

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



LED 旋轉廣告顯示燈

學生姓名： 呂 宗 翰

曾 柏 霖

洪 啓 原

洪 偉 晟

指導老師： 葉 忠 賢 老師

中 華 民 國 104 年 05 月

## 誌 謝

在高英工商三年來吸收到不少專業知識與技能，每當碰到困難，主任、老師們總會為大家解決難題，並讓我們了解其中的特性。使我們學到了不少，知識上的增加也獲得不少，對自我的肯定又更上了一層樓！感謝老師每次在我們遇到難題時，總是告訴我們這句話：（試了也許不一定會成功，但不試就完全不會成功）。

此次專題我們學習到組員們團隊合作的重要性，有了團隊合作才能使專題更完整的呈現。實作中用了不少材料、工具，讓我們認識零件與它的特性。感謝老師協助建議，製作上的難題與錯誤，也有老師為我們協助減少我們的困難與阻礙，使我們更有信心的製作，謝謝老師。

呂宗翰、曾柏霖、洪啟原、洪偉晟謹上 2015/05

# LED 旋轉廣告顯示燈

## 摘 要

現今生活中有許多產品都是以視覺暫留的特性來完成的，例如：小綠人、電子時鐘、電子廣告看板等。為了探討視覺暫留的概念，於是我們利用自身所學的 LED 應用技術、程式語言、電路繪製等，配合專題課程加以研究，製作簡易型的視覺暫留器，模擬市面上精緻的視覺暫留所呈現出來的效果，並經我們實際操作後深入了解其中的原理以及相關概念，以達到此次專題的研究目的。此專題製作主要利用視覺暫留原理的特性產生圖像，其原理類似於電影一般的連續畫面。我們將撰寫好的程式輸入至 STC12C5A32S2 內，搭配一排 LED 在不同的時間閃爍不同的 LED 燈，加上馬達的高速旋轉，產生對人眼的視覺暫留，顯示出我們所要求的畫面。

**關鍵詞：**LED、STC12C5A32S2、高速旋轉、視覺暫留

# 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 .....	2
四、製作預期成效 .....	3
貳、理論探討 .....	4
參、專題製作 .....	9
一、設備及器材 .....	9
二、製作方法與步驟 .....	9
三、專題製作 .....	12
四、旋轉 LED 改字軟體-平面立體版 .....	13
肆、製作成果 .....	20
伍、結論與建議 .....	22
一、結論 .....	22
二、建議 .....	22
參考文獻 .....	23
附錄一 LED 旋轉廣告顯示燈程式碼 .....	24

## 表目錄

表 3-1-1 專題製作流程圖 .....	15
表 3-3-2 LED 廣告顯示燈之材料表.....	22

## 圖目錄

圖 1-3-1 專題製作流程圖 .....	2
圖 2-1-1 DS1302 時鐘芯片 .....	5
圖 2-1-2 DS1302 時鐘芯片電路 .....	5
圖 2-1-3 32.768KHZ 晶體振盪器 .....	5
圖 2-1-4 32.768KHZ 晶體振盪器 .....	6
圖 2-1-5 單晶片圖 .....	6
圖 2-1-6 單晶片電路圖 .....	6
圖 2-1-7 LED 視覺暫留原理圖 .....	7
圖 2-1-8 LED 視覺暫留原理圖 .....	7
圖 2-2-1 正面俯視圖 .....	8
圖 2-2-2 背面俯視圖 .....	8
圖 2-2-3 焊接圖 .....	8
圖 2-2-4 側面圖 .....	8
圖 2-3-1 下載器連接方法 .....	10
圖 2-3-2 遙控器使用說明 .....	10
圖 3-1-1 軟體使用說明(一).....	16
圖 3-1-2 軟體使用說明(二).....	18
圖 3-1-3 校準時間 .....	18
圖 3-1-4 STC12C5A32S2 晶片程式碼 .....	18
圖 3-1-5 編寫程式碼 .....	18
圖 3-1-6 LED 電路圖 .....	18
圖 3-1-7 紅外接收二極體 .....	18
圖 3-1-8 文字燒錄部分電路 .....	19
圖 3-1-9 時鐘芯片電路 .....	19
圖 3-1-10 STC12C5A32S2 電路 .....	20
圖 3-1-11 外部組裝成果圖(一).....	21
圖 3-1-12 外部組裝成果圖(二).....	23

圖 4-1-1 外部包裝完成 .....	24
圖 4-1-2 外部包裝轉動圖 .....	24
圖 4-1-3 成果側面圖 .....	24
圖 4-1-4 實際轉動側面圖 .....	24

# 壹、前言

## 一、製作動機

二十一世紀是高科技的時代，隨著 LED 技術的普及化，LED 背光液晶顯示器廣泛的應用。各種我們熟知的電子產品上的指示燈、螢幕、照明燈、液晶螢幕用的 LED 背光源、手機上按鍵的背光、新世代的 OLED 螢幕、PLED 螢幕等，各式各樣關於 LED 的應用正逐漸穩定的發展中，在生活上如此普及的應用使我們的生活更加便利，因此本組對於 LED 的相關原理及應用想更深入地進行了解。

近幾年在街道的路旁、公車上、醫院裡日常生活中都看到對 LED 顯示的相關產品。在詳細的觀察與研究後發現，現今的平面顯示器，普通只能以單面觀看。有時因角度不同而使得的觀看者不易觀看的情形，因此本組在查閱相關文獻後，嘗試製作一個可旋轉式的 LED 顯示器，期待能提升觀看者在觀賞時的方便性。傳統的點矩陣 LED 顯示方式不僅成本較多，並將造成電能源大量耗損。

本專題製作的旋轉式 LED 顯示器，改變以往 LED 點陣方式顯示方式，除了減少 LED 使用數量外，並可將可視角度大大的擴展，可以 360°觀看完全無死角。並用少數的 LED 取代大量的 LED 點矩陣，因減少普通點矩陣顯示器因 LED 損壞而造成顯示故障的機率，進而減少維修成本，也可以提高顯示面的穩定性與色彩的均勻性，帶給人們一種新的視覺體驗。

## 二、製作目的

本專題研究平面顯示器的缺點並加以改善，可讓觀看者可以更方便以不同角度觀看顯示器，本專題將進行以下幾點進行研究。

- 改善目前平面顯示器單面顯示的缺點，製作一個可環形觀看的顯示器。
- 運用視覺暫留原理，設計只使用一排 LED 即可達成 360 度環型顯示器。
- 採用無線方式，來當做為本專題的電源供應方式。
- 本專題能夠降低能源的耗損以達成節能減碳的效果。



- 探討視覺暫留與單晶片之計算時間兩者之關係，並完成視覺暫留的效果。
- 藉由 CNY-70 感測器與單晶片結合以提供穩定的視覺暫留。

### 三、製作架構

#### (一) 專題製作流程

我們小組成員確定後，經小組一再的問題討論與溝通後，訂下了此次專題製作的題目。題目確定後，我們小組便開始構思如何去完成 LED 旋轉廣告顯示的輪文報告，首先，透過網路以及灣考書籍蒐集有用的資料，在整個專題應用過程中，如發現錯誤或者將不懂的理論以及製作上的問題細節，即會與指導老師進行討論，想辦法如何去解決且了解單晶片以及其他電子零組件是否能夠運用自如。當以上幾點都能如預期想法製作完成後即算是踏出了第一步。

## （二）製作流程圖



圖 1-3-1 專題製作流程圖

## 四、製作預期成效

本專題目前正朝著更高亮度，承受高氣候變化性，更高發光密度，發光均勻性，可靠性及全彩化等方向發展。未來也可以安裝在自行車或其他交通工具的車輪上可隨著車輪的旋轉來顯示圖案，透過他，可以使我們的訊號傳播方式更多樣化讓生活更加豐富。本專題的 LED 旋轉廣告燈可以利用軟體來更改字體與圖案，並用遙控器控制顯示方式，來呈現多樣化的圖案與文字。

## 貳、理論探討

### 一、電子相關零組件

#### (一) DS1302 時鐘芯片

DS1302 的引腳排列，其中 Vcc1 為後備電源，VCC2 為主電源。在主電源關閉的情況下，也能保持時鐘的連續運行。DS1302 由 Vcc1 或 Vcc2 兩者中的較大者供電。當 Vcc2 大於 Vcc1+0.2V 時，Vcc2 給 DS1302 供電。當 Vcc2 小於 Vcc1 時，DS1302 由 Vcc1 供電。

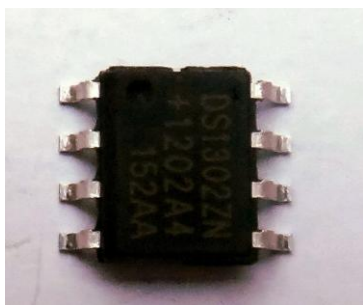


圖 2-1-1 DS1302 時鐘芯片

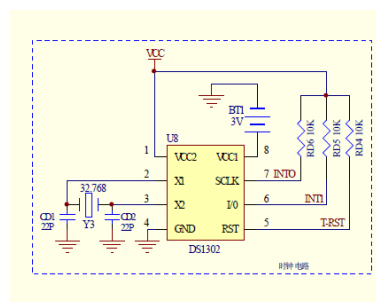


圖 2-1-2 DS1302 時鐘芯片電路

#### (二) 32.768KHZ 晶體振盪器

32.768KHZ 是一個標準的頻率，晶體振盪器頻率的應用主要有以下幾個方面的參數：尺寸、負載電容、頻率偏差、應用範圍。按尺寸外形來分主要分為插件和貼片的；插件的主要有 2\*6、3\*8、49s 等，貼片的就有很多種了，根據各公司的設計可用的型號有很多，例如：日本 KDS 晶體振盪器就有 49SMD、DST310S、SM—14J、DST520、DST410S 等。應用的範圍廣泛。也給人類帶來了時間的重要意義。也有直插的 DT-14、DT-26、DT-38，主要頻率是 32.768KHZ，廣泛應用在電錶、水錶、燃氣表、熱量表、氣表、工控儀表等。



圖 2-2-3 32.768KHZ 晶體振盪器

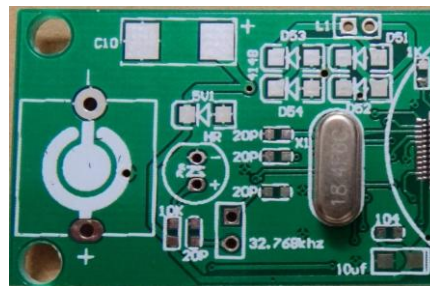


圖 2-2-4 32.768KHZ 晶體振盪器安裝位置

## 二、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 2-1-5 單晶片圖

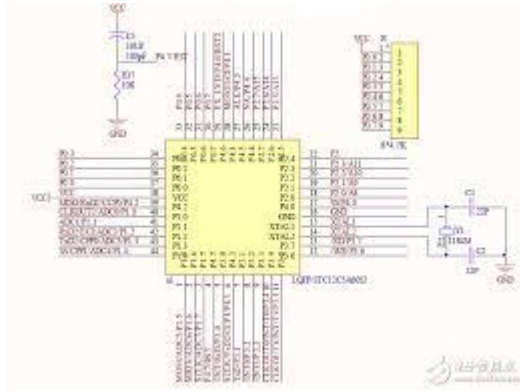


圖 2-1-5 單晶片電路圖

## 三、TC12C5A32S2 特性介紹

- STC12C5A32S2 系列主要性能：高速：1 個時 ca 鐘/機器週期，增強型 8051 內核，速度比普通 8051 快 8~12 倍，寬電壓：5.5~3.3V，2.2~3.6V( STC12LE5A60S2 系列 )
- 增加第二復位功能(高可靠復位，可，頻率<12MHz 吋，無需此功能)
- 增加外部掉電檢測電路,可在掉電時,及時將數據保存進 EEPROM,正常工作時無需操作 EEP
- 低功耗設計：空閒模式，(可由任意一個中斷喚醒)
- 低功耗設計：掉電模式(可由外部中斷喚醒)，可支持下降沿/上升沿和遠程喚醒
- 工作頻率：0~35MHz 時，相當於普通 8051：0~420MHz
- 時鐘：外部晶體或內部 RC 振盪器可選，在 ISP 下載編程用戶程序時設置 32K 字節片內閃存程序存儲器，擦寫次數 10 萬次以上
- 1280 字節片內 RAM 數據存儲器
- 芯片內 EEPROM 功能，擦寫次數 10 萬次以上
- ISP / IAP，在系統可編程/在應用可編程，無需編程器/仿真器
- 8 通道，10 位高速 ADC，速度可達 25 萬次/秒，2 路 PWM 還可當 2 路 D/A 使用
- 2 通道捕獲/比較單元 (PWM/PCA/CCP)，也可用來再實現 2 個定時器或 2 個外部中斷(支持上升沿/下降沿中斷)
- 4 個 16 位定時器，兼容普通 8051 的定時器 T0 / T1,2 路 PCA 實現 2 個定時

器可編程時鐘輸出功能，T0 在 P3.4 輸出時鐘，T1 在 P3.5 輸出時鐘，BRT 在 P1 0.0 輸出時鐘

- (WDT) 利用一個定時器，來監控主程式的運行
- 高速 SPI 串行通信端口
- 全雙工異步串行口(UART)，兼容普通 8051 的串口
- 先進的指令集結構，兼容普通 8051 指令集，有硬件乘法/除法指令通用 I/O 口，復位後為：准雙向口/弱上拉（普通 8051 傳統 I/O 口）
- 可設置成四種模式：准雙向口/弱上拉，推挽/強上拉，僅為輸入/高阻，開漏每個 I/O 口驅動能力均可達到 20 毫安，但整個芯片最大不得超過 100 毫安

#### 四、零件故障識別

觀察法：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。

#### 五、視覺暫留(persistence of vision)

當我們注視某物體一段時間後將視線移開，在短時間內仍可看到物體的形像，是一種視覺暫留的現象。可分為兩種：如果影像的顏色與原物同，這種影像稱為「正片後像」(positive afterimage)；如果顏色與原物不同，則稱為「負片後像」(negative afterimage)，產生負片後像的原因除視覺暫留的作用外，還與視覺疲勞有關。眼睛是人體所有的器官中最精緻、最細密的組織結構，其內部的構造像是電腦設定好的程式，具有自動化調節功能（accommodation），可接受來自四面八方不同的訊息，並自然地做出相對的反應，其中之所以能讓人看到物品(光的反射)，最 LED 電子鐘 2 重要的是「視網膜」跟「視神經系統」，當你看見物品時，視網膜負責將通過瞳孔的光線（也就是捕捉到的映像），經過視神經給予大腦視覺中樞精確的刺激，做出正確的判斷。映象處理過程中(視網膜→視神經→大腦)所產生的「惰性」和「時差」，即是視覺暫留(或稱視覺滯留)。視覺暫留例子：電影、走馬燈、煙火

如要想使顯示的字體和圖案不產生變形，還需要專門軟體對發光點進行的座標轉換。製作調試時運轉的振動現象易於控制。這種運動方式常見

的制作有：

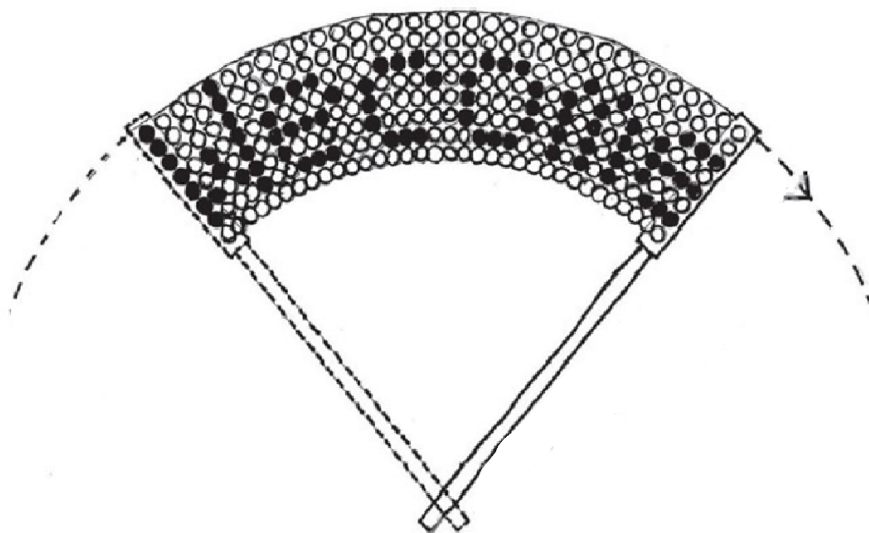


圖2-1-7 LED視覺暫留原理圖

由於顯示面不是一個平面，一般無法完整觀察整個顯示面，多採取畫面滾動顯示來彌補。

- 1：車輪上的LED顯示。
- 2：風扇上的LED顯示。
- 3：數位/類比雙模式顯示時鐘。

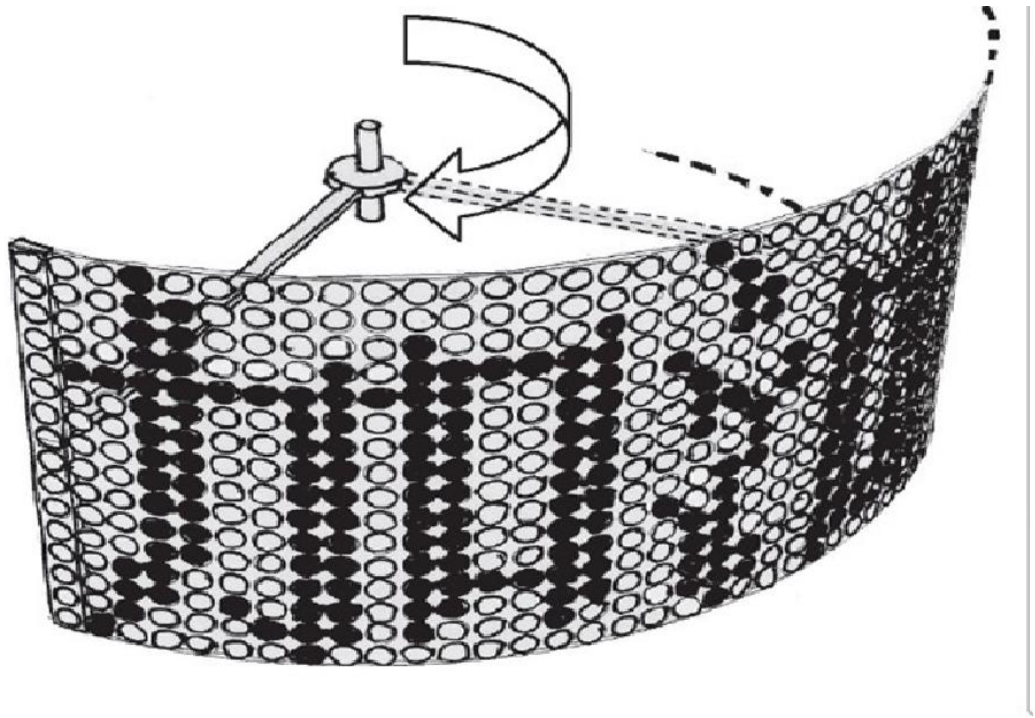


圖2-1-8 LED視覺暫留原理圖



## 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
電源供應器	供應所需電源
麵包版	模擬電路
三用電錶	檢測
PC 電腦	編譯程式
燒錄器	程式燒錄 STC12C5A32S2
Word	書面報告
Powerpoint	上台報告之簡報
示波器	觀察波行，找出錯誤
程式編譯器	編好之軟體除錯即轉燒錄檔

### 二、製作方法和製作步驟

旋轉LED的製作分為供電板、底座製作、主控板及立體轉接板四部份，以下為各部分焊接的部分圖

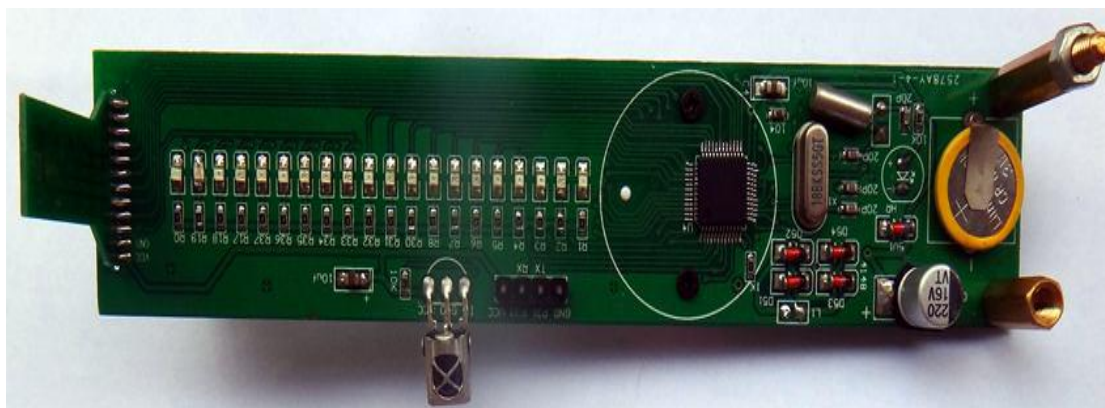


圖 2-2-1 正面俯視圖





圖 2-2-2 背面焊接圖

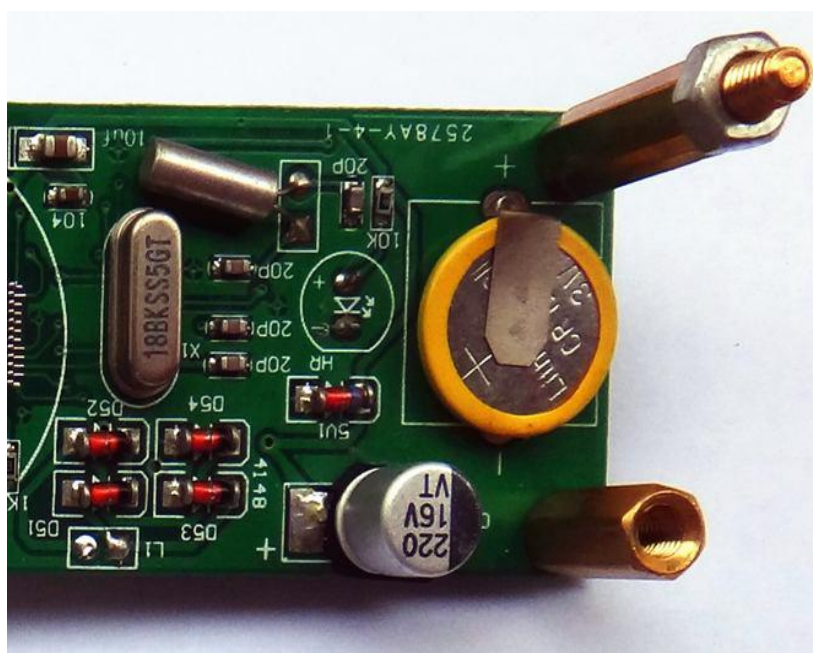


圖 2-2-3 焊接圖

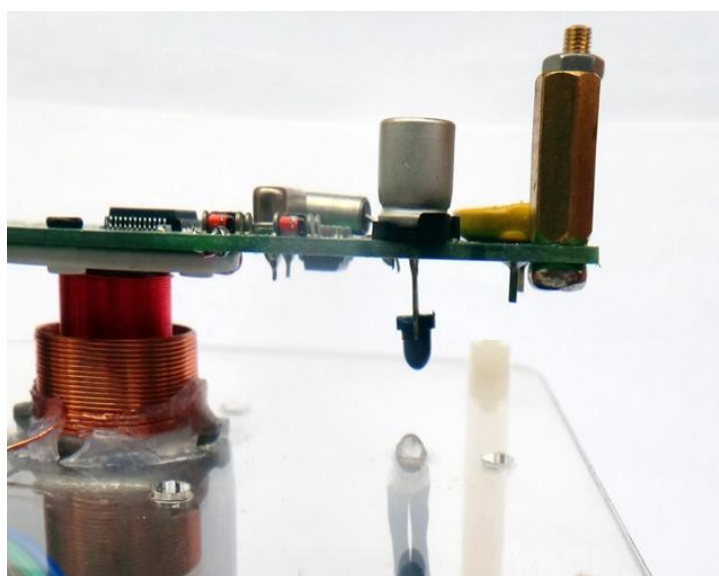


圖 2-2-4 側面圖

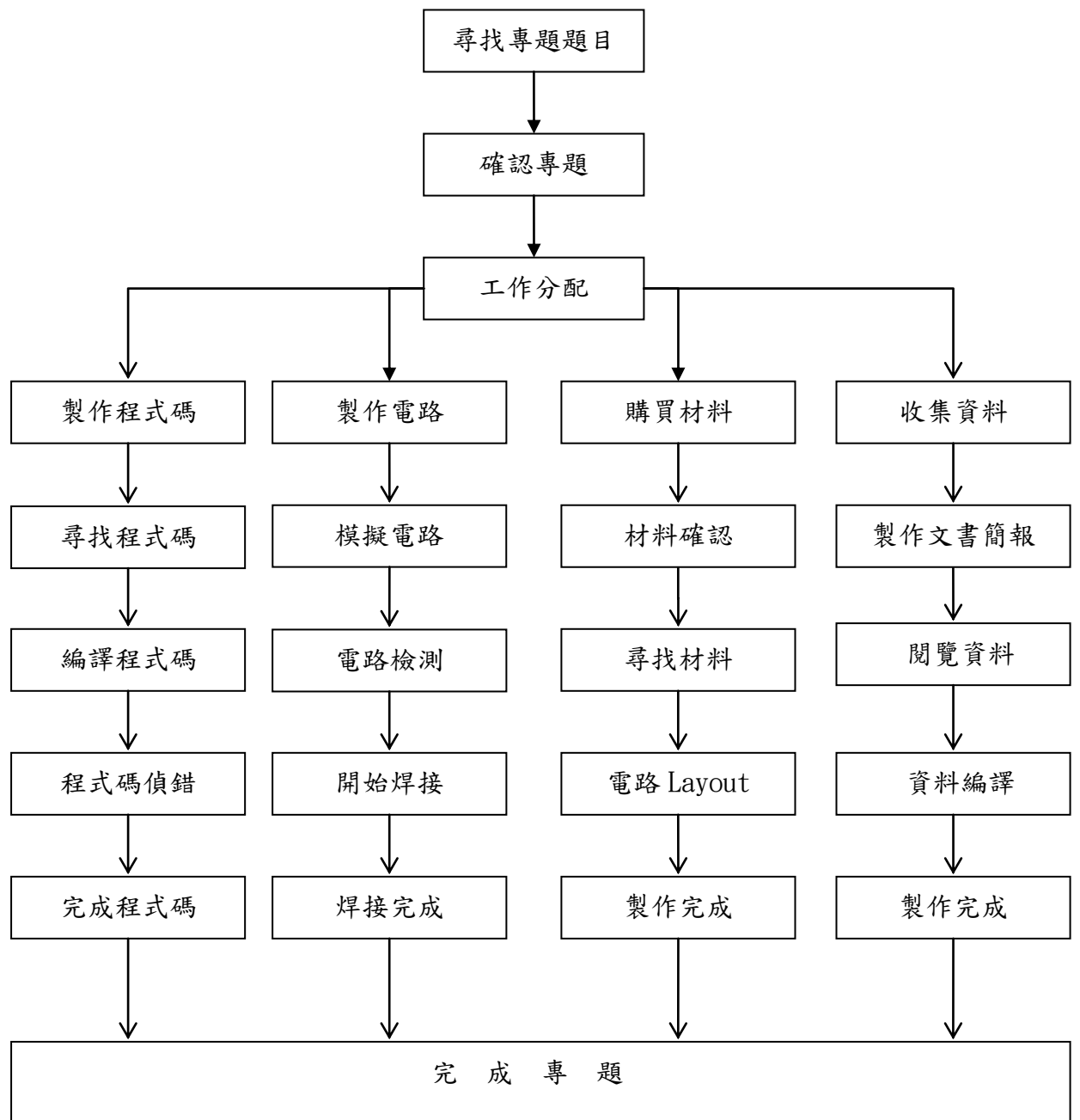


圖 2-3-1 製作方法及步驟

### 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		資訊科三年級	
專題名稱	中文名稱	LED 旋轉廣告顯示燈	
	英文名稱	LED indicator lights rotating advertising	
專題內容簡述		<p>現今生活中有許多產品都是以視覺暫留的特性來完成的，例如：小綠人、電子時鐘、電子廣告看板等。為了探討視覺暫留的概念，於是我們利用自身所學的 LED 應用技術、程式語言、電路繪製等，配合專題課程加以研究，製作簡易型的視覺暫留器，模擬市面上精緻的視覺暫留所呈現出來的效果，並經我們實際操作後深入了解其中的原理以及相關概念，以達到此次專題的研究目的。此專題製作主要利用視覺暫留原理的特性產生圖像，其原理類似於電影一般的連續畫面。我們將撰寫好的程式輸入至 STC12C5A32S2 內，搭配一排 LED 在不同的時間閃爍不同的 LED 燈，加上馬達的高速旋轉，產生對人眼的視覺暫留，顯示出我們所要求的畫面。</p>	
指導老師姓名		葉忠賢 老師	
參與同學姓名	呂宗翰組長		曾柏霖組員
	洪偉晟組員		洪啟原組員
專題執行日期		103 年 9 月 1 日至 104 年 5 月 31 日	

#### 四、旋轉 LED 改字軟體-平面立體版

改字前需連接USB 轉串口與主控板的下載口，看板上標號連線，先留空 GND 線，如下圖所示：

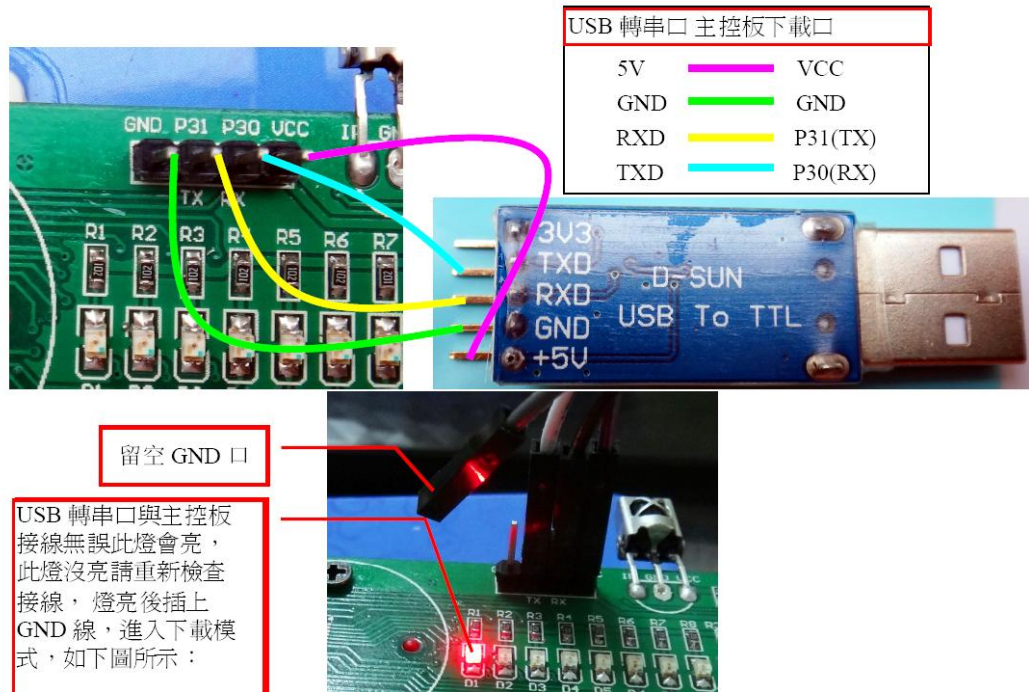


圖2-3-2 下載器連接方法



圖 2-3-3 遙控器使用說明



圖 3-2-1 軟體使用說明(一)



圖 3-1-2 軟體使用說明(二)

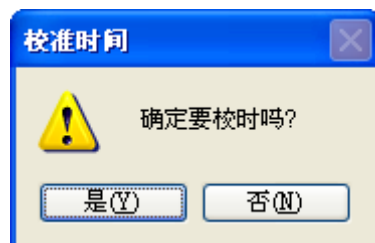


圖 3-1-3 校準時間



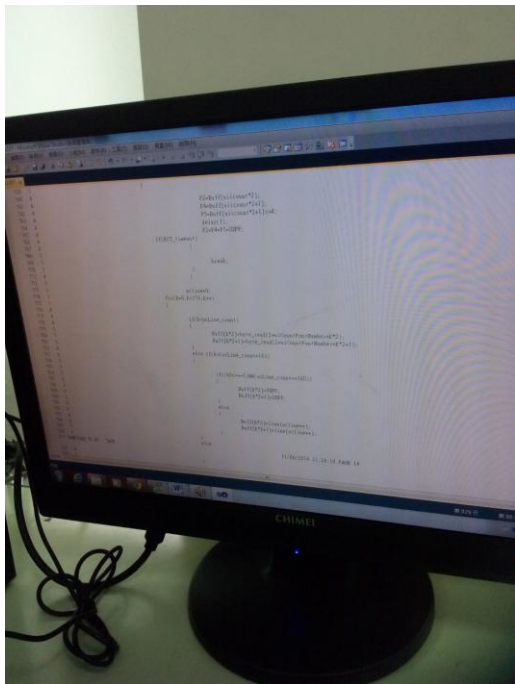


圖 3-1-4 STC12C5A32S2 晶片程式碼

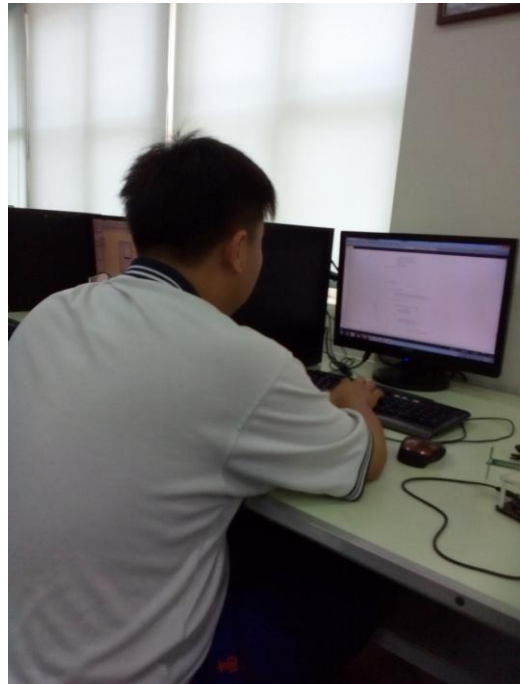


圖 3-1-5 編寫程式碼

#### (一) LED 旋轉廣告顯示燈功能及設定

透過這個專題製作，去了解如何運用單晶片在紅外線遙控器控制上，當在使用紅外線遙控器時，能進行遠距離遙控，且經由透過程式碼，來更改旋轉LED的顯示方式，使LED旋轉顯示燈有更多的顯示方式。

本專題的內容由89C51為基礎的做電路架構，其中功能、設定：

##### 1. LED旋轉廣告顯示燈開關設定

- (1) 送電時，馬達開始旋轉。
- (2) 使用遙控器切換1~9。
- (3) 晶片會讀取燒錄好的程式顯示

(二) 硬體電路圖：LED旋轉廣告顯示燈

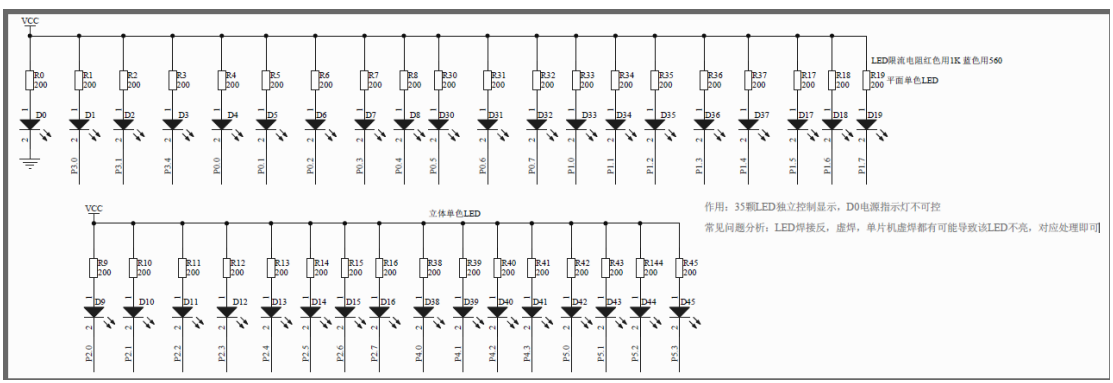


圖3-1-6 LED電路圖

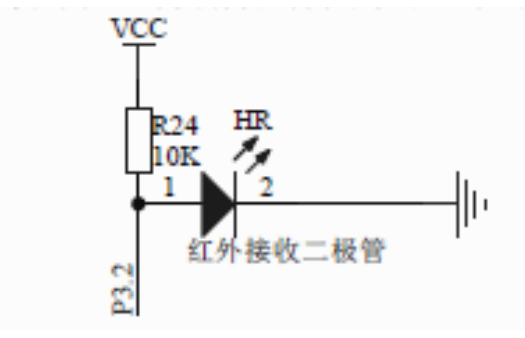


圖3-1-7 紅外接收二極體

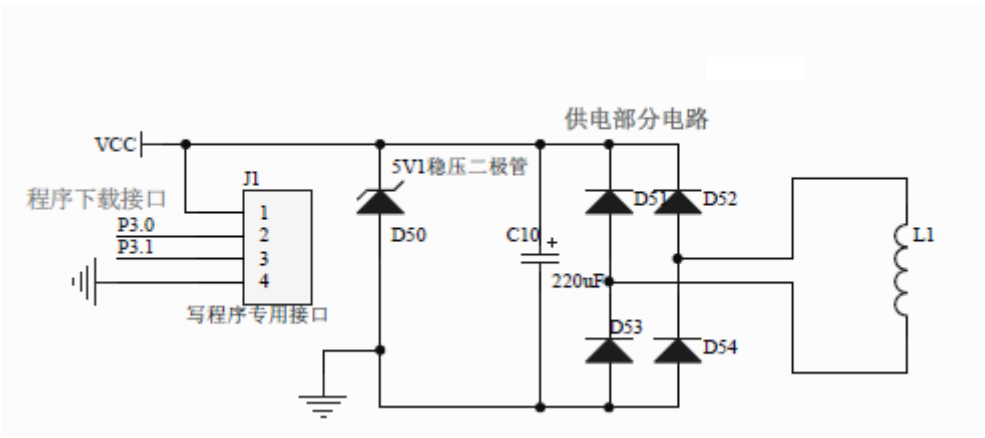


圖3-1-8 文字燒錄部分電路

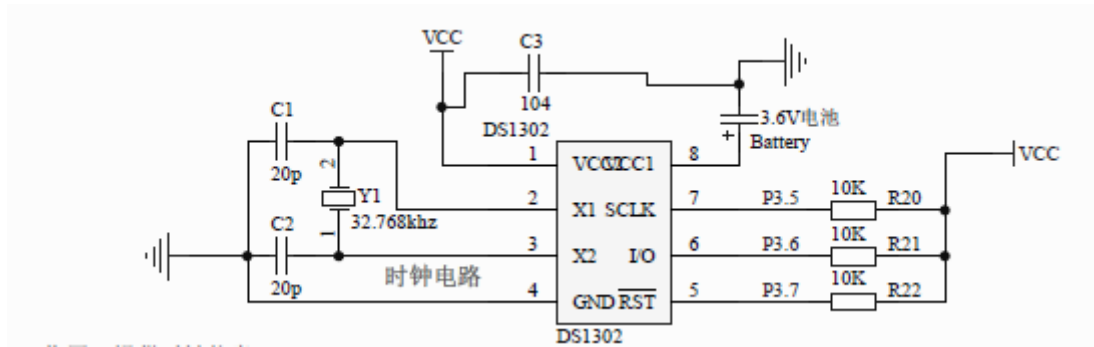


圖3-1-9 時鐘芯片電路

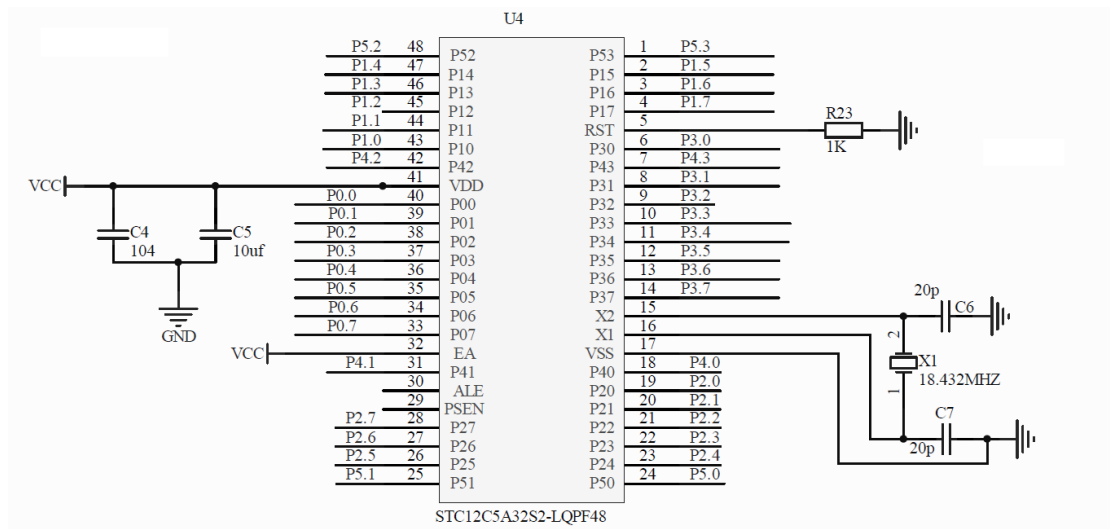


圖3-1-10 STC12C5A32S2電路



表3-1-11 LED旋轉廣告顯示燈之材料表

材料名稱	單位	數量	材料名稱	單位	數量
單晶片 STC12C5A32S2	個	1	4148 二極體	個	4
3.6V電池	顆	1	4P排針	個	1
限流電阻	個	36	紅外接收二極體	個	1
104電容	個	2	220uF電容	個	1
5V1穩壓管	個	1	10K電阻	個	5
一體化紅外接收器	個	1	旋轉主板 立體轉接板PCB板	套	1
32.768KHZ晶體振盪器	個	1	電機	個	1
0805 LED	個	35	配重銅柱、螺絲	套	2
20pF電容	個	4	大線圈	個	1
小線圈、塑料件	套	1	磁芯	個	1
DS1302 時鐘芯片	個	1	470電阻	個	3
10uF電容	個	2	10K電阻	個	1
18.432MHZ晶體振盪器	個	1	B772 中功率電晶體	個	1
主板與 塑料件固定螺絲	個	2	8cm熱縮管	條	1

8050電晶體	個	1	470uF電容	個	1
B772散熱片	個	1	4148二極體	個	2
222電容	個	1	壓克力底座	套	1
紅外線發光二極體	個	1	遙控器	個	1
5.0電源插頭	個	1	Usb電源線	條	1
5cm熱熔膠	條	1	無線供電PCB板	片	1

## 肆、製作成果

我們小組決定題目後，開始製作LED旋轉廣告顯示燈的電路、外型包裝、繪製設計電路圖；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3-1-12 外部組裝成果圖(一)



圖3-1-13 外部組裝成果圖(二)



圖3-1-13 實際轉動成果圖(一)



圖3-1-13 實際轉動成果圖(二)

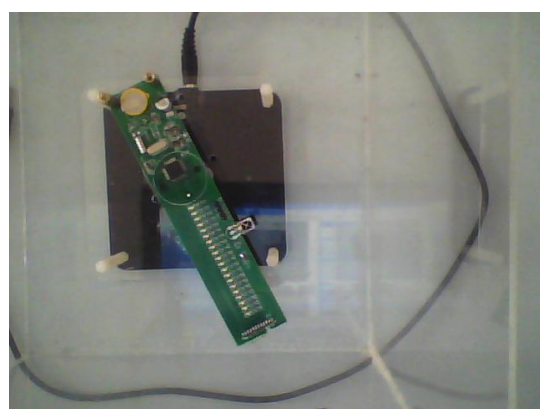


圖3-1-14 外部包裝完成

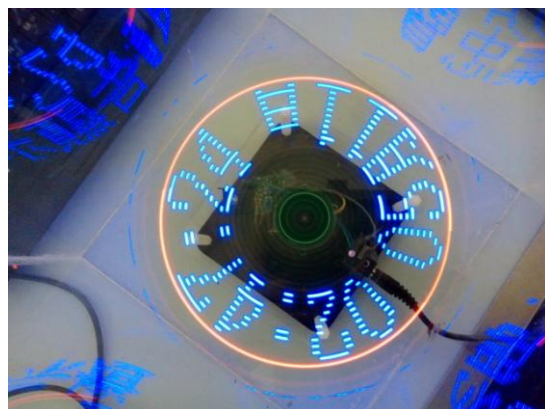


圖3-1-15 外部包裝轉動圖

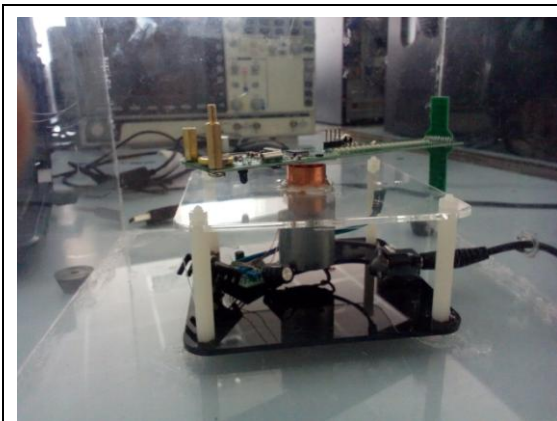


圖4-1-1 成果側面圖

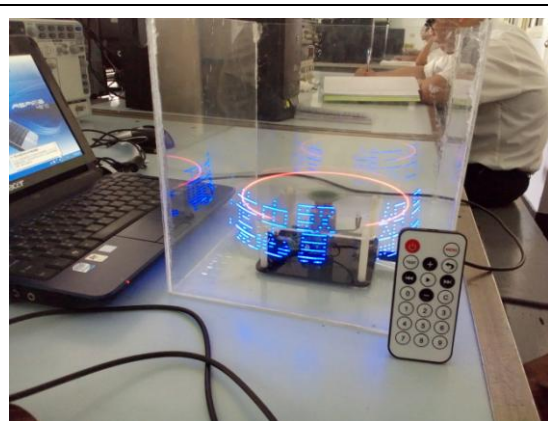


圖4-1-2實際轉動側面圖

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，由於有牽涉到木工配置的問題，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。

## 伍、結論與建議

本次的單元使我們將對整體我們小組來說是一個完整專題製作的學習過程，在過程中，我了解團隊合作的重要，而也進一步更了解在LED方面的知識。

### 一、結論

1. 本次專題我們遇到許多錯誤，還好有老師的幫忙，且在製作過程中，我們也意識到團隊合作的重要性，像外部包裝的壓克力板，在黏合過程中，就需要用到團隊合作。
2. 我們了解到STC12C5A32S2單晶片的特性
3. 這次LED旋轉廣告顯示燈相信未來可以用在大型廣告方面取代多數平面顯示燈，利用3D顯示的方式，來吸引目光。

### 二、建議

- 1.在旋轉過程中，請勿將手伸入展示盒內，以免造成割傷。
- 2.經過這次專題啟發後，我們認為未來可以改成投影式的LED旋轉廣告。

## 參考文獻

1. 陳宏明、游文億、許丁格、沈柏璵、廖崇筌，神奇 LED 顯示器設計與製作，建國科技大學。
2. 陳宏明、王毅賢、陳威孫、楊淳雅、楊國璋，旋轉輪型LED顯示器之控制與實現，建國科技大學。
3. 8051專題 – LED005 全彩廣告燈14顆RGB，LED(七色)。
4. 申寶琪 LED 電子時鐘。
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、吳金戊(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. 徐弘洋 教授 電子鐘專題報告 逢甲大學。

## LED 旋轉廣告顯示燈程式碼

```
#include "NEW_8051.H"
#include "task.h"
bit flag;
uchar id;
void Show_line_time(void)
{
    uchar i;
    for(i=0;i<8;i++)
    {
        P2=shuzi[Hour/16][i*2];
        P4=shuzi[Hour/16][i*2+1];
        P5=shuzi[Hour/16][i*2+1]>>4;
        delay_PIC(2);

    }
    P2=P4=P5=0XFF;
    for(i=0;i<8;i++)
    {
        P2=shuzi[Hour%16][i*2];
        P4=shuzi[Hour%16][i*2+1];
        P5=shuzi[Hour%16][i*2+1]>>4;
        delay_PIC(2);

    }
    P2=P4=P5=0XFF;
    /*****/

    P2=P4=P5=0XFF;
    for(i=0;i<8;i++)
    {
        P2=shuzi[10][i*2];
        P4=shuzi[10][i*2+1];
        P5=shuzi[10][i*2+1]>>4;
        delay_PIC(2);
    }
```

```

    }
    P2=P4=P5=0XFF;
    /***/
    for(i=0;i<8;i++)
    {
        P2=shuzi[Cent/16][i*2];
        P4=shuzi[Cent/16][i*2+1];
        P5=shuzi[Cent/16][i*2+1]>>4;
        delay_PIC(2);

    }
    P2=P4=P5=0XFF;
    for(i=0;i<8;i++)
    {
        P2=shuzi[Cent%16][i*2];
        P4=shuzi[Cent%16][i*2+1];
        P5=shuzi[Cent%16][i*2+1]>>4;
        delay_PIC(2);

    }
    P2=P4=P5=0XFF;
    for(i=0;i<8;i++)
    {
        P2=shuzi[10][i*2];
        P4=shuzi[10][i*2+1];
        P5=shuzi[10][i*2+1]>>4;
        delay_PIC(2);

    }
    P2=P4=P5=0XFF;
    /***/
    for(i=0;i<8;i++)
    {
        P2=shuzi[Sec/16][i*2];
        P4=shuzi[Sec/16][i*2+1];
        P5=shuzi[Sec/16][i*2+1]>>4;
        delay_PIC(2);
    }

```



```

    }
    P2=P4=P5=0XFF;
    for(i=0;i<8;i++)
    {
        P2=shuzi[Sec%16][i*2];
        P4=shuzi[Sec%16][i*2+1];
        P5=shuzi[Sec%16][i*2+1]>>4;
        delay_PIC(2);

    }

}

void display(void)
{
    uchar i;

    for(i=0;i<8;i++)
    {
        P1=tab[Sec%16][i*2];
        P0=tab[Sec%16][i*2+1];
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        if(BIT_timeout)
        {

            return;
        }
    }
}

```

```

}
P1=0XFF;
P0=0XFF;

for(i=0;i<8;i++)
{
    P1=tab[Sec/16][i*2];
    P0=tab[Sec/16][i*2+1];
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;
    }
}
P1=0XFF;
P0=0XFF;

for(i=0;i<8;i++)
{
    P1=tab[10][i*2];
    P0=tab[10][i*2+1];
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);

```

```

P2=P4=P5=0XFF;
P2=byte_read(2+uiicount*2);
P4=byte_read(2+uiicount*2+1);
P5=byte_read(2+uiicount*2+1)>>4;
uiicount++;
delay(1);
P2=P4=P5=0XFF;
if(BIT_timeout)
{

    return;

}
}
P1=0XFF;
P0=0XFF;

if((flag==1)&&(id==2))
{
    for(i=0;i<8;i++)
    {
        P1=0XFF;
        P0=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        if(BIT_timeout)
        {

```

```

        return;
    }
}
P1=0XFF;
P0=0XFF;

for(i=0;i<8;i++)
{
    P1=0XFF;
    P0=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;
    }
}
P1=0XFF;
P0=0XFF;

}
else
{
    for(i=0;i<8;i++)
    {
        P1=tab[Cent%16][i*2];

```

```

P0=tab[Cent%16][i*2+1];
P2=byte_read(2+uiicount*2);
P4=byte_read(2+uiicount*2+1);
P5=byte_read(2+uiicount*2+1)>>4;
uiicount++;
delay(1);
P2=P4=P5=0XFF;
P2=byte_read(2+uiicount*2);
P4=byte_read(2+uiicount*2+1);
P5=byte_read(2+uiicount*2+1)>>4;
uiicount++;
delay(1);
P2=P4=P5=0XFF;
if(BIT_timeout)
{

    return;

}
}
P0=0XFF;
P1=0XFF;

for(i=0;i<8;i++)
{
    P1=tab[Cent/16][i*2];
    P0=tab[Cent/16][i*2+1];
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;

```

```

        if(BIT_timeout)
        {

            return;

        }
    }
    P1=0XFF;
    P0=0XFF;

}
for(i=0;i<8;i++)
{
    P1=tab[10][i*2];
    P0=tab[10][i*2+1];
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;

    }
}
P1=0XFF;
P0=0XFF;

if((flag==1)&&(id==1))

```

```

{
    for(i=0;i<8;i++)
    {
        P0=0xFF;
        P1=0xFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0xFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0xFF;
        if(BIT_timeout)
        {

            return;

        }
    }
    P1=0xFF;
    P0=0xFF;

    for(i=0;i<8;i++)
    {
        P1=0xFF;
        P0=0xFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0xFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
    }
}

```

```

        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        if(BIT_timeout)
        {

            return;
        }
    }
    P1=0XFF;
    P0=0XFF;

}
else
{
    for(i=0;i<8;i++)
    {
        P1=tab[Hour%16][i*2];
        P0=tab[Hour%16][i*2+1];
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        if(BIT_timeout)
        {

            return;
        }
    }
}

```



```

P1=0XFF;
P0=0XFF;

for(i=0;i<8;i++)
{
    P1=tab[Hour/16][i*2];
    P0=tab[Hour/16][i*2+1];
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;
    }
}
P1=0XFF;
P0=0XFF;

}

for(i=10;i>0;i--)
{
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;

```

```

}

if((flag==1)&&(id==3))
{
    for(i=8;i>0;i--)
    {
        P1=0XFF;
        P0=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        if(BIT_timeout)
        {

            return;
        }
    }
    P1=0XFF;
    P0=0XFF;

    for(i=8;i>0;i--)
    {
        P1=0XFF;
        P0=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
    }
}

```

```

        P2=P4=P5=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        if(BIT_timeout)
        {

            return;

        }
    }
    P1=0XFF;
    P0=0XFF;

}
else
{
    for(i=8;i>0;i--)
    {
        P1=tab1[Month/16][i*2];
        P0=tab1[Month/16][i*2+1];
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        if(BIT_timeout)
        {

```

```

        return;
    }
}
P1=0XFF;
P0=0XFF;

for(i=8;i>0;i--)
{
    P1=tab1[Month%16][i*2];
    P0=tab1[Month%16][i*2+1];
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;
    }
}

P1=0XFF;
P0=0XFF;

```

```

P2=byte_read(2+uiicount*2);
P4=byte_read(2+uiicount*2+1);
P5=byte_read(2+uiicount*2+1)>>4;
uiicount++;

```

```

delay(1);
P2=P4=P5=0XFF;
for(i=8;i>0;i--)
{
    P1=tab1[10][i*2];
    P0=tab1[10][i*2+1];
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;
    }
}
P1=0XFF;
P0=0XFF;

if((flag==1)&&(id==4))
{
    for(i=8;i>0;i--)
    {
        P1=0XFF;
        P0=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
    }
}

```

```

    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;

}
P1=0XFF;
P0=0XFF;

for(i=8;i>0;i--)
{
    P1=0XFF;
    P0=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;
    }
}
P1=0XFF;
P0=0XFF;

```

```

}
else
{
    for(i=8;i>0;i--)
    {
        P1=tab1[Day/16][i*2];
        P0=tab1[Day/16][i*2+1];
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
    }
    P1=0XFF;
    P0=0XFF;
    for(i=8;i>0;i--)
    {
        P1=tab1[Day%16][i*2];
        P0=tab1[Day%16][i*2+1];
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
        P2=byte_read(2+uiicount*2);
        P4=byte_read(2+uiicount*2+1);
        P5=byte_read(2+uiicount*2+1)>>4;
        uiicount++;
        delay(1);
        P2=P4=P5=0XFF;
    }
}

```

```

        if(BIT_timeout)
        {

            return;
        }
    }
}
P1=0XFF;
P0=0XFF;

```

```

P2=byte_read(2+uiicount*2);
P4=byte_read(2+uiicount*2+1);
P5=byte_read(2+uiicount*2+1)>>4;
uiicount++;
delay(1);
P2=P4=P5=0XFF;
for(i=8;i>0;i--)
{
    P1=tab1[11][i*2];
    P0=tab1[11][i*2+1];
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;
    }
}

```



```

    }
}
P1=0XFF;
P0=0XFF;

for(i=8;i>0;i--)
{
    P1=0xFF;
    P0=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    P2=byte_read(2+uiicount*2);
    P4=byte_read(2+uiicount*2+1);
    P5=byte_read(2+uiicount*2+1)>>4;
    uiicount++;
    delay(1);
    P2=P4=P5=0XFF;
    if(BIT_timeout)
    {

        return;
    }
}

P1=0xFF;
P0=0XFF;

}

```