

跑馬燈(marquee)

使用 LED 燈左右移動，或具有樣式移動

使用：陣列(array)

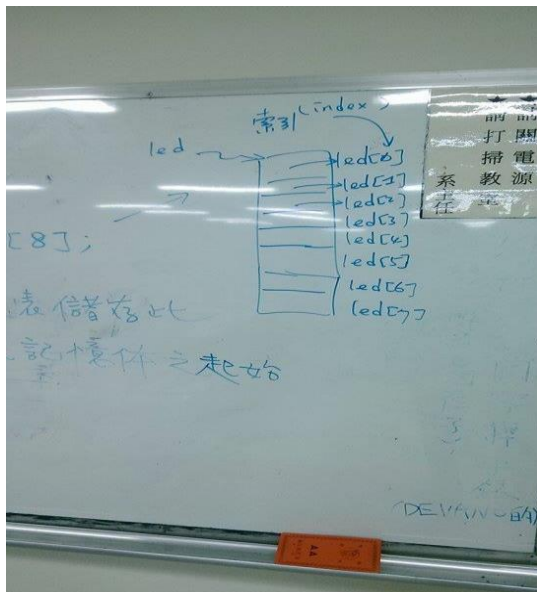
「一維陣列」

宣告方式：

資料型態 陣列變數名[陣列長度]

例：

Char led[8]; ←宣告 LED 的八個位置



陣列變數代表儲存此陣列資料之記憶體之起始位置

陣列變數名[索引值]，代表陣列記憶體中第”索引值”個記憶體之資料值，所有值從 0 開始，到最大陣列長度-1

LED 位置

LED[0]=1

LED[1]=2

LED[2]=3

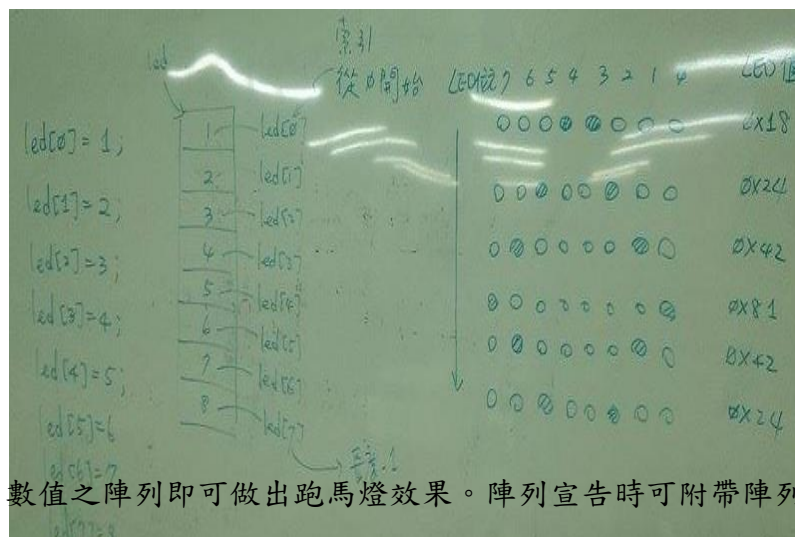
LED[3]=4

LED[4]=5

LED[5]=6

LED[6]=7

LED[7]=8



設定---LED 數值之陣列即可做出跑馬燈效果。陣列宣告時可附帶陣列之起始

值。

帶起始值之陣列宣告

資料型態 陣列名[ ] = {索引=0 之值 索引=1 之值  
或陣列名[長度] 索引=長度-1 之值}

則

```
Char ledVal[6]={0x18 0x24 0x42 0x81 0x42 0x24};
```

控制迴圈：

```
While (迴圈條件){
```

迴圈條件成立(條件值≠0)

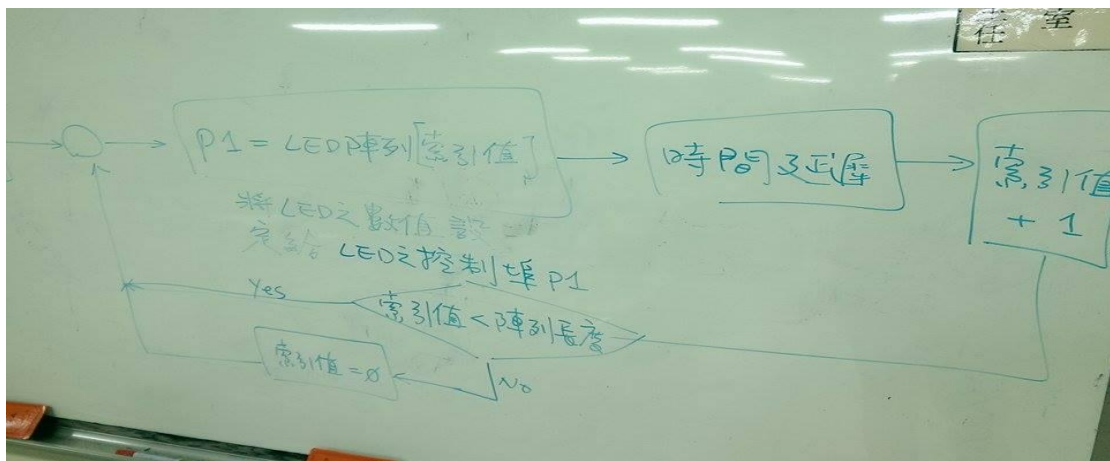
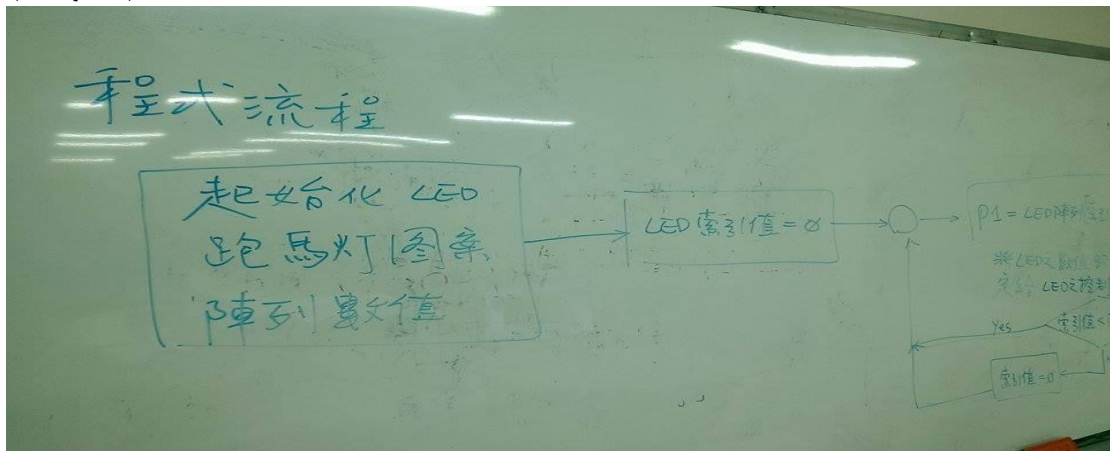
執行{...}內之程式;

If (離開條件) break;

當離開條件成立時，或迴圈條件不成立時離開迴圈

```
}
```

程式流程：



## Ubuntu 步驟流程

### 1. 編輯 ex2.c 之步驟

```
cd micro
mkdir ex2
cd ex2
gedit ex2.c &
```

### 2. ex2.c 內容

```
#include <8051.h>
void delay(unsigned char n){
    unsigned char i ;
    for(i=0; i<n; i++);
}
void main(void){
char ledVal[6]={0x18, 0x24, 0x42, 0x81, 0x42, 0x24};
char index =0;
while(1){
    P1=~ledVal[index];
    delay(1);
    index++;
    if(index==6) index=0;
}
}
```

### 3. 儲存後編譯

```
sdcc ex2.c
```

### 4. 模擬

```
cd ..
cd edsim51di
java -jar edsim51di.jar
在模擬視窗
Load ex2.ihx
```

If 敘述

```
If(條件 1){  
    符合條件 1 時執行處程式 A  
}  
else if(條件 2){  
    符合條件 2 時執行此處程式 B  
}  
else {  
    不符合條件 1 與條件 2 時執行此處程式  
}
```

例如：

```
A=3 → B=10  
if(A==1){  
    B=5;  
}  
else if(A==2){  
    B=7;  
}  
else B=10;
```