

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

專題製作報告



Android 藍芽遙控自走車

學生姓名： 羅 建 甫

蔡 政 翰

何 恩 富

指導老師： 葉 忠 賢 老師

中 華 民 國 104 年 05 月



## 誌 謝

我們要特別感謝指導老師葉忠賢老師，在專題寫作過程中，從資料蒐集與分析、甚至書寫修辭，盡心盡力的指導，逐字逐句、嚴謹的審閱修改，付出極大的辛勞，使我們獲益良多；在忠賢老師適時的鼓勵之下，專題終能如期完成。

在過程當中，遇到許多的問題，成品一直功能不正常，老師努力想辦法去修理它，直到最後功能正常了。同時，也感謝組員們的分工合作及資料收集，在學習期間的互相砥礪及陪伴與彼此加油打氣。

最後，還是要謝謝老師，正因為老師非常積極處理，才能使我們的專題製作更加完整，也能更順利的完成，在此本組特別的感謝，僅致上最高的敬意及謝意。在專題製作這門課，讓我們對於學習有更多的成長、思考，其中有苦也有樂，再畫上完美句點的同時，感謝曾經有幫助的同學。

組長：羅建甫、組員：蔡政翰、何恩富 謹上 2015/05

# Android 藍芽遙控自走車

## 摘 要

本篇研究報告旨在透過單晶片 89C51 的學習，了解單晶片的功能及使用方法，且經由實際製作電路的過程中去對單晶片運作有更深入的了解。會想製作藍芽遙控自走車，原因就是在這個多元且資訊化的世界，手機愈來愈普遍因此才想到利用藍芽來操控自走車，我們便利用專題製作的機會，研究藍芽的操作方式，探討如何設計一輛方便操控的自走車；故想要藉由設計一個單晶片電路，配合語言程式去達到自動控制的目的，也因為透過設計及製作過程中，可以知道及了解，如何透過語言程式去設計、控制一自走車電路。小組成員同學預期此設計能先達成做到啟動的功能及目的，若此階段沒問題了，再會進階地針對自走車電路再去做修改，再加入藍芽的技術，可以讓自走車更好操作，故現行之小組專題製作的目標即是想透過單晶片的學習，去製作一個自動化的自走車，以達到科技與生活相結合的目的。

**關鍵詞：**微電腦單晶片、自走車、ZK-05 藍芽

# 目 錄

誌謝 .....	I
摘要 .....	II
目錄 .....	III
表目錄 .....	IV
圖目錄 .....	V
壹、前言 .....	1
一、製作動機 .....	1
二、製作目的 .....	1
三、製作架構 .....	1
四、製作預期成效 .....	2
貳、理論探討 .....	4
參、專題製作 .....	14
一、設備及器材 .....	14
二、製作方法與步驟 .....	14
三、專題製作 .....	16
肆、製作成果 .....	21
伍、結論與建議 .....	26
一、結論 .....	26
二、建議 .....	26
參考文獻 .....	27
附錄一 Android 藍芽遙控自走車程式碼 .....	28

## 表目錄

表 3-1-1 專題製作使用儀器（軟體）設備一覽表 .....	14
表 3-1-2 藍芽自走車之材料表 .....	19

## 圖目錄

圖 1-3-1 專題製作流程圖 .....	2
圖 2-1-1 藍芽網路架構 .....	4
圖 2-1-2 馬達控制電路 .....	5
圖 2-2-1 單晶片內部構造 .....	5
圖 2-2-2 單晶片 .....	6
圖 2-2-3 單晶片電路圖 .....	6
圖 2-2-4 基本電路連接圖 .....	7
圖 2-2-5 8051 記憶體結構示意圖 .....	7
圖 2-2-6 程式記憶體(ROM) .....	8
圖 2-2-7 程式記憶體(RAM) .....	9
圖 2-2-8 計時器說明圖 .....	10
圖 2-2-9 工作模式 .....	11
圖 2-2-10 計時器控制( Timer Control , TCON )暫存器 .....	11
圖 2-2-11 串列埠控制( Serial Port Control )暫存器 .....	12
圖 2-2-12 串列資料緩衝( Serial Data Buffer , SBUF )暫存器 .....	12
圖 2-2-13 SFR 記憶體分佈圖 .....	13
圖 2-2-14 同步定向傳輸示意圖 .....	13
圖 3-1-1 設備及材料 .....	14
圖 3-2-1 製作方法及步驟 .....	14
圖 3-3-1 麵包板上電路模擬(一) .....	17
圖 3-3-2 麵包板上電路模擬(二) .....	17
圖 3-3-3 製作 Microsoft Office Power Point 報告 .....	17
圖 3-3-4 製作 Microsoft Office Word 簡報 .....	17
圖 3-3-5 藍芽自走車電路圖 .....	18
圖 4-1-1 藍芽自走車開關製作過程(一) .....	21
圖 4-1-2 藍芽自走車開關製作過程(二) .....	21
圖 4-1-3 藍芽自走車開關製作過程(三) .....	21

圖 4-1-4 藍芽自走車開關製作過程(四).....	21
圖 4-1-5 藍芽自走車開關製作過程(五).....	21
圖 4-1-6 完成藍芽自走車專題 .....	21
圖 4-1-7 藍芽遙控自走車說明圖 .....	22
圖 4-1-8 藍芽遙控自走車手機說明 .....	23
圖 4-1-9 手機操作步驟(一).....	24
圖 4-1-9 手機操作步驟(二).....	24
圖 4-1-9 手機操作步驟(三).....	24
圖 4-1-9 手機操作步驟(四).....	24
圖 4-1-9 手機操作步驟(五).....	25
圖 4-1-9 手機操作步驟(六).....	25



# 壹、前言

## 一、製作動機

藍芽系統是一項低功率、低成本、低耗電的短距離無線射頻介面且製成單一晶片的門檻較低。也因為藍芽系統為一種短距離的傳輸技術，所消耗的電力較小，故單一晶片技術製作較其它系統容易。且通訊頻帶最一般化，在全世界幾乎皆可通，故較易製造此收發通信模組。因此，本專題使用藍芽模組系統來做自走車之無線控制系統，有別於一般利用紅外線或超音波的方式進行遙控，期能發展出遙控車有更多控制方式的未來性。

## 二、製作目的

此專題是利用89C51單晶片和藍芽系統(bluetooth)模組，設計遠控制器。由藍芽系統傳輸指令至89C51單晶片對自走車進行控制，達到無線搖控功能，並藉由89C51單晶片將程式指令燒入晶片，達到智慧型自走車無人駕駛系統。

因此，本專題使用藍芽模組系統來做搖控之無線控制系統，有別於一般利用紅外線或超音波的方式進行遙控，更重要的是因其低功率的特性，所以對人體的影響相對降低。

## 三、製作架構

### (一) 專題製作流程

由組長決定專題的主題後，就已經邁向專題的第一步了，我們就開始討論到底該怎麼作，可是一想到電路圖跟程式之後就開始頭痛起來，因為專題的電路圖跟程式碼與之前所學的大大不同，所以我們便開始蒐集研究相關資料及相關的資訊，就開始研究，由於老師說硬體的部分不會影響到，有這樣的想法之後就開始放心規劃電路圖了。首先把電路圖畫出來，之後有哪些材料就開始購買，購買完畢之後，就可以開始著手，在麵包版上做出功能之後，就開始進行焊接。

## (二) 製作流程圖

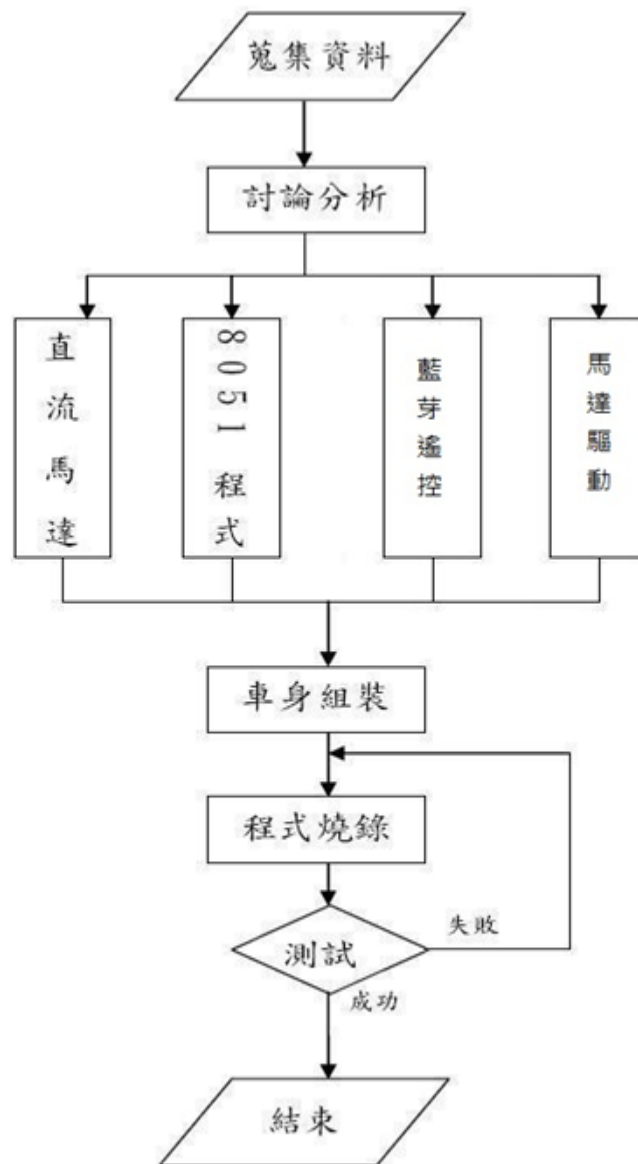


圖 1-3-1 專題製作流程圖

## 四、製作預期成效

我們這一個小組是第一次一起進行專題製作當時覺得非常困難的樣子還很擔心做不出來，但是在真正的去設計，去碰觸，去踏進這一個微電腦 8051 的世界，才了解這一小小的單晶片原來在我們的生活中扮演著這樣的角色，想不到的是我們無時無刻的在與它做朋友。此次的專題報告，發現到我們還

有有此地地方還是不太純熟，還有待我們去尋找出來，經過這一次的從尋找題目、設計、操作、……。我們都有長足的進步，對我們來說，這一次專題報告讓我們收獲良多。

將程式燒錄至 89C51 單晶片，再透過藍芽連線模組傳輸至手機裡面，再把手機的程式打開設定，完成後，就會出現操作介面，然而按照螢幕的方向讓車子行走。

## 貳、理論探討

### 一、電子相關零組件

#### (一) 藍芽

藍芽技術是一種小範圍的無線電頻率技術，裝置間透過晶片可互相溝通，不必再透過纜線傳輸。目前，低功率的無線傳輸仍以紅外線為主，應用的層面仍侷限在行動電話、PDA、電腦及其相關產品上。

此外，跳頻展頻在同步且同時的情況下，發射與接收兩端以特定型式的窄頻電波來傳送訊號，為了避免在一特定頻段受其他雜訊干擾，收發兩端傳送資料經過一段極短的時間後，便同時切換到另一個頻段，由於不斷的切換頻段，因此較能減少在一個特定頻道受到的干擾，也不容易被竊聽。跳頻展頻所展開的訊號，可依特別設計來規避雜訊或重覆的頻道，並且跳頻訊號必須遵守 FCC（Federal Communications Commission）的要求，使用 75 個以上的跳頻訊號，且跳頻至下一個頻率的最高時間間隔為 400ms，在 IEEE 802.11 中最高時間間隔通常定為 250ms，也就是每秒跳頻 4 次。而且，藍芽所發出的信號需低於 10 pico 瓦，而 GSM 手機啟動時會產生一至三瓦不等的發射功率，相較之下安全許多。

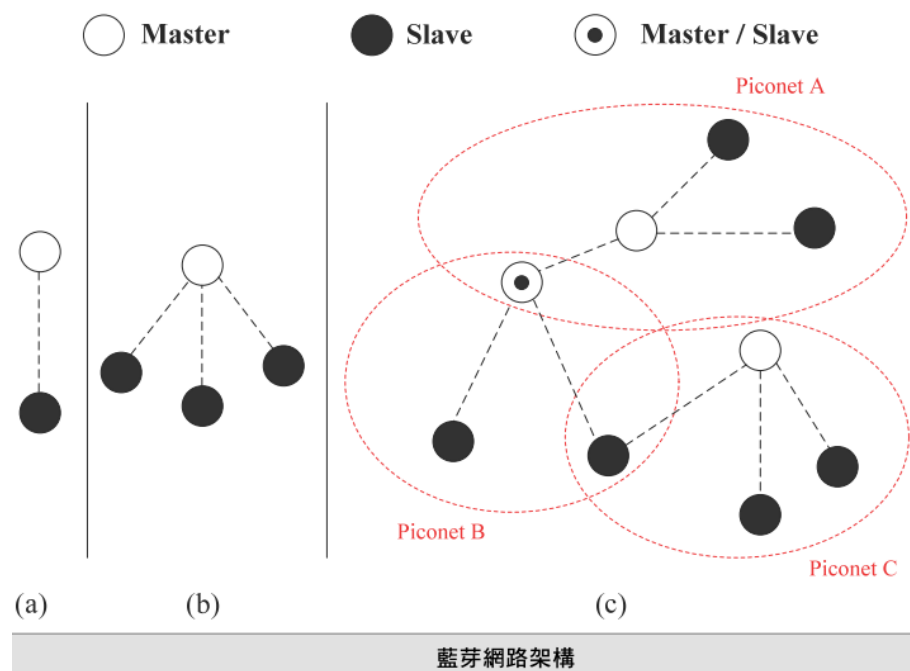


圖 2-1-1 藍芽網路架構

#### (二) 直流馬達

##### (1) 直流馬達原理：

由外部電源提供電流使通過轉子導線，以產生磁場與定子磁場相互作用而轉動。

##### (2) 馬達控制 IC(L293D)：

訊號從接腳 2、接腳 7、接腳 11、接腳 15 輸入

L1 與 L2 為左右馬達

當接腳 2 輸入訊號，L1 馬達正轉

接腳 7 輸入訊號，L1 馬達反轉

接腳 10 輸入訊號，L1 馬達正轉

接腳 15 輸入訊號，L1 馬達反轉

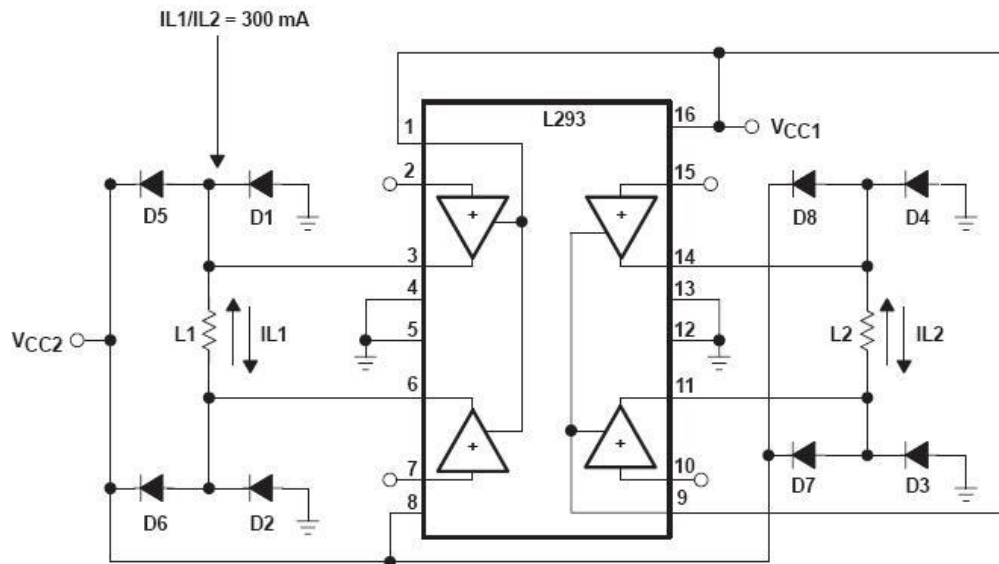


圖 2-1-2 馬達控制電路

## 二、8051單晶片介紹

微電腦普遍應用在日常生活中的一些自動化設備中，可以說帶動了整個人類科技的進步。本章主要介紹微電腦組成的基本架構，單晶片微電腦的特點及其應用領域使初學者可以很快地了解一套控制系統是如何構成的。

(1)8051內部結構：

8051 為Intel 公司推出的MCS-51 系列產品之一，其結構如下圖：

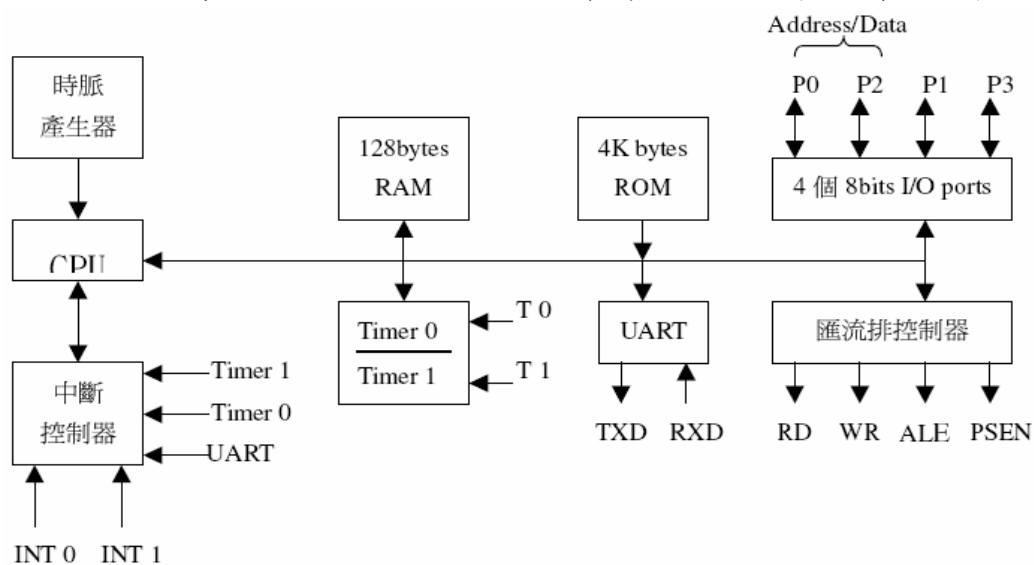


圖 2-2-1 單晶片內部構造

8051 主要功能列舉如下：

- ☐ 為一般控制應用的8位元單晶片。
- ☐ 晶片內部具時脈振盪器（傳統最高工作頻率可至12MHz）。
- ☐ 內部程式記憶體（ROM）為4K位元組。
- ☐ 內部資料記憶體（RAM）為128位元組。
- ☐ 外部程式記憶體可擴充至64K位元組。
- ☐ 外部資料記憶體可擴充至64K位元組。
- ☐ 32條雙向輸入輸出線，且每條均可以單獨做I/O的控制。
- ☐ 5個中斷向量源。
- ☐ 2組獨立的16位元定時器。
- ☐ 1個全多工串列通信埠。
- ☐ 8751及8752單晶片具有資料保密的功能
- ☐ 單晶片提供位元邏輯運算指令

## (2)單晶片微電腦

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其中，中央處理單元則是由



圖 2-2-2 單晶片

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
RST	9	1	32	P0.7/AD7
RXD/P3.0	10		31	$\overline{EA}$
TXD/P3.1	11	單	30	ALE
INT0/P3.2	12	晶	29	$\overline{PSEN}$
INT1/P3.3	13	片	28	P2.7/A15
T0/P3.4	14		27	P2.6/A14
T1/P3.5	15		26	P2.5/A13
WR/P3.6	16		25	P2.4/A12
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 2-2-3 單晶片電路圖

## 三、零件故障識別

有無燒糊、燒斷、起泡、板面斷線、插口鏽蝕。表測法：+5V、GND 由於這些電路的固有頻率不同，波形不同，所以所測的數據是近似值，只能供參考。總電流測量法該法是通過檢測 ic 電源進線的總電流，來判 ic 好壞的一種方法。由於 ic 內部絕大多數為直接耦合，ic 損壞時（如某一個 pn 結擊穿或開路）會引起後級飽和與截止，使總電流發生變化。所以通過測量總電流的方法可以判 ic 的好壞。也可用測量電源通路中電阻的電壓降，用歐姆定律計算出總電流值。

由 85C51 中撰寫程式碼使，電路中送出藍芽連線模組，經由手機連線，使藍芽接收產生作用。

基本電路連接如下圖

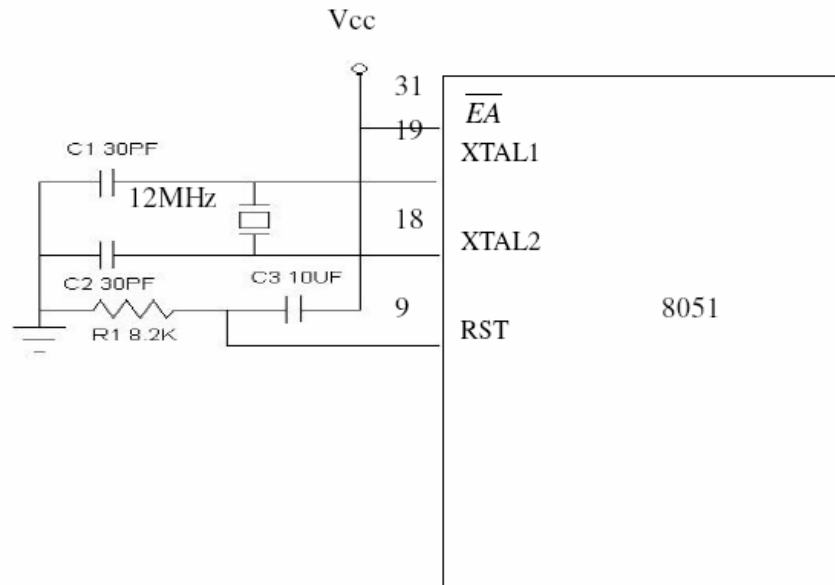


圖2-2-4 基本電路連接圖

### 8051的記憶體結構

8051的記憶體分為程式記憶體及資料記憶體，所以程式與資料是分開存放的，又可在分為內部及外部，下圖為8051記憶體結構示意圖：

8051記憶體結構示意圖：

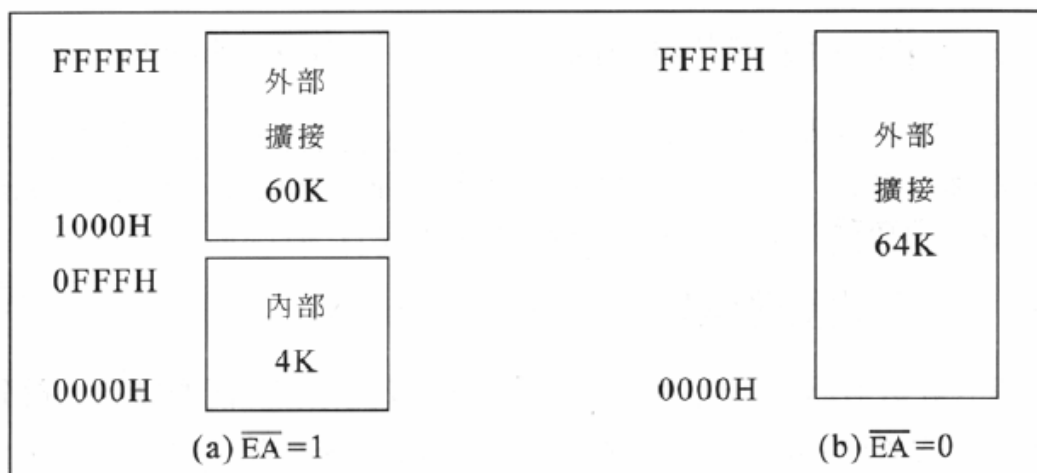


圖2-2-5 8051記憶體結構示意圖

### 程式記憶體(ROM)

程式記憶體主要是存放程式碼，CPU 會自動到程式記憶體提取程式碼，並依照程式內容執行工作，所以CPU 無法存入或更新程式記憶體的資料。

8051 的程式記憶體可以分成內部(4K)及外部(64K)，利用 EA 接腳來做設定，若 EA 接高電位則表示使用內部程式記憶體，反之，EA 接低電位則表示使用外部記憶體。

程式記憶體用來存放 8051 程式，可使用內部的 4K 位元組（EA 接腳為高電位）若使用者在做內部記憶體的讀取時，程式區超 4Kbytes 後，CPU 自動會送出 Low 的訊號至 PSEN 接腳進而繼續至外部 ROM 提取程式指令使用外部的 64K 位元組則將 EA 接腳，接在寫 8051 的程式時，必須知道幾個程式記憶體的特別位址，這些位址是各種中斷服務程式的進入點下表列出了各種中斷的進入點位址，其中位址 0000H 是重置(RESET)的進入點，這意思是說，8051 被重置時，從位址 0000H 開始執行程式。

中 斷 源	向 量 位 址
RESET	0000h
TNT0	0003h
Timer0	000Bh
INT1	000Bh
Timer1	001Bh
UART	0023h
Timer2	002Bh

圖2-2-6 程式記憶體(ROM)

### 資料記憶體(RAM)

#### ◎內部資料記憶體

8051 內部有一塊 256 個 byte 的位址空間，這塊空間是存放資料記憶體(RAM)和特殊功能暫存器(SFR)的地方。這塊記憶體空間雖然只有 256byte，但是 8051 將其中位指數高的 128byte(80H~FFH)採用 不同的定址方式而容納了兩組 128byte 的記憶體空間，因此總共的空間為  $128+128+128=384$  byte。



以下三個部分開加以解說：

1. 位址 00H~7FH 的RAM
2. 位址 80H~FFH 的RAM
3. 位址 80H~FFH 的 SFR
4. 位址 00H~7FH 的 RAM：

此 128 位元組可以使用直接或間接定址法的方式來存取其內部的資料，以下圖來說明，此位址可以分為以下三部分：

A.暫存器庫 00H~1FH

B.可用位元定址區 20H~2FH

C.一般用途空間 30H~7FH

7FH	一般資料存放區或堆疊區							
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00
1FH	RB3 ( 8 bytes )							
	RB2 ( 8 bytes )							
	RB1 ( 8 bytes )							
00H	RB0 ( 8 bytes )							

圖2-2-7 程式記憶體(RAM)

### 位址80H~FFH的RAM：

只有8052, 8752 和8032 的內部RAM 有這128byte, 8031, 8051 和8751 則沒有。這塊RAM 的內容必須使用間接定址法。

### 位址80H~FFH的SFR：

特殊功能暫存器是一塊 128byte 的記憶空間，它是存放 8051 內部的週邊所使用的暫存器的地方，例如 I/O port 的輸出栓鎖器(P0, P1, P2, P3)，計時器的 counter，致能中斷系統的 IE 暫存器等。因為 8051 的週邊設備並不多，因此SFR 裡128 個位址空間並未用完，這些目前沒有用到的位址，裡面是空的。SFR 所使用個位址是 80H~FFH，這塊區域與8051 的較高128 位元組的RAM 使用了同一塊記憶空間8051 採用了不同的指令的定址法來區分這兩塊記憶體，如前面所述，RAM 是使用間接定址法，SFR 是使用直接定址法。在 SFR 裡的各種位元組都有其個別的名稱，在寫程位元組，可直接呼叫其名稱，而不需要使用位址。

### 特殊功能暫存器( Special Function Register, SFR )

#### (1) 累加器( Accumulator, ACC )：

累加器又可稱之為 ACC 或 A 暫存器，這是一個使用頻率頗高的一個通用暫存器，而有許多指令是以其為操作對象。

#### (2) P0、P1、P2、P3埠暫存器：

這四個埠暫存器可存放 8051 單晶片的 4 個 I/O 埠的輸出門鎖 (Latch)，主要是存放並保持I/O 的輸出資料。

#### (3) TH0~TH2、TL0~TL2計時器/計數時暫存器：

這 3 組 16 位元的暫存器是分別用來儲存計時器/計數器的計時/計數值。TH0、TH1、TH2 為高位元組，TL0、TL1、TL2 為低位元組。TH0 及 TL0 對應於計時器/計數器 0，TH1 及 TL1

對應於計時器/計數器 1，TH2 及 TL2 對應於計時器/計數器 2(8052 系列)。

TH0及TL0對應於計時器/計數器0，TH1及TL1

對應於計時器/計數器1，TH2及TL2對應於計時器/計數器2(8052系列)。

#### (4)計時器模式控制(Timer/Counter Mode Control, TMOD)暫存器：

位元	7	6	5	4	3	2	1	0
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0
計時器1					計時器0			

圖2-2-8 計時器說明圖

GATE：計時器動作閘控位元，當GATE=1時，INT0或INT1接腳為高電位，同時 TCON中的 TR0或 TR1控制位元為1時，計時/計數器 0或 1才會動作。若 GATE=0，則只要將TR0或TR1控制位元設為1，計時/計數器0或1即可動作。

C / T：做計時器或計數器功能之選擇位元。C/T=1為計數器，由外部接腳T0

或T1 輸入計數脈波。C/T=0為計時器，由內部系統時脈提供計時工作脈波。

M1：模式選擇位元1。

M0：模式選擇位元0。

M1	M0	工作模式
0	0	0 13-bit 計時器
0	1	1 16-bit 計時器/計數器
1	0	2 8-bit 自動載入
1	1	3 (Timer 0)TL0 為 8 位元計時器/計數器，由標準之計時器 0 之控制位元控制，TH0 為 8 位元計時器，且由計時器 1 控制位元控制。
1	1	3 (Timer 1)計時器/計數器 1 停止。

圖2-2-9 工作模式

(5) 計時器控制( Timer Control，TCON )暫存器：

位元	7	6	5	4	3	2	1	0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

圖2-2-10 計時器控制( Timer Control，TCON )暫存器

TF1(TCON.7)：計時器 1 溢位旗號，當計時溢位時，由硬體設定為 1，在執行過相對的中斷服務常式後則自動清除為 0。

TR1(TCON.6)：計時器 1 啟動控制位元，可以由軟體來設定或清除。

TF0(TCON.5)：計時器 0 溢位旗號，當計時溢位時，由硬體設定為 1，在執行過相對的中斷服務常式後則自動清除為 0。

TR0(TCON.4)：計時器 0 啟動控制位元，可以由軟體來設定或清除。

IE1(TCON.3)：外部中斷 1 動作旗號，當外部中斷被偵測出來時，硬體自動設定此位元，在執行過中斷服務常式後，則消除為 0。

IT1(TCON.2)：外部中斷 1 動作型態選擇，當 IT1=1 時，中斷型態為負緣觸發，當 IT1=0 時，中斷型態則為低準位觸發。

IE0(TCON.1)：外部中斷 0 動作旗號，當外部中斷被偵測出來時，硬體自動設定此位元，在執行過中斷服務常式後，則消除為 0。

IT0(TCON.0)：外部中斷 0 動作型態選擇，當 IT1=1 時，中斷型態負緣觸發，當 IT1=0 時，中斷型態則為低準位觸發。

(6) 串列埠控制( Serial Port Control )暫存器：

位元	7	6	5	4	3	2	1	0
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

圖2-2-11 串列埠控制( Serial Port Control )暫存器

SM0(SCON.7)：串列傳輸模式選擇，共有 4 種模式。SM1(SCON.6)：串列

傳輸模式選擇，共有4種模式。SM2(SCON.5)：在串列傳輸動作模式2或模式3時，作多處處機控制功能用。

REN(SCON.4)：串列介面接收位元，當REN=1時表示接收致能。

TB8(SCON.3)：在模式2或3時，所送出的第9個資料位元，可以由軟體指令來做控制設定或清除。

RB8(SCON.2)：在模式2或3時，所接收到的第9個資料位元，存放在此位元中。

TI(SCON.1)：串列資料傳送中斷旗號，在工作模式0時，送出8個資料位元後，TI 設為 1，而在其他模式時，在送出停止位元時，TI 也會被設為 1；此位元必須由軟體來清除。

RI(SCON.0)：串列資料接收中斷旗號，在工作模式0時，收到第8個串列輸入資料位元後，RI會設為 1，在其他模式時，收到停止位元的一半時，硬體會自動將此位元設為1。此位元必須由軟體來清除。

SM0	SM1	模式	說明	鮑率
0	0	0	移位暫存器	Fosc./12
0	1	1	8-bit UATR	可變
1	0	2	9-bit UATR	Fosc./64 或 Fosc./32
1	1	3	9-bit UATR	可變

圖2-2-12 串列資料緩衝( Serial Data Buffer, SBUF )暫存器

(7)串列資料緩衝( Serial Data Buffer, SBUF )暫存器：

8051 單晶片的串列埠是全雙工的，故實際上SBUF 暫存器分開為兩個不同的暫存器，一個是當作UART 傳送資料的緩衝區，另一個是當作UART接收資料的緩衝區。若將資料寫到SBUF 時，就會將資料放入傳送緩衝區，UART就會將TXD 傳出去。若去讀SBUF，就會讀到接收緩衝區的資料。

下圖為SFR的記憶體分佈圖：

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW							D7
C8								CF
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90	P1							97
88	TCON	TMOD	TLO	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH			PCON	87

此行記憶體位元組可做位元定址。

圖2-2-13 SFR記憶體分佈圖

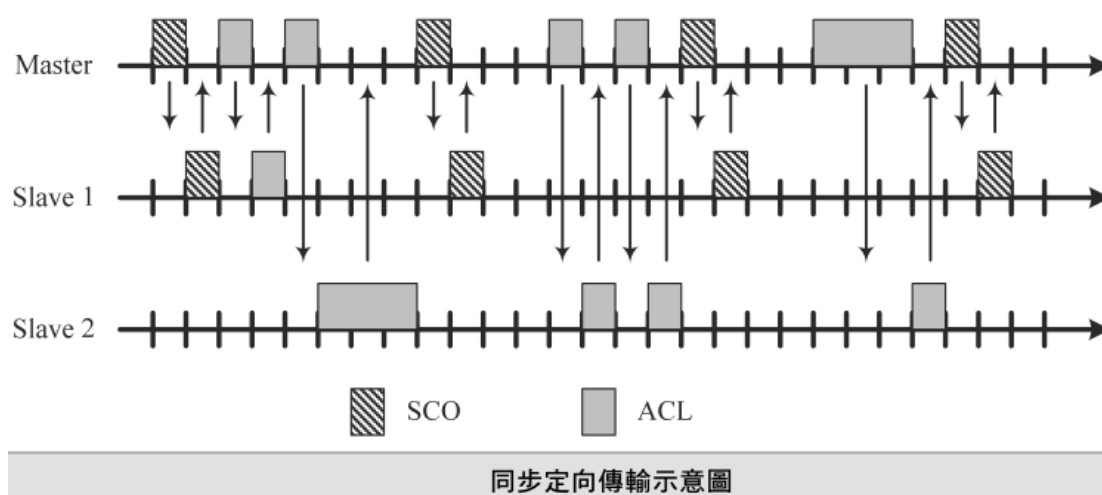


圖2-2-14 同步定向傳輸示意圖

## 參、專題製作

本章分成三個階段由說明此專題所應用到的設備和器材及製作方法與步驟專題製作等等。

### 一、設備及材料

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
麵包版	模擬電路
數位相機	拍攝專題功能
三用電錶	檢測
PC 電腦	編譯程式
燒錄器	程式燒錄 89C51
Word	書面報告
Powerpoint	上台報告之簡報
程式編譯器	編好之軟體除錯即轉燒錄檔
Protel 99SE	繪畫電路圖

### 二、製作方法和製作步驟

這次專題利用的是行動並且研究，由循環的反覆研究及製作過程所構成，包含準備零件、實驗模擬、電路資料查詢及報告篇寫等階段。此研究之製作方法與製作步驟。

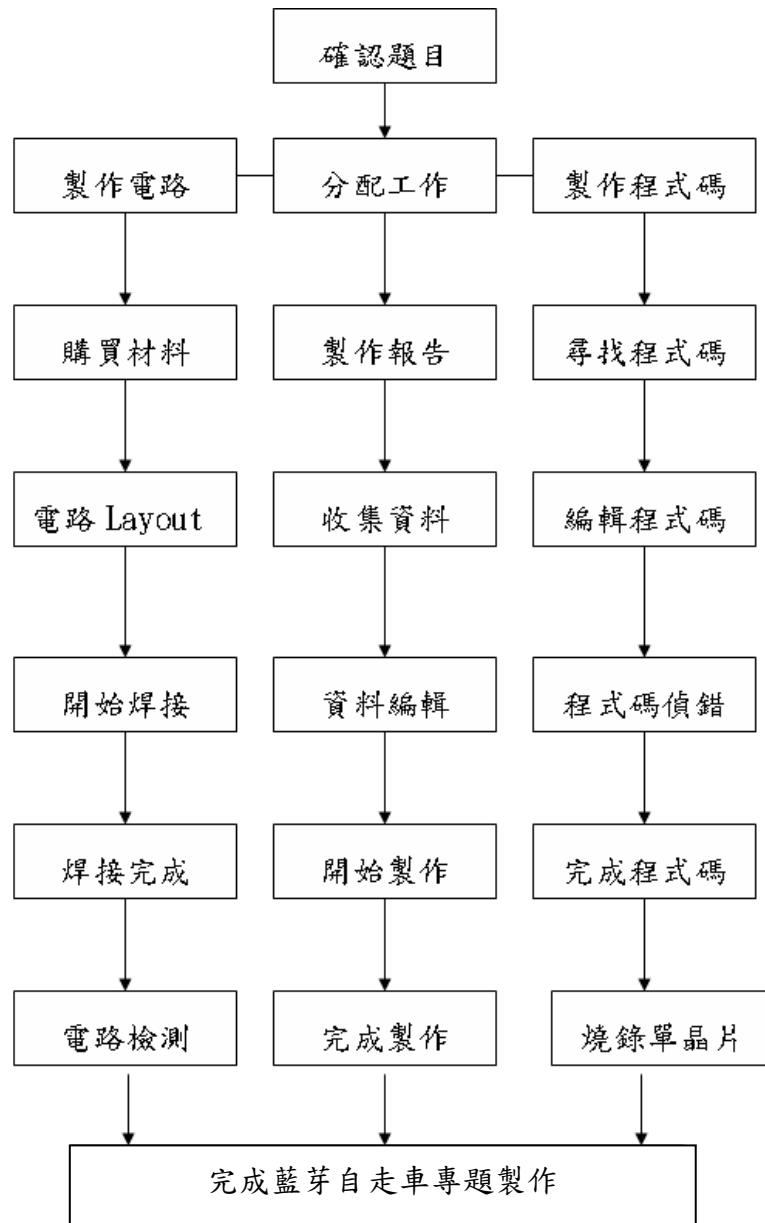


圖 3-2-1 製作方法及步驟

### 三、專題製作

表 3-3-1 專題製作計畫書

專題型別		<input type="checkbox"/> 個人型專題 <input checked="" type="checkbox"/> 團隊型專題	
專題性質		單晶片控制研究	
科別／年級		電子科三年級	
專 題 名 稱	中文名稱	藍芽自走車	
	英文名稱	Bluetooth self-propelled vehicle	
專題內容簡述		<p>透過 89C51 單晶片學習控制硬體了解單晶片的使用、以及 89C51 運用設計一個無人操控的專題，我們選擇了運用藍芽連線模組作一感測的方式，有別於紅外線或超音波感測使自走車應用在生活中、使生活更加便利，學到新知識。</p> <p>由單晶片判讀並控制自走車前進、後退、左轉、右轉。希望能夠利用自走車上的藍芽模組與智慧型手機進行配對，來設計出更加方便，更好操控的自走車。</p>	
指導老師姓名		葉忠賢 老師	
參與同學姓名		羅建甫組長	蔡政翰組員
		何恩富組員	
專題執行日期		103 年 9 月 1 日至 104 年 5 月 31 日	



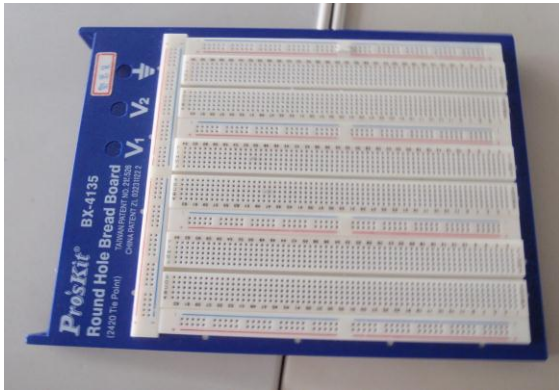


圖 3-3-1 麵包板上電路模擬(一)

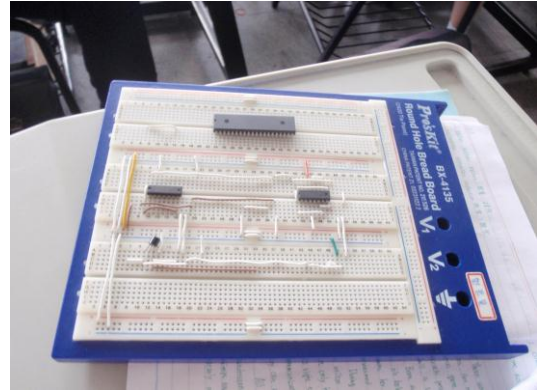


圖 3-3-2 麵包板上電路模擬(二)



圖 3-3-3 製作 Power Point 簡報



圖 3-3-4 製作 Word 報告書

### (一)藍芽自走車功能及設定

透過這個專題製作，去了解如何運用單晶片連接藍芽自動控制上，當在使用藍芽接收時，能進行無人感測，且經由透過程式碼，來更改行走方向，使藍芽的表示方法：剛開啟開關送電時，藍芽會接收訊號，等訊號接收到時，然而進而操作。

## (二) 硬體電路圖：藍芽自走車

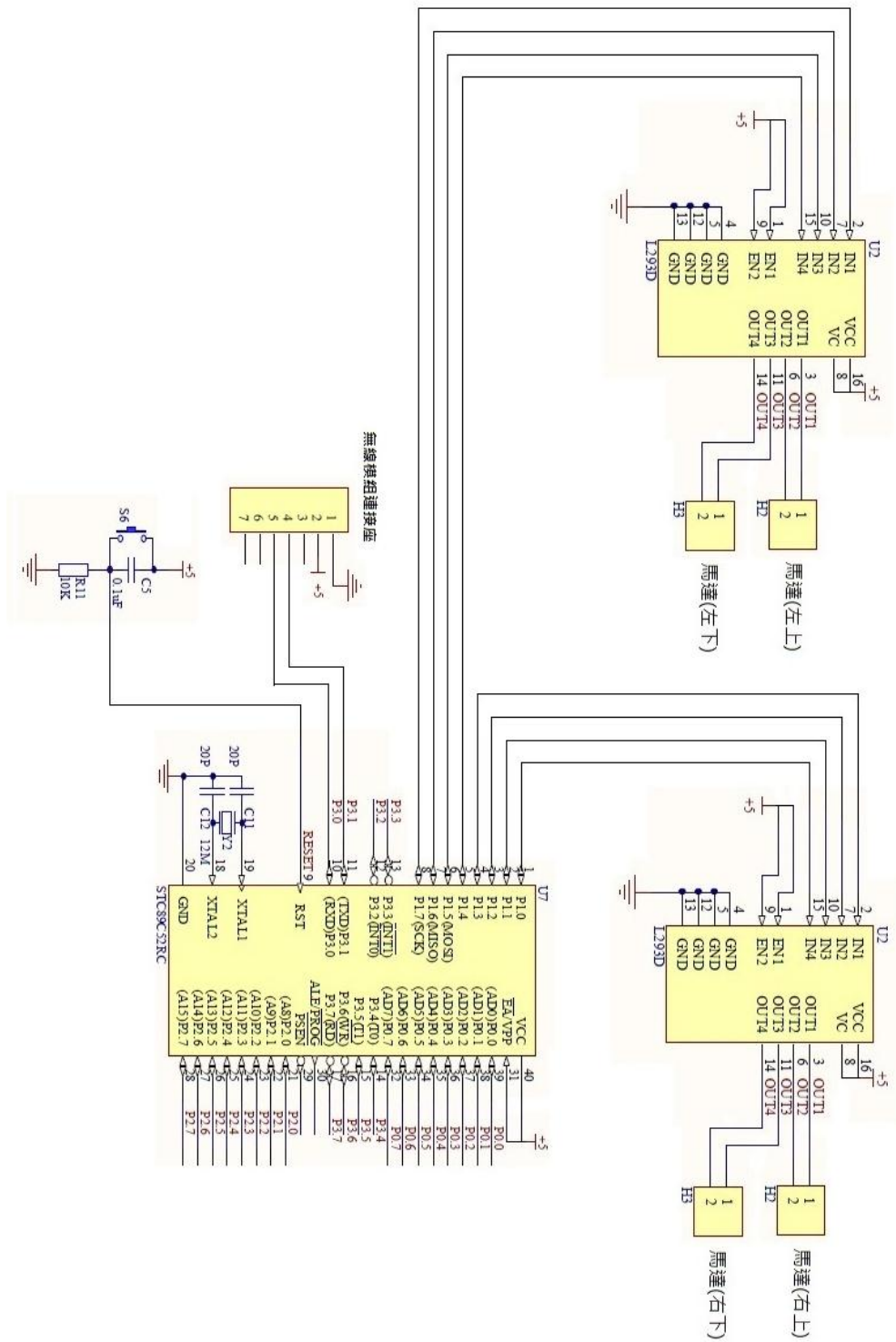


圖 3-3-5 藍芽自走車電路圖

表3-3-2 藍芽自走車之材料表

項目	品名	規格	數量
1	電阻	330 ohm, 1/4W	1
2	電阻	10K ohm, 1/4W	1
3	電解電容	10u/50V	1
4	陶瓷電容	22p, pitch=2.5mm	2
5	單晶片	89C51(含程式)	1
6	發光二極體	藍色, 5mm	1
7	DIP IC 座	16Pin, 600mil	1
8	DIP IC 座	40Pin, 600mil	1
9	開關	2 段 6Pin	2
10	驅動 IC	L293D	1
11	XTAL	11.0592M Hz	1
12	PCB	洞洞板	1
13	馬達組	3V 含減速齒輪	2
14	螺帽	3mm 用	12
15	銅柱		8
16	輔助滾輪		1
17	單排針座	2Pin	2
18	單排針座	3Pin	1
19	輪子		2
20	藍芽模組	ZK-07	1

在高雄的各電子材料行都難以買的到這些東西，經由我們小組討論後由鑑輔和恩赴去長民街買材料結果都要等一個禮拜，一個禮拜後買完後就開始製做我們這組的專題製作，藍芽在連線時是發出肉眼所看不到的東西、發射出去後由手機接收到訊號就能達成你所要的目的了。

小組分工的配置：

建甫負責找資料，及製作簡報內容，過程中亦有購買相關書籍當成參考資料，選擇要如何去製作專題，讓專題，讓恩富和政翰知道要作何專題，然後再經過小組討論、商量，有問題時，會再去徵詢老師的意見。

是要上網查有關專題相關的資料，負責把相關的資料和圖片下載下來，在去製作文書軟體得作業，如有發生困難時可以找小組討論或者找老師諮詢，解決你的所不懂的是事情。

建甫和恩富負責去買電路中所需之零件，來製作我們這組的專題報告書政翰負責把電路畫出來，在讓恩富看著電路圖去和接我們的專題製作的成品，建甫負責製作簡報讓政翰和恩富上台發表此專題內容。

## 肆、製作成果

我們小組決定題目藍芽自走車，開始製作並且模擬麵包板、設計Layout圖，進而完成焊接且製作整個完整電路；這整個流程中，我們小組都用相機及相關電腦設備將之拍照下來，經將這些資料完整處理過後，我們呈現在這個專題報告之中，如下所示：

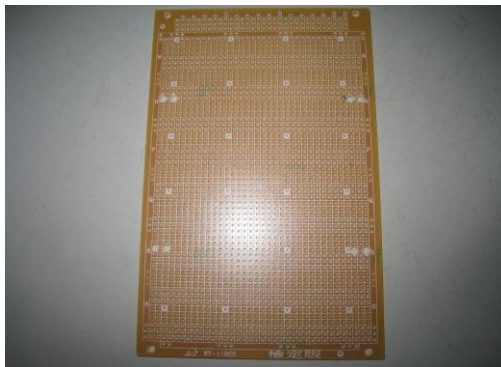


圖4-1-1藍芽自走車製作過程(一)

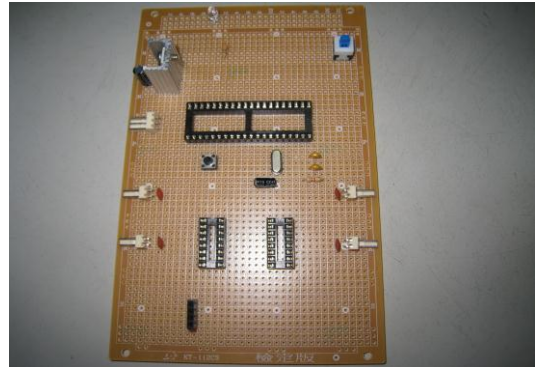


圖4-1-2紅外線感測開關製作過程(二)

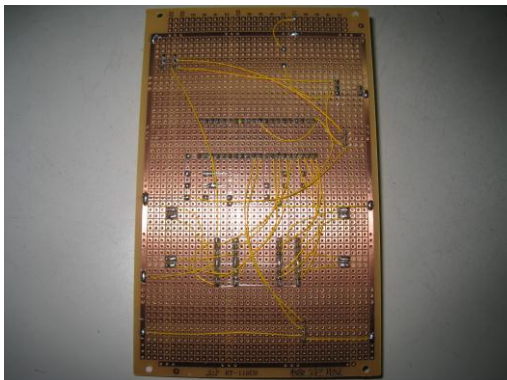


圖4-1-3藍芽自走車製作過程(三)

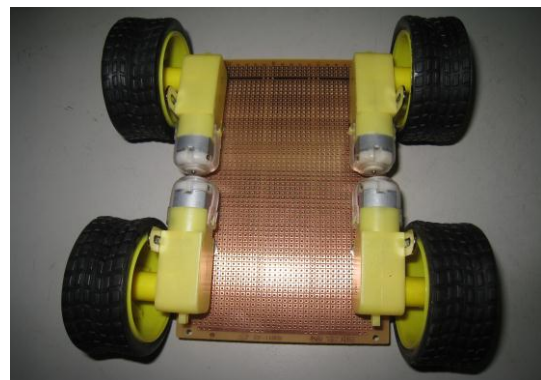


圖4-1-4藍芽自走車製作過程(四)

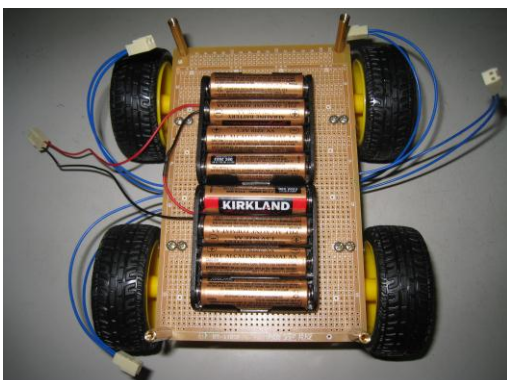


圖4-1-5藍芽自走車製作過程(五)

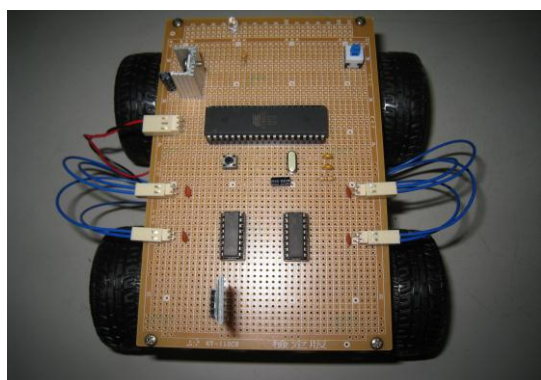


圖4-1-6完成藍芽自走車專題



目前為止，我們所製作的電路實作部分，面對我們所遭遇的困難以及問題，都已克服了，緊接著，我們的下一步驟即是將我們開始製作外面包裝的成品製作，將之接到我們的，由於有牽涉到配置的問題，所以待我們將之完成後，會再以照片及實體的方式慢慢呈現之。

藍芽自走車理論操作：

- (1) 透過藍芽發送接收器，了解感測時間。
- (2) 透過89C51了解接腳該如何運作；程式如何進行更改的方法。

### 一、操作方法及步驟

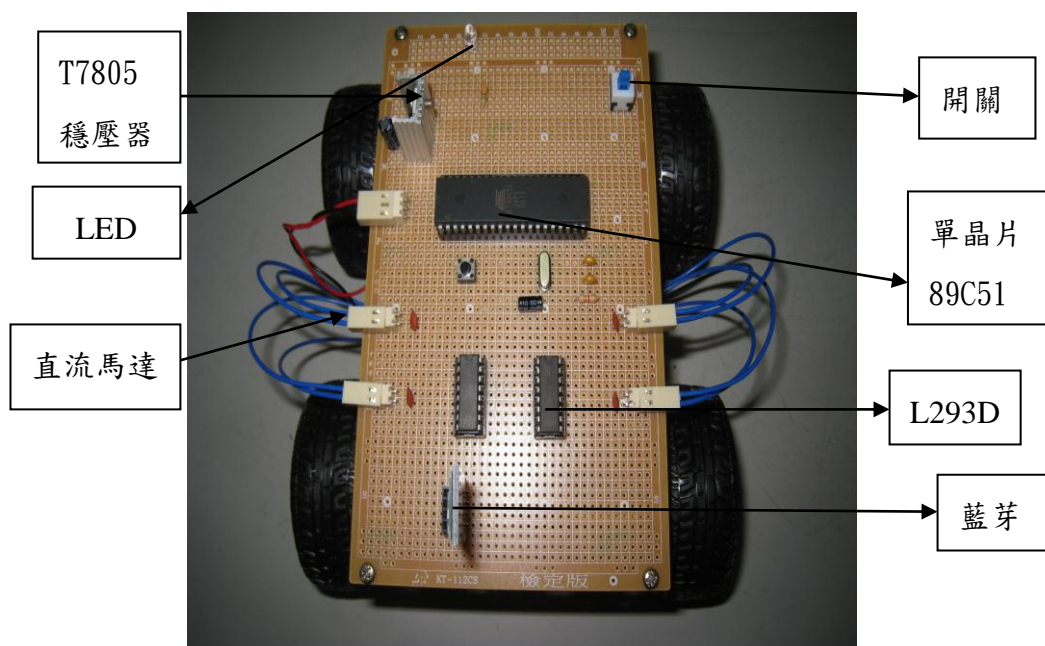


圖4-1-7 藍芽遙控自走車說明圖

本專題的內容由89C51為基礎的做電路架構，其中功能、設定：

#### 1. 藍芽自走車開關設定：

- (1) 打開開關時，LED就會亮燈。
- (2) 接下來打開手機勾選藍芽。
- (2) 與手機上的藍芽連接，之後輸入密碼1234。
- (3) 配對成功後，打開手機程式ZK-Remotecar。
- (4) 進行最後一次配對點擊右上角的藍芽連接。
- (5) 點選HC-06(每支手機的名稱不一定相同)。
- (6) 配對成功後，可用手機遙控自走車前進。

(7)25頁為完整圖片說明。

## 二、下圖為手機操作介面說明

我們以中國大陸的國產華為手機U8650為例為大家講解如何與藍牙模組連接控制小車，不同手機操作大同小異。

先安裝軟體，之後如下圖所示

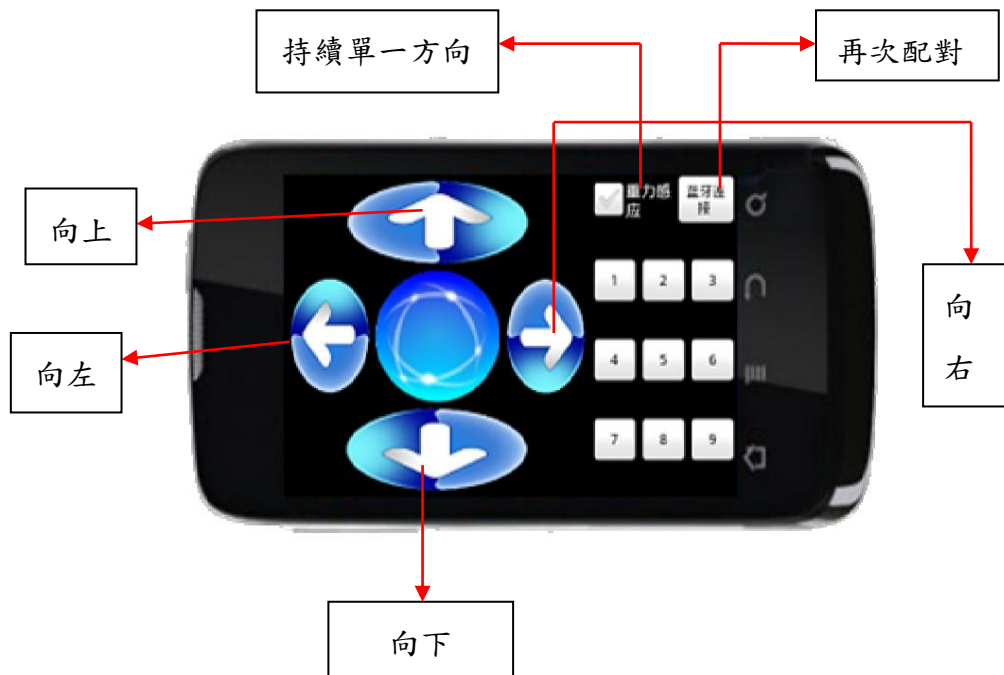


圖4-1-8 藍芽遙控自走車手機說明

手機上面的數字鍵為下文字敘述：

按键1 发送 ASCII 按下发ON1 弹起来ONa  
按键2 发送 ASCII 按下发ON2 弹起来ONb  
按键3 发送 ASCII 按下发ON3 弹起来ONc  
按键4 发送 ASCII 按下发ON4 弹起来ONd  
按键5 发送 ASCII 按下发ON5 弹起来ONe  
按键6 发送 ASCII 按下发ON6 弹起来ONf  
按键7 发送 ASCII 按下发ON7 弹起来ONg  
按键8 发送 ASCII 按下发ON8 弹起来ONh  
按键9 发送 ASCII 按下发ON9 弹起来ONi

### 三、下圖為手機操作步驟



圖4-1-9開啟藍芽介面(一)



圖4-1-10勾選藍芽(二)



圖4-1-11點選HC-06(三)



圖4-1-12輸入密碼1234(四)





圖4-1-13打開ZK-Remotecar (五)



圖4-1-14最後點選藍芽連接按HC-06 (六)

藍芽自走車開關設定：按下HC-06後，藍芽會呈現出不斷閃爍的狀態，表示未連接到，等到閃爍停止後，即可操作。

## 伍、結論與建議

### 一、結論

本次製作專題時，小組常常出現毫無方向，不知道接下來該怎麼辦的情形常常出現，例如：資料查詢、電路參考、該從何找尋論點之類的，這些都要一一突破，因為我們主要是缺乏找尋適合我們的電路程式等等。

1. 學校裡所學的皆以理論為主，將理論結合電路應用在日常生活中將有效提升學習樂趣。
2. 藍芽可以控制的實際距離能到50m公尺以內。
3. 專題製作可培養團隊合作的精神，因彼此都會有自己的意見，學習如何去合作把意見統一。
4. 透過此次專題製作學習的方式更能幫助我們獲得知識的建構及整合，且亦可以幫助我們提升其創造思考的能力。
5. 專業知識的不足是一般專題製作過程中常遇到的事，此時老師所扮演的角色就相對重要。
6. 專題的製作可以帶來很大的成就感，當完成專題時的那份喜悅只有認真負責的人能感受的到。
7. 小組同學認同資訊科技的知識在專題製作學習過程中，扮演著重要的角色，因其對電路製作、資料呈現及成果報告製作是很有幫助的。
8. 專題製作學習可以培養我們學習者具備問題解決、研究、反省、團體合作及應用資訊科技等多項能力。

### 二、建議

在學習過程中給予回饋：同學建議，在專題製作學習研究過程中，老師能否可以在學習的進行過程，給予立即性的回饋，讓學生可以及早發現其缺失，盡早進行改善。

1. 在操作藍芽自走車時，要注意行走方向，避免撞毀。
2. 此專題仍然還需要用到電池，並沒有達到環保的目的，期待下次能使用風力轉換成電力來行走，以達到環保。
3. 隔著一層地板不能連線，因為藍芽是平行傳輸，因此不能連接。
4. 改善的方法是把藍芽連接波幅變高或許能夠解決上述問題。

## 參考文獻

- 1.楊明峰，2008，8051 入門輕鬆學，台北市：碁峰資訊。
- 2.鍾富昭，2003，8051 專題製作，台北市：全華文化。
- 3.陳俊榮，2007，組合語言，台北市：全華文化。
- 4.林豐隆，2007，專題製作，台北市：全華文化。
- 5.傅榮鈞・林偉政，專題製作8051單晶片篇，台北縣：台科大圖書。
- 6.徐椿樑・陳輔賢，2004，8051/8951理論與實物應用，台北市：全華文化。
- 7.蔡朝洋單晶片微電腦8051 / 8951原理與應用。台北市：全華科技圖書股份有限公司。
- 8.郭庭吉、吳金戌（2008），單晶片8051 專題製作-使用keil AX51。台北市：松崗電腦圖書有限公司。
- 9.葉文聰、WonDerSun（2008）。程式設計應用電子篇。台北市：台科大圖書股份有限公司。
- 10.許桂樹、范逸之Visual Basic-網路監控實務。台北縣：新文京開發出版股份有限公司。

## 附錄一 藍芽自走車程式碼

/\*\*\*\*\*\*

\*\*\*\*\*

藍芽遙控車

插入藍牙模塊：

晶振：11.0592M 晶振

\*\*\*\*\*

\*\*\*\*\*/

#include<AT89x51.H>

#define Left\_moto\_go {P1\_0=1,P1\_1=0,P1\_2=1,P1\_3=0;} //左邊馬達

向前走

#define Left\_moto\_back {P1\_0=0,P1\_1=1,P1\_2=0,P1\_3=1;} //左邊馬達

向後轉

#define Left\_moto\_Stop {P1\_0=0,P1\_1=0,P1\_2=0,P1\_3=0;} //左邊馬達

停轉

#define Right\_moto\_go {P1\_4=1,P1\_5=0,P1\_6=1,P1\_7=0;} //右邊馬達

向前走

#define Right\_moto\_back {P1\_4=0,P1\_5=1,P1\_6=0,P1\_7=1;} //右邊馬達

向前走

#define Right\_moto\_Stop {P1\_4=0,P1\_5=0,P1\_6=0,P1\_7=0;} //右邊馬達

停轉

#define left 'C'

#define right 'D'

#define up 'A'

#define down 'B'

```

#define stop      'F'

char code str[] = "收到指令，向前!\n";
char code str1[] = "收到指令，向後!\n";
char code str2[] = "收到指令，向左!\n";
char code str3[] = "收到指令，向右!\n";
char code str4[] = "收到指令，停止!\n";

bit  flag_REC=0;
bit  flag      =0;

unsigned char  i=0;
unsigned char  dat=0;
unsigned char  buff[5]=0; //接收緩衝字節

/*****
**/
//延時函數
void delay(unsigned int k)
{
    unsigned int x,y;
    for(x=0;x<k;x++)
        for(y=0;y<2000;y++);
}

/*****
**/
//字符串發送函數
void send_str( )

```

```

        // 傳送字串
    {
        unsigned char i = 0;
        while(str[i] != '\0')
        {
            SBUF = str[i];
            while(!TI);          // 等待數據傳送
            TI = 0;              // 清除數據傳送標誌
            i++;                 // 下一個字符
        }
    }

```

```

void send_str1()
    // 傳送字串
{
    unsigned char i = 0;
    while(str1[i] != '\0')
    {
        SBUF = str1[i];
        while(!TI);          // 等待數據傳送
        TI = 0;              // 清除數據傳送標誌
        i++;                 // 下一個字符
    }
}

```

```

void send_str2()
    // 傳送字串
{
    unsigned char i = 0;
    while(str2[i] != '\0')

```

```

{
    SBUF = str2[i];
    while(!TI);          // 等待數據傳送
    TI = 0;               // 清除數據傳送標誌
    i++;                  // 下一個字符
}
}

```

```

void send_str3()
    // 傳送字串
{
    unsigned char i = 0;
    while(str3[i] != '\0')
    {
        SBUF = str3[i];
        while(!TI);      // 等待數據傳送
        TI = 0;          // 清除數據傳送標誌
        i++;              // 下一個字符
    }
}

```

```

void send_str4()
    // 傳送字串
{
    unsigned char i = 0;
    while(str4[i] != '\0')
    {
        SBUF = str4[i];
        while(!TI);      // 等待數據傳送
        TI = 0;          // 清除數據傳送標誌
    }
}

```

```

        i++;                // 下一個字符
    }
}

/*****
**/

//前速前進
void run(void)
{
    Left_moto_go;  //左馬達往前走
    Right_moto_go; //右馬達往前走
}

//前速後退
void backrun(void)
{
    Left_moto_back;  //左馬達往前走
    Right_moto_back; //右馬達往前走
}

//左轉
void leftrun(void)
{
    Left_moto_back;  //左馬達往前走
    Right_moto_go;   //右馬達往前走
}

//右轉
void rightrun(void)
{
    Left_moto_go;    //左馬達往前走
    Right_moto_back; //右馬達往前走
}

```



```

}
//STOP

void stoprun(void)
{

    Left_moto_Stop;    //左馬達往前走
    Right_moto_Stop;   //右馬達往前走
}

/*****
**/

void sint() interrupt 4    //中斷接收 3 個字節
{
    if(RI)                //是否接收中斷
    {
        RI=0;
        dat=SBUF;
        if(dat=='O'&&(i==0)) //接收數據第一幀
        {
            buff[i]=dat;
            flag=1;          //開始接收數據
        }
        else
        if(flag==1)
        {
            i++;
            buff[i]=dat;
            if(i>=2)
            {i=0;flag=0;flag_REC=1 ;} // 停止接收
        }
    }
}

```

```

}

/*****

/*--主函數--*/

void main(void)
{
    TMOD=0x20;
    TH1=0xFd;          //11.0592M 晶振，9600 波特率
    TL1=0xFd;
    SCON=0x50;
    PCON=0x00;
    TR1=1;
    ES=1;
    EA=1;
    while(1)           /*無限循環*/
    {
        if(flag_REC==1) //
        {
            flag_REC=0;
            if(buff[0]=='O' && buff[1]=='N') //第一個字節為 O，第二個字節為 N，第
三個字節為控制碼
                switch(buff[2])
                {
                    case up :           // 前進
                        send_str( );
                        run();
                        break;
                    case down :         // 後退
                        send_str1( );
                        backrun();

```

```
break;
case left :                               // 左轉
    send_str2( );
    leftrun();
break;
case right :                             // 右轉
    send_str3( );
    rightrun();
break;
case stop :                               // 停止
    send_str4( );
    stoprun();
break;
```