錶	測量零件有無損壞及專題電路板各信號之量測
IC 萬用燒錄器	利用燒錄器將程式燒錄至 89C51 單晶片中
電源供應器	提供專題成品所需之電源
Microsoft Office Word	專題報告、製作過程的撰寫
Microsoft Office Power Point	進行口頭報告、製作及專題成品報告呈現
Keil-C	單晶片組合語言程式之編輯、燒錄軟體
Protel 99SE	繪畫專題電路之線路圖

二、製作方法與步驟

本專題研究採用的是行動研究法，主要是由循環的研究歷程所構成，包括準備、實驗教學、電路資料分析及報告撰寫等階段。本研究之製作方法與步驟，如圖 3-2-1 所示。

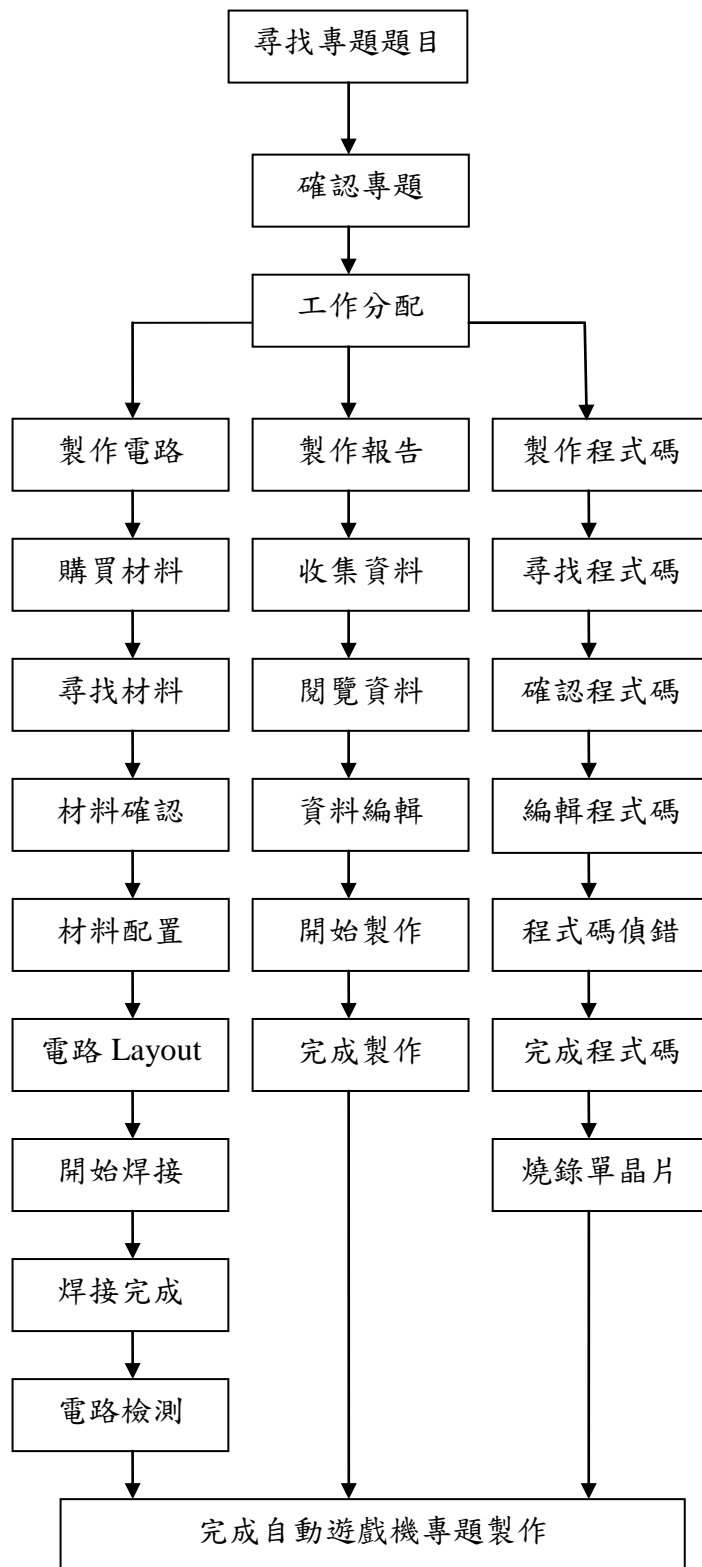


圖 3-2-1 製作方法與步驟

三、專題製作

(一) 微電腦自動遊戲機設定

我們將此專題看成兩部分，一部分為輸入部分，另一部分為輸出部分，其功能說明如下：

- 1.輸入：以 12 顆按鍵來模擬牙齒，微動開關偵測嘴巴是否開到最高。
- 2.輸出：直流馬達控制嘴巴打開及關閉，LED 顯示牙齒狀態，有按過則暗，沒按過則亮，其他的 LED、蜂鳴器增加聲光效果。

(二) 微電腦自動遊戲機功能

1. 開啟電源後，使用者啟動遊戲時，能自動產生亂數值，馬達控制嘴巴開啟，並播放音樂與 LED 全部亮起。
2. 使用者按任意鍵後，該按鍵搭配的 LED 熄滅，並進行掃描比對
 - (1) 如果與亂數值不同，能產生 LED 閃爍一下、嘴巴抖一下，用以驚嚇使用者，以提高遊戲的刺激性。
 - (2) 如果與亂數值吻合，馬達控制嘴巴關閉，並撥放音樂，LED 全部閃爍。

(二) 電路圖

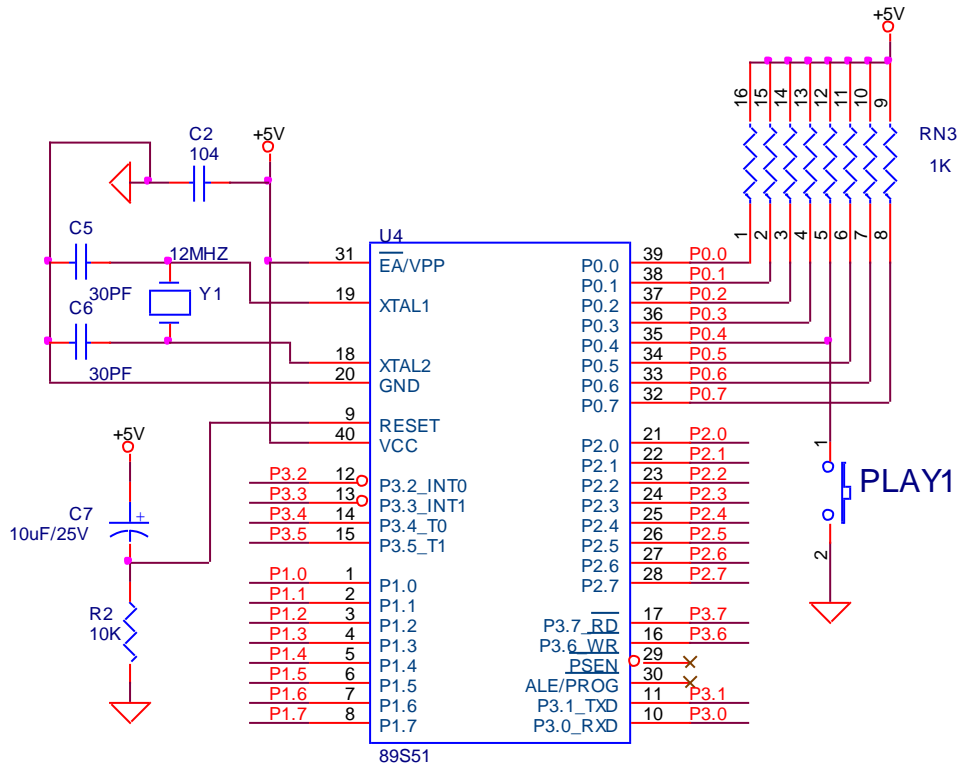


圖 3-3-1 8051 主電路

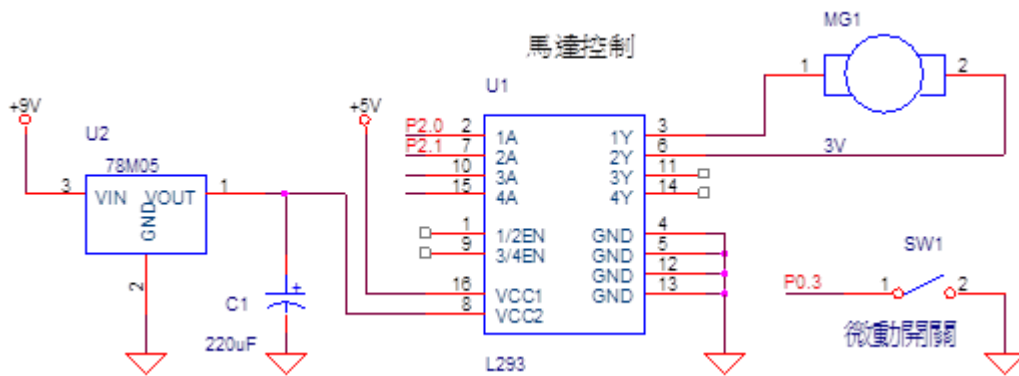


圖 3-3-2 馬達控制電路

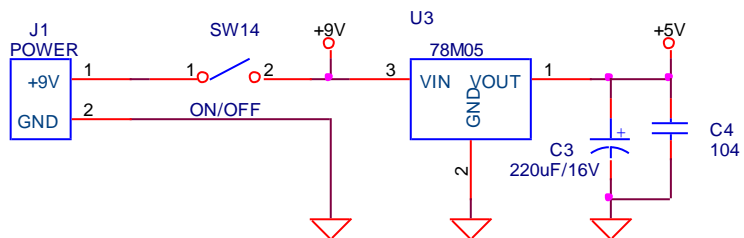


圖 3-3-3 電源電路

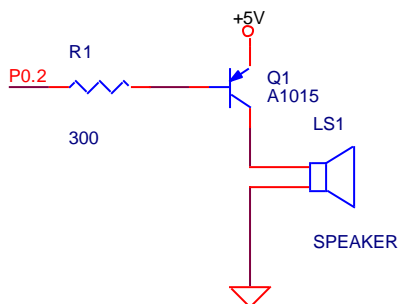


圖 3-3-4 蜂鳴器電路

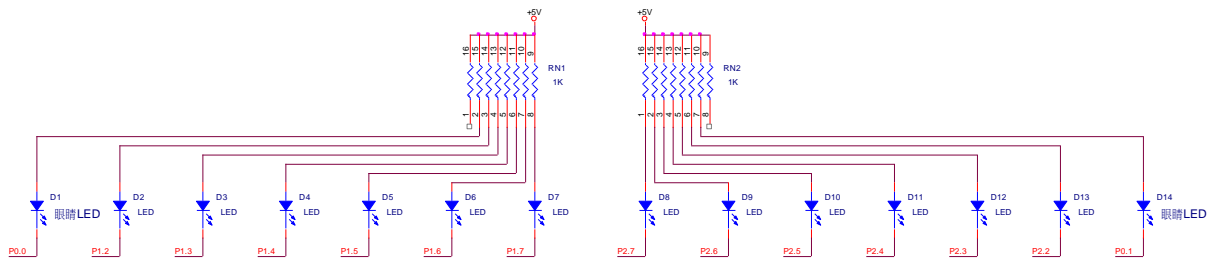


圖 3-3-5 LED 電路

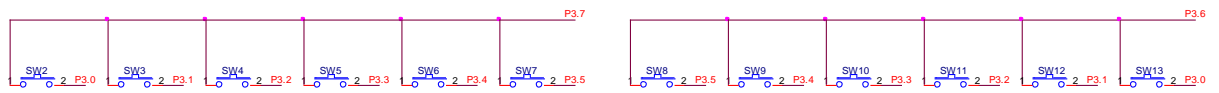


圖 3-3-6 按鍵電路

表 3-3-3 微電腦自動遊戲機之材料表

材料名稱	規 格	單 位	數 量	備註
無段開關	PLAY, SW2, SW3, SW4, SW5, SW6, SW7, SW8, SW9, SW10, SW11, SW12, SW13	個	13	
220uF 電解 電容	C1	個	1	
220uF 電解 電容	C2, C4	個	2	
220uF/16V 電解電容	C3	個	1	
30PF 陶瓷 電容	C5, C6	個	2	
10uF/25V 電解電容	C7	個	1	
眼睛 LED	D1, D14	個	2	
LED	D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13	個	12	
變壓器插座	J1	個	1	
蜂鳴器	LS1	個	1	
3V 小馬達 組	MG1	個	1	
A1015 電晶 體 PNP	Q1	個	1	
1K 電阻	RN1, RN2, RN3	個	3	
300 電阻	R1	個	1	
10K 電阻	R2	個	1	
微動開關	SW1	個	1	
ON/OFF	SW14	個	1	
L293	U1	個	1	
78M05	U3, U2	個	2	
89S51	U4	個	1	
12MHZ	Y1	個	1	

肆、製作成果

由決定題目，製作模擬電路、繪製設計電路圖，進而完成焊接製作整個電路；這整個流程，都使用數位相機及相關電腦設備將之紀錄下來，經將這些資料整理過後，呈現在專題報告之中，如下所示：

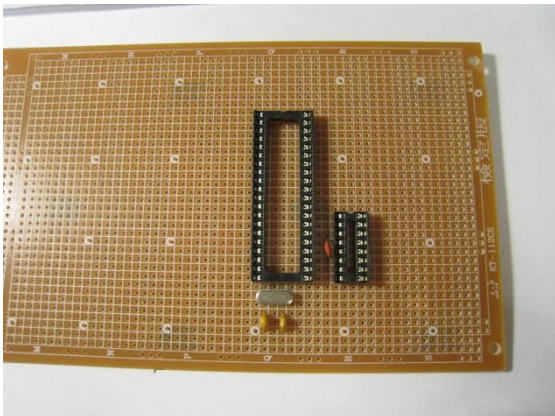


圖 4-1-1 遊戲機電路板製作過程(一)

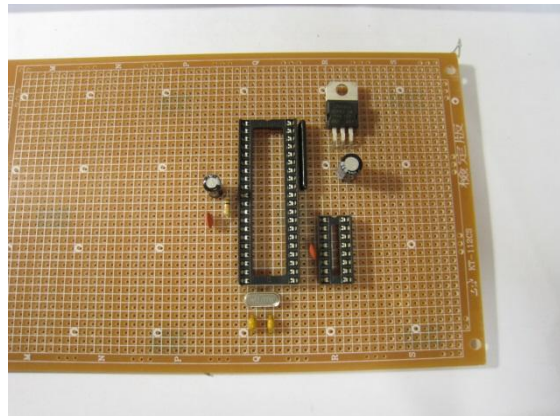


圖 4-1-2 遊戲機電路板製作過程(二)

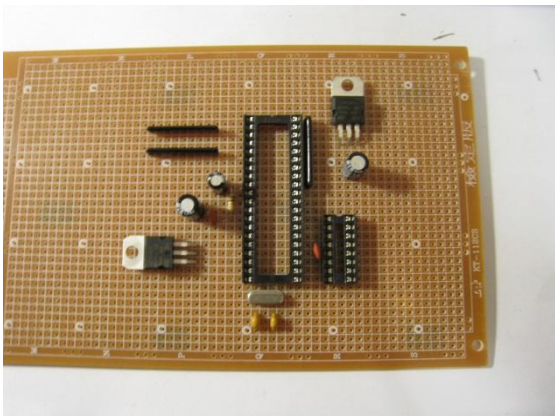


圖 4-1-3 遊戲機電路板製作過程(三)

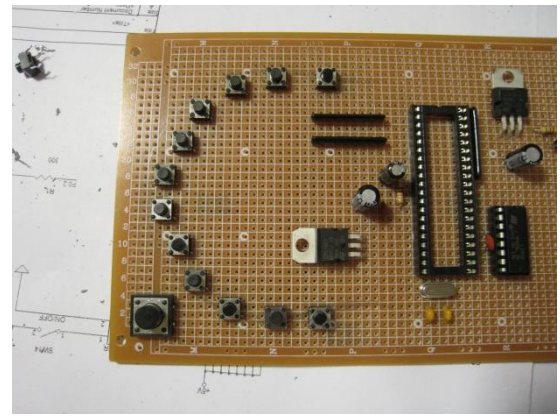


圖 4-1-4 遊戲機電路板製作過程(四)

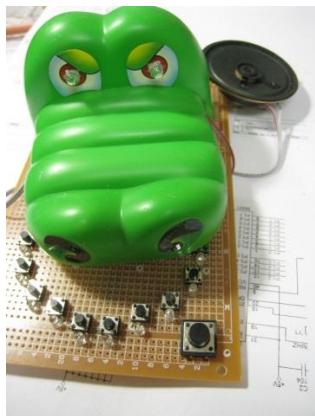


圖 4-1-5 微電腦自動遊戲機成品

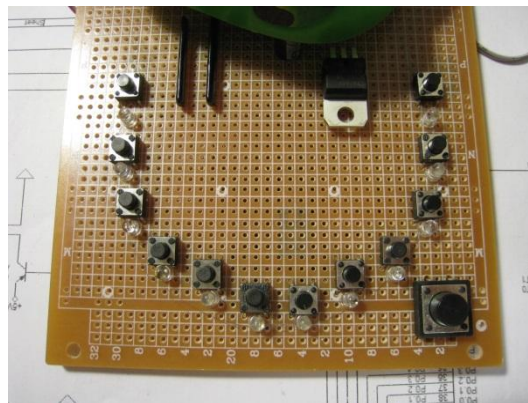


圖 4-1-6 測量按鈕(一)

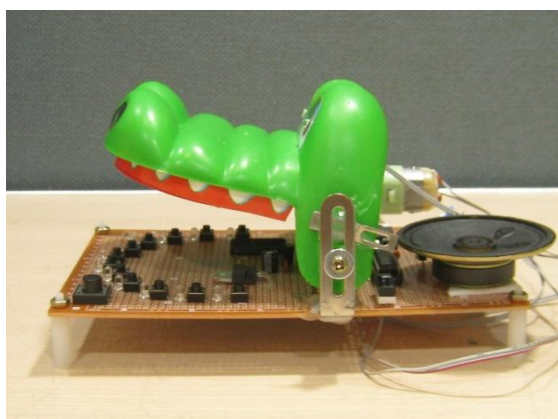


圖 4-1-7 測量音量(二)

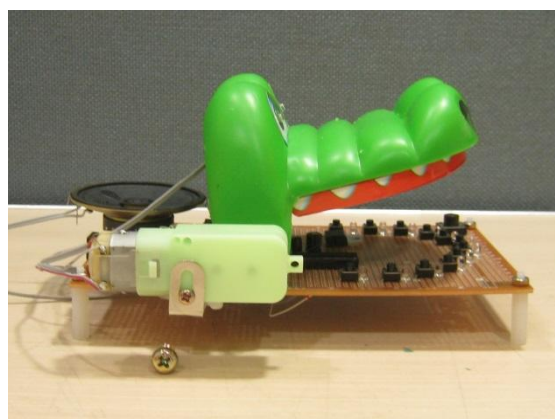


圖 4-1-8 外觀檢測(三)

參考文獻

1. 電子實習與專題製作-感測器應用篇，盧明智，全華圖書股份有限公司，97年10月。
2. 單晶片微電腦 8051 原理與應用，蔡朝洋，全華圖書股份有限公司，92年8月。
3. 單晶片 8051 實作入門，陳明熒，文魁，94年1月。
4. SN74LS47N 的 DATA SHEET。
5. 89C51 的 DATA SHEET。
6. 郭庭吉，2008，8051 單晶片微電腦專題製作，台北縣：台科大圖書公司。
7. 創意導航核心，2007，PhotoImpact 12 影像哈燒秀，台北市：全華圖書公司。
8. 蔡朝洋，2007，單晶片電腦 8051/8951 原理與應用，台北縣：全華圖書公司。
9. 鄧明發，陳茂璋，2000，微電腦專題製作應用電路，台北市：知行文化公司。
10. 鍾明政，1999，單晶片 8051 原理與實作，台中市：長高企業公司。

附錄一 微電腦自動遊戲機之程式碼

SCAN:

```
CLRP3.6    ;掃描訊號
SETB  P3.7
CALL  DELAY2    ;延時
JB  P3.0 , SC1    ;偵測按鍵 1
SETB  P2.2    ;LED 燈暗 1
MOV  31H , #1    ;31H 設為 1
CALL  JUDGE    ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.0 , $    ;等待放開按鍵
CALL  DELAY2    ;延時
JMP SCOK    ;跳 SCOK
```

SC1:

```
JB  P3.1 , SC2    ;偵測按鍵 2
SETB  P2.3    ;LED 燈暗 2
MOV  31H , #2    ;31H 設為 2
CALL  JUDGE    ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.1 , $    ;等待放開按鍵
CALL  DELAY2    ;延時
JMP SCOK    ;跳 SCOK
```

SC2:

```
JB  P3.2 , SC3    ;偵測按鍵 3
SETB  P2.4    ;LED 燈暗 3
MOV  31H , #3    ;31H 設為 3
CALL  JUDGE    ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.2 , $    ;等待放開按鍵
CALL  DELAY2    ;延時
JMP SCOK    ;跳 SCOK
```

SC3:

```
JB  P3.3 , SC4    ;偵測按鍵 4
```

```

SETB  P2.5      ;LED 燈暗 4
MOV    31H, #4   ;31H 設為 4
CALL   JUDGE     ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB   P3.3, $    ;等待放開按鍵
CALL   DELAY2    ;延時
JMP   SCOK       ;跳 SCOK

```

SC4:

```

JB    P3.4, SC5  ;偵測按鍵 5
SETB  P2.6      ;LED 燈暗 5
MOV    31H, #5   ;31H 設為 5
CALL   JUDGE     ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB   P3.4, $    ;等待放開按鍵
CALL   DELAY2    ;延時
JMP   SCOK       ;跳 SCOK

```

SC5:

```

JB    P3.5, SCOK ;偵測按鍵 6
SETB  P2.7      ;LED 燈暗 6
MOV    31H, #6   ;31H 設為 6
CALL   JUDGE     ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB   P3.5, $    ;等待放開按鍵
CALL   DELAY2    ;延時
JMP   SCOK       ;跳 SCOK

```

SCOK:

```

RET

```

```

;=====
;====掃描左半部 6 鍵副程式=====
;=====

```

SCAN2:

```

SETB  P3.6      ;掃描訊號

```

CLRP3.7

```
CALL DELAY2 ;延時
JB P3.0, SC7 ;偵測按鍵 7
SETB P1.2 ;LED 燈暗 7
MOV 31H, #7 ;31H 設為 7
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.0, $ ;等待放開按鍵
CALL DELAY2 ;延時
JMP SCOK2 ;跳 SCOK2
```

SC7:

```
JB P3.1, SC8 ;偵測按鍵 8
SETB P1.3 ;LED 燈暗 8
MOV 31H, #8 ;31H 設為 8
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.1, $ ;等待放開按鍵
CALL DELAY2 ;延時
JMP SCOK2 ;跳 SCOK2
```

SC8:

```
JB P3.2, SC9 ;偵測按鍵 9
SETB P1.4 ;LED 燈暗 9
MOV 31H, #9 ;31H 設為 9
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.2, $ ;等待放開按鍵
CALL DELAY2 ;延時
JMP SCOK2 ;跳 SCOK2
```

SC9:

```
JB P3.3, SCA ;偵測按鍵 10
SETB P1.5 ;LED 燈暗 10
MOV 31H, #10 ;31H 設為 10
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
```

JNB P3.3, \$;等待放開按鍵

CALL DELAY2 ;延時

JMP SCOK2 ;跳 SCOK2

SCA:

JB P3.4, SCB ;偵測按鍵 11

SETB P1.6 ;LED 燈暗 11

MOV 31H, #11 ;31H 設為 11

CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)

JNB P3.4, \$;等待放開按鍵

CALL DELAY2 ;延時

JMP SCOK2 ;跳 SCOK2

SCB:

JB P3.5, SCOK2;偵測按鍵 12

SETB P1.7 ;LED 燈暗 12

MOV 31H, #12 ;31H 設為 12

CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)

JNB P3.5, \$;等待放開按鍵

CALL DELAY2 ;延時

JMP SCOK2 ;跳 SCOK2

SCOK2:

RET

START:

CLRP2.0 ;馬達初始狀態(停止)

CLRP2.1

MOV 30H, #1 ;初始值 30H 設為 1

MOV 31H, #0 ;初始值 31H 設為 0

MOV 32H, #0 ;初始值 32H 設為 0

LOOP:

INC 30H ;30H 為亂數值

MOV A, 30H ;從 1 加到 13

CJNE A, #13, LOP2 ;超過 13 時，變 1

MOV 30H, #1

LOP2:

JB P0.4, LOOP ;1 加到 13 的期間按下 PLAY 鍵時，程式往下，否則跳 LOOP

JNB P0.4, \$;等待放開按鍵

CLRP2.0 ;馬達控制嘴巴打開

SETB P2.1

JB P0.3, \$;嘴巴打開的期間，壓到微動開關時，程式往下，否則原地

跳

CLRP2.0 ;馬達停止

CLRP2.1

MOV DPTR, #MUSIC2 ;DPTR 指向樂譜之開頭

CALL MUSICPLAY ;撥放音樂

MOV P2, #00000000B ;LED 全亮

MOV P1, #00000000B ;LED 全亮

LOOP3:

CALL SCAN2 ;呼叫掃描左半部 6 鍵副程式

CALL SCAN ;呼叫掃描右半部 6 鍵副程式

MOV A, 32H ;判斷 32H

CJNE A, #1, LOOP3 ;32H 若為 1，程式往下，否則跳 LOOP3

MOV 32H, #0 ;表示遊戲結束，跳回 LOOP 等待下一次遊戲

JMP LOOP

=====

=====掃描右半部 6 鍵副程式=====

=====

SCAN:

CLRP3.6 ;掃描訊號

SETB P3.7

CALL DELAY2 ;延時

JB P3.0, SC1 ;偵測按鍵 1

SETB P2.2 ;LED 燈暗 1

```
MOV    31H, #1    ;31H 設為 1
CALL   JUDGE      ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB   P3.0, $     ;等待放開按鍵
CALL   DELAY2     ;延時
JMP    SCOK       ;跳 SCOK
```

SC1:

```
JB    P3.1, SC2   ;偵測按鍵 2
SETB  P2.3        ;LED 燈暗 2
MOV    31H, #2    ;31H 設為 2
CALL   JUDGE      ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB   P3.1, $     ;等待放開按鍵
CALL   DELAY2     ;延時
JMP    SCOK       ;跳 SCOK
```

SC2:

```
JB    P3.2, SC3   ;偵測按鍵 3
SETB  P2.4        ;LED 燈暗 3
MOV    31H, #3    ;31H 設為 3
CALL   JUDGE      ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB   P3.2, $     ;等待放開按鍵
CALL   DELAY2     ;延時
JMP    SCOK       ;跳 SCOK
```

SC3:

```
JB    P3.3, SC4   ;偵測按鍵 4
SETB  P2.5        ;LED 燈暗 4
MOV    31H, #4    ;31H 設為 4
CALL   JUDGE      ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB   P3.3, $     ;等待放開按鍵
CALL   DELAY2     ;延時
JMP    SCOK       ;跳 SCOK
```

SC4:

```

JB P3.4, SC5 ;偵測按鍵 5
SETB P2.6 ;LED 燈暗 5
MOV 31H, #5 ;31H 設為 5
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.4, $ ;等待放開按鍵
CALL DELAY2 ;延時
JMP SCOK ;跳 SCOK

```

SC5:

```

JB P3.5, SCOK ;偵測按鍵 6
SETB P2.7 ;LED 燈暗 6
MOV 31H, #6 ;31H 設為 6
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.5, $ ;等待放開按鍵
CALL DELAY2 ;延時
JMP SCOK ;跳 SCOK

```

SCOK:

```
RET
```

```

;=====
;====掃描左半部 6 鍵副程式=====
;=====

```

SCAN2:

```

SETB P3.6 ;掃描訊號
CLR P3.7
CALL DELAY2 ;延時
JB P3.0, SC7 ;偵測按鍵 7
SETB P1.2 ;LED 燈暗 7
MOV 31H, #7 ;31H 設為 7
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.0, $ ;等待放開按鍵

```

CALL DELAY2 ;延時
JMP SCOK2 ;跳 SCOK2

SC7:

JB P3.1, SC8 ;偵測按鍵 8
SETB P1.3 ;LED 燈暗 8
MOV 31H, #8 ;31H 設為 8
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.1, \$;等待放開按鍵
CALL DELAY2 ;延時
JMP SCOK2 ;跳 SCOK2

SC8:

JB P3.2, SC9 ;偵測按鍵 9
SETB P1.4 ;LED 燈暗 9
MOV 31H, #9 ;31H 設為 9
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.2, \$;等待放開按鍵
CALL DELAY2 ;延時
JMP SCOK2 ;跳 SCOK2

SC9:

JB P3.3, SCA ;偵測按鍵 10
SETB P1.5 ;LED 燈暗 10
MOV 31H, #10 ;31H 設為 10
CALL JUDGE ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB P3.3, \$;等待放開按鍵
CALL DELAY2 ;延時
JMP SCOK2 ;跳 SCOK2

SCA:

JB P3.4, SCB ;偵測按鍵 11
SETB P1.6 ;LED 燈暗 11
MOV 31H, #11 ;31H 設為 11

```

CALL  JUDGE      ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB  P3.4, $     ;等待放開按鍵
CALL  DELAY2    ;延時
JMP  SCOK2      ;跳 SCOK2

```

SCB:

```

JB   P3.5, SCOK2;偵測按鍵 12
SETB P1.7      ;LED 燈暗 12
MOV  31H, #12  ;31H 設為 12
CALL  JUDGE    ;呼叫比對副程式 31H 和 30H 作比對(相同表按到蛀牙)
JNB  P3.5, $   ;等待放開按鍵
CALL  DELAY2   ;延時
JMP  SCOK2     ;跳 SCOK2

```

SCOK2:

```
RET
```

```

;=====
;====比對副程式 31H 和 30H 作比對=====
;=====

```

JUDGE:

```

MOV  A, 30H     ;30H->A
MOV  B, 31H     ;31H->B
CJNE A, B, SCLOOP ;A=B?, 是則程式往下, 否則跳 SCLOOP
SETB P2.0      ;馬達控制嘴巴關閉
CLRP2.1
CALL  DELAY    ;延時
CLRP2.0       ;馬達停止
CLRP2.1
MOV  DPTR, #MUSIC ;DPTR 指向樂譜之開頭
CALL MUSICPLAY ;撥放音樂
MOV  31H, #0    ;31H 回原值

```

MOV 32H, #1 ;32H 設為 1 作為判斷

MOV P2, #11111111B ;LED 全暗

MOV P1, #11111111B

RET

SCLOOP:

CPL P1.2 ;LED 閃一下 驚嚇使用者

CPL P1.3

CPL P1.4

CPL P1.5

CPL P1.6

CPL P1.7

CPL P2.2

CPL P2.3

CPL P2.4

CPL P2.5

CPL P2.6

CPL P2.7

SETB P2.0 ;嘴巴抖一下 驚嚇使用者

CLRP2.1

JNB P0.3, \$

CLRP2.0

SETB P2.1

CALL DELAY2

JB P0.3, \$

CLRP2.0

CLRP2.1

CPL P1.2 ;LED 閃一下 驚嚇使用者

CPL P1.3

CPL P1.4

CPL P1.5

CPL P1.6

CPL P1.7

CPL P2.2

CPL P2.3
CPL P2.4
CPL P2.5
CPL P2.6
CPL P2.7
RET

```
DELAY2:  MOV    R6 , #100
DL:      MOV    R7 , #200
         DJNZ   R7 , $
         DJNZ   R6 , DL
         RET
```

```
; =====
; ==延時副程式 R5 來定時==
; =====
```

```
DELAY:  MOV    R5 , #2
DL0:    MOV    R6 , #250
DL1:    MOV    R7 , #200
DL2:    DJNZ   R7 , DL2
         DJNZ   R6 , DL1
         DJNZ   R5 , DL0
         RET
```

```
;=====
;撥放音樂副程式=====
;=====
```

MUSICPLAY:

CONT:

```

CLR      A                ;讀取樂譜內之音階代碼，
MOVC     A, @A+DPTR      ;A=音階代碼
CJNE     A, #40, OK      ;若音階代碼為 40，則結束
RET

```

;讀取樂譜內之音拍代碼

```

OK:      CPL P0.0
          CPL P0.1
          CPL P1.2
          CPL P1.3
          CPL P1.4
          CPL P1.5
          CPL P1.6
          CPL P1.7
          CPL P2.2
          CPL P2.3
          CPL P2.4
          CPL P2.5
          CPL P2.6
          CPL P2.7

```

```

PUSH     ACC              ;保存 A 之內容
INC      DPTR             ;讀取樂譜內知音拍代碼
CLR      A
MOVC     A, @A+DPTR
MOV      R4, A            ;R4=音拍代碼
POP      ACC              ;取回 A 之內容

```

```

;=====
;==      依據音階代碼發出相對應之聲音      ==
;=====

```

```

CHK11:  CJNE     A, #11, CHK12 ;若音階代碼為 11，則產生中音 DO 之聲音

```

ACALL DDO

CHK12: CJNE A, #12, CHK13 ;若音階代碼為 12，則產生中音 RE 之聲音

ACALL RE

CHK13: CJNE A, #13, CHK14 ;若音階代碼為 13，則產生中音 MI 之聲音

ACALL MI

CHK14: CJNE A, #14, CHK15 ;若音階代碼為 14，則產生中音 FA 之聲音

ACALL FA

CHK15: CJNE A, #15, CONT2 ;若音階代碼為 15，則產生中音 SO 之聲音

ACALL SO

CONT2:

INC DPTR ;繼續讀取樂譜

LJMP CONT

```
;=====
;== 設定各音階代碼之相對應參數 ==
;=====
```

;設定中音 DO 之相對應參數

DDO: MOV R6, #126

MOV R5, #33

LJMP OUTPUT

;設定中音 RE 之相對應參數

RE: MOV R6, #113

MOV R5, #37

LJMP OUTPUT

;設定中音 MI 之相對應參數

```
MI:      MOV      R6, #100
         MOV      R5, #41
         AJMP     OUTPUT
```

;設定中音 FA 之相對應參數

```
FA:      MOV      R6, #95
         MOV      R5, #44
         AJMP     OUTPUT
```

;設定中音 SO 之相對應參數

```
SO:      MOV      R6, #85
         MOV      R5, #49
         AJMP     OUTPUT
```

```
;=====
;==      輸出 R5 乘以 R4 週的方波      ==
;=====
```

```
OUTPUT:  PUSH     05                ;保存 R5 之內容
MLOOP:   CJNE    A, #00, SOUND      ;若音階代碼為 00，
         LJMP    MUTE              ;則不讓揚聲器通電
SOUND:   CLR     P0.2              ;令揚聲器通電
MUTE:    ACALL   DELAYM            ;延時半週期
         SETB   P0.2              ;令揚聲器斷電
         ACALL   DELAYM            ;延時半週期
         DJNZ   R5, MLOOP          ;輸出一個音長，共 R5 週的長波
```

```

POP      05          ;取回 R5 之內容
DJNZ    R4 , OUTPUT ; 一共輸出 R4 個音長

```

```

REST:    MOV      R6 , #170
         MOV      R5 , #50
WAIT:    ACALL   DELAYM
         DJNZ    R5 , WAIT
         RET

```

;延遲半週期 $t=15\mu s * R6 + 9\mu s$

```

DELAYM:  MOV      B , R6
DLL:     MOV      R7 , #6
         DJNZ    R7 , $
         DJNZ    R6 , DLL
         MOV      R6 , B
         RET

```

```

;=====
;==          樂  譜          ==
;=====

```

;以下是[我是一隻小小鳥]的樂譜

```

MUSIC:  DB      11 , 02 , 11 , 02 , 11 , 02
         DB      13 , 03 , 12 , 01 , 11 , 02
         DB      13 , 02 , 13 , 02 , 13 , 02
         DB      15 , 03 , 14 , 01 , 13 , 02
         DB      15 , 02 , 14 , 02 , 13 , 02
         DB      12 , 04 , 00 , 02
         DB      40          ;停止演奏

```

MUSIC2: DB 11 , 01 , 12 , 01 , 13 , 01
DB 40 ;停止演奏
END

附錄二 單晶片 89C51 特性介紹

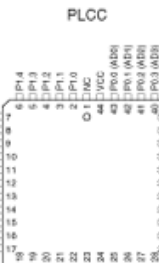
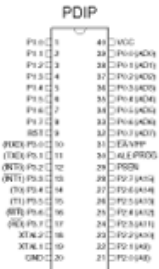
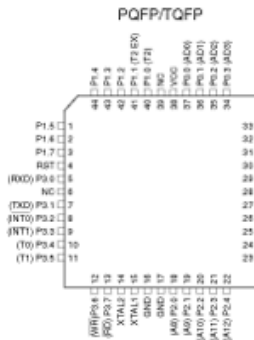
Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
 - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations



**8-bit
Microcontroller
with 4K Bytes
Flash**

AT89C51

**Not Recommended
for New Designs.
Use AT89S51.**

Rev. 0265G-02/00

