1. **教學內容/目標。**

* The study of the design of microprocessor-based systems.
* 嵌入式系統。
* 組合語言。

1. **說明微處理機與微控制器之間差異與定位。**

**微處理機**計算力強，價格貴，只有ALU(算術邏輯單元)、CCU(中央控制單元)。

**微控制器**功能多，價格便宜。各個部門都有，ALL IN ONE的設計。

1. **pendant的微處理器和AT家族的控制器有什麼不同？**

AT家族的控制器本身已經有輸出跟輸入，

而Pendant微處理器的輸出跟輸入要另外接在PC版上。

1. **請說明AT89C/S51內部有多少程式記憶體？有多少資料記憶體？有幾組Timer/Counter？有幾組UART？有多少I/O接腳？**

* 內部程式記憶體4KB，最大可擴充至 64KB。
* 內部資料記憶體128Bytes，最大可擴充至 64KB。
* 2 組16位元計時／計數器〈T0、T1〉。
* 1 組全雙工串列埠〈UART〉。
* 32支(4個8位元雙向I/O〈P0、P1、P2、P3〉)。

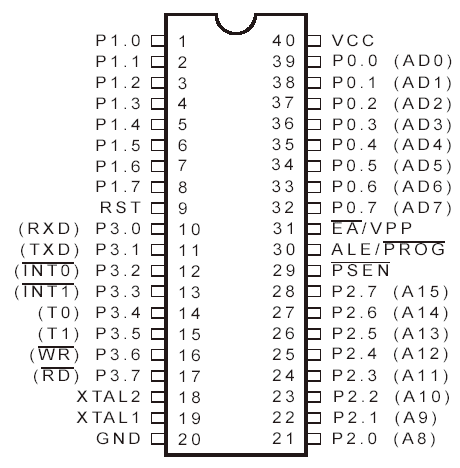
1. **請問(port0) 和(port1、port2、port3)有什麼差別？**

port0："0"為0，"1"為斷路(open dran)。

port1、port2、port3："0"為0，"1"為1。

1. **請說明下列AT89C/S51接腳功能：**

**RxD**：串列輸入。(UART的接收腳)。

**TxD**：串列輸出。(UART的傳送腳)。

**/INT0**：外部中斷 0 信號輸入端。

**/INT1**：外部中斷 l 信號輸入端。

**T0**：計時器/計數器 0 外部信號輸入端。

**T1**：計時器/計數器 l 外部信號輸入端。

**VCC**：接正電源5V。

**GND**：接地。

**/EA**：接高電位時，讀取內部程式記憶體；

接低電位時，讀取外部程式記憶體。

**/PSEN**：程式儲存致能閃控輸出信號。

**ALE**：位址閂鎖致能輸出信號。

**XTAL1, XTAL2**：接石英晶體振盪器，需接兩個30p

的電容，目的是改變整個系統時脈

( XTAL1:反相震盪放大器的輸出端

XTAL2:反相震盪放大器的輸入端)。

**RST**：重置信號輸入端。

**/WR**：外部資料記憶體寫入激發信號。

**/RD**：外部資料記憶體讀取激發信號。

1. **請說明下列AT89C/S51特殊功能暫存器之功能：**

**PCON** **(電源控制暫存器)**：控制電源的工作模式。

**SCON(串列埠控制暫存器)**：控制非同步接收傳送器(UART)的工作模式。

**SCON.TI**：UART送完資料就會設定這個旗標。

**SCON.RI**：UART收到資料就會設定這個旗標。

**TCON(計時/計數控制暫存器)**：控制計時/計數器 0 與計時/計數器 1 工作狀態。

**TMOD(計時/計數模式控制暫存器)**：設定第0個和第1個計時/計數器的工作模式 - 較高和較低的4位元完全相同 - 高位元記錄Timer1，低位元記錄Timer0。

**SBUF** **(串列資料緩衝器)**：兩個暫存器構成，一個是當作傳送資料緩衝器，另一個是當成接收資料的緩衝器。(透過P3.0、P3.1做接收和傳送)。

**IP(中斷優先暫存器)**：設定中斷源的優先權 - 對應的位元為1時，表示有較高的優先權。

**IE(中斷致能暫存器)**：指定中斷致能或禁能 - 8051有五個中斷源 - bit7為EA，若是0的話，則表示禁止所有的中斷要求。

**TH0/TL0+TH1/TL1**：組成兩組16位元計時/計數器。

**P0、P1、P2、P3**：專門做輸出及輸入的動作。

**PC(程式計數暫存器)**：記載著程式下一個待執行指令位址。

**SP(堆疊指標暫存器)**：指出堆疊空間的起始位址 - CPU reset時，SP的內含值是07h，表示堆疊從08h開始。

**PSW(程式狀態暫存器)**：記錄程式運作時，CPU各種狀態。(CY、AC、OV、P為狀態旗標位元)

**PSW.CY**：進位旗標，當做加法運算時，產生進位時，CY自動設為1，反之則為0。

**PSW.** **AC**：輔助進位。

**PSW.OV**：檢測是否有溢位。

**PSW.P**：同位位元(用來檢測是奇同位或偶同位)。

**PSW.** **RS0、RS1**：用來選擇Register Bank。

**PSW.** **F0、F1**：沒有特定功能，單純用來儲存位元。

**ACC：**簡稱A，最重要的暫存器，運算與資料轉移都透過 ACC。

**B 暫存器**：用於乘、除法中所使用到之暫存器。

**DPTR(資料指標暫存器)：**16位元暫存器，由兩個8bit的DPH、DPL組成，用來指定外部資料記憶體存取時的位址。

1. **請問Timer跟Counter有什麼相同之處、有什麼不同之處？**

**相同之處**：兩者都會往上數。

**不同之處**：Timer經過一個機器週期(machine cycle)會增加1。

Counter要從外面輸入一個訊號才會增加1。

1. **將SMOD和CF0設為1和清除為0？**

因為SMOD和CF0都不是bit addressable，所以不可以用STEB和CLR。

**SETB SMOD要改成：**

MOV A,PCON

ORL A,#10000000b

MOV PCON,A

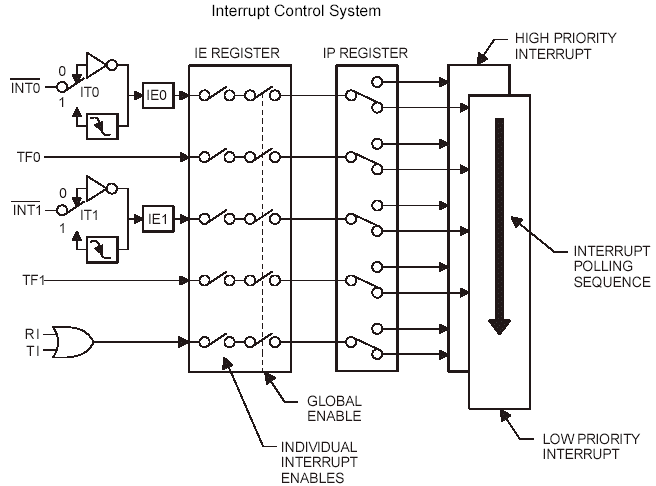
**CLR CF0 要改成：**

MOV A,PCON

AND A,#11111011b

MOV PCON,A

1. **說明8051的中斷控制機制。**



IE為中斷致能暫存器，決定是否致能中斷。

IP為中斷優先權暫存器，決定高優先權。

如同一等級優先權的中斷同時發生，先檢查到的中斷優先處裡，而檢查的方向由上而下，若優先權皆相同，INT0優先權最高，R1、T1優先權最低。

1. **請說明下列8051定址法/指令差異：**

* **立即定址(Immediate Addressing)**
  1. 將某個位址或暫存器直接存入一個值。
  2. 立即定址的前置符號為#。
* **直接定址(Direct Addressing)**

直接將某個位址的值傳給CPU中的某一個暫存器。

* **間接定址(Indirect Addressing)**
  1. 利用暫存器R0，R1，SP及DPTR當成指標，間接取得指標內的資料。
  2. 間接定址的前置符號為@。
* **暫存器定址(Register Addressing)**凡透過R0~R7 暫存器作資料的搬移或加減運算統稱為暫存器定址。
* **索引定址法**
  1. 利用DPTR或PC為基底值然後加上ACC所得位址。

**MOVC A,@A+DPTR與MOVX A,@DPTR**

**MOVC A,@A+DPTR**：將程式記憶體內容A+DPTR移入累加器A。

**MOVX A,@DPTR**：外部資料記憶體內容移入累加器A。

**RL A與RLC A**

**RL A**：累積器左旋一位元。

**RLC A**：含進位左旋。

**MOV A,R0與MOV A,@R0**

**MOV A,R0**：將暫存器 R0內的資料存入累加器A。

**MOV A,@R0**：把內部 RAM 位址的資料存入累加器 A。

**MOV A,#10h與MOV A,10h**

**MOV A,#10h**：將常數值”10”存入累加器A。16進制。

**MOV A,10h**：把位址10H的資料存入累加器A。

**MOV A,#10h與MOV A,#10**

**MOV A,#10h**：將常數值”10”存入累加器A。16進制。

**MOV A,#10**：將常數值”10”存入累加器A。10進制。

1. **請說明下列8051指令/假指令的動作**

**假指令//**

**ORG:**ORG表示程式的起始位址，程式經過組譯後是由ORG所指令的位址開始存放機器碼。

**EQU**：指定EQU後面的數值給EQU前面的標記。例:VAR1 EQU 50H 即VAR1=50H

**END**：程式結束時要加END假指令，表示組譯到此結束。

**REG**：由使用者自己定義暫存器。

**VAR**：指定VAR後面的數值給VAR前面的變數。

**DB**：定義一個位元組(Byte)的資料給某記憶體位址。DB(Define Byte)。

**DW**：定義兩個位元組資料(16Bit)給某記億體位址。DW(Define Word)。

**LONG**：定義四個位元組(32Bit)的資料給某一記憶體位址。

**---------------------------------------------------------------------------------------------------**

**指令//**

**MUL AB**：累加器A乘以暫存器B，相乘結果之高 8位元存入B，低8位元存入A。

**CJNE A,#10h,SOMEWHERE**：若A與10H不相同就跳躍至SOMEWHERE。

**RET**：由副程式返回。

**RETI**：由中斷副程式返回。

**DJNZ**：減1，若不等於0，跳回後方指定的地方，若為0就往下執行。

**DJNZ RO,SOMEWHERE**：暫存器內容減1，若不等於0，則跳至SOMEWHERE。

**DJNZ RO,LOOP**：暫存器內容減1，若不等於0，則跳至LOOP。

**INC A**：累加器A內容加 1。

**MOVC A,@A+DPTR**：將程式記憶體內容移入累加器A內。

**LCALL**：長程式呼叫。

**CALL SUB**：呼叫副程式SUB。

**SETB**：將指定設為1。

**SETB P1.0**：將 P1.0 設為 1。

**SWAP A**：累加器A的高低 4 位元互相交換。

**RR A**：累加器A內容向右旋轉 1 位元。

**LJMP**：長程跳耀。

**LJMP TARGET**：遠程跳耀，跳到64K範圍內的位址TARGET。

**CLR**：清除為0。

**JB**：若被定址之位元(bit)是1，則跳往後面的標名(rel)處執行，否則直接往下一指令執行。

**JNB**：若被指定的位元(bit)是0，則跳到標名(rel)執行。

1. **請說明下列軟體之用途：**

**MIDE-51 Studio**：免費軟體、程式編譯。

**ASEM-51 v1.3**：免費的組譯程式。

**Emulator V1.0**：軟體模擬器。

**SDCC(Small Device C Compiler)**：功能強大而且免費的C語言編譯器。

1. **請說明下列積體電路用途：**

**7805**：把輸入7~25 V的電壓，調整成5V輸出的電壓，為穩壓用。

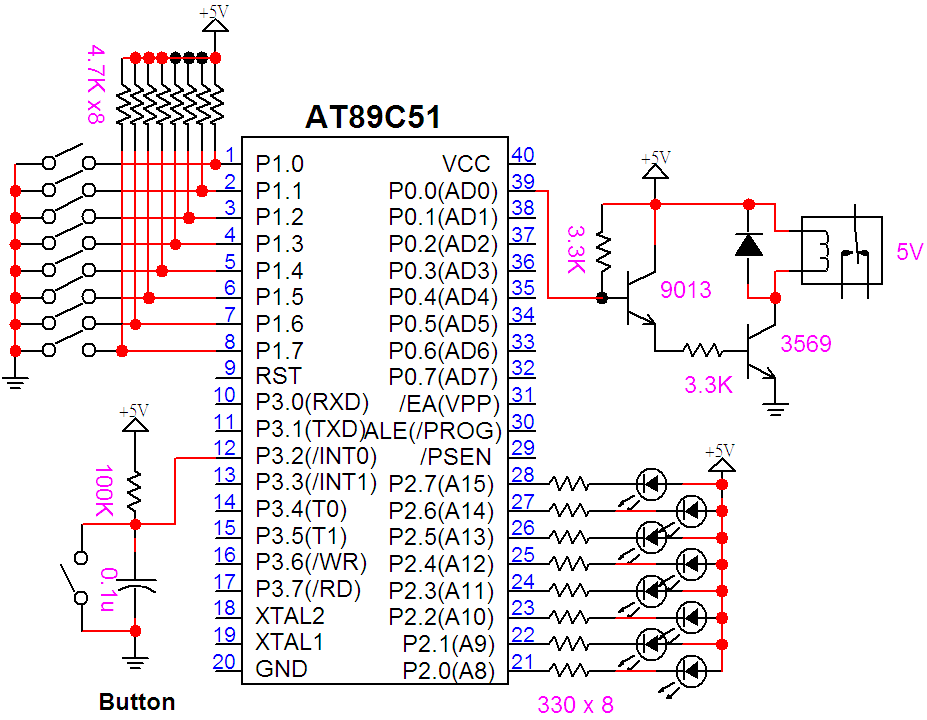
**ICL232**：為89S51的UART與RS-232連接時，必須作電壓準位的轉換。

**HIN232**：為RS-232 的傳送和接收介面 IC，用於電壓準位轉換。

**LM317**：可用來調整電壓源，也能作為定電流的裝置。

1. **請問下圖的接線哪邊有錯？**

老師可能故意把LED的電源接地，或者把開關的電阻跟電容接反，或是電晶體反接。

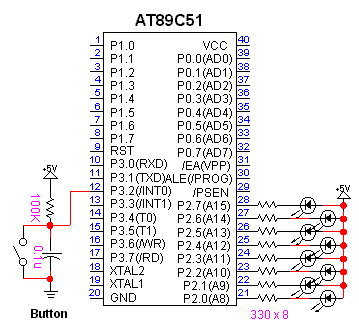


1. **中斷的處理過程跟副程式的呼叫有那些同異之處？**

**相同**：程式執行的流程會轉移到另一個地方，執行完之後再回到原來的地方繼續下去。

**相異**：控制權轉移的時機不一樣。副程式呼叫是因為執行副程式呼叫所導致的。中斷是因為外來的非同步事件所引發的。

1. **請參考以下電路，設計一8051組合語言程式片段，以肉眼可見速度閃爍LED。**

** ORG 0000h**

**REPEAT： MOV P2,#00h**

**LCALL DELAY**

**MOV P2,#0FFh**

**LCALL DELAY**

**LJMP REPEAT**

**DELAY： MOV R0,#0FFh**

**LOOP1： MOV R1,#0FFh**

**LOOP2： DJNZ R1,LOOP2**

**DJNZ R0,LOOP1**

**RET**

**END**

1. **請設計一8051組合語言程式片段，從P1、P3讀入被乘數與乘數，將乘積從P0、P2輸出。**

**ORG 0000h**

**REPEAT： MOV P1,#0FFh**

**MOV A,P1**

**MOV P3,#0FFh**

**MOV B,P3**

**MUL AB**

**MOV P0,B ;高位元**

**MOV P2,A ;低位元**

**LJMP REPEAT**

**END**