**B. Program Code**

.model small

.stack

.data

a0 db " START YOUR ENGINE$"

a1 db " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ $"

a2 db " /###############################\ $ "

a3 db " /W/ \W\ $"

a4 db " /W/ \W\ $"

a5 db " /W/ \W\ $"

a6 db " @@@|HH\HHHHHHHHHHHHHHHHHHHHHHH/HH|@@@ $"

a7 db " @@|HHH\\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/HHH |@@ $"

a8 db " |HHHH\\_\_@@\_\_\_\======/\_\_@@\_\_\_/HHHH| $"

a9 db " |HHHHHHHHHHH CPE-143 HHHHHHHHHH| $"

a10 db " |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX| $"

a11 db " HHHHH \+++++++++++++++++++/ HHHHH $"

a12 db " HHHHH HHHHH$"

msg1 db " SELECT RC CAR TRANSMISSION$"

msg2 db " [A] AUTOMATIC$"

msg3 db " [M] MANUAL$ "

msg4 db " [X] EXIT$"

msg5 db "ENTER YOUR CHOICE : $"

msg6 db " AUTOMATIC MODE$"

msg7 db " MANUAL MODE$"

msg8 db "MANUAL RC CAR CONTROLS$"

msg9 db " [W] FORWARD$"

msg10 db " [Q] Forward/LEFT$"

msg11 db " [A] Backward/LEFT$"

msg12 db " [E] Forward/RIGHT$"

msg13 db " [D] Backward/RIGHT$"

msg14 db " [S] BACKWARD$"

msg15 db " [B] BACK TO MAIN MENU$"

msg16 db " MANUAL MODE$"

msg17 db " FORWARD$"

msg18 db"Backward/LEFT$"

msg19 db"Backward/RIGHT$"

msg20 db " FORWARD$"

msg21 db " BACKWARD$"

msg22 db " FORWARD/LEFT$"

msg23 db " FORWARD/RIGHT$"

.code

start:

mov ax,@data

mov ds,ax

main proc

call backg

call exit

main endp

cls proc

mov ah, 06h

mov bh, 07

mov cx, 0000h

mov dx, 184fh

int 10h

cls endp

backg proc

mov ah, 06h

mov bh, 0fh

mov cx, 0000h

mov dx, 184fh

int 10h

mov bh, 0f0h ;1

mov cx, 0000h

mov dx, 0305h

int 10h

mov bh, 0f0h;2

mov cx, 000bh

mov dx, 0310h

int 10h

mov bh, 0f0h;3

mov cx, 0016h

mov dx, 031bh

int 10h

mov bh, 0f0h;4

mov cx, 0032h

mov dx, 0337h

int 10h

mov bh, 0f0h;5

mov cx, 003eh

mov dx, 0343h

int 10h

mov bh, 0f0h;6

mov cx, 004ah

mov dx, 034fh

int 10h

mov bh, 0fh;<7

mov cx, 0300h

mov dx, 0606h

int 10h

mov bh, 0f0h;7

mov cx, 0306h

mov dx, 0610h

int 10h

mov bh, 0fh ;<8

mov cx, 030bh

mov dx, 0610h

int 10h

mov bh, 0f0h ;8

mov cx, 0311h

mov dx, 0615h

int 10h

mov bh, 0eh;<9;<------------

mov cx, 0316h

mov dx, 0637h

int 10h

mov bh, 0f0h;<12

mov cx, 0338h

mov dx, 063eh

int 10h

mov bh, 0fh;12

mov cx, 033eh

mov dx, 0643h

int 10h

mov bh, 0f0h;<13

mov cx, 0344h

mov dx, 064ah

int 10h

mov bh, 0fh;13

mov cx, 034ah

mov dx, 064fh

int 10h

mov bh, 10h;<7;3layer

mov cx, 0600h

mov dx, 0906h

int 10h

mov bh, 1fh;7

mov cx, 0606h

mov dx, 0910h

int 10h

mov bh, 1eh ;<8

mov cx, 060bh

mov dx, 0910h

int 10h

mov bh, 0eh ;8

mov cx, 0611h

mov dx, 0915h

int 10h

mov bh, 0eh;<9;<------------

mov cx, 0616h

mov dx, 0937h

int 10h

mov bh, 0eh;<12

mov cx, 0638h

mov dx, 093eh

int 10h

mov bh, 1eh;12

mov cx, 063eh

mov dx, 0943h

int 10h

mov bh, 1fh;<13

mov cx, 0644h

mov dx, 094ah

int 10h

mov bh, 10h;13

mov cx, 064ah

mov dx, 094fh

int 10h

mov bh, 1fh;<7;4layer

mov cx, 0900h

mov dx, 0c06h

int 10h

mov bh, 1fh;7

mov cx, 0906h

mov dx, 0c10h

int 10h

mov bh, 1fh ;<8

mov cx, 090bh

mov dx, 0c10h

int 10h

mov bh, 0eh ;8;<\_\_\_\_\_\_\_\_

mov cx, 0911h

mo v dx, 0c3eh

int 10h

mov bh, 1fh;12

mov cx, 093eh

mov dx, 0c43h

int 10h

mov bh, 10h;<13

mov cx, 0944h

mov dx, 0c4ah

int 10h

mov bh, 1fh;13

mov cx, 094ah

mov dx, 0c4fh

int 10h

mov bh, 10h;<7;5layer

mov cx, 0c00h

mov dx, 0f06h

int 10h

mov bh, 11h;7

mov cx, 0c06h

mov dx, 0f10h

int 10h

mov bh, 10h ;<8

mov cx, 0c0bh

mov dx, 0f10h

int 10h

mov bh, 0eh ;8;<------------

mov cx, 0c11h

mov dx, 0f3eh

int 10h

mov bh, 10h;12

mov cx, 0c3eh

mov dx, 0f43h

int 10h

mov bh, 11h;<13

mov cx, 0c44h

mov dx, 0f4ah

int 10h

mov bh, 10h;13

mov cx, 0c4ah

mov dx, 0f4fh

int 10h

mov bh, 0fh;<7;6layer

mov cx, 0f00h

mov dx, 1206h

int 10h

mov bh, 0efh;7

mov cx, 0f06h

mov dx, 1010h

int 10h

mov bh, 0fh;7

mov cx, 1006h

mov dx, 1210h

int 10h

mov bh, 0efh;7

mov cx, 1206h

mov dx, 1410h

int 10h

mov bh, 0fh;7

mov cx, 1406h

mov dx, 1610h

int 10h

mov bh, 0efh;7

mov cx, 1606h

mov dx, 1810h

int 10h

mov bh, 0fh ;<8

mov cx, 0f0bh

mov dx, 1210h

int 10h

mov bh, 80h ;8;<\_\_\_\_\_\_\_\_

mov cx, 0f11h

mov dx, 123eh

int 10h

mov bh, 0fh;12

mov cx, 0f3eh

mov dx, 1243h

int 10h

mov bh, 0f0h;<13

mov cx, 0f44h

mov dx, 124ah

int 10h

mov bh, 0efh;7

mov cx, 0f44h

mov dx, 104ah

int 10h

mov bh, 0fh;7

mov cx, 1044h

mov dx, 124ah

int 10h

mov bh, 0efh;7

mov cx, 1244h

mov dx, 144ah

int 10h

mov bh, 0fh;7

mov cx, 1444h

mov dx, 164ah

int 10h

mov bh, 0efh;7

mov cx, 1644h

mov dx, 184ah

int 10h

mov bh, 0fh;13

mov cx, 0f4ah

mov dx, 124fh

int 10h

mov bh, 0f0h;rightline

mov cx, 0f3bh

mov dx, 183ch

int 10h

mov bh, 0f0h;leftline

mov cx, 0f12h

mov dx, 1813h

int 10h

mov bh, 0f0h;line

mov cx, 0f25h

mov dx, 1126h

int 10h

mov bh, 80h;line

mov cx, 1125h

mov dx, 1326h

int 10h

mov bh, 0f0h;line

mov cx, 1325h

mov dx, 1526h

int 10h

mov bh, 80h;line

mov cx, 1725h

mov dx, 1826h

int 10h

mov ah, 02

mov bh, 0

mov dh, 1

mov dl, 5

int 10h

mov ah, 09

mov dx, offset a0

int 21h

mov ah, 02

mov bh, 0

mov dh, 3

mov dl, 5

int 10h

mov ah, 09

mov dx, offset a1

int 21h

mov ah, 02

mov bh, 0

mov dh, 4

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a2

int 21h

mov ah, 02

mov bh, 0

mov dh, 5

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a3

int 21h

mov ah, 02

mov bh, 0

mov dh, 6

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a4

int 21h

mov ah, 02

mov bh, 0

mov dh, 7

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a5

int 21h

mov ah, 02

mov bh, 0

mov dh, 8

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a6

int 21h

mov ah, 02

mov bh, 0

mov dh, 9

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a7

int 21h

mov ah, 02

mov bh, 0

mov dh, 10

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a8

int 21h

mov ah, 02

mov bh, 0

mov dh, 11

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a9

int 21h

mov ah, 02

mov bh, 0

mov dh, 12

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a10

int 21h

mov ah, 02

mov bh, 0

mov dh, 13

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a11

int 21h

mov ah, 02

mov bh, 0

mov dh, 14

mov dl, 5

int 10h

mov ah, 09h

mov dx, offset a12

int 21h

mov ah, 02

mov bh, 0

mov dh, 16

mov dl, 17

int 10h

mov ah, 09

mov dx, offset msg1

int 21h

mov ah, 02

mov bh, 0

mov dh, 17

mov dl, 21

int 10h

mov ah, 09

mov dx, offset msg2

int 21h

mov ah, 02

mov bh, 0

mov dh, 18

mov dl, 21

int 10h

mov ah, 09h

mov dx, offset msg3

int 21h

mov ah, 02

mov bh, 0

mov dh, 19

mov dl, 21

int 10h

mov ah, 09h

mov dx, offset msg4

int 21h

mov ah, 02

mov bh, 0

mov dh, 21

mov dl, 27

int 10h

mov ah, 09h

mov dx, offset msg5

int 21h

mov ah, 01h

int 21h

cmp al, 'a'

je automatic

jne nxt1

nxt1:

cmp al, 'A'

je automatic

jne nxt2

nxt2:

cmp al, 'm'

je gotomanual

jne nxt3

nxt3:

cmp al, 'M'

je gotomanual

jne nxt4

nxt4:

cmp al, 'x'

je gotoexit

jne nxt5

nxt5:

cmp al, 'X'

je gotoexit

jne gotoerror

backg endp

gotoexit:

call exit

gotoerror:

call cls

call start

gotomanual:

call goto2manual

automatic proc

mov ah, 06h

mov bh, 07

mov cx, 0000h

mov dx, 184fh

int 10h

MOV AH,06H;cardesign

MOV bh,0e0h

mov cx,0225h

mov dx, 0829h

int 10h

MOV AH,06H;cardesign

MOV bh,0fh

mov cx,0825h

mov dx, 0a29h

int 10h

MOV AH,06H;cardesign

MOV bh,0e0h

mov cx,0a25h

mov dx, 1429h

int 10h

MOV AH,06H;cardesign

MOV bh,0fh

mov cx,1425h

mov dx, 1629h

int 10h

MOV AH,06H;cardesign

MOV bh,0e0h

mov cx,1625h

mov dx, 1829h

int 10h

MOV AH,06H

MOV bh,0e0h

mov cx,0010h

mov dx, 1819h

int 10h

MOV AH,06H

MOV bh,0e0h

mov cx,0035h

mov dx, 183eh

int 10h

mov ah, 02

mov bh, 0

mov dh, 5

mov dl, 25

int 10h

mov ah, 09

mov dx, offset msg6

int 21h

automaticinput:

mov dx,379h

in al,dx

test al, 00001000b

je leftauto

test al, 00010000b

je rightauto

jnz straight

straight:

mov dx,378h

mov al,00000001b

out dx,al

mov al,00000000b

out dx,al

call delay1

mov ah, 02

mov bh, 0

mov dh, 7

mov dl, 25

int 10h

mov ah, 09

mov dx, offset msg17

int 21h

jmp automaticinput

leftauto:

mov dx,378h

mov al,00001010b

out dx,al

mov ah, 02

mov bh, 0

mov dh, 7

mov dl, 25

int 10h

mov ah, 09

mov dx, offset msg18

int 21h

call delay2

jmp automaticinput

rightauto:

mov dx,378h

mov al,00000110b

out dx,al

mov ah, 02

mov bh, 0

mov dh, 7

mov dl, 25

int 10h

mov ah, 09

mov dx, offset msg19

int 21h

call delay2

jmp automaticinput

automatic endp

goto2manual:

call manual

manual proc

input:

mov ah, 06h

mov bh, 07h

mov cx, 0000h

mov dx, 184fh

int 10h

mov ah, 06h

mov bh, 0e0h

mov cx, 0000h

mov dx, 1828h

int 10h

mov ah,06h

mov bh,41h

mov cx,0028h

mov dx,084fh

int 10h

mov ah,06h

mov bh,1fh

mov cx,0828h

mov dx,104fh

int 10h

mov ah,06h

mov bh,41h

mov cx,1028h

mov dx,184fh

int 10h

mov ah, 02

mov bh, 0

mov dh, 3

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg7

int 21h

mov ah, 02

mov bh, 0

mov dh, 7

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg8

int 21h

mov ah, 02

mov bh, 0

mov dh, 8

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg9

int 21h

mov ah, 02

mov bh, 0

mov dh, 9

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg10

int 21h

mov ah, 02

mov bh, 0

mov dh, 10

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg11

int 21h

mov ah, 02

mov bh, 0

mov dh, 11

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg12

int 21h

mov ah, 02

mov bh, 0

mov dh, 12

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg13

int 21h

mov ah, 02

mov bh, 0

mov dh, 13

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg14

int 21h

mov ah, 02

mov bh, 0

mov dh, 14

mov dl, 6

int 10h

mov ah, 09

mov dx, offset msg15

int 21h

mov ah, 02

mov bh, 0

mov dh, 16

mov dl, 7

int 10h

mov ah, 09

mov dx, offset msg5

int 21h

mov ah, 02h

mov ah, 02

mov bh, 0

mov dh, 16

mov dl, 26

int 10h

mov ah, 01

int 21h

cmp al, 'w'

je forward

jne nxtd1

nxtd1:

cmp al, 'W'

je forward

jne nxtd2

nxtd2:

cmp al, 's'

je gotobackward

jne nxtd3

nxtd3:

cmp al, 'S'

je gotobackward

jne nxtd4

nxtd4:

cmp al, 'q'

je gotoforwardleft

jne nxtd5

nxtd5:

cmp al, 'Q'

je gotoforwardleft

jne nxtd6

nxtd6:

cmp al, 'a'

je gotobackwardleft

jne nxtd7

nxtd7:

cmp al, 'A'

je gotobackwardleft

jne nxtd8

nxtd8:

cmp al, 'e'

je gotoforwardright

jne nxtd9

nxtd9:

cmp al, 'E'

je gotoforwardright

jne nxtd10

nxtd10:

cmp al, 'd'

je gotobackwardright

jne nxtd11

nxtd11:

cmp al, 'D'

je gotobackwardright

jne nxtd12

nxtd12:

cmp al, 'b'

je gotomain

jne nxtd14

nxtd14:

cmp al, 'B'

je gotomain

jne error

gotomain:

call start

error:

call manual

ret

manual endp

gotobackward:

call backward

gotoforwardleft:

call forwardleft

gotobackwardleft:

call backwardleft

gotoforwardright:

call forwardright

gotobackwardright:

call backwardright

forward proc

mov dx, 378h

mov al, 00000001b

out dx, al

mov ah, 02

mov bh, 0

mov dx,0b32h

int 10h

mov ah, 09

mov dx, offset msg20

int 21h

call delay1

call reset

call input

forward endp

backward proc

mov dx, 378h

mov al, 00000010b

out dx, al

mov ah, 02

mov bh, 0

mov dx,0b32h

int 10h

mov ah, 09

mov dx, offset msg21

int 21h

call delay1

call reset

call input

backward endp

forwardleft proc

mov dx, 378h

mov al, 00001001b

out dx, al

mov ah, 02

mov bh, 0

mov dx,0b32h

int 10h

mov ah, 09

mov dx, offset msg22

int 21h

call delay1

call reset

call input

forwardleft endp

forwardright proc

mov dx, 378h

mov al, 00000101b

out dx, al

mov ah, 02

mov bh, 0

mov dx,0b32h

int 10h

mov ah, 09

mov dx, offset msg23

int 21h

call delay1

call reset

call input

forwardright endp

backwardleft proc

mov dx, 378h

mov al, 00000110b

out dx, al

mov ah, 02

mov bh, 0

mov dx,0b32h

int 10h

mov ah, 09

mov dx, offset msg18

int 21h

call delay1

call reset

call input

backwardleft endp

backwardright proc

mov dx, 378h

mov al, 00001010b

out dx, al

mov ah, 02

mov bh, 0

mov dx,0b32h

int 10h

mov ah, 09

mov dx, offset msg19

int 21h

call delay1

call reset

call input

backwardright endp

reset proc

mov ah, 02h

mov bh, 0

mov dx, 0000h

int 21h

mov dx, 378h

mov al, 00000000h

out dx, al

ret

reset endp

delay1 proc

push bx

push cx

mov bx, 4500

BB1:

mov cx, 8000

C1:

NOP

Loop C1

dec bx

jnz BB1

pop cx

pop bx

ret

delay1 endp

delay2 proc

push bx

push cx

mov bx, 18000

BB2:

mov cx, 30000

C2:

NOP

Loop C2

dec bx

jnz BB2

pop cx

pop bx

ret

delay2 endp

exit proc

mov ah,4ch

int 21h

exit endp

end start