

立體停車場暨動畫監控管理系統 之研究與模擬

林 基 源

摘 要

爲了解決停車問題，本系統利用電腦倉儲觀念，設計出一套既快速又安全的“立體停車場模擬系統”。於本系統中實際製作了自動立體停車場模型，並藉由PC來做管理、控制及實際運作之動態畫面安全監視。停車時並以X軸、Y軸兩軸同時傳送方式，自動的找尋最短路徑之停車位。經實作模擬證明本系統是可行的，相當適合國情需要。

STUDY AND SIMULATION ON CUBIC PARKING BUILDING

Chi-Yuan Lin

Department of Electronic Engineering
Chin-Yi Institute of Technology and Business
Taichug, Taiwan 41111, R.O.C.

Key Words: dynamic screen monitor, management system, computer control automatically.

ABSTRACT

In order to solve the problem of parking lot, the system takes advantage of the idea that computers can help control the storage of the car and designs a set of "Cubic parking lot demonstrative system" which is fast and safe. In the system, we make an automatically cubic parking lot model and by the control of

personal computer, the system can manage, control and keep an eye on the present moving picture on the screen. If someone is going to park his car, the system will find a short cut for him automatically by means of the transmission of both X axis and Y axis.

Through the practical experiment, the system is proved to be usable and meet our people's need.

一、前 言

現今的工商業突飛猛進，相對的人民生活水準也隨之提高，而汽車也成為人們常用的交通工具，雖然汽車給我們帶來很大的便利，但是也留下不少問題，譬如：環保上的空氣污染、噪音的干擾、交通的擁擠、停車位……等種種問題。尤其是停車所產生的問題更是不勝枚舉，因為台灣土地本來就小，到處都是寸土寸金，想擁有自己的停車位置，或想在市區內找一處停車位，都得花費相當的代價，造成駕駛者無論是在精神上，或是物質上都是一個很大的負擔。

究竟有那些問題，以及本文製作的動機 [1]，和解決之道為何？簡要說明如下：

1. 停車問題：

- (1) 停車空間的浪費：停車空間固然需賴政府有計劃的規劃，但是如何有效的利用停車空間更是重要。臺灣土地有限，車子密度的成長率又急速在增加，隨著而來的問題也就產生了，那便是停車位置的缺乏。
- (2) 浪費時間及車子的損毀：工商業發達的今日所講究的是“時間就是金錢”，因為停車位置的缺乏，造成大家於停車時必須花時間找一個停車位，且泊車時往往短時間可以完成的停車，卻因泊車技術的不純熟及經驗不足，造成時間上的浪費。除此之外，也會因駕駛技術不佳，注意力不集中造成汽車的碰撞，形成無謂的損失。

2. 製作動機

很多人都有一個共同的經驗，當您駕駛車子到市區時，想停下車去買東西或辦事情，有時因找不到可停車的地方，而花費相當長的時間，或事情因

此而耽擱。如果一定要找一個停車的地方，常得在市區兜了好多地方才找到一處停車位，而且停在馬路上車子既沒有保障，又得擔心汽車是否受損。如果想停在停車場上，在人多車多的臺灣，停車位實在一位難求，基於以上之原因，這就是本文研究動機，要來解決這些問題。

3. 解決之道

為了解決上述停車位的缺乏、時間的浪費以及車子的毀損等種種問題，本系統所製作的“立體停車場模擬系統”乃是藉由電腦來做管理、控制及實際運作動態畫面監視。當有車子駛入停車場定位停車台後，由電腦做資料管理、安全監視及透過介面電路來控制驅動電路以驅動三軸之步進馬達（stepping motor）將車子移至最近之空位，如圖 1 所示。其簡要說明如下：

- (1) 將平面式的停車場，改成立體式，使原本只可停一輛車的空位，變成可停多輛車位，以改善停車位不足的問題。
- (2) 當車子開到本系統中之定位停車台後，就由電腦來控制列印有關停車資料，如表 1 所示，並自動以 X 軸(Axis)、Y 軸兩軸同時傳送方式將車子置於最短路徑的停車位。而且當取車時，電腦將先自動列印取車有關資料，如表 2 所示，並自動從選擇停車位上取車，在時間上可節省很多，並避免車子無謂受損。

本文的設計可解決很多問題，只要缺乏停車空間的地方都改成本文所設計的方式，必定可改善車位不足的問題，同時車子也受到保障，不必將車子停放在馬路上而耽心受到損壞，所以證明本文的可行性是相當高的。

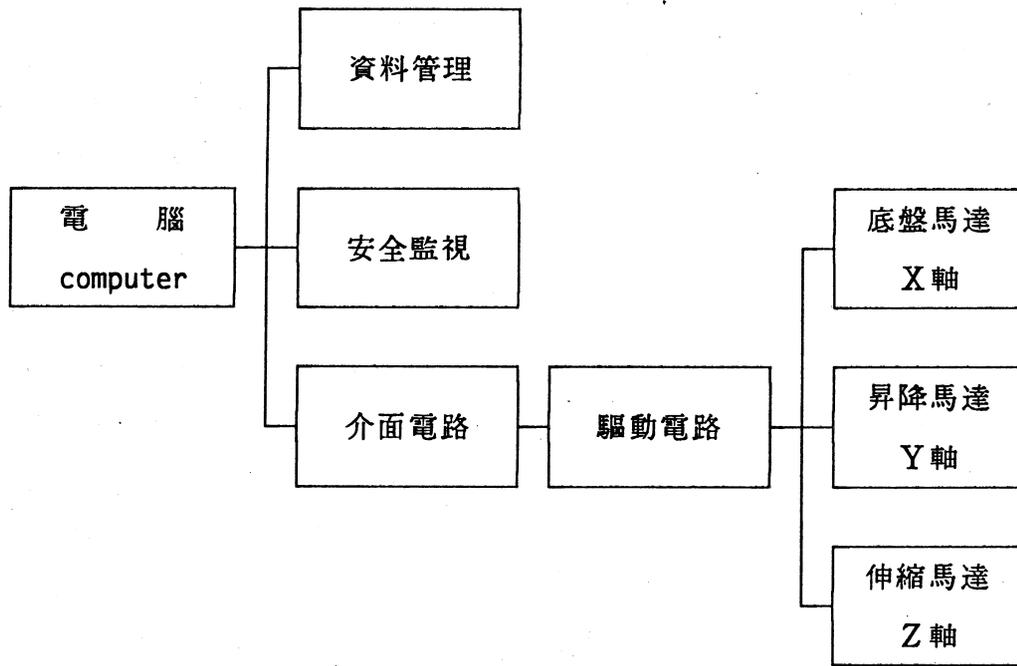


圖 1 自動立體停車場監控管理系統方塊圖

電腦自動立體停車場	
車位編號：	4
停車時間：	1998年 6月14日 20:53:15
密碼：	1234

表一 停車時列印給車主收執之單據

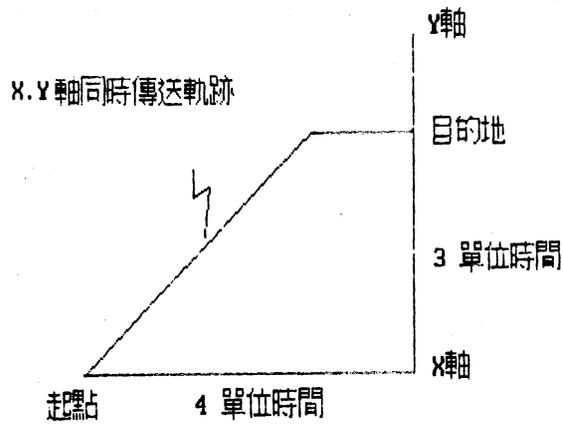
電腦自動立體停車場收據
車位編號：3
停車時間：1998年 6月14日 20:54:33
取車時間：1998年 6月14日 20:55:5
停車費用：10元整

表二 取車時列印給
車主之收據

二、機械架構設計原理

本機械架構設計原理及建立大體上是採用自動儲運系統原理[2,3]，構建成6*6 空間架構，其所佔用的停車面積僅相當於停放6 部汽車空間，卻可獲得36部停車位，所以非常節省空間。存放方式，即駕駛者把汽車駛到定位停車台位置，則昇降架構將會把汽車自動提起然後運往最短路徑的停車空位上，且爲了提高工作效率，本系統是以 X 軸，Y 軸同步轉動方式來運送，如圖 2 所示，如此可節省相當多的時間。本系統可分爲走行（X 軸）、昇降（Y 軸）與叉動（Z 軸）三動力，並以尼隆(nylon)線採用調諧器(tuner)原理，以環繞的方式達到驅動的功能。以下分三軸說明：

1. X 軸所需驅動馬力最大，因具備帶動整體之架構與汽車，所以在此選用了 3 伏特(Voltage)，2.7 安培(Ampere)之規格施以 5 伏特電壓，以求得較大馬力。
2. Y 軸所需馬力比 X 軸來得小，其功能爲帶動存取汽車之機械架構與汽車的昇與降，以達到上昇與下降之目的。再配合上 X 軸就可以活動於 36 部停車位之中。同時有一點須注意的就是，當上昇或下降到達定位時，必須剎車，否則架構將會往下滑落，其剎車原理是將控制步進馬達的 4 個位元持續送出訊號即可達成。Y 軸之步進馬達是採用 24 伏特，0.18 安培之規格。
3. Z 軸之功能是用來將找到定位的汽車，伸出放至指定的儲位或在指定儲位



X、Y軸同時傳送需4 單位時間

X、Y軸各別傳送需4+3=7 單位時間

圖 2 X、Y軸同時運轉解說圖

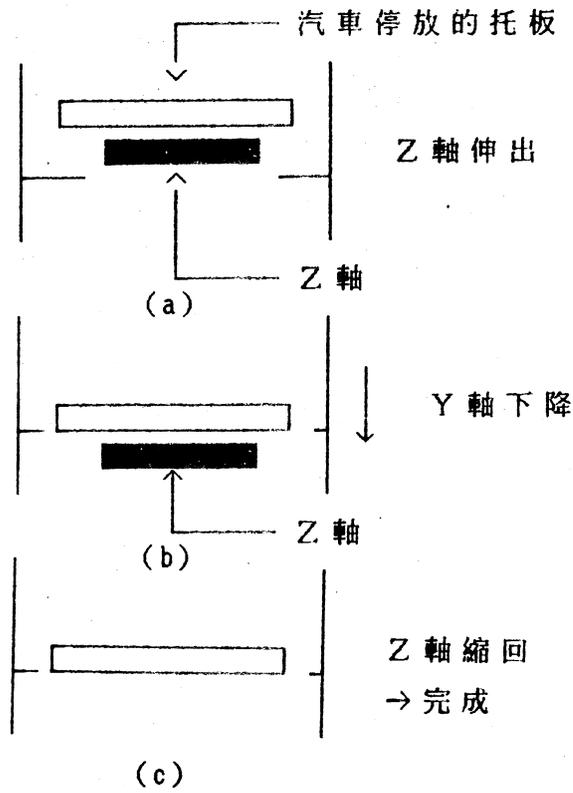


圖 3 存車動作之解說圖

取出汽車，其存與取的動作須配合 Y 軸來進行。當存車時其動作是 Z 軸先伸出到定點之後，Y 軸下降到定點接著 Z 軸又縮回到起點，如此來達到存車的功能，如圖 3 所示。而取車之動作，即為存車動作之反向動作。

4. 載車架三軸之架構上加有極限開關(limit switch)，以使存取車輛後做載車架歸位控制。如此不但可提高準確度以減少誤差或誤動作，更可發揮監控程式的效率。

三、系統電路原理和分析

1. I/O 介面卡電路原理

- (1) 本介面卡是用 74LS138 解碼器(decoder) 做為位址解碼的選擇，當要對三軸之步進馬達做控制時要先藉由 74LS138 解碼器解碼出 8255 PIO(Parallel input output) 控制步進馬達埠(port)位址。當選擇到 8255 PIO 後再將控制訊號籍 8255 PIO 送至步進馬達驅動(stopping motor driver) 電路，如圖 4 所示。
- (2) 當 74LS138 解碼器位址解碼選擇到 8255 PIO 被規劃的位址時，表示 8255 PIO 能夠用軟體做規劃，看是要選擇 PORT A 或 PORT B 或 PORT C 來做輸入或輸出[4]。其中本電路解碼結果是 PORT A 位址(address) 為 300H、PORT B 位址為 301H、PORT C 位址為 302H 與控制暫存器(control register) 位址為 303H。
- (3) 在 8255 PIO 後的 74LS244 緩衝器(buffer) 是做電流放大以及保護電路的功用，最後將控制資料送至步進馬達驅動電路。
- (4) 圖中 Out1 是 8255 PIO 的 PORT A，其規劃為輸出模式(output mode)。PA₀ - PA₃ 是控制 X 軸(底盤)的步進馬達，PA₄ - PA₇ 是控制 Y 軸(昇降臺)的步進馬達。如此利用一個 PORT 同時控制兩軸驅動步進馬達，可穩定達到 X 軸、Y 軸同步運轉功能。
- (5) 圖中 Out2 是 8255 PIO 的 PORT B，其規劃為輸入模式(input mode)。PB₀ - PB₂ 是規劃給三軸的極限開關用，PB₃ - PB₇ 未使用。
- (6) 圖中 Out2 PB₀ 接一微動開關(Micro switch)負責底盤 X 軸馬達定位感測，PB₁ 接一微動開關負責昇降 Y 軸馬達定位感測及 PB₂ 接一光遮斷器(photo interrupter)負責伸縮 Z 軸馬達定位感測[5]。

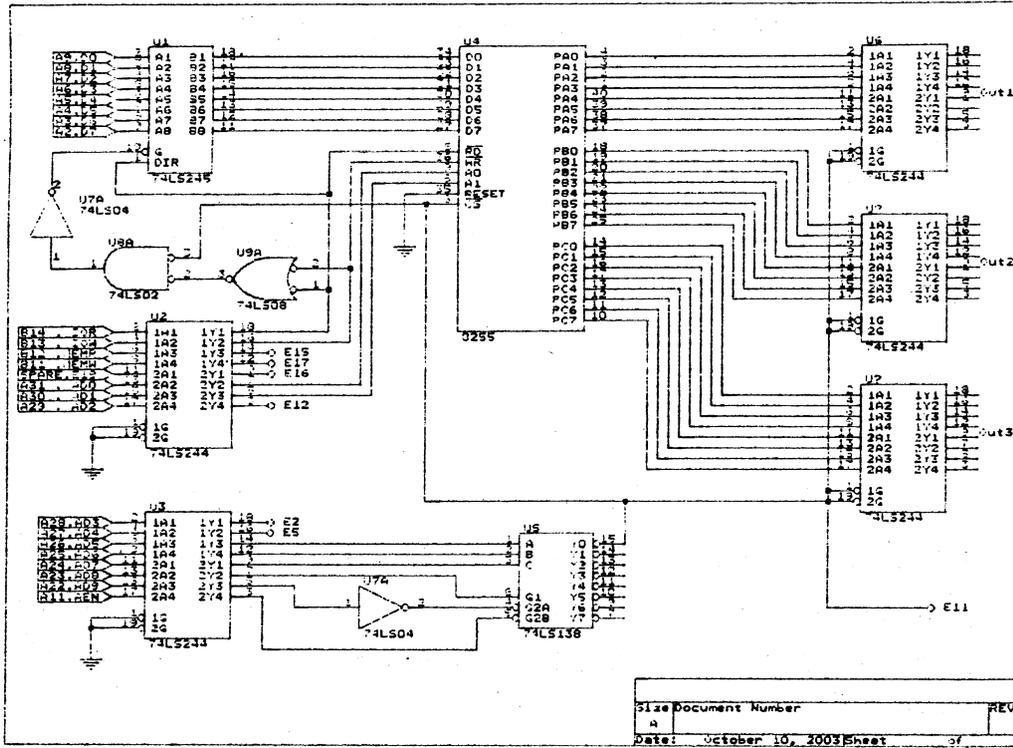


圖 4 I/O 介面卡電路圖

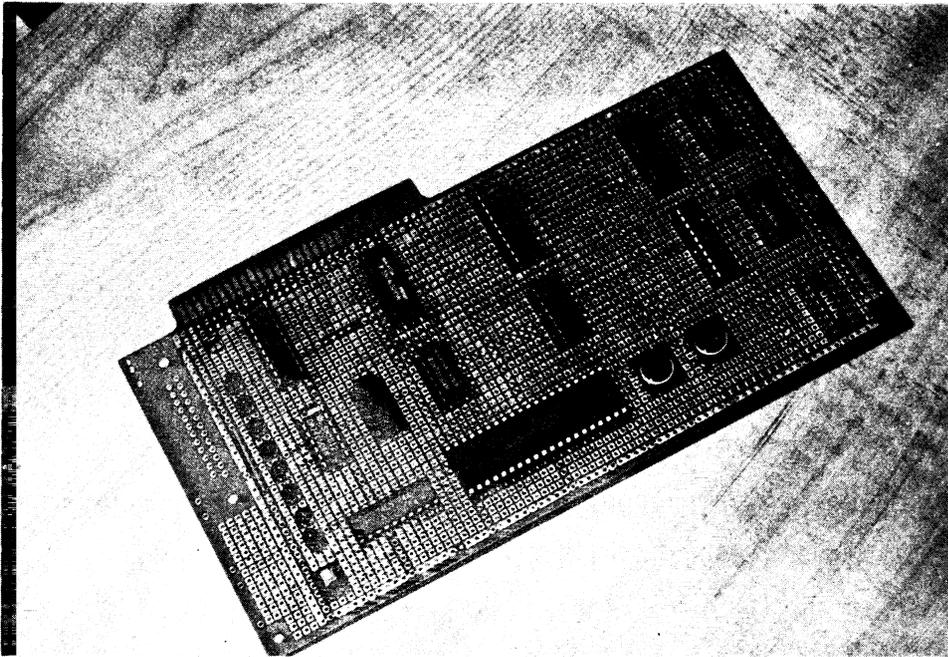


圖 5 I/O 介面電路實體圖

(7)圖中Out3是8255 PIO的PORT C，其規劃為輸出模式。

PC₀ -PC₃ 是控制 Z 軸（伸縮臺）的步進馬達，PC₄ -PC₇ 未使用。

(8)圖 5 為 I/O 介面卡電路之實體圖。

2.步進馬達驅動電路原理與Test Step 程式分析

(1)本驅動路如圖 6 所示，為能使步進馬達有較大的扭力，因此採用“2相激磁法”[6]。

(2)驅動電路中使用達靈頓電路(Darlington CKT)來做電流放大，並且加入二極體做為驅動電路的保護。

(3)整個驅動電路之動作是由8255 PIO送出信號推動由電晶體2N3569及2N3055所組成的達靈頓放大電路，使其能夠提供足夠的電流驅動步進馬達。

(4)本模擬系統使用到 X、Y、Z 軸，因此必須有三組驅動電路。

(5)利用步進馬達Test Step程式[7]，測得每一停車位之步進馬達應走步數數據如表 3 所示。

(6)圖 7 為步進馬達驅動電路實體圖。

車位編號	X 軸需走步數	Y 軸需走步數
0	290	1115
1	290	895
2	290	670
3	290	440
4	290	220
5	290	0
6	540	1115
7	540	895
8	540	670
9	540	440
10	540	220
11	540	0
12	793	1115

13	793	895
14	793	670
15	793	440
16	793	220
17	793	0
18	1045	1115
19	1045	895
20	1045	670
21	1045	440
22	1045	220
23	1045	0
24	1290	1115
25	1290	895
26	1290	670
27	1290	440
28	1290	220
29	1290	0
30	1545	1115
31	1545	895
32	1545	670
33	1545	440
34	1545	220
35	1545	0

表 3 步進馬達對各停車位應走步數實驗所得數據表

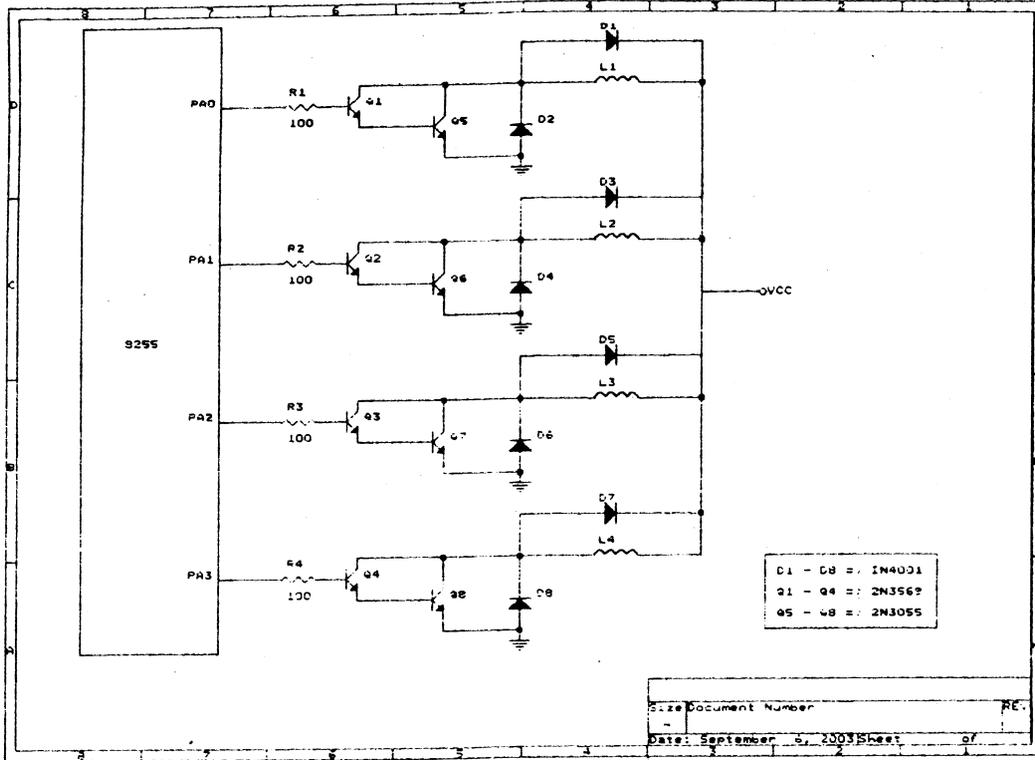


圖 6 步進馬達驅動電路圖

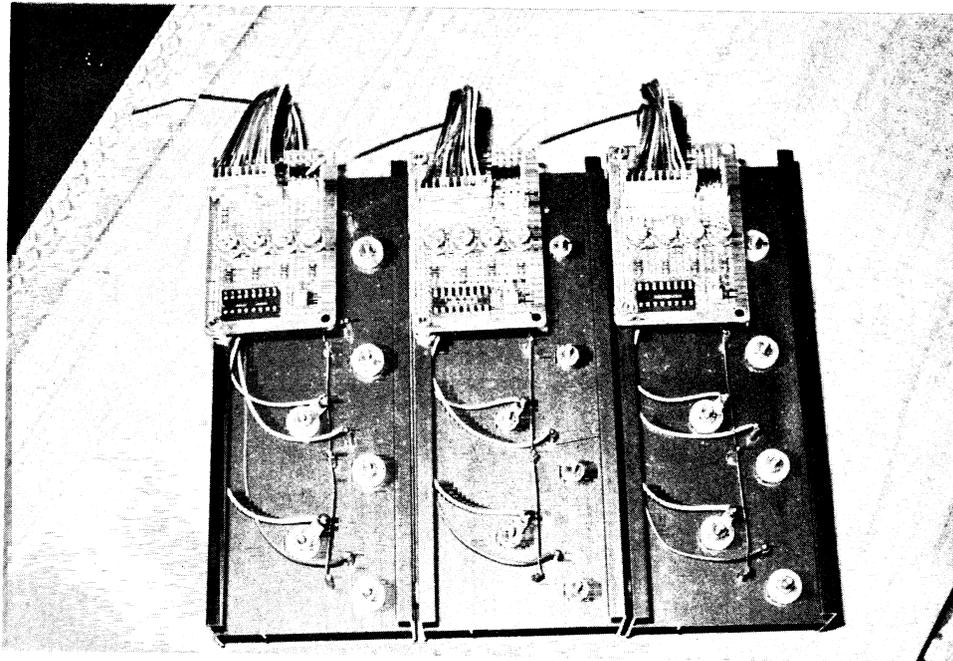


圖 7 步進馬達驅動電路實體圖

四、軟體程式分析

1. 電腦自動立體停車系統主程式各部階層(Hierarchical)之說明：

本動畫監控管理程式係以Turbo Pascal程式語言在IBM PC/AT 機器上執行[8]，功能包含有有機架結構之驅動[9]、實際運作之安全監視與管理資料之處理等。程式與註解說明請參閱附錄所示。其軟體程式之各部階層說明如圖8所示。

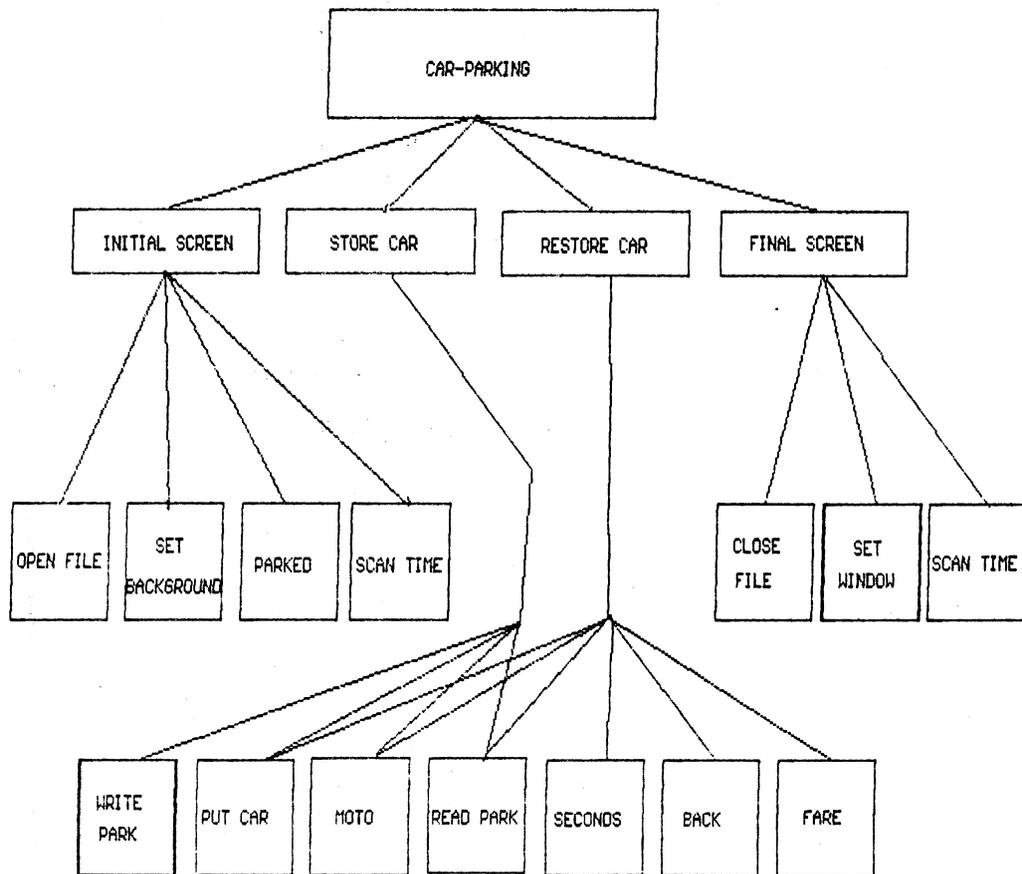


圖 8 動畫監控管理程式階層圖

階層圖各部之說明：

Car-Parking：電腦自動停車系統之主程式。

Initial Screen：規劃初始之畫面、檔案、資料。

Store Car：處理停車作業。

Restore Car：處理取車作業。

Final Screen：還原初始畫面，關閉檔案。

細部說明：

Open File：開啓停車狀況資料檔。

Set Background：設定背景畫面。

Parked：顯示目前停車狀況

Scan Time：設定倚天中文系統掃描時間[10]。

Close File：關閉資料檔。

Set Window：設定視窗。

Write Park：寫入資料至資料檔。

Put Car：放入車輛或取出車輛。

Moto：移動載車架至指定位置。

Read Park：讀取資料檔中的資料。

Seconds：計算停車時間。

Back：將載車架返回原處。

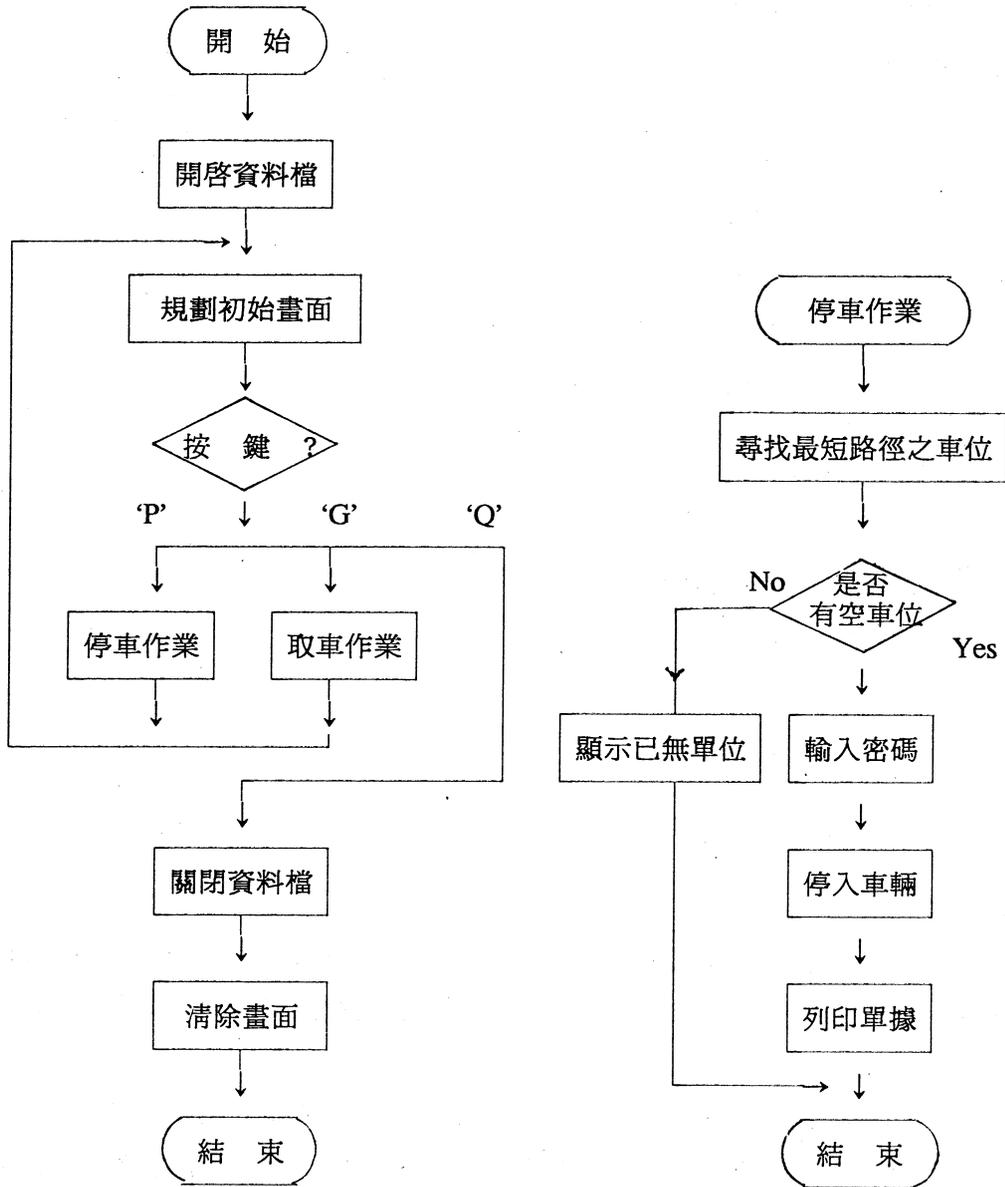
Fare：計算停車費用。

2.軟體作業流程

當起動立體停車系統後，系統進入監控畫面，並顯示告知是要選擇停放車輛(P)或取出車輛(G)或離開主程式(Q)，其整個監控管理過程如圖 9 (a) 所示。

(1)若按下P 鍵表示停放車輛，其作業流程如圖 9 (b) 所示。其中輸入密碼 (password)一項其處理方式是以字串(string)形態處理，大小可達20 byte 長度，主要是防止錯誤取車以增加寄車之安全性。列印單據之有關資料如前表 1 所示。

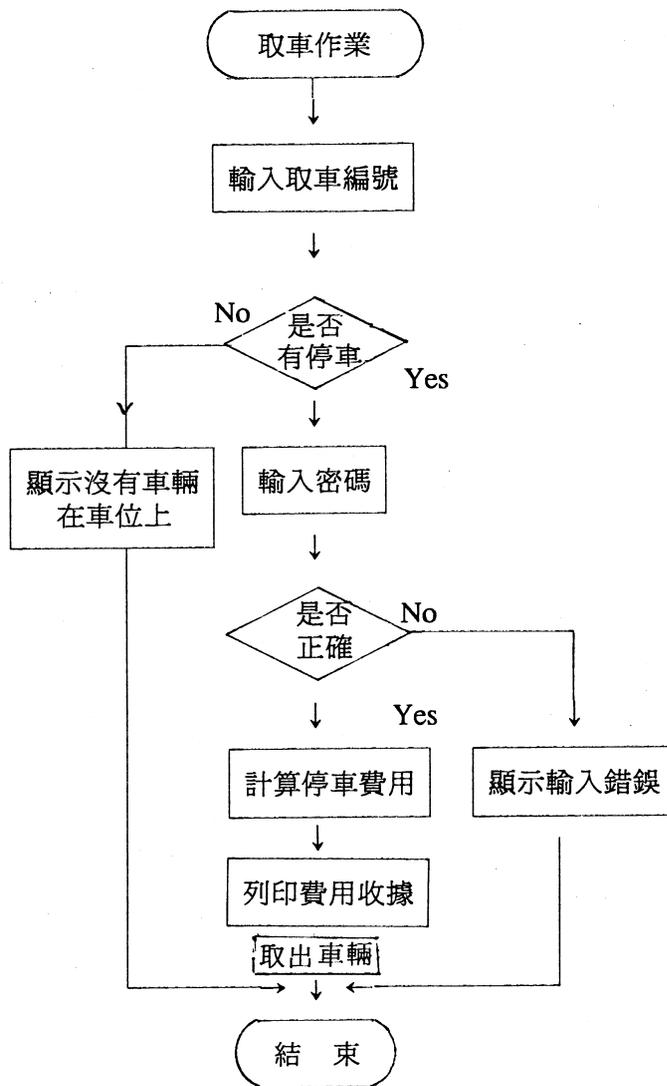
(2)若按下G 鍵表示取出車輛，其作業流程如圖 9 (c) 所示。其中輸入密碼一項若車主遺失自己原先設定的密碼，管理人員可依序按下B、O、S、S鍵取代之，以便順利取車。並且本程式對停車費用是以每小時10元計，且若停車不超過 1 小時則以基本費用10元計。其中列印費用收據之有關資料則如前表 2 所示。



(a) 主程式作業流程圖

(b) 停放車輛作業流程圖

圖 9 監控管理程式作業流程圖



(c) 取出車輛作業流程圖

圖 9 監控管理程式作業流程圖

3.資料結構

本程式所使用的資料結構有二：

(1)時間的資料結構包含有年、月、日、時、分、及秒等六項資料，如表 4 所示。

名稱	形態	大小 (byte)
Year	word	2
Month	word	2
Day	word	2
Hour	word	2
Minute	word	2
Second	word	2

表 4 時間資料結構表

(2)停車狀況的資料結構包含有停車資料、停車時間及密碼等三個資料欄位，如表 5 所示。

名稱	形態	大小 (byte)
Parked	Boolean	1
Prak Time	Time	12
Password	String	20

表 5 停車狀況資料結構表

五、實作與模擬結果

爲能證實本立體停車系統理論之可行性，本文自製了立體停車場模型並與IBM PC/AT 連接做實際運作之安全監視、控制與管理，所得結果如下：

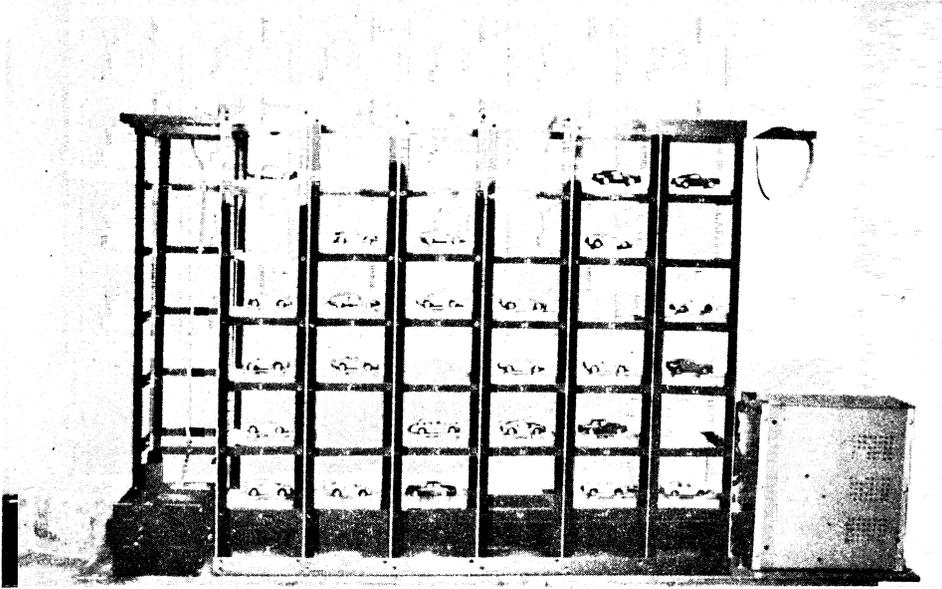


圖 10 自動立體停車場模擬系統實體圖（正視圖）

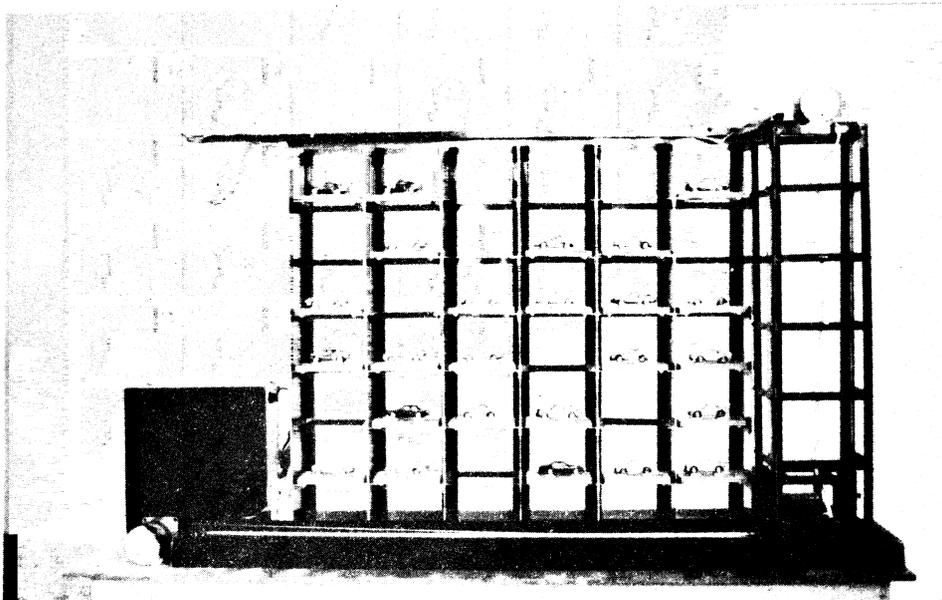


圖 11 自動立體停車場模擬系統實體圖（背視圖）

電腦自動立體停車場動畫監控管理系統

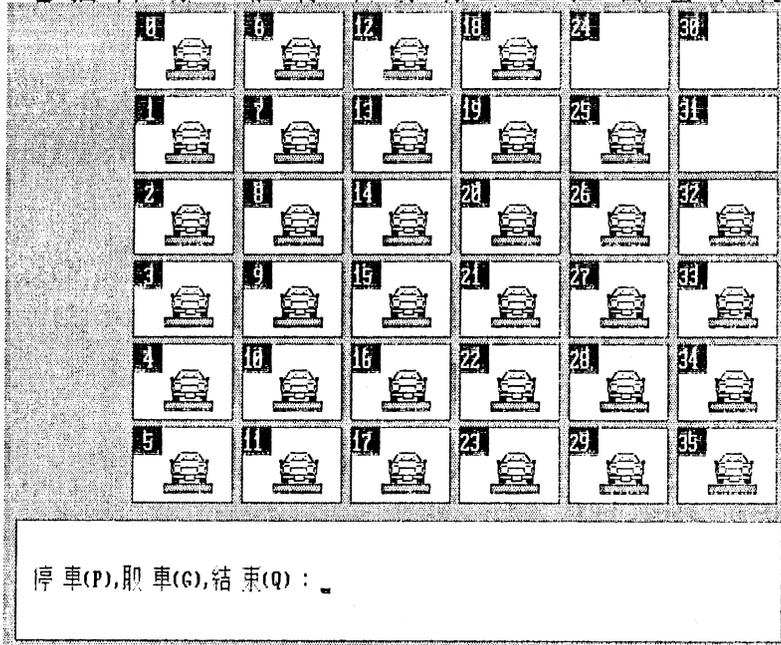


圖 12 系統模擬安全監視畫面

電腦自動立體停車場動畫監控管理系統

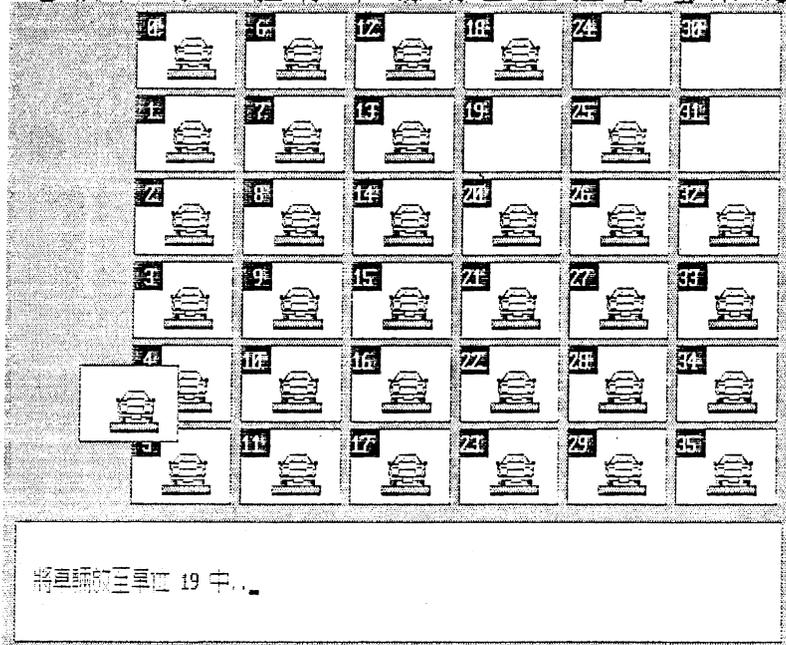


圖 13 系統存車中實際運作動態監視畫面 (去程)

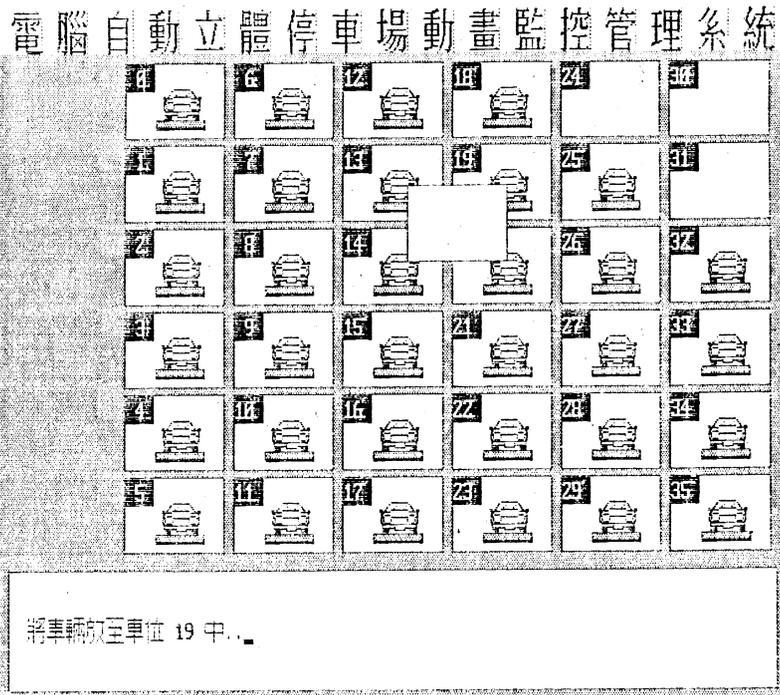


圖 14 系統存車中實際運作動態監視畫面（回程）

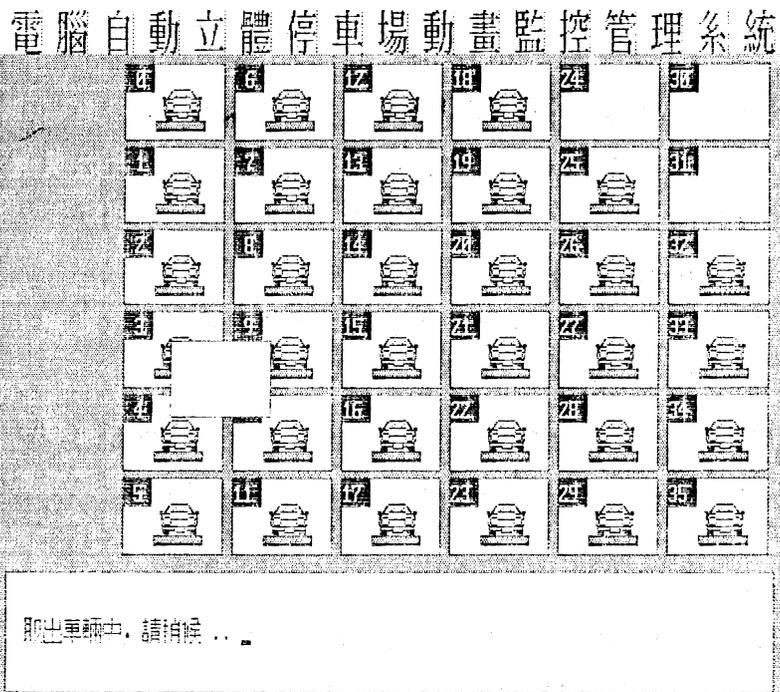


圖 15 系統取車中實際運作動態監視畫面（去程）

電腦自動立體停車場動畫監控管理系統



圖 16 系統取車中實際運作動態監視畫面 (回程)

六、結論與建議

有了汽車後，便開始擔心停車位等種種問題，而這對於地少人多的臺灣來說，更給汽車擁有者與政府造成各種困擾。而本文所設計之電腦自動立體停車場暨動畫監控管理系統經實作模擬結果，證實其可行的。

本研究所獲致的成果：

1. 將空間立體化，使原本可停一輛車位的空間，改成可停放數輛，這對停車位一位難求的臺灣來說，實有其必要性。
2. 利用電腦管理，記錄車輛存取時間並自動換算所需繳付的金額，可節省許多人力、時間及減少人為誤差。其中密碼(password)管理更提高了寄車安全性，因不正確資料是無法取車的。且斷電時電腦管理資料可被完整且正確的保護。
3. 利用TURBO PASCAL程式語言設計動畫監控程式，直接進行存車及取車作業的設備驅動控制，並對整個立體停車場實際運作安全進行動態監督。
4. 爲了節省存車、取車作業時間，本系統以走行(X軸)、昇降(Y軸)同時進行方式，並自動找尋最靠近輸入口之最短路徑停車空位存放。

5.使用極限開關(limit switch)對每次存取動作後做載車架歸位控制，以充分發揮監控程式之效率。

隨著電腦技術的不斷創新，未來的自動儲運系統在功能上將達到更令人滿意的境界，希望藉由本“自動立體停車場暨動畫監控管理系統”的研究，做一拋磚引玉的引領。然相關專業人員技術、經驗的提供更是重要的環節。最後本文提出兩個建議如下：

- 1.實際製作或再次實作模擬時，可將步進馬達(steping motor)改以直流馬達(DC motor)或伺服馬達(Servo motor)取代，且由於載車架機體高，且具走行(X軸)、昇降(Y軸)及叉動(Z軸)三種運轉動作，故可再加入額外設備如：捲揚上限極限開關、捲揚下限極限開關、防止載車架逸走極限開關、碰撞緩衝器(buffer)及昇降台緊急剎車器等做為安全設施。
- 2.本文討論於停車時係以最近空位存放方式，然若汽車欲停置時間可以確定，則可改採分級式存放。亦即根據欲停置時間的長短予以分等級，停置時間短者放於近出入口位置，在同一等級裡，則以最近空位方式作業，使停車時可更具智慧的找尋最有效率的停車空位。

七、附 錄

Turbo Pascal 語言程式：

```
uses crt,dos, et, printer;
```

```
type
```

```
Time=record (*時間的資料型態*)
```

```
    Year, Month, Day, Hour, Minute, Second: word;
```

```
end;
```

```
ParkData=record (*停車位的資料型態*)
```

```
    Parked: boolean; (*是否已經停車*)
```

```
    ParkTime: Time; (*停車時間*)
```

```
    Password: string[20]; (*密碼*)
```

```
end;
```

```
const
```

```
    CountX : array [0..5] of word =
```

```
        (290,540,793,1045,1290,1545);
```

```
    CountY : array[0..1] of array [0..5] of word =
```

```
        ((1115,895,670,440,220,0),(1100,925,700,475,250,30));
```

```
(*控制步進馬達之數據*)
```

```
    StepData : array [0..2] of array [0..3] of byte =
```

```
        (($c9,$63,$36,$9c),($09, $03,$06,$0c), ($c3,$63,$33,$9
```

```
3));
```

```
    Model=3;
```

```
    Mode2=6;
```

```
var
```

```
    ParkDataFile : file of ParkData; (*儲存停車位狀態的檔案*)
```

```
    Buf:ParkData;
```

```
    CarImage : array [0.. 134] of byte; (*儲存車輛造型之Buffer*)
```

```
    ImageBuf, CarBuf : array [0..466] of byte;
```

```
    ImageSur:array [0..1] of array [0..466] of byte;
```

```
    Ch : char;
```

```

Procedure delay (no : integer);
var
  i : integer;
  oo : real;
begin
  for i:=0 to no*100 do    (*利用執行Floating Point 運算以延遲時間*)
    oo:=-oo*0.0001;
  end;
  (*等待按鍵*)
  procedure WaitKey;
  begin
    ch:=readkey;
  end;
  (*顯示訊息*)
  procedure Message (str : string);
  begin
    clrscr;
    write (str);
  end; (* Message *)
  procedure OpenParkFile;      (*打開檔案*)
  var
    no : integer;
    carfile : file;
  begin
    assign (ParkDataFile, 'park. dat');
    {$i-}
    reset (ParkDataFile);
    {$i+}
    if ioresult<>0 then        (*若檔案不存在*)
      begin

```

```

rewrite (ParkDataFile);      (*則建立新的資料檔*)
Buf.Parked: = false;
for no:=0 to 36 do
    write (ParkDataFile, Buf);
end;
(*讀取車輛造型 *)
assign (carfile, 'car. img');
reset (carfile, 1);
blockread (carfile, CarImage, filesize (carfile));
close (carfile);
end; (* OpenParkFile *)
(*讀取資料*)
procedure ReadParkData (no : word;var Data : ParkData);
begin
    seek (ParkDataFile, no);
    read (ParkDataFile, Data);
end; (* ReadParkData *)
(* 儲存資料 *)
procedure WriteParkData(no : word;var Data : ParkData);
begin
    seek (ParkDataFile, no);
    write (ParkDataFile, Data);
end; (* WriteParkData *)
(*顯示背景 *)
procedure SetBackground;
var
    no, offset : word;
begin
    fillchar (mem[$b800:00],32768,00);
    for no : 0 to 374 do

```

```

begin
  offset := (no div 4) * 80 + (no mod 4) * $2000;
  if (no mod 2) <> 0 then
    fillchar (mem[$b800:offset],80,$aa)
  else
    fillchar (mem[$b800:offset],80, $55);
  end;
end; (* SetBackground *)
(* 移動載車架 *)
procedure Moto (ToNo, Mode : word);
var
  x,y,fx,fy,cx,cy : word;
  no, cl :integer;
begin
  port [$303]:-$80;
  x:-0;
  y:-0;
  fx:-CountX[ToNo div 6];
  fy:-CountY[Mode][ToNo mod 6];
  no:-0;
  cl:-0;
  cx:-16;
  cy:-254;
  port[$300]:-0;
  getimage(cx,cy,cx+80, cy+41, imagebuf);
  putimage(cx,cy,3,ImageSur[mode]);
  while (x<>fx) or (y<>fy) do
  begin
    (* 移動硬體之載車架 *)
    if x<fx then inc(x);

```

```

if x<fy then inc(y);
if x=fx then port[$300]:=-StepData[1][no] else
if Y=fy then port[$300]:=-StepData[2][no] else
  port [$300]:=-StepData [0] [no];
inc (no) ;
if no > 3 then no:=-0;
delay (7);
(* 在螢幕上顯示模擬動作 *)
inc (c1);
if c1>20 then
begin
  c1:=-0;
  putimage(cx,cy,3,imagebuf);
  cx:-trunc(x*0.35059+3);
  cy:-trunc (254-(y-mode*30)*0.2);
  getimage(cx,cy,cx+80,cy+41,imagebuf);
  putimage(cx,cy,3,ImageSur[mode]);
end;
end;
putimage(cx,cy,3,imagebuf);
port[$300]:=-3;
end; (* Moto *)

procedure Back (no1 : integer;mode : byte);
var
  no :integer;
  x,y,fx,fy,c1,cx,cy :integer;
  mode3 : byte;
begin
  fx:=0;

```

```

fy:= 0;
if mode=3 then mode3:-1 else mode3:-0;
x:=-CountX[no1 div 6];
y:=-CountY[0][no1 mod 6];
c1:-0;
cx:=-trunc (x*0.35059+3);
cy:=-trunc(254-y*0.2);
getimage(cx,cy,cx+80, cy+41, imagebuf);
putimage (cx,cy,3,ImageSur[mode3]);
no:=-3;
while ((Port[$301] and mode)<> mode) do
begin
  (* 移動載車架 *)
  case port [$301]and mode of
    0 : port [$300] :-StepData[0][no];
    1 : port [$300] :-StepData[2][no];
    2 : port [$300] :-StepData[1][no];
    4 : port [$300] :-StepData[2][no];
  end;
  dec (no,1);
  if no<0 then no:=-3;
  if x>fx then dec (x);
  if y>fy then dec (y);
  inc(c1);
  if c1>20 then
  begin
    c1:=0;
    putimage (cx,cy,3,imagabuf);
    cx:=-trunc(x*0.35059+3);
    cy:=-trunc(254-y*0.2);

```

```

    getimage (cx,cy,cx+80,cy+41,imagebuf);
    putimage (cx,cy,3,ImageSur[mode3]);
end else
delay (15);
end;
putimage(cx,cy,3,imagebuf);
port. [$300]:=0;
end; (* Back *)
procedure PutCar (No1 : integer;mode : integer);
const
    StepData2 : array[0..1] of array[0..3] of byte =
        (($c0,$60,$30,$90),($09,$03,$06,$0c));
var
    x,y : integer;
    Count,no : integer;
begin
    port [$303]:=$82;
    port [$300]:=$3;
    no:=0;
    for Count:=0 to 155 do
    begin
        port [$302]:=StepData2[0][no];
        inc (no);
        if no>3 then no:=0;
        delay(15);
    end;
    port [$302]:=0;
    for Count:= 0 to 100 do
    begin
        port [$300]:=StepData2[1][no];

```

```

if mode=-1 then
  begin
    inc(no);
    if no>3 then no:=0;
  end
else
  begin
    dec(no);
    if no<0 then no:=-3;
  end;
  DELAY(15);
end;
prot[$300]:=-3;
for Count:-0 to 155 do
begin
  port[$302]:=-StepData2[0][no];
  dec(no);
  if no<0 then no:=-3;
  DELAY(15);
end;
port[$302]:=0;
if no1<>-1 then
begin
  inc (No 1, 6);
  x:=-16+(No1 div 6)*88;
  y:=-29+(No1 mod 6)*45;
  if mode = -1 then
    putimage (X+25,y+11,3,CarImage)
  else
    line (x+24,y+11,x+79,y+40,1,0,$d);

```

```

end
and; (* PutCar *)
procedure InitScreen;          (*規劃初始畫面*)
var
  i,j : word;
begin
  ScanTime(0);
  directvideo:=false;
  textcolor(7);
  texbackground(0);
  clrscr;
  SetBackground;
  writeln(0 to 5 do
    for j:-0 to 5 do
      begin
        line (104+i*88,29+j*45,184+i*88,70+j*45,1,0,$d);
        line (104+i*88,29+j*45,184+i*88,70+j*45,0,0,9);
        gotoxy(14+i*11,3+j*3);
        write(i*6+j:2,' ');
      end;
    getimage (16,254,96,295, ImageBuf);
    line(16,254,96,295,1,0,$d);
    line(16,254,96,295,0,0,9);
    getimage(16,254,96,295,ImageSur[1]);
    putimage(41,265,3,CarImage);
    getimage(16,254,96,295,ImageSur[0]);
    line (17,255,95,294,1,0,$d);
    putimage(16,254,3,ImageBuf);

  (* 繪出停車場狀況 *)

```

```

for i:-0 to 5 do
for j:-0 to 5 do
begin
  seek(ParkDataFile, i*6+j);
  read(ParkDataFile, Buf);
  if Buf.Parked then
    putimage(129+i*88,40+j*45,3,CarImage);
end;
line(10,305,629,370,1,0,$d);
line(10,305,629,370,0,0,9);
textcolor(0);
textbackground(7);
window(4,23,78,23);
end; (* InitScreen *)
(* 停車作業 *)
procedure StoreCar;
const
  Store Order : array[0..36] of word =
    (5,4,11,3,10,17,2,9,16,23,1,8,
     27,34,12,19,26,33,18,25,32,24,31,30,36);
var
  no, nouse : word;
begin
  (* 找尋空的車空 *)  (* 找尋空的車位 *)
  no:-0;
  ReadParkData (Store Order[no],Buf);
  with Buf do
    while Parked and (no <= 35) do
      begin
        inc (no);

```

```

    ReadParkData (StoreOrder[no].Buf);
end;
if no <- 35 then
begin
    no:=StoreOrder[no];
    Message('請輸入密碼 : ');
    readln (Buf. Password);
    (* 更改資料內容 *)
    Message('列印停車單據中..');
    with Buf.ParkTime do
    begin
        gettime (Hour, Minute, Second, nouse);
        getdate (Year, Month, Day, nouse);
        writeln (lst, '
');
        writeln(lst,'
');
        writeln(lst,'          電腦自動立體停車場          ');
        writeln(lst,'          ');
        writeln(lst,'          車位編號 : ',no:2,'          | ');
        writeln(lst,'          ');
        writeln(lst,'          停車時間 : ',year:4,'年',Month:2,'月',day:2,'日'
                Hour:2,':',Minute:2,':',Second:"2,' | ');
        writeln(lst,'          ');
        writeln(lst,'          密碼 : ',Buf.Password:20,'          ');
        writeln(lst,'          ');
        writeln(lst,'          -----');
    end;
    Buf.Parked :=true;
    WriteParkData (no,Buf);

```

```

clrscr;
write('將車輛放至車位' ,no:2,' 中..');
PutCar(-1,1);
woto(no,0);
PutCar(no,-1);
delay (100);
Back(no,Model);
end
else
begin
  Message('抱歉, 目前已經沒有車位了! '#7);
  WaitKey;
end;
end; (* StoreCar *)
(* 計算停車的秒數 *)
function Seconds (year, month,day,hour,minute,second : word) :real;
const
  months : array[1..12] of byte =
    (31,28,31,30,31,30,31,31,30,31,30,31);
var
  days : real;
  i : word;
begin
  (* 計算是否為閏年 *)
  if (((year mod 4)=0) and ((year mod 100)≠0))
    or (((year mod 4)=0) and ((year mod 400)=0)) then
    months[2]:=-29 else months[2]:28;

```

```

(* 計算總天數 *)
dec (year);
days:=(year div 4) + (year div 100) - (year div 400) + year *365;
for i:-1 to month-1 do
    days:=-days+months[i];
days:=-days+day-1;
(* 計算總秒數 *)
Seconds:=(((days*24+hour)*60+minute)*60+second);
end; (* Seconds *)
(* 取車作業 *)
procedure RestoreCar;
var
    nostr : string;
    no.p : word;
    y1,m1,d1,h1,m1,s1,u1,u2 : word;
    pass : string[20];
(* 計算停車費用 *)
function Fare : real;
var
    m :real;
begin
    (* 計算停車時間 *)
    gettime (h1,m1,s1,u1);
    getdate (y1,m1,d1,u1);
    with Buf.ParkTime do
        t1:=-Seconds(year,month,day,hour,minute,second);
        t2:=-Seconds(y1,m1,d1,h1,m1,s1);
    (* 計算停車費用 *)
    m:=(int ((t2-t1)/360+9)/10)*10; (* 停車費用每小時10元 *)
    if m<10 then m:=10; (* 基本費用10元 *)

```

```
Fare := m;
end; (* Fare *)
begin
repeat
  Message ('請輸入欲取出的車位編號 :');
  readln (nostr);
  if nostr='' then exit;
  val (nostr, no, p);
  if p<> 0 then
    no:=-36;
  if (no<0) or (no>35) then
  begin
    Message ('編號超出範圍! '#7);
    WaitKey;
  end;
until
ReadParkData (no, Buf);
if Buf.Parked then
  begin
    Message ('請輸入密碼:');
    readln(pass);
    if (pass=Buf.Password) OR (pass = 'boss') then
      begin
        with Buf.ParkTime do
          begin
            money:=-Fare;
            clrscr;
            write('請付停車費用 ',money:4:0,'元');
            WaitKey;
            Message('列印收據中，請稍候..');
```

```

WriteLn (lst,
writeln(lst,' ');
writeln(lst,' 電腦自動立體停車場收據 ');
writeln(lst,' ');
writeln(lst,' 車位編號: ',no:2,' ');
writeln(lst,' ');
writeln(lst,' 停車時間: ;year:4,'年',Month:2,'月',day:2,'日'
Hour:2,':',Minute:2,':',Second"2,' ');
writeln(lst,' ');
writeln(lst,' 取車時間: ',y1:4,'年',M1:2,'月',d1:2,'日',
H1:2,':',Mi1:2,':',S1:2,' ');
writeln(lst,' ');
writeln(lst,' 停車費用: ',money:5:0,'元整 ');
writeln(lst,' ');
writeln(lst,' -----');
Message('取出車輛中，請稍候...');

```

```
end;
```

```

Moto (no,1); (* 移動載車架至 no 號車位 *)
PutCar (no,1);(* 取出車輛 *)
Back (no, Mode2);(* 移動載車架至定位停車台 *)
PutCar (-1, -1); (* 放下車輛 *)
Buf.Parked := false;
WriteParkData(no,Buf);
Message ('謝謝光臨，祝您旅途愉快!');
delay (1500);

```

```
end
```

```
else
```

```
begin
```

```

Message('密碼輸入錯誤，請再輸入! '#7);
WaitKey;

```

```
end;
else
begin
  clrscr;
  write('抱歉,在 ',no:2,' 車位中,並無任何車輛!','#7);
  WaitKey;
end;
end; (* RestoreCar *)
procedure FinalScreen; (* 結束作業處理 *)
begin
  close (ParkDataFile); (* 關閉資料檔 *)
  window (1,1,80,25);
  textcolor (7);
  textbackground(0);
  clrscr;
  EtCmd('a');
  scantime(5);
end; (* FinalScreen *)
begin
  Back (35, model);
  OpenParkFile; (* 打開資料檔 *)
  InitScreen; (* 規劃初始畫面 *)
  repeat
    Message ('停車(P), 取車(G), 結束(Q):');
    repeat
      Ch:=upcase (readkey);
    until Ch in ['P','G', 'Q'];
    case Ch of
      'P' :StoreCar; (* 執行停車功能 *)
```

```
'G' :RestoreCar; (* 執行取車功能 *)  
end;  
until Ch = 'Q';  
FinalScreen; (* 結束作業處理 *)  
end.
```

參考文獻

- 1.唐煬烈，「路邊自動停車系統之研究與模擬」，碩士論文，逢甲大學，臺中(1987)。
- 2.關剛石，「自動儲運系統規劃」，工研院機械所講義，新竹，第1-1~1-88頁(1989)。
- 3.吳志信、葉忠闊、許敦年，「自動儲運專集」，機械工業雜誌，第十八期，第33-49 頁(1984)。
- 4.Intel Corporation, 8085 Microcomputer system user's Manual, Taipei, pp.4-27~4-48 (1977)。
- 5.許書務，光電元件應用技術，第87-89 頁，全華圖書，台北(1986)。
- 6.白中和，步進馬達控制電路設計，第27-28頁，建興出版社，台北(1988)。
- 7.林永裁、鄭美珠，微電腦系統I/O 控制實驗，第9-10~9-17頁，松崗圖書，台北(1989)。
- 8.曹德貴、王順賢，TURBO PASCAL 5.0 使用手冊，第43-68 頁，明道出版社，台北(1989)。
- 9.曹德貴、王順賢，TURBO PASCAL 5.0參考手冊，第243 頁，明道出版社，台北(1989)。
- 10.倚天資訊，倚天中文系統使用手冊，第9章第3頁，倚天資訊，台北(1989)。