

Ch1 可程式控制器簡介

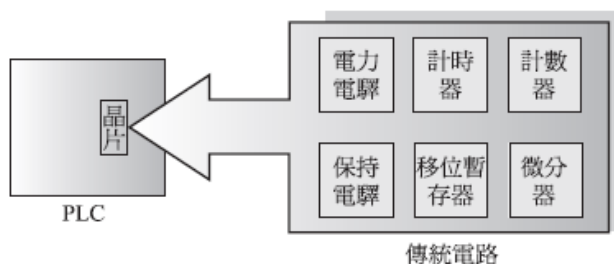
班級：電機二 姓名：_____

一、什麼是可程式控制器(Programmable Logic Controller)

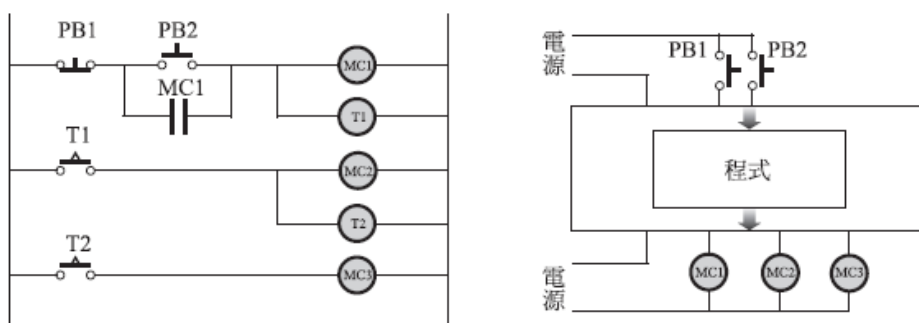
1. 就英文字義來解釋，可程式控制器(簡稱_____)是一個可程式的，可規劃的邏輯控制器。
2. 可程式控制器是一台精簡化的微電腦，不同功能有不同的程式內容，由使用者決定
3. 傳統工業配線電路是將所有輸出入依功能作整體的連線。而可程式控制器是以本體為核心，將線路分成_____與_____兩部份，輸入接點共_____個，輸出接點共_____個

二、可程式控制器的特點

1. 體積小，功能強



2. 硬體電路配線容易，維修簡單

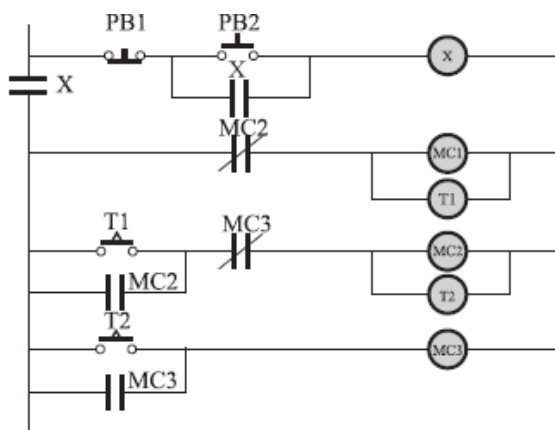


3. 量產時，可省下更多時間與經費
4. 可視需要更改程式，不像傳統工業配線，必須重新佈置及組裝配件

EX：若需將上圖功能更改

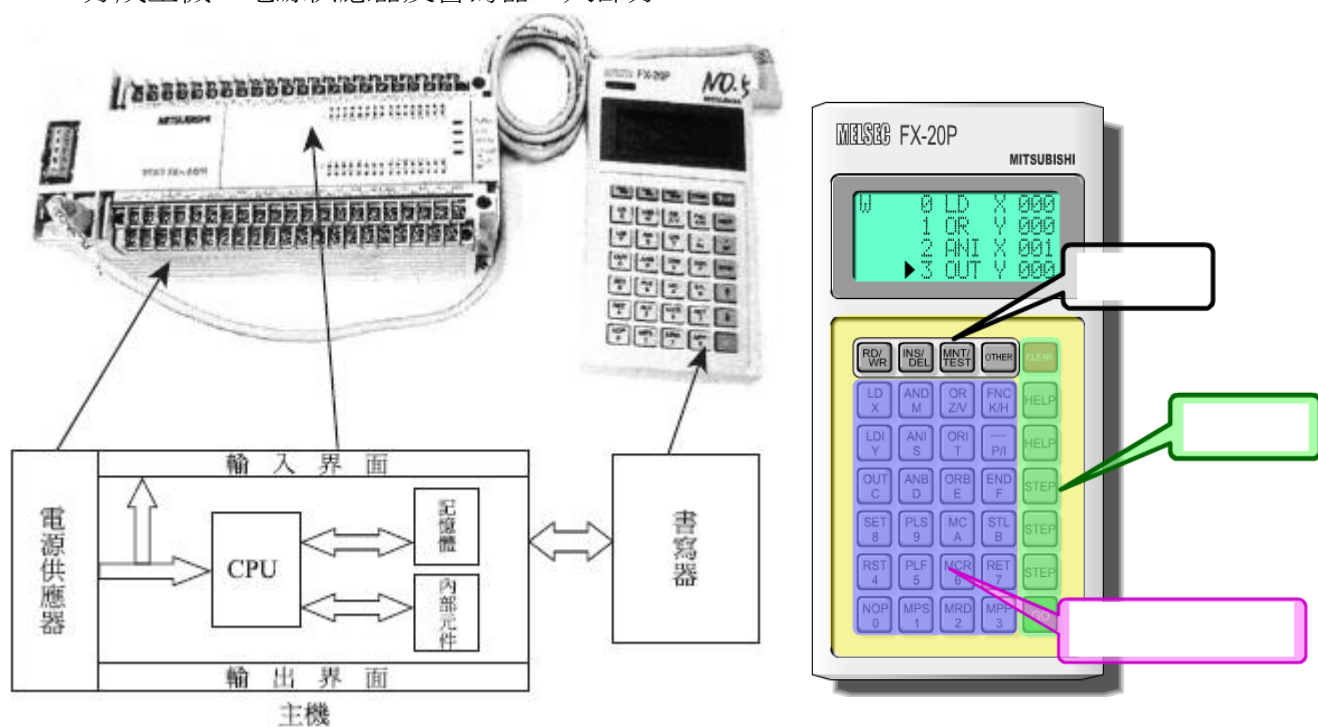
PB2 按下 → MC1 動作 $\xrightarrow{T1}$ MC2 動作，MC1 跳脫 $\xrightarrow{T2}$ MC3 動作，MC2 跳脫

PB1 按下 → MC1, MC2, MC3 皆跳脫



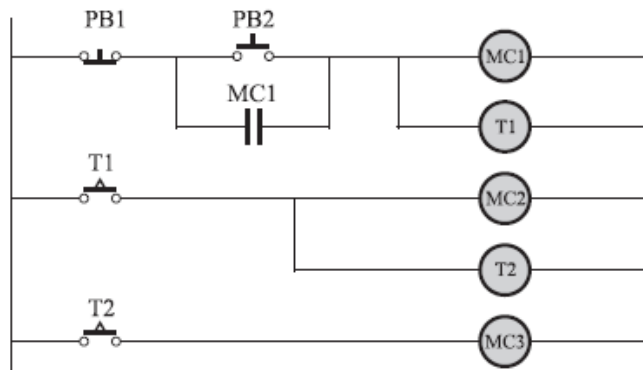
三、硬體結構

1. 分成主機、電源供應器及書寫器三大部分

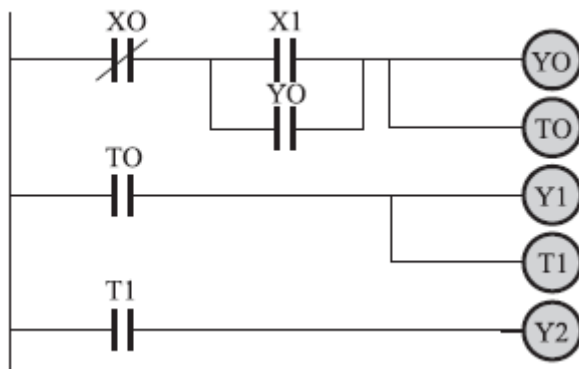


四、階梯圖繪製

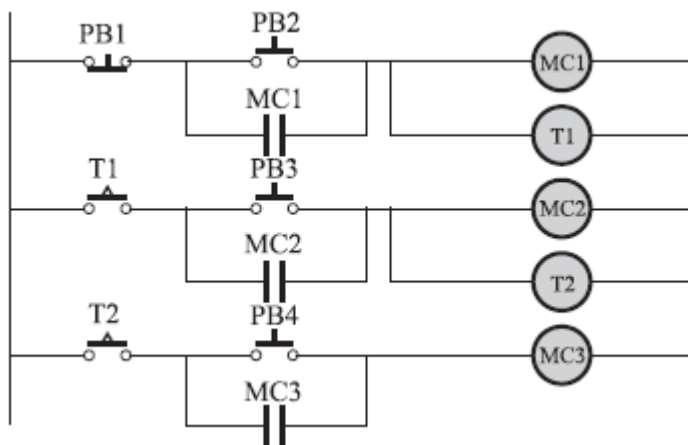
1. 將傳統工業配線圖，改成可程式控制器階梯圖
 - (1) 先分辨輸出入元件、內部元件後分別編號
 - (2) A 接點以___表示，B 接點以___表示
 - (3) 輸出元件都是以一個圓圈填入元件編號即可



輸入元件	輸出元件	內部元件
PB1→X0	MC1→Y0	T1→T0
PB2→X1	MC2→Y1	T2→T1
	MC3→Y2	



2. 請將傳統工業配線圖，改成階梯圖繪出



輸入元件	輸出元件	內部元件

實習心得：

日期：_____

Ch2 可程式控制器基本操作

一、基本操作

1. 程式全部清除：

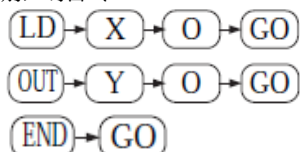
(1) 選擇寫入模式

(2) 鍵入指令 **NOP** **A** **GO**

2. 程式鍵入：

(1) 選擇寫入模式

(2) 輸入指令

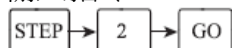


位址	指 令
0	LD X0
1	OUT Y0
2	END

3. 直接跳到某一位址：

(1) 選擇讀取模式

(2) 輸入指令



**利用此指令找找看書寫器行數：_____行

4. 程式修改

(1) 選擇寫入模式

(2) 直接輸入新指令

5. 尋找某一指令

(1) 選擇讀取模式

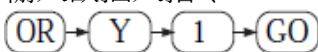
(2) 輸入欲尋找之指令



6. 插入一行指令

(1) 選擇插入模式

(2) 輸入欲插入指令



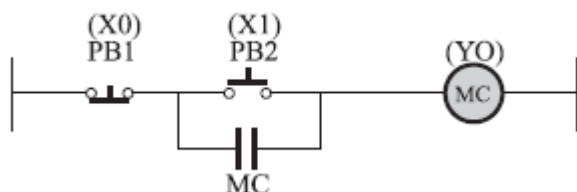
7. 刪除某一指令

(1) 尋找指令

(2) 選擇刪除模式

(3) 按執行鍵

範例：請將以下自保持電路以可程式控制器控制



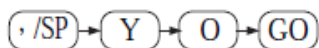
位址	指 令
0	LD X1
1	OR Y0
2	ANI X0
3	OUT Y0
4	END

8. A.程式監視

- (1) 模式開關搬到 RUN 模式
- (2) 選擇監視模式

B.元件監視

- (1) 模式開關搬到 RUN 模式
- (2) 輸入欲監視元件



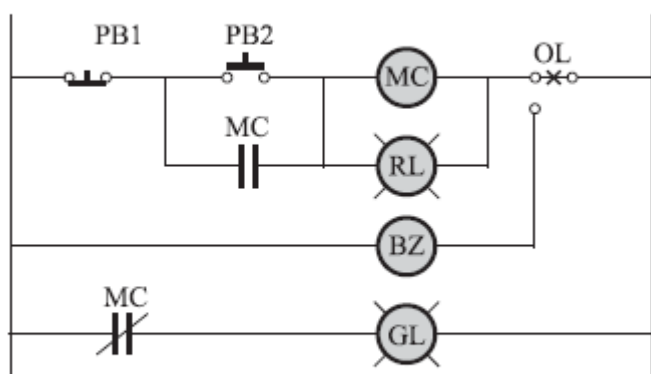
9. 強制驅動及復歸

- (1) 選擇測試模式
- (2) 強制輸出 **SET**
- (3) 強制復歸 **RST**

補充：檢查所有元件



實習：



位址	指 令
0	LD X1
1	OR Y0
2	ANI X0
3	ANI X2
4	OUT Y0
5	OUT Y1
6	LD X2
7	OUT Y2
8	LDI Y0
9	OUT Y3
10	END

1. 執行：將模式開關搬到 RUN 位置。

- (1) 送電後，綠燈 GL(Y3)亮
- (2) 按下 PB2(X1)開關，MC(Y0)動作，同時紅燈 RL(Y1)亮，綠燈 GL(Y3)熄，馬達運轉
- (3) 按下 PB1(X0)開關，MC(Y0)跳脫，同時紅燈 RL(Y1)熄，綠燈 GL(Y3)亮，馬達停止運轉
- (4) 過載時(拉起 OL—X2)，蜂鳴器 BZ(Y2)響、其他動作同(3)

Ch3 可程式控制器基本指令與內部元件

一、基本指令介紹

指令名稱	功能	指定對象	指令名稱	功能	指定對象
LD	母線開始 a 接點	X、Y、M、S、T、C	ORB	區塊並聯	
LDI	母線開始 b 接點	X、Y、M、S、T、C	ANB	區塊串聯	
OUT	輸出線圈	Y、M、S、T、C	PLS/PLF	上緣/下緣微分輸出	Y、M
OR	並聯連接的 a 接點	X、C	MPS	分歧點的開始	
ORI	並聯連接的 b 接點	X、Y、M、S、T、C	MRD	分歧點的中間	
AND	串聯連接的 a 接點	X、Y、M、S、T、C	MPP	分歧點的結束	
ANI	串聯連接的 b 接點	X、Y、M、S、T、C	NOP	不執行運算	
SET	強制驅動	Y、M、S	END	程式結束回到位址 0	
RST	強制復歸	Y、M、S、T、C、D、 V、Z			

二、內部元件及指令

1. 一般電驛

電驛編號	用途	注意事項
M0-M383	一般用途	可使用參數變更停電保持用的範圍
M384-M1153	停電保持	

2. 限時電驛

電驛編號	用途	數值的範圍	單位時間	預設值範圍
T0-T199	一般用途	1-32767	0.1 秒	0.1-3276.7
T200-T245			0.01 秒	0.01-327.67
T246-T249	停電保持		0.001 秒	0.001-32.767
T250-T255			0.1 秒	0.1-3276.7

3. 計數器

電驛編號	種類	用途	預設值範圍
C0-C15	加算型	一般用途	16 位元
C16-C199		停電保持	1~32767
C200-C219	加算/減算型	一般用途	32 位元
C220-C234		停電保持	-21464836~2147483647
C235-C255	高速型 加算/減算型	停電保持	32 位元 -21464836~2147483647

4.特殊電驛

電驛編號	用途
M8000	常時 ON 電驛
M8001	常時 OFF 電驛
M8002	第一次掃瞄動作電驛
M8011	0.01 秒時鐘脈波電驛
M8012	0.1 秒時鐘脈波電驛
M8013	1 秒時鐘脈波電驛
M8014	1 分鐘時鐘脈波電驛

5.其他元件

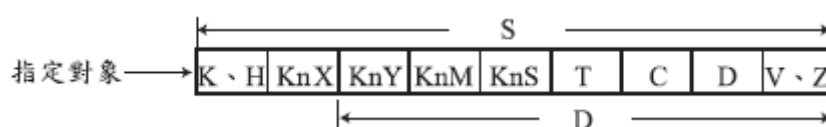
元件編號	種類	用途
D0~D127	資料暫存器	一般用途
D128~D1099		停電保持
D1000-D7999		檔案用途
D8000-D8511		特殊用途
V, Z	索引暫存器	間接定址用
P	跳躍指標	跳躍指引
I	中斷指標	中斷指引

Ch4 可程式控制器應用指令與設計

常用應用指令

指令名稱	編號	功能	指令	備註
MOV	FNC 12	將來元元件狀態或數值傳送至目標元件	MOV S D	
CML	FNC 14	將來源狀態或數值反向後再傳送到目標元件	CML S D	
INC	FNC 24	將目標元件值加 1 後再存回目標元計	INC S D	通常使用 INCP
DEC	FNC 25	將目標元件值減 1 後再存回目標元計	INC D	通常使用 DECP
ADD	FNC 20	將被加數與加數相加，結果存入目標元件	ADD 被加數 加數 D	
SUB	FNC 21	將被減數與減數相減，結果存入目標元件	SUB 被減數 減數 D	
CMP	FNC 10	將 S1 S2 相互比較，並指定目標元件來反應	CMP S1 S2 D	
ZRST	FNC 40	將 D1 至 D2 間所有目標元件予以復歸	ZRST D1 D2	
SFTL/SFTR	FNC 35/ FNC 36	將目標元件內的值左/右移 N2 個單位，空位元由補位元件填入	SFTL S D N1 N2 SFTR S D N1 N2	N1：移位目標元件總數
ROL/ROR	FNC 31/ FNC30	將 D 中的值向左/右旋轉 N 個位元	ROL D N ROR D N	N：旋轉位元數
ALT	FNC 66	交替驅動與復歸目標元件	ALT D	
WAND	FNC 26	將兩個來源元件或數值進行邏輯 AND 運算	WAND S1 S2 D	
WOR	FNC 27	將兩個來源元件或數值進行邏輯 OR 運算	WAOR S1 S2 D	
WXOR	FNC 28	將兩個來源元件或數值進行邏輯 XOR 運算	WXOR S1 S2 D	

S：來源元件或數值 D：目標元件或數值



狀態設計法—條件控制

設計一電路控制，條件如下：按 PB1 則 M1、M2 動作；按 PB2 則 M2、M3 動作；按 PB3 則 M3、M1 動作。

1.狀態分析並直接繪出階梯圖：

(1)將電路功能分析成若干狀態，每一個狀態都以一個一般電驛來代表。上述功能可以分成三種狀態，亦即

A.按 PB1 時為第一種狀態 S1，並以一般電驛 M0 來代表

B.按 PB2 時為第二種狀態 S2，並以一般電驛 M1 來代表

C.按 PB3 時為第三種狀態 S3，並以一般電驛 M2 來代表

輸入元件	輸出元件	內部元件
PB1→X0	MC1→Y0	S1→M0
PB2→X1	MC2→Y1	S2→M1
PB3→X2	MC3→Y2	S3→M2
OL1→X3		
OL2→X4		
OL3→X5		

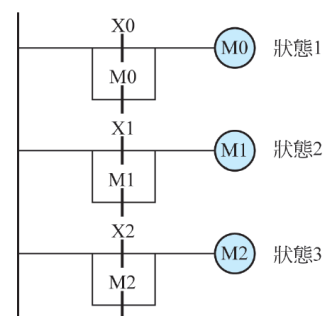


圖 3-67

(2)因為馬達的動作均為持續運轉，所以每一種狀態都應該能自保持，因此階梯圖如上圖

(3)因為三種狀態中，在一段時間只能有一種存在，所以假若 S1 動作，則 S2、S3 必須切斷；同理 S2 動作，則 S1、S3 必須切斷；S3 動作，則 S1、S2 必須切斷。這是一種互鎖電路，最好的方法是利用開關的 b 接點。也就是用 a 接點接通自己而用 b 接點切斷別人，如圖 3-68。

(4)加上 OL 得到下圖（圖 3-69）

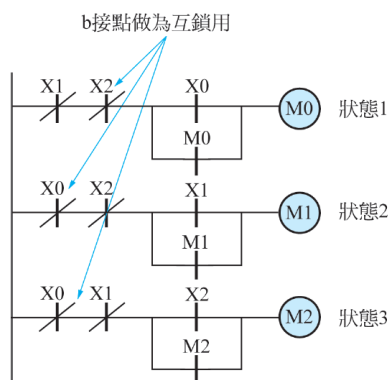


圖 3-68

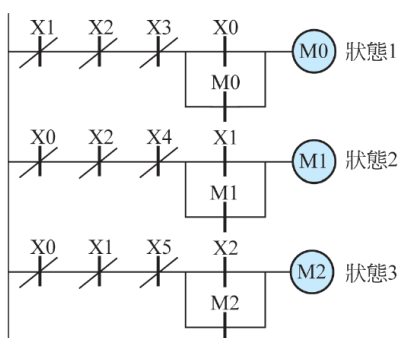


圖 3-69

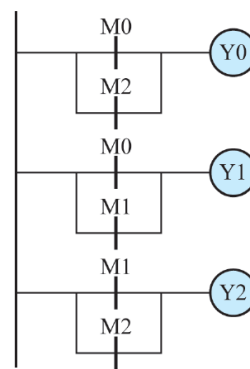


圖 3-70

(5)利用狀態電驛的 a 接點來驅動輸出：狀態 S1 (M0) 時，M1、M2 (Y0、Y1) 動作；狀態 S2 (M1) 時，M2、M3 (Y1、Y2) 動作；狀態 S3 (M2) 時，M3、M1 (Y2、Y0) 動作。交叉分析得到如圖 3-70。

(6)將上兩圖組合以後得到完整的階梯圖

狀態設計法—**程序控制**

設計一電路控制，條件如後：按 PB2 則 PL1、PL2 動作；經 1 秒後改成 PL2、PL3 動作；再經 1 秒後改成 PL3、PL1 動作；再經 1 秒後又改成 PL1、PL2 動作。如此重覆執行，直到按下 PB1 之後全部復歸。

(1) 狀態分析同上例，是為三個依設定時間而順序執行的狀態。

輸入元件	輸出元件	內部元件
PB1→X0	PL1→Y0	S1→M0 T0
PB2→X1	PL2→Y1	S2→M1 T1
	PL3→Y2	S3→M2 T2

(2) 因此階梯圖如圖 3-73

(3) 因為三種狀態中，在一段時間內只能有一種存在，而且狀態是依序進行的，所以假若 S1 動作，則 S3 必須切斷；同理 S2 動作，則 S1 必須切斷；S3 動作，則 S2 必須切斷。另外為了避免在狀態執行中又按下 X1 開關，因此在 S1 迴路上必須串上其它狀態的 b 接點，如圖 3-74。

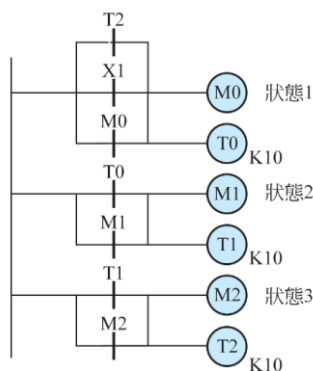


圖 3-73

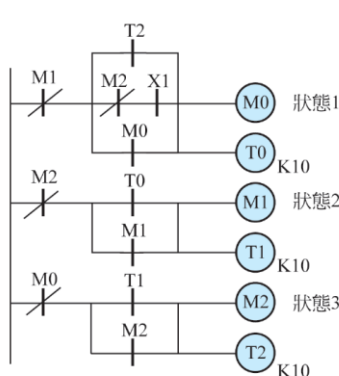


圖 3-74

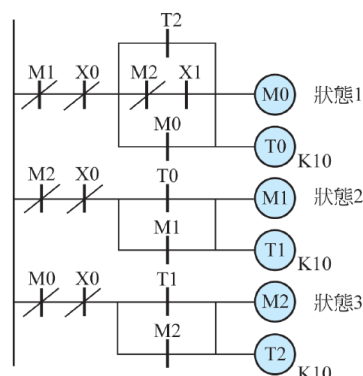


圖 3-75

(4) 因為在任何狀態下，按下 PB1 時，都應能切斷電路，因此必須在每一狀態行，都插入 PB1 的 b 接點，如圖 3-75。

(5) 利用狀態電驛的 a 接點來驅動輸出：狀態 S1 (M0) 時，PL1、PL2 (Y0、Y1) 動作；狀態 S2 (M1) 時，PL2、PL3 (Y1、Y2) 動作；狀態 S3 (M2) 時，PL3、PL1 (Y2、Y0) 動作。交叉分析得到圖 3-76。

(6) 將上兩圖組合以後得到完整的階梯圖，如圖 3-77

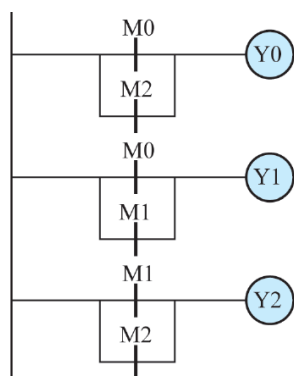
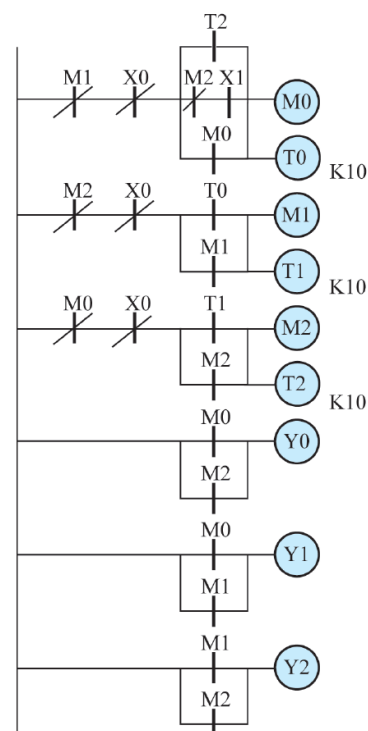


圖 3-76

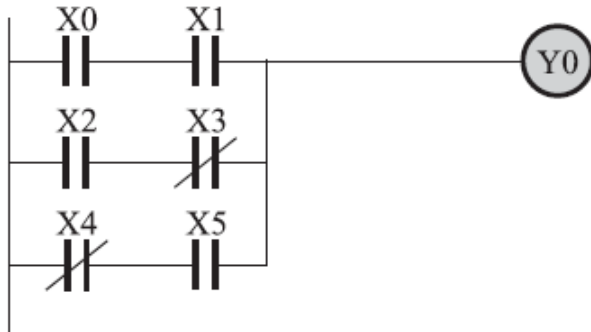


課前練習

班級：_____ 姓名：_____

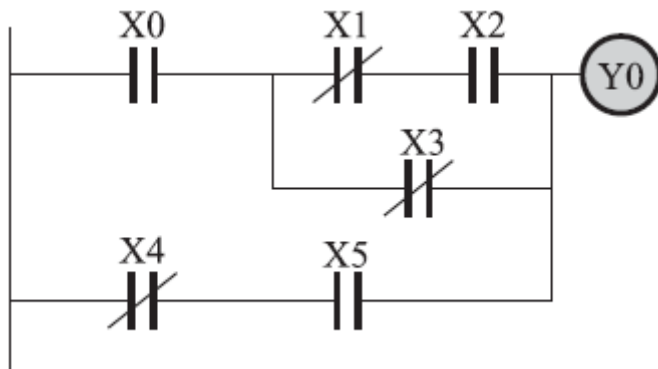
上課前，請先複習一下上次教的指令吧！把練習完成的程式寫在右方的欄位中。

1.



Sol1 :

2.



Sol2 :

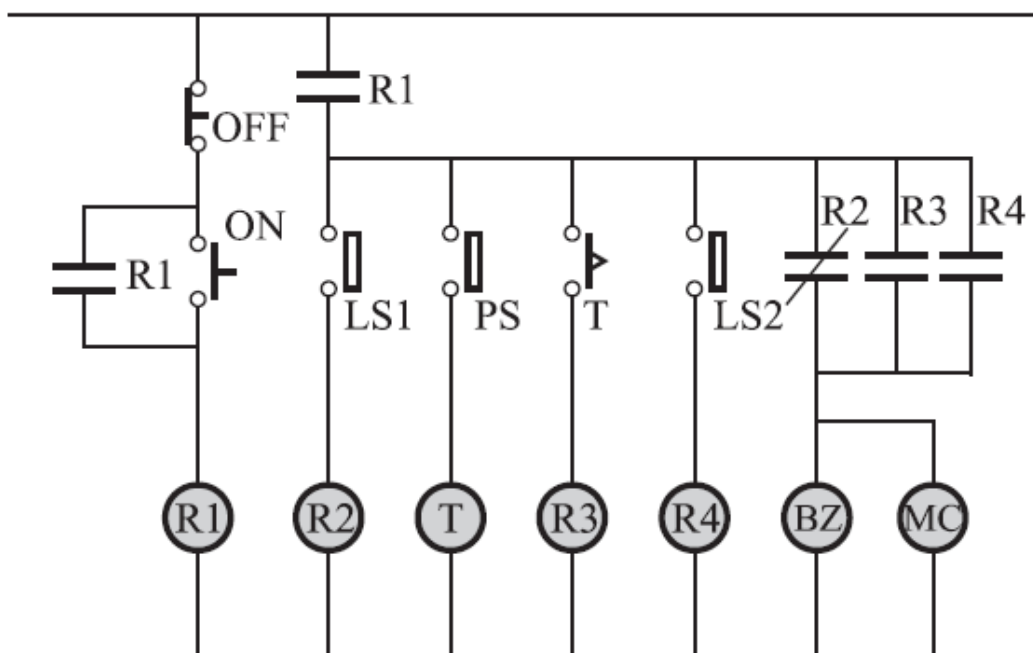
實習

1. 請使用限時電驛設計一個閃爍電路，執行時 Y0 ON→(1 秒)→Y0 OFF→(2 秒)→循環
 - 1.1 畫出電路階梯圖

1.2 將 PLC 程式碼寫下

班級：_____ 姓名：_____ 實習日期：_____

1. 請將傳統工業配線圖如下，改成使用可程式控制器後的階梯圖繪出
(提示：R1、R2、R3、R4 為電力電驛)



- 1.1 請將控制電路圖之元件分別填入並編號

輸入元件	輸出元件	內部元件

- 1.2 將控制電路圖轉成階梯圖畫出

2. 設計一個兩燈交互閃爍迴路，功能如下(不需按順序)：

- 按 PB1(X0)時，三個燈亮 (Y0、Y1、Y2)
- 按 PB2(X1)時，兩個燈亮 (Y0、Y1)
- 按 PB3(X2)時，一個燈亮 (Y0)
- 按 PB4(X3)時，燈號全熄

2.1 畫出階梯圖

2.2 寫出 PLC 程式碼

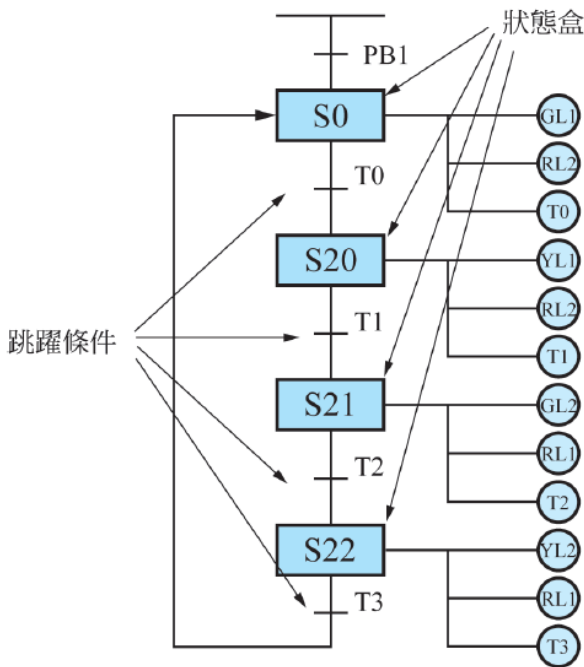
實習心得：

Ch5 狀態流程圖設計

一、狀態流程圖(SFC)解說

1. 在經過這麼多的程式設計的練習，各位應該可以發現，幾乎所有程式的設計，都是由不同的狀態所組合而成的。如果所有的狀態是依一定的順序來逐步進行的，則稱之為程序控制。若狀態的變化需依條件而改變，則為條件控制 (第三章都已介紹過)。不管是程序控制或條件控制，其實也都是由狀態所組合成的，所以使用狀態設計法來設計題目，最為恰當。但是如果狀態之間的變化愈趨複雜，則使用狀態設計法所設計出來的程式，閱讀起來就愈不容易，偵錯的工作也就不容易進行了。

為了改善這個缺失，另外提出一種較新的技巧--狀態流程圖設計法。它是狀態設計法的延伸，是將狀態設計法所分析出來的各種狀態，放入一狀態盒中，每一個狀態盒都以一狀態電驛來代表，例如 S0、S20...等，而狀態之間的關係則使用流程圖來說明。因為流程圖很容易閱讀，偵錯也很容易，所以有愈來愈多的程式設計師，都採用這種方法來設計程式。



重點說明：

- (1) S0 執行時，GL1、RL2 亮，T0 開始計時，時間到後跳到狀態 S20 執行
- (2) 狀態變更後，原狀態之電驛、輸出元件均被復歸

2. 與一般階梯圖之比較

- A. 可直接看出各狀態的變化流程，不像階梯圖要仔細的分析電路。
- B. 狀態間可依某些條件來決定它的流向。
- C. 在執行完某一狀態後，其原有的動作可以選擇自動復歸或繼續保持 (依使用的指令而定)。
- D. 同一內部元件可在不同的狀態中重覆出現 (但在同一狀態中不能出現兩次以上)，不像階梯圖只能出現一次。

3. FX3G 狀態電驛編號

編號	S0-S9	S10-S999	S1000-S4095
用途	起始狀態用	停電保持用	一般用

4. 指令編寫方法

指令名稱	動作說明	備註
SET	啟動步進點	同時關閉先前的步進點
STL	進入步進點(內部階梯圖)	
RST	關閉步進點	同時關閉先前的步進點
RET	脫離步進點(內部階梯圖) 返回系統的主要母線	一連串 SET、STL 指令後， 必須寫入 RET 指令

5. 設計方法

- (1) 在設計初，以 SFC 來紙上作業，做設計規劃。
- (2) 將 SFC 轉換為步進階梯圖，並寫出程式指令。
- (3) 以書寫器或軟體將指令鍵入。
- (4) RUN 之後使用步進階梯圖來偵錯 (如果機種沒提供 SFC 軟體偵錯功能)。

二、單一順序流程

1. 流程說明

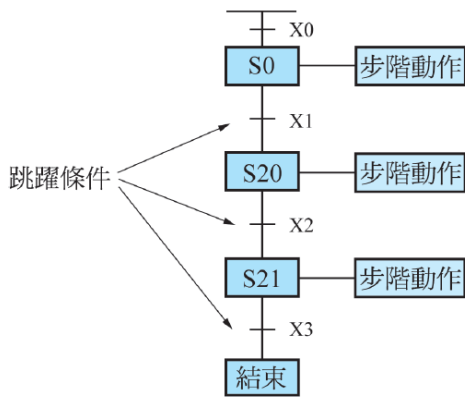
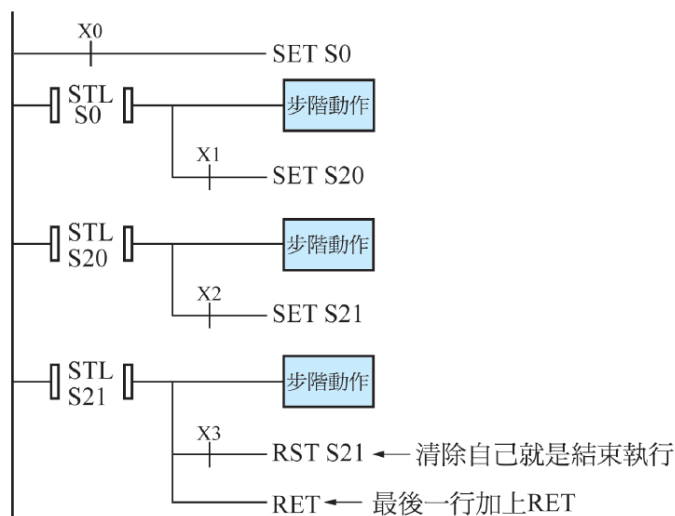


圖 5-4

從頭到尾只有一條路可走，這稱為單一順序流程圖。依一定順序逐步執行且沒有分歧，所以屬於單一順序流程。右圖中在 S21 執行完後即結束。在步進階梯圖中是以清除 (RST) 正在執行的步階來結束步進動作。

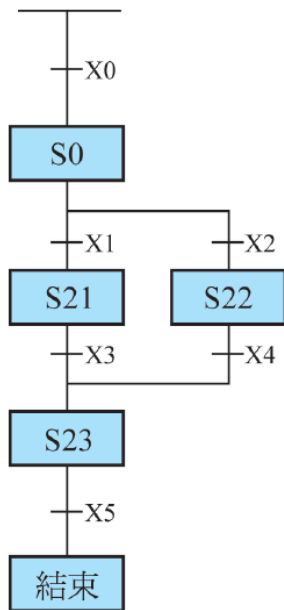
2. 步進階梯圖



指 令	
LD	X0
SET	S0
STL	S0
步 階 動 作	
LD	X1
SET	S20
STL	S20
步 階 動 作	
LD	X2
SET	S21
STL	S21
步 階 動 作	
LD	X3
RST	S21
RET	
END	

二、選擇分歧、合流流程設計

1. 流程說明



左邊的流程圖，當 S0 執行完後，分歧成兩條路徑，而且只能選擇其中一條路徑來走，這種有分歧的方式稱為選擇分歧。流程說明如下：

- A. 當 S0 執行後，若 X1 先按下，則跳到 S21 執行，此時就算按下 X2 開關，S22 也無法動作，因為程序已到了 S21。之後若 X3 按下，則脫離 S21 而跳到 S23 執行，再按下 X5 開關後，則程式結束。
- B. 若 S0 執行後，X2 先按下，則跳到 S22 執行，此時就算按下 X1 開關，S21 也無法動作，因為程序已到了 S22。之後若 X4 按下，則脫離 S22 而跳到 S23 執行，再按下 X5 開關後，則程式結束。

2. 步進階梯圖

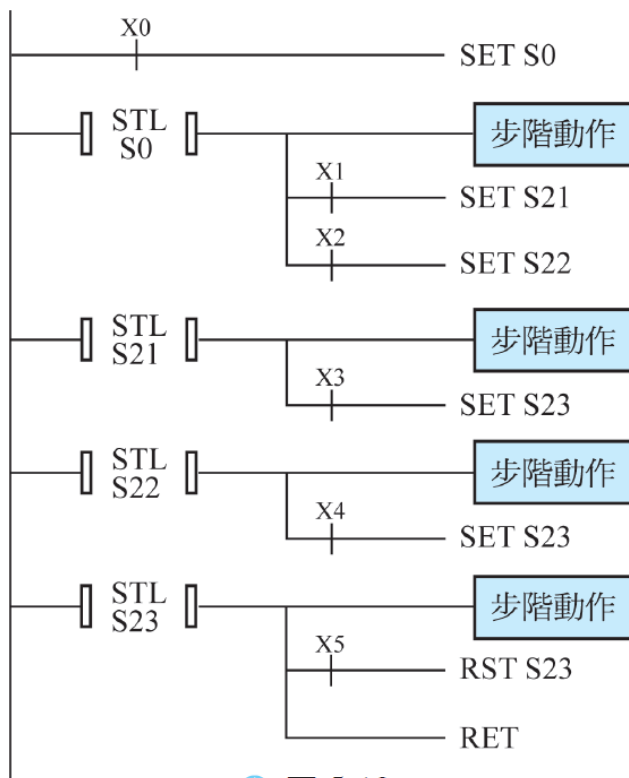
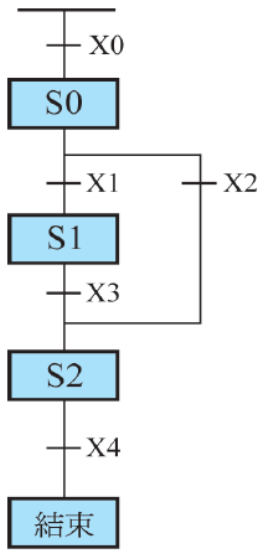


圖 5-10

指令	
LD	X0
SET	S0
STL	S0
步階動作	
LD	X1
SET	S21
LD	X2
SET	S22
STL	S21
步階動作	
LD	X3
SET	S23
STL	S22
步階動作	
LD	X4
SET	S23
STL	S23
步階動作	
LD	X5
RST	S23
RET	
END	

三、狀態跳躍流程設計

1. 流程說明



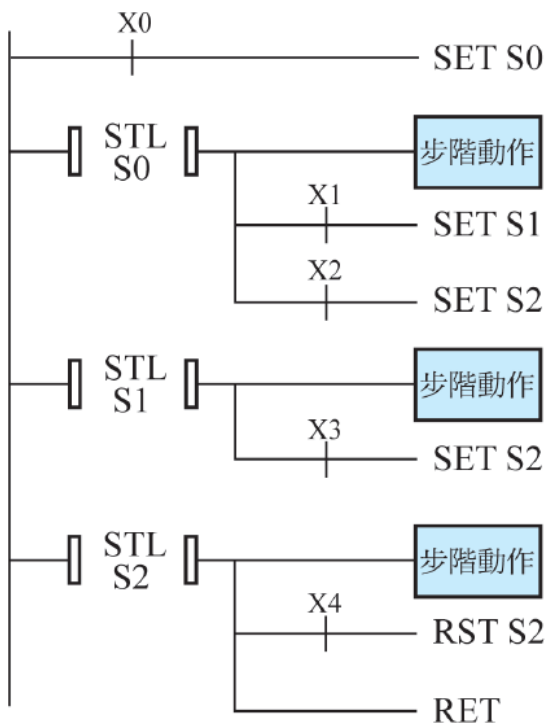
左邊的流程圖，當 S0 執行完後，分歧成兩條路徑：

A.當 S0 執行後，若 X1 先按下，則跳到 S1 執行，此時就算按下 X2 開關，S2 也無法動作，因為程序已到了 S1。之後若 X3 按下，則脫離 S1 而跳到 S2 執行，再按下 X4 開關後，則程式結束。

B.若 S0 執行後，X2 先按下，則直接脫離 S0 而跳到 S2 執行，再按下 X4 開關後，則程式結束。像這種跨躍某些步階而直接跳到另一步階的動作稱之為狀態跳躍。

C.狀態跳躍的目的地，一般均無限制，只要條件符合，就可以跳離原步階而進入另一步階。

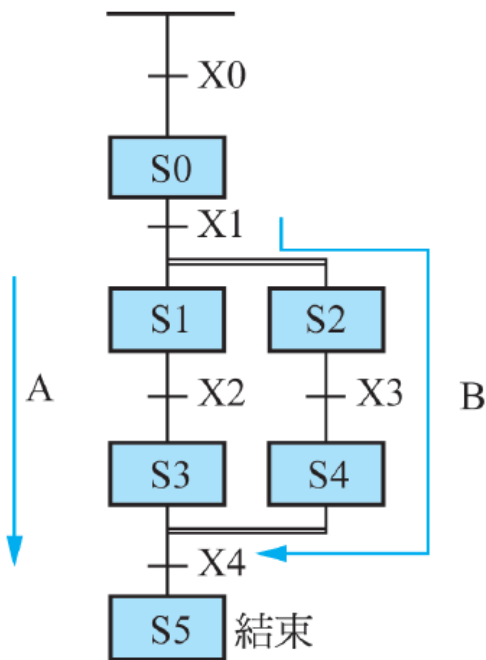
2. 步進階梯圖



指令	
LD	X0
SET	S0
STL	S0
步階動作	
LD	X1
SET	S1
LD	X2
SET	S2
STL	S1
步階動作	
LD	X3
SET	S2
STL	S2
步階動作	
LD	X4
RST	S2
RET	
END	

四、並進分歧、合流流程設計

1. 流程說明



左邊的流程圖，當 S0 執行完後，若按下 X1 開關，則 S1 及 S2 同時執行，也就是箭頭所指的兩條路徑可以同時執行。這種分歧的方式稱為並進分歧流程。流程說明如下：

A. 當 S0 執行後，若 X1 按下，分成兩條路徑同時執行：

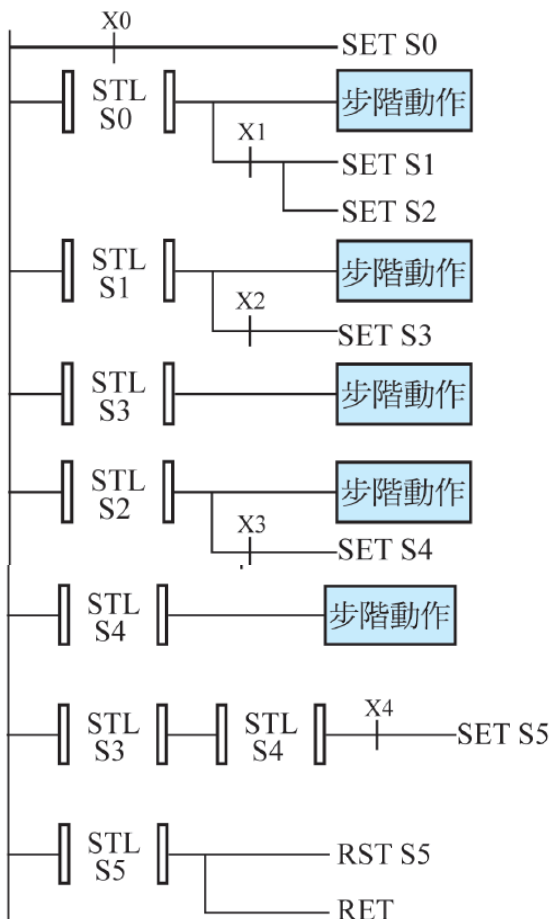
a. 跳到 S1 執行，之後若 X2 按下，則脫離 S1 而跳到 S3 執行。

b. 跳到 S2 執行，之後若 X3 按下，則脫離 S2 而跳到 S4 執行。

B. 當 S3 及 S4 都已執行後，若 X4 按下，則脫離 S3 及 S4 而跳到 S5 執行，程式結束。

C. 假若左邊路徑已執行到 S3，而右邊路徑尚停留在 S2 時，就算你按了 X4，也不會跳到 S5 執行，必須等到 S4 執行後，才會繼續往下執行，像這種有等待功能的方式，稱之為並進合流。

2. 步進階梯圖



指令	
LD	X0
SET	S0
STL	S0
步階動作	
LD	X1
SET	S1
SET	S2
STL	S1
步階動作	
LD	X2
SET	S3
STL	S3
步階動作	
STL	S2
步階動作	
LD	X3
SET	S4
STL	S4
步階動作	
STL	S3
STL	S4
LD	X4
SET	S5
STL	S5
RST	S5
RET	
END	