

รายการอ้างอิง

1. ฟุจิอิ ซาโตะ. การทดสอบแบบไม่ทำลาย (Non destructive testing). แปลโดย ปรีทรศน์ พันธุ์บรรยงค์ และคณะ. กรุงเทพมหานคร:บริษัท ที.พี.พรินท์ จำกัด, พ.ศ. 2539.
2. Lamarsh J.R. Introduction to nuclear Engineering. New York:Addison-Wesley Publishing Co., 1975 .
3. Rafael C. Gonzalez, Richard E. Woods. Digital image processing. the United States of America: Addison-Wesley Publishing Company, 1992.
4. วิรุฬห์ ตรีศฤง. การออกแบบและสร้างเกราะกำบังรังสีสำหรับคั่นกำเนิดรังสีแกมมาซีเซียม-137 ความแรง 5 คูรี เพื่อใช้ในงานถ่ายภาพรังสี. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2533.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บรรณานุกรม

ภาษาไทย

- เจนวิทย์ เหลืองอร่าม. การใช้ Turbo C++ เขียนโปรแกรมภาษา C. กรุงเทพมหานคร: ซีเอ็ดดูเคชั่น, พ.ศ. 2538.
- นเรศร์ จันทน์ขาว. การตรวจสอบโดยวิธีการถ่ายภาพด้วยรังสี. เอกสารประกอบการอบรมทางวิชาการ ชมรมการตรวจสอบโดยไม่ทำลาย, พ.ศ. 2528.
- นุชบา แซ่ลิ้ม. การพัฒนาระบบสร้างภาพสองมิติด้วยไมโครคอมพิวเตอร์จากการส่งผ่านรังสีแกมมา. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2535.
- ฟูจิอิ ซาโตะ. การทดสอบแบบไม่ทำลาย (Non destructive testing). แปลโดย ปรีทรรคนันท์ พันธุบรรยงก์ และคณะ. กรุงเทพมหานคร: บริษัท ที.พี.พรินท์ จำกัด, พ.ศ. 2539.
- วิรุฬห์ ดริสกุล. การออกแบบและสร้างเกราะกำบังรังสีสำหรับต้นกำเนิดรังสีแกมมาซีเซียม-137 ความแรง 5 คูรี เพื่อใช้ในงานถ่ายภาพรังสี. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2533.
- มงคล วรรณประภา. การพัฒนาระบบสแกนด้วยรังสีแกมมาเพื่อคำนวณสร้างภาพโทโมกราฟีของเสาคอนกรีตเสริมเหล็ก. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, พ.ศ. 2536.
- สุวัฒน์ เลิศถ้ำ. เรียนการเขียนโปรแกรมด้านกราฟิกด้วยเทอร์โบปาสคาล เวอร์ชัน 6.0. กรุงเทพมหานคร: บริษัท พี.โอ.เอ็น.วาย จำกัด.

ภาษาอังกฤษ

- International Atomic Energy Agency. Industrial radiography. Vienna: the IAEA in Austria., 1992. (Mimeographed).
- John R. Lamarsh. Introduction to nuclear engineering. 2nd ed. London: Addison-Wesley Publishing Company, 1993.
- Rafael C. Gonzalez, Richard E. Woods. Digital image processing. the United States of America: Addison-Wesley Publishing Company, 1992.
- Thomas S. Curry III, James E. Dowdey, Robert C. Murry. Christensen's introduction to the physics of diagnostic radiology. Philadelphia: Lea & Febiger, 1984.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ต้นกำเนิดรังสีแกมมาซีเซียม-137

Caesium-137
Gamma sources

Sources contain the radionuclide as compressed pellets of caesium chloride. Encapsulation is in welded stainless steel.

Nominal equivalent activity*	Nominal air kerma rate at 1 metre mSv/hr	Capsule type	Double encapsulation Code
5	14.4	X.60/2	CXC.0024
10	28.8	X.00/1	CDC.0025
30	86.4	X.00/1	CDC.001

*Tolerance -0, +25%
for definition of equivalent activity, see page 55

Availability: on request

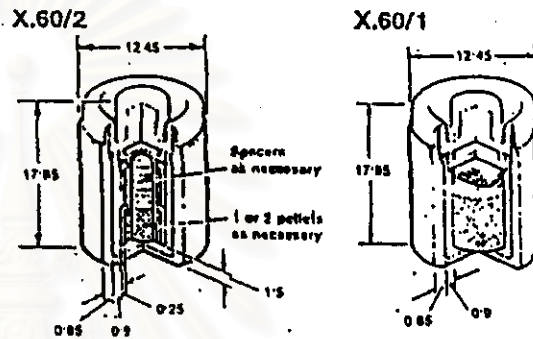
Recommended working life: 15 years

Quality Control

Wipe test A

Bubble test D

Helium leak test I



Safety performance testing

Capsule	ISO classification	IAEA special form
X00/1	EG3505	GB/190/S
X00/2	EG3035	GB/190/S

¹³⁷Cs (30.0 2 yr)
⁵⁵ Cs (30.0 2 yr)
 Mode: β-
 E_{max} 505.1 keV
 Sp. A 37.0 Ci/g
 Prod. fission

Gamma Emission (¹³⁷Cs)
 (γ)-560.2 keV

γ-ray	E _γ (keV)	I _γ (%)
γ ₁	560.2	0.00100
γ ₂	4.51	0.00000
γ ₃	0.463	0.00000
γ ₄	0.554	0.00000
γ ₅	0.620	0.00000
γ ₆	0.617	0.00000
γ ₇	2.190	0.00000
γ ₈	26.552	0.00000
γ ₉	39.420	0.00000
γ ₁₀	60.400	0.00000

(10,000% uncertainty)
 * with ¹³⁷Mo (2,552 min) in equilibrium

Continuous Radiation (¹³⁷Cs)
 (β)-182 keV; (β⁻)-0.134 keV

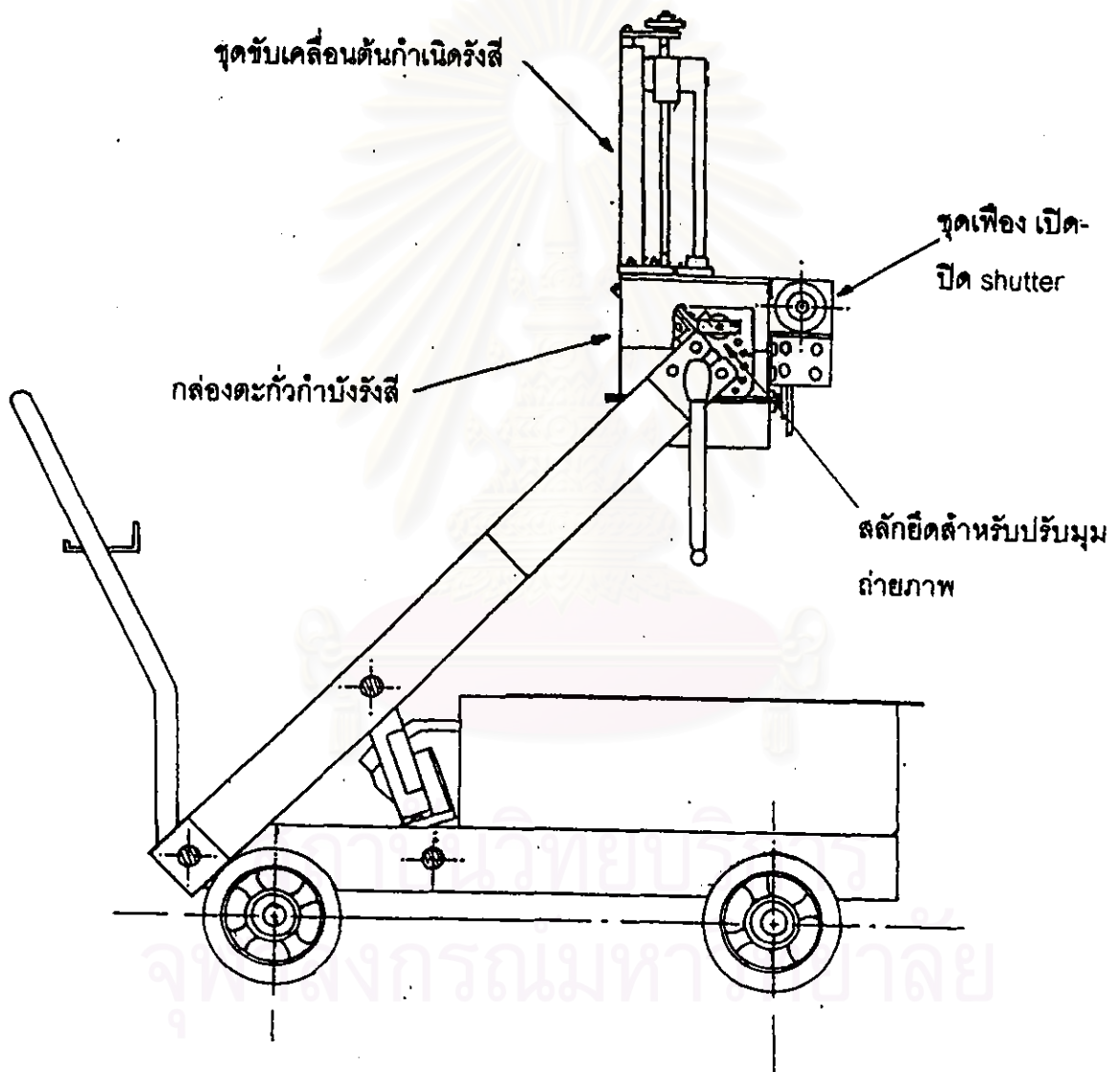
E _{max} (keV)	I (%)	β (%)
0-182	0.170	3.42
18	0.0049	0.10
10-20	0.000	0.01
10	0.000	0.00
70-90	1.05	6.5
90	0.0124	0.012
10-100	0.2	1.0
10	0.011	0.014
100-200	0.2	0.2
10	0.043	0.020
200-300	0.0	0.2
10	0.0019	0.0023
300-400	0.0	0.0
10	0.0007	0.0008
400-500	0.0	0.0
10	0.0001	0.0001

Atomic Electrons (¹³⁷Cs)
 (e⁻)-61.9 keV

E _{max} (keV)	I (%)	α (%)
1	0.199	3.8
6	0.107	2.0
23	0.0139	0.042
16	0.001	0.017
27	0.076	0.76
30	0.0106	0.010
31	0.007	0.010
28	0.0101	0.010
33	0.0071	0.010
34	0.0017	0.0010
674	0.0	7.91
616	0.16	1.44
640	1.39	0.241
621	0.418	0.130
641	0.0110	0.00421

ภาคผนวก ข

ลักษณะของชุดกำกับและบังคับลำรังสีของต้นกำเนิดรังสีแกมมาซีเซียม -137



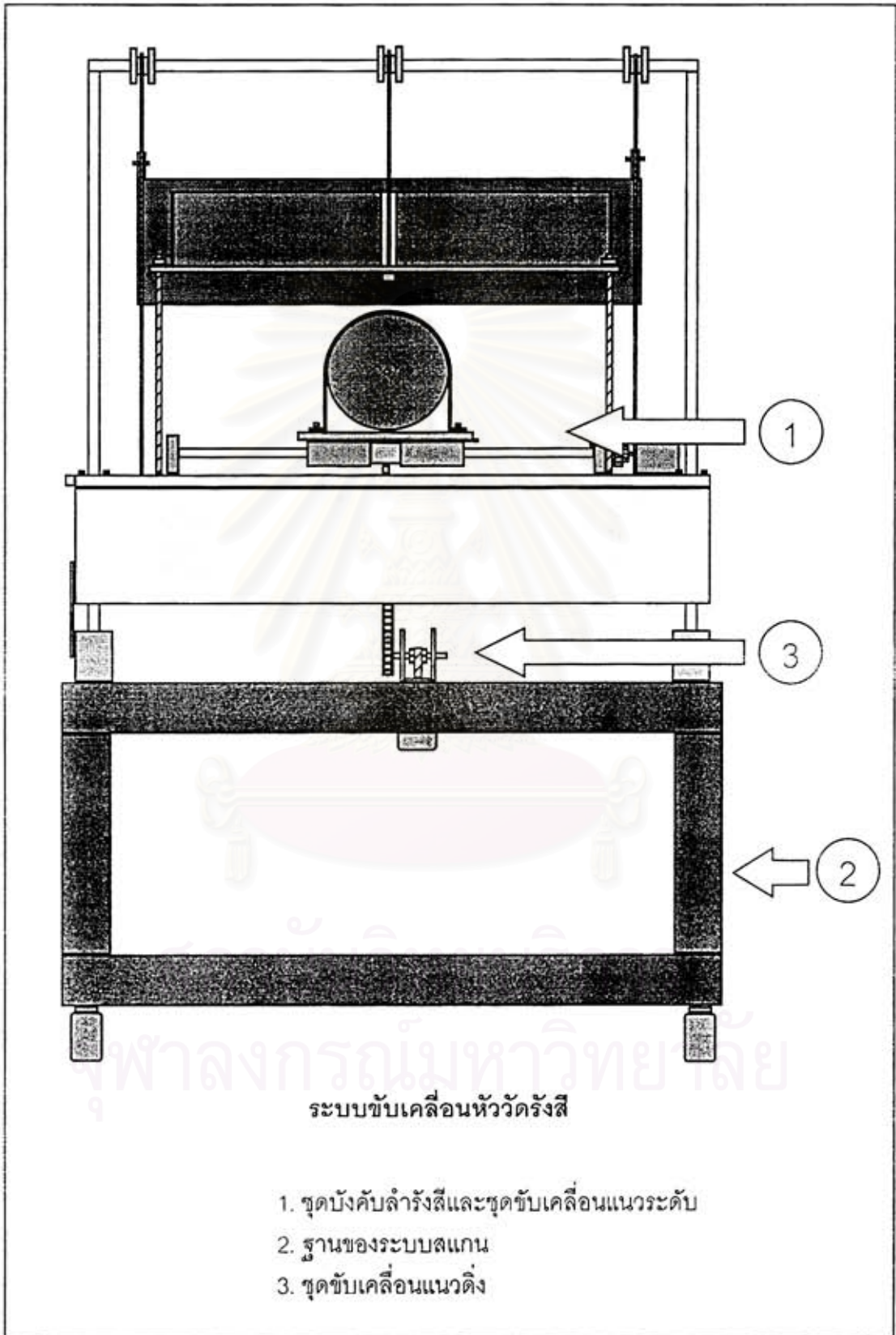
ภาคผนวก ก

ระบบขับเคลื่อนทางกล

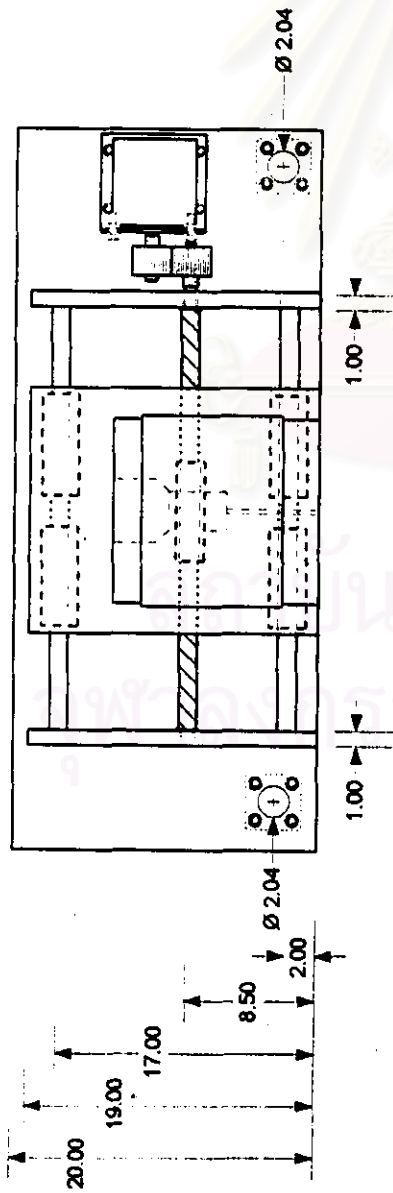
1. ระบบขับเคลื่อนหัววัดรังสี
2. ฐานของระบบและชุดขับเคลื่อนในแนวตั้ง
3. ชุดบังคับตัวรังสีและชุดขับเคลื่อนในแนวระดับ



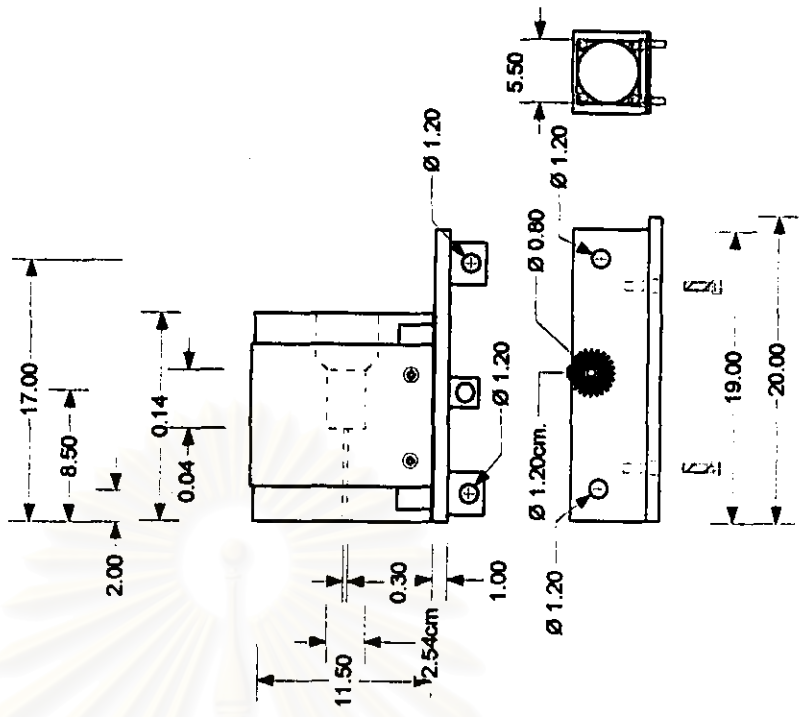
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



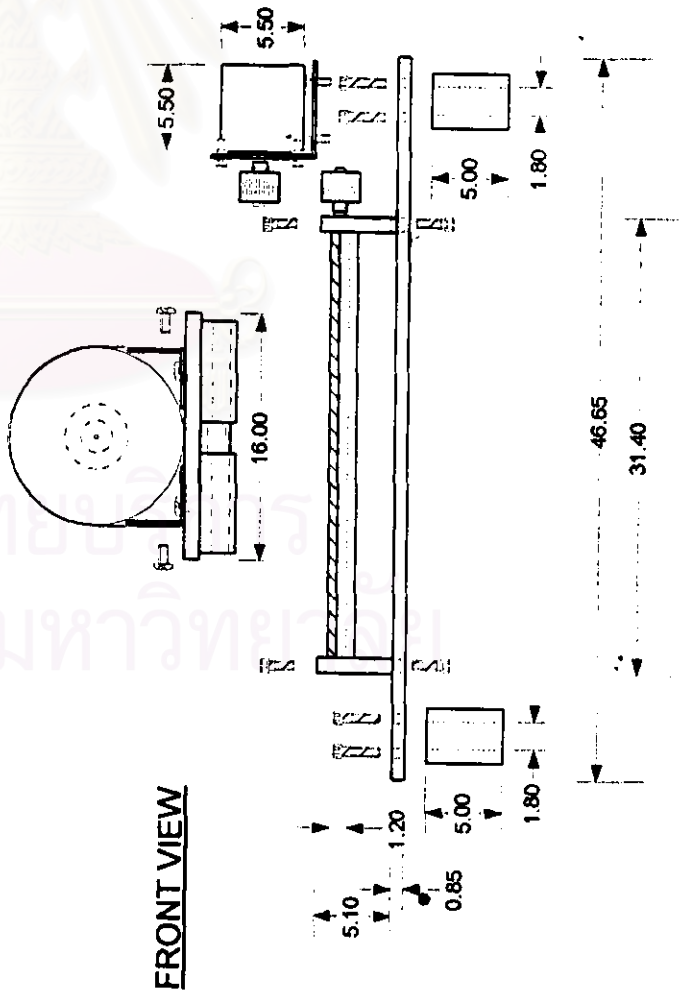
TOP VIEW



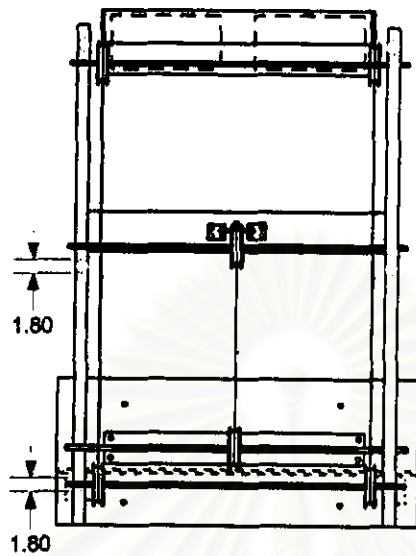
1 RATIO 1:5



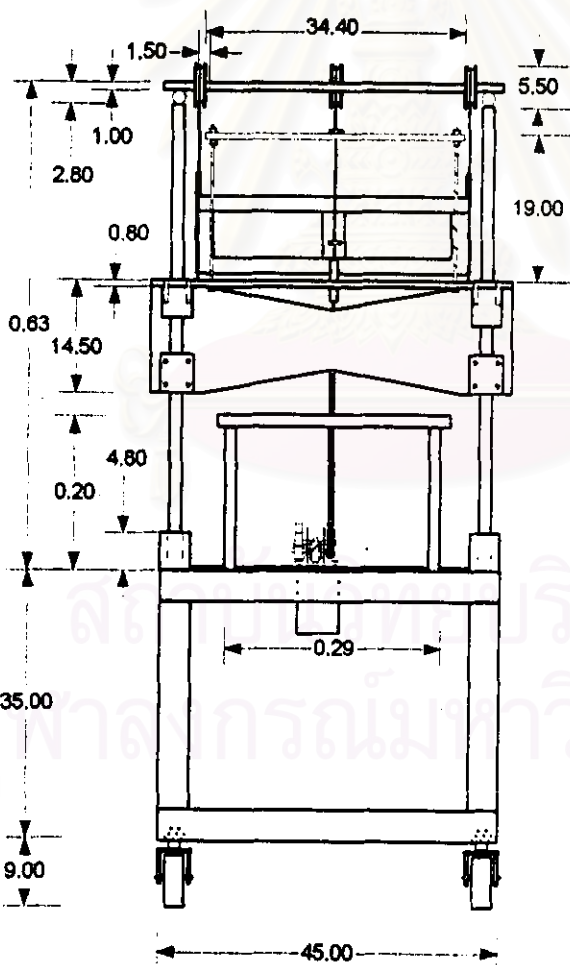
SIDE VIEW



FRONT VIEW



TOP VIEW



FRONT VIEW

2&3

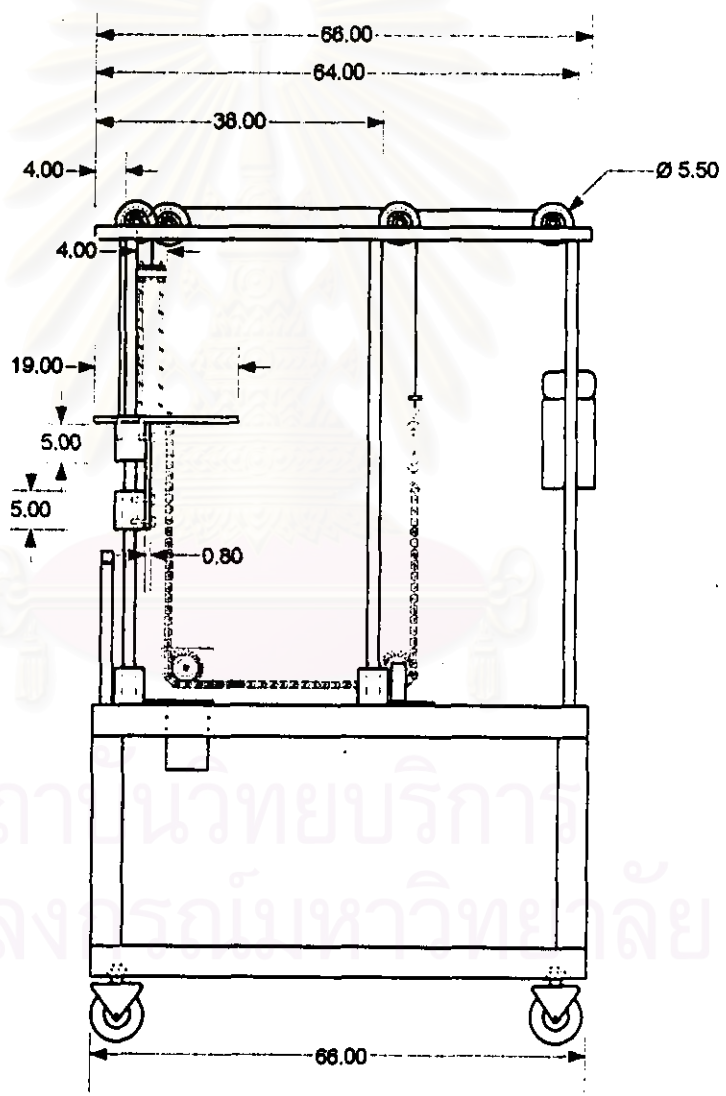
RATIO 1 : 10

บริการ
จุฬาลงกรณ์มหาวิทยาลัย



2&3

RATIO 1 : 10



SIDE VIEW

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง

ฟีดเตอร์ฟังก์ชัน Shepp-Logan



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ฟิลเตอร์ฟังก์ชัน Shepp-Logan

ฟิลเตอร์ฟังก์ชัน Shepp-Logan เป็นฟังก์ชันที่ใช้ในการปรับแก้ข้อมูลภาพเชิงตัวเลขเพื่อให้คุณภาพของภาพดีขึ้น โดยฟังก์ชันของ Shepp-Logan จะทำการปรับแก้ข้อมูลความเข้มของรังสีซึ่งมีผลอันเนื่องมาจากระยะทางของรังสีที่ตกกระทบหัววัดมีระยะทางมากขึ้น แต่จะเป็นการปรับแก้เฉพาะเส้นภาพแต่ละเส้นหรือแต่ละแถวของเมตริกซ์ C

$$\sum_{m=1}^i \sum_{n=1}^j G_{mn} = \sum_{m=1}^i \sum_{n=1}^j C_{mn} f(m_{mn})$$

- โดย C คือข้อมูลภาพเชิงตัวเลขก่อนผ่านฟิลเตอร์ฟังก์ชัน
 G คือข้อมูลภาพเชิงตัวเลขหลังผ่านฟิลเตอร์ฟังก์ชัน
 f(m) คือฟิลเตอร์ฟังก์ชันของ Shepp-Logan

โดยที่

$$f(m_{ji}) = \left(\frac{2}{\pi^2 d(1 - 4m_{ji}^2)} \right)$$

และ d = ช่วงหาของช่องบังคับรังสี หรือขนาดของช่องบังคับลำรังสี (mm.)

$$m = \begin{bmatrix} 0 & 1 & 2 & \dots & (n-1) \\ -1 & 0 & 1 & \dots & (n-2) \\ -2 & -1 & 0 & \dots & (n-3) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -(n-1) & -(n-2) & -(n-3) & \dots & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \dots & c_{1j} \\ c_{21} & c_{22} & c_{23} & \dots & c_{2j} \\ c_{31} & c_{32} & c_{33} & \dots & c_{3j} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{i1} & c_{i2} & c_{i3} & \dots & c_{ij} \end{bmatrix}$$



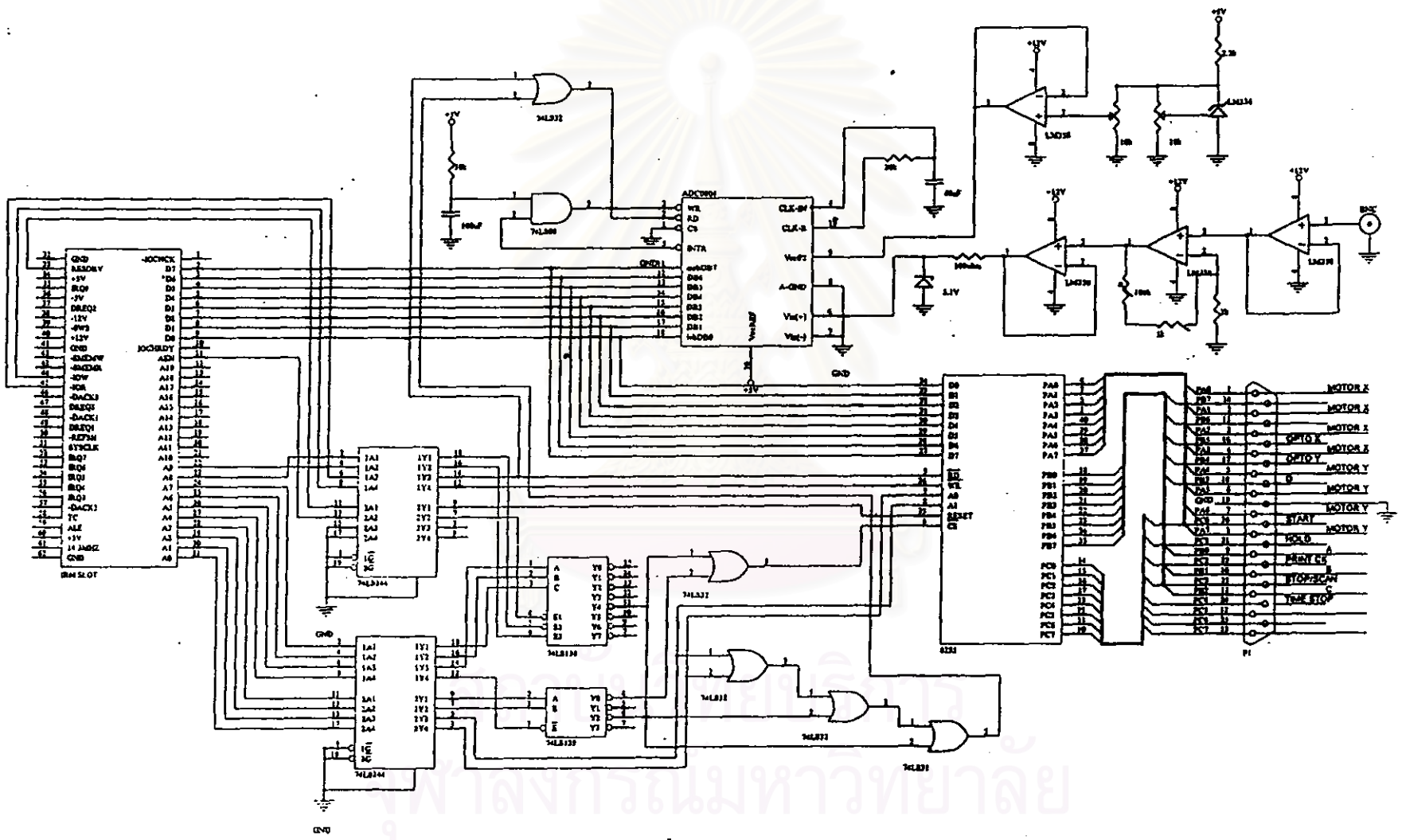
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก จ

วงจรมโหสถูปญาณ



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



วงจรเชื่อมโยงสัญญาณ

ภาคผนวก ง

โปรแกรมควบคุมระบบสแกน
และสร้างภาพตองมิตีของคอนกรีตเสริมเหล็ก



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย


```

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

#define F1 59
#define F2 60
#define F3 61
#define F4 62
#define F9 67
#define F10 68
#define F11 133
#define F12 134
#define HOME 71
#define UP 72
#define PgUp 73
#define END 79
#define DOWN 80
#define PgDn 81
#define INS 82
#define DEL 83
#define RIGHT 77
#define LEFT 75
#define ENTER 13
#define ESC 27
#define TRUE 1
#define CLIP_ON 1
#define CLIP_OFF 0

#define MENU_WIDTH 180 // width of menu
#define MAX_ITEMS 6 // number of menu items

char *menu_item[MAX_ITEMS] = // menu item names
    {"PRE-SCAN", "NEW SCAN", "CORRECT DATA",
    "VIEW PROFILE", "VIEW IMAGE", "QUIT"};

void menu_action(char *item_name);
char get_code(void);
void smooth(void);
void smooth1(void);
void display_menu(void);
void correct_data(void);
int axisX(int &st1);

int axisX1(int &st1,int direct,int n);
int axisY(int &st2);
int axisY1(int &st2,int direct,int n);
void init_counter(void);
void screen2(void);
int huge detectsvga256(void);
int huge detectsvga2562(void);
void pre_scan(void);
void scan(void);
void view_profile(void);
void view_image(void);
void title(void);
void Lnl_I0(void);
void Shepp_Logan(void);
void screen(char TEXTS[15]);
void clearsreen(void);
void disp_status(int x,int y);
void disp_status1(int x,int y);
void digi_disp(int xv1,int yv1,int xyv);
void digi_disp1(int xv1,int yv1,int xyv,int yy,int xcp);
void checkerrors(void);
void menu_box(int left, int top, int width, int height);
void display(int x1,int y1,int n,int r_mj,int r_mn,int step,int mag,int
scalexy);
void markscaleX(int start,int n,int y,int r_mj,int r_mn,int size,int
step,int tb,int ctext,int mag);
void markscaleY(int start,int n,int x,int r_mj,int r_mn,int size,int
step,int tb,int ctext,int mag);
void box1(int x1,int y1,int x2,int y2,int color1,int color2,int color3);
/* x1=Left,x2=Right,y1=Top,y2=Bottom,color1=outside
color,color2=inside color,color3=BG color*/
const portA = 0x300;
const portB = 0x301;
const portC = 0x302;
const portControl = 0x303;
const s1=0x03,s2=0x06,s3=0x0c,s4=0x09;
const s5=0x30,s6=0x60,s7=0xc0,s8=0x90;
int d1,d2,d3,d4,s1,s2,count,mag;
int lt,yy=156,xx=61,ycp,xcp=180,yp=200,px=82,py=158;
const l1=10,l2=15,l3=5;
char text1[16]=" IMAGE DISPLAY",text2[16]="PROFILE
DISPLAY",
text3[16]=" PRE-SCANNING ",text4[16]=" CORRECT
DATA ";
char *Fonts[] = {"0","1","2","3","4","5","6","7","8","9","10","11","12"};

void main(void)

```

```

{
int xr,;
/* request auto detection */
int gdriver, gmode;
struct palettetype pal;
gdriver = installuserdriver("svga256", detectsvga256);
gdriver = DETECT;
/* initialize graphics and local variables */
checkerrors();
initgraph(&gdriver, &gmode, "");
checkerrors();
for (i=0; i<64; i++)
    setrgbpalette(i, i*1, i*1, i*1);
title();
getch();
clearscreen();
do{
    tt=1;
    screen2();
    display_menu();
}while(1);
}

void title(void)
{
    setviewport(0,0,799,599,CLIP_ON);
    box1(0,0,799,599,15,9,8);
    setfillstyle(2, 113);bar(3,3,797,597);
    setcolor(76);rectangle(0,0,799,599);
    setcolor(79);settextstyle(TRIPLEX_SCR_FONT, HORIZ_DIR, 4);
    outtextxy(130,50, "DEVELOPMENT OF A GAMMA-RAY");
    outtextxy(50,105, "TRANSMISSION SCANNING SYSTEM FOR");
    outtextxy(130,160, "TWO-DIMENSIONAL IMAGING OF");
    outtextxy(200,215, "REINFORCED CONCRETE");
    setcolor(87);settextstyle(TRIPLEX_FONT, HORIZ_DIR, 4);
    outtextxy(100,285,"DEPARTMENT OF NUCLEAR
TECHNOLOGY");
    settextstyle(TRIPLEX_SCR_FONT, HORIZ_DIR, 3);
    outtextxy(285,330,"GRADUATE SCHOOL");
    outtextxy(205,370,"CHULALONGKORN UNIVERSITY");
    setcolor(64);settextstyle(TRIPLEX_SCR_FONT, HORIZ_DIR, 2);
    outtextxy(385,420,"BY");
    outtextxy(175,495,"ADVISOR :");
    outtextxy(205,530,"CO-ADVISOR :");
    setcolor(79);
    outtextxy(315,495,"ATTAPORN PATTARASUMUNT");
    outtextxy(375,530,"DECHO THONG-ARAM");

    setcolor(74);
    outtextxy(235,455,"THITIKORN HENSUPPAIBOON");
}

void screen2(void)
{
    setviewport(0,0,799,599,CLIP_ON);
    box1(0,0,799,599,15,9,8);
    setfillstyle(1, 113);bar(3,3,797,179);
    setfillstyle(2, 104);bar(3,183,799,599);
    setlinestyle(0,0xFFFF, 4);
    setcolor(76);rectangle(0,0,799,599);line(0,180,799,180);
    setcolor(70);settextstyle(TRIPLEX_SCR_FONT, HORIZ_DIR, 1);
    outtextxy(55,8,"DEVELOPMENT OF A GAMMA-RAY
TRANSMISSION SCANNING SYSTEM");
    outtextxy(100,40,"FOR TWO-DIMENSIONAL IMAGING OF
REINFORCED CONCRETE");
    setcolor(64);
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 1);
    outtextxy(225,75,"DEPARTMENT OF NUCLEAR
TECHNOLOGY");
    outtextxy(270,100,"CHULALONGKORN UNIVERSITY");
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 1);
    setcolor(120);
    setcolor(64);
    outtextxy(65,125,"Designed by :");
    outtextxy(355,125," THESIS ADVISOR :");
    outtextxy(350,150,"THESIS CO-ADVISOR :");
    setcolor(79);
    outtextxy(550,125,"Attaporn Pattarasumunt");
    outtextxy(550,150,"Decho Thong-Aram");
    settextstyle(TRIPLEX_SCR_FONT, HORIZ_DIR, 2);
    setcolor(79);
    outtextxy(10,150,"Thitikorn Hansuppaiboon");
    return;
}

void display_menu(void)
{
    void *cursorh;
    int l,pyh=58,tt=1,width=180,vpos=0;
    unsigned int size;
    setcolor(140);
    size = imagesize(3, pyh, width+28, pyh+34);
    cursorh = malloc(size);
    getimage(3, pyh+30, width+28, pyh+64, cursorh);
    /* repeat until a key is pressed */
}

```

```

while (tt==1)
{
    menu_box(295, 250, MENU_WIDTH, MAX_ITEMS+37); // draw
the box
    for(i=0; i<=MAX_ITEMS; i++) // for each menu item
    {
        if(i==pyh) // if active item.
            cputs( menu_item[i] );
    }
    putimage(3, pyh, cursorh, XOR_PUT);
(switch (get_code())
{
    case UP :{ vpos=(vpos>0) ? -- vpos : MAX_ITEMS-1;pyh=pyh-
34;
                if (pyh<= 45)
                    pyh=228;};break;
    case DOWN :{ vpos=(vpos<MAX_ITEMS-1) ? ++vpos:
0;pyh=pyh+34;
                if (pyh>=247)
                    pyh =58;};break;
    case 'r' : {menu_action(menu_item[vpos]);tt=2;};break;
}
}
/* plot new image */
putimage(3, pyh, cursorh, XOR_PUT);
}
free(cursorh);
}

void menu_box(int left, int top, int width, int height)
{
    int i,yr=64;
    // define box-size window
    setviewport(left, top, left+width+30,
top+(height*MAX_ITEMS)+7,0);
    setfillstyle(1,220);bar(0, 0, width+30, (height*MAX_ITEMS)+7);
    setfillstyle(9,120);bar(1, 1, width+29, 54);
    setlinestyle(0,0xFFFF, 3);
    setcolor(79);
    rectangle(0,0, width+30, (height*MAX_ITEMS)+7);
    line(0.55,width+30,55);
    setcolor(74);
    settxtstyle(TRIPLEX_FONT, HORIZ_DIR, 2);
    settxtjustify(0,2); outtextxy(48,15,"MAIN MENU");
    setcolor(68);
    settxtstyle(SMALL_FONT, HORIZ_DIR, 7);
    for (i=0;i<=5;i++)

```

```

{
    outtextxy(15,yr,menu_item[i]);
    yr=yr+34;
}
return ;
} // end draw_box()

void menu_action(char *item_name)
{
    int i;
    if( strcmp(item_name, "PRE-SCAN") == 0 )
        pre_scan();
    else if( strcmp(item_name, "NEW SCAN") == 0 )
        scan();
    else if( strcmp(item_name, "CORRECT DATA") == 0 )
        correct_data();
    else if( strcmp(item_name, "VIEW PROFILE") == 0 )
        view_profile();
    else if( strcmp(item_name, "VIEW IMAGE") == 0 )
        view_image();
    else if( strcmp(item_name, "QUIT") == 0 )
    {
        clrscr();
        int gdriver, gmode;
        struct palettetype pal;
        gdriver = installuserdriver("svga256",
detectsvga256);
        gdriver = DETECT;
        /* initialize graphics and local
variables */
        checkererrors();
        inigraph(&gdriver, &gmode, "");
        checkererrors();
        for (i=0; i<16; i++)
            setrgbpalette(i, i, i);
        closegraph();
        exit; // exit to DOS
    }
    clrscr();
    return ;
} // end menu_action

char get_code(void)
{
    char key;
    int i;
    if( (key=getch()) == 0 ) // if extended code,

```



```

return( getch() ); // return the code
else if ( key == 'r' ) // if [Enter],
    return(key); // return it
else if (key == ESC )
{
    clearsreen();
    int gdriver, gmode;
    struct palettetype pal;
    gdriver = installuserdriver("svga256",
detectsvga2562);
    gdriver = DETECT;
    /* initialize graphics and local
variables */
    checkerrors();
    initgraph(&gdriver, &gmode,
"D:\Softdos\Tc3");
    checkerrors();
    for (i=0; i<16; i++)
        setrgbpalette(i, i, i, i);
    closegraph();
    exit; // exit to DOS
}
else // if anything else.
return (1);
} // end get_code()

void screen(char TEXTS[17])
{
    setviewport(0,0,799,599,CLIP_ON);
    box1(0,0,799,599,76,76,245);
    box1(6,6,793,593,104,104,245);
    /* Title */
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 4);
    box1(255,27,555,95,69,64,0);
    setcolor(69);outtextxy(275,40,TEXTS);
}

void init_counter(void)
{
    int i,nn,timest=0,D[6];//count=0,D[6];
    for(j=0;i<300;i++);
    output(portC,0x00);delay(13);
    output(portC,0xA0);delay(13);
    output(portC,0xB0);delay(13);
    timest=(inport(portB)&0x40);delay(13);
while(timest=0)
{
    timest=(inport(portB)&0x40);
    if (timest==0) /*check Time Stop*/
    {
        output(portC,0x90);delay(13);
        for(nn=1;nn<7;nn++)
        {
            output(portC,0xD0); /*sent PRINT CLK
            output(portC,0x90);
            D[nn] = (inport(portB)&0xF);delay(13);
        }
        count=((D[1]*100000)+(D[2]*10000)+(D[3]*1000)+(
D[4]*100)+(D[5]*10)+D[6]);
        output(portC,0x00);
    }
}

void digi_disp(int xv1,int yv1,int xyv)
{
    int xv2,yv2;
    char msg[6];
    xv2=xv1+60;yv2=yv1+20;
    setviewport(xv1,yv1,xv2,yv2,CLIP_ON);
    clearviewport();
    setviewport(xv1,yv1,xv2,yv2,CLIP_ON);
    setfillstyle(1,125);bar(0,0,60,20);
    setcolor(68);rectangle(0,0,60,20);
    setcolor(86);settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    // create a message string
    sprintf(msg, "%d", xyv);
    // output the message
    outtextxy(22, 7, msg);
    setviewport(0,0,799,599,CLIP_OFF);
    moveto(xx,ycp);
}

void digi_disp1(int xv1,int yv1,int xyv,int yy,int xcp)
{
    int xv2,yv2;
    char msg[8];
    xv2=xv1+60;yv2=yv1+20;
    setviewport(xv1,yv1,xv2,yv2,CLIP_ON);
    clearviewport();
    setviewport(xv1,yv1,xv2,yv2,CLIP_ON);
    setfillstyle(1,125);bar(0,0,60,20);
}

```

```

setcolor(68);rectangle(0,0,60,20);
setcolor(86);settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
// create a message string
sprintf(msg, "%d", xyv);
// output the message
outtextxy(22, 7, msg);
setviewport(0,0,799,599,CLIP_OFF);
moveTo(xcp,yy);
}

void display(int x1,int y1,int n,int r_mj,int r_mn,int step,int mag,int
scalexy)
{
int x2,y2;
/* Create Image Display */
x2=x1+30;y2=y1+30;

box1(x1,y1,x2+33+(n*r_mj*mag),y2+33+(n*r_mj*mag),64,119,10
3);
setfillstyle(1, 0);
bar(x2-1,y2-1,x2+2+(n*r_mj*mag),y2+4+(n*r_mj*mag));
setlinestyle(0,0xFFFF, 1);
setcolor(68);
rectangle(x2-1,y2-1,x2+2-(n*r_mj*mag),y2+4+(n*r_mj*mag));
if (scalexy == 2) {
markscaleX(x2+1,n,y2+1,r_mj,r_mn,-4,step,-10,200,mag);

markscaleX(x2+1,n,y2+4+(n*r_mj*mag),r_mj,r_mn,4,step,5,200,
mag);
markscaleY(y2+2,n,x2,r_mj,r_mn,-4,step,-17,200,mag);

markscaleY(y2+2,n,x2+1+(n*r_mj*mag),r_mj,r_mn,4,step,5,200,
mag);
}
if (scalexy == 1) {
markscaleX(x2+1,n,y2+1,r_mj,r_mn,-3,step,-10,200,mag);

markscaleX(x2+1,n,y2+4+(n*r_mj*mag),r_mj,r_mn,3,step,5,200,
mag);
}
if (scalexy == 0) {
markscaleY(y2+2,n,x2+1+(n*r_mj*mag),r_mj,r_mn,3,step,5,200,
mag);
}
}

void disp_status1(int x,int y)
{
int i;
char msg[3];
settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
box1(x,y,x+100,y+303,64,119,120);
setcolor(103);
outtextxy(x+8, y+30,"PROFILE No.");
outtextxy(x+10, y+90,"X POSITION");
outtextxy(x+10,y+150,"Y POSITION");
outtextxy(x+10,y+210," VALUE ");
outtextxy(x+10,y+220," (COUNTS)");
}

void disp_status(int x,int y)
{
settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
box1(x,y,x+220,y+150,69,64,0);
setcolor(103);
outtextxy(x+15,y+30,"PROFILE NUMBER : ");
outtextxy(x+15,y+60,"X POSITION : ");
outtextxy(x+15,y+90,"Y POSITION : ");
outtextxy(x+15,y+120,"VALUE (COUNTS) : ");
setcolor(69);
}

void pre_scan(void)
{
int tt=1,xr,yr,xcp,yy,xpp,mag=4;
char axis;
setviewport(0,0,799,599,CLIP_ON);
screen(tex13);
display(30,125,4,30,3,3,2,2);
disp_status(500,125);
box1(500,300,720,460,69,64,0);
while (tt==1)
{
settextstyle(TRIPLEX_FONT, HORIZ_DIR, 1);
setcolor(78);
outtextxy(517,315,"Select Axis-Scanning");
setcolor(68);
outtextxy(570,350,"1.");
outtextxy(570,380,"2.");
outtextxy(570,410,"3.");
setcolor(142);
outtextxy(595,350,"X-AXIS");
}
}

```

```
outtextxy(595,380,"Y-AXIS");
outtextxy(595,410,"QUIT");
axis=getch();
switch (axis) {
case '1':
{
    display(30,125,4,30,3,3,2,1);
    setfillstyle(1,0);
    bar(61,158,61+237,397);
outport(portControl, 0x82);
axisY(s5);delay(13); // move det. to origin on axis Y
axisY1(st2,2,800);delay(13); // move det. to origin on
axis X
axisX(s1);delay(13); // move det. to origin on axis X
delay(13);
xx=61,ycp=350;
digi_disp(645,148,1);
digi_disp(645,208,1);
moveto(xx,ycp);
for(xr=1;xr<=20;xr++)
{
    digi_disp(645,178,xr);
    init_counter();
    digi_disp(645,238,count);
    setlinestyle(0,0xFFFF , 3);
    ycp=(ypp-
(count*(0.0015))) *mag;
    setcolor(68);
    if (xr==1) putpixel(xx,ycp,2);
    else {xx=xx+(3*mag);lineto(xx,ycp);}
    delay(13);
    moveto(xx,ycp);
    axisX1(st1,2,75*2);delay(13);
    outport(portC, 0x00);
}
};break;
case '2':
{
    yy=158;xpp=40;
    display(30,125,4,30,3,3,2,1);
    setfillstyle(1,0);
    bar(61,158,61+237,397);
outport(portControl, 0x82);
delay(13);
axisY(s5);delay(13); // move det. to origin on axis Y
axisY1(st2,2,250);delay(13); // move det. to origin on
axis Y
    digi_disp(645,148,1);
    digi_disp(645,178,1);
    moveto(xcp,yy);
    for(yr=1;yr<=20;yr++)
    {
        digi_disp(645,208,yr);
        init_counter();
        digi_disp(645,238,count,yy,xcp);
        setlinestyle(0,0xFFFF , 3);
        xcp=(xpp+(count*(0.001))) *mag;
        setcolor(68);
        if (yr==1) putpixel(xcp,yy,2);
        else {yy=yy+(3*mag);lineto(xcp,yy);}
        delay(13);
        axisY1(st2,2,28*2);delay(13);
        outport(portC, 0x00);
    }
};break;
case '3':{tt=2;exit ;};break;
case ESC : {tt=2;exit ;};break;
default: {settextstyle(DEFAULT_FONT, HORIZ_DIR,
1,2,3)};
outtextxy(540,440,"Please Select
");setcolor(76);
}
}
void scan(void)
{
int xr,yr;
char oldname[15] = "image1.dat",newname[15];
FILE *fp;
setviewport(0,0,799,599,CLIP_ON);
screen(text2);
display(30,125,4,30,3,3,2,1);
dispo_status(500,125);
outport(portControl, 0x82);
axisX(s1);delay(13); // move det. to origin on axis X
delay(13*10);
axisX1(st1,2,1500);delay(13); // move det. to middle
on axis X
axisY(s5);delay(13); // move det. to origin on axis Y
axisY1(st2,2,250);delay(13); // move det. to origin on
axis Y
    delay(13);
}
}
}
```



```

axisX(st1); delay(l3); // move det. to origin on axis X
if ((fp = fopen("image1.dat","w+")) != NULL)
{
    for(yr=1;yr<=40;yr++)
    {
        xx=61,ycp=350;
        setfillstyle(1,0);
        bar(xx,158,xx+237,395);
        moveto(xx,ycp);
        digi_disp(645,148,yr);
        digi_disp(645,208,yr);
        for(xr=1;xr<=40;xr++)
        {
            digi_disp(645,178,xr);
            init_counter();
            if (xr==40)
            {
                fprintf(fp,"%6d\n",count); delay(l3);
            }
            else
            {
                fprintf(fp,"%6d\n",count); delay(l3);
            }
            /*plot Profile */
            digi_disp(645,238,count);
            setlinestyle(0,0xFFFF , 3);
            ycp=(ypp-
(count*(0.015)))*mag;
            setcolor(68);
            if
(xr==1){putpixel(xx,ycp,2);}
            else {
                xx=xx+(3*mag);lineto(xx,ycp);}
            delay(l3);
            axisX1(st1,1,75);delay(l3);
            outport(portC, 0x00);
        }
        delay(l3);
        axisX1(st1,2,1500);delay(l3); /* move det. to
middle on axis X */
        axisY1(st2,2,28);delay(l3);
        axisX(st1);delay(l3); /* move det. to origin on axis
X */
    }
}
fclose(fp);
gotoxy(65,25);
printf("Save File Name : ");

setcolor(64);
gets(newname);
/* Rename the file */
if (rename(oldname, newname) == 0) printf("");
else perror("");
}
void correct_data(void)
{
    int tt=1;
    char choice;
    setviewport(0,0,799,599,CLIP_ON);
    screen(text4);
    tt=1;
    while(tt==1){
        setfillstyle(1,245);
        bar(200,190,820,410);
        box1(300,200,500,360,69,64,0);
        settextstyle(TRIPLEX_FONT,
HORIZ_DIR, 1);
        setcolor(78);
        outtextxy(340,220,"Select
Method");
        setcolor(68);
        outtextxy(330,250,"1.");
        outtextxy(330,280,"2.");
        outtextxy(330,310,"3.");
        setcolor(142);
        outtextxy(355,250,"In IO/I");
        outtextxy(355,280,"Shepp-
Logan");
        outtextxy(355,310,"Quit");
        choice=getch();
        switch (choice) {
            case '1': {Lni_IO();tt=2;};break;
            case '2': {Shepp_Logan();tt=2;}; break;
            case '3': {tt=2;exit ;};break;
            case ESC : {tt=2;exit ;};break;
            default: {settextstyle(DEFAULT_FONT, HORIZ_DIR,
1);setcolor(76);
outtextxy(320,340,"Please Select
1,2,3");getch();};
        }
    }
}
return;
}
void Lni_IO(void)

```

```

{
float count2[2000],count3[2000];
float op;
int d,i,count4=0,NumLn,DpL;
char s_name[15],t_name1[15],t_name[15],count1[8],msg[3];
FILE *fp;

    setfillstyle(1,245);
    bar(200,190,620,410);
    box1(210,200,570,375,64,0,0);
    setcolor(68);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(240,225,"Source File Name  :");
    outtextxy(240,250,"Target File Name  :");
    outtextxy(240,275,"Io File Name  :");
    outtextxy(240,300,"Number of Lines  :");
    outtextxy(240,325,"Number of Data/Line  :");
    setcolor(64);
    gotoxy(60,14);
    gets(s_name);
    outtextxy(450,225,s_name);
    gotoxy(60,14);
    gets(t_name1);
    outtextxy(450,250,t_name1);
    gotoxy(60,14);
    gets(t_name);
    outtextxy(450,275,t_name);
    gotoxy(60,14);
    scanf("%d",&NumLn);
    sprintf(msg, "%d", NumLn);
    outtextxy(450,300,msg);
    gotoxy(60,14);
    scanf("%d",&DpL);
    sprintf(msg, "%d", DpL);
    outtextxy(450,325,msg);
    if ((fp = fopen(s_name,"rt")) != NULL)
    {
        for(i=0;i<(NumLn*DpL);i++)
        {
            count3[i]=atof(fgets(count1,sizeof(count1),fp));
        }
        fclose(fp);
    }
    if ((fp = fopen(t_name,"rt")) != NULL)
    {
        for(i=0;i<(NumLn*DpL);i++)
        {
            count2[i]=atof(fgets(count1,sizeof(count1),fp));
        }
        fclose(fp);
    }
    if ((fp = fopen(t_name1,"w+")) != NULL)
    {
        for(i=0;i<(NumLn*DpL);i++)
        {
            op=0;
            op=640*(log(count2[i])-
            log(count3[i]));
            count4=op/1;
            fprintf(fp,"%d\n",count4);
        }
        fclose(fp);
        setfillstyle(1,0);
        bar(217,427,563,463);
        box1(210,420,570,470,64,0,0);
        setcolor(68);
        settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
        outtextxy(250,440,"Correct Data OK!");
        getch();
        return;
    }
}

void Shepp_Logan(void)
{
    double count3[40][40];
    int count1[1600],count2[40][40];
    double l_LOWEST,f,f1,sum1,factor,min;
    int d,i,xr,yr,zr,NumLn,DpL,m;
    char s_name[15],t_name1[15],count1[8],msg[3];
    FILE *fp;

    setfillstyle(1,245);
    bar(200,190,620,410);
    box1(210,200,570,375,64,0,0);
    setcolor(68);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    outtextxy(240,225,"Source File Name  :");
    outtextxy(240,250,"Target File Name  :");
    outtextxy(240,275,"Number of Lines  :");
    outtextxy(240,300,"Number of Data/Line  :");
    outtextxy(240,325,"Diameter of Collimator (mm.)  :");
    setcolor(64);

```



```

gotoxy(60,14);
gets(s_name);
outtextxy(450,225,s_name);
gotoxy(60,14);
gets(t_name1);
outtextxy(450,250,t_name1);
gotoxy(60,14);
scanf("%d",&NumLn);
sprintf(msg,"%d",NumLn);
outtextxy(450,275,msg);
gotoxy(60,14);
scanf("%d",&DpL);
sprintf(msg,"%d",DpL);
outtextxy(450,300,msg);
gotoxy(60,14);
scanf("%d",&d);
sprintf(msg,"%d",d);
outtextxy(500,325,msg);
if ((fp = fopen(s_name,"r")) != NULL)
{
    i=0;
    for(yr=0;yr<NumLn;yr++)
    {
        for(xr=0;xr<DpL;xr++)
        {
            count1[i]=atoi(fgets(count1,sizeof(count1),fp));
            count2[yr][xr]=count1[i];
            i++;
        }
    }
    fclose(fp);
}

count3[yr][xr]=sum1;
}
}
min=0;
for(yr=0;yr<NumLn;yr++)
{
    for(xr=0;xr<DpL;xr++)
    {
        min=(min<count3[yr][xr]) ? min : count3[yr][xr];
    }
    i_LOWEST = (-1)*min;
    if ((fp =
        fopen(t_name1,"w+") != NULL)
    {
        for(yr=0;yr<NumLn;yr++)
        {
            for(xr=0;xr<DpL;xr++)
            {
                count2[yr][xr]=64*(i_LOWEST+count3[yr][xr])/1;
                fprintf(fp,"%d\n",count2[yr][xr]);
            }
        }
        fclose(fp);
        setfillstyle(1,0);

        for(yr=0;yr<NumLn;yr++)
        {
            bar(217,427,563,463);
            for(xr=0;xr<DpL;xr++)
            {
                box1(210,420,570,470,64,0,0);
                setcolor(68);

                sum1=0.0;
                for(zr=0; zr<DpL; zr++)
                {
                    m=xr-zr;
                    f=(1-(4*m*m));
                    outtextxy(250,440,"Correct Data OK");
                    getch();
                }
            }
        }
        return;
    }
}

void view_profile(void)
{
    factor=2/(M_PI*M_PI*d*f);
    sum1 +=
    count2[yr][zr]*factor,

```

```

float c;
int xr,yr=1,d,lmax,lmin,xx,ypp=390;
long int ycp=350,count2[1600];
char newname[15],count1[8],mag[12];
FILE *fp;

setviewport(0,0,799,599,CLIP_ON);
screen(text2);
display(30,125,4,30,3,2,1);
disp_status(500,125);
box1(500,345,730,380,89,0,0);
settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
setcolor(64);
outtextxy(510,380,"File Name : ");
gotoxy(80,23);
scanf("%s",&newname);
setcolor(69);
outtextxy(600,380,newname);

// output the message
if ((fp = fopen(newname,"rt")) != NULL)
{
    for(xr=1;xr<=1600;xr++)
    {
count2[xr]=atoi(fgets(count1,sizeof(count1),fp));
        lmin=(lmin<count2[xr]) ? lmin :
count2[xr];
        lmax=(lmax>count2[xr]) ? lmax :
count2[xr];
    }
fclose(fp);
if ((fp = fopen(newname,"rt")) != NULL)
{
    for(yr=1;yr<=40;yr++)
    {
        xx=61;
        setfillstyle(1,0);
        bar(xx,158,xx+237,397);
        moveto(xx,ycp);
        digi_disp(645,148,yr);
        digi_disp(645,208,yr);
        for(xr=1;xr<=40;xr++)
        {
count2[xr]=atoi(fgets(count1,sizeof(count1),fp));
            digi_disp(645,178,xr);
            digi_disp(645,238,count2[xr]);
setlinestyle(0,0xFFFF,3);
moveto(xx,ycp);
            d = (232*(count2[xr]-
lmin))/(lmax-lmin);
            ycp= ypp-d;
            setcolor(68);
            if (xr==1) putpixel(xx,ycp,2);
            else if (xr>1)
                {xx=xx+(6);lineto(xx,ycp);}
            delay(50);
        }
        getch();
    }
fclose(fp);
}

void view_image(void)
{
    int
    x,y,xr,gl=63,yb=128,yc,cp,M=0,N=0,m,c,lmax=0,lmin=0,n,CM=0
    ,inv=0,count4[1600];
    long int count2[1600],count3[1600];
    char newname[15],count1[8],newname1[15];
    mag=1;
    FILE *fp;

    setviewport(0,0,799,599,CLIP_ON);
    screen(text1);
    display(119,125,4,10,1,3,mag+1,2);
    disp_status(500,125);
    settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
    box1(50,500,750,570,69,64,0);
    setcolor(64);
    outtextxy(75,520,"INS: HOME: PgUP:
F1: F11: ");
    outtextxy(75,545,"DEL: END: PgDN:
F2: F12: ");
    setcolor(69);
    outtextxy(77,520," ZOOM IN MAX LEVEL+
MIN LEVEL+ BRIGHT+ NORM./INV.");
    outtextxy(77,545," ZOOM OUT MAX LEVEL-
MIN LEVEL- BRIGHT- SAVE FILE ");
    box1(30,125,51,450,89,0,0);
    for (xr=63;xr>=0;xr-)
    {
        yc=256/64;
        setfillstyle(SOLID_FILL, xr);

```

```

bar(33,yb,48,yb+yc);
yb=yb+yc+1;
}
// box1(70,125,91,450,69,0,0);

box1(500,345,730,380,69,0,0);
setcolor(64);
outtextxy(510,380,"File Name : ");
setcolor(69);
gotoxy(80,23);
scanf("%s",newname);
outtextxy(600,360,newname);
if ((fp = fopen(newname,"r")) != NULL)
{
    for(xr=0;xr<1600;xr++)
    {
        count2[xr]=atoi(fgets(count1,sizeof(count1),fp));
        lmin=(lmin<count2[xr]) ? lmin :
count2[xr];
        lmax=(lmax>count2[xr]) ? lmax :
count2[xr];
        count3[xr]=count2[xr];
    }
}
fclose(fp);

M=lmax;N=lmin;cp=(lmax-lmin)/64;

y=157;
x=150;
for(xr=0;xr<1600;xr++)
{
    if ((xr>=40)&&(xr%40==0))
    {x=150;y=y+(mag+1);}
    c = 63-((63*(count2[xr]-lmin))/(lmax-
lmin))/1;
    setfillstyle(SOLID_FILL, c);
    bar(x,y,x+(mag),y+(mag));
    x=x+(mag+1);
}

while (getch() == 0)
{
    switch (getch()){
        case F9 : {gl=gl+1;
                    if (gl>=63) gl=63;};break;
        case F10 : {gl=gl-1;
                    if (gl<=1) gl=1;};break;
        case F11 : {if (inv==0) inv=1;
                    else inv = 0;};break;
        case F12 : { setfillstyle(1,245);
                    bar(500,345,730,380);
                    box1(500,345,730,380,69,0,0);
                    setcolor(64);
                    outtextxy(510,380,"File
                    Name : ");
                    setcolor(69);
                    gotoxy(80,23);
                    scanf("%s",newname1);
                    outtextxy(600,360,newname1);
                    if ((fp = fopen(newname1,"w+")) !=
                    NULL)
                    {
                        for(xr=0;xr<1600;xr++)
                        {
                            count4[xr]=64*count4[xr];
                            fprintf(fp,"%d\n",count4[xr]);
                        }
                    }
                    fclose(fp);};break;
        case INS : {mag=mag+1;
                    setfillstyle(1,245);
                    bar(117,123,464,470);
                    if (mag>=6) mag=6;
                    display(119,125,4,10,1,3,mag+1,2);};break;
        case DEL : {mag=mag-1;
                    setfillstyle(1,245);
                    bar(117,123,464,470);
                    if (mag<=1) mag=1;
                    display(119,125,4,10,1,3,mag+1,2);};break;
        case HOME : {M=M-cp;m=m-1;
                    if (M<=N) M=N+cp;
                    if (m<=1) m=1;};break;
        case END : {M=M+cp;m=m+1;
                    if (M>=lmax) M=lmax;
                    if (m>=63) m=62;};break;
        case PgUp : {N=N+cp;n=n-1;
                    if (N>=M) N=M-cp;

```

```

        if (n<=0) n=0;);break;
    }
    case PgDn : (N=N-cp;n=n+1;
        if (N<=lmin) N=lmin;
        if (n>=63) n=62;);break;
        return;
    }
    case F2 : (CM=CM-1;
        if (CM<=0) CM=0;);break;
        void markscaleX(int start,int n,int y,int r_mj,int r_mn,int size,int
    case F1 : (CM=CM+1;
        if (CM>=63)
        step,int tb,int ctext,int mag)
        {
    CM=63;);break;
        int i,c,end,fonts=0;
    case ESC : exit;
        r_mn=r_mn*mag;
    case ENTER : exit;
        r_mj=r_mj*mag;
    } //END CASE
        end=(r_mj*n)+start;
        y=157;
        if (r_mn!=0)
        x=150;
        {
            c=r_mj/r_mn;
        }
        for(xr=0;xr<1600;xr++)
        {
            setcolor(ctext);
            count3[xr]=count2[xr];
            settxtstyle(DEFAULT_FONT, HORIZ_DIR, 1);
        }
        for(xr=0;xr<1600;xr++)
        {
            setlinestyle(0,0xFFFF, 1);
            line(start,y,start,y+(2*size));
            outtextxy(start-2,y+(2*size+tb),Fonts[0]);
            if (count3[xr]>=M)
            while ((r_mn!=0) && (start < end ))
            {
                count3[xr]=M;
                if (count3[xr]<=N)
                for(i=0;i<c;i++)
                {
                    line(start,y,start,y+size);
                    count3[xr]=N;
                    start=start+r_mn;
                }
                if (inv==1)
                {
                    c = ((gl*(count3[xr]-N))/(M-
                    fonts=fonts+step;
                    if (fonts < 10)
                    outtextxy(start-2,y+(2*size+tb),Fonts[fonts]);
                    else
                    outtextxy(start-6,y+(2*size+tb),Fonts[fonts]);
                    line(start,y,start,y+(2*size));
                }
                if (inv==0)
                {
                    c = gl - ((gl*(count3[xr]-
                    }
                    while ((r_mn==0) && (start < end ))
                    {
                        N))/(M-N))/1;
                    line(start,y,start,y+(2*size));
                    if (fonts >= 10)
                    outtextxy(start-6,y+(2*size+tb),Fonts[fonts]);
                    else
                    outtextxy(start-2,y+(2*size+tb),Fonts[fonts]);
                    start=start+r_mj;
                    fonts=fonts+step;
                }
            }
            if ((xr>=40)&&(xr%40==0))
            {
                line(start,y,start,y+(2*size));
                if (fonts >= 10)
                outtextxy(start-6,y+(2*size+tb),Fonts[fonts]);
                else
                outtextxy(start-2,y+(2*size+tb),Fonts[fonts]);
                start=start+r_mj;
                fonts=fonts+step;
            }
            (x=150;y=y+(mag+1);)
            c=c+CM;
            if (c>63) c=63;
            if (c<0) c=0;
            setfillstyle(SOLID_FILL, c);
            count4[xr]=c;
            bar(x,y,x+(mag),y+(mag));
            x=x+(mag+1);
        }
    }

```

```

void markscaleY(int start,int n,int x,int r_mj,int r_mn,int size,int
step,int tb,int ctext,int mag)
{
int i,c,end,fonts=0;
    r_mn=r_mn*mag;
    r_mj=r_mj*mag;
    end = (r_mj*n)+start;
    if (r_mn!=0)
    {
        c=r_mj/r_mn;
    }
setcolor(ctext);
settextstyle(DEFAULT_FONT, HORIZ_DIR, 1);
setlinestyle(0,0xFFFF, 1);
    line(x,start,x+(2*size),start);
    outtextxy(x+(2*size+tb)+5,start-2,Fonts[0]);
while ((r_mn!=0) && (start < end))
{
    for(i=0;i<c;i++)
    {
        line(x,start,x+size,start);
        start=start+r_mn;
    }
    fonts=fonts+step;
    if (fonts < 10)
        outtextxy(x+(2*size+tb)+5,start-2,Fonts[fonts]);
    else
        outtextxy(x+(2*size+tb),start-2,Fonts[fonts]);
    line(x,start,x+(2*size),start);
}
while ((r_mn==0) && (start < end))
{
    line(x,start,x+(2*size),start);
    if (fonts >= 10)
        outtextxy(x+(2*size+tb),start-2,Fonts[fonts]);
    else
        outtextxy(x+(2*size+tb),start-2,Fonts[fonts]);
    start=start+r_mj;
    fonts=fonts+step;
}
}

void box1(int x1,int y1,int x2,int y2,int color1,int color2,int color3)
{
    setlinestyle(0,0xFFFF,
3);setcolor(color1);rectangle(x1,y1,x2,y2);

```

```

    setlinestyle(0,0xFFFF,
3);setcolor(color2);rectangle(x1+3,y1+3,x2-3,y2-3);
    setfillstyle(1, color3);bar(x1+6,y1+6,x2-6,y2-6);
}

int axisX(int &st1)
{
int i=1,chk2=0,chk_opto=0;
    switch (st1) {
        case 0x03 : {d1=s1;d2=s4;d3=s3;d4=s2;};break;
        case 0x06 : {d1=s2;d2=s1;d3=s4;d4=s3;};break;
        case 0x0C : {d1=s3;d2=s2;d3=s1;d4=s4;};break;
        case 0x09 : {d1=s4;d2=s3;d3=s2;d4=s1;};break;
        default : {d1=s4;d2=s3;d3=s2;d4=s1;};
    }
    while (i <= 3650)
    {
        outport(portA, d1); delay(11);
        chk2=inport(portB);
        chk_opto=(chk2 & 0xF0);
        if ((chk_opto == 0xD0) || (chk_opto == 0xF0))
            {st1=d1;return 0;}
        outport(portA, d2); delay(11);
        chk2=inport(portB);
        chk_opto=(chk2 & 0xF0);
        if ((chk_opto == 0xD0) || (chk_opto == 0xF0))
            {st1=d2;return 0;}
        outport(portA, d3); delay(11);
        chk2=inport(portB);
        chk_opto=(chk2 & 0xF0);
        if ((chk_opto == 0xD0) || (chk_opto == 0xF0))
            {st1=d3;return 0;}
        outport(portA, d4); delay(11);
        chk2=inport(portB);
        chk_opto=(chk2 & 0xF0);
        if ((chk_opto == 0xD0) || (chk_opto == 0xF0))
            {st1=d4;return 0;}
        i++;
    }
    return 0;
}

int axisX1(int &st1,int direct,int n)
{
int i=1;
    switch (st1) {
        case 0x03 : {if (direct ==
1){d1=s1;d2=s2;d3=s3;d4=s4;};

```

```

                if (direct ==
2)(d1=s1;d2=s4;d3=s3;d4=s2;);break;
                case 0x06 : {if (direct ==
1)(d1=s2;d2=s3;d3=s4;d4=s1;);
                if (direct ==
2)(d1=s2;d2=s1;d3=s4;d4=s3;);break;
                case 0x0C : {if (direct ==
1)(d1=s3;d2=s4;d3=s1;d4=s2;);
                if (direct ==
2)(d1=s3;d2=s2;d3=s1;d4=s4;);break;
                case 0x09 : {if (direct ==
1)(d1=s4;d2=s1;d3=s2;d4=s3;);
                if (direct ==
2)(d1=s4;d2=s3;d3=s2;d4=s1;);break;
                default : {d1=s1;d2=s2;d3=s3;d4=s4;);
                }
                delay(13);
while (i <= n)
{
    outport(portA, d1); delay(11);
    outport(portA, d2); delay(11);
    outport(portA, d3); delay(11);
    outport(portA, d4); delay(11);
    i++;
}
return 0;
}

int axisY(int &st2)
{
int i=1,chk2=0,chk_opto=0;
switch (st2) {
    case 0x30 : {d1=s6;d2=s7;d3=s8;d4=s5;};break;
    case 0x60 : {d1=s7;d2=s8;d3=s5;d4=s6;};break;
    case 0xc0 : {d1=s8;d2=s5;d3=s6;d4=s7;};break;
    case 0x90 : {d1=s5;d2=s6;d3=s7;d4=s8;};break;
    default : {d1=s8;d2=s7;d3=s6;d4=s5;};
}
while (i <= 1660)
{
    outport(portA, d1); delay(12);
    chk2=inport(portB);
    chk_opto=(chk2 & 0xF0);
    if ((chk_opto == 0xE0) || (chk_opto == 0xF0))
        {st2=d1;return 0;};
    outport(portA, d2); delay(12);
    chk2=inport(portB);
    chk_opto=(chk2 & 0xF0);
    if ((chk_opto == 0xE0) || (chk_opto == 0xF0))
        {st2=d2;return 0;};
    outport(portA, d3); delay(12);
    chk2=inport(portB);
    chk_opto=(chk2 & 0xF0);
    if ((chk_opto == 0xE0) || (chk_opto == 0xF0))
        {st2=d3;return 0;};
    outport(portA, d4); delay(12);
    chk2=inport(portB);
    chk_opto=(chk2 & 0xF0);
    if ((chk_opto == 0xE0) || (chk_opto == 0xF0))
        {st2=d4;return 0;};
    i++;
}
return 0;
}

int axisY1(int &st2,int direct,int n)
{
int i=1;
switch (st2) {
    case 0x30 : {if (direct == 1)
        {d1=s6;d2=s7;d3=s8;d4=s5;};
        if (direct == 2)
        {d1=s8;d2=s7;d3=s6;d4=s5;};break;
    case 0x60 : {if (direct == 1)
        {d1=s7;d2=s8;d3=s5;d4=s6;};
        if (direct == 2)
        {d1=s5;d2=s8;d3=s7;d4=s6;};break;
    case 0xC0 : {if (direct == 1)
        {d1=s8;d2=s5;d3=s6;d4=s7;};
        if (direct == 2)
        {d1=s6;d2=s5;d3=s8;d4=s7;};break;
    case 0x90 : {if (direct == 1)
        {d1=s5;d2=s6;d3=s7;d4=s8;};
        if (direct == 2)
        {d1=s7;d2=s6;d3=s5;d4=s8;};break;
    default : {d1=s8;d2=s7;d3=s6;d4=s5;};
}
while (i <= n)
{
    outport(portA, d1); delay(12);
    outport(portA, d2); delay(12);
    outport(portA, d3); delay(12);
    outport(portA, d4); delay(12);
    i++;
}
}
}

```

```

    }
    return 0;
}

int huge_detectsvga256(void)
{
    int driver, mode, sugmode = 3;
    detectgraph(&driver, &mode);
    return sugmode;
    /* return grError; */
}

int huge_detectsvga2562(void)
{
    int driver, mode, sugmode = 2;
    detectgraph(&driver, &mode);
    return sugmode;
    /* return grError; */
}

void checkerrors(void)
{
    int errorcode;
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }
}

void clearscreen(void)
{
    int xmax, ymax, x=0, y;
    setviewport(0,0,799,599,CLIP_ON);
    xmax=getmaxx();
    ymax=getmaxy();
    setfillstyle(1,144);
    for(i=0;i<=xmax/4;i++)
    {
        bar(x,0,x+2,ymax);
        bar(xmax,0,xmax+2,ymax);delay(15);
        x=x+4;xmax=xmax-4;
    }
    delay(50);
    setfillstyle(1,0);
    for(i=xmax/4;i<=xmax;i--)
    {
        bar(x,0,x+4,ymax);
        bar(xmax,0,xmax+4,ymax);delay(15);
        x=x+4;xmax=xmax-4;
    }
    cleardevice();
    clearviewport();
    return;
}

```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ประวัติผู้เขียน

นายฐิติกร เห็นทรัพย์ไพบุลย์ เกิดเมื่อวันที่ 12 กันยายน พ.ศ. 2514 ภูมิลำเนาเดิมแขวงบ้านช่างหล่อ เขตบางกอกน้อย กรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาตรี จากภาควิชาฟิสิกส์อุตสาหกรรมและอุปกรณ์การแพทย์ (ฟิสิกส์อุตสาหกรรม) คณะวิทยาศาสตร์ประยุกต์จากสถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ เมื่อปีการศึกษา 2536 จากนั้นเข้าศึกษาต่อภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2537



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย