

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



藍芽無線控制跑馬燈

學生姓名： 吳育誠

鄭連豐

潘冠屹

指導老師： 葉忠賢 老師

中華民國 105 年 05 月

誌 謝

首先要感謝教導我們三年的忠賢老師，在這三年我們學到很多東西，而我們也將學到的東西結合在這次的專題中，另外也嘗試了許多沒碰過的元件，將雙方結合起來產生了這次的專題成品。

再來，要感謝我們的組員，雖然有摩擦但還是成功完成這次的專題，我們分工合作發揮自己所擅長的部份，首先感謝育誠的硬體方面的處理，有他在，我們在硬體方面真的如魚得水，他為我們解決成品的問題，再來感謝連豐，他為我們撰寫了軟體方面如程式之類的，接著文書方面由冠屹來編輯。每個人分工合作，也一同參與這次的專題製作，雖然不時很完美，但大老師說過：『只有神才是完美的，所以，不要向別人苛求任何東西。』他說的沒錯，雖然不能苛求別人，但可以要求自己，要求自己在時間內做好做滿，在以上這些條件下我們完成了這次的專題製作。

要感謝的人相信雙手來數都不夠，但能將自己所學之事發揮到極限，我想就是對他們最好的感謝方法之一，所以在這裡再次的感謝，教導我們的所有人，有你們的指導才有現在的我們。

吳育誠、鄭連豐、潘冠屹 謹上 2016/05

藍芽無線控制跑馬燈

摘 要

透過本篇研究微電腦 Arduino 的學習，Arduino，是一個開放原始碼的單晶片微控制器，簡單明了的編程環境 Arduino 的編程環境易於初學者使用，同時對高級用戶來講也足夠靈活。Arduino 軟件(IDE)能在 Windows、Macintosh OSX 和 Linux 操作系統中運行，而大多數其他單片機系統僅限於在 Windows 操作系統中運行。

現在有許多廣告看板都是使用 LED 矩陣列來進行運作，如果能隨時隨地來更改顯示的文字豈不是方便許多，所以就產生了這次的專題題目，我們將手機 APP 與 Arduino 和藍芽模組做個結合，手機 APP 輸入文字，將立即修改跑馬燈上的文字，這是一個非常方便的設計，也可以利用手機 APP 才控制跑馬燈亮度、速度等等。

透過 Arduino 的學習、以及 MAX7219 的運用設計一個無線跑馬燈的專題，我們選擇運用手機 APP 將文字軟體解碼再傳送到藍芽與 MAX7219 LED 進而顯示文字、能應用在生活中需要立即修改跑馬燈文字的時候。

關鍵詞：Arduino、藍芽模組、MAX7219 等等

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.....	19
一、設備及器材.....	19
二、製作方法與步驟.....	19
三、專題製作.....	21
肆、製作成果.....	28
伍、結論與建議.....	30
一、結論.....	30
二、建議.....	30
參考文獻.....	31
附錄一 Arduino UNO 程式碼.....	32

表目錄

表 2-1-1 HC-06 腳位定義說明表	4
表 2-1-2 低耗電藍芽以及典型藍芽對比表	7
表 2-2-1 MAX7219 腳位定義說明表	14
表 3-1-1 專題製作使用儀器（軟體）設備一覽表	19
表 3-3-2 藍芽無線控制跑馬燈之材料表	26

圖目錄

圖 1-1-1 專題製作流程圖	2
圖 2-1-1 HC-06 正面元件面	4
圖 2-1-2 HC-06 背面腳位定義	4
圖 2-1-3 藍芽標誌	7
圖 2-1-4 LED 發光實體圖	8
圖 2-2-1 Arduino 主控版	9
圖 2-2-2 Arduino 標誌	9
圖 2-2-2-3 Arduino IDE 編寫程式	11
圖 2-2-4 8X8 點陣 正面	12
圖 2-2-5 8X8 點陣 背面	12
圖 2-2-6 8X8 點陣 內部結構	12
圖 2-2-7 MAX7219 IC 接腳說明圖	13
圖 2-2-8 MAX7219 模組接腳說明圖	15
圖 2-2-9 MAX7219 模組接腳圖	15
圖 2-2-10 MAX7219 模組圖	15
圖 2-2-11 MAX7219 輸出入腳圖	16
圖 2-2-12 MAX7219 LED 資料輸出的引腳	16
圖 2-2-13 MAX7219 LED 串聯電路圖	16
圖 2-2-14 MAX7219 LED 麵包板接線圖	17
圖 2-2-15 宣告兩個字元陣列	18
圖 3-1-1 製作方法及步驟	20
圖 3-1-2 麵包板上電路模擬(一)	22
圖 3-1-3 麵包板上電路模擬(二)	22
圖 3-1-4 麵包板上電路模擬(三)	22
圖 3-1-5 麵包板上電路模擬(四)	22
圖 3-1-6 麵包板製作完成	22
圖 3-1-7 測試麵包板有無功能	22

圖 3-1-8 藍芽無線控制跑馬燈程式碼.....	23
圖 3-1-9 編寫程式碼.....	23
圖 3-1-10 藍芽無線控制跑馬燈電路圖.....	24
圖 3-1-11 藍芽無線控制跑馬燈完整圖(正面).....	25
圖 3-1-12 藍芽無線控制跑馬燈完整圖(背面).....	25
圖 3-1-13 藍芽接腳圖.....	27
圖 4-1-1 藍芽無線控制跑馬燈製作過程(一).....	28
圖 4-1-2 藍芽無線控制跑馬燈製作過程(二).....	28
圖 4-1-3 藍芽無線控制跑馬燈成品圖.....	28
圖 4-1-4 藍芽無線控制跑馬燈測試功能.....	28
圖 4-1-5 製作成品過程.....	28
圖 4-1-6 完成藍芽無線控制跑馬燈.....	28
圖 4-1-7 Bluetooth MCU 介面圖.....	29

壹、前言

一、製作動機

由於節能環保意識興起，許多的機車騎士紛紛改騎較環保的自行車(俗稱腳踏車)，也有許多人利用腳踏車來代步或者利用其來健身以保持身體健康，所以路上隨處可見腳踏車穿梭在車水馬龍中或是人行道兼自行車道上，雖說不僅環保也可以鍛鍊身體，但是也衍伸出一些安全上的問題，就像常常看到一些騎士轉彎時不慎與後方慢跑的跑者撞在一起，導致雙方的皮肉傷，更嚴重者還有如電視新聞上一些因轉彎而不慎被後方車輛擦撞或追撞的一些事故，有些甚至還奪走了腳踏車騎士的性命，大部分的肇事原因都是因為腳踏車並沒有猶如機車上的方向燈，所以導致許多腳踏車騎士想轉彎，卻因缺乏與後方車輛明確的溝通管道，產生後車撞前車的一個事故。

二、製作目的

最近幾年來因為購買腳踏車的人，比以前多了，也因為這樣出車禍的人也變多了，其中有 21.7%是因違反交通管制而出事的，其中就有多數人是因為晚上照明不足或尾燈燈光不足而發生車禍的，在快速駕駛或轉彎時是沒有辦法馬上看到位於坐墊下的燈的，也因為這樣，我們構思了直接在背上裝上 LED 燈，外加上可以控制左右燈的開關，要轉彎時只要扳一下開關，就可以讓後面的騎士知道你要左或右轉，主要目的如以下說明：

- (一)增加自行車騎士的安全性。
- (二)提醒後方來車了解自行車的行駛方向。
- (三)減少自行車事故的發生。
- (四)讓用路人能更安心行駛。

三、製作架構

(一) 專題製作流程

首先，我們先選定專題題目，在尋找如何製作此電路的材料，然後再進行零件模組的配線設計，然後在麵包板上配線、進行測試，測試成功後進行電路板 Layout 線路設計，在進行焊接作業，完成後再彙整文書參考資料，並開始將此文書檔輸出裝訂成本。

(二) 專題製作流程圖

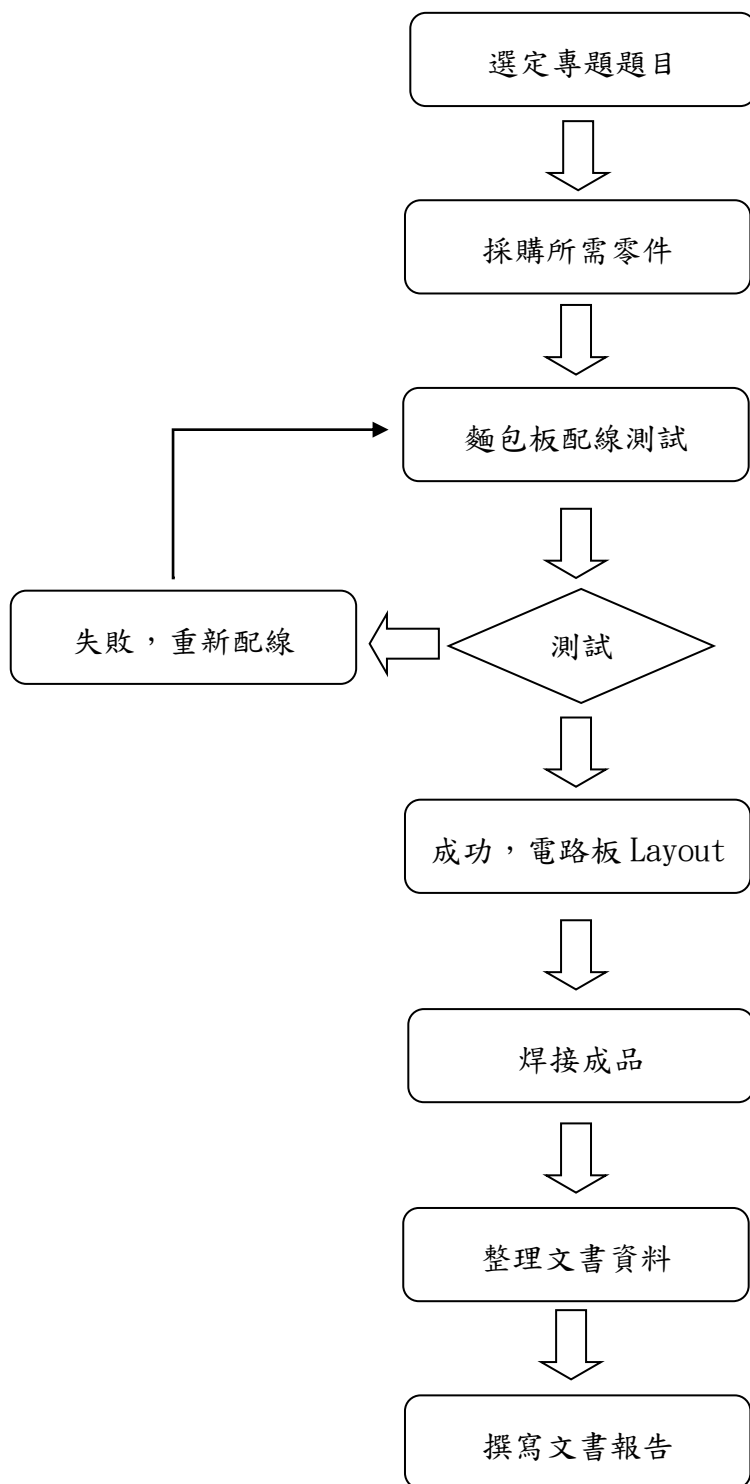


圖 1-3-1 專題製作流程圖

四、製作預期成效

我們這一個小組雖然是第一次一起進行專題製作，過程中有些許溝通上的摩擦，但到最後還是訂定了目標，初步定義成效如下：

- (一)當要左轉時，僅需按下手機 app 上的左轉按鈕。
- (二)當按下左轉按鈕後，手機 app 就會將訊號傳遞至藍芽模組上。
- (三)手機上的藍芽模組會把訊號傳遞至方向燈上的藍芽模組。
- (四)當接收電路板收到左轉的訊號時就會將訊號傳遞至方向燈上顯示。
- (五)LED 方向燈板上即會顯示向左遞增的箭型燈號。
- (六)當左轉完後恢復直行時，再按下直行按鈕。
- (七)LED 方向燈板上就會顯示出一些簡易的文字廣告。

貳、理論探討

一、電子相關零組件

(一) HC-06 藍芽模組

HC-06 藍芽模組的硬體是採用英國劍橋的 CSR 公司的 BC417143 的晶片，支援藍芽 2.0+EDR 規範，而工作電壓為 3.3 至 6 伏特，工作頻率由 2.4GHz 至 2.485GHz 的 ISM 頻段來進行通信，而傳輸距離在空曠地有效距離為 10 公尺。



圖 2-1-1 HC-06 正面元件面

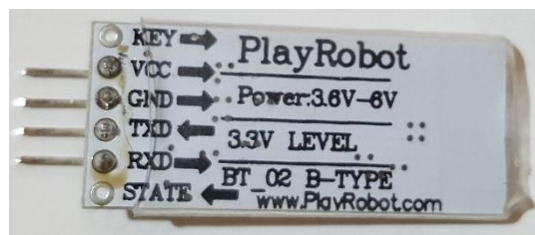


圖 2-1-2 HC-06 背面腳位定義

表 2-1-1 HC-06 腳位定義說明表

Pin	ID	Description
1	KEY	AT 設定模式致能腳
2	VCC	輸入電壓 3.6V 至 6V
3	GND	接地
4	TXD	UART TXD 輸出
5	RXD	UART RXD 輸入
6	STATE	藍芽連線狀態指示

HC-06 藍芽模組的兩種操作模式簡介：

- 1、自動連線 (Automatic Connection)，又稱為透通模式 (Transparent Communication)。
- 2、命令回應 (order-response)，又稱為 AT 模式 (AT mode)。

平時使用的「自動連線」模式只是把 RxD 腳傳入的資料，轉成藍芽無線訊號傳遞出去；或者將接收到的無線資料，從 TxD 腳傳給 Arduino，模組本身不會解讀資料，也不接受控制，操控藍芽模組的指令統稱 AT 命令 (AT-command)。AT 命令並非透過藍芽無線傳輸，而是模組的 TxD 和 RxD 接腳。藍芽模組只有在 AT 模式，才能接收 AT 命令。

而 HC-06 模組在與其他裝置連線之前，都是處於 AT 模式狀態，所以只要一通電，HC-06 模組就進入 AT 模式。HC-06 的 AT 命令，採用 9600bps 的速率傳送，一通電，HC-06 模組就進入 AT 模式，在尚未與裝置連線之前，

HC-06 板子上的 LED 將不停地閃爍；一旦與其他藍芽裝置連線（如：手機或電腦），LED 將維持點亮狀態。

藍芽簡介：

藍芽（英語：Bluetooth），一種無線技術標準，用來讓固定與行動裝置，在短距離間交換資料，以形成個人區域網路（PAN）。其使用短波特高頻（UHF）無線電波，經由 2.4 至 2.485 GHz 的 ISM 頻段來進行通信。1994 年由電信商愛立信（Ericsson）發展出這個技術。它最初的設計，是希望建立一個 RS-232 資料線的無線通訊替代版本。它能夠連結多個裝置，克服同步的問題。

藍芽技術最初由愛立信創製。技術始於愛立信公司的 1994 方案，它是研究在行動電話和其他配件間進行低功耗、低成本無線通信連線的方法。發明者希望為裝置間的通訊創造一組統一規則（標準化協定），以解決使用者間互不相容的移動電子裝置。1997 年前愛立信公司以此概念接觸了行動裝置製造商，討論其專案合作發展，結果獲得支援。

1999 年 5 月 20 日，索尼易立信、國際商業機器、英特爾、諾基亞及東芝公司等業界龍頭創立「特別興趣小組」（Special Interest Group, SIG），即藍芽技術聯盟的前身，目標是開發一個成本低、效益高、可以在短距離範圍內隨意無線連線的藍芽技術標準。

這項無線技術的名稱取自古代丹麥維京國王 Harald Blåtand 的名字，他以統一了因宗教戰爭和領土爭議而分裂的挪威與丹麥而聞名於世，而這個名字的英文便是 Harald Bluetooth。

1998 年時藍芽推出 0.7 規格，支援 Baseband 與 LMP（Link Manager Protocol）通訊協定兩部份。1999 年推出先後 0.8 版，0.9 版、1.0 Draft 版，1.0a 版、1.0B 版。1.0 Draft 版，完成 SDP（Service Discovery Protocol）協定、TCS（Telephony Control Specification）協定。1999 年 7 月 26 日正式公布 1.0 版，確定使用 2.4GHz 頻譜，最高資料傳輸速度 1Mbps，同時開始了大規模宣傳。和當時流行的紅外線技術相比，藍芽有著更高的傳輸速度，而且不需要像紅外線那樣進行介面對介面的連線，所有藍芽裝置基本上只要在有效通訊範圍內使用，就可以進行隨時連線。

當 1.0 規格推出以後，藍芽並未立即受到廣泛的應用，除了當時對應藍芽功能的電子裝置種類少，藍芽裝置也十分昂貴。2001 年的 1.1 版正式列入 IEEE 標準，Bluetooth 1.1 即為 IEEE 802.15.1。同年，SIG 成員公司超過 2000 家。過了幾年之後，採用藍芽技術的電子裝置如雨後春筍般增加，售價也大幅下降。為了擴寬藍芽的應用層面和傳輸速度，SIG 先後推出了 1.2、2.0 版，以及其他附加新功能，例如 EDR（Enhanced Data Rate，配合 2.0 的技術標準，將最大傳輸速度提高到 3Mbps）、A2DP（Advanced Audio Distribution Profile，一個控音軌分配技術，主要應用於立體聲耳機）、AVRCP（A/V Remote Control

Profile) 等。Bluetooth 2.0 將傳輸率提升至 2Mbps、3Mbps，遠大於 1.x 版的 1Mbps (實際約 723.2kbps)。

而 Bluetooth 在 2.4GHz 的電波干擾問題一直為大家所詬病，特別和無線區域網路間的互相干擾問題。有干擾發生時，就以重新傳送封包的方法來解決干擾。

從藍芽推出至今，簡單描述大約有以下幾種版本：

- 1、藍芽 1.0
- 2、藍芽 1.0B
- 3、藍芽 1.1
- 4、藍芽 1.2：可向下相容藍芽 1.1。
- 5、藍芽 2.0+EDR：加入了非跳躍窄頻通道，以及更高的連線速度 (實際測試速度為 280KB/s=2240Kbps)。
- 6、藍芽 2.1+EDR：向下對藍芽 1.2 版本完全相容。
- 7、藍芽 3.0+HS：更高的資料傳輸速率，整合 802.11PAL 最高速度可達 24Mbps。是 2.0 速度的 8 倍，以及引入了增強電源控制，實際空閒功耗明顯降低。
- 8、藍芽 4.0：超低的峰值、平均和待機模式功耗，覆蓋範圍增強，最大範圍可超過 100 米，以及提出了「低功耗藍芽」、「傳統藍芽」和「高速藍芽」三種模式，用途以及特色分別說明如下：
 - 高速藍芽：資料交換與傳輸。
 - 傳統藍芽：資訊溝通、裝置連線點
 - 低功耗藍芽：藍芽低功耗，以不需占用太多頻寬的裝置連線為主。前身其實是 NOKIA 開發的 Wibree 技術，本是作為一項專為行動裝置開發的極低功耗的移動無線通信技術，在被 SIG 接納並規範化之後重新命名為 Bluetooth Low Energy (後簡稱低功耗藍芽)。而這三種協定規範還能夠互相組合搭配、從而實作更廣泛的應用模式
- 9、藍芽 4.1：支援多裝置連接，以及其為藍芽 4.0 的軟體更新版本，搭載藍芽 4.0 裝置的終端可通過軟體更新獲得此版本。
- 10、藍芽 4.2

表 2-1-2 低耗電藍芽以及典型藍芽對比表

技術規範	典型藍芽	低耗電藍芽
無線電頻率	2.4 GHz	2.4 GHz
距離	10 米/100 米	30 米
空中資料速率	1-3 Mb/s	1 Mb/s
應用吞吐量	0.7-2.1 Mb/s	0.2 Mb/s
節點／單元	7-16,777,184	未定義（理論最大值為 2^{32} ）
安全	64/128-bit 及用戶自定義的應用層	128-bit AES 及用戶自定義的應用層
強健性	自動適應快速跳頻，FEC，快速 ACK	自動適應快速跳頻
發送資料的總時間	100 ms	<6 ms
政府監管	全球	全球
認證機構	藍芽技術聯盟（Bluetooth SIG）	藍芽技術聯盟（Bluetooth SIG）
語音能力	有	沒有
網路拓撲	分散網	Star-bus
耗電量	1（作為參考）	0.01 至 0.5（視使用情況）
最大操作電流	<30 mA	<15 mA（最高運行時為 15 mA）
服務探索	有	有
簡介概念	有	有
主要用途	手機，遊戲機，耳機，立體聲音頻串流，汽車和 PC 等	手機，遊戲機，PC，錶，體育和健身，醫療保健，汽車，家用電子，自動化和工業等



圖 2-1-3 藍芽標誌

(二)發光二極體

發光二極體 (Light-Emitting Diode, LED) 是一種能發光的半導體電子元件，透過三價與五價元素所組成的複合光源。此種電子元件早在 1962 年出現，早期只能夠發出低光度的紅光，被 hp 買下專利後當作指示燈利用。及後發展出其他單色光的版本，時至今日，能夠發出的光已經遍及可見光、紅外線及紫外線，光度亦提高到相當高的程度。用途由初時的指示燈及顯示板等；隨著白光發光二極體的出現，近年逐漸發展至被普遍用作照明用途。

發光二極體只能夠往一個方向導通 (通電)，叫作正向偏置 (正向偏壓)，當電流流過時，電子與電洞在其內重合而發出單色光，這叫電致發光效應，而光線的波長、顏色跟其所採用的半導體物料種類與故意滲入的元素雜質有關。具有效率高、壽命長、不易破損、反應速度快、可靠性高等傳統光源不及的優點。白光 LED 的發光效率近年有所進步；每千流明成本，也因為大量的資金投入使價格下降，但成本仍遠高於其他的傳統照明。雖然如此，近年仍然越來越多被用在照明用途上。

LED 即發光二極體，是嵌入式系統中常用的輸出設備。單個 LED 通常用作報警之顯示，故障指示或提示信號等。LED 是一個半導體設備，當電流通過它的時候，可以產生可見光。LED 的發光強度與通過的電流強度成正比，由 64 個 LED 組成，8 行 8 列。由於 LED 的發光顏色和發光效率與製作 LED 的材料和工藝有關，目前廣泛使用的有紅、綠、藍三種。由於 LED 工作電壓低 (僅 1.5-3V)，能主動發光且有一定亮度，亮度又能用電壓 (或電流) 調節，本身又耐衝擊、抗振動、壽命長 (10 萬小時)，所以在大型的顯示設備中，目前尚無其他的顯示方式與 LED 顯示方式匹敵。



圖 2-1-4 LED 發光實體圖

二、單晶片微電腦

(一) Arduino

Arduino，是一個開放原始碼的單晶片微控制器，它使用了 Atmel AVR 單片機，採用了開放原始碼的軟硬體平台，建構於簡易輸出/輸入（simple I/O）介面板，並且具有使用類似 Java、C 語言的 Processing/Wiring 開發環境。

Arduino 可以使用 Arduino 官方的 Arduino IDE 進行指令碼撰寫 & 編輯，結合其他電子元件，例如開關或感測器或其他控制器件、LED、馬達或其他輸出裝置，作出互動作品。Arduino 亦也可以獨立運作成為一個可以跟軟體溝通的介面。

Arduino 可以使用 Arduino 語言與 Macromedia Flash、Processing、Max/MSP、Pure Data 和 SuperCollider 跟 Java 和 make block.cc 等軟體，結合電子元件，例如開關或感測器或其他控制器件、LED、步進馬達或其他輸出裝置，作出互動作品。Arduino 也可以獨立運作成為一個可以跟軟體溝通的介面。



圖 2-2-1 Arduino 主控版



圖 2-2-2 Arduino 標誌

Arduino 開發沿革：

Arduino 的核心開發團隊成員包括：馬西莫·班齊（Massimo Banzi）、大衛·奎提耶斯（David Cuartielles）、湯姆·伊果（Tom Igo）、贊布羅塔·馬提諾（Gianluca Martino）、大衛·梅利斯（David Mellis）和尼可拉斯·蘭比提（Nicholas Zambetti）。

據說馬西莫·班齊之前是義大利 Ivrea 一家高科技設計學校的老師。他的學生們經常抱怨找不到便宜好用的微控制器。2005 年冬天，馬西莫·班齊跟大衛·奎提耶斯討論了這個問題。大衛·奎提耶斯是一個西班牙籍晶片工程師，當時在這所學校做訪問學者。兩人決定設計自己的電路板，並引入了馬西莫·班齊的學生大衛·梅利斯為電路板設計編程語言。兩天以後，大衛·梅利斯就寫出了程式碼。又過了三天，電路板就完工了。這塊電路板被命名為 Arduino。幾乎任何人，即使不懂電腦編程，也能用 Arduino 做出很酷的東西，比如對感測器作出回應，閃爍燈光，還能控制馬達。隨後馬西莫·班齊、大衛·奎提耶斯和大衛·梅利斯把設計圖放到了網上。保持設計的開放源碼理念，因為版權法可以監管開源軟體，卻很難用在硬體上，他們決定採用創用 CC 許可。[1]創用 CC 是為保護開放版權行為而出現的類似 GPL 的一種許可（license）。在創用 CC 許可下，任何人都被允許生產印刷電路板的複製品，還能重新設計，甚至銷售原設計的複製品。你不需要付版稅，甚至不用取得 Arduino 團隊的許可。然而，如果你重新發布了引用設計，你必須說明原始 Arduino 團隊的貢獻。如果你調整或改動了電路板，你的最新設計必須使用相同或類似的創用 CC 許可，以保證新版本的 Arduino 電路板也會一樣的自由和開放。唯一被保留的只有 Arduino 這個名字。它被註冊成了商標。如果有人想用這個名字賣電路板，那他們可能必須付一點商標費用給 Arduino 的核心開發團隊成員。

贊布羅塔·馬提諾創立的 Arduino Srl 被 2009 年創立的 Arduino LLC 控告侵犯了他們的著作權，這第二家被控侵權的 Arduino（也就是 Arduino Srl），原先叫做 Smart Projects Srl，以前是在義大利負責生產製造 Arduino 控制板的公司。而第一家 Arduino（也就是我們較熟悉的 Arduino LLC）則負責開發控制板，並管理周邊的開放原始碼專題與社群。在之前，原本雙方是合作的關係；2014 年 Martino 與另外四位共同創辦人，對於 Arduino 品牌的發展方向意見不合，導致 Martino 另外請費德里科·穆斯托擔任 Smart Projects 的新執行長，並把公司名稱改為 Arduino Srl。

Arduino 特色：

(一)基於創用 CC 開放原始碼的電路圖設計。

- (二)免費下載，也可依需求自己修改，但需遵照姓名標示。您必須按照作者或授權人所指定的方式，表彰其姓名。
- (三)依相同方式分享，若您改變或轉變著作，當散布該衍生著作時，您需採用與本著作相同或類似的授權條款。
- (四)Arduino 可使用 ICSP 線上燒入器，將 Bootloader 燒入新的 IC 晶片。可依據 Arduino 官方網站，取得硬體的設計檔，加以調整電路板及元件，以符合自己實際設計的需求。
- (五)可簡單地與感測器，各式各樣的電子元件連接，如 紅外線、超音波、熱敏電阻、光敏電阻、伺服馬達...等。
- (六)支援多樣的互動程式，如 Adobe Flash, Pure Data, C, Processing...等。
- (七)使用低價格的微處理控制器 (Atmel AVR) (ATMEGA 8,168,328 等)。USB 介面，不需外接電源。另外有提供直流(DC)電源輸入。

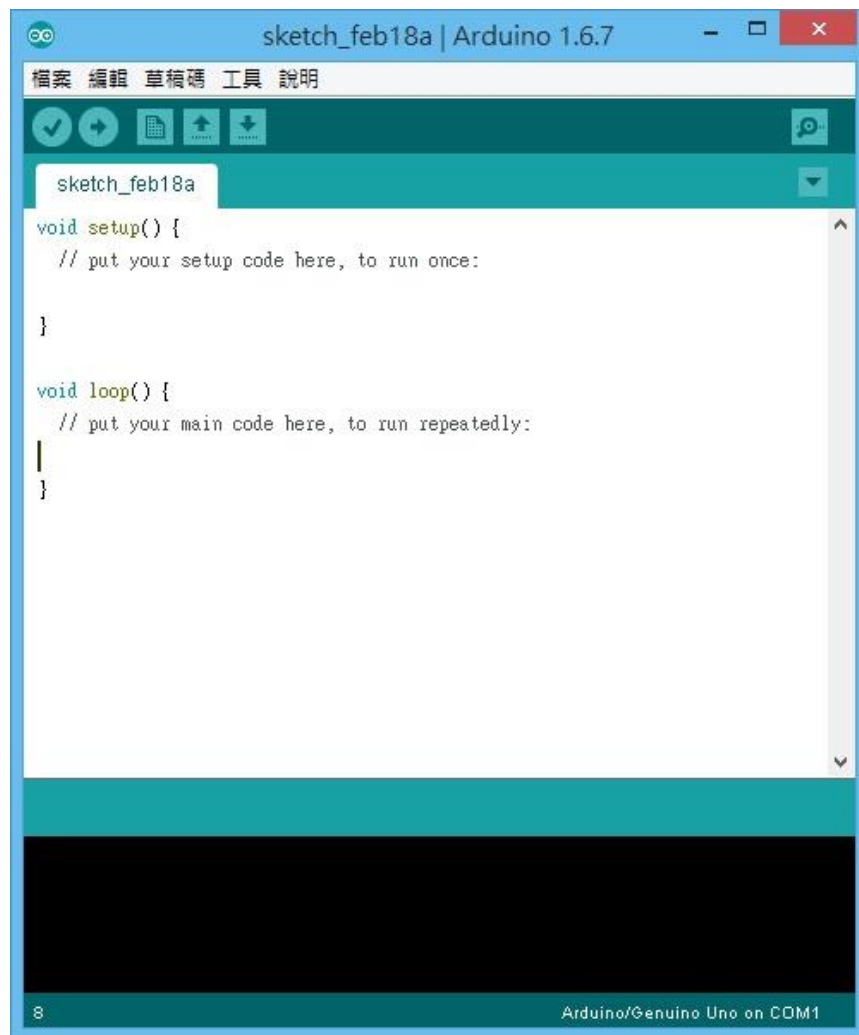


圖 2-2-3 Arduino IDE 編寫程式

(二) MAX7219 點陣模組

1、8X8 點陣模組

透過掃描式的方式來更新每一個 LED 點，透過人眼的視覺暫留的特性與原理，所有的點在十分之一秒內可以全部輪流亮過，對人眼的視覺來說，自動會組合成一個完整的畫面，所以利用這樣的原理來設計與控制矩陣式 LED，就可以拼湊出一個完整的畫面。

但是矩陣式 LED 是製造廠商先行將 LED 置入如圖 2-2-1 所示的矩陣式 LED 顯示模組之中，並如圖 2-2-2 所示，將矩陣式 LED 所有的接腳整理成『列接腳』與『行接腳』。

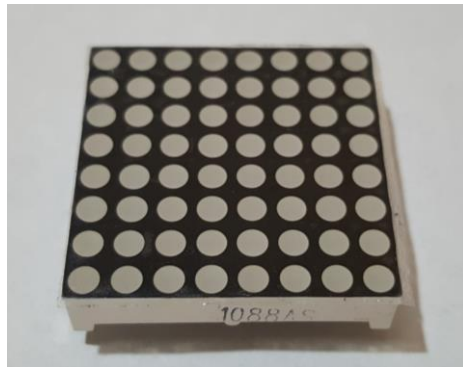


圖 2-2-4 8X8 點陣 正面

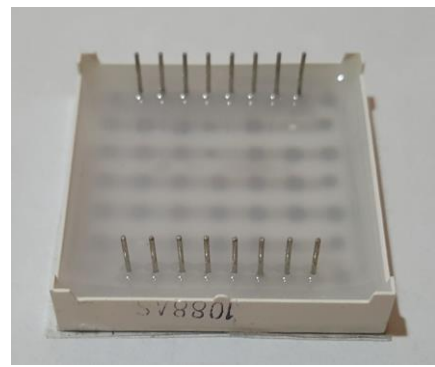
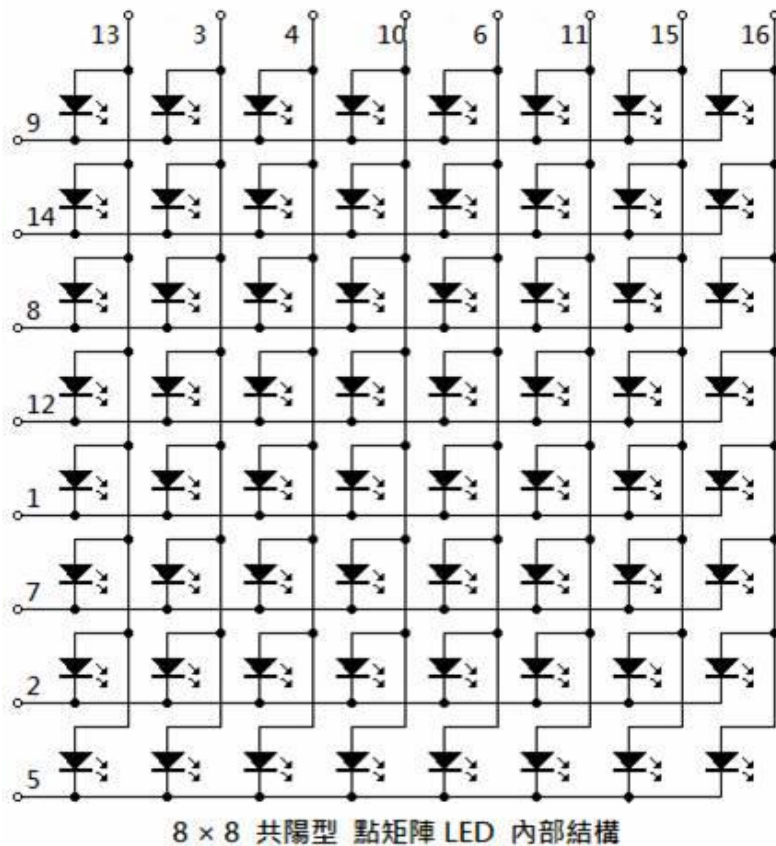


圖 2-2-5 8X8 點陣 背面



8 × 8 共陽型 點矩陣 LED 內部結構

圖 2-2-6 8X8 點陣 內部結構

2、MAX7219

MAXIM 公司生產的整合的 IC，由下圖所示，MAX7219 是一個 24 接腳的 DIP 封裝或 DIP 包裝，透過串列輸入／輸出共陰極顯示之驅動晶片 (Driver IC)，它可以連接微處理器與 8 位數字的 7 段數字 LED 顯示，也可以連接條線圖顯示器或者 64 個獨立的 LED：包括一個 B 型 BCD 編碼器、多路掃描迴路，還有一個 8*8 的靜態 RAM 用來存儲每一個數據。只需要一個外部暫存器用來設置各個 LED 的電流。

該晶片具有 10MHz 傳輸率的 Serial Peripheral interface (SPI) 三線串列介面，並可以和任何微處理器 (包括單晶片) 相連，只需一個外接電阻即可設置所有 LED 的電流。它的操作很簡單，單晶片只需透過 SPI 三線介面就可以將相關的指令寫入 MAX7219 的內部指令和資料暫存器，同時它還允許使用者選擇多種解碼方式和譯碼位元。此外它還支援多片 7219 串聯方式，這樣單晶片就可以通過 3 根線 (即串列資料線：Din、串列時鐘線：Clock 和晶片：CS 擇線) 控制更多的 7 段顯示器或 8*8 矩陣式 LED 顯示器。

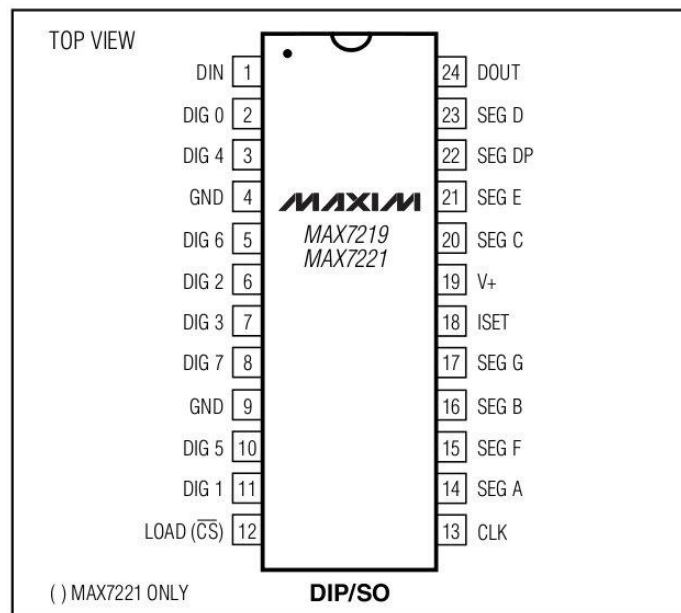


圖 2-2-7 MAX7219 IC 接腳說明圖

3、MAX7219 LED 顯示模組介紹

MAX7219 是一種整合的串列輸入/輸出共陰極顯示驅動器，它連接微處理器與 8 位元數位的 7 段數位 LED 顯示，也可以連接條線圖顯示器或者 64 個獨立的 LED。其上包括一個片上的 B 型 BCD 編碼器、多路掃描迴路，段字驅動器，而且還有一個 8*8 的靜態 RAM 用來存儲每一個資料。只有一個外部暫存器用來設置各個 LED 的段電流。

只要使用四線串列介面可以聯接所有通用的微處理器。每個資料可以定址在更新時不需要改寫所有的顯示。MAX7219 同樣允許使用者對每一個資料選擇編碼或者不編碼。整個設備包含一個 150 μA 的低功耗關閉模式，類比和數位亮度控制，一個掃描數量暫存器允許使用者顯示 1-8 位元資料，還有一個讓所有 LED 發光的檢測模式。

MAX7219 模組參數：

- 1、單個模組可以驅動一個 8*8 共陰 LED 矩陣顯示器
- 2、模組工作電壓：5V
- 3、模組尺寸：5mm (長) X 2mm (寬) X 1.5mm(高)
- 4、帶 4 個固定螺絲孔，孔徑 3 mm，可使用 M3 銅柱固定
- 5、模組帶輸入輸出介面，支援多個模組級聯

MAX7219 模組接線說明：

- 1、模組左邊為輸入埠，右邊為輸出埠。
- 2、控制單個模組時，只需要將輸入埠接到微處理機（單晶片）
- 3、多個模組級聯時，第 1 個模組的輸入端接微處理機（單晶片），輸出端接第 2 個模組的輸入端，第 2 個模組的輸出端接第 3 個模組的輸入端，以此類推。

表 2-2-1 MAX7219 腳位定義說明表

ID	Description
VCC	輸入電壓 4.5V 至 5V
GND	接地
CLK	串列時脈信號，最高頻率 10MHz
DIN	串列資料登錄端
DOUT	串列資料輸出端，用於級連擴展
CS	讀取腳，致能端

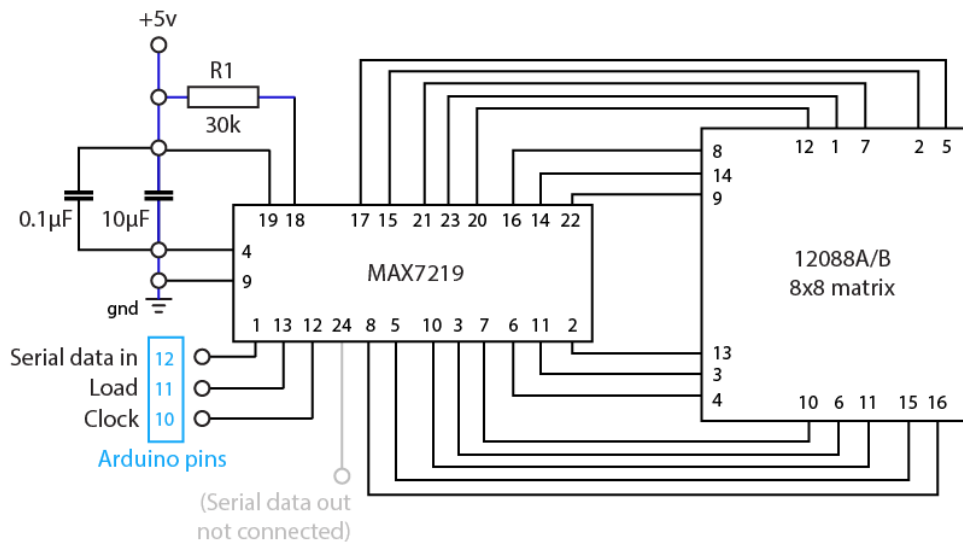


圖 2-2-8 MAX7219 模組接腳說明圖

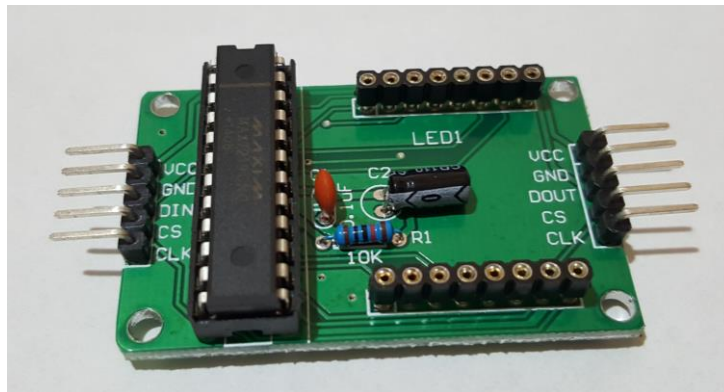


圖 2-2-9 MAX7219 模組接腳圖

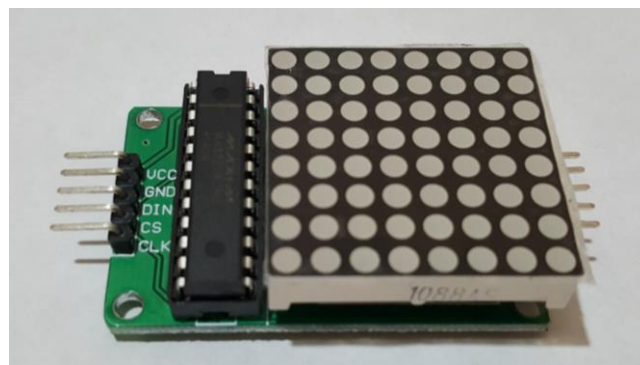


圖 2-2-10 MAX7219 模組圖

4、串連 MAX7219 LED 點陣模組

MAX7219 IC 第 1 和 24 腳，分別是序列資料輸入及資料輸出，因此這個 IC 和 74HC595 一樣，都能串接在一起

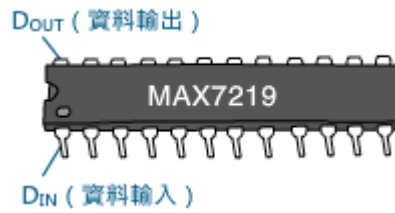


圖 2-2-11 MAX7219 輸出入腳圖

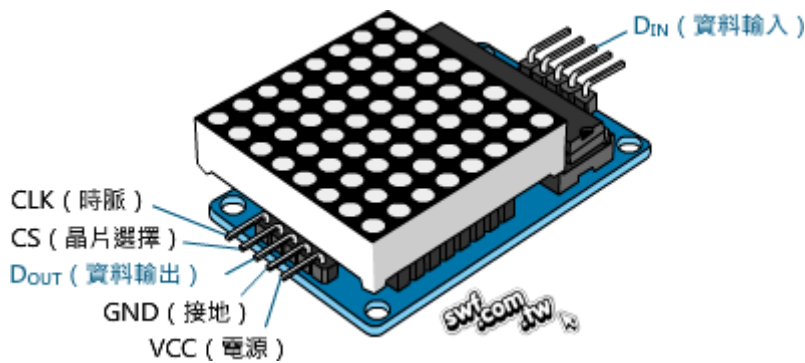


圖 2-2-12 MAX7219 LED 資料輸出的引腳

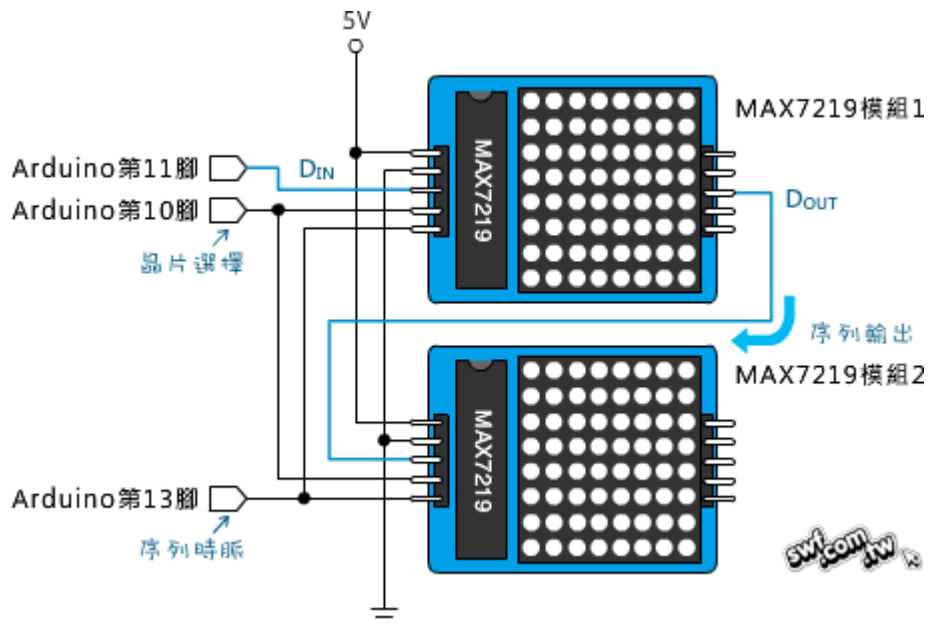


圖 2-2-13 MAX7219 LED 串聯電路圖

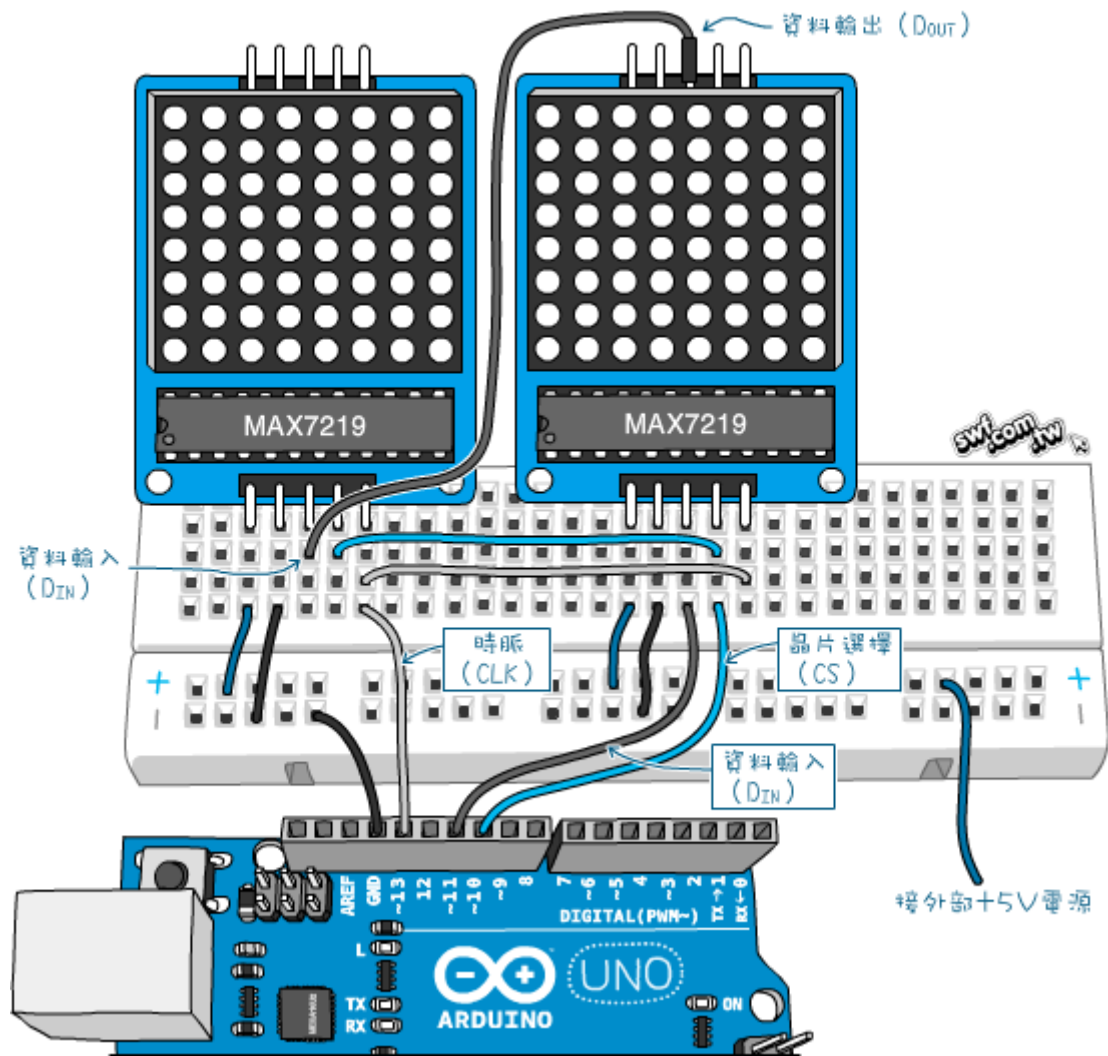


圖 2-2-13 MAX7219 LED 麵包板接線圖

4、MaxMatrix 程式庫建立捲動文字

香港城市大學的區建忠教授 (Kin-Chung Au) 寫了一個 MaxMatrix 程式庫，可大幅簡化 LED 點矩陣的程式碼、支援串連顯示並提供四種移動效果：

- shiftLeft (左)
- shiftRight (右)
- shiftUp (上)
- shiftDown (下)

5、複製陣列元素

使用 MaxMatrix 程式庫建立文字捲動程式，需要用到複製陣列元素指令，在此先說明，複製陣列不像複製簡單型變數，無法用一個指派（“=”）運算子搞定。

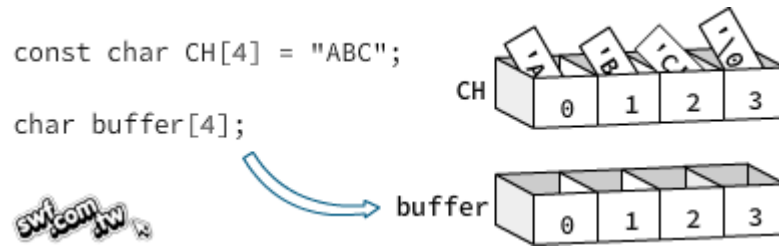


圖 2-2-14 宣告兩個字元陣列

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
Arduino UNO	專題之微型處理器
Arduino IDE	專題之微處理器程式碼編寫
MAX7219	專題之 32*8 點矩陣顯示器
Word	書面報告
Powerpoint	上台報告之簡報

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。

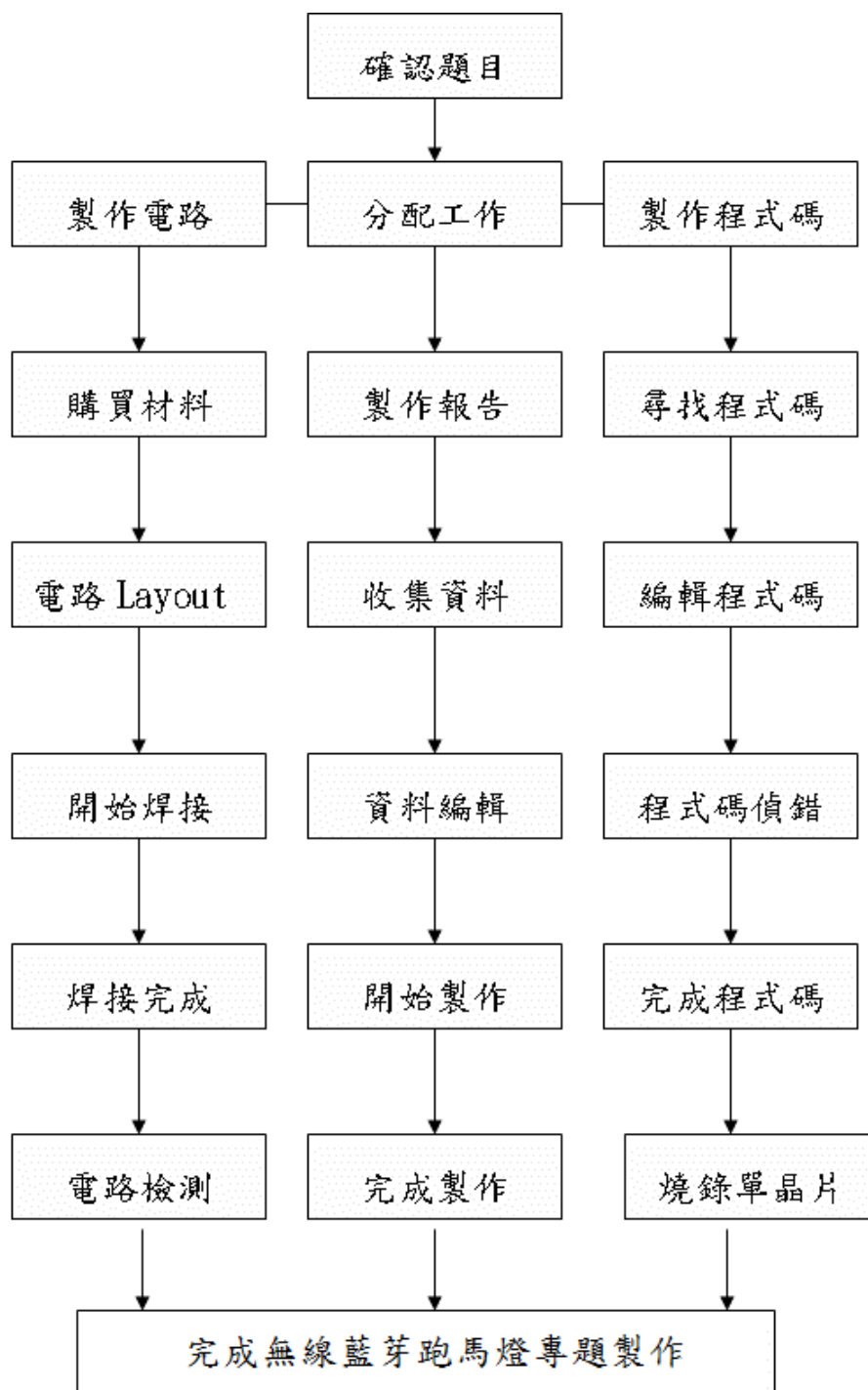


圖 3-1-1 製作方法及步驟

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		Arduino 進階控制研究	
科別／年級		資訊科三年級	
專題名稱	中文名稱	藍芽無線控制跑馬燈	
	英文名稱	Bluetooth marquees	
專題內容簡述		<p>透過 Arduino UNO 板的線路配置、以及 MAX7219 點矩陣的時脈研究，運用 Arduino IDE 設計以及撰寫一個無線跑馬燈的專題，我們選擇運用手機應用程式將文字軟體解碼成 HEX 碼再傳送到藍芽接收經由 Arduino UNO 與 MAX7219 LED 進而顯示文字、能應用在生活中需要立即修改跑馬燈文字的時候。</p>	
指導老師姓名		葉忠賢	老師
參與同學姓名	吳育誠組長		鄭連豐組員
	潘冠屹組員		
專題執行日期		104 年 9 月 1 日至 105 年 5 月 31 日	

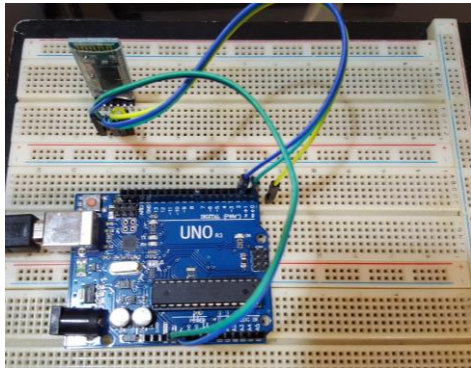


圖 3-1-2 麵包板上電路模擬(一)

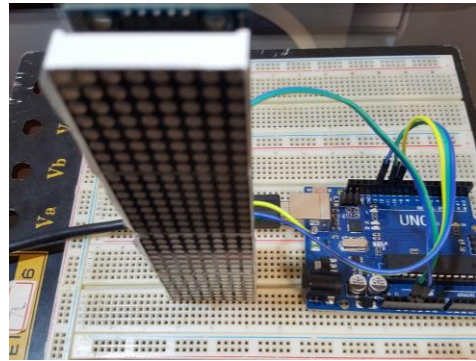


圖 3-1-3 麵包板上電路模擬(二)

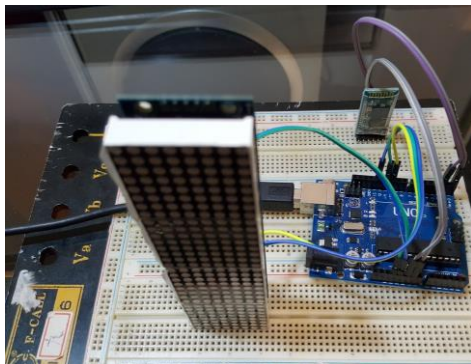


圖 3-1-4 麵包板上電路模擬(三)

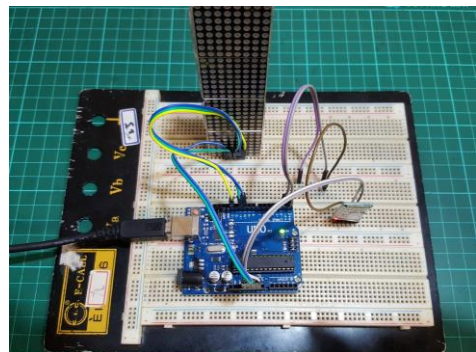


圖 3-1-5 麵包板上電路模擬(四)

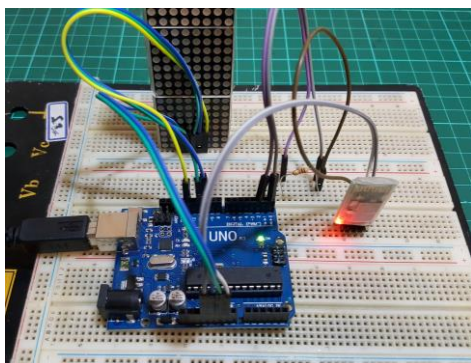


圖 3-1-6 麵包板製作完成

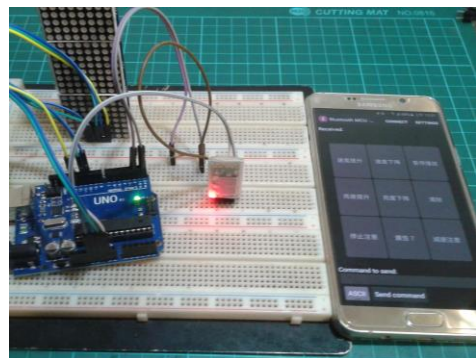


圖 3-1-7 測試麵包板有無功能

```

The UART communication happens at 115200 baud.
Written by: Dajendra Bhatt
*/
//
// 1. 注意藍牙模組 ESP8266 插板位置，在 getty() 程式
// 2. 使用 Arduino IDE 3.0 以上編譯上傳，請務必安裝 USB
// 3. ESP8266、ESP-01 擴強光緒請用 AT 轉 AT 軟體才可使用此程式
//
// Result: Work !!!
//
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_NeoPixel.h>
int pinCS = 10; // Attach CS to this pin, DIN to MOSI and CLK to SCK (cf http://arduino.cc/en/Reference/SPI)
int nPixels = 64; // Number of LEDs in the strip
int nChannels = 3; // RGB
Adafruit_NeoPixel strip = Adafruit_NeoPixel(nPixels, pinCS, nChannels, NEO_GRB + NEO_DMA);
strip.begin(); // Initialize the strip
//
// Matrix dimensions
int numRows = 8; // Number of rows
int numCols = 16; // Number of columns
//
// Matrix data
char matrix[numRows][numCols];
//
// Matrix settings
int scrollSpeed = 50; // Default scroll speed (milliseconds)
int space = 1; // Space between two characters
int width = 0; // The font width in pixels
bool invert = false; // Invert the matrix
bool dataAvailable = false;
char input[5];
int count = 0;
int time = 0;

void setup() {
  matrix.setIntensity(1); // Use a value between 0 and 15 for brightness

```

圖 3-1-8 藍芽無線控制跑馬燈程式碼

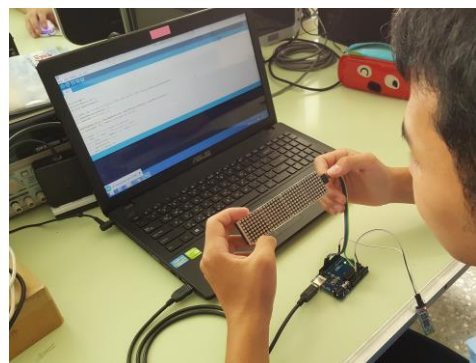


圖 3-1-9 編寫程式碼

(一)藍芽無線控制跑馬燈功能及設定

透過這個專題製作，去了解如何運用藍芽模組在Arduino上進行資料傳輸，當在使用手機App時，能進行遠距離操控，且經由透過程式碼，來更改無限跑馬燈的字幕。藍芽無線控制跑馬燈的表示方法：當開啟跑馬燈時自動顯示預設文字，可進行字幕速度、亮度等設定，也可以立即修改跑馬燈字幕。

本專題的內容由Arduino為基礎的電路架構，其中功能、設定：

1.HC-06藍芽模組設定：

- (1) 送電時，藍芽模組開啟。
- (2) 設定藍芽模組速率，使藍芽模組與Arduino連線。
- (3) 資料庫會自動編譯藍芽模組接收到的文字。

2.Arduino理論操作：

- (1) 透過Arduino UNO，了解Arduin的操作。
- (2) 透過 Arduino IDE 來進行程更改的操作。

(二) 硬體電路圖:藍芽無線控制跑馬燈

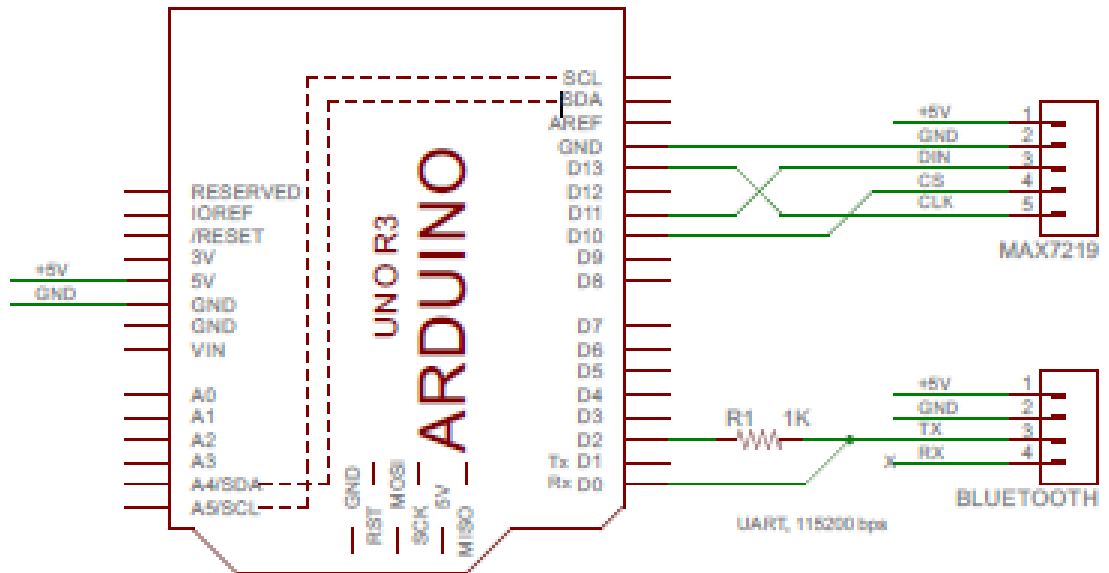


圖 3-1-10 藍芽無線控制跑馬燈電路圖

由 Arduino IDE 中撰寫程式碼，電路中的 MAX7219 點陣模組接收資料與輸出，經由藍芽模組接收訊號，使 MAX7219 點陣模組產生作用，當手機 App 輸出資料，由藍芽模組接收再傳至 MAX7219 這塊 IC 上編譯。。

(三) 藍芽無線控制跑馬燈電路板完整圖及材料表：

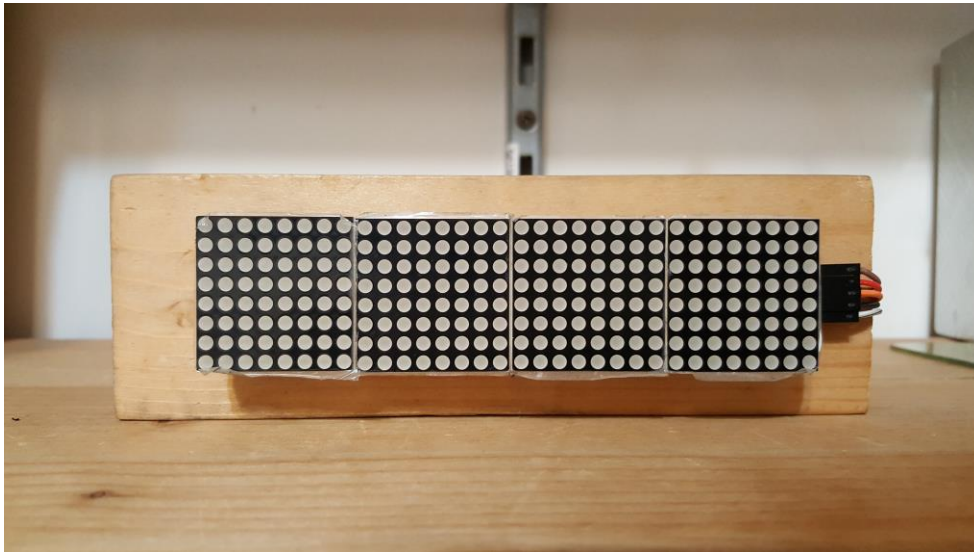


圖3-1-11 藍芽無線控制跑馬燈完整圖(正面)

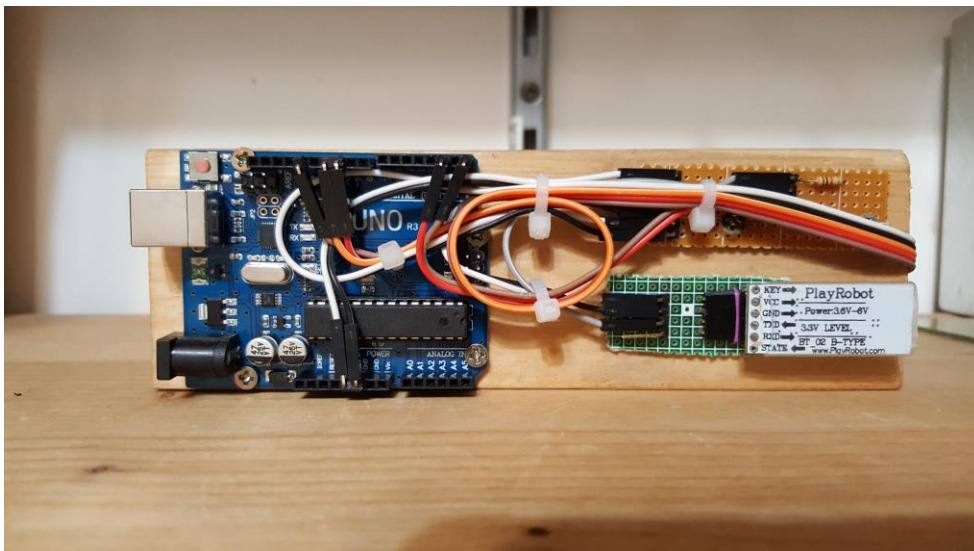


圖3-1-12 藍芽無線控制跑馬燈完整圖(背面)

表3-3-2 藍芽無線控制跑馬燈之材料表

材料名稱	規格	單位	數量	備注
8*8模組	MAX7219	個	4	
杜邦線	公對母	個	1	(包)
杜邦線	公對公	個	1	(包)
杜邦線	母對母	個	1	(包)
Arduino	UNO	個	1	
藍芽模組	HC-06	個	1	
USB Cable	Type-B	條	1	
電阻	1K Ω	個	1	
插銷	公插	個	1	(包)
電路板	900孔	片	1	(裁切)
木板	松木板	塊	1	
螺絲	平頭螺絲	個	1	(包)

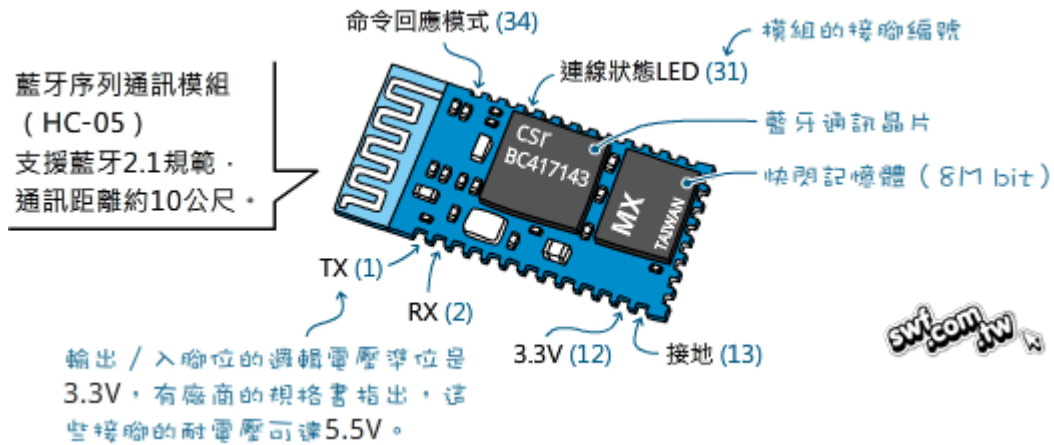


圖3-1-13 藍芽接腳圖

在高雄的各電子材料行都可以買的到這些東西，經由我們小組討論後由選用網路訂購材料，三天後收到材料便開始我們的專題製作，藍芽模組是接收肉眼所看不到的東西、手機App發射訊號出去後由藍芽模組接收就能達成所要的目的。

小組分工的配置：

育誠負責製作硬體配置，過程中亦有使用網路資料當成參考資料，讓專題能更加順利完成，而連豐負責程式碼的編寫，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。

冠屹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，在去製作文書報告，如有發生困難時可以找小組討論或者找老師諮詢，解決你的所不懂的事情。

連豐負責去訂購電路中所需之零件，在讓育誠看著Layout圖去完成我們的專題製作的成品，接著製作簡報讓連豐上台發表此專題內容。

肆、製作成果

我們小組決定題目藍芽無線控制跑馬燈，開始製作並且模擬麵包板、設計 Layout圖，進而去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藍芽無線控制跑馬燈製作過程(一)

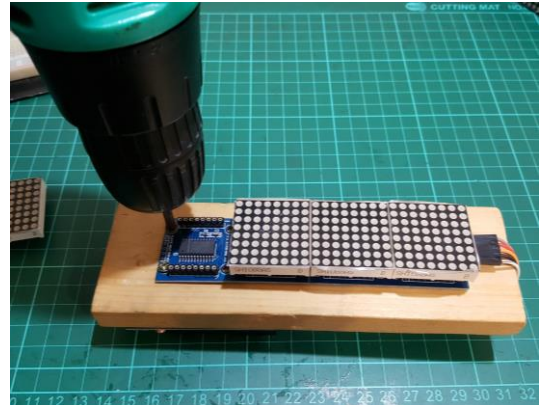


圖4-1-2藍芽無線控制跑馬燈製作過程(二)

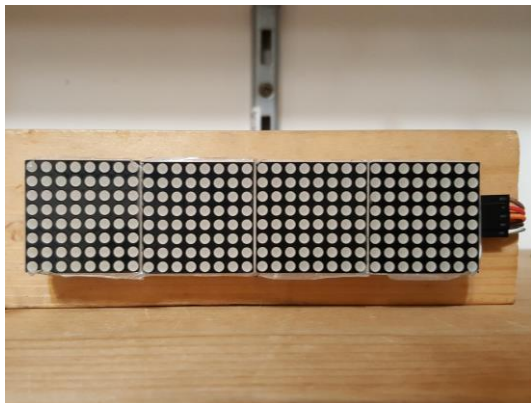


圖4-1-3藍芽無線控制跑馬燈成品圖



圖4-1-4藍芽無線控制跑馬燈測試功能



圖4-1-5製作成品過程

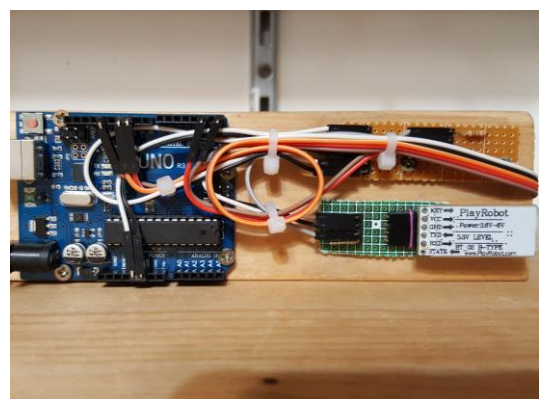


圖4-1-6完成藍芽無線控制跑馬燈

操作說明如下:

- 1、首先先進入Android手機的藍芽設定選項
- 2、開啟藍芽，並搜尋，待搜尋到“rudy”，並輸入“1234”作為配對密碼
- 3、配對成功後打開Android手機上已安裝的應用程式“Bluetooth MCU”
- 4、進入Bluetooth MCU程式主畫面後，點選“CONNECT(連接)”
- 5、如果應用程式要求藍芽使用權限，請點選“允許”
- 6、接下來請點選“rudy”，並按下方的“Select(選擇)”
- 7、連接後即會顯示“Hold On(請稍後)”的字樣，稍等一下
- 8、連線成功後，即可開始使用藍芽無線跑馬燈



圖4-1-7 Bluetooth MCU介面圖

伍、結論與建議

這次的專題製作使我們學到許多課程外的知識，將課程學到的知識與課程外的結合，所做出來的專題就是我們的成果，將自身能力發會到極限，在製作過程中邊思考、嘗試、研究，為了完成目標中的成品，發揮自己所長，也為了讓學弟妹看到我們的成就。

一、結論

- (1) 學校裡所學的理论為主，將理論結合電路應用在日常生活中，提升學習的樂趣。
- (2) 透過專題製作，以邊製作邊學習的方式來獲得新知識，整合藍芽相關的資料傳遞方式，以及矩陣的顯示原理，再將獲得的知識整合為一個專題。
- (3) 進行分工合作，找出各個組員的專長，將此專長發揮到極限，以此完成此專題。
- (4) 透過此專題了解市場需求，將此專題結合到日常生活所需，擴展更方便的生活。

二、建議

- (1) 藍芽模組接收訊號距離有限，須注意距離。
- (2) 將成品更輕量化，達成隨身攜帶這個條件。
- (3) 專題製作中著重能從做中學學習，需要團隊每位成員共同配合參與，並且分工合作一齊完成。在專題製作中遇到任何問題可先自行搜尋資料再來請教老師，立即解決問題並定期與老師討論進度，以期達到研究學習之成效。
- (4) 科技始終來自人性，如何達成我們心中的成效，來自我們努力的程度，為了不被淘汰，我們將會持續學習。

參考文獻

1. 王小建，2011 單片機設計應用，清華大學出版。
2. Tom Igoe，2014，互動裝置專題製作，歐萊禮出版。
3. 徐毅、張正賢，2015，主題式單晶片原理與專題製作，台科大出版。
4. 趙英傑，2014，超圖解Arduino 互動設計入門，旗標出版。
5. 楊明豐，Arduino最佳入門與應用，碁峰出版。
6. 陳明熒，2016，Arduino語音互動專題製作與應用，松崗出版。
7. 蔡朝洋，Arduino從零開始學，碁峰出版。
8. Mark Geddes，2015，Arduino自作專案大百科，碁峰出版。
9. Don Wilcher，Make：簡易的Arduino專題製作，歐萊禮出版。
10. 蕭獻賦，實用IC封裝，五南出版。

附錄一 藍芽無線控制跑馬燈程式碼

/*

Project: Scrolling Message Display using Easy Matrix modules in cascade

Description: This is demo project for Easy Matrix modules cascaded together to make a scrolling

message display. It uses Adafruit's GFX and Mark Ruys' Max72xxPanel libraries both.

The project receives display data and commands through UART. You can send a character stream through Serial Monitor

tool from the Arduino IDE. You also need a 1K resistor that is used to connect Arduino Pin 0 (Rx) and Pin 2 (External Interrupt).

This interrupts the Arduino every time a new command or character stream is received through UART so that it could respond to that.

Here are the commands Arduino recognizes.

1. Any character stream must be sent enclosed inside parenthesis.

Example: Suppose if you want to display a message "Hello! How are you doing?"

You should send it like this: (Hello! How are you doing?)

When the Arduino receives the first character as '(', it understands that the upcoming characters are display data until ')'

is received. The message is displayed as scrolling from right to left.

2. Display intensity can be incremented or decremented by sending /+ or /- respectively. Every time the Arduino receives '/',

it understands it is a command. It supports 16 brightness levels.

3. Similarly, scrolling speed can be increased or decreased by sending /< or />

respectively.

4. /p is used to pause the scrolling message. Sending this command again resumes the scroll.

5. /e is used to erase the buffer and blank the display

Connections:

Easy Matrix	Arduino
VCC	+5V
DIN	MOSI (Pin 11)
LOAD	pinCS defined in the program below
CLK	SCK (Pin 13)
GND	Gnd

The UART communication happens at 115200 baud.

Written by: Rajendra Bhatt

```
*/  
  
//  
// ruten.proteus 2015/08/28 修改  
// 修改文字顯示的方向以符合四合一 MAX7219 LED 點矩陣驅動控制模組  
//  
// 1. 注意藍芽與 ESP8266 通訊速度，在 setup() 修改  
// 2. 使用 Arduino IDE 1.5.8 以上編譯通過，其餘版本未測試  
// 3. ESP8266, ESP-01 需預先燒錄 UART 轉 WiFi 韌體才可使用此程式  
//  
// Result: Work !!!
```

```

//

#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>

int pinCS = 10; // Attach CS to this pin, DIN to MOSI and CLK to SCK (cf
    http://arduino.cc/en/Reference/SPI )
int numberOfHorizontalDisplays = 4;
int numberOfVerticalDisplays = 1;

Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays,
    numberOfVerticalDisplays);
int scrollspeed = 50; // Default scrollspeed (milliseconds)

int spacer = 1; // Space between two characters
int width = 5 + spacer; // The font width is 5 pixels
boolean inChar = false, NewData = false, pause = false;
boolean dataAvailable = false;
char inputString[512];
int count = 0, BTvalue = 5;

void setup() {

    matrix.setIntensity(1); // Use a value between 0 and 15 for brightness

// Adjust to your own needs
    matrix.setPosition(0, 3, 0); // The first display is at <3, 0>
    matrix.setPosition(1, 2, 0);

```

```

matrix.setPosition(2, 1, 0);
matrix.setPosition(3, 0, 0);
matrix.setRotation( 0, 3 );
matrix.setRotation( 1, 3 );
matrix.setRotation( 2, 3 );
matrix.setRotation( 3, 3 );

matrix.fillScreen(0);
matrix.write();

Serial.begin(115200); // serial communication initialize

// Using interrupt 0 on digital pin 2.
pinMode(2, INPUT);
digitalWrite(2, LOW);
attachInterrupt(0, serialInterrupt, CHANGE);

}

void loop(){
  if(dataAvailable){
    display_data();
  }
}

void display_data(){
  for ( int i = 0 ; i < width * count + matrix.width() - 1 - spacer; i++ ) {

    matrix.fillScreen(0);

```

```

int letter = i / width;
int x = (matrix.width() - 1) - i % width;

int y = (matrix.height() - 8) / 2; // center the text vertically

while ( x + width - spacer >= 0 && letter >= 0 ) {
    if ( letter < count ) {
        matrix.drawChar(x, y, inputString[letter], HIGH, LOW, 1);
    }

    letter--;
    x -= width;
}
matrix.write(); // Send bitmap to display
if(!dataAvailable){
    i = width * count + matrix.width() - 1 - spacer;
    matrix.fillScreen(LOW);
    matrix.write();
}
if(NewData){
    i = 0;
    matrix.fillScreen(LOW);
    matrix.write();
    NewData = false;
}
while(pause == true){
    delay(0);
} // Pause here if pause command is received

delay(scrollspeed);

```

```

    }
}

// Volatile, since it is modified in an ISR.
volatile boolean inService = false;

void serialInterrupt(){

    if (inService) return;

    // You was not in service. Now you are.
    inService = true;

    // Reenable interrupts, to allow Serial to work. We do this only if inService is false.
    interrupts();
    while(!Serial.available());
    while (Serial.available()) {
        char ch = Serial.read();
        if(ch == '{'){
            count = 0;
            inChar = true;

            while(inChar){
                if (Serial.available()) {
                    ch = Serial.read();
                    if(ch == ' '){
                        inChar = false;
                        dataAvailable = true;
                    } else{
                        inputString[count] = ch;

```

```

        count++;
    }
    if(count > 0) {
        NewData = true;
    }
}
}
}

if(ch == '/') { // Command mode
    inChar = true;
    while(inChar) {
        if (Serial.available()) {
            ch = Serial.read();
            // '/' is for brightness increment
            if(ch == '+') {
                if(BTvalue < 15) {
                    BTvalue ++;
                    matrix.setIntensity(BTvalue);
                }
                break;
            }
            // '/' is for brightness decrement
            if(ch == '-') {
                if(BTvalue > 0) {
                    BTvalue --;
                    matrix.setIntensity(BTvalue);
                }
                break;
            }
        }
    }
}
}
}

```

```

// '>' is to slow down scroll
if(ch == '>'){
    if(scrollspeed < 500) {
        scrollspeed = scrollspeed + 50;
    }
    break;
}

// '<' is to make scroll faster
if(ch == '<'){
    if(scrollspeed > 50){
        scrollspeed=scrollspeed-50;
    }
    break;
}

// 'e' is to erase display
if(ch == 'e'){
    dataAvailable = false;
    break;
}

// 'p' is to pause display
if(ch == 'p'){
    if(pause == false){
        pause = true;
    }
    else {
        pause = false;
    }
}

```

```
        break;
    }

    else {
        break; // Unrecognized command
    }
}
}
} // while(serial.available)
inService = false;
}
```