

ภาคผนวก

ภาคผนวก ก

โปรแกรมการสร้างภาพจำลองการเคลื่อนที่ของหุ่นยนต์

KAWASAKI UNIMATE MACHINE

ROBOT.CPP

ROBOT.CPP

```
#include<stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <graphics.h>
#include <alloc.h>
#include <bios.h>
#include <dos.h>
#include "m2-d.cpp"
#include <process.h>
#include "hscreen.cpp"
#include "psc.cpp"
#define NO_CHOICE 6
#define NO_MENU 5

#define UP_ARROW 0x4800
#define DOWN_ARROW 0x5000
#define LEFT_ARROW 0x4B00
#define RIGHT_ARROW 0x4D00
#define RETURN 0x1C0D
#define ESCAPE 0x011b
#define F10 0x4400
#define ALT_X 0x2d00

void screen()
{
    setcolor(15);
    if(getmaxcolor()==1)setfillstyle(SOLID_FILL,0);
    else setfillstyle(SOLID_FILL,LIGHTBLUE);
```

```

bar3d(0,0,getmaxx(),getmaxy()*19/20,10,1);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setcolor(RED);
outtextxy(getmaxx()/2,getmaxy()*18/20,"press any key to continou...");
setfillstyle(SOLID_FILL,RED);
bar3d(getmaxx()*1/4-50,getmaxy()*1/4,getmaxx()*3/4+50,getmaxy()*2/4,0,1);
settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
setcolor(getmaxcolor());
setviewport(getmaxx()*1/4-50,getmaxy()*1/4,getmaxx()*3/4+50,getmaxy()*2/4-35,1);
outtextxy(45,45,"ROBOT SIMULATION PROGRAM");

setviewport(0,0,getmaxx(),getmaxy(),1);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setfillstyle(SOLID_FILL,LIGHTMAGENTA);
bar3d(0,getmaxy()*19/20,getmaxx(),getmaxy(),0,1);
setcolor(15);
outtextxy(10,465,"F10-menu ALT-X to quit esc_to Unpopup Window");
getch();
closegraph();
}

void nscr()
{
setviewport(0,0,getmaxx(),getmaxy(),1);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar3d(0,0,getmaxx(),19,0,1);
/* setviewport(0,0,getmaxx(),getmaxy(),1);
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
setfillstyle(SOLID_FILL,LIGHTBLUE);
bar3d(0,19,getmaxx(),getmaxy()*19/20,0,1);

```

```
*/  
setviewport(0,0,getmaxx(),getmaxy(),1);  
settextstyle(DEFAULT_FONT,HORIZ_DIR,1);  
setfillstyle(SOLID_FILL,LIGHTMAGENTA);  
bar3d(0,getmaxy()*19/20,getmaxx(),getmaxy(),0,1);  
setcolor(15);  
outtextxy(10,465,"F10-menu ALT-X to quit esc_to Unpopup Window");  
}
```

```
// initialize graphics
```

```
void init()
```

```
{
```

```
int gdriver=DETECT,gmode,errorcode;
```

```
initgraph(&gdriver,&gmode,"");
```

```
errorcode=graphresult();
```

```
if(errorcode != grOk)
```

```
{
```

```
printf("Graphics error:%s \n" ,grapherrormsg(errorcode));
```

```
printf("press any key to halt");
```

```
getchar();
```

```
closegraph();
```

```
exit(1);
```

```
}
```

```
}
```

```
typedef struct heading
```

```
{
```

```
char *choice;
```

```
};
```

```
/* data structure menu */

typedef struct menu_struct
{
    int frame[4];
    int row[NO_CHOICE];
    int col;
    struct heading item[NO_CHOICE];
    int last_choice;
};

void Initialize_Graphics_Mode(void);
void Close_Graphics_Mode(void);
void Draw_Rectangle(int lt,int tp,int rt,int bt);
void Draw_Fill_Rectangle(int lt,int tp,int rt,int bt);
void Erase_Fill_Rectangle(int lt,int tp,int rt,int bt);
void Set_Fill_Pattern(int pattern,int color);
void Set_Color(int color);
void Out_Text_XY(int x,int y,char text[]);
int Get_Max_X(void);
int Get_Max_Y(void);

void Menu_Assignment(void);
void Display_main_Menu(void);
void Display_Menu(int menu_no);
void Select_Menu(int menu_no,int choice_no);
void Unselect_Menu(int menu_no,int choice_no);
int Read_Key(int key);
void Inverse(int menu_no,int choice_no);
```

```
void Normal(int menu_no,int choice_no);
```

```
struct menu_struct menus[NO_MENU];
```

```
int start = 5,
```

```
one_choice_width = 20;
```

```
void Initialize_Graphics_Mode(void)
```

```
{
```

```
int gdriver = DETECT, gmode, errorcode;
```

```
// char msg[80];
```

```
/* initialize graphics and local variables */
```

```
initgraph(&gdriver, &gmode, "");
```

```
/* read result of initialization */
```

```
errorcode = graphresult();
```

```
if (errorcode != grOk) /* an error occurred */
```

```
{
```

```
printf("Graphics error: %s\n", grapherrormsg(errorcode));
```

```
printf("Press any key to halt:");
```

```
getch();
```

```
exit(1); /* terminate with an error code */
```

```
}
```

```
}
```

```
void Close_Graphics_Mode(void)
```

```
{
    closegraph();
}

void Draw_Rectangle(int lt,int tp,int rt,int bt)
{
    rectangle(lt,tp,rt,bt);
}

void Draw_Fill_Rectangle(int lt,int tp,int rt,int bt)
{
    Set_Fill_Pattern(1,15);
    bar(lt,tp,rt,bt);
}

void Erase_Fill_Rectangle(int lt,int tp,int rt,int bt)
{
    Set_Fill_Pattern(1,0);
    bar(lt,tp,rt,bt);
}

void Set_Fill_Pattern(int pattern,int color)
{
    setfillstyle(pattern,color);
}

void Out_Text_XY(int x,int y,char text[])
{
    outtextxy(x,y,text);
}

void Set_color(int color)
{
    setcolor(color);
}

int Get_Max_X(void)
```

```
{
    return(getmaxx());
}

int Get_Max_Y(void)
{
    return(getmaxy());
}

void Menu_Assignment(void)
{
    int one_part;
    one_part=(Get_Max_X()/NO_MENU);

    menus[0].frame[0] = start;
    menus[0].frame[1] = one_choice_width;
    menus[0].frame[2] = start+one_part;
    menus[0].frame[3] = one_choice_width+one_choice_width*5;
    menus[0].frame[4] = start+one_part+one_part;

    menus[0].row[0] = 6;
    menus[0].row[1] = 26;
    menus[0].row[2] = 46;
    menus[0].row[3] = 66;
    menus[0].row[4] = 86;
    menus[0].row[5] = 106;

    menus[0].col = start;
    start+=one_part;
```

```
menus[0].item[0].choice=" File ";
menus[0].item[1].choice=" New ";
menus[0].item[2].choice=" Open ";
menus[0].item[3].choice=" Save ";
menus[0].item[4].choice=" Dosshell ";
menus[0].item[5].choice=" Quit ";
```

```
menus[0].last_choice=5;
```

```
menus[1].frame[0] = start;
menus[1].frame[1] = one_choice_width;
menus[1].frame[2] = start+one_part;
menus[1].frame[3] = one_choice_width+one_choice_width*4;
```

```
menus[1].row[0] = 6;
menus[1].row[1] = 26;
menus[1].row[2] = 46;
menus[1].row[3] = 66;
menus[1].row[4] = 86;
```

```
menus[1].col = start;
start+=one_part;
```

```
menus[1].item[0].choice=" Edit ";
menus[1].item[1].choice=" Cut ";
menus[1].item[2].choice=" Copy";
menus[1].item[3].choice=" Paste";
```

```
menus[1].item[4].choice=" Clear";
```

```
menus[1].last_choice=4;
```

```
menus[2].frame[0] = start;
```

```
menus[2].frame[1] = one_choice_width;
```

```
menus[2].frame[2] = start+one_part;
```

```
menus[2].frame[3] = one_choice_width+one_choice_width*3;
```

```
menus[2].row[0] = 6;
```

```
menus[2].row[1] = 26;
```

```
menus[2].row[2] = 46;
```

```
menus[2].row[3] = 66;
```

```
menus[2].row[4] = 86;
```

```
menus[2].col = start;
```

```
start+=one_part;
```

```
menus[2].item[0].choice=" Simulation ";
```

```
menus[2].item[1].choice=" Plot all link ";
```

```
menus[2].item[2].choice=" Plot some link ";
```

```
menus[2].item[3].choice=" All position ";
```

```
menus[2].item[4].choice=" ";
```

```
menus[2].last_choice=3;
```

```
menus[3].frame[0] = start;
```

```
menus[3].frame[1] = one_choice_width;
```

```
menus[3].frame[2] = start+one_part-1;
```

```
menus[3].frame[3] = one_choice_width+one_choice_width*2;
```

```
menus[3].row[0] = 6;
```

```
menus[3].row[1] = 26;
```

```
menus[3].row[2] = 46;
```

```
menus[3].row[3] = 66;
```

```
menus[3].row[4] = 86;
```

```
menus[3].col = start;
```

```
start+=one_part;
```

```
menus[3].item[0].choice=" Help ";
```

```
menus[3].item[1].choice=" Content ";
```

```
menus[3].item[2].choice=" Index";
```

```
menus[3].item[3].choice="";
```

```
menus[3].item[4].choice="";
```

```
menus[3].last_choice=2;
```

```
menus[4].frame[0] = start;
```

```
menus[4].frame[1] = one_choice_width;
```

```
menus[4].frame[2] = start+one_part;
```

```
menus[4].frame[3] = one_choice_width+one_choice_width*2;
```

```
menus[4].row[0] = 6;
```

```
menus[4].row[1] = 26;
```

```
menus[4].row[2] = 46;
```

```
menus[4].row[3] = 66;
```

```
menus[4].row[4] = 86;
```

```
menus[4].col = start;
start+=one_part;
```

```
menus[4].item[0].choice=" About ";
menus[4].item[1].choice=" Information ";
menus[4].item[2].choice=" Programer";
menus[4].item[3].choice="";
menus[4].item[4].choice="";
```

```
menus[4].last_choice=2;
```

```
}
```

```
void Display_Main_Menu(void)
```

```
{
```

```
int i;
```

```
Draw_Rectangle(0,0,Get_Max_X(),19);
```

```
for(i=0;i<NO_MENU; i++){
```

```
setcolor(0);
```

```
Out_Text_XY(menus[i].col,menus[i].row[0],menus[i].item[0].choice);
```

```
}
```

```
}
```

```
void Display_Menu(int menu_no)
```

```
{
```

```
int i;
```

```
Draw_Rectangle(menus[menu_no].frame[0],menus[menu_no].frame[1],
```

```
menus[menu_no].frame[2],menus[menu_no].frame[3]);
```

```
for(i=0;i<NO_CHOICE; i++)/* ..... */
```

```
        Out_Text_XY(menus[menu_no].col,menus[menu_no].row[i],
        menus[menu_no].item[i].choice);
    }
}
```

```
void Inverse(int menu_no,int choice_no)
{
    Draw_Fill_Rectangle(menus[menu_no].col+2,
        menus[menu_no].row[choice_no],
        menus[menu_no].col+124,
        menus[menu_no].row[choice_no]+10);
    setcolor(LIGHTBLUE);
    Out_Text_XY(menus[menu_no].col,menus[menu_no].row[choice_no],
        menus[menu_no].item[choice_no].choice);
    setcolor(RED);
}
```

```
void Normal(int menu_no,int choice_no)
{
    Erase_Fill_Rectangle(menus[menu_no].col+2,
        menus[menu_no].row[choice_no],
        menus[menu_no].col+124,
        menus[menu_no].row[choice_no]+10);

    Out_Text_XY(menus[menu_no].col,menus[menu_no].row[choice_no],
        menus[menu_no].item[choice_no].choice);
}
```

```
void Unselect_Menu(int menu_no,int choice_no)
{
    Normal(menu_no,choice_no);
    setcolor(0);
    Display_Menu(menu_no);
    setcolor(RED);
}

int Read_Key(int key)
{
    return bioskey(key);
}

void workscr()
{
    //nscr();

    setviewport(0,0,getmaxx(),getmaxy(),1);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setfillstyle(SOLID_FILL,LIGHTBLUE);
    bar3d(0,19,getmaxx(),getmaxy()*19/20,0,1);
    setcolor(15);
    rectangle(0,0,getmaxx(),getmaxy());
    Display_Main_Menu();
    setviewport(0,0,getmaxx(),getmaxy(),1);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setfillstyle(SOLID_FILL,LIGHTMAGENTA);
    bar3d(0,getmaxy()*19/20,getmaxx(),getmaxy(),0,1);
    setcolor(15);
    outtextxy(10,465,"F10-menu ALT-X to quit esc_to Unpopup Window");
}
```

```
}  
open_menu()  
{  
int i;  
setcolor(15);  
do{  
i=bioskey(0);  
switch(i){  
    case F10 : Select_Menu(0,1);  
                break;  
    case ALT_X : closegraph();exit(0);  
                break;  
    case ESCAPE : cleardevice();nscr();  
                open_menu();  
                break;  
};  
}while(1);  
}
```

```
void edit_file()  
{  
  
setcolor(15);  
  
outtextxy(50,55,"Input Filename...");  
  
gscanfxy(50,80,"%s",filename);
```

```
}  
void Select_Menu(int menu_no,int choice_no)  
{  
    int new_menu_no,  
        new_choice_no =0;  
    int i;  
  
    Display_Menu(menu_no);  
    Inverse(menu_no,choice_no);  
  
    for(;;) {  
        i = Read_Key(0);  
        switch(i) {  
            case LEFT_ARROW :if(menu_no == 0)  
                new_menu_no = NO_MENU-1;  
                else new_menu_no = menu_no-1;  
                Unselect_Menu(menu_no,choice_no);  
                Select_Menu(new_menu_no,1);  
                break;  
            case RIGHT_ARROW :if(menu_no == NO_MENU-1)  
                new_menu_no = 0;  
                else new_menu_no = menu_no+1;  
                Unselect_Menu(menu_no,choice_no);  
                Select_Menu(new_menu_no,1);  
                break;  
            case UP_ARROW :if(choice_no == 1)  
                new_choice_no =menus[menu_no].last_choice;  
                else new_choice_no = choice_no-1;  
                Normal(menu_no,choice_no);
```

```

        Inverse(menu_no,new_choice_no);
        choice_no = new_choice_no;
        break;

    case DOWN_ARROW :if(choice_no ==menus[menu_no].last_choice)
        new_choice_no=1;
        else new_choice_no = choice_no+1;
        Normal(menu_no,choice_no);
        Inverse(menu_no,new_choice_no);
        choice_no = new_choice_no;
        break;

    case RETURN : switch(menu_no){
        case 0 : switch(choice_no){
            case 1 : cleardevice();clrscr();system("edit data.dat");

/*spawnlp(P_WAIT,"c:\\qbasic.exe","qbasic.exe",file,NULL);*/
                init();nscr();open_menu();
                break;
            case 2 : cleardevice();clrscr();system("edit ");

/*spawnlp(P_WAIT,"c:\\qbasic.exe","qbasic.exe",file,NULL);*/
                init();nscr();open_menu();
                break;
            case 3 : open_menu();
                break;
            case 4 : workscr();closegraph();clrscr();
                spawnlp(P_WAIT,
"C:\command.com","c:\command.com",NULL);
                init();nscr();open_menu();
                break;
            case 5 : closegraph(); exit(0);

```

```
        break;
    };
    break;
case 1: switch(choice_no){
    case 1 :
        break;
    case 2 :
        break;
    case 3 :
        break;
    case 4 : break;
};
break;
case 2: switch(choice_no){
    case 1 : cleardevice();
        plot_all_link();nscr();open_menu();
        break;
    case 2 : cleardevice();
        plot_some_link();nscr();open_menu();
        break;
    case 3 : cleardevice();
        viewgoal();nscr();open_menu();
        break;
    case 4 : cleardevice();ginput();trans();
        gprintfxy(50,50,"%0.2f",Q[1]);
        gprintfxy(50,80,"%0.2f",Q[2]);
        gprintfxy(50,110,"%0.2f",Q[3]);
        gprintfxy(50,140,"%0.2f",Q[4]);
        gprintfxy(50,180,"%0.2f",Q[5]);
        gprintfxy(50,210,"%0.2f",Q[6]);
```

```
        nscr();open_menu();
        break;
    };
    break;
case 3: switch(choice_no){
    case 1 : hscr();getch();
        cleardevice();
        nscr();open_menu();
        break;
    case 2 :
        break;
    case 3 :
        break;
    };
    break;

case 4: switch(choice_no){
    case 1 :
        break;
    case 2 : pscr();getch();
        cleardevice();
        nscr();open_menu();
        break;
    };
    break;

case ESCAPE : exit(0);
    break;
}
```

```
}  
}
```

```
main()  
{  
    Initialize_Graphics_Mode();  
    screen();  
    delay(20);  
    init();  
    nscr();  
    Menu_Assignment();  
    Display_Main_Menu();  
    open_menu();  
    return(0);  
}
```

M2-D.CPP

M2-D.CPP

```
#include<graphics.h>
```

```
#include<math.h>
```

```
#include<dos.h>
```

```
#include<conio.h>
```

```
#include<stdio.h>
```

```
#include<ctype.h>
```

```
#include<tran.cpp>
```

```
#include<gtext.cpp>
```

```
#include<stdlib.h>
```

```
typedef struct{
```

```
    double x,y,z;
```

```
    }point;
```

```
point p[10];
```

```
double xw1,xwr,ywt,ywb,a,b,c,d;
```

```
int xv1,xvr,yvt,yvb,x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5;
```

```
int color;
```

```
extern POSITION P,p_org[7],Tt;
```

```
FILE *file;
```

```
char filename[50];
```

```
int num;
```

```
//open_menu();
```

```
void set_window(double xmin,double xmax,double ymin,double ymax)
```

```
{
```

```
xw1=xmin;xwr=xmax;
```

```
ywb=ymin;ywt=ymax;
```

```
}
```

```
void set_viewport(int xmin,int xmax,int ymin,int ymax)
```

```

{
xvl=xmin;xvr=xmax;
yvb=ymin;yvt=ymax;
a=(xvr-xvl)/(xwr-xwl);b=xvl-a*xwl;
c=(yvt-yvb)/(ywt-ywb);d=yvb-c*ywb;
rectangle(xvl,yvb,xvr,yvt);
}

//change world coordinate to screen coordinate
void WORLDtoPC(double xw,double yw,int &xpc,int &ypc)
{ int ypc_temp;
  xpc=(int)(a*xw+b+0.5);ypc_temp=(int)(c*yw+d+0.5);
  if(yvb>=240) ypc=479-(ypc_temp-yvb);
    else ypc=yvt-yvb-ypc_temp;
}

//draw xy plane
void xy_plane()
{
set_window(-150,150,-150,150);
set_viewport(0,319,0,239);
WORLDtoPC(p_org[0].x,p_org[0].y,x1,y1);
WORLDtoPC(p_org[2].x,p_org[2].y,x2,y2);
WORLDtoPC(p_org[3].x,p_org[3].y,x3,y3);
WORLDtoPC(p_org[6].x,p_org[6].y,x4,y4);
color=getcolor();
setcolor(3);
line(x1,y1,x1+30,y1);outtextxy(x1+30,y1,"x");
line(x1,y1,x1,y1-30);outtextxy(x1,y1-30,"y");
setcolor(14);
circle(x1,y1,8);
setcolor(2);

```

```

circle(x1,y1,5);setfillstyle(SOLID_FILL,2);floodfill(x1,y1,2);
setcolor(color);
line(x1,y1,x2,y2);
setcolor(13);
circle(x2,y2,5);setfillstyle(SOLID_FILL,13);floodfill(x2,y2,13);
setcolor(color);
line(x2,y2,x3,y3);
//setcolor(color);
setcolor(4);
circle(x3,y3,5);setfillstyle(SOLID_FILL,4);floodfill(x3,y3,4);
//setcolor(2);
setcolor(color);
line(x3,y3,x4,y4);
//setcolor(color);
setcolor(9);
circle(x4,y4,3);setfillstyle(SOLID_FILL,9);floodfill(x4,y4,9);
setcolor(color);
outtextxy(10,220,"xy plane");

}

//draw xz plane
void xz_plane()
{
set_window(-150,150,-50,150);
set_viewport(0,319,240,479);
WORLDtoPC(p_org[0].x,p_org[0].z,x1,z1);
WORLDtoPC(p_org[2].x,p_org[2].z,x2,z2);
WORLDtoPC(p_org[3].x,p_org[3].z,x3,z3);
WORLDtoPC(p_org[6].x,p_org[6].z,x4,z4);

```

```
color=getcolor();
setcolor(3);
line(x1,z1,x1+30,z1);outtextxy(x1+30,z1,"x");
line(x1,z1,x1,z1-30);outtextxy(x1,z1-30,"z");
setcolor(RED);
rectangle(x1-7,z1,x1+7,z1+20);

setcolor(2);
circle(x1,z1,5);setfillstyle(SOLID_FILL,2);floodfill(x1,z1,2);
setcolor(color);
line(x1,z1,x2,z2);
setcolor(13);
circle(x2,z2,5);setfillstyle(SOLID_FILL,13);floodfill(x2,z2,13);
setcolor(color);
line(x2,z2,x3,z3);

setcolor(4);
circle(x3,z3,5);setfillstyle(SOLID_FILL,4);floodfill(x3,z3,4);
//setcolor(2);
setcolor(color);
line(x3,z3,x4,z4);

setcolor(9);
circle(x4,z4,3);setfillstyle(SOLID_FILL,9);floodfill(x4,z4,9);
setcolor(color);
outtextxy(10,460,"xz plane");
}

//draw yz plane
void yz_plane()
```

```

{
set_window(-150,150,-50,150);
set_viewport(320,639,240,479);
WORLDtoPC(p_org[0].y,p_org[0].z,y1,z1);
WORLDtoPC(p_org[2].y,p_org[2].z,y2,z2);
WORLDtoPC(p_org[3].y,p_org[3].z,y3,z3);
WORLDtoPC(p_org[6].y,p_org[6].z,y4,z4);
color=getcolor();
setcolor(3);
line(y1,z1,y1+30,z1);outtextxy(y1+30,z1,"y");
line(y1,z1,y1,z1-30);outtextxy(y1,z1-30,"z");
setcolor(RED);
rectangle(y1-7,z1,y1+7,z1+20);
//circle(y1,z1,8);
setcolor(2);
circle(y1,z1,5);setfillstyle(SOLID_FILL,2);floodfill(y1,z1,2);
setcolor(color);
line(y1,z1,y2,z2);
setcolor(13);
circle(y2,z2,5);setfillstyle(SOLID_FILL,13);floodfill(y2,z2,13);
setcolor(color);
line(y2,z2,y3,z3);
setcolor(color);
setcolor(4);
circle(y3,z3,3);setfillstyle(SOLID_FILL,4);floodfill(y3,z3,4);
setcolor(2);
setcolor(color);
line(y3,z3,y4,z4);
setcolor(color);
setcolor(9);

```

```
circle(y4,z4,3);setfillstyle(SOLID_FILL,9);floodfill(y4,z4,9);
```

```
setcolor(color);
```

```
outtextxy(330,460,"yz plane");
```

```
}
```

```
void present()
```

```
{
```

```
xy_plane();delay(500);
```

```
xz_plane();delay(500);
```

```
yz_plane();delay(500);
```

```
delay(20);
```

```
}
```

```
//read file
```

```
infile()
```

```
{int i;
```

```
setcolor(15);
```

```
if ((file=fopen(filename,"r"))==NULL)
```

```
{
```

```
gprintfxy(330,40,"No file %s",filename);
```

```
getch();setcolor(15);
```

```
// setup_screen();
```

```
// open_menu();
```

```
}
```

```
fscanf(file,"%d",&num);
```

```
fscanf(file,"%lf%lf%lf",&Tt.x,&Tt.y,&Tt.z);
```

```
for (i=0;i<num;i++)
```

```
fscanf(file,"%lf%lf%lf",&p[i].x,&p[i].y,&p[i].z);
```

```
fclose(file);
return 0;
}

//read file
infile2()
{int i;
setcolor(15);
if ((file=fopen(filename,"r"))==NULL)
{
gprintfxy(20,15,"No file %s",filename);
getch();setcolor(15);
// setup_screen();
// open_menu();
}
fscanf(file,"%d",&num);
fscanf(file,"%lf%lf%lf",&Tt.x,&Tt.y,&Tt.z);
for (i=0;i<num;i++)
fscanf(file,"%lf%lf%lf",&p[i].x,&p[i].y,&p[i].z);
fclose(file);
return 0;
}

//read file
void ginput()
{ int i,j,height;
rectangle(0,0,getmaxx(),getmaxy());
set_window(-150,150,-150,150);
set_viewport(320,639,0,239);
setcolor(15);
gprintfxy(330,40,"Input file name:...");
```

```

height=textheight("1")+25;
gscanfxy(330,40+height,"%s",filename);
for (i=321;i<639;i++)
    for(j=38;j<100;j++)putpixel(i,j,0);
setcolor(15);
infile();
setcolor(15);
}
void drawpath(int k)
{
set_window(-150,150,-150,150);/*xy*/
set_viewport(0,319,0,239);
WORLDtoPC(p[k-1].x,p[k-1].y,x1,y1);
WORLDtoPC(p[k].x,p[k].y,x2,y2);
color=getcolor();
setcolor(10);
line(x1,y1,x2,y2);
setcolor(color);
delay(500);

set_window(-150,150,-50,150);/*xz*/
set_viewport(0,319,240,479);
WORLDtoPC(p[k-1].x,p[k-1].z,x1,z1);
WORLDtoPC(p[k].x,p[k].z,x2,z2);
color=getcolor();
setcolor(10);
line(x1,z1,x2,z2);
setcolor(color);
setcolor(color);
delay(500);
}

```

```
set_window(-150,150,-50,150);/*yz*/
set_viewport(320,639,240,479);
WORLDtoPC(p[k-1].y,p[k-1].z,y1,z1);
WORLDtoPC(p[k].y,p[k].z,y2,z2);
color=getcolor();
setcolor(10);
line(y1,z1,y2,z2);
setcolor(color);
delay(500);
}
```

```
plot_all_link()
```

```
{int k;
ginput();
for(k=0;k<num;k++)
{P.x=p[k].x; P.y=p[k].y; P.z=p[k].z;
trans();
present();
if((k>0)&&(k<num)) drawpath(k);
}
setcolor(15);
gprintfxy(330,40,"Press any key...");
getch();
}
```

```
plot_some_link()
```

```
{int k;
ginput();
```

```

for (k=0;k<num;k++)
{P.x=p[k].x; P.y=p[k].y; P.z=p[k].z;
trans();
if((k==num-1)||(k==0)) present();
if((k>0)&&(k<num)) drawpath(k);
}
setcolor(15);
gprintfxy(330,40,"Press any key...");
getch();

}

```

```

viewgoal()
{int i,j,height,ypos=15;
setcolor(15);
rectangle(0,0,getmaxx(),getmaxy());
gprintfxy(50,55,"Input file name :...");
gscanfxy(50,80,"%s",filename);
for (i=10;i<300;i++)
for (j=10;j<100;j++) putpixel(i,j,0);
setcolor(15);
infile2();
setcolor(15);
for (i=0;i<num;i++){
height=textheight("H");
ypos+=height*2;
if(ypos>=450)ypos=15;
gprintfxy(20,ypos,"%d. x%.2fy%.2fz%.2f",i+1,p[i].x,p[i].y,p[i].z);
}
setcolor(15);

```

```
gprintfxy(20,ypos+50,"Press any key...");
```

```
getch();
```

```
}
```

GTEXT.CPP

GTEXT.CPP

#include<bios.h>

#include<ctype.h>

#include<math.h>

#include<stdio.h>

#include<string.h>

#include<conio.h>

#include<graphics.h>

#include<stdarg.h>

#include <stdlib.h>

#define TEXTSIZE 8

#define BUFSIZE 140

#define TAB 0x0f09

#define BS 8

#define CR 13

#define ESC 27

#define BSPACE 0x0e08

#define ENTER 0x1c0d

#define ESCAPE 0x011b

#define DEL 0x5300

#define UP_ARROW 0x4800

#define DOWN_ARROW 0x5000

#define LEFT_ARROW 0x4B00

#define RIGHT_ARROW 0x4D00

```
#define F1 0x3b00
#define F10 0x4400
#define ALT_X 0x2d00

int grtextbkcolor=0;
int grtextcolor=13;
int settextcolor(int color)
{
    int oldcolor=grtextcolor;
    grtextcolor=color;
    return oldcolor;
}

int settextbkcolor(int color)
{
    int oldcolor=grtextbkcolor;
    grtextbkcolor=color;
    return oldcolor;
}

/*..*/
int gprintfxy(int xloc,int yloc,char *fmt,...)
{
    va_list argptr;
    char str[BUFSIZE];
    int cnt;
    struct fillsettingstype oldfill;
    char userfillpattern[8];
    va_start(argptr,fmt);
    cnt=vsprintf(str,fmt,argptr);
```

```

if(str[0]==NULL) return 0;
getfillsettings(&oldfill);
if(oldfill.pattern==USER_FILL) getfillpattern(userfillpattern);
setfillstyle(SOLID_FILL,grtextbkcolor);
setcolor(grtextcolor);
bar(xloc,yloc,xloc+textwidth(str),yloc+textheight("H")*5/4);
if (oldfill.pattern==USER_FILL)
    setfillpattern(userfillpattern,oldfill.color);
else
    setfillstyle(oldfill.pattern,oldfill.color);
outtextxy(xloc,yloc,str);
va_end(argptr);
return cnt;
}

/*...*/
int gprintf(char *fmt,...)
{
va_list argptr;
char str[BUFSIZE];
int cnt,xloc,yloc;
struct fillsettingstype oldfill;
char userfillpattern[8];
va_start(argptr,fmt);
cnt=vsprintf(str,fmt,argptr);
if(str[0]==NULL) return 0;
xloc=getx();yloc=gety();
getfillsettings(&oldfill);
if(oldfill.pattern==USER_FILL) getfillpattern(userfillpattern);
setfillstyle(SOLID_FILL,grtextbkcolor);

```

```

setcolor(grtextcolor);
bar(xloc,yloc,xloc+textwidth(str),yloc+textheight("H")*5/4);
if (oldfill.pattern==USER_FILL)
    setfillpattern(userfillpattern,oldfill.color);
else
    setfillstyle(oldfill.pattern,oldfill.color);
    outtext(str);
    va_end(argptr);
    return cnt;
}
int ggetche()
{
char ch;
ch=getch();
gprintf("%c",ch);
return ch;
}
int gputch(int c)
{
char buffer[2];
sprintf(buffer,"%c",c);
gprintf(buffer);
return c;
}
char *ggets(char *buffer)
{
int currlloc,maxchars,oldcolor,ch;
struct viewporttype view;
char charbuff[3];
buffer[0]='\0';

```

```

currloc=0;
getviewsettings(&view);
maxchars=(view.right-getx())/textwidth("M")-1;
if (maxchars<=0) return (NULL);
gprintfxy(getx(),gety(),"_");
while((ch=getche())!=CR && ch!=ESC) {
if (ch==BS) {
if (currloc>0) {
currloc--;
if (currloc<=maxchars) {
oldcolor=setttextcolor(grtextbkcolor);
sprintf(charbuff,"%c",buffer[currloc]);
gprintfxy(getx()-textwidth(charbuff),gety(),"%c_",buffer[currloc]);
setttextcolor(oldcolor);
moveto(getx()-textwidth(charbuff),gety());
}
}
}
else {
if(currloc<maxchars) {
oldcolor=setttextcolor(grtextbkcolor);
gprintfxy(getx(),gety()," ");
setttextcolor(oldcolor);
buffer[currloc]=ch;
gputch(ch);
currloc++;
}
else
putch(0x07);
}
}
}

```

```

    if (currloc < maxchars) gprintfxy(getx(), gety(), "_");
}

if (currloc <= maxchars) {
    oldcolor = settextcolor(grtextbkcolor);
    gprintfxy(getx(), gety(), "_");
    settextcolor(oldcolor);
}

if (ch == ESC) buffer[0] = '\0';
    else buffer[currloc] = '\0';
return buffer;
}

```

```

int gscanf(char *fmt, ...)
{
    va_list argptr;
    char str[BUFSIZE];
    int cnt;
    va_start(argptr, fmt);
    ggets(str);
    cnt = vsscanf(str, fmt, argptr);
    va_end(argptr);
    return cnt;
}

/* ... */

int gscanfxy(int xloc, int yloc, char *fmt, ...)
{
    va_list argptr;
    char str[BUFSIZE];
    int cnt, oldx, oldy;
    oldx = xloc;

```

```

oldy=yloc;
moveto(xloc,yloc);
va_start(argptr,fmt);
ggets(str);
cnt=vsscanf(str,fmt,argptr);
va_end(argptr);
moveto(oldx,oldy);
return cnt;
}

```

```

void drawcurs(int locx,int locy,unsigned char erase)
{
static int lastx=0,lasty=0;
if(erase) {
setcolor(grtextbkcolor);
moveto(lastx*TEXTSIZE,(lasty+1)*TEXTSIZE+1);
lineto(locx*TEXTSIZE+7,(locy+1)*TEXTSIZE+1);
lastx=locx;lasty=locy;
}
}

```

```

void displaystr(int locx,int locy,int dlen,char str[])
{
int i;
for (i=0;i<dlen && str[i] !='\0';i++)
gprintfxy((locx+i)*TEXTSIZE,locy*TEXTSIZE,"%c",str[i]);
for (;i<dlen;i++)
gprintfxy((locx+i)*TEXTSIZE,locy*TEXTSIZE," ");
}

```

GPOPUP.CPP

GPOPUP.CPP

```
#include<graphics.h>
```

```
#include<alloc.h>
```

```
#include<conio.h>
```

```
#include "gtext.cpp"
```

```
#define NUMWINDOWS 1
```

```
#define THREED 10
```

```
struct GRAPHICSWINDOW {
```

```
    int vleft,vtop,vright,vbottom;
```

```
    int fleft,ftop,fright,fbottom;
```

```
    int cpx,cpy;
```

```
    void *savearea1;
```

```
    void *savearea2;
```

```
    int drawcolor;
```

```
    int bt,bc,bft,bfc;
```

```
    int dc;
```

```
    int origbfc,origdc;
```

```
};
```

```
int wptr=0;
```

```
struct GRAPHICSWINDOW *wstack[NUMWINDOWS];
```

```
paintwindow(int left,int top,int right,int bottom,
```

```
            int bt,int bfc,int bft,int bc)
```

```
{
```

```
    int i;
```

```
    if (bt==THREED) {
```

```
        setlinestyle(SOLID_LINE,0,NORM_WIDTH);
```

```

for(i=0;i<2;i++) {
    setcolor(DARKGRAY);
    line(right-i,top+i,right-i,top-i);
    line(right+i,top-i,right-i,top-i);
    setcolor(WHITE);
    line(left+i,top+i,right-i,top+i);
    line(left+i,top+i,left+i,bottom-i);
}
setcolor(bfc);
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
setfillstyle(bft,bfc);
bar3d(left+2,top+2,right-2,bottom-2,0,0);
}
else {
    setcolor(bc);
    setlinestyle(bt,0,NORM_WIDTH);
    setfillstyle(bft,bfc);
    bar3d(left,top,right,bottom,0,0);
}
}

int savewindow(int left,int top,int right,int bottom)
{
    int halfpoint;
    halfpoint=(top+bottom)/2;
    wstack[wptr]->savearea1=malloc(imagesize(left,top,right,halfpoint));
    wstack[wptr]->savearea2=malloc(imagesize(left,halfpoint+1,right,bottom));
    if (wstack[wptr]->savearea1==NULL||wstack[wptr]->savearea2==NULL) return 0;
    getimage(left,top,right,halfpoint,wstack[wptr]->savearea1);
    getimage(left,halfpoint+1,right,bottom,wstack[wptr]->savearea2);
}

```

```

return 1;
}
/*...*/
int gpopup(int left,int top,int right,int bottom,int bordertype,
           int bordercolor,int backfill,int fillcolor,int drawcolor)
{
    struct viewporttype oldview;
    int oldx,oldy,savecolor;
    struct linesettingstype saveline;
    struct fillsettingstype savefill;
    if (wptr>=NUMWINDOWS) return 0;
    wstack[wptr]=(struct GRAPHICSWINDOW *)malloc(sizeof(struct
GRAPHICSWINDOW));
    if(wstack[wptr]==0) return 0;
    getviewsettings(&oldview);
    oldx=getx();
    oldy=gety();
    setviewport(0,0,getmaxx(),getmaxy(),1);
    getlinesettings(&saveline);
    savecolor=getcolor();
    getfillsettings(&savefill);
    if (savewindow(left,top,right,bottom)==0) {
        setviewport(oldview.left,oldview.top,oldview.right,oldview.bottom,1);
        moveto(oldx,oldy);
        free(wstack[wptr]);
        return 0;
    }
    wstack[wptr]->bc=bordercolor;
    wstack[wptr]->bt=bordertype;
    wstack[wptr]->bft=backfill;

```

```

wstack[wptr]->bfc=fillcolor;
wstack[wptr]->dc=drawcolor;
paintwindow(left,top,right,bottom,bordertype,fillcolor,backfill,bordercolor);
if(bordertype==THREED)
    setviewport(left+2,top+2,right-2,bottom-2,1);
else
    setviewport(left+1,top+1,right-1,bottom-1,1);
wstack[wptr]->fleft=left;
wstack[wptr]->ftop=top;
wstack[wptr]->fbottom=bottom;
wstack[wptr]->fright=right;

    wstack[wptr]->vleft=oldview.left;
    wstack[wptr]->vtop=oldview.top;
    wstack[wptr]->vbottom=oldview.bottom;
    wstack[wptr]->vright=oldview.right;
wstack[wptr]->cpx=oldx;
wstack[wptr]->cpy=oldy;
    wstack[wptr]->drawcolor=savecolor;
    wstack[wptr]->origbfc=setttextbkcolor(fillcolor);
    wstack[wptr]->origdc=setttextcolor(drawcolor);
setlinestyle(saveline.linestyle,saveline.upattern,saveline.thickness);
setcolor(savecolor);
setfillstyle(savefill.pattern,savefill.color);wptr++;
return 1;
}

/*...*/
int gunpop()
{

```

```

struct viewporttype currview;
struct GRAPHICSWINDOW*w;
if(wptr<=0) return 0;
wptr--;
w=wstack[wptr];

setviewport(w->fleft,w->ftop,w->fright,w->fbottom,1);
getviewsettings(&currview);
putimage(0,0,w->savearea1,COPY_PUT);
putimage(0,(currview.bottom-currview.top)/2+1,w->savearea2,COPY_PUT);
setviewport(w->vleft,w->vtop,w->vright,w->vbottom,1);
moveto(w->cpx,w->cpy);
setcolor(w->drawcolor);
settextbkcolor(w->origbfc);
settextcolor(w->origdc);
free(w->savearea1);
free(w->savearea2);
free(wstack[wptr]);
return 1;
}

void unpopallwindows()
{
while(gunpop());
}

int printfxy(int xloc,int yloc,char *fmt,...)
{
int oldbc=settextbkcolor(wstack[wptr-1]->bfc);
int olddc=settextcolor(wstack[wptr-1]->dc);
int cnt=gprintfxy(xloc,yloc,fmt);
settextbkcolor(oldbc);

```

```
settextcolor(olddc);
```

```
return cnt;
```

```
}
```

TRAN.CPP

TRAN.CPP

```
#include <string.h>
```

```
#include<graphics.h>
```

```
#include<math.h>
```

```
#include<dos.h>
```

```
#include<conio.h>
```

```
#include<stdio.h>
```

```
#include<ctype.h>
```

```
#define Pi 3.141592654
```

```
#define d2 7.2 /*cm*/
```

```
#define a2 22 /*cm*/
```

```
#define a3 20 /*cm*/
```

```
#define d3 6 /*cm*/
```

```
#define d6 15 /*cm*/
```

```
typedef struct {
```

```
double x,y,z;
```

```
} POSITION;
```

```
/* Operating Function.....*/
```

```
double sqr(double);
```

```
double Q4,Q5,Q6;
```

```
double Q[7],T[2][5][5];
```

```
POSITION P,Tt,p_org[7];
```

```
void init();
```

```
//void screen();
```

```
void nscr();
```

```
open_menu();
```

```
//calculate all point in world coordinate
```

```
void mcs()
```

```
{
```

```
p_org[0].x=0.0; p_org[0].y=0.0; p_org[0].z=0.0;
```

```
p_org[1].x=0.0; p_org[1].y=0.0; p_org[0].z=0.0;
```

```
p_org[2].x=a2*cos(Q[1])*cos(Q[2])+d2*sin(Q[1]);
```

```
p_org[2].y=a2*sin(Q[1])*cos(Q[2])-d2*cos(Q[1]);
```

```
p_org[2].z=a2*sin(Q[2]);
```

```
p_org[3].x=P.x-cos(Q[4])*sin(Q[5])*d6;
```

```
p_org[3].y=P.y-sin(Q[4])*sin(Q[5])*d6;
```

```
p_org[3].z=P.z-cos(Q[5])*d6;
```

```
p_org[4].x=cos(Q[1])*(a3*cos(Q[2]+Q[3])+a2*cos(Q[2]))+sin(Q[1])*(d3+d2);
```

```
p_org[4].y=sin(Q[1])*(a3*cos(Q[2]+Q[3])+a2*cos(Q[2]))-cos(Q[1])*(d3+d2);
```

```
p_org[4].z=a3*sin(Q[2]+Q[3])+a2*sin(Q[2]);
```

```
p_org[5].x=cos(Q[1])*(a3*cos(Q[2]+Q[3])+a2*cos(Q[2]))+sin(Q[1])*(d3+d2);
```

```
p_org[5].y=sin(Q[1])*(a3*cos(Q[2]+Q[3])+a2*cos(Q[2]))-cos(Q[1])*(d3+d2);
```

```
p_org[5].z=a3*sin(Q[2]+Q[3])+a2*sin(Q[2]);
```

```
p_org[6].x=P.x;
```

```
p_org[6].y=P.y;
```

```
p_org[6].z=P.z;
```

```
}
```

```
double sqr(double x)
{
    return pow(x,2.00);
}
```

```
unsigned short int inverse_link_kinematic(int v){
double B1,B2,C3,C2,C5,Q23;
double Pwx,Pwy,Pwz;
Pwx=T[1][1][4]-(T[1][1][3]*d6);
Pwy=T[1][2][4]-(T[1][2][3]*d6);
Pwz=T[1][3][4]-(T[1][3][3]*d6);

B1=(-1*Pwx)+sqrt((sqr(Pwx)+sqr(Pwy)-sqr(d2+d3))));
B2=(Pwy-(d2+d3));

Q[1]=2*atan2((B1),(B2));

C3=(sqr(Pwx)+sqr(Pwy)+sqr(Pwz)-sqr(a2)-sqr(a3)-sqr(d2+d3))/(2*a3*a2);
if(sqr(C3)>1.0){
    closegraph();
    printf("\n\n\t\n angle1 error in data file input !!!\n\n\n");
    printf("\t\tPlease check your data again.\n\n\n");
    printf("\t\tYour data may be out of range of robot to operate.");
    getch();
    init();
    nscr();
    open_menu();
}
```

```

}
```

```

B1=sqrt((-1+sqr(B2)));
```

```

B2=(C3);
```

```

Q[3]=atan2(B1,B2);
```

```

C2=(a2+a3*cos(Q[3]))*sqrt((sqr(Pwx)+sqr(Pwy)-sqr(d3+d2)))+
```

```

(a3*sin(Q[3])*Pwz)/((sqr(Pwx)+sqr(Pwy)+sqr(Pwz)-sqr(d2+d3)));
```

```

if(sqr(C2)>1.0){
```

```

    closegraph();
```

```

    printf("\n\n\n Error0 in data file input !!!\n\n\n");
```

```

    printf("\tPlease check your data again.\n\n\n");
```

```

    printf("\tYour data may be out of range of robot to operate.");
```

```

    getch();
```

```

    init();
```

```

    screen();
```

```

    open_menu();
```

```

}
```

```

B1=sqrt((1-sqr(B2)));
```

```

B2=(C2);
```

```

Q[2]=atan2(B1,B2);
```

```

B1=((T[v][3][4])-Pwz);
```

```

B2=sqrt(sqr((T[v][1][4])-Pwx)+sqr((T[v][2][4])-Pwy));
```

```

Q[5]=atan2(B1,B2);
```

```

B1=abs((cos(Q[4])*(sin(Q[1])*(T[v][1][2])-cos(Q[1])*(T[v][2][2]))-
(sin(Q[4])*cos(Q[2]+Q[3])*(cos(Q[1])*(T[v][1][2])+sin(Q[1])*(T[v][2][2]))-
(sin(Q[4])*sin(Q[2]+Q[3])*(T[v][3][2])));

```

```

B2=abs((cos(Q[4])*(sin(Q[1])*(T[v][1][1])-cos(Q[1])*(T[v][2][1]))-
(sin(Q[4])*cos(Q[2]+Q[3])*(cos(Q[1])*(T[v][1][1])+sin(Q[1])*(T[v][2][1]))-
(sin(Q[4])*sin(Q[2]+Q[3])*(T[v][3][1])));

```

```

Q[6]=atan2(B1,B2);

```

```

B1=abs((T[v][2][4])-Pwy);

```

```

B2=abs((T[v][1][4])-Pwx);

```

```

Q[4]=atan2(B1,B2);

```

```

return 0;

```

```

}

```

```

unsigned short int tool_input(void) {

```

```

if(Tt.x==0.0 && Tt.y==0.0)

```

```

{Q4=0.0; Q5=0.0;}

```

```

else{

```

```

    Q5=acos(Tt.z/sqrt(sqrt(Tt.x)+sqrt(Tt.y)+sqrt(Tt.z)));

```

```

    Q4=acos(Tt.x/(sqrt(sqrt(Tt.x)+sqrt(Tt.y))));

```

```

}

```

```

Q6=Q6*Pi/180.0;

```

```

if(((sqrt(a2-a3-d6)>=(sqrt(P.x)+sqrt(P.y)+sqrt(P.z))))

```

```

        &&(sqr(a2+a3+d6)<=(sqr(P.x)+sqr(P.y)+sqr(P.z))))
    {

        closegraph();
        printf("\n\n\t\tError2 indata file input !!!\n\n\n");
        printf("\t\tPlease check your data again.\n\n\n");
        printf("\tyou data maybe out of range of robot to operate.");
        getch();
        init();
        nscr();
        open_menu();

    }

    return 0;

}

```

```

unsigned short int forward_kinematic(void)

```

```

{
    T[0][1][1]= cos(Q4)*cos(Q5)*cos(Q6)-sin(Q4)*sin(Q6);
    T[0][2][1]= sin(Q4)*cos(Q5)*cos(Q6)+cos(Q4)*sin(Q6);
    T[0][3][1]= -sin(Q5)*cos(Q6);
    T[0][4][1]= 0.0;

    /*.....*/

    T[0][1][2]= -cos(Q4)*cos(Q5)*sin(Q6)-sin(Q4)*cos(Q6);
    T[0][2][2]= -sin(Q4)*cos(Q5)*sin(Q6)+cos(Q4)*cos(Q6);

```

```

T[0][3][2]= sin(Q5)*sin(Q6);
T[0][4][2]= 0.0;

/*.....*/

T[0][1][3]= cos(Q4)*sin(Q5);
T[0][2][3]= sin(Q4)*sin(Q5);
T[0][3][3]= cos(Q5);
T[0][4][3]= 0.0;

/*.....*/

T[0][1][4]=P.x;
T[0][2][4]= P.y;
T[0][3][4]= P.z;
T[0][4][4]= 1.0;

/*.....*/

return 0;
}

int forward(void)

{

T[1][1][1]= cos(Q[1])*(cos(Q[2]+Q[3])*(cos(Q[4])*cos(Q[5])*cos(Q[6])-
sin(Q[4])*sin(Q[6]))-sin(Q[2]+Q[3])*sin(Q[5])*cos(Q[6]))
+sin(Q[1])*(sin(Q[4])*cos(Q[5])*cos(Q[6])+cos(Q[4])*sin(Q[6]));

```

$$\begin{aligned}
T[1][2][1] &= \sin(Q[1]) * (\cos(Q[2]+Q[3]) * (\cos(Q[4]) * \cos(Q[5]) * \cos(Q[6]) - \\
&\sin(Q[4]) * \sin(Q[6])) - \sin(Q[2]+Q[3]) * \sin(Q[5]) * \cos(Q[6])) \\
&\quad - \cos(Q[1]) * (\sin(Q[4]) * \cos(Q[5]) * \cos(Q[6]) + \cos(Q[4]) * \sin(Q[6])); \\
T[1][3][1] &= \sin(Q[2]+Q[3]) * (\cos(Q[4]) * \cos(Q[5]) * \cos(Q[6]) - \\
&\sin(Q[4]) * \sin(Q[6])) + \cos(Q[2]+Q[3]) * \sin(Q[5]) * \cos(Q[6]); \\
T[1][4][1] &= 0;
\end{aligned}$$

$$\begin{aligned}
T[1][1][2] &= \cos(Q[1]) * (- \\
&1 * \cos(Q[2]+Q[3]) * (\cos(Q[4]) * \cos(Q[5]) * \cos(Q[6]) + \sin(Q[4]) * \cos(Q[6])) + \sin(Q[2]+Q[3]) \\
&* \sin(Q[5]) * \sin(Q[6])) \\
&\quad + \sin(Q[1]) * (-1 * \sin(Q[4]) * \cos(Q[5]) * \sin(Q[6]) + \cos(Q[4]) * \cos(Q[6]));
\end{aligned}$$

$$\begin{aligned}
T[1][2][2] &= \sin(Q[1]) * (- \\
&1 * \cos(Q[2]+Q[3]) * (\cos(Q[4]) * \cos(Q[5]) * \cos(Q[6]) + \sin(Q[4]) * \cos(Q[6])) + \sin(Q[2]+Q[3]) \\
&* \sin(Q[5]) * \sin(Q[6])) \\
&\quad - \cos(Q[1]) * (-1 * \sin(Q[4]) * \cos(Q[5]) * \sin(Q[6]) + \cos(Q[4]) * \cos(Q[6]));
\end{aligned}$$

$$\begin{aligned}
T[1][3][2] &= (-1) * \sin(Q[2]+Q[3]) * (\cos(Q[4]) * \cos(Q[5]) * \sin(Q[6]) + \sin(Q[4]) * \sin(Q[6])) - \\
&\cos(Q[2]+Q[3]) * \sin(Q[5]) * \sin(Q[6]); \\
T[1][4][2] &= 0;
\end{aligned}$$

$$\begin{aligned}
T[1][1][3] &= \cos(Q[1]) * (\cos(Q[2]+Q[3]) * \cos(Q[4]) * \sin(Q[5]) + \sin(Q[2]+Q[3]) * \cos(Q[5])) \\
&\quad + \sin(Q[1]) * \sin(Q[4]) * \sin(Q[5]);
\end{aligned}$$

$$\begin{aligned}
T[1][2][3] &= \sin(Q[1]) * (\cos(Q[2]+Q[3]) * \cos(Q[4]) * \sin(Q[5]) + \sin(Q[2]+Q[3]) * \cos(Q[5])) \\
&\quad - \cos(Q[1]) * \sin(Q[4]) * \sin(Q[5]);
\end{aligned}$$

$$T[1][3][3] = \sin(Q[2]+Q[3]) * \cos(Q[4]) * \sin(Q[5]) - \cos(Q[2]+Q[3]) * \cos(Q[5]);$$

```
T[1][4][3]= 0;

T[1][1][4]= P.x;
T[1][2][4]= P.y;
T[1][3][4]= P.z;
T[1][4][4]= 1;
/* double C3;
C3=(sqr(T[1][1][4])+sqr(T[1][2][4])+sqr(T[1][3][4])-sqr(a2)-sqr(a3)-
sqr(d2+d3))/(2*a3*a2);
printf("\n %d :=%5.5f",C3);*/
return 0;
}
```

```
void trans(void)
```

```
{
    tool_input();
    forward();
    inverse_link_kinematic(0);
    mcs();
}
```

HSCREEN.CPP

HSCREEN.CPP

```
#include<graphics.h>
#include<math.h>
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<dos.h>
void hscr()
{
    setcolor(15);
    if(getmaxcolor()==1)setfillstyle(SOLID_FILL,0);
    else setfillstyle(SOLID_FILL,LIGHTBLUE);
    bar3d(0,0,getmaxx(),getmaxy()*19/20,10,1);
    setfillstyle(SOLID_FILL,RED);
    bar3d(getmaxx()*1/8,getmaxy()*1/8,getmaxx()*7/8,getmaxy()*3/4,0,1);
    settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setcolor(getmaxcolor());
    setviewport(getmaxx()*1/8,getmaxy()*1/8,getmaxx()*7/8,getmaxy()*3/4,1);
    outtextxy(70,45," This SIMULATION PROGRAM use for ");
    outtextxy(65,70," KAWASAKI UNTIMATE MACHINE 260 R13 ");
    outtextxy(25,120," Plot all link - use for show all link in all position ");
    outtextxy(25,145," Plot some link - use for show first & last position ");
    outtextxy(25,170," All position - use for display position to go ");
    outtextxy(170,250,"Press any key to quit ...");
    setviewport(0,0,getmaxx(),getmaxy(),1);
    settxtstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setfillstyle(SOLID_FILL,LIGHTMAGENTA);
    bar3d(0,getmaxy()*19/20,getmaxx(),getmaxy(),0,1);
}
```

PSCR.CPP

PSCR.CPP

```
#include<graphics.h>
#include<math.h>
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<dos.h>

void pscr()
{
    setcolor(15);
    if(getmaxcolor()==1)setfillstyle(SOLID_FILL,0);
    else setfillstyle(SOLID_FILL,LIGHTBLUE);
    bar3d(0,0,getmaxx(),getmaxy()*19/20,10,1);
    setfillstyle(SOLID_FILL,RED);
    bar3d(getmaxx()*1/8,getmaxy()*1/8,getmaxx()*7/8,getmaxy()*3/4,0,1);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setcolor(getmaxcolor());
    setviewport(getmaxx()*1/8,getmaxy()*1/8,getmaxx()*7/8,getmaxy()*3/4,1);
    outtextxy(150,45," This SIMULATION PROGRAM ");
    outtextxy(185,70," written by ");
    outtextxy(155,120," WORAPONG THONGTAKORN ");
    outtextxy(155,145," WIPUTYA SEATONG ");
    outtextxy(155,170," INDUSTRIAL ENGINEERING");
    outtextxy(25,220,"FACULTY OF ENGINEERING ");
    outtextxy(190,250," NARESUAN UNIVERSITY");
    setviewport(0,0,getmaxx(),getmaxy(),1);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
    setfillstyle(SOLID_FILL,LIGHTMAGENTA);
    bar3d(0,getmaxy()*19/20,getmaxx(),getmaxy(),0,1);
}
```

ภาคผนวก ข
วิธีการใช้โปรแกรมการสร้างภาพจำลองการเคลื่อนที่ของหุ่นยนต์
KAWASAKI UNIMATE MACHINE

วิธีการใช้โปรแกรม ROBOT.EXE

1. เรียกโปรแกรมที่ prompt C:\> robot
2. เมื่อเข้าโปรแกรมแล้วจะเห็น Comment แสดงขึ้นมา หลังจากนั้นให้กดปุ่มใด ๆ เพื่อเริ่มใช้งานเมนูในโปรแกรม
3. เรียกใช้เมนูโดยการกด F10 และเลือกเมนูโดยใช้ลูกศร up , down , left ,right ไปที่เมนู กด ENTER เมื่อเลือกเมนูใช้งานได้แล้ว
4. menu ที่มีใน โปรแกรมจะประกอบด้วย

New จะเป็นเมนูที่ใช้สร้าง Data File (*.dat) โดยจะสร้างด้วยโปรแกรม Editor ของ Dos
Open เป็นเมนูที่ใช้แก้ไข Data File โดยทำการแก้ไขด้วยโปรแกรม Editor ของ Dos เช่น เดียวกัน

Dosshell เป็นเมนูที่ใช้ติดต่อกับระบบปฏิบัติการของ Dos เพื่ออาจจะ load โปรแกรม Data File จาก Drive A: หรือ directory อื่น

Quit เป็นเมนูที่ใช้เมื่อต้องการจบการทำงานของโปรแกรม

View all link เป็นเมนูที่จำลองการเคลื่อนที่ของหุ่นยนต์ โดยแสดง link ทุก link เมื่อ Robot เคลื่อนตามแต่ละ position โดยผู้ใช้จะต้องใส่ชื่อ Data File ของ Robot เข้าไปเมื่อมีการเรียกใช้เมนูนี้

View some link เป็นเมนูที่จำลองการเคลื่อนที่ของหุ่นยนต์ โดยแสดง link บาง link คือ เฉพาะ link ของ position แรกกับ position สุดท้าย โดยมีรายละเอียดการทำงานเหมือนกับ View all link คือผู้ใช้จะต้องใส่ชื่อ Data File

All position เป็นเมนูที่ใช้ดู position ใน Data File ซึ่งผู้ใช้จะต้องใส่ชื่อ Data File ที่ต้องการเปิดดูข้อมูลด้วย

Help เป็นเมนูแสดงรายละเอียดการใช้งานของแต่ละเมนู

Programmer เป็นเมนูแสดงชื่อของผู้เขียน โปรแกรม

5. รูปแบบของ Data File ซึ่งสามารถใช้โปรแกรม Editor ได้ ๆเขียนขึ้นมาได้

3→คือจำนวน position

1 2 1 → คือ vector แสดง Rotation ปลาย link ของ Robot

10 20 30 → คือ position ที่ 1

15 30 25 → คือ position ที่ 2

-10 25 -35 → คือ position ที่ 3

<*** End Of File ***> → คือการแสดงการจุดสิ้นสุดของ Data File