

## 【範例】專題製作報告書格式例範

一、書背：如範例(一)封面：如範例(二)

二、口試委員簽名：如範例(三)

三、誌謝：如範例(四)目錄：如範例(五)

四、內文：

(一)如範例六

(二)每頁字數：

約一千六百字（一行40字\*40行），以A4紙之鐳射或噴墨印表機列印，但每頁上、下、左、右請各留2.5公分，以便編排頁碼及裝訂。

(一)字體大小及行距規定：

1.標題：中文用標楷體、英文用Times New Roman，16字。

2.內文：中文用標楷體、英文用Times New Roman，12字。

3.行間距：採用單行行距打字(SingleSpace)。

三、文字分段敘述之編號：

以一、(一)、1、(1)、(i)為序，文字敘述中之數字一律用阿拉伯數字表示。

四、請以 Office2003 或以上版本之軟體。

範例：(一)書背

建國科技大學自動化系暨機電光所專題製作

樓梯行走車

一〇三學年度

專題題目

專題製作完成學年度

範例：(二) 封面

# 建國科技大學 自動化系暨機電光所

請以淡黃色封面裝訂

※若您找不到相同的顏色  
紙張，請洽建國科技大學  
下坡左轉→內堡潭實業社  
→告知要用淡黃色 12 號

103學年度

專題製  
作完成  
學年度

## 專題製作報告書

### 樓梯行走車

指導教授：謝○○ 老師

班 級：日間部(四技) 三年甲班

日間部四技  
三年甲班

組 長：郭○○ 10109001

組 員：施○○ 10109002

葉○○ 10109003

廖○○ 10109004

陳○○ 10109005

中華民國○○○年○○月

專題製作完成年/月

範例：(三) 指導教師簽名

建國科技大學

自動化系暨機電光所一〇三學年度專題製作報告

題目：樓梯行走車

姓名：洪○○

學號 10109001

姓名：林○○

學號：10109002

姓名：鍾○○

學號：10109003

姓名：黃○○

學號：10109004

姓名：趙○○

學號：10109005

指導教師：\_\_\_\_\_

中華民國 ○○ 年○ 月

## 範例：（四）誌謝

### 誌謝

感謝指導教授 謝○○主任一年來的細心指導與教誨，並不時花費時間在觀念上進行討論，使本專題得以順利完成，且在報告撰寫過程中費心指出許多問題，讓報告的基礎得以成型，而不論在研究態度與待人接物方面均受益良多，並常提供一些日常生活中的經驗談，在此謹致上最誠摯之敬意與謝忱。而口試期間，還要感謝兩位口試委員 詹○○老師及 林○○老師對於本專題的指正和建議，使得本專題不完美之處得以更加完整，在此亦對兩位老師致上謝意。

另外，特別感謝系上老師們的指導與鞭策。更有賴許多朋友在精神上給予鼓勵與支持，甚或實質協助，我們才能順利完成專題

最後要感謝我們的父母與家人，若不是他們長久以來的支持，不可能有今天的小小成果，僅以此篇報告表達內心最誠摯的感激。

## 目錄

指導教師簽名表	
誌謝	
目錄	I
圖目錄、表目錄	II
中文摘要	III
英文摘要	IV
第一章 緒論	
1-1 研究背景	1
1-2 研究目的	1
1-3 製作流程	2
1-4 系統架構	2
1-5 系統操作流程圖	3
第二章 可程式控制器之簡介	
2-1 可程式控制器與 VB 之通訊架構	
(一)通訊協定	4
(二)R S -232	4
2-2 可程式控制器配線圖	7
2-3 門鎖鑽孔 X - Y 軸定位平台系統內外部實際圖簡介	
(一)控制面板	8
(二)控制箱內部	9
(三)門鎖鑽孔 X - Y 定位平台	10
2-4 伺服馬達驅動器位置模式相關參數設定	11
2-5 可程式控制器程式說明	14
2-6 S F C 架構圖	17
2-7 P L C 階梯圖	19
第三章 VB 圖控畫面設計	
3-1 系統主畫面	27
3-2 系統監控畫面	28
3-3 程式說明	
(一)系統主畫面程式內容	29
(二)系統監控畫面程式內容	30
第四章 結論	36
參考文獻	37
附錄 A 可程式控制器性能規格	38
附錄 B Visual Basic 元件介紹	40
附錄 C F X 2 可程式控制器下達命令的種類	41
附錄 D V B 與 P L C 間的通訊—M S c o m m 控制項	42
附錄 E P L C 特殊指令說明程式	43

## 圖目錄

圖 1.1	系統操作流程圖-----	3
圖 2.1	非平衡性 RS-232-----	5
圖 2.2	RS-232 信號傳遞方塊示意-----	6
圖 2.3	RS-232 位元組資料格式-----	6
圖 2.4	傳送過程時序圖-----	6
圖 2.5	可程式控制器配線圖-----	7
圖 2.6	控制面板-----	8
圖 2.7	控制箱內部圖-----	9
圖 2.8	X-Y 軸定位平台-----	10
圖 2.9	加工定位點座標圖-----	13
圖 2.10	SFC 架構圖-----	17
圖 3.1	系統主畫面-----	27
圖 3.2	系統監控畫面-----	28

## 表目錄

表 2.1	RS-232 接腳(DB9 與 DB25 腳位) -----	5
表 2.2	位置模式之參數 -----	11
表 2.3	輸出輸入點之規劃 -----	14
表 2.4	PLC 之程序說明 -----	15
表 A.1	可程式控制器功能簡介 -----	38
表 C.1	元間名稱與位址對照表 -----	41
表 D.1	串列埠通訊工具屬性說明 -----	42
表 E.1	串列埠通訊工具屬性說明 -----	43

## 中文摘要

本專題結合人機介面 MMI(Man-Machine Interface)，PLC 及伺服馬達來製作門鎖鑽孔定位監控系統。目前，PC-Based 及可程式控制器(PLC)為核心所建構的自動化工廠的整合生產型態，正在我們國家的整個產業結構的變化過程中擴散生根。對於有一定規模的生產系統一般採用分散式控制系統，在這種系統中，長久以來 PLC 擔任著第一線的控制器，負責單能機械或系統局部的步序控制。PLC 的功能常被定位在「控制機台動作」。而在人機介面上的功能一直相當薄弱，以至於以往機台開發工程師經常要設計偌大的操作面板，以作為人機的溝通管道，但都未能善盡人意。

隨著電腦的普及，及軟體的進步，促使 PLC 朝向結合電腦科技方向發展。近幾年，windows 的人機介面軟體已陸續被開發出來，以及視覺化的程式設計軟體日趨實用化，使得友善的圖形監控方式已非遙不可及的事。圖形監控已成為目前線上自動化工程的一個熱門話題，本專題計畫以可程式控制器控制 XY 平台及鑽床構成之鑽孔加工平台，進行門鎖零件之多孔定位及鑽孔加工。並以 VB 設計圖形監控程式來達到 PC 監控 PLC 的目地，以強化系統之人機介面功能。



## **Abstract**

The purpose of this project is to design an automated machine for precisely position and drilling parts of door locks. At present, the automatic production integrated by PC-based and Programmable Logic Controller (PLC) has general been accepted during the nation-wide industrial transformation, and the separate controlling system has generally been adopted by production systems in large size. In such a system, the PLC has long played a role of the preliminary controller which is responsible for controlling the order of the mono-functional machines or a certain area in that system. The function of PLC is often defined as to control the operation of the machines, and its function in the MMI aspect, on the other hands, is unknown. Therefore, in the past R&D engineers often had to design a huge operation panel as the means for communications between human beings and machines, which seemed not desirable.

With the popularity of computers and the improvement of software, the PLC has also been developed to be combined with the computer technology. Nowadays, since the MMI was invented continuously and the visual programming software was designed to be more practical, graphic monitoring has become foreseeable. Graphic monitoring is now a hot topic when studying automation engineering. In this project, we will perform multi-hole positioning and machining on the lock components with PLC to control the XY panel and the drilling surface. In addition, we will design a graphic monitor program with VB to meet our goal of monitoring the PLC with PC, which will eventually strengthen the function of MMI.

## 第一章 緒論

### 1-1 研究背景

大約在 1968 年美國最大的汽車工廠，推出的汽車款式越來越多，在生產線上，為應付各種款式的控制，其控制器需面臨經常更改的情況，而且計時器或計數器也無法達到同步操作，因此發展出了可程式控制器，用來取代傳統繼電器控制配電盤。配電盤控制設計上極為方便、簡潔，然而對於功能要求稍複雜或動作要求稍多，傳統配電盤控制器將會因大量的配線而險的體積龐大及無效率，同時成本增加維修亦困難，因此如何將生產線自動化，開啟電腦除了商業、科學外的另一個重要用途。

隨著經濟的發展及產業結構的改變，工廠朝著使用自動化生產的方向是無可避免的。因此可程式控制器(Programmable Logic Controller, 簡稱 PLC)有著優異功能及高穩定性，所以在工廠自動化方面一直扮演著重要的角色，常被用來建構分散式控制系統，用來控制現場設備。

傳統的鑽床工作皆須以人工方式或按鈕方式來移動工件位置，管理人員也無法了解生產現況及突發狀況，如此的方式已經不符合現在競爭的社會，而且人工成本過高，獲利有限。

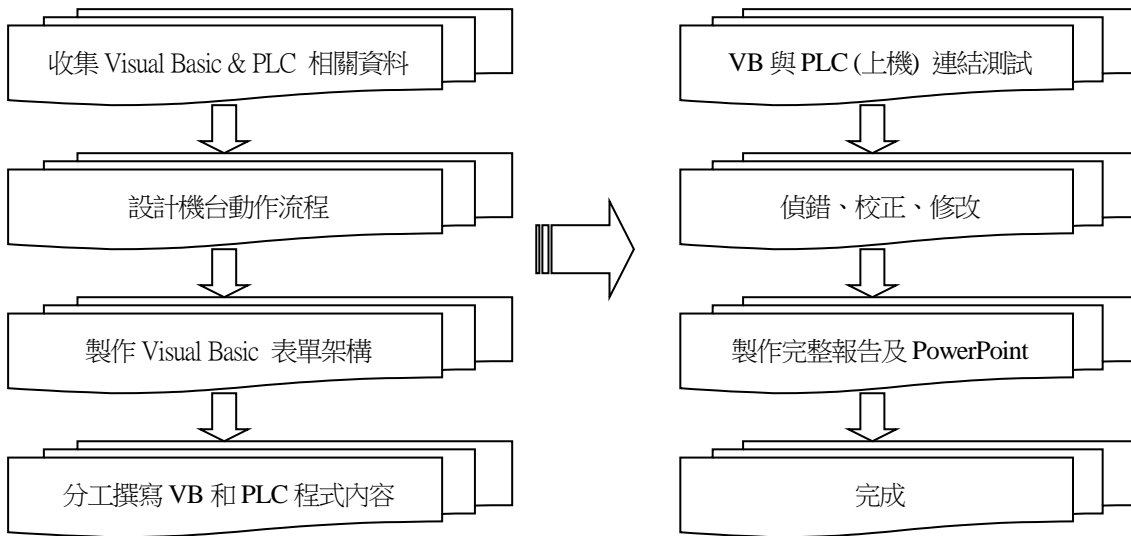
有鑑於此，運用 VISUAL BASIC(簡稱 VB)設計出適合的人機介面來做圖形監控，利用個人電腦的 RS232 通訊埠，使控制的範圍擴大，且不必侷限於現場才能操作，只要有電腦的地方安裝 VB 的人機介面，透過網路的通訊協定便可達到及時監控的效果，並監控系統及生產保全，使生產的成本減低。採用一般 windows 作業系統所能接受的視覺化物件導向程式設計來發展圖控系統，是未來的趨勢，也是工廠自動化的理想解決途徑。

### 1-2 研究目的

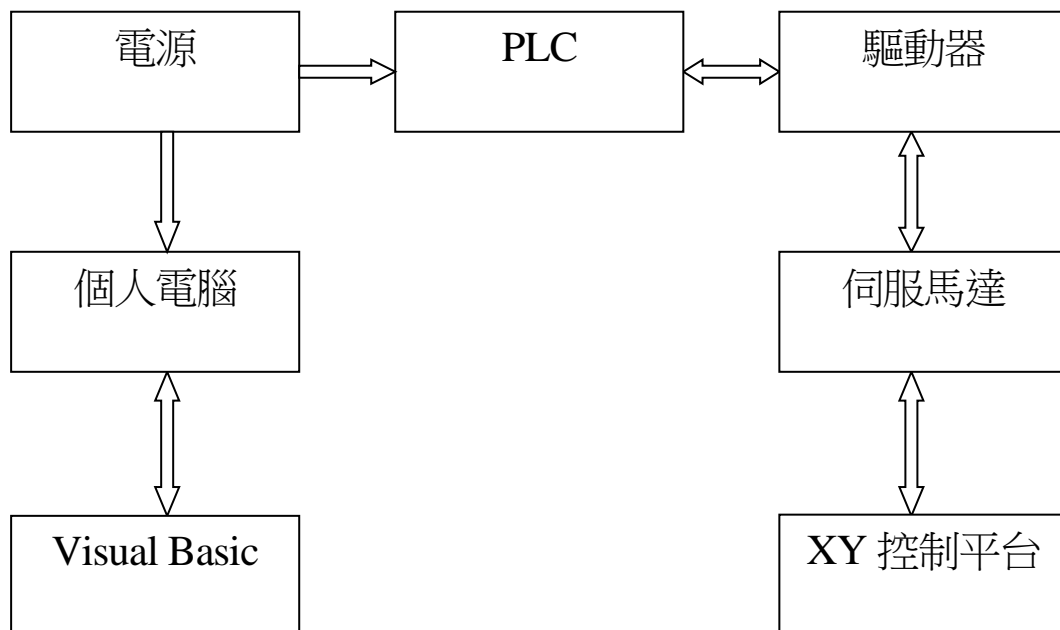
目前圖形監控在生產設備和自動化控制等應用非常廣泛，現今市面上人機介面也有許多形式的模組被應用在生產設備中。運用個人電腦撰寫 Visual Basic 物件導向程式及設計可程式控制器電路，利用個人電腦的 RS-232 通訊埠或是網路通訊協定透過可程式控制器及 PC 來控制設備或遠端負載。近代 PLC 大都具有 RS232/RS485 串列通訊埠，PC 可透過此通訊埠和 PLC 達成資料傳輸。要事先制定通訊協定，以便 PLC 在接收到資料後能解讀資料的涵義。PC 和 PLC 是主從關係，PC 算是主端(master)，為主動端；而 PLC 算是從端(slave)，為從動端。PC 端基於輪詢(polling)方式向 PLC 讀取或寫入資料。每一次溝通的達成，首先是由 PC 端主動發出命令(command)，在 PLC 端依據接收到的命令進行解讀，之後回應(response)給 PC，命令和回應都有一定格式(format)。

本專題是在 Visual Basic 6.0 下進行設計圖形監控程式來達到 PC 監控 PLC，以 PLC 控制雙軸鑽床多孔定位，並提升速度及準確度，以利業界快速精密加工。

### 1-3 製作流程



### 1-4 系統架構



### 1-5 系統操作流程圖

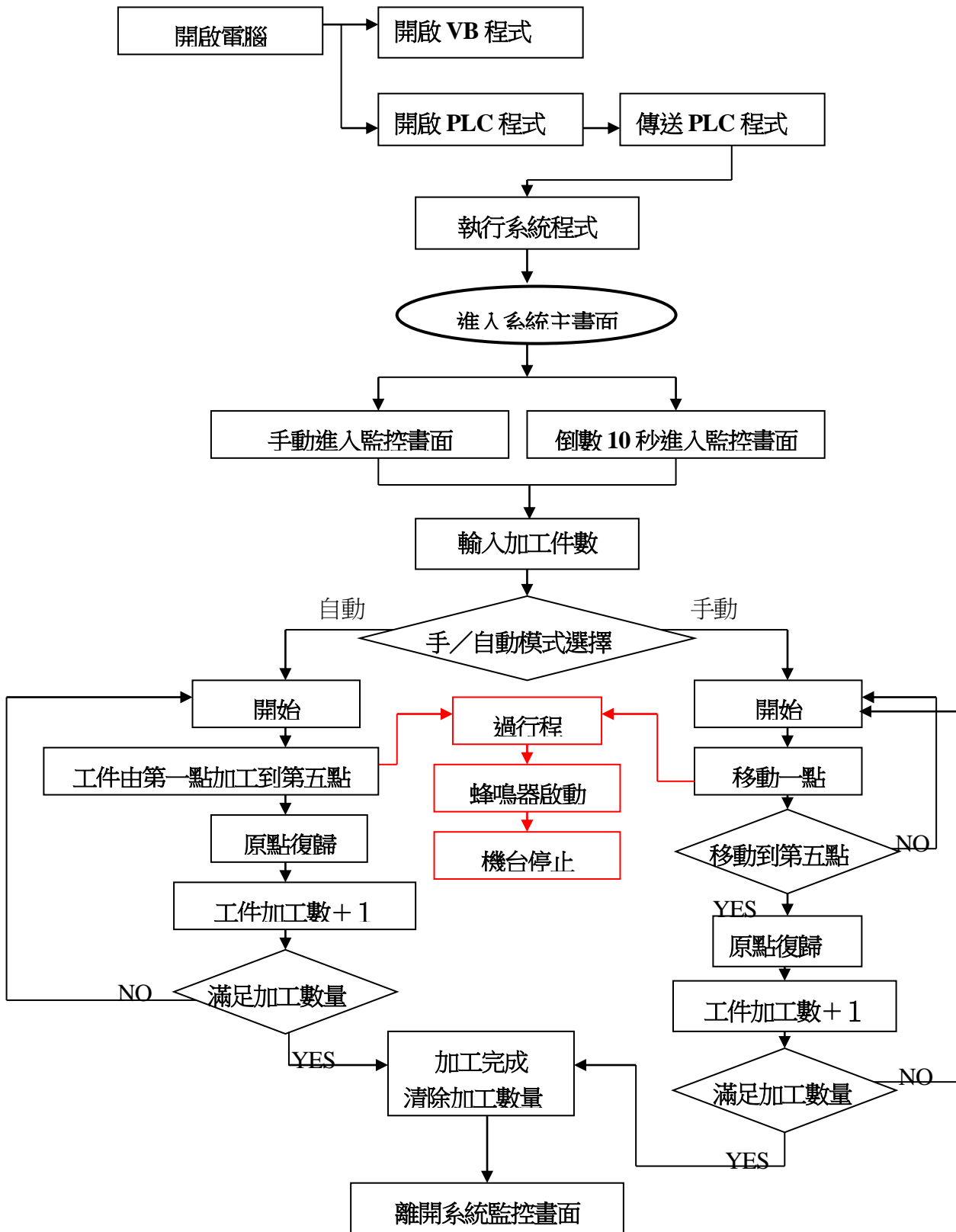


圖 1.1 系統操作流程圖

## 第二章 可程式控制器之簡介

### 2-1 可程式控制器與 VB 之通訊架構

#### (一) 通訊協定

通訊協定是兩裝置之間為達到資料傳輸所制定的傳輸格式。

系統整合的目的是在各部門(單元)專業分工之下資源共享，以締造最佳工作效率。由前幾段述及的，不管是分散式控制的，或是圖形監控的，都需要 PLC 和電腦連線，這之中，兩裝置之間訊息互傳技術是系統整合的基本需要。PLC 圖形監控就是一種系統整合意義，PC 和 PLC 分工合作，PC 負責 MMI，而 PLC 負責機台控制。

PC 和 PLC 兩裝置之間的資料傳輸主要是靠兩層通訊協定(protocol)達成的，所以兩裝置與兩層通訊協定必須一致。第一層是指硬體層級的通訊協定，如常用的標準介面有串列式的 RS-232、RS-422 或 RS-485 等等，他制定兩方電器信號、準位、接線法、以及字元資料包裝格式的規範，串列式傳輸他每次一位元，又以傳送一位元組為一單位(有頭有尾的包裝)。第二層是指字串傳送層級通訊協定，如何將一筆有意義的字元的字串傳送給對方解讀。

#### (二) RS-232

在第一層級的串列通訊協定中，RS-232 是一種較早發現的介面，現今被廣泛應用於個人電腦，辦公室各裝置之間的資料傳輸方式，諸如滑鼠、繪圖機和量測儀錶等等都提供 RS-232 埠，以便能和其他裝置連線。但 RS-232 他是一種非平衡行(unbalance)的傳輸型態，如(圖 2.1)所示，導線分佈有電阻和雜散電容，而導致兩端地線無法獲得一致準位，可能會有回路回授(loop back)現象，因而較易受到 EMI 和叉音(cross talk)的干擾，信號減弱(S/N 比較小)以至於傳輸頻寬較低(保持率低)、最大傳輸距離較短等缺點。

鑒於此，後來就發展差動式的 RS-422/RS-485，以改善頻寬和傳輸距離。RS-232 只能一裝置對一裝置傳輸，最高速度限制為 19200 率(baud rats, dits/sec)，傳送資料的最大距離為 15m。雙工的 RS-232 最少要 3 條線，一條地線和兩條信號線傳送收用，如(表 2.1)中的 SG, TXD, RXD 等接腳，接線法如(圖 2.2)所示。在使用 RS-232 時，裝置雙方必須設定一致的通訊協定，如起始位元(start bit)、資料位元數(data bits)、同位元(parity)和停止位元(stop bit)等等，在雙方有一致的通訊協定下，接收方始能檢知信號意義。資料組成格式如(圖 2.3)，其中同位元分奇位元(odd)和偶位元(even)兩種，此位元為一錯誤查碼用途。傳送過程之時序如(圖 2.4)所示，傳送方以起始重置對方震盪脈波，以獲得兩地有一致相同的工作時脈(clock)，逐一傳送位元，直至停止位元，完成一位元組(byte)之傳送。傳送過程兩裝置若有做交握檢查，則要再增加 RTS、CTS、EQR 等接線，詳細資料得查介紹 RS-232 的專書。

表(2.1) RS-232 接腳(DB9 與 DB25 腳位)

信號	符號	DB9 接腳	DB25 接腳	說明
Carrier detector	CD	1	8	載波檢測
Transmit data	TXD	2	3	發送資料
Receive data	RXD	3	2	接收資料
Data Terminal ready	DTR	4	20	資料終端備妥
Signal ground	SG	5	7	信號接地
Data set ready	DSR	6	6	資料備妥
Request to send	RTS	7	4	要求發送
Clear to send	CTS	8	5	清除發送
Frame ground	FG		1	機殼接地

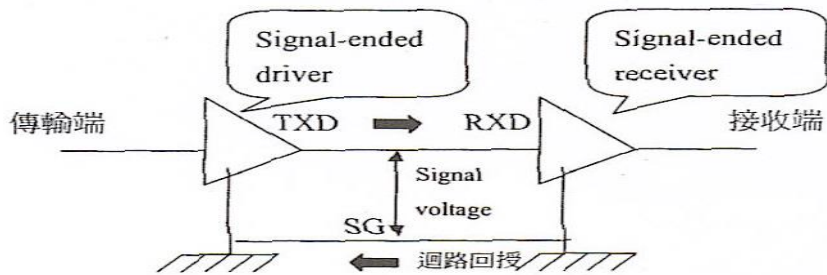


圖 2.1 非平衡型 RS232

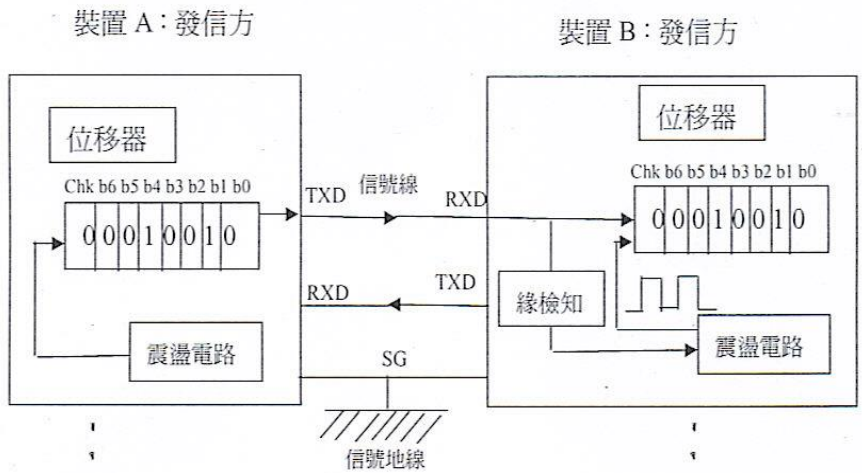


圖 2.2 RS232 信號傳遞方塊示意

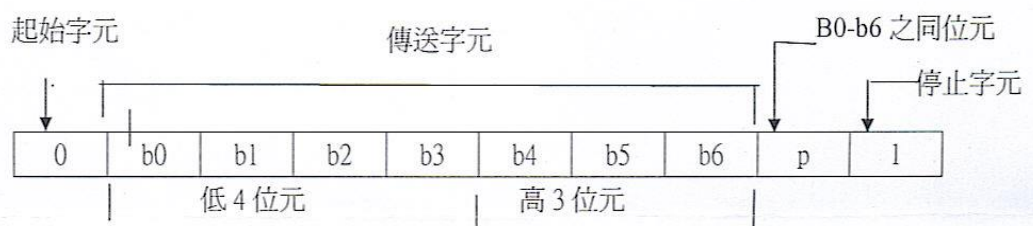


圖 2.3 RS232 位元組資料格式

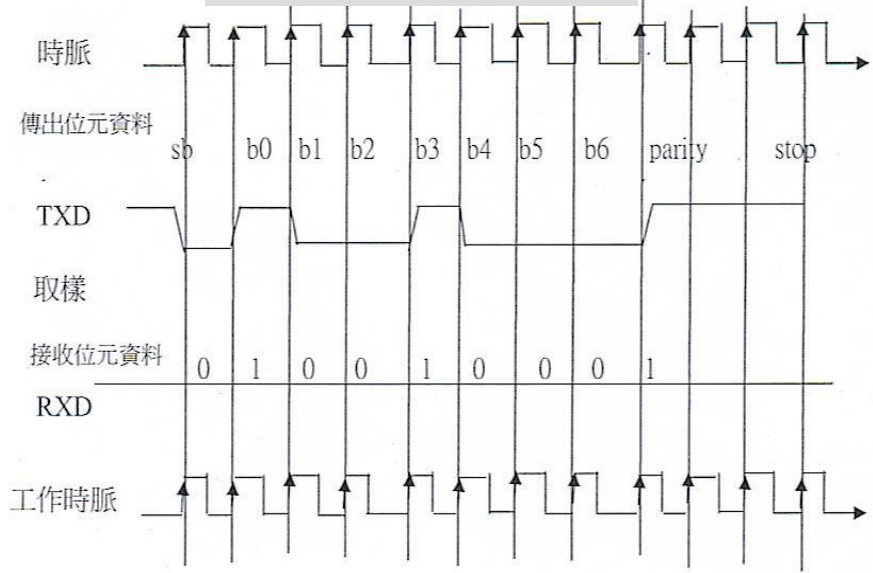
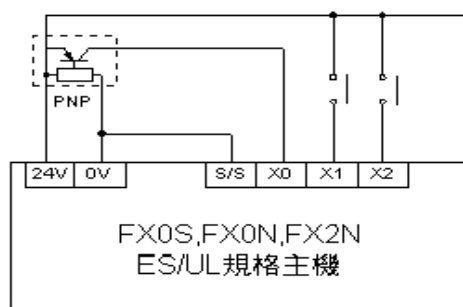


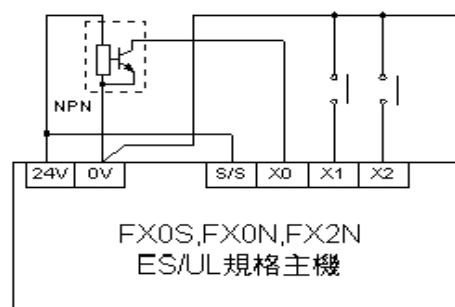
圖 2.4 傳送過程時序圖

## 2-2 可程式控制器配線圖：

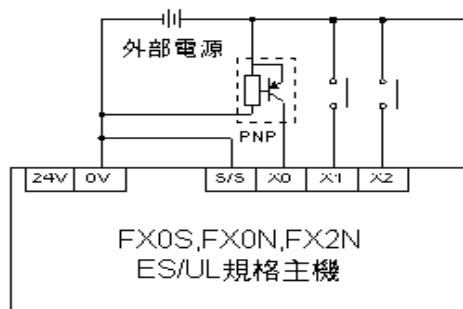
PNP元件使用主機的DC24V電源



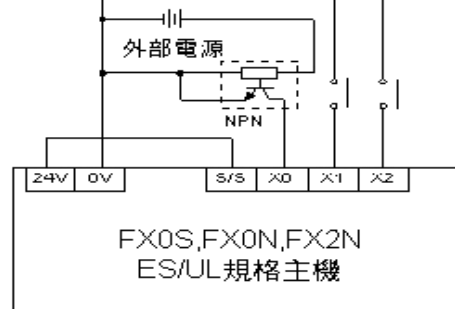
NPN元件使用主機的DC24V電源



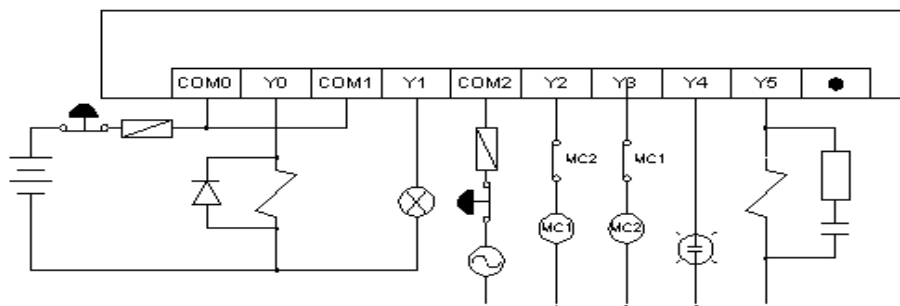
PNP元件使用外部的DC24V電源



NPN元件使用外部的DC24V電源



繼電器(Realy)輸出配線範例



電晶體(Transistor)輸出配線範例

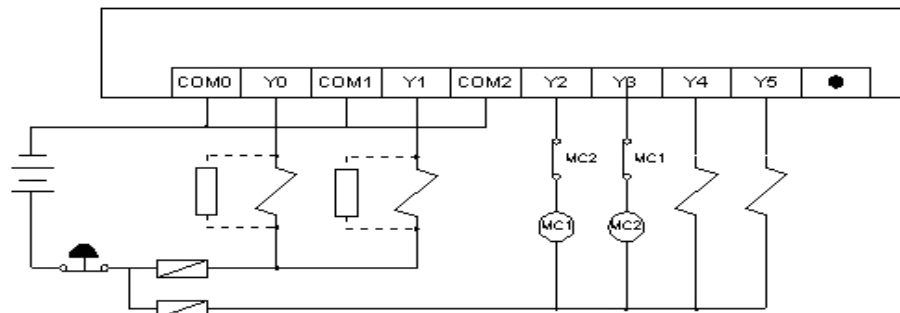


圖2.5 可程式控制器配線圖圖



## 2-3 門鎖鑽孔 X-Y 軸定位平台系統內外部實際圖簡介



圖 2.6 控制箱面板

控制箱面板說明：

按 鈕 開 關 (ON) =====作用為啟動電源，啟動後亮燈。

按 鈕 開 關 (啟動) =====作為執行程式按鈕，啟動後亮燈。

蜂 鳴 器 (BZ) =====系統過載，發出警報。

緊 急 停 止 鈕 (STOP) =====緊急斷電，機台停止動作。

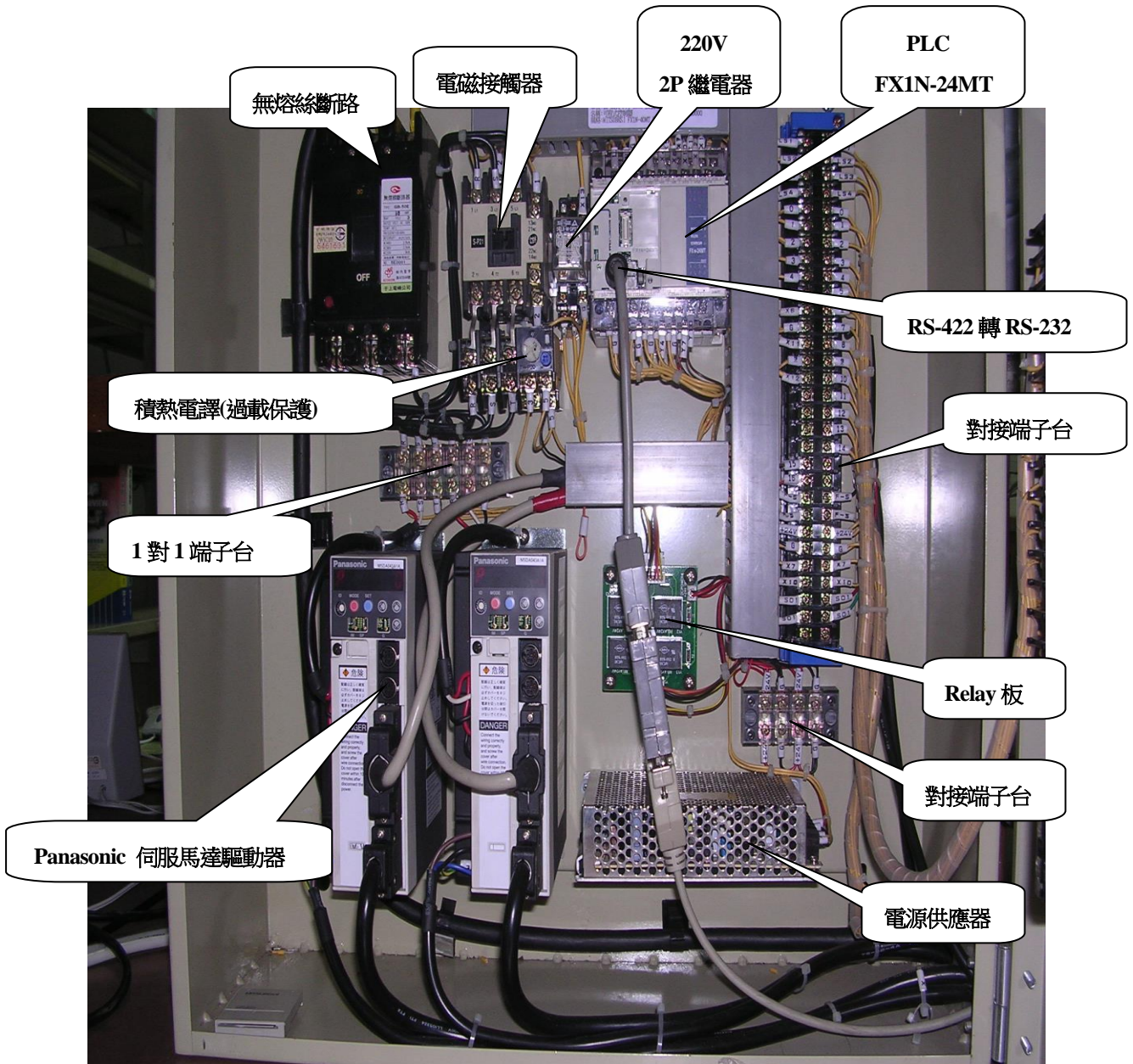
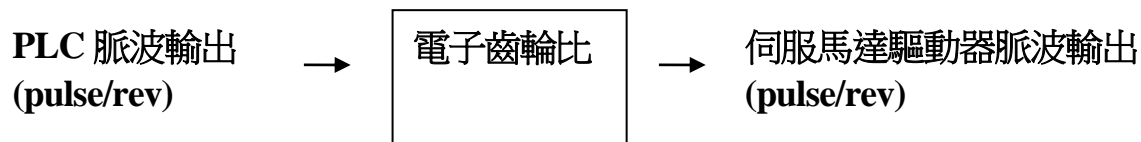


圖 2.7 控制箱內部圖



## 2-4 伺服馬達驅動器位置模式相關參數設定:

滾珠螺桿規格：1 pitch/rev, 5 mm/pitch. PLC 之脈波輸出(pulse/rev), 電子齒輪比及伺服馬達驅動器脈波輸出(pulse/rev)三者關係如下：



其中電子齒輪比計算式為:

$$\text{電子齒輪比} = \frac{\text{Pr } 46 * 2^{\text{Pr } 4A}}{\text{Pr } 4B}$$

式中 Pr46, Pr4A 及 Pr4B 為驅動器位置模式之參數，如表 2.2 所示：

表(2.2)位置模式之參數

參數號碼 (Pr□□)	參數功能	設定範圍	內定值	單位
40	脈波倍頻倍數設定	1 ~ 4	4	-----
44	每轉輸出脈波數	1 ~ 16384	2500	pulse/rev
46	指令分同倍數分子 (電子齒輪比分子)	1 ~ 10000	10000	-----
4A	指令分同倍數分子倍率 (電子齒輪比分子 * (2 <sup>n</sup> ))	0 ~ 17	0	2 <sup>n</sup>
4B	指令分同倍數分母 (電子齒輪比分母)	1 ~ 10000	10000	-----

查表若內定值 Pr44 = 2500 pitch/rev，Pr40 = 4，則驅動器馬達旋轉一圈所需之脈波數為 2500 \* 4 = 10000。又若電子齒輪比設為 1：1，則應設定相關參數 Pr46 = 10000，Pr4A = 0，Pr4B = 10000，即：

由上述得知欲驅動馬達轉動一圈，PLC 須送給驅動器 10000 個脈波，

$$\begin{aligned}
 \text{電子齒輪比} &= \frac{\text{Pr } 46 * 2^{\text{Pr } 4A}}{\text{Pr } 4B} \\
 &= \frac{10000 * 2^0}{10000} \\
 &= \frac{1}{1}
 \end{aligned}$$

即 10000 pitch/rev。

由於馬達旋轉一圈，相當於滾珠螺桿一個 pitch，即 1 pitch/rev。

而滾珠螺桿為 5mm / pitch，則 PLC 每送出一個 pulse 使加工平台之移動距離為：

$$\begin{aligned}
 & (5\text{mm} / \text{pitch}) * (1 \text{ pitch/rev}) / (10000) \text{ pulse/rev} \\
 &= (5 \text{ mm} / \text{pitch}) * (1 \text{ pitch/rev}) * (1/10000 \text{ rev/pulse}) \\
 &= (5 / 10000) \text{ mm} / \text{pulse} \\
 &= 0.5 \mu\text{m} / \text{pulse}
 \end{aligned}$$

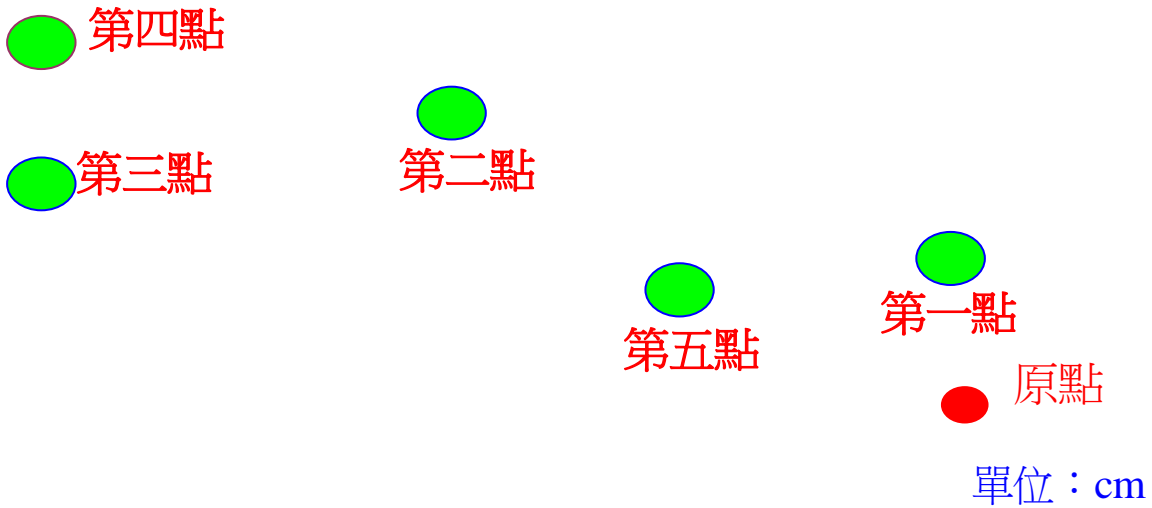


圖 2.9 加工定位點座標圖

送出脈波的計算：

Ex：平台移動 10mm，要送出多少脈波？

$$1\text{cm} = 10\text{ mm} = 10000\text{ }\mu\text{m}$$

$$10000\text{ }\mu\text{m} / 0.5\text{ }\mu\text{m} / \text{pulse} = 20000\text{ pulse}$$

Ex: 第一點至第二點距離計算：

$$(X2-X1) = (37.96\text{ mm} - 27.75\text{ mm}), \text{脈波數} = 204200\text{ pulse}$$

$$(Y2-Y1) = (64.15\text{ mm} - 46.65\text{ mm}), \text{脈波數} = 35000\text{ pulse}$$

## 2-5 可程式控制器程式說明

本節是針對 PLC 程式，INPUT、OUTPUT 接點，見表(1.3)、階梯圖程序內容，見表(1.4)、SFC 架構圖(見 2-6 節)、以及 PLC 階梯圖作簡介，(見 2-7 節)。

表(2.3)為輸出輸入點之規劃

INPUT	功能	備註	OUTPUT	功能	備註
X0	S-RDY	X 軸	Y0	PULS	X 軸
X1	ALM		Y1	PULS	Y 軸
X2	COIN		Y2	SIGN	X 軸
X3	S-RDY	Y 軸	Y3	SIGN	Y 軸
X4	ALM		Y4	SRV-ON	X 軸
X5	COIN		Y5	SRV-ON	Y 軸
X6	啟動鈕		Y6	蜂鳴器	
X7	X 軸原點		Y7	啟動燈	
X10	Y 軸原點				
<b>暫存器</b>					
接點	功能	備註	接點	功能	備註
D0	X 軸 Ø 6.5 第一點		D42	控制速度用	
D2	X 軸 Ø 6.5 第二點		D44	控制速度用	
D4	X 軸 Ø 6.5 第三點		D46	控制速度用	
D6	X 軸 Ø 6.5 第四點		D48	控制速度用	
D8	X 軸 Ø 6.5 第五點		D50	X 軸相差距離暫存器	
D10	定位暫存器		D70	Y 軸相差距離暫存器	
D20	Y 軸 Ø 6.5 第一點		D90	X 軸速度暫存器	
D22	Y 軸 Ø 6.5 第二點		D92	X 軸距離暫存器	
D24	Y 軸 Ø 6.5 第三點		D94	Y 軸速度暫存器	
D26	Y 軸 Ø 6.5 第四點		D96	Y 軸距離暫存器	
D28	Y 軸 Ø 6.5 第五點		D98	回 X、Y 軸原點	
D40	控制速度用		D100	加工次數	
<b>輔助電譯</b>					
接點	功能	備註	接點	功能	備註
M0	驅動 X 軸		M6	啟動	
M1	驅動 Y 軸		M7	停止	
M3	控制 X 軸反轉		M8	手動模式	
M5	控制 Y 軸反轉		M22	定位完成	

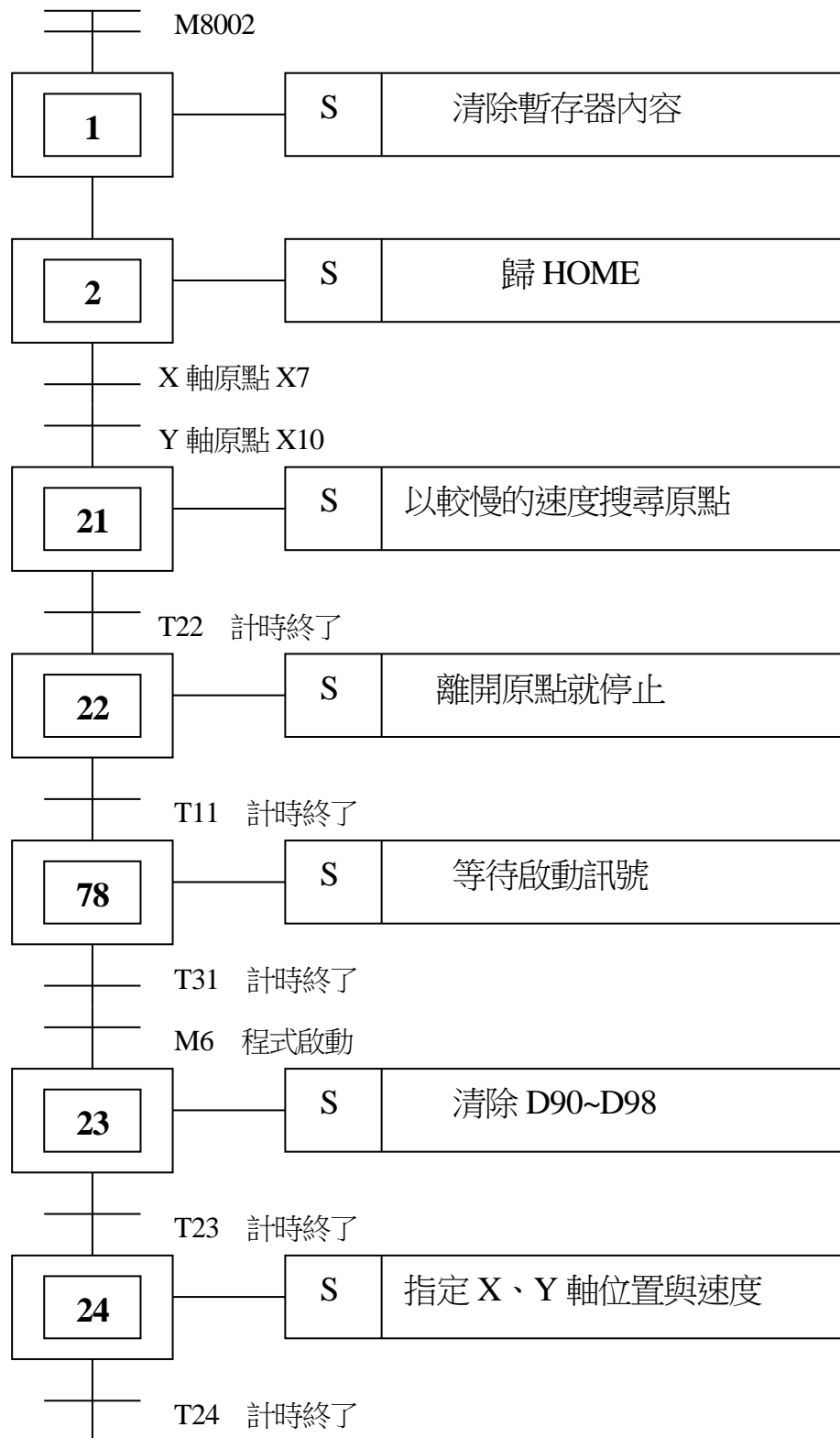
表(2.4) PLC 之程序說明

S 機 能代 號	功 能 說 明
S0	清除D90~D98 (速度, 距離, 位置暫存器) 的值 清除D10 (定位暫存器) 的值 清除M24~M27 的值 清除 Z (變數) 解除M6 保持S20
S20	指定回原點速度及距離, 讓X, Y往原點方向移動 碰到原點近接時就停止移動並解除M30 保持S21
S21	指定X軸移動速度, 並移動2500 (PULSE) 指定Y軸移動速度, 並移動5000 (PULSE) 以較慢的速度往正方向移動 保持S22
S22	XY往方向移動, 碰到原點近接後立即停止 (此為原點) 保持S78
S78	需按啟動鈕(M6) , 才可繼續動作 保持S23
S23	當沒有驅動X, Y時, 清除D90-D98 的值 保持S24
S24	指定X, Y軸移動速率及距離 保持S25
S25	驅動X, Y軸 觸發M7時, 馬達會停止動作 到達定位時, 保持S26
S26	當X軸定位成功, 將M0~M5 全部解除 連續執行比較D2Z與D0Z 當 $D2Z > D0Z$ 時, 把D2Z-D0Z 的值存在D50 $D2Z < D0Z$ 時, 把D0Z-D2Z 的值存在D50 並控制X軸反轉 $D2Z = D0Z$ 時, 停止X軸輸出



<b>S27</b>	<p>當Y軸定位成功，比較D22Z與D20Z</p> <p>當 <math>D22Z &gt; D20Z</math> 時，把<math>D22Z - D20Z</math>的值存在D70</p> <p><math>D22Z &lt; D20Z</math> 時，把<math>D20Z - D22Z</math>的值存在D70</p> <p>並控制Y軸反轉</p> <p><math>D22Z = D20Z</math> 時，控制Y軸輸出</p> <p>保持S28</p>
<b>S28</b>	<p>移動 " X軸相差距離暫存器 " 的值移到 " X軸距離暫存器 "</p> <p>移動 " Y軸相差距離暫存器 " 的值移到 " Y軸距離暫存器 "</p> <p>解除M6</p> <p>保持S29</p>
<b>S29</b>	<p>判斷是否執行手動模式(M8)</p> <p>是：必須按啟動鈕(M6)才可繼續下一孔加工，保持S30</p> <p>否：直接進行下一孔加工，保持S30</p>
<b>S30</b>	<p>驅動X，Y軸</p> <p>M7觸發時，馬達會停止動作</p> <p>當XY到達定位時，保持S31</p>
<b>S31</b>	<p>解除M0~M5</p> <p>將Z與K2相加 (<math>Z = Z + K2</math>)</p> <p>比較Z與K8的值，當</p> <p><math>K8 &gt; Z</math> 時，解除M6狀態，保持S26 (繼續下一點加工)</p> <p><math>K8 = Z</math> 時，清除D10的資料，保持S0並清除D10資料</p> <p>(回到原點等待加工)</p>

## 2-6 SFC 架構圖



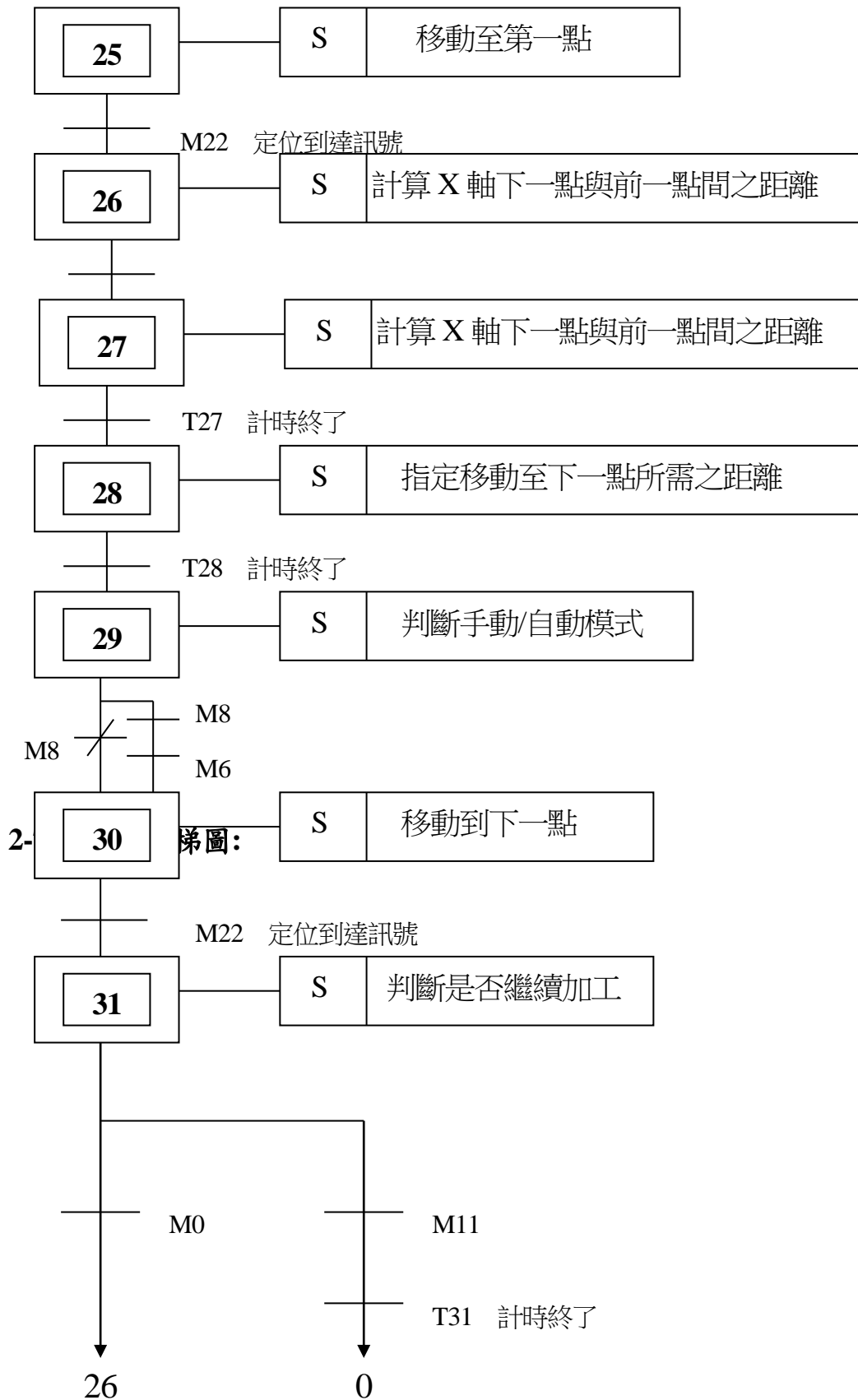
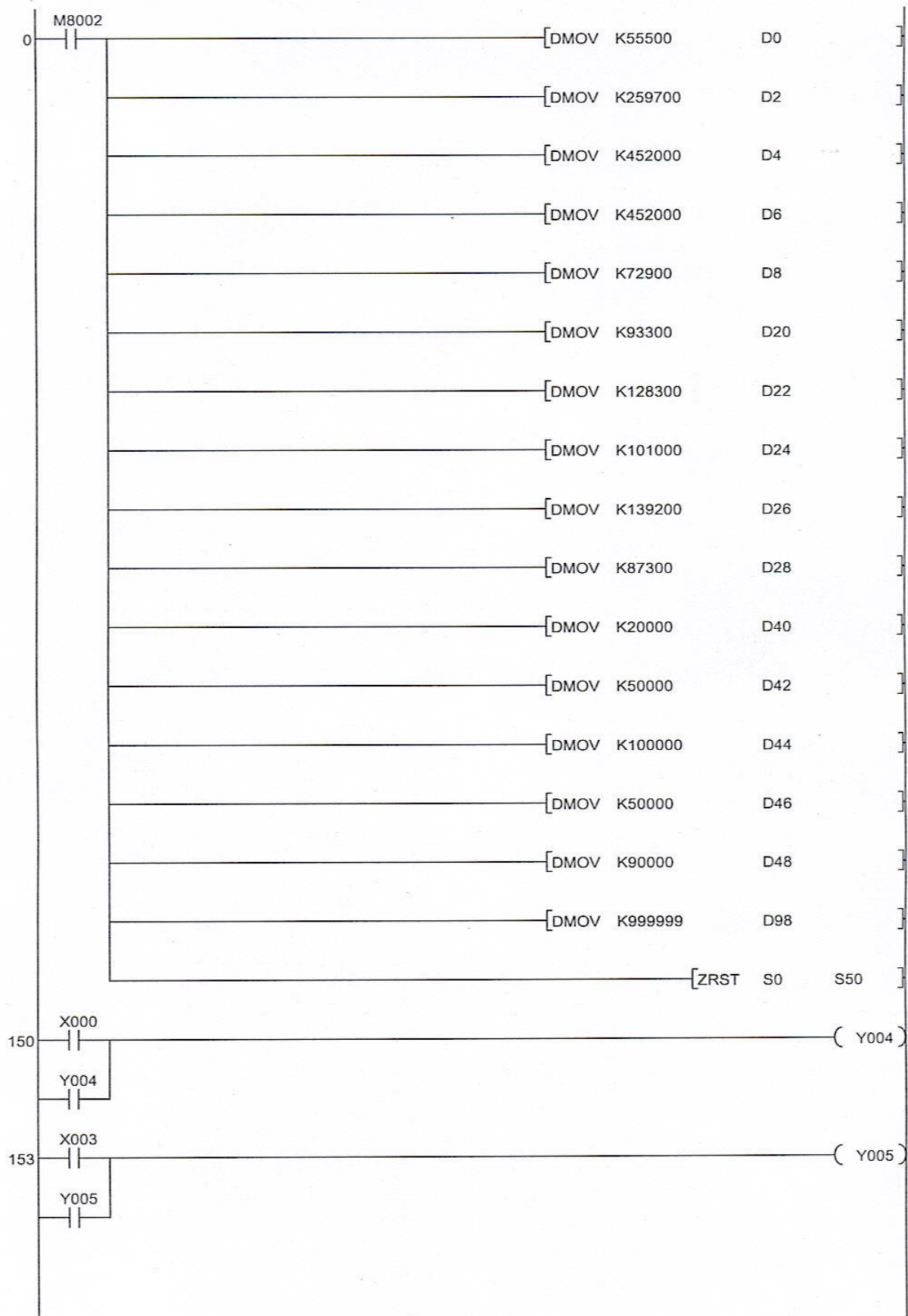
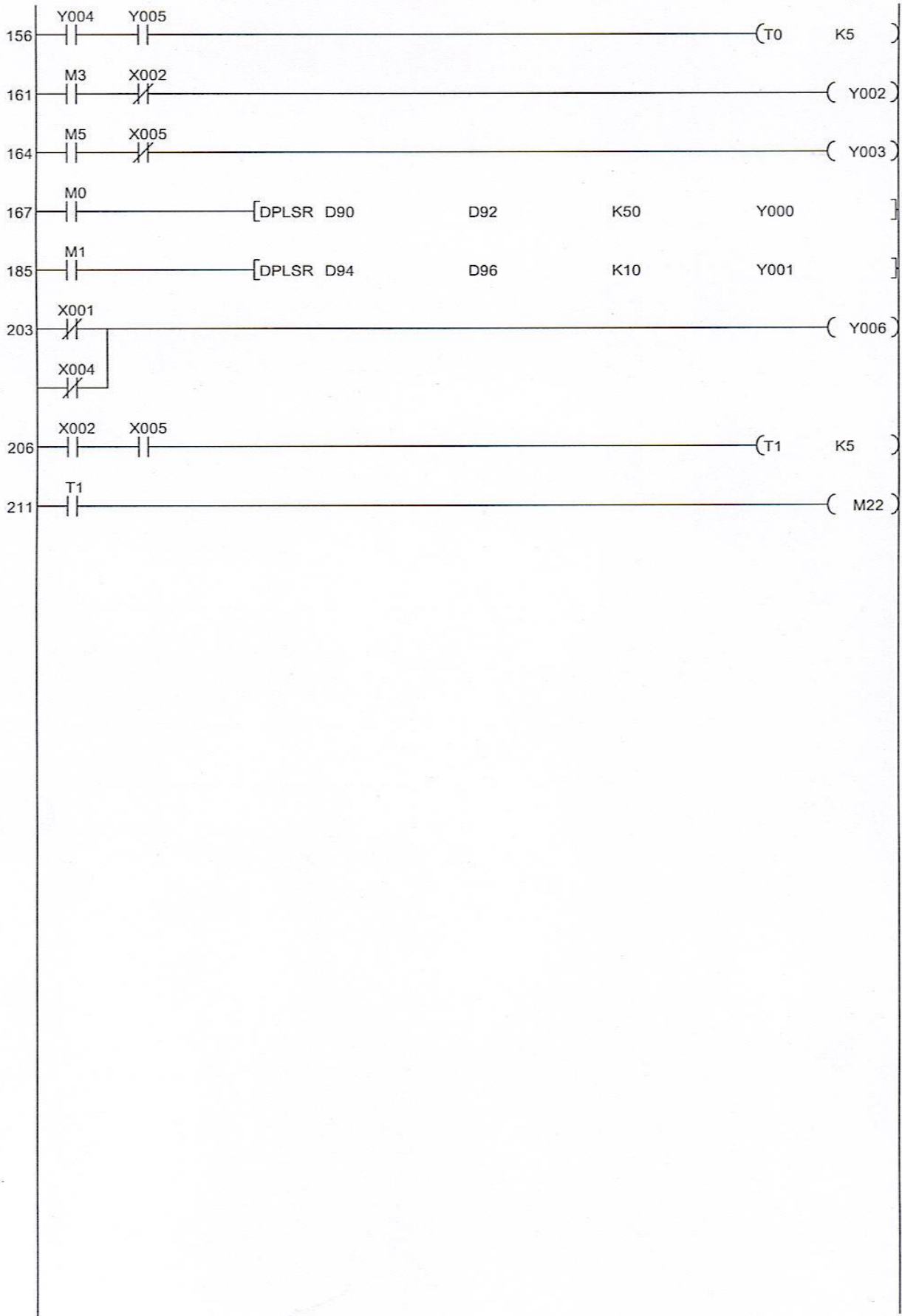
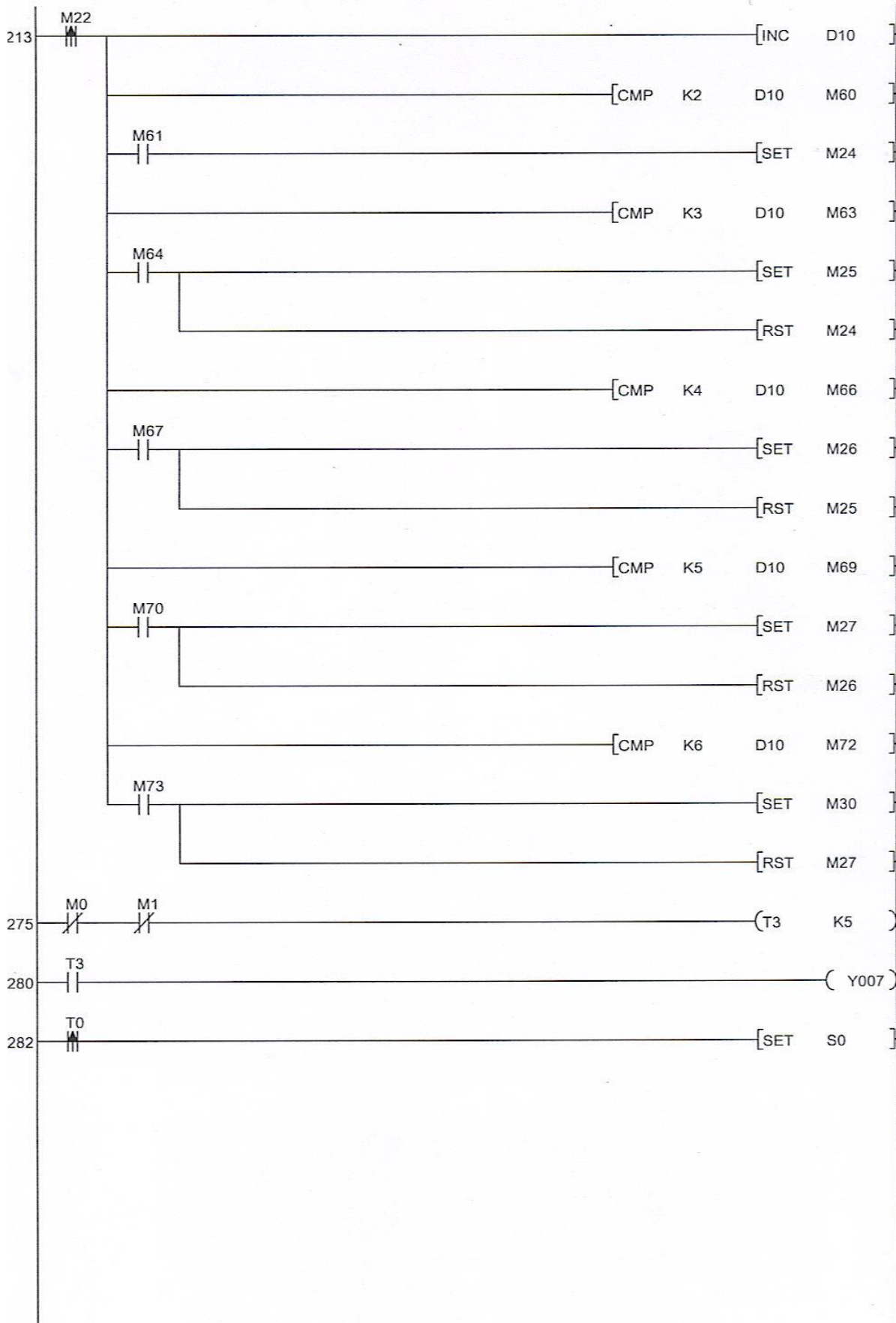


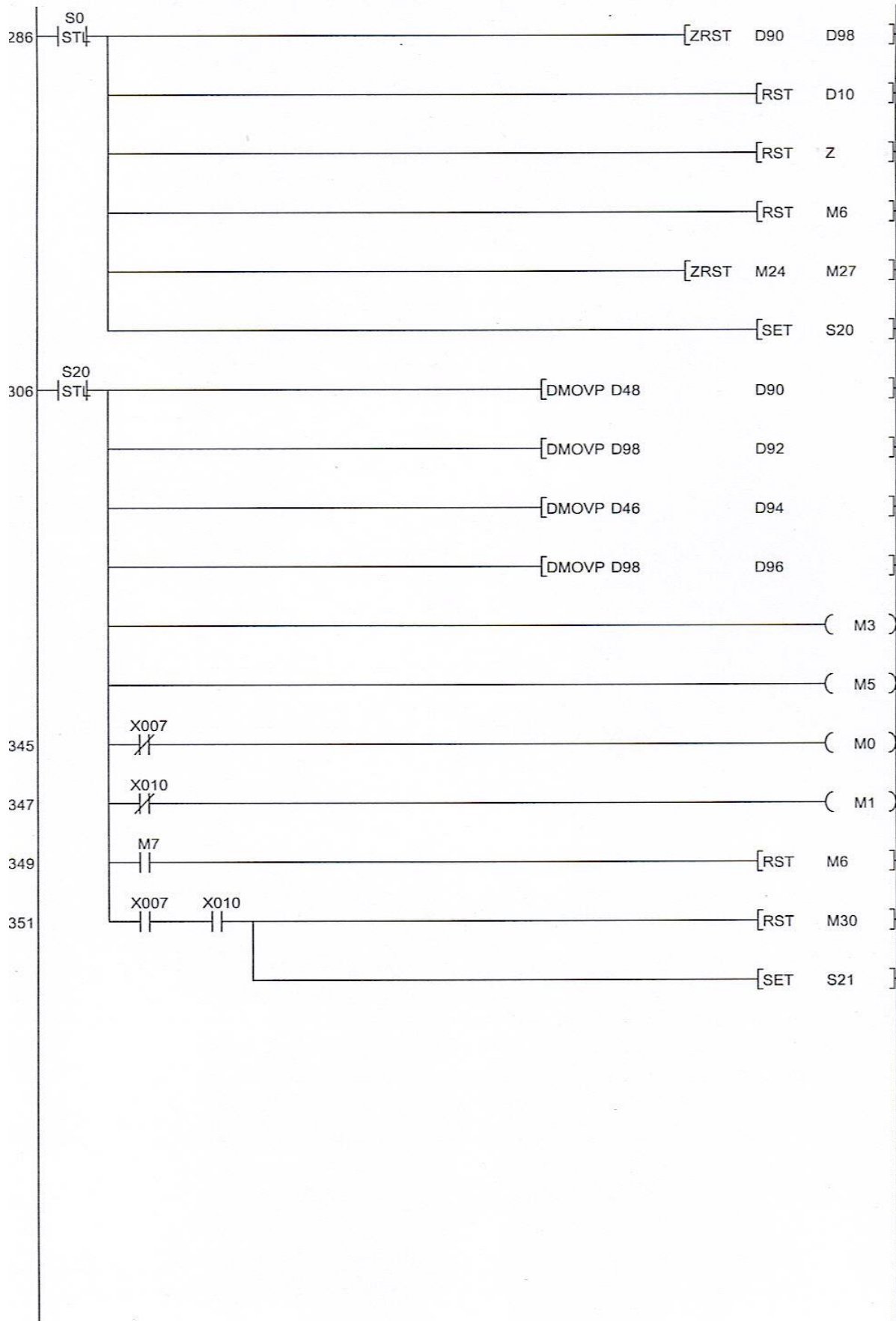
圖 2.10 SFC 架構圖

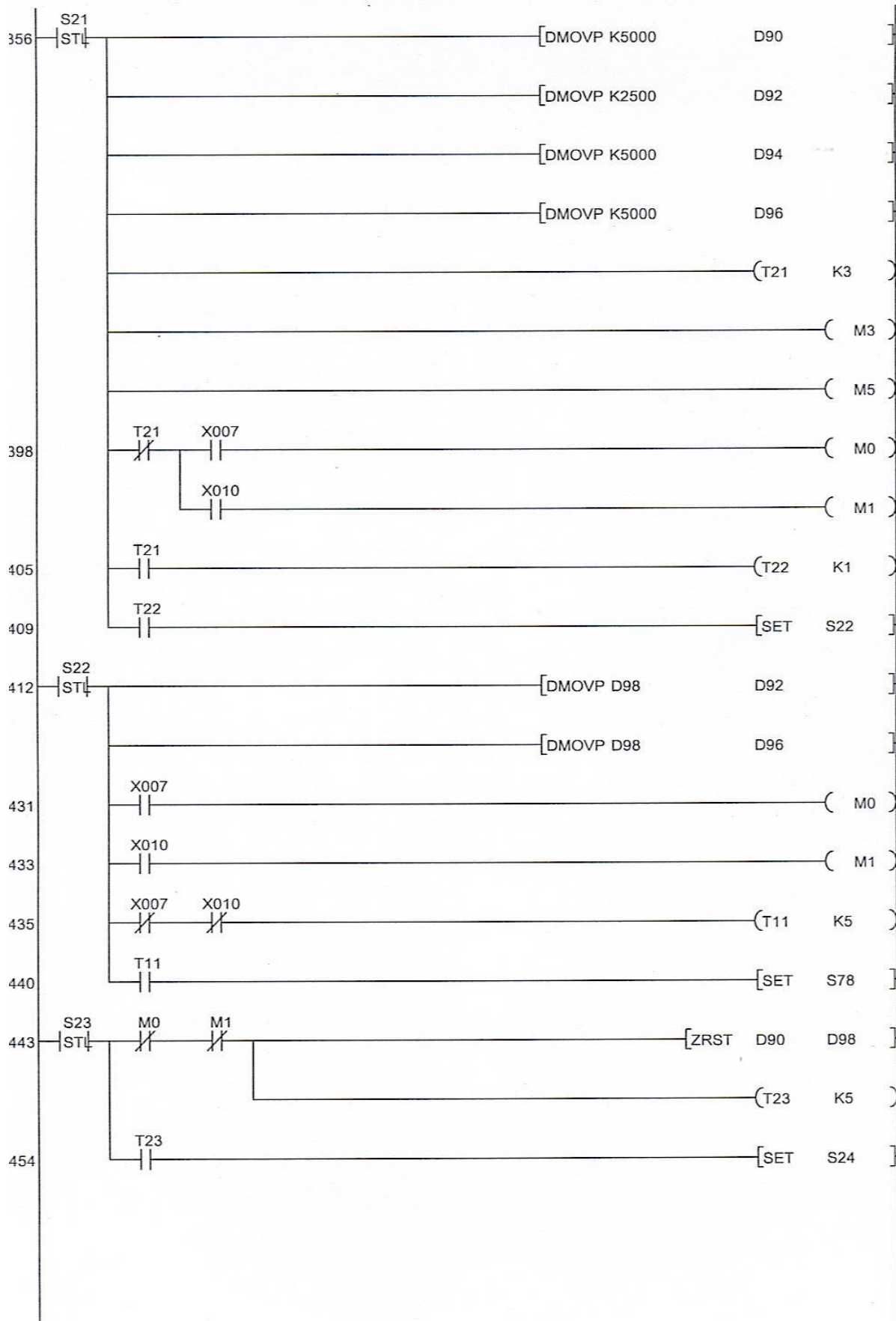
## 2-7 PLC 階梯圖



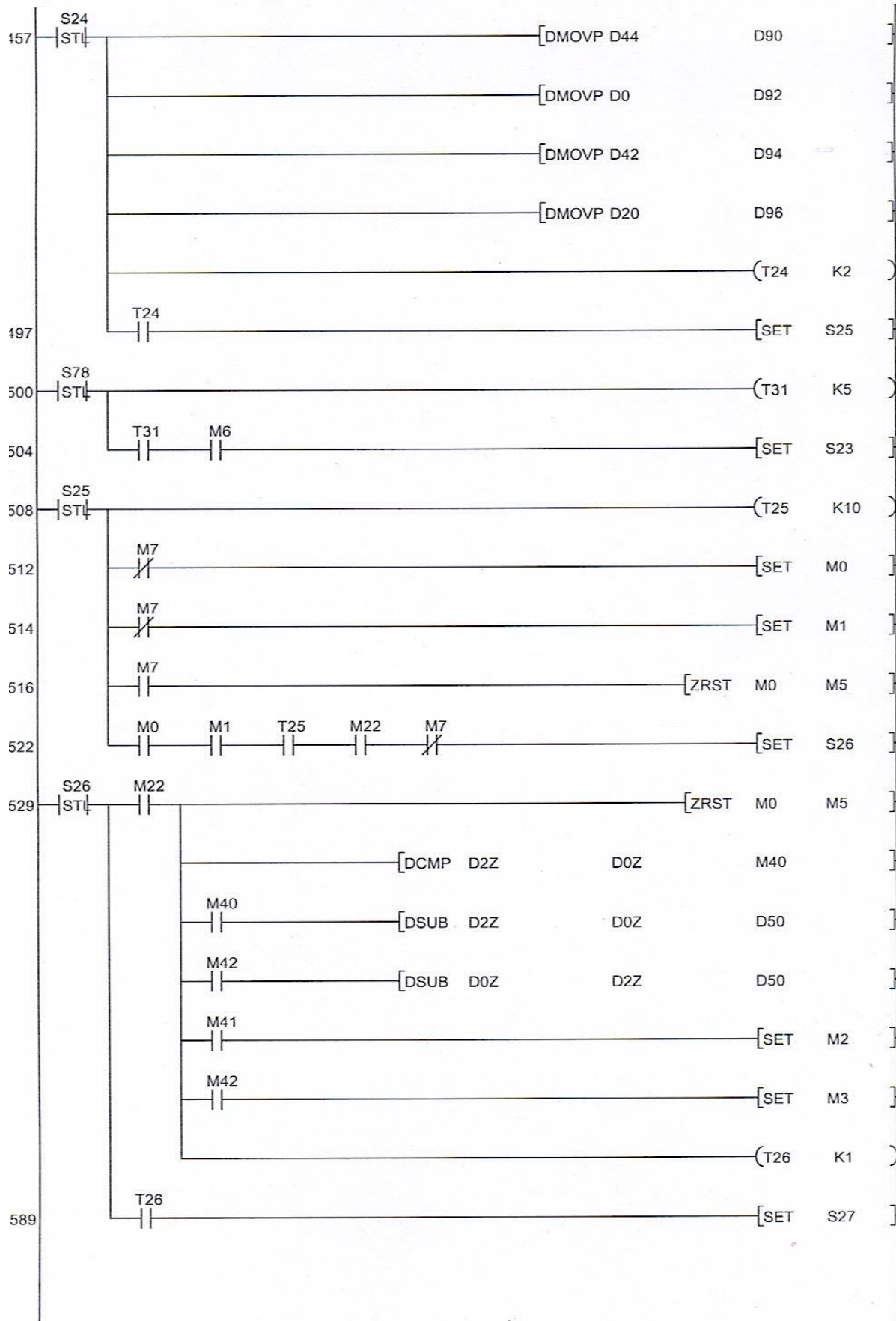


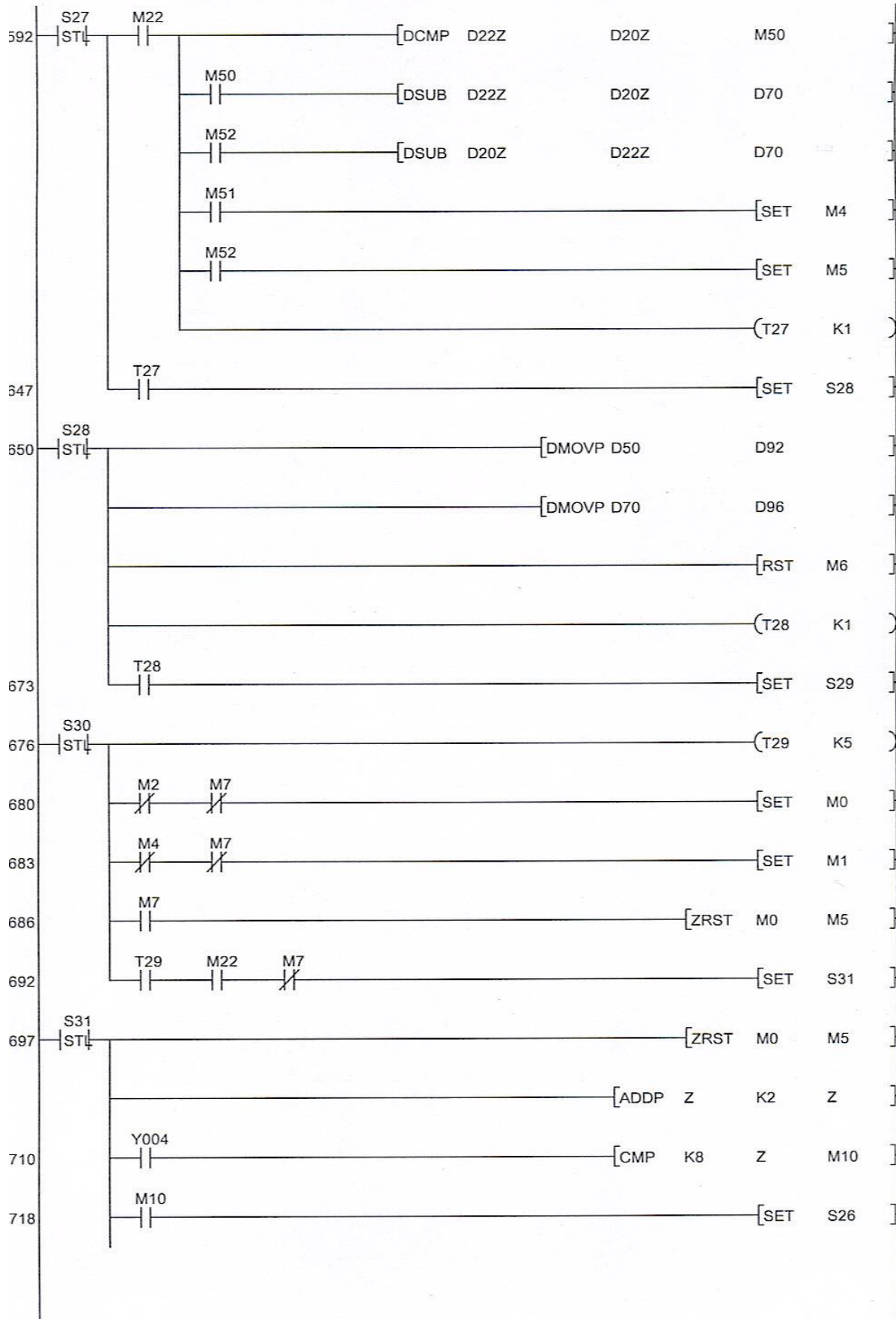


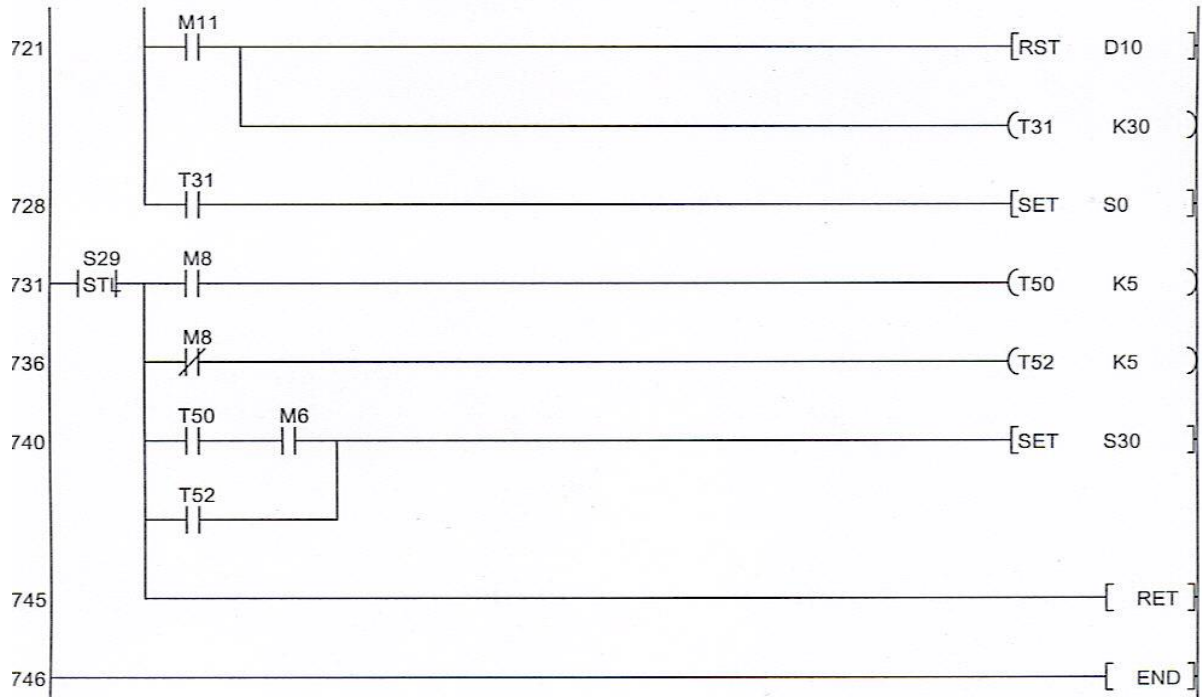












### 第三章 VB 圖控畫面簡介

#### 3-1 系統主畫面



圖 3.1 系統主畫面

本程式主畫面採用本校之徽章

程式執行時

徽章大小會隨著視窗大小變化

此畫面顯示現在時間、本製作小組介紹以及倒數計時器

倒數計時 10 秒後自動進入監控畫面

或直接按監控畫面左上方按鈕直接進入監控畫面

### 3-2 系統監控畫面

畫面簡介:



圖 3.2 系統監控畫面

本畫面為系統之監控畫面

由畫面中可清楚看到加工點的狀態

X-Y 軸正反轉狀態

X-Y 軸驅動器之啟動與否

警報燈於系統過載時會亮燈，並發出警報音

由手動模式按鈕可切換手／自動模式

開始按鈕啟動程式，可隨時以暫停按鈕來暫停機台動作

輸入欲加工件數，每加工完成一次，會於畫面下方顯示已加工／未加工件數

若未輸入欲加工件數而啟動，系統會出現訊息框提示

清除按鈕為清除文字框中之數值

於畫面中也會顯示現在日期／時間

### 3-3 程式設計內容簡介

#### (一)系統主畫面程式內容:

Dim timedelay

---

Private Sub Command1\_Click()

Form2.Show                                    按下 Command1 按鈕(進入系統監控畫面)

Form1.Hide                                    Form1(系統主畫面)關閉

timedelay = 0

End Sub

---

Private Sub Form\_Load()

timedelay = 11                                載入表單時，設定延遲時間

End Sub

---

Private Sub Form\_Resize()

Image1.Left = 0

Image1.Top = 0                                Image1(校徽圖案)左邊界及上邊界為 0

Image1.Width = Form1.ScaleWidth            下邊界及右邊界隨著表單大小改變跟著變化

Image1.Height = Form1.ScaleHeight - StatusBar1.Height

End Sub

---

Private Sub Form\_Unload(Cancel As Integer)

End    結束

End Sub

---

Private Sub Timer1\_Timer()

timedelay = timedelay - 1                    倒數 10 秒

StatusBar1.Panels(1).Text = Now              呈現現在時間

StatusBar1.Panels(2).Text = "93 年夜二技自動化二甲專題小組製"            呈現本組介紹

StatusBar1.Panels(3).Text = "進入系統畫面倒數" & timedelay & "秒"            進入監控畫面剩餘時間

If timedelay = 0 Then

Form2.Show                                    呈現系統監控畫面

Form1.Hide                                    系統主畫面關閉

End If

End Sub

## (二)系統監控畫面程式內容:

Dim w As Integer

Dim tex As Integer

設定變數 w 和 tex 為整數

---

Private Sub Form\_Load()

With MSComm1

.Settings = "9600,e,7,1"

.CommPort = 1

設定 VB 與 PLC 可程式控制器的串列控制項

.PortOpen = True

MSComm1 通訊協定

.RTSEnable = True

End With

Timer1.Interval = 150

Timer2.Interval = 50

設定計時器時間

Timer3.Interval = 50

StatusBar1.Panels(2).Text = "Visual Basic 監控畫面"

End Sub

---

Private Sub Form\_Unload(Cancel As Integer)

MSComm1.PortOpen = False

Unload Form1

載出表單，關閉 MSComm1 的通訊埠

End Sub

---

Private Sub Command1\_Click()

Timer2.Enabled = Not Timer2.Enabled

If Timer2.Enabled = True Then

Command1.Caption = "自動模式"

cmdstr = check\_sum(Chr\$(2) + "80808" + Chr\$(3))

MSComm1.Output = cmdstr 8 為強制 OFF

Else

切換手/自動模式

Command1.Caption = "手動模式"

cmdstr = check\_sum(Chr\$(2) + "70808" + Chr\$(3))

MSComm1.Output = cmdstr 7 為強制 ON

End If

End Sub

```
Private Sub Command2_Click()
```

```
    Call ClearAllControls(Me)
```

呼叫 ClearAllControls(Me)，清除文字框數值

```
End Sub
```

---

```
Private Sub Command3_Click()
```

```
    If Text1.Text = "" Then
```

```
        MsgBox "請先輸入欲加工數量"
```

啟動程式按鈕，若未輸入欲加工數量

```
    GoTo qwe
```

程式跳到 qwe

```
    End If
```

```
    If Command1.Caption = "手 動 模 式" Then
```

```
        GoTo hand
```

若選擇手動模式，程式跳至 hand

```
    End If
```

```
    w = w + 1
```

```
    Text2.Text = w
```

```
    tex = Text1.Text
```

```
    Text3.Text = tex - w
```

```
    If tex - w < 0 Then
```

```
        Text3.Text = 0
```

```
        Text2.Text = tex
```

```
        MsgBox "已經完成加工數量"
```

```
        GoTo qwe
```

```
    End If
```

計算已加工、未加工數量，  
加工完成後顯示訊息方塊

```
hand:
```

```
    cmdstr = check_sum(Chr$(2) + "70608" + Chr$(3))
```

Chr\$(2)：起始碼

```
    MSComm1.Output = cmdstr
```

"70608"：強制 M6 ON

```
    t1 = Timer: Do: Loop While (Timer - t1 < 0.15)
```

Chr\$(3)：結束碼

```
    ackstr = MSComm1.Input
```

```
    If Asc(ackstr) = 6 Then
```

延時 0.15 秒，以接收 PLC 回應訊息

```
        Shape5.BackColor = vbRed
```

```
        Shape6.BackColor = vbBlack
```

```
    End If
```

```
qwe:
```

```
End Sub
```



```

Private Sub Command4_Click()
    Timer3.Enabled = Not Timer3.Enabled
    If Timer3.Enabled = True Then
        Shape5.BackColor = vbBlack
        Shape6.BackColor = vbRed
        cmdstr = check_sum(Chr$(2) + "70708" + Chr$(3))
        MSComm1.Output = cmdstr    強制 M7 ON
    Else
        Shape5.BackColor = vbRed
        Shape6.BackColor = vbBlack
        cmdstr = check_sum(Chr$(2) + "80708" + Chr$(3))
        MSComm1.Output = cmdstr    強制 M7 OFF
    End If
End Sub

```

暫停

---

```

Private Sub Command5_Click()
    End
End Sub

```

系統結束

---

```

Private Sub Timer1_Timer()
    StatusBar1.Panels(1).Text = Now    現在時間
    '讀取馬達狀態
    cmdstr = check_sum(Chr$(2) + "000A001" + Chr$(3))    "000A001"中：0=讀取元件群
    MSComm1.Output = cmdstr    00A0=Y0~Y7
    MSComm1.InputLen = 8    01=一組
    t1 = Timer
    Do
        DoEvents    延時 0.1 秒
    Loop Until (Timer - t1 > 0.1) Or MSComm1.InBufferCount >= 8    等待 0.1 秒 or 暫存區等待的
    ackstr = MSComm1.Input    字元數為 8 時，便離開迴圈
    str1 = val_to_bin(val("&H" & Mid(ackstr, 2, 2)), 8)    自回應字串中取出所需的資料，將原來的
    For i = 0 To 8    16 進制轉換成 10 進制，再由 Function val_to_bin
        On Error Resume Next    轉成 2 進制，取字串第二個起的兩位數值
        Select Case i
            Case 0: Shape2(i).FillColor = IIf(Mid(str1, 8 - i, 1), vbRed, vbBlack)
            Case 1: Shape2(i).FillColor = IIf(Mid(str1, 8 - i, 1), vbRed, vbBlack)
            Case 2: Shape2(i).FillColor = IIf(Mid(str1, 8 - i, 1), vbRed, vbBlack)
            Case 3: Shape2(i).FillColor = IIf(Mid(str1, 8 - i, 1), vbRed, vbBlack)

```

```
Case 4: Shape2(i).FillColor = IIf(Mid(str1, 8 - i, 1), RGB(0, 255, 255), vbBlack)
Case 5: Shape2(i).FillColor = IIf(Mid(str1, 8 - i, 1), RGB(0, 255, 255), vbBlack)
Case 6: Shape2(i).FillColor = IIf(Mid(str1, 8 - i, 1), RGB(255, 0, 255), vbBlack)
```

```
End Select
```

```
Next i
```

```
If Shape2(6).FillColor = RGB(255, 0, 255) Then
```

```
OLE1.DoVerb
```

```
End If
```

#### 讀取定位點

```
cmdstr2 = check_sum(Chr$(2) + "0010301" + Chr$(3)) "0010301"讀取 M24~M31 元件群
```

```
MSComm1.Output = cmdstr2
```

```
MSComm1.InputLen = 8
```

```
t1 = Timer
```

```
Do
```

```
DoEvents
```

```
Loop Until (Timer - t1 > 0.1) Or MSComm1.InBufferCount >= 8
```

```
ackstr2 = MSComm1.Input
```

```
str2 = val_to_bin(val("&H" & Mid(ackstr2, 2, 2)), 8)
```

```
For j = 0 To 8
```

```
On Error Resume Next
```

```
Select Case j
```

```
Case 0: Shape1(j).FillColor = IIf(Mid(str2, 8 - j, 1), vbRed, vbBlack) M24
```

```
Case 1: Shape1(j).FillColor = IIf(Mid(str2, 8 - j, 1), vbRed, vbBlack) M25
```

```
Case 2: Shape1(j).FillColor = IIf(Mid(str2, 8 - j, 1), vbRed, vbBlack) M26
```

```
Case 3: Shape1(j).FillColor = IIf(Mid(str2, 8 - j, 1), vbRed, vbBlack) M27
```

```
Case 4: Shape1(j).FillColor = IIf(Mid(str2, 8 - j, 1), vbRed, vbBlack)
```

```
Case 5: Shape1(j).FillColor = IIf(Mid(str2, 8 - j, 1), vbRed, vbBlack)
```

```
Case 6: Shape1(j).FillColor = IIf(Mid(str2, 8 - j, 1), vbRed, vbBlack) M30
```

```
Case 7: Shape1(j).FillColor = IIf(Mid(str2, 8 - j, 1), vbRed, vbBlack)
```

```
End Select
```

```
Next j
```

一次讀取 8 個輸入接點，但只顯示出對應的 5 個  
控制項陣列，因此加上 Error Resume Next 可避免  
程式執行錯誤，讓程式簡化

```
If Shape2(2).FillColor = vbRed Then
    Shape2(0).FillColor = vbBlack
End If
If Shape2(3).FillColor = vbRed Then
    Shape2(1).FillColor = vbBlack
End If
End Sub
```

表達正反轉

---

```
Public Sub ClearAllControls(frmFORM As Form)
```

```
    Dim ctlControl As Object
    On Error Resume Next
    For Each ctlControl In frmFORM.Controls
        ctlControl.Text = ""
        ctlControl.ListIndex = -1
        DoEvents
    Next
```

清除 Form1 上的 Text 內容

```
End Sub
```

---

```
Public Function check_sum(str1)
```

```
    n = Len(str1)
    str2 = Mid(str1, 2, n - 1)
    Sum = 0
    For i = 1 To n - 1
        Sum = Sum + Asc(Mid(str2, i, 1))
    Next i
    sum_hex = Hex(Sum)
    check_code = Right(sum_hex, 2)
    check_sum = str1 + check_code
```

總合檢查函式

將命令碼(CMD)到結束碼(ETX)間的每一個字元  
化成對應的 16 進位 ASCII 碼，然後相加在一起  
最後取其總合值的後兩位數作為其檢查碼，附加  
在結束位元後面，一併傳送

```
End Function
```

---

```
Public Function val_to_bin(val, bit_no) As String
```

```
    i = 0
```

```
    val1 = val
```

```
    Do While ((2 ^ i) <= val): i = i + 1: Loop
```

```
    For j = 1 To i
```

```
        q = val1 \ 2
```

10 進位轉 2 進位函式

```
        r = val1 Mod 2
```

```
        val_to_bin = Trim(Str(r)) & val_to_bin
```

```
        val1 = q
```

```
    Next j
```

```
    val_to_bin = String(bit_no - i, "0") + val_to_bin
```

```
End Function
```

---

```
Sub Text1_KeyPress(KeyAscii As Integer)
```

```
    If KeyAscii < 48 Or KeyAscii > 57 Then
```

```
        KeyAscii = 0
```

防止在文字框中輸入文字

```
    End If
```

只可輸入數字

```
End Sub
```

## 第四章 結論

由於 VB 的程式並不是直覺化的設計，所以要撰寫這類程式必須克服資料傳輸上的瓶頸，也必須對資料傳輸的方式有所了解才行，畢竟電腦語言跟機器語言是有所不同的。由於 PLC 的掃描速度比電腦快，因此要成功的接收到 PLC 的訊號就必須讓 PLC 的作動時間延長。如果 PLC 元件作動時間過短，PC 就會錯過這個訊號。如此一來，整個的執行效率會因此而降低，也就是說 PC 畫面的顯示會比 PLC 的執行還慢。

換另一個角度看，因為 VB 的語言較容易理解，所以撰寫起來也比較容易，且 VB6.0 以後採用 GUI 介面(圖形化使用者介面)，也讓程式撰寫更加容易，可發揮的空間相當大。

PLC 方面，我們採用 FX1N-24MT 可程式控制器，直接與國際牌驅動器作開迴路的控制，其中並無使用任何定位模組，大大降低了設計的成本，對於業界來講，最需要這種產品了。撰寫程式的過程中，必須清楚知道解析度的算法，以及伺服驅動器的相關接點，這樣才能使定位更準確。再來就是流程的設計，要怎樣才能使階梯圖不那麼複雜，程序該如何縮到最短並符合動作要求，因此採用了迴圈的方式下去做運算，且做完一個迴圈後歸回原點，省掉了累積誤差的麻煩；為了擴充性的問題，我們也在程式裡做了一些處理，即使插入外加的步驟，也不會遺失定位的位置。往後希望能夠由 VB 的畫面直接輸入定位點的位置來做定位，並以圖形移動的方式將值傳送到 PLC 的暫存器，相信這樣更能符合業界的需求。

此專題，我們使用可程式控制器 (PLC) 及 Visual Basic (VB) 互相配合，來進行鑽孔機台多孔定位之遠端監控的目地。在過程中，我們運用個人電腦撰寫 VB 物件導向程式，設計出適合的人機介面來做圖形監控，再利用電腦的 RS-232 通訊埠與 PLC 達成資料傳輸，使 PC 監控 PLC 而 PLC 控制伺服馬達，成功的控制機台的作動，也成功的接收到 PLC 所送出的訊號，讓鑽孔機台做精密而準確的定位。

## 參考文獻

### 參考書籍:

1. 博士群工作室編著/黃明達主編，『Visual Basic 6.0 範例與解析』，全華出版社，(民 91)
2. 廖文輝/周志宏編著，『圖形監控』，全華出版社，(民 86)
3. 郭昌榮編著，『FX 系列 PLC 之連線，通訊及 VB 圖形監控』，全華出版社，(民 92)
4. 宓哲民編著，『PLC\_VB 圖形監控』，全華出版社，(民 91)
5. 黃世陽/吳明哲主編，『Visual Basic 中文版學習範本』，松崗電腦圖書資料股份有限公司，(1998)
6. 吳目誠/吳秉柔編著，『精彩 Visual Basic 6.0 程式設計』，知城數位科技股份有限公司，(2002)
7. 盧毅編著，『Visual Basic 6 中文版程式設計學習寶典』，碁峰資訊股份有限公司，(1999)
8. 陳義分編著，『Power Point 2002 新世紀應用』，全華科技圖書股份有限公司，(民 90)
9. 榮欽科技主筆室編著，『最新 Power Point 2002 彩色書』，松崗電腦圖書資料股份有限公司，(2001)

### 參考網站:

1. <http://www.8051.com.tw/guestbook.php?lang=1&bydate=1>
2. <http://www.two-way.com.tw/vbb/index.php>
3. <http://www.gpp-w.com/>
4. <http://infoserv.com.tw/phorum/list.php?f=1>
5. <http://myweb.hinet.net/home2/jaric/vb/index4m.htm>
6. <http://www.vbguide.com.tw/Default.asp>
7. <http://come.to/ippan>
8. <http://www.hosp.ncku.edu.tw/~cww/oldguy/oldguy.htm>
9. <http://myweb.hinet.net/home2/jaric/vb/indexNew.htm>
10. <http://www.hosp.ncku.edu.tw/~cww/html/site.html>
11. <http://www.hosp.ncku.edu.tw/~cww/html/vb.html>
12. <http://home.educities.edu.tw/vbtester/indexw1024.htm>
13. <http://vb.infoserv.com.tw/>

## **附錄 A 可程式控制器性能規格**

本專題之PLC控制器採用FX1N-24MT型控制器，下表(1.5)為此控制器之功能簡介：

表(A.1)可程式控制器功能簡介

項目		FX1N 系列	
演算控制方式		往覆來回掃描(專用 LSI)	
I/O 處理方式		結束再生方式	
演算處理速度	基本命令	0.55~0.7 $\mu$ s	
	應用命令	數 10~數 100 $\mu$ s	
程式語言		階梯圖+步進階梯圖(SFC 表現可)	
記憶容量・記憶體種類		8000 STEP EEPROM 內藏(可裝置 EEPROM、EEPROM 記憶卡匣)	
指令數命令的種類	基本・步進階梯圖指令	基本指令 27 個 步進階梯圖指令 2 個	
	應用指令	89 種 177 個	
輸入點		84 點 X10~X127	合計 128 點
輸出點		64 點 Y0~Y77	
輔助繼電器	一般用	M0~M383 384 點	
	EEPROM 保持	M384~M511 128 點	
	停電保持	M512~M1535 1024 點	
	特殊用	M8000~M8255 256 點	
步進點	初始狀態用	S0~S9 10 點	
	EEPROM 保持	S10~S127 118 點	
	停電保持	S128~S999 872 點	
計時器	100ms	T0~T199 200 點	
	10ms	T200 ~ T245 46 點	
	1ms(停電保持)	T246 ~ T249 4 點	
	100ms(停電保持)	T250 ~ T255 6 點	
類比旋鈕		VR1:D8030 VR2:D8031 2 點	
計數器	16 位元加算型	C0~C15 16 點	
	16 位元加算型(EEPROM 保持)	C16~C31 16 點	
	16 位元加算型(停電保持)	C32~C199 168 點	
	32 位元加算/減算型	C200~C219 20 點	
	32 位元加算/減算型(停電保持)	C220~C234 15 點	
	高速用計數器(EEPROM 保持)	C235~C255	1 相 60KHz / 2 點 10KHz/1 點 2 相 30KHz / 1 點 5KHz/1 點
資料暫存器	16 位元(一般用)	D0~D127 128 點	
	16 位元(EEPROM 保持)	D128~D255 128 點	
	16 位元(停電保持)	D256~D7999 7744 點	
	檔案暫存器(EEPROM 保持)	D1000~D7999 最多 7000 點	
	特殊用	D8000~D8255 256 點	
	間接指定用	V0~V7 Z0~Z7 16 點	



指標	JUMP, CALL 用	P0~P127 128 點
	插斷用	I0 □ □ ~ I5 □ □ 6 點
主控點		N0~N7 8 點
定數	10 進位數 K	16 位元：-32,768 ~ +32,767 32 位元：-2,147,483,648 ~ +2,147,483,647
	16 進位數 H	16 位元：0000~FFFF(H) 32 位元：00000000~FFFFFFFF(H)

## 附錄 B Visual Basic 元件介紹

 (Command Button) 按鈕 供建立一

方盒，當使用者按此功能鈕時，將可以執行寫在功能鈕的程式碼。CommandButton 控制項可以開始、中斷或者結束一個動作。利用「屬性視窗」，可以修改按鈕的名稱、字形大小、顏色....等。將游標移到按鈕上雙擊，就可叫出程式視窗，按鈕最常使用的程序是「Click」。

 (Frame) 框架


建立框架，使應用程式有條理。

Frame 控制項為控制項提供可識別的分組。


Frame 控制項用來提供其它控制項的識別群組之用。例如，您可以使用 Frame 控制項來子劃分表單功能 — 劃分為選項按鈕控制項的個別群組。

 (Label) 標籤 用於顯示字串 Label 控制

項可以用來顯示使用者無法直接改變的文字。文字標籤的用途是將一些說明文字（無法利用輸入或輸出方式變更）放在表單上。Label 控制項用來顯示文字，但使用者無法編輯這些控制項。可用程式碼來改變 Label 控制項中顯示的文字，以回應執行階段的事件。如果希望 Label 顯示的長度可以改變，就要利用 AutoSize 和 WordWrap 屬性。

 (text box) 文書盒

供使用者輸入和顯示資料，文字方塊多是用來作輸入與輸出之用。


 (timer) 定時裝置

可用於在指定的時間內做某樣工作。

Timer 控制項可以透過 Timer 事件，有規律地每隔一段時間執行一次程式碼。用於背景處理中，它是非視覺化的（程式執行時，不會在表單中呈現控制項圖案）。

 (image) 影像工具

用於顯示圖片（picture）或圖示（icon）或是圖像（metafile）。這個工具具有和功能鈕相同的功能。

 (shape) 外型工具

可以利用這工具在表格內建立矩形、圓形或多邊形。

## 附錄 C FX2可程式控制器下達命令的種類

- ◆ 元件群讀取：命令碼為『0』，適用的元件為X、Y、T、C、M、S接點及T、C、D的目前值，也就是以元件群的方式讀取上述元件內容。
- ◆ 元件群寫入：命令碼為『1』，適用的元件為Y、T、C、M、S接點及T、C、D的目前值，也就是以元件群的方式將設定值寫入上述元件。
- ◆ 元件強制ON：命令碼為『7』，適用元件為Y、T、C、M及S的單一接點。
- ◆ 元件強制OFF：命令碼為『8』，適用元件為Y、T、C、M及S的單一接點。

表(C.1) 元件名稱與位址對照表

1. 元件名稱與編號	2. 元件位址
3. Y0 ~ Y7	4. 00A0
5. M24 ~ M31	6. 0103
7. M0 ~ M15	8. 0800 ~ 080F

## 附錄 D VB 與 PLC 間的通訊—MSComm 控制項



(MSComm) 串列埠通訊工具

要以 Visual Basic 設計程式對 PLC 進行監控，需使用串列埠通訊控制項 MSComm 控制項。MSComm 控制項為 VB 專業版中的控制項，在 VB 的工作環境下建立新專案，並直接選擇專業版控制項，MSComm 控制項就會出現在工具箱中。

下表中為 MSComm 常用屬性：

表(D.1) 串列埠通訊工具屬性說明

屬性	說明
<b>CommPort</b>	設定通訊連接埠的代號。CommPort=1 即表示設定 COM1 為通訊埠。
<b>Settings</b>	設定通訊埠的傳輸速率。預設值為“9600, n, 8, 1”，意思為以每秒 9600bit 的傳輸數率，沒有同位元檢查，每個資料 8 位元，停止位元為 1 個 bit。
<b>PortOpen</b>	設定通訊埠的開啟或關閉。在傳輸之前須開啟通訊埠，而在通訊完畢後須將通訊埠關閉。
<b>Input</b>	從輸入暫存區讀取字串，並將暫存區裡的內容清除。為一種先進先出 (First In First Out FIFO) 的機制。
<b>Output</b>	將資料寫入輸出暫存器。
<b>InputLen</b>	設定由輸入暫存區讀入的字串長度。
<b>InBufferCount</b>	傳回輸入暫存區已接收並在等待中的字元數。
<b>RTSEnable</b>	發送請求致能。一般由電腦傳送 Request To Send (RTS) 信號到數據通訊設備，以請示准許傳送資料。

利用 Visual Basic 設計程式對 PLC 進行 RS-232 串列通訊埠的資料傳輸及接收，其步驟如下：

1. 於表單中加入通訊控制項——MSComm 控制項。
2. 設定通訊埠連接號碼，即設定 Commport 屬性。
3. 設定傳輸參數，即 Settings 屬性。
4. 開啟通訊埠，即令屬性 PortOpen=True。
5. 發送請求致能，即令屬性 RTSEnable=True。
6. 使用 Input 屬性讀取輸入暫存區的字串，或利用 Output 屬性送出字串到輸入暫存區。
7. 根據回應字串的內容的到傳輸的狀況，並以此作為後續處理的依據，同時也可在電腦中顯示出相關訊息。
8. 7 程式結束後須將 MSComm 控制項關閉，即令屬性 PortOpen=False。

## 附錄 E PLC 程式特殊指令說明

表 E.1 PLC 指令說明

符號，稱呼	機能	類型
SET	動作保持	
RST	動作保持解除 現在值和暫存器清除	
STL	步進階梯圖開始	
RET	步進階梯圖中了	
END	程式終了	
DPLSR	付加減速的脈波輸出	32 位元命令，連續執行型
DMOVE	轉送	32 位元命令，連續執行型
DCMP	比較	32 位元命令，連續執行型
CMP	比較	16 位元命令，連續執行型
ZRST	全部重置	16 位元命令，連續執行型
INC	BIN 增加	16 位元命令，連續執行型
DSUB	BIN 減算	32 位元命令，連續執行型
ADDP	BIN 加算	16 位元命令，脈波執行型

