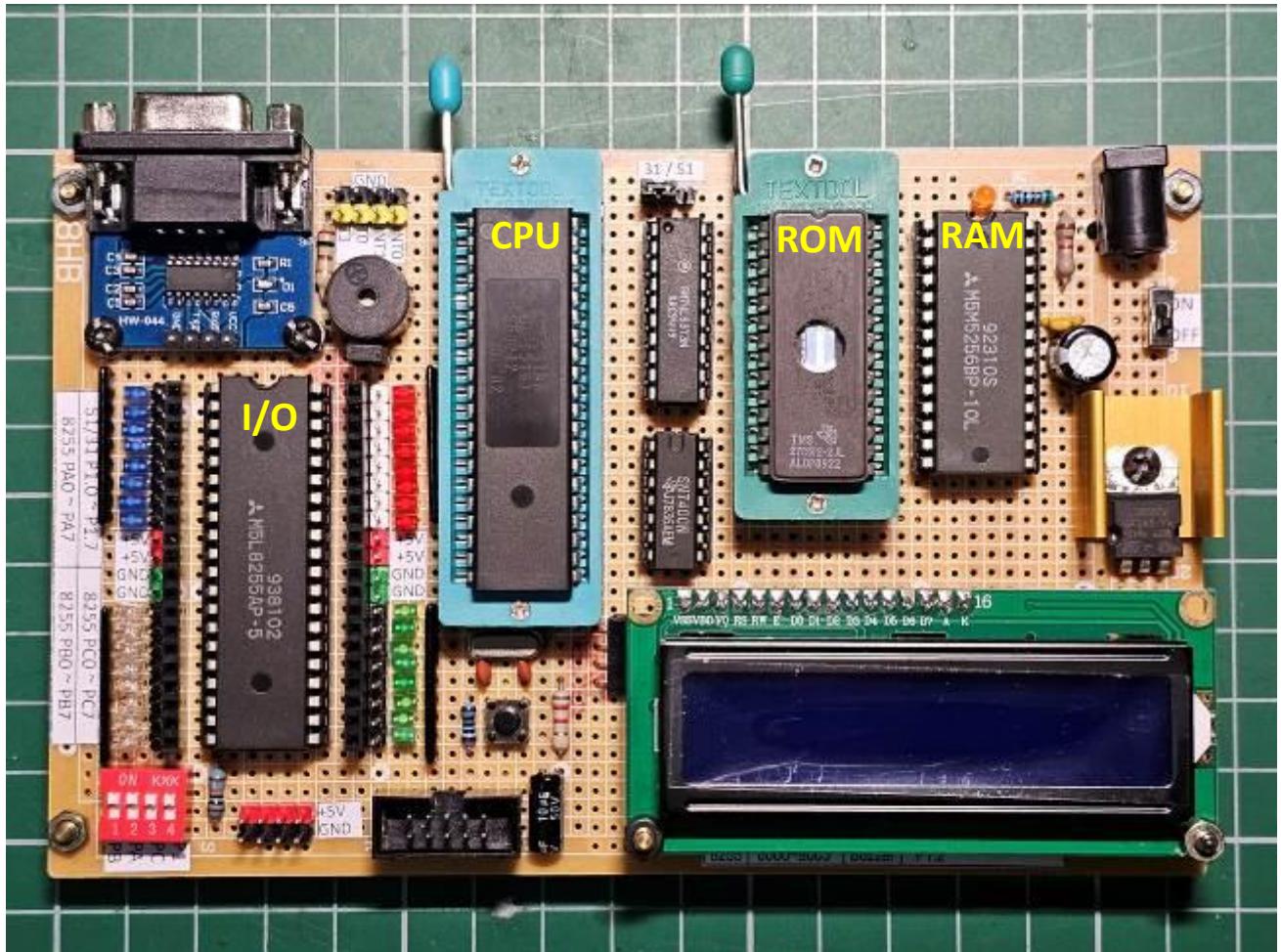


# 製作一個 8031/51 單板微電腦

原來將近 50 年前的 Z80 一直生產到 2024 年 5 月啊，既然這麼經典又歷史悠久的晶片還在生產，那我們前面介紹了單晶片的發展由來，然後手頭邊也有一些舊零件，我們就組裝一個以 8031/51 為主的單晶片系統來向經典致敬吧~。



## 一、系統規劃

如上圖，這樣一個符合有 CPU、ROM、RAM、I/O 的微型電腦系統，稱作單板電腦(Single-Board Computer, SBC)，雖然沒什麼大用途，但它很適合新手了解一個電腦系統硬體的運作，包括 CPU 各個接腳的訊號模式，ROM、RAM 如何讀取、寫入，I/O 如何控制，以及各個元件的位址解碼電路等，當它通電後依照我們的程式跑起來，LED 閃閃發光，還是很有成就感的。

這樣沒啥特定目的，只把 IO 腳接出來、通用的板子有時也稱實驗板或開發板，之後可以利用它的 I/O 來控制其他元件或感測器，就像我們使用現成的 Arduino 一樣。

我們選用 8031/51 當 CPU 是因為 8031 是一顆沒有 ROM 的單晶片，一定要外接 ROM，它的 RAM 也很小，大一點的資料應用也是需要外接 RAM，很符合一般 CPU 的狀況，如早期的 Z80、8080/8085、8086/8088 等，而且 MCS-51 系列的接腳都有包含 Address Bus、Data

Bus、Control Bus 的功能了，可以很輕鬆地建立起一個電腦系統。

另外最大的好處是發展環境較簡單建立，可以先使用 AT89S51 (Flash ROM 版本的 8051)，加上 Keil C51 環境用 C 語言來寫程式，然後直接燒錄來測試板子的各項功能是否正確，再將正確的程式燒錄到 ROM，然後切換到 31 模式或換成 8031 來執行，否則每次測試程式就要燒一次 EPROM，然後抓 bug，紫外線清除，再燒...，真的會昏倒吧。

8031/51 外接記憶體時 ROM 和 RAM 的位址是分開獨立的，所以 ROM 最大可以到 64KB，RAM 也是可到 64KB，但因為 8031/51 僅支援記憶體對映輸入輸出 (Memory-mapped I/O, MMI/O，簡稱為記憶體對映 I/O)，不支援埠對映輸入輸出 (Port-mapped I/O, PMI/O，也叫作獨立輸入輸出 (isolated I/O))，所以得留一點記憶體位址空間給 I/O 用，我們選用 32KB 的 RAM，應該夠了，剩下的記憶體位址給 I/O 用。

接下來就要畫線路圖了。

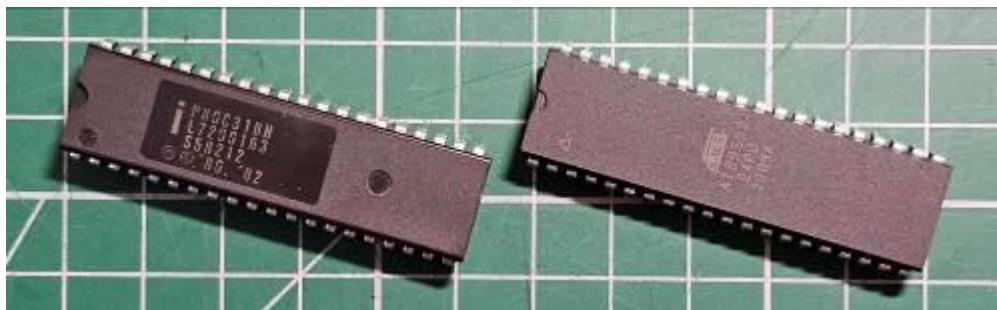
## 二、線路圖

有了線路圖才能開始施工焊接線路，如果線路不複雜，就用手畫一畫就行了，如果想要漂亮，或線路實在太多，手繪容易出錯，可以使用免費的 KiCAD( <https://www.kicad.org> ) 軟體，它是功能強大的線路圖製作軟體，完全免費，推薦！，或使用試用版的 OrCAD ( <https://www.orcad.com/orcad-free-download> )，都很方便，因為我們沒有要量產，就這麼一片，所以就不要管 PCB 佈線的部分了，會畫線路就行了，而且各元件功能不先正確跑起來就去製作 PCB 板也沒用啊，萬一線路是錯的勒，還是要先用「雛形」驗證一下的。

線路圖如附件，請參閱。主要元件及參數如下：

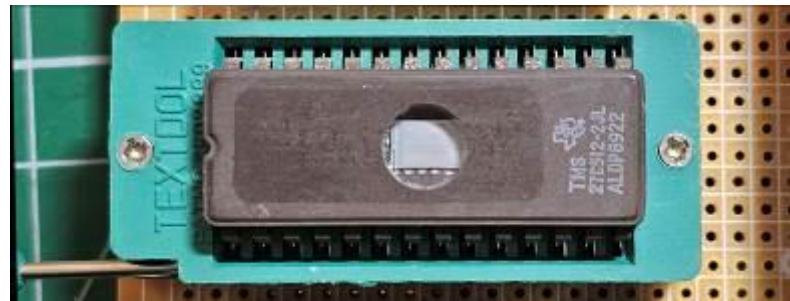
### (一)、CPU：8031 或 AT89S51。

(每次親手拿著 8031，這種工藝發展初期，引領人類步入微電腦時代的晶片，心中就充滿了懷舊的崇敬，歷史啊~)



- 最重要的是第 31 腳 (/EA)，決定了工作模式，若接地則為 8031 模式，使用外部 ROM 程式記憶體，忽略內部的 ROM。若接 Vcc，則為 8051 模式，使用內部的 ROM 來執行程式。也就是當 /EA 接地時，8051/AT89S51 等同於 8031。
- 當 8031/51 要外接記憶體時，P0 (Port 0) 為 Data bus (D0-D7) 及低位 Address bus (A0-A7)，我使用一顆 74LS373 來將位址訊號鎖住。P2 (Port 2) 為高位 Address bus (A8-A15)。P3 (Port 3) 為 Control bus 的一些接腳。P1 (Port 1) 未使用，仍可當一般 IO 腳用。

### (二)、ROM：使用 27C512 EPROM。



1. 這是一顆非常經典的 EPROM，有 64KB 的容量，反正也沒什麼成本考量，就給它用到最大吧。解碼位址為 0x0000-0xFFFF。(看圖這麼舊就知道是拆舊貨啦)。我個人覺得 EPROM 應該是最漂亮的 IC 了，因為它可以直接看到 IC 內部的晶片！
2. 8031/51 是用第 29 腳 (/PSEN) 來當作讀取訊號，和 RAM 使用/RD 來讀取無關，8031/51 自己會依照指令分辨現在要讀取程式記憶體 (ROM) 還是資料記憶體 (RAM)。

(三)、RAM：使用 M5M5256 SRAM。



1. 這也是一顆非常經典的 SRAM，因為剛好手邊有啦，就是它了。有時通稱 62256 系列，它有 32KB 的容量。解碼位址為 0x0000-0x7FFF。
2. 它由 8031/51 的第 16 腳/WR 及第 17 腳/RD 來控制讀寫。

(四)、I/O：使用 8255 PIO。



1. 這是一顆使用非常廣泛的 IO 晶片，雖然已經歷史悠久了，但仍然可以在很多應用上看到它。它有三個 8bit 的埠：PA、PB、PC，每個 IO 腳都可以獨立控制使用，其中 PC 可以拆開來配合 PA 和 PB 做其它更複雜的應用。
2. 它有一個控制暫存器，用來決定 8255 的工作模式，另外 PA、PB、PC 各有一個資料暫存器，用做輸出入資料緩衝。解碼位址如下：

A <sub>1</sub>	A <sub>0</sub>	Port selected	Address
0	0	Port A	0x8000
0	1	Port B	0x8001
1	0	Port C	0x8002
1	1	Control register	0x8003

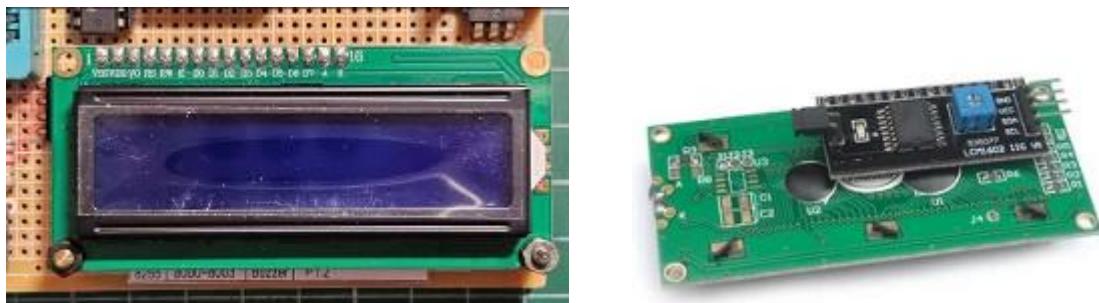
3. 我把每個 IO 腳都直接接上了 LED，方便測試及將來控制外部元件時可以方便視覺檢測訊號是否正常。

#### (五)、通訊介面：RS232 模組。



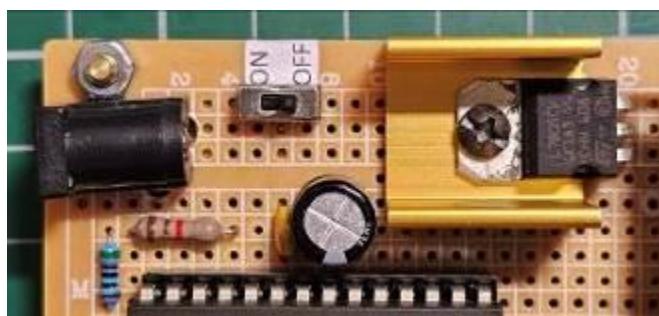
1. 使用了現成的 DB-9 接頭 MAX232 模組來轉換 RS232 訊號（當然要自己接線做一組也是可以啦），因為這只是一個通訊週邊，不是系統的重點，直接用現成的模組就好，這樣我們的板子就可以和電腦溝通了。
2. 雖然目前大部分的 PC 裝置已經不使用 DB-9 的接頭來傳輸訊號（就是以前的 COM1），但 RS232 的訊號抗雜訊較好，傳輸距離相對較遠，還可以把接頭鎖住，在複雜的環境不會一扯就掉，在工業上還是蠻有用的。

#### (六)、顯示介面：使用 1602 LCD。



1. 1602LCD 是 Maker 們常用的顯示器，它可以顯示 16 字\*2 行，內建編碼，只要把要顯示的字元送過去就行了。它的介面分 8bit 和 4bit，這裡為了節省 IO 腳，我加上了專用的 IIC 模組，讓 LCD 透過 IIC 協定兩隻腳就可以控制了。

#### (七)、電源供應：使用 7805。

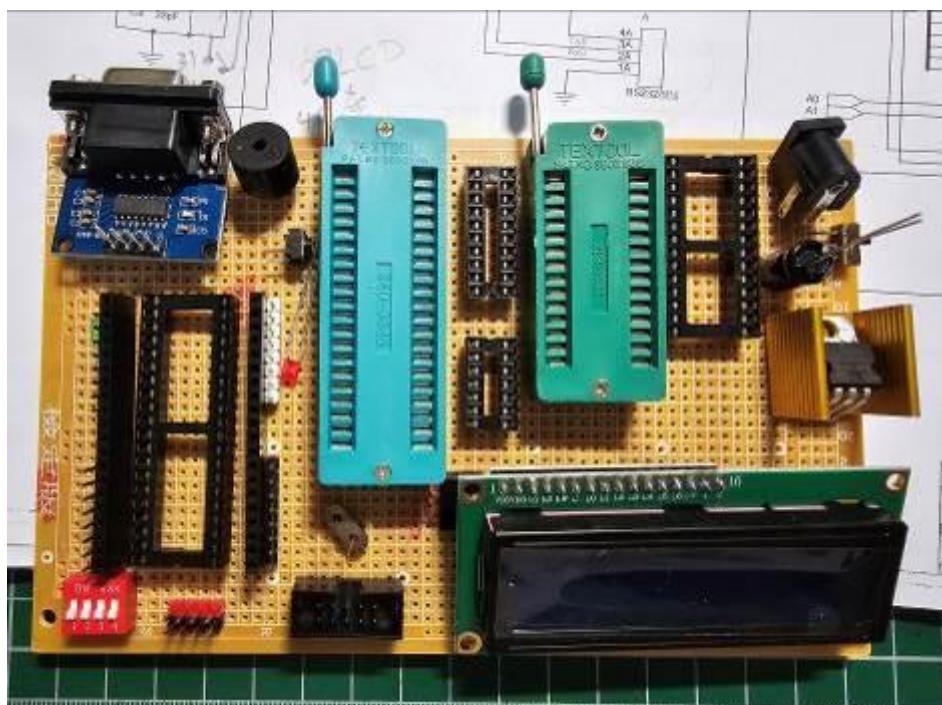


1. 我們需要一個簡單的電源供應電路，在不使用 USB 供電時可以外接 9~12V 的電源，7805 是一個非常簡單的電源穩壓 IC，只有三隻腳，輸入、輸出、接地，只要輸入電壓在額定的範圍內，都可以提供穩定的+5V 輸出。

### 三、零件布置

依照自己設計的線路圖，把零件找齊，然後試著擺放一下，看看大概需要多大的空間，因為我們要純手工來打造，最方便的是使用洞洞板，或稱萬用板，這些販售的洞洞板是有固定尺寸的，找一個大小最接近我們需求的。

還要考慮元件之間布線的問題，好的元件位置可以減少拉線焊接的數目及長短，也不能太接近，還要考慮留下的空間夠不夠走線，可別連烙鐵頭都伸不進去了。還好我們不是高頻應用的電路，對於線與線之間的訊號干擾可以暫不考慮。



### 四、動手實作

這裡需要手工焊接線路，烙鐵、焊錫是必備的，只要熟練焊接技巧，線路、各 IC 接腳不要看錯，照著線路圖施工應該是沒有什麼問題，從底板背面看 IC 腳位是相反的，要留意。

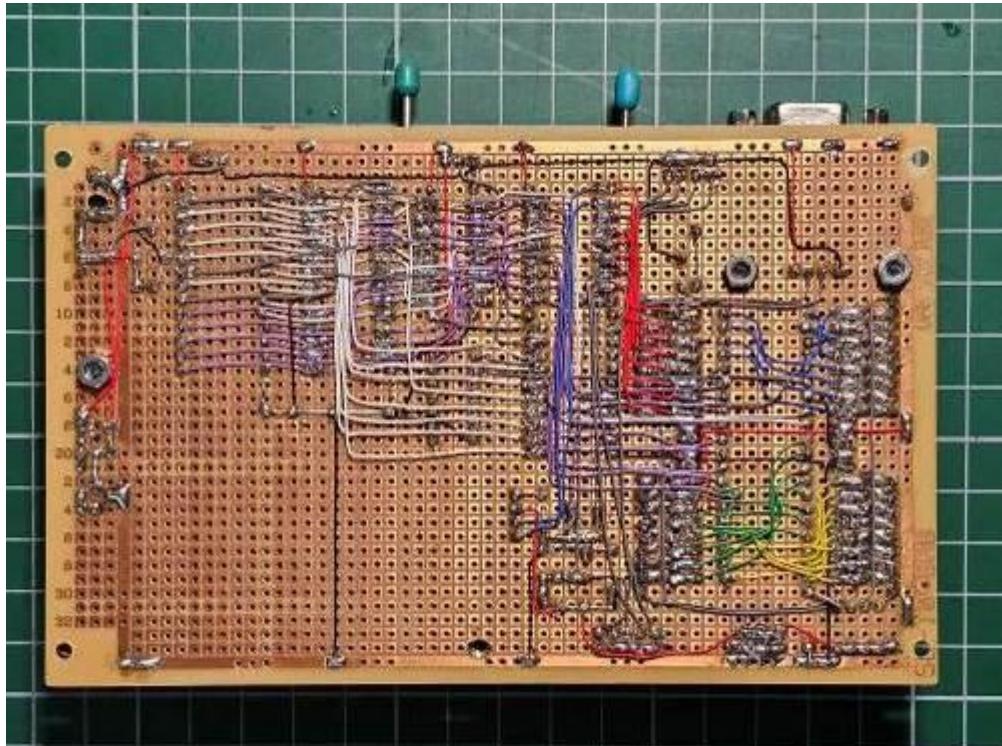
我會先把所有元件接點都焊好後再慢慢拉線。這種很細很細不用剝皮直接焊的線材俗稱 OK 線，很方便，好好走線不要讓它太凌亂，不然萬一要查修就麻煩了。

我習慣每焊好一個點就用力扯一下，確定不是冷焊，不然這麼多的接點接觸不良要查修要花很大的功夫，也會導致系統不穩定。至於



顏色建議多買一點，好處是可以用線的顏色來區分這是 Data bus 還是 Address bus，還是電源線等，像我習慣+5V 都用紅線，接地都用黑線，一來容易區分，二來看上去也美觀（我自己覺得啦）。

在焊接一個段落後我就會用三用電表測試一下剛才焊的每一個接點是否都有正確連接，是否有跟旁邊的其它接腳短路，不要太自信，發生斷路、短路的情況都是有可能的，這個階段要有耐心，要細心。最後都焊接完後會再全部檢查一次線路才準備通電。



線路圖看起來線很多，實際焊起來還好。在不洗 PCB 板，也沒什麼雕刻機啥的輔助下，這是純手製作常用的方法，一旦確定設計合理，線路正常，元件都依設計運作，再去考慮量產或洗 PCB 板的問題。

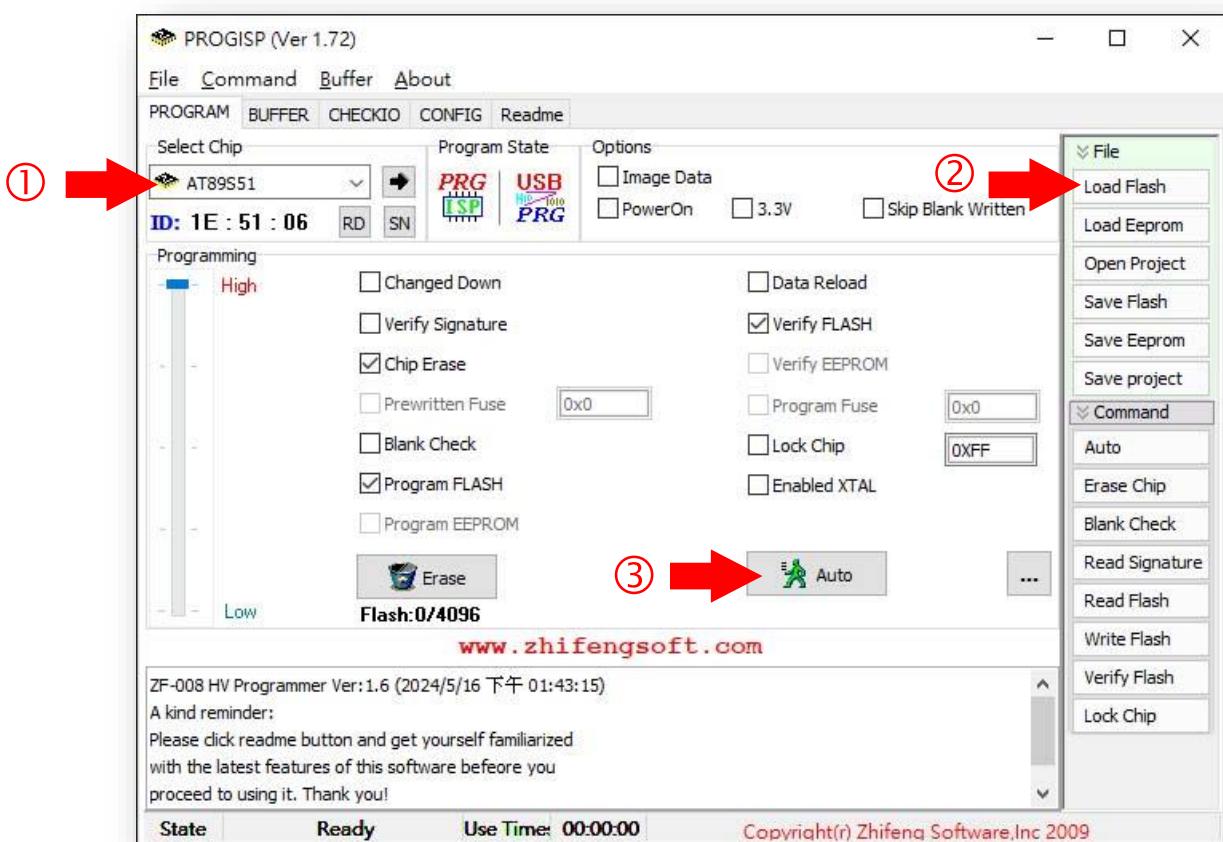
## 五、程式及測試

(一)、先用 AT89S51，ROM 先不要插，這兩個元件使用了活動 IC 座就是因為可能會換來換去，跳線開關接到 51 模式。第一次通電後先靜置幾秒，看看有沒有哪裡冒煙、有沒有異味，摸一下各主要元件有沒有異常發熱的，如果都正常，那應該就沒有什麼大問題，可以開始寫程式測試各功能。

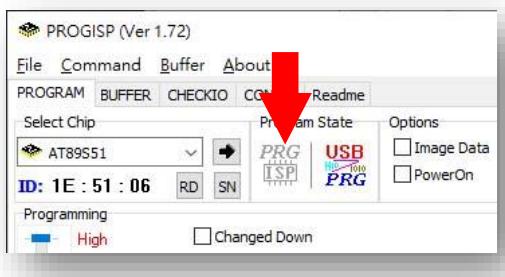
(二)、AT89S51 支援 In-System Programmable (ISP)，即不用拔下 AT89S51 另外用燒錄器燒錄，直接在系統板子中即可進行燒錄，它使用 USBISP 介面，透過 USB 即可進行燒錄，網路上可以很容易買到 USBISP (也稱 USBASP) 燒錄器。只要在板子上加上 10pin 或 6pin 的牛角座來連接 USBISP 就可以了。



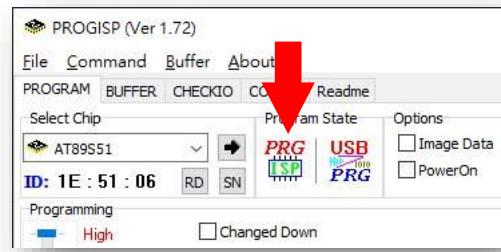
燒錄軟體我是使用 progisp 這個免費軟體（可以在這裡取得：<https://github.com/ioelectro/avr-progisp-programmer/tree/main/Software>），或網路搜尋一下，很多地方可以下載。不需要安裝，也不需要驅動程式，除非你的電腦不認得 USBISP 裝置。只要①選好晶片型號，②載入要燒錄的程式，③按下 Auto 即可。至於詳細設定及使用請查詢一下網路教學。

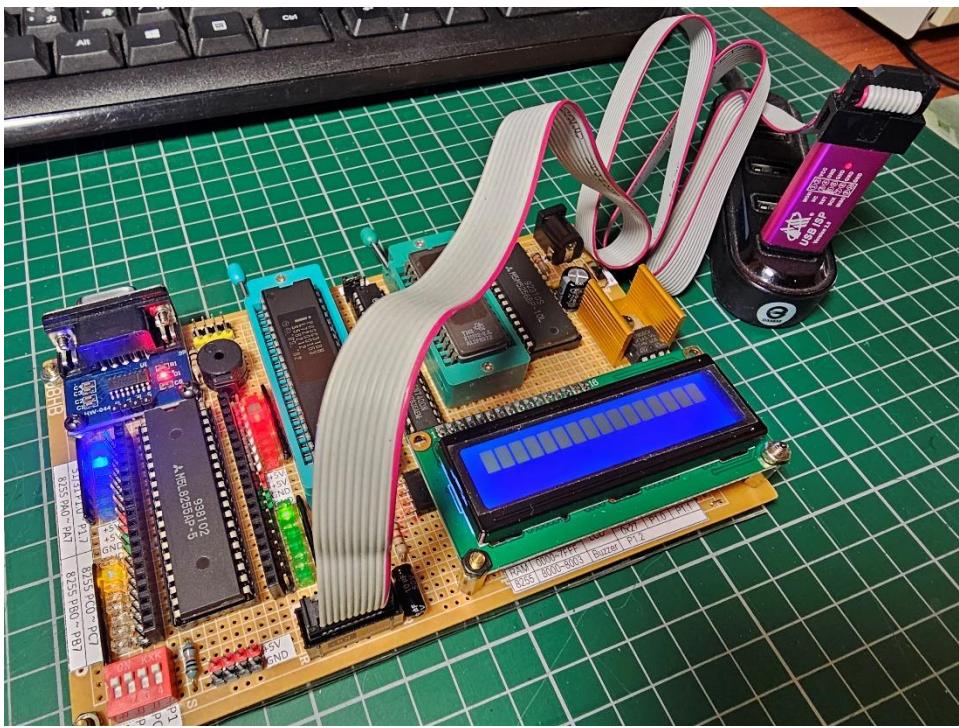


沒有插入 USBISP 或電腦無法辨識



正確測得 USBISP，可使用





(三)、至於編寫程式，不管使用什麼軟體，最後能生成 Intel 16 進制檔 (\*.hex) 或二進制檔 (\*.bin) 給燒錄軟體即可。

(四)、工科的學生建議要會組合語言，那是了解及控制硬體的精隨。其他愛好者一般使用 C 語言來實做會比較方便。8051 的 C 語法和一般的 C 語言差不多，但還是有一些差異，最重要的是一些為了控制硬體而多出來的資料型態，還有晶片各個 PORT 或暫存器的特定名稱（都已經事先定義好了），你可以參看 reg51.h 或 reg52.h 。

## 六、心得

現今的晶片工藝已經進展到手工無法製作了，我個人能“純手工”做一個微電腦系統的極限大概就是 Intel 80188 了，再上去的 CPU 如 80286、80386 就看網路神人了。

如果你能夠理解這樣一個微電腦系統的設計及運作，相信其他如 Arduino、MicroBit、Raspberry Pi、NodeMCU 等都不是問題。

雖然現在 Arduino、ESP32 等開發板都已經做得小巧精緻，也有蓬勃發展的應用環境和支援，但我們要發展一些自己的東西時，總不能用 Arduino 插一堆杜邦線來當作成品吧，最後還是要直接使用單晶片來設計並實現我們的創意，才有機會大量生產啊。

另外要學會看各個晶片的手冊，不管是幾頁，要耐心查閱，一般廠商都會把他們的產品如何使用說明得很清楚，包含各項規格、設定等，甚至還會提供範例接線圖，是很有用的資料。

