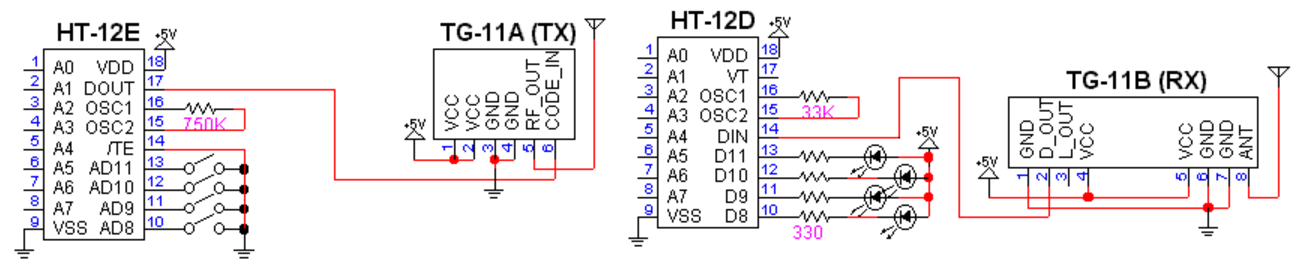
1. **請參考下列電路，說明射頻模組(RF Module)如何運作。**



**HT-12為編碼IC，其中A0~A7為RF傳輸中的密碼設定，AD8~AD11為要送出的資料，DOUT將密碼及資料送到TG-11A，TG-11A為紅外線發送模組，將接受到的資料用紅外線發送，TG-11B為與TG-11A成對的紅外線接收模組，將接受到的密碼傳送到HT-12D，HT-12D為解碼IC，根據A0~A7的密碼設定，與接收的資料核對密碼，若密碼正確則將剛設定的資料作輸出。解碼 IC 的工作頻率約為編碼 IC 的 50 倍，編碼 IC 連接 750kΩ，解碼 IC 連接 33KΩ即可。**

1. **請參考下列電路，**
2. **請說明SEEPROM(Serial Electrical Erasable Programmable Read-Only Memory)、RAM(Random Access Memory)、ROM(Read-Only Memory)、UVEPROM(UV Erasable Programmable Read-Only Memory)間之差異及優缺點。**

**SEEPROM為電子抹除式可複寫唯讀記憶體 ，用特定的電壓，來抹除晶片上的資訊，以便寫入新的資料。優點是存取快，但相對缺點是當關閉電源時則記憶體揮發，無法儲存。**

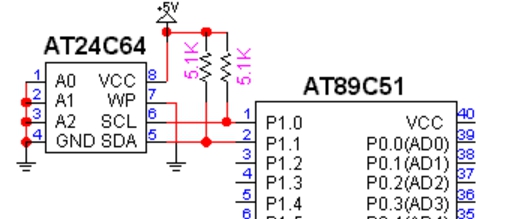
**RAM為隨機存取記憶體，是當關閉電源時就會揮發的記憶體。雖克服ROM只能寫入一次的缺點，但寫入時間極慢且抹除方式麻煩。**

**UVEPROM為紫外線抹除式可複寫唯讀記憶體 ，用紫外線，來抹除晶片上的資訊，以便寫入新的資料。大容量價格低速度，相對其他種類ROM快，缺點較新式NAND快閃記憶體慢。**

1. **請設計一8051C語言程式引用下列函式，將AT24C64最高位址與最低位址內容互換。**

**void seeprom\_write\_byte(unsigned char seeprom\_device\_address, unsigned int seeprom\_memory\_address, unsigned char seeprom\_byte\_data);**

**unsigned char seeprom\_read\_byte(unsigned char seeprom\_device\_address, unsigned int seeprom\_memory\_address);**



**程式碼:**

**#define SEEPROM\_SCL P1\_0**

**#define SEEPROM\_SDA P1\_1**

**void main(){**

**char tmp;**

**seeprom\_write\_byte(0x00,0,'B');**

**seeprom\_write\_byte(0x00,8191,'A');**

**tmp=(char)seeprom\_read\_byte(0x00,0);**

**seeprom\_write\_byte(0x00,0,(char)seeprom\_read\_byte(0x00,8191));**

**seeprom\_write\_byte(0x00,8191,tmp);**

**}**

1. **請說明下列SDCC敘述功能:**
   1. **#include<8051.h>**

**引入8051.h檔案函示庫的標頭檔。**

* 1. **void func(void)\_\_interrupt 0;**

**宣告外部中斷0的中斷服務常式。**

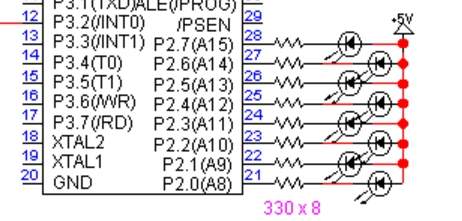
* 1. **\_\_xdata\_\_at 0xD000 unsigned char x[8192];**

**宣告無號字元陣列x[8192]，指定從外部RAM的D000H單元開始。**

* 1. **sdcc --code-loc 0x4000 --xram-loc 0x8000 test.c**

**使用sdcc編譯並設定ROM的起始位置為0x4000及設定外部記憶體起始位置從0x8000開始**

1. **請參考以下電路，設計一8051C語言程式，以肉眼可看見速度閃爍LED。**



**#include <at89x51.h>**

**void delay\_ms(unsigned int i);**

**void main(void){**

**unsigned int i = 0;**

**P2 = ~(0xFF);**

**while(1){**

**P2 = ~(P2);**

**delay\_ms(1000);**

**}**

**}**

**void delay\_ms(unsigned int n){**

**unsigned char i;**

**n\*=2;**

**while(n--)for(i=0;i<200;i++);**

**}**

1. **請說明8051執行MOVX A,@DPTR時，如何依序送出哪些控制信號來存取外部資料記憶體。**

**將DPTR+A的LOW BYTE擺到P0 ，High Byte 擺到P2，ALE>>將Low Byte 鎖住，PSEN>>將指定位址的資料送至DATA Bus上，A<P0，Data Bus 上的資料讀進來到A裡面。**

1. **請解釋下列名辭、積體電路、工具軟體:**
   1. **DTMF**

**雙音多頻信號，電話系統中**[**電話機**](https://zh.wikipedia.org/wiki/%E7%94%B5%E8%AF%9D%E6%9C%BA)**與**[**交換機**](https://zh.wikipedia.org/wiki/%E7%94%B5%E4%BF%A1%E4%BA%A4%E6%8D%A2)**之間的一種**[**用戶信令**](https://zh.wikipedia.org/w/index.php?title=%E7%94%A8%E6%88%B7%E4%BF%A1%E4%BB%A4&action=edit&redlink=1)**，通常用於發送被叫號碼。**

* 1. **Stack**

**堆疊，在8051中利用堆疊指標暫存器(SP)來標記目前儲存的位置，按照後進先出（LIFO, Last In First Out）的原理運作，組合語言中利用PUSH和POP的指令來達成。**

* 1. **ICL232**

**達靈頓電路，作為電流放大。**

* 1. **ULN2803A**

**8051程式模擬器，可以詳細觀察暫存器的變化。**

* 1. **Emulator 8051 V1.0**

**免費的8051軟體開發環境，提供編譯、組譯等功能。**

* 1. **MIDE-51 Studio**

**免費的8051軟體開發環境，提供編譯、組譯等功能。**

* 1. **ASEM-51 v1.3：**

**免費的8051組譯程式。**

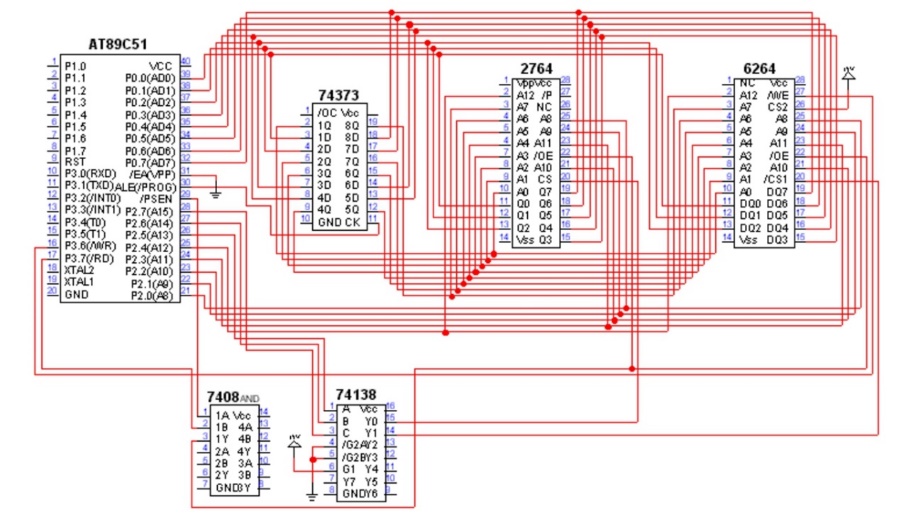
* 1. **Terminal Emulator(Tera Term Pro)：**

**終端機 可讀取和傳輸串列傳輸的資料**

* 1. **74C922**

**為 16 鍵編碼IC,固有(X1~X4)和(Y1~Y4)兩組輸入腳,共可以組成16個開關接點,故可有 16 組編碼。**

1. **請參考下列電路:**



* 1. **請說明 74373、74138之用途**

**74373為擁有8組三態輸出D型正反器，透過共同時脈腳(CLK)來達成栓鎖功能，另外為控制三態輸出的腳位。**

**74138為3對8線解碼/多工器，為高準位輸入、低準位輸出，其中G1為致能腳。**

* 1. **請說明2764、6264之定址範圍**

**2764定址範圍為0000H~1FFFH**

**6264定址範圍為2000H~3FFFH**

* 1. **請撰寫一8051 C語言程式將6264資料內容全部設為0**

**\_\_xdata\_\_at 0x2000 unsigned char x[8192];**

**void main(){**

**for(i=0;i<8192;i++)**

**x[i]=0;**

**}**

* 1. **請撰寫一8051 C語言程式將2764內容寫到6264：**

**\_\_code unsinged char x[8192];**

**\_\_xdata unsinged char y[8192];**

**Void main(){**

**for(i=0;i<8192;i++)**

**y[i]=x[i];**

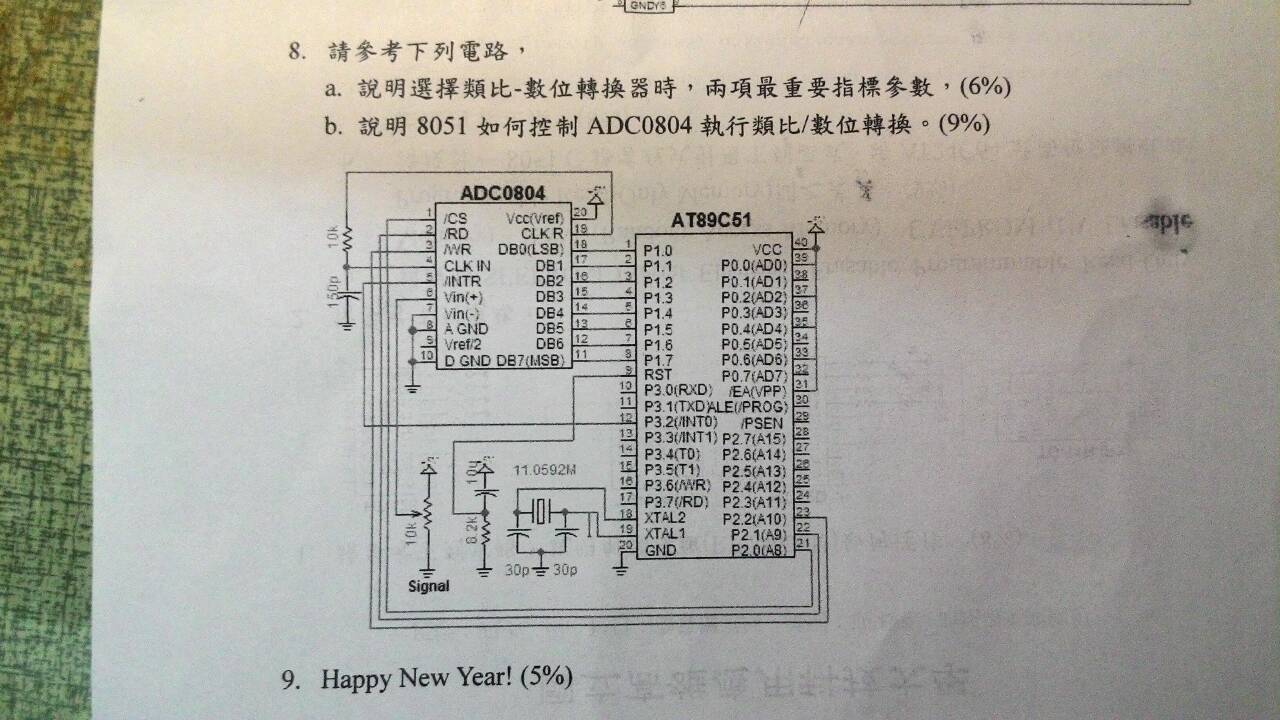
**}**

1. **請參考下電路**
   1. **說明選擇類比-數位轉換器時，兩項最重要指標參數**

**解析度、轉換時間。**

* 1. **說明8051如何控制ADC0804執行類比/數位轉換。**

**ADC0804數位訊號輸出端DB0~DB7連接8051的P0埠，讀取轉換為8位元的數位資料。 控制線接地(＝0)，表示晶片永遠處於被選擇狀態，與兩接腳連接8051的P3埠的兩個接點(P2.2，P2.3)，INTR接8051的中斷接腳INT0。**

****

1. **請參考下列UART與CLCD函式庫，設計一8051程式從終端機輸入二數，將兩數相乘之機於CLCD**

**//UART function prototypes**

**void uart\_initialize(void);**

**void uart\_put\_char(char c);char\_get\_char(void);char uart\_get\_char\_echo(void);**

**void uart\_put\_string(char \*s);void uart\_get\_string(char \*s);**

**void i2s(int I,char \*s);int s2i(char \*s);**

**void uart\_put\_integer(int i);int uart\_get\_ineger();**

**//Character LCD (HD44780U-compatible) function prototypes**

**void clcd\_initialize(void);**

**void clcd\_write\_instruction(unsigned char instruction);**

**void clcd\_wait(void); void clcd\_cls(void);**

**void clcd\_set\_cursor(unsigned int row,unsigned int column);**

**void clcd\_put\_char(char c); void clcd\_put\_string(char \*s);**

**void clcd\_put\_integer(int i);**

**程式碼:**

**void main(){**

**int i,j;**

**uart\_initialize();**

**uart\_put\_string("i: ");i=uart\_get\_integer();uart\_put\_char('\n');**

**uart\_put\_string("j: ");j=uart\_get\_integer();uart\_put\_char('\n');**

**clcd\_initialize();**

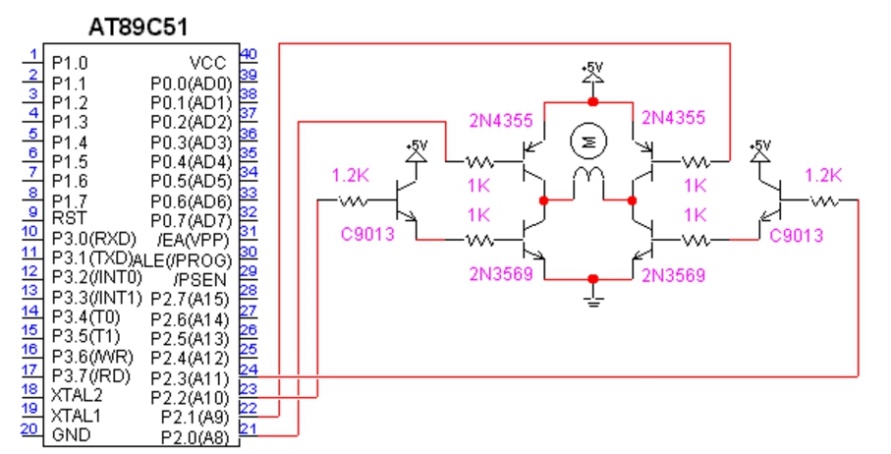
**clcd\_set\_cursor(1,0);**

**clcd\_put\_integer(i\*j);**

**}**

1. **參考以下電路，請說明如何操控直流馬達之旋轉方向及速度。**

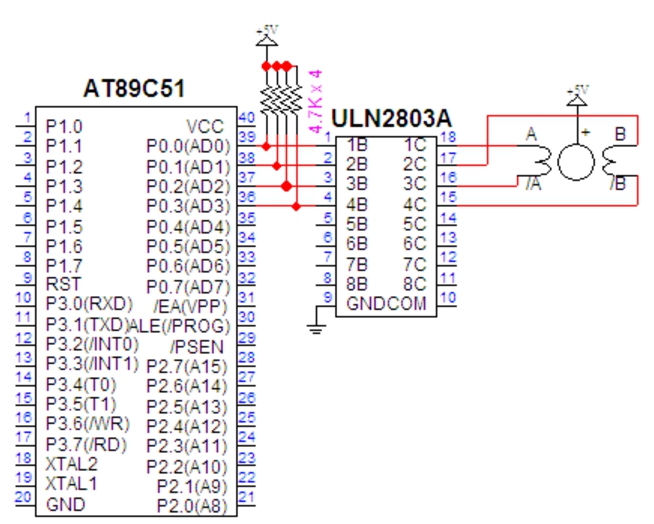
**P2\_0、P2\_1、P2\_2、P2\_3送出1100馬達停止轉動，分別先送出0101在延遲一段時間把P2\_3清除為0，再延遲一段時間，則會右轉。分別先送出1010在延遲一段時間把P2\_2清除為0，再延遲一段時間，則會左轉，其中兩次延遲時間的總和要固定，第一次的時間越長速度越快。**



1. **參考電路，說明步進馬達的運作原理，並請說明如何操作其旋轉方向及速度**

**步進馬達由2組線圈組成，六支腳分別為及兩個COM腳，其中COM腳需供應12V的電源，再透過這四隻腳的使得線圈產生電流讓馬達轉動。**

**控制方向按照順序送出1000->0100->0010->0001會順時針旋轉，反之送出0001->0010->0100->1000則逆時針旋轉，再根據下一輪動作的延遲來控制速度，周而復始。**



**744444444**