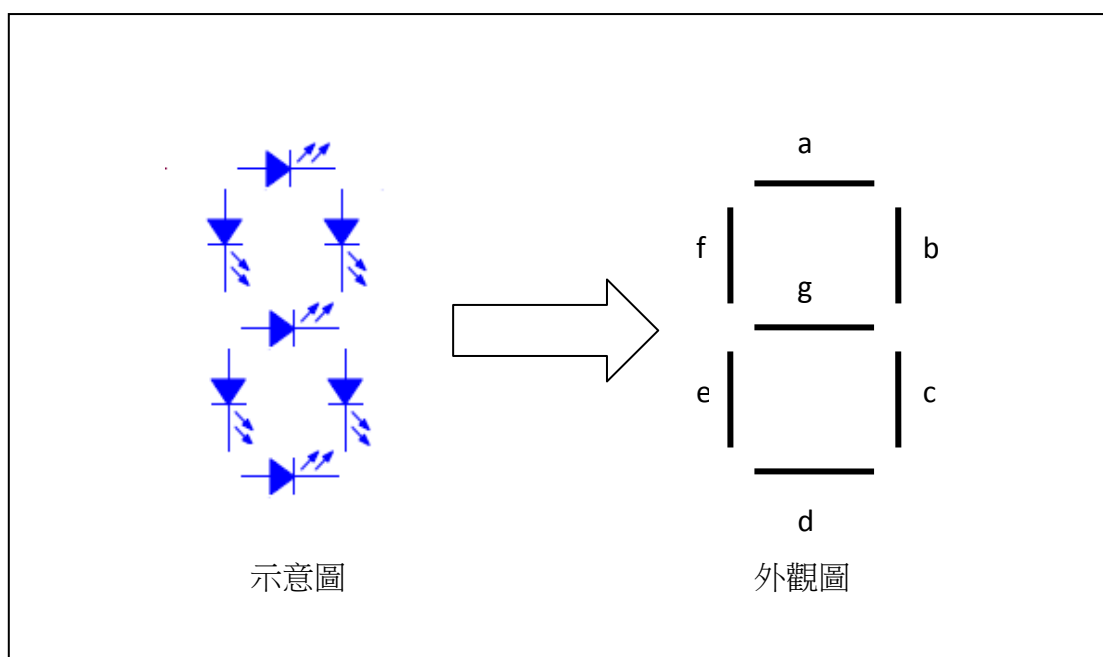


uClinux 作業系統下利用應用程式

驅動七段顯示器(uClinux-4)

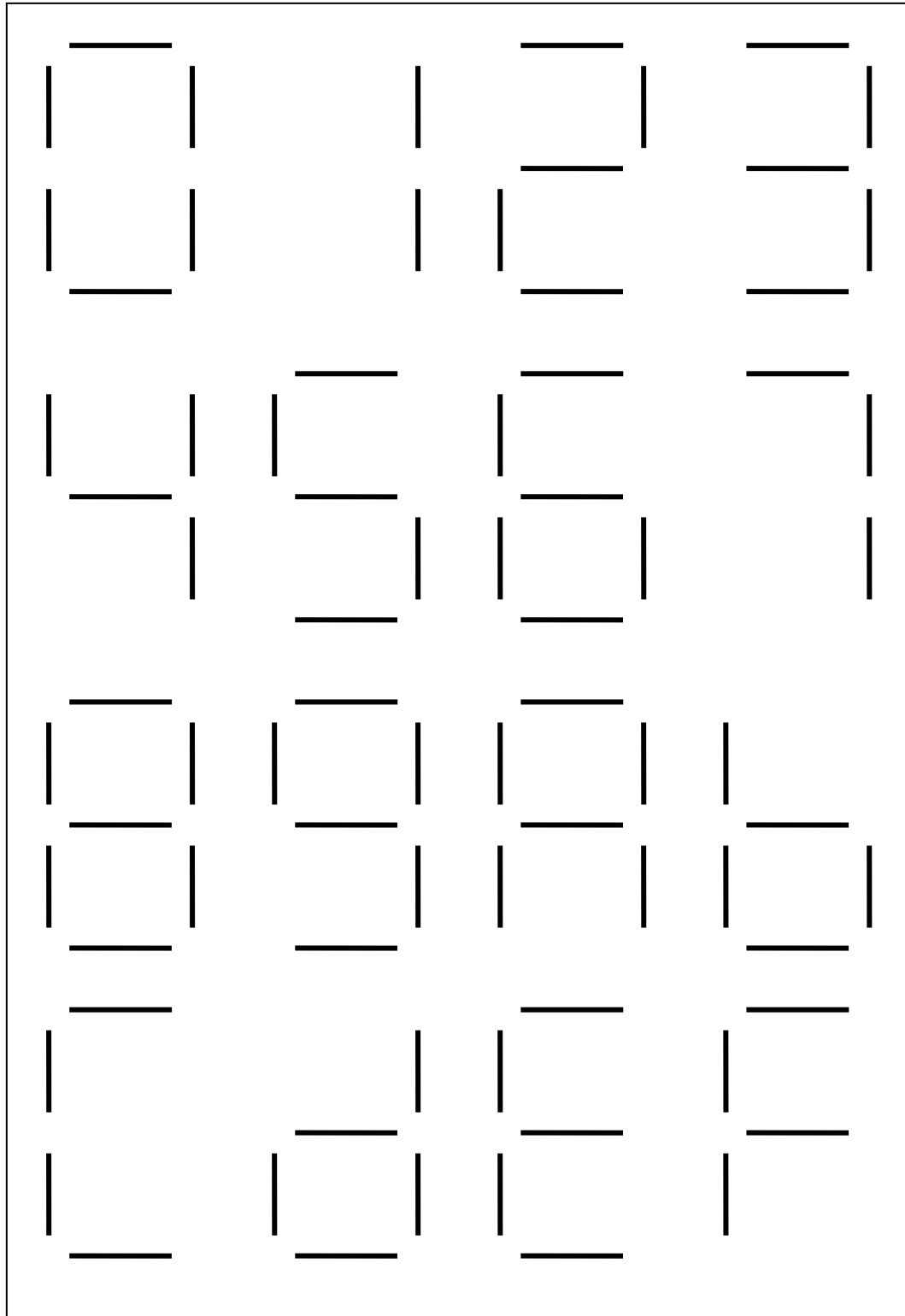
七段顯示器是發光二極體的組合，由於一組發光二極體所能表示的訊息，只有亮與滅，如果要表示數字或文字就沒辦法了，七段顯示器就是為了要表現文數字而設計的，它的原理很簡單，就是多組發光二極體的組合，如下圖所示：



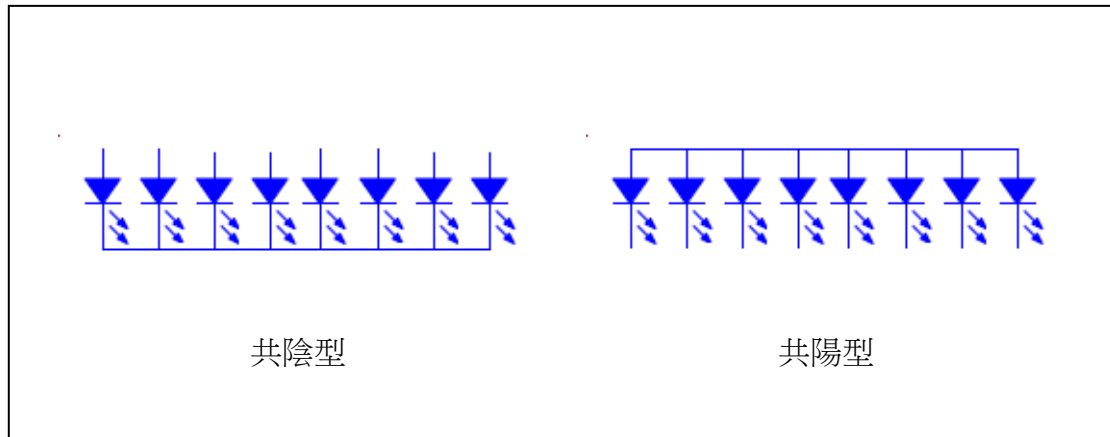
上圖左邊為七段顯示器之示意圖，由於所要呈現的為文數字，所以每一發光二極體所顯示的如上圖右側，為一直線，透過不同發光二極體的組合，即可顯示不同的文數字，如下表所示：

顯示文數 字	發光二極體顯示與否(1 表顯示，0 表不顯示)						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1
A	1	1	1	0	1	1	1
B	0	0	1	1	1	1	1
C	1	0	0	1	1	1	0
D	0	1	1	1	1	0	1
E	1	0	0	1	1	1	1
F	1	0	0	0	1	1	1

依照上表，顯示之文數字如下圖：

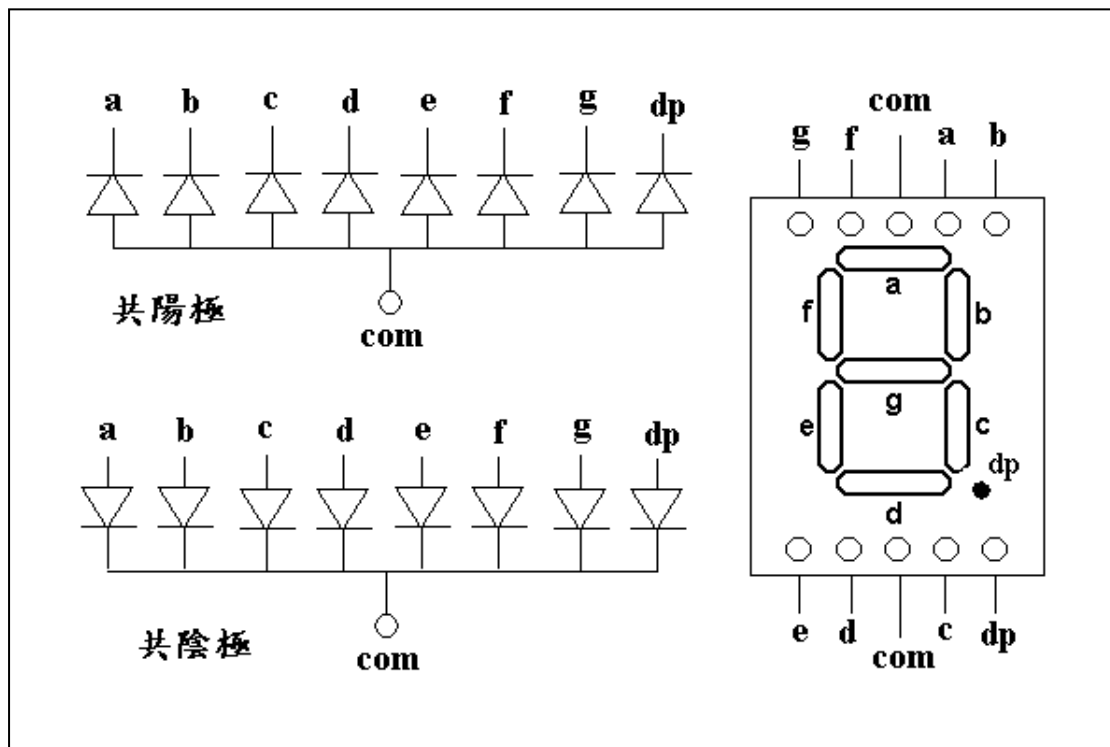


由上述七段顯示器之示意圖來看，七段顯示器必然有七段控制接腳，另一接腳為共用腳，如下圖所示：



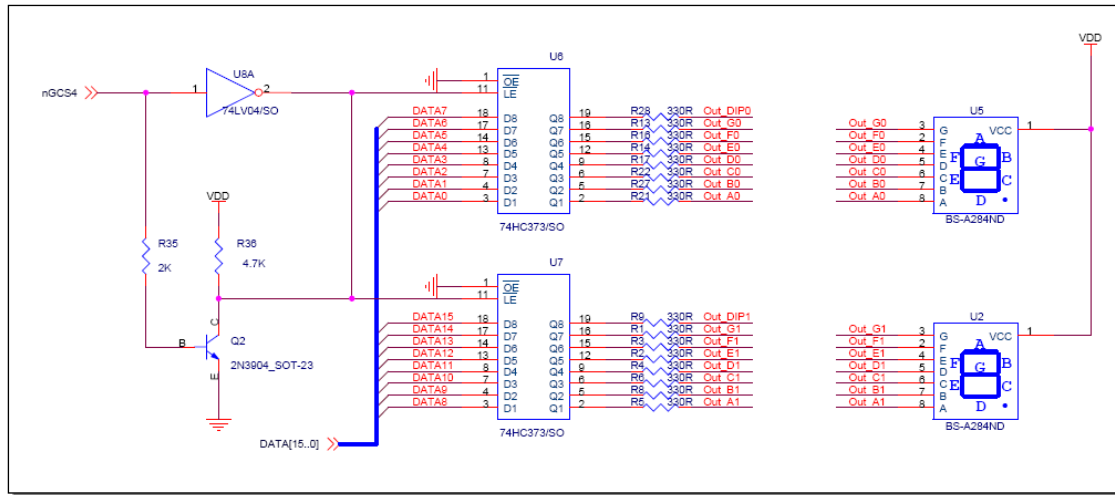
如上圖所示，共同接腳接至地稱之為共陰型，而共同接腳接至電源稱之為共陽型，而共陰型的控制方式為控制電源輸出至 a-g 發光二極體，至於共陽型的控制方法為控制地之輸出至 a-g 發光二極體。

七段顯示器為利用上述所說之原理，製作而成之元件，如下圖所示：



設計電路的原理，就是將七段顯示器之 com 接腳，依照共陽或共陰型，分別接至電源或是地，再利用控制接腳 a-g 控制所欲顯示之字型。

目標板 ESD44B0_B 所選用之七段顯示器，為共陽型，如下圖所示：



ESD44B0_B 上共有兩組七段顯示器，分別接至一組 74373 門門晶片，每一晶片控制 8 組輸出，由於七段顯示器除了七段外，尚有一小數點，目標板並不控制此接腳，所以此兩組門門晶片等於分別控制一組七段顯示器之 a-g 接腳，當相對應之接腳為低電位時，七段顯示器之對應發光二極體會被點亮，門門晶片之輸出訊號為 Out_An-Out_Gn(其中 n 為 0 或 1)，分別代表不同門門晶片之輸出，以控制七段顯示器，控制真值表如下：

顯示文數 字	門門晶片 74373 輸出訊號(H 表高電位，L 表低電位)						
	Out_An	Out_Bn	Out_Cn	Out_Dn	Out_En	Out_Fn	Out_Gn
0	L	L	L	L	L	L	H
1	H	L	L	H	H	H	H
2	L	L	H	L	L	H	L
3	L	L	L	L	H	H	L
4	H	L	L	H	H	L	L
5	L	H	L	L	H	L	L
6	L	H	L	L	L	L	L
7	L	L	L	H	H	H	H
8	L	L	L	L	L	L	L
9	L	L	L	L	H	L	L
A	L	L	L	H	L	L	L
B	H	H	L	L	L	L	L
C	L	H	H	L	L	L	H
D	H	L	L	L	L	H	L
E	L	H	H	L	L	L	L
F	L	H	H	H	L	L	L

ESD44B0_B 利用 S3C44B0X 之記憶體控制器的第四區塊記憶體，來控制門門晶片，S3C44B0X 片之第四區塊記憶體的啟動/停止控制腳為 nGCS4，上圖中利用 nGCS4 控制門門晶片之鎖定(LE)接腳，資料匯流排之 DATA[7:0]及 DATA[15:8]接腳，分別接至兩組門門晶片。

處理器將資料寫入第四區塊記憶體區間(0x800_0000 - 0xA00_0000)時，nGCS4 接腳處於低電位，經過反向接至門門晶片(74373)之鎖定接腳(LE)，兩組門門晶片會同時，將 DATA[7:0]及 DATA[15:8]資料匯流排之狀態鎖定，並輸出至 Out_A0 - Out_G0 及 Out_A1 - Out_G1 接腳，如此便可驅動七段顯示器，其真值表如下表：

nGCS4	LE	DATA[15:0]	Out_An - Out_Gn
H	L	無用	未鎖定
L	H	鎖定	輸出

資料匯流排被鎖定後，送至 Out_An - Out_Gn，其對應關係如下表：

資料匯流排腳位	鎖定接腳
DATA7	無用
DATA6	Out_G0
DATA5	Out_F0
DATA4	Out_E0
DATA3	Out_D0
DATA2	Out_C0
DATA1	Out_B0
DATA0	Out_A0
DATA15	無用
DATA14	Out_G1
DATA13	Out_F1
DATA12	Out_E1
DATA11	Out_D1
DATA10	Out_C1
DATA9	Out_B1
DATA8	Out_A1

資料匯流排 DATA[7:0]及 DATA[15:8]分別控制兩組門門電路，也就是說，結果分別顯示於兩組七段顯示器，顯示之字型與匯流排間之關係為反向，下表為字型在匯流排之代碼及顯示字型：

顯示字型	顯示代碼	資料匯流排
0	0xC0	DATA[7:0] / DAT[15:0]
1	0xF9	DATA[7:0] / DAT[15:0]

2	0xA4	DATA[7:0] / DAT[15:0]
3	0xB0	DATA[7:0] / DAT[15:0]
4	0x99	DATA[7:0] / DAT[15:0]
5	0x92	DATA[7:0] / DAT[15:0]
6	0x82	DATA[7:0] / DAT[15:0]
7	0xD8	DATA[7:0] / DAT[15:0]
8	0x80	DATA[7:0] / DAT[15:0]
9	0x98	DATA[7:0] / DAT[15:0]
A	0x08	DATA[7:0] / DAT[15:0]
B	0x03	DATA[7:0] / DAT[15:0]
C	0x46	DATA[7:0] / DAT[15:0]
D	0x21	DATA[7:0] / DAT[15:0]
E	0x06	DATA[7:0] / DAT[15:0]
F	0x0E	DATA[7:0] / DAT[15:0]

舉例來說，如果資料匯流排之資料為 **0xC0C0** 代表兩組七段顯示器顯示'00'，如果為 **0xC0F9** 代表顯示'01'之字型，以下類推。

瞭解原理以後，在 **uClinux** 作業系統下纂寫驅動七段顯示器的程式就變得相當的簡單，就是在 **0x800_0000** 記憶體位置寫入相對應的資料，不再列出程式原始碼，有興趣的讀者請下載附件。

Victor 於加拿大