

C 語言學習網頁: www.tutorialspoint.com/cprogramming/

一、C 語言開發環境如下:

1. 文字編輯器(Text editor):----- gedit or geany
2. C 語言編輯器:(C compiler):-----sdcc(8051 C 編輯器)

二、語言之程式碼檔案副檔名通常為 .C (例如 io.c 或 control.c)

三、sdcc 的編譯指令:

sdcc -c 此 C 程式碼檔案 => 會產生一個 C 程式碼 .rel

sdcc C 程式碼檔案=>會產生 16 進制的機械碼檔案

狀況一: 只有一個程式碼 foo.c

將 foo.c 編譯為機械碼之指令為 sdcc foo.c 可產生 foo.ihx 檔

狀況二: 有兩個程式碼

副程式檔名為 fool.c

主程式檔名為 main.c

編譯成機械碼之指令需:

1. 將副程式編譯成 .rel 檔
2. 將主程式與副程式之 .rel 一起編成機械碼

步驟一 sdcc c fool.c =>產生 fool.rel

步驟二 sdcc main.c fool.rel=>會產生 main.ihx

四、C 語言架構---主要有五大指令

1. Preprocessor 前置處理 如 #include, #define
2. Function 函式 如 main(), printf()
3. Variable 變數
4. Statement & Expression 敘述/表示式
5. Comment 註解 //雙斜線為當行註解
/*.....*/為方塊註解, 裏頭的字為註解

五、主程式(主函式) main()

任何 C 語言必有一 main() 函式, 且程式一定會從 main() 開始運行

六、C 語言程序

```
#include 檔頭檔(header file)
#define 巨集/常數
全域變數宣告;
副程式宣告;
void <=無須回傳時
Int main(void)
↑有宣告回傳值時
{區域變數宣告 ;
敘述表示式;
迴圈;
.
.
.
Void ; ←宣告時，需刪掉此行
return 0 ; ←有宣告回傳時需如此行
}
變數宣告 ↓分號為行結束
資料型態 變數辨識字 ;
↑變數名
```

※辨識字允許 A~Z、a~z 0~9，例如:A123、Aa_Bb

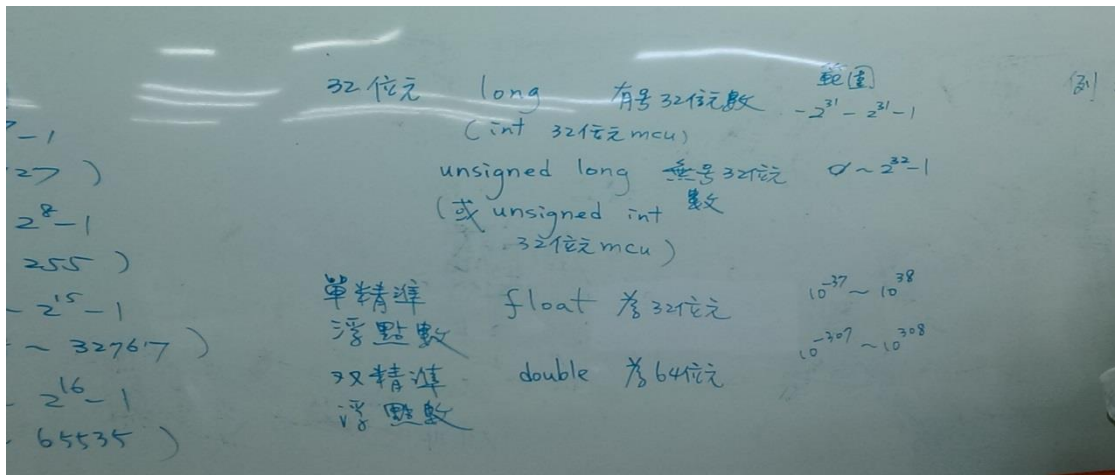
a123 與 A123 是不同的變數，b 與 aB 也為不同的變數

辨識字不可為關鍵字，常用的關鍵字如下

If、else、for、while、switch、case、break、comtinue、do、union、strnct、
return、typedef、sigeof、char、int、long……

資料型態(Data Type)

資料型態 (Data Type)	數值範圍	32位元
8位元 char 有8位元數	$-2^7 - 2^7 - 1$ (-128 ~ 127)	
unsigned char 無8位元數	$0 \sim 2^8 - 1$ (0 ~ 255)	
16位元 int 有16位元數 (或 short)	$-2^{15} \sim 2^{15} - 1$ (-32768 ~ 32767)	
unsigned int 無16位元數 (或 unsigned short)	$0 \sim 2^{16} - 1$ (0 ~ 65535)	



例:宣告一變數其範圍 0~1000，則可用之資料型態為:

int 或 unsigned int

可宣告

Int myVar ;

全域變數(global variable)

宣告在 int main()函式之外之變數即為全域變數，全域變數之有效範圍為整個程式

區域變數(local variable)

再函式內宣告之變數(包含 main()函式)均為區域變數，其有效範圍僅在其宣告之函式內有用，在此函式外即失效

※當全域變數名與區域變數名相同時，在宣告區域變數之函數內以區域變數有效

例:

char A=1; (全域)

int main() { ;

char A=2 ; (區域) ※在 main 內 A=2

char C, D;

C=A+2; C=2+2=4

D=add(); D=1+2+3

{

Int add() {

Return (A+2); ※A=1

}

extern (外部變數宣告)

當 A 程式碼回到 B 程式碼之全域變數時需加上 extern 來宣告此變數在其他程式已宣告

例:

```
extern int C;  ※在其他程式已有宣告
int add(int B){
    return(B+C);
}
```

A 程式碼

```
int C;
```

.

B 程式碼

例

在軟件模擬環境中
P1為輸出埠連接到
8個LED，可將P1
視為8位元無符號數
寫一程式將P1由0開
始加1直到255後再重
覆上述動作。

```
#include <8051.h>
void main(void)
{
    P1 = 0;
    while(1)
    {
        P1++;
    }
}
```

開啟 Term

```
cd micro
```

```
mkdir p1
```

```
cd p1
```

gedit pl.c &

寫程式

寫好後

Sdcc pl.c