รายการอ้างอิง

1. ธีระวัฒน์ ประกอบผล. การพัฒนาเก็บข้อมูลด้วยเทคนิคโทรทัศน์สำหรับคำนวณสร้างภาพโทโมกราฟี. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชานิวเคลียร์เทคโนโลยี บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2536.
2. สมยศ ศรีสถิตย์, อรรถพร ภัทรสุมันต์. การคำนวณ สร้างภาพโทโมกราฟีด้วยเทคนิคฟิล์มเพื่อการตรวจสอบโดยไม่ทำลาย. งานวิจัย ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2538.
3. มงคล วรรณประภา. การพัฒนาระบบสแกนด้วยรังสีแกมมาเพื่อคำนวณการสร้างภาพโทโมกราฟีของเสาคอนกรีตเสริมเหล็ก. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชานิวเคลียร์เทคโนโลยี บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2536.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บรรณานุกรม

1. Ernest O. Doebelin, Measurement Systems Application and Design, 4th ed. (McGraw-Hill,Inc., 1990)
2. Frank Baeseler, Bruce Bovill, Scanning & Image Processing for the PC (McGraw-Hill,Inc,1993)
3. James Maas, Industrial Electronics (Prentice Hall International ,Inc.,1995)
4. John C. Russ, Computer-Assisted Microscopy: *The Measurement and Analysis of Images* (New York : Plenum Press,1990)
5. Mohammad A. Karim, Electro-optical Devices and Systems, (PWS-KENT Publishing Company, 1990)
6. Paul L. DeVries, A First Course in Computational Physics, (John Wiley & Sons,Inc.,1994)
7. Thomas S. Curry III et. al. , Christensen's Introduction to the Physics of Diagnostic Radiology, 3rd. ed. (Lea & Febiger , Philadelphia., 1984)
8. Wolfram Stadler, Analytical Robotics and Mechantronics, (McGraw-Hill.Inc.,1995)
9. Yasushi IKEDA , Atsuhisa ANDO , Kohei OHKUBO , Masanobu YOKOI , A New Imaging Device for neutron CT , Collected Papers of Research Activities on Neutron Radiography In Japan , May 14-18 ,1989 , (Department of Nuclear Engineering , Faculty of Engineering , Nagoya University , Furo-cho , Chikusa-ku , Nagoya 464-01 , Janpan AND Institute for Atomic Energy , Rikkyo University 2-5-1 Nagasaka , Yokosuka , Kanagawa 240-01 , Janpan)

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

โปรแกรมสนับสนุนการทำงานของระบบเก็บข้อมูลโพรไฟล์จากภาพถ่ายรังสีบนฟิล์ม

โปรแกรมการทำงานของระบบเก็บข้อมูลโพรไฟล์จากภาพถ่ายรังสีบนฟิล์มเพื่อคำนวณสร้างภาพโทโมกราฟี เขียนด้วยภาษา C และ Assembly ร่วมกันโดยในส่วนของภาษา Assembly นั้นจะถูกกำหนดโดยใช้การ inline ด้วยคำสั่ง asm แล้วใช้ debugger ของภาษา C

ตัวแปร

EMS_SEG	เป็นค่าเซกเมนต์ของหน่วยความจำ
offset	เป็นค่าออฟเซ็ตของหน่วยความจำ
line_max	เป็นจำนวนครั้งที่อ่านต่อหนึ่งโพรไฟล์และโพรไฟล์สูงสุด
line_no	เป็นจำนวนครั้งที่อ่านต่อหนึ่งโพรไฟล์
profile	เป็นจำนวนโพรไฟล์
file_name	ชื่อแฟ้มข้อมูลที่ใช้บันทึก
port_no	พอร์ตของไมโครคอมพิวเตอร์ที่ใช้ติดต่อกับสแกนเนอร์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
/*  PROGRAM AQUISITION PROFILE FROM FILM SYSTEM

            :
            : Program link subroutine from inline Assembly
            :
    Object  : Use Gray scale DMA3 IRQ5
            :Scan value from DMA to EMS memory display graphic
            :400 dpi = 8Eh buff = 677h  DOTS/LINES
            :
            :Control port 108h for out 4 bit control stepping motor
            :109h for in  2 bit limit moving
            :109h for out set NMI
*/

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

asm     IO_NMI      EQU  0A0h
asm     ENA_NMI     EQU  80h
asm     DIS_NMI     EQU  0
asm     P_ICWA      EQU  20h
asm     P_ICWB      EQU  21h
asm     OCW1        EQU  11011111b
asm     EOI         EQU  20h
asm     SEG_VEC     EQU  0000h
asm     OFF_VEC     EQU  0034h
asm     PAGE_REG    EQU  82h        /* DMA1=80h  DMA3=82h */
asm     BASEAD_REG  EQU  6          /* DMA1= 2   DMA3= 6 */
asm     BASEWC_REG  EQU  7          /* DMA1= 3   DMA3= 7 */
asm     MODE_REG    EQU  0Bh
asm     D_MODE      EQU  00000111b  /* Demand Mode Tranfer = 000001__b */
asm     COMM_REG    EQU  8
asm     COMM_DAT    EQU  0
```

```
asm     CMASK_REG   EQU 0Eh
asm     MASK_REG    EQU 0Ah
asm     ENA_MASK    EQU 00000011b   /* DMA1= 001b DMA3= 011b */
asm     MAST_REG    EQU 0Dh
asm     FL_FF       EQU 0Ch

unsigned char buff,dat;
unsigned int i,j,k,l;
unsigned int OLD_SEG,OLD_OFFS,page_r,base_add,rec_no,offset;
unsigned int p_handle,EMS_SEG,free_p,tot_p,map_page,al_page;
unsigned int f_handle,port_no,sc_mode,buff_no,line_max,line_no;
unsigned int c_port,m_relay,m_loop,profile;
int x,y,z,no,fd;
int x_max,y_max,c_max,gdriver,gmode;
char file_name[20];
char *err,*text1,*text2,*text3;

#define menu    "Aquisition Profile data form Film system"
#define choice0 "Main Menu"
#define choice1 "Initial Film and Varies Intensity of Light"
#define choice2 "Input No.line scan  No.profile Filename"
#define choice3 "Correct Profile Data"
#define choice4 "Exit Program"
#define pointer "\020"
#define cr_up   "\030"
#define cr_down "\031"


void NON()
   {
asm     PUSH AX
asm     PUSH DX
asm     POP  DX
asm     POP  AX
   }
```

```
void SETSCAN(unsigned char dat)
  {  if((dat&2)==2)                          /* Set scanner active */
          { gotoxy(74,1);
            printf(" on  ");
            outp(port_no,2);
            for(i=0;i<0x1C;i++)
              for(j=0;j<0x3FFF;j++)
                  NON();
          }
         else
          { outp(port_no,0);                 /* Set scanner off */
            gotoxy(74,1);
            printf(" off ");
          }
  }

void INI_SCAN()
  {  port_no=0;
         sc_mode=0;
         buff=0xFF;
         do
          { port_no=port_no+0x100;
            buff=inp(port_no)&0xFE;
            SETSCAN(2);
            sc_mode=inp(port_no)&0xFE;
            SETSCAN(0);
          } while(sc_mode==buff);
         SETSCAN(2);
         do
          { sc_mode=inp(port_no)&8;
            if(sc_mode==8)
              { gotoxy(1,1);
                printf("                    ");
              }
            else
              { gotoxy(1,1);
```

```
                printf(" Set scanner to gray Scale ");
              }
            } while(sc_mode==0);
           do
            { sc_mode=inp(port_no)&0xFE;
              if(sc_mode==0x8E)
               { gotoxy(1,1);
                 printf("                          ");
               }
              else
               { gotoxy(1,1);
                 printf(" Set scanner to 400 dpi    ");
               }
            } while(sc_mode!=0x8E);
           SETSCAN(0);
           buff_no=0x677;
  }

void ENA_INT()                          /* Enable Interrupt NMI */
  {
asm       CLI
asm       PUSH AX

asm       IN   AL,P_ICWB                /* Program OCW1 */
asm       MOV  AH,OCW1
asm       AND  AL,AH                    /* Enable IRQ_NO */
asm       OUT  P_ICWB,AL

asm       POP  AX
asm       STI
  }

void DIS_INT()                          /* Disable INT Request */
  {
asm       CLI
asm       PUSH AX
```

```
asm     IN   AL,P_ICWB              /* Program OCW1 */
asm     MOV  AH,OCW1
asm     NOT  AH                     /* Disable IRQ_NO */
asm     OR   AL,AH
asm     OUT  P_ICWB,AL

asm     POP  AX
asm     STI
   }

void SET_NMI()                      /* Set NMI Interupt Routine Scaner */
  {
asm     CLI
asm     PUSH DS
asm     PUSH ES
asm     PUSH SI
asm     PUSH AX
asm     PUSH DX

        DIS_INT();
asm     MOV  AL,ENA_NMI
asm     OUT  IO_NMI,AL              /* Enable Nonmaskable Interupt */

asm     MOV  AX,SEG_VEC
asm     MOV  ES,AX
asm     MOV  AX,OFF_VEC
asm     MOV  SI,AX
asm     MOV  DX,ES:[SI]
asm     MOV  AX,ES:[SI+2]
        OLD_OFFS=_DX;
        OLD_SEG=_AX;
asm     MOV  AX,SEG _INTROUT
asm     MOV  DX,OFFSET _INTROUT
asm     MOV  ES:[SI],DX
asm     MOV  ES:[SI+2],AX
```

```
asm     POP  DX
asm     POP  AX
asm     POP  SI
asm     POP  ES
asm     POP  DS
asm     STI
  }

void RST_NMI()                          /* Reset NMI */
  {
asm     CLI
asm     PUSH DS
asm     PUSH ES
asm     PUSH SI
asm     PUSH AX
asm     PUSH DX

        DIS_INT();
asm     MOV  AL,DIS_NMI                 /* Disable Nonmaksable Interupt */
asm     OUT  IO_NMI,AL
asm     MOV  AL,EOI                     /* Program OCW2 Send EOI signal */
asm     OUT  P_ICWA,AL

asm     MOV  AX,SEG_VEC
asm     MOV  ES,AX
asm     MOV  AX,OFF_VEC
asm     MOV  SI,AX
asm     MOV  AX,SEG _OLD_OFFS
asm     MOV  DS,AX
        _DX=OLD_OFFS;
asm     MOV  ES:[SI],DX
        _AX=OLD_SEG;
asm     MOV  ES:[SI+2],AX

asm     POP  DX
```

```
asm     POP  AX
asm     POP  SI
asm     POP  ES
asm     POP  DS
asm     STI
  }

void DIS_DMA()                          /* Disable DMA */
  {
asm     PUSH AX

asm     MOV  AL,ENA_MASK
asm     OR   AL,00000100b
asm     OUT  MASK_REG,AL

asm     POP  AX
  }

void ENA_DMA()                          /* Enable DMA */
  {
asm     PUSH AX

asm     MOV  AL,ENA_MASK
asm     OUT  MASK_REG,AL

asm     POP  AX
  }

void SET_DMA()                          /* Program address to DMA process */
  {
asm     PUSH AX
asm     PUSH DX

        _AX=EMS_SEG;                    /* Set Start Address */
        _DX=0x10;
asm     MUL  DX
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
        page_r=_DX;
        base_add=_AX+rec_no;
        _AX=page_r;
asm     OUT  PAGE_REG,AL
asm     OUT  FL_FF,AL
        _AX=base_add;
asm     OUT  BASEAD_REG,AL
asm     MOV  AL,AH
asm     OUT  BASEAD_REG,AL
        outp(port_no+3,_AL);
        outp(port_no+5,3);
        _AX=buff_no;
asm     PUSH AX
        outp(port_no+1,_AL);
asm     POP  AX
        _AL=_AH;
        outp(port_no+4,_AL);
        outp(port_no+2,3);

        NON();
        inp(8);
        NON();

asm     POP  DX
asm     POP  AX
  }

void INI_DMA()                  /* Direct Memory Access Program */
  {
asm     CLI
asm     PUSH AX
asm     PUSH DX

        DIS_INT();
        DIS_DMA();
asm     OUT  CMASK_REG,AL       /* Clear bit Mask = Enable DMA 4 channel */
```

```
asm     MOV  AL,D_MODE          /* Program Mode Register */
asm     OUT  MODE_REG,AL
asm     MOV  AL,COMM_DAT        /* Program Command Register */
asm     OUT  COMM_REG,AL

        rec_no=0;
        SET_DMA();
        _AX=buff_no;
asm     OUT  FL_FF,AL
asm     OUT  BASEWC_REG,AL
asm     MOV  AL,AH
asm     OUT  BASEWC_REG,AL

asm     POP  DX
asm     POP  AX
asm     STI
  }

void INI_EMS()                  /* Initial EMS */
  {

asm     CLI
asm     PUSH AX
asm     PUSH BX
asm     PUSH DX

        _AH=0x40;               /* CHECK EMS */
        geninterrupt(0x67);
        if(_AH!=0)
         {
asm       PUSH AX
          err=" LIM ";
          goto err_EMS;
         }

        _AH=0x42;
```

```
            geninterrupt(0x67);
            if(_AH!=0)
             {
asm            PUSH AX
               err=" Det.";
               goto err_EMS;
             }
            free_p=_BX;tot_p=_DX;

            _AH=0x41;
            geninterrupt(0x67);
            if(_AH!=0)
             {
asm            PUSH AX
               err="Start";
               goto err_EMS;
             }
            EMS_SEG=_BX;
            goto succ_EMS;

err_EMS:
asm         POP  AX
            dat=_AH;
            map_page=0xFF;

succ_EMS:
asm         POP  DX
asm         POP  BX
asm         POP  AX
asm         STI
   }

void SET_EMS()                          /* Mapping page */
   {

asm         PUSH AX
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
asm     PUSH BX
asm     PUSH DX

        _AH=0x43;
        _BX=al_page;
        geninterrupt(0x67);
        p_handle=_DX;
        if(_AH!=0)
         {
asm       PUSH AX
          err="Alloc";
          goto err_Map;
         }

        _AL=0;
        _BX=0;
        for(i=0;i<=al_page-1;i++)
         { _AH=0x44;
           _DX=p_handle;
           geninterrupt(0x67);
           if(_AH!=0)
            {
asm                 PUSH AX
                    err=" Map ";
                    goto err_Map;
            }
           _AL++;
           _BX++;
         }
        for(i=0;i<=al_page-1;i++)
          for(offset=0;offset<=0x3FFF;offset++)
            pokeb(EMS_SEG,(i*0x4000)+offset,0x80);
        map_page=0;
        goto succ_Map;

err_Map:
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
asm     POP  AX
        dat=_AH;
        map_page=0xFF;

succ_Map:
asm     POP  DX
asm     POP  BX
asm     POP  AX
  }

void RST_EMS()
  {
asm     PUSH AX

        DIS_DMA();
        _AH=0x45;
        _DX=p_handle;
        geninterrupt(0x67);

asm     POP  AX
  }

void STEP_MOTOR(unsigned char dat,unsigned int m_loop)
  {     if(dat==1)
         { for(i=1;i<=m_loop;i++)
            { if((inp(c_port+1)&2)==2)
               continue;
              outp(c_port,5);
              for(m_relay=0;m_relay<=0x3FFF;m_relay++);
              if((inp(c_port+1)&2)==2)
                continue;
              outp(c_port,6);
              for(m_relay=0;m_relay<=0x3FFF;m_relay++);
              if((inp(c_port+1)&2)==2)
                continue;
              outp(c_port,10);
```

```
            for(m_relay=0;m_relay<=0x3FFF;m_relay++);
            if((inp(c_port+1)&2)==2)
              continue;
            outp(c_port,9);
            for(m_relay=0;m_relay<=0x3FFF;m_relay++);
          }
        }
      if(dat==0xFE)
        { for(i=1;i<=m_loop;i++)
          { if((inp(c_port+1)&1)==1)
              continue;
            outp(c_port,9);
            for(m_relay=0;m_relay<=0x3FFF;m_relay++);
            if((inp(c_port+1)&1)==1)
              continue;
            outp(c_port,10);
            for(m_relay=0;m_relay<=0x3FFF;m_relay++);
            if((inp(c_port+1)&1)==1)
              continue;
            outp(c_port,6);
            for(m_relay=0;m_relay<=0x3FFF;m_relay++);
            if((inp(c_port+1)&1)==1)
              continue;
            outp(c_port,5);
            for(m_relay=0;m_relay<=0x3FFF;m_relay++);
          }
        }
      outp(c_port,0);
  }

void PLOT_GRAPH(unsigned int offset,unsigned int color_no)
  {  setviewport(2,152,415,413,0);
      k=offset;
      for(x=1;x<=414;x++)
        { y=0;
          for(no=1;no<=4;no++)
```

```
                { buff=peekb(EMS_SEG,k);
                  y=y+buff;
                  k++;
                }
               y=y/4;
               putpixel(x,261-y,color_no);
              }
   }

void INI_FILM()
   {
film_1:    setcolor(9);
           setviewport(0,422,417,479,0);
           clearviewport();
           rectangle(0,0,417,57);
           text1="Program Initial Film";
           text2="Press <\030> step up <\031> step down";
           text3="Press <Enter> veries intensity <ESC> quit";
           setcolor(14);
           outtextxy(100,10,text1);
           outtextxy(10,25,text2);
           outtextxy(10,40,text3);
           do
            { _AH=0;
              geninterrupt(0x16);
              buff=_AH;
              dat=_AL;
              m_loop=50;
              switch(buff)
                { case 0x48    : STEP_MOTOR(1,m_loop);
                                 break;
                  case 0x50    : STEP_MOTOR(0xFE,m_loop);
                                 break;
                  default      : NON();
                }
            } while((buff!=0x1C)&&(dat!=0x1B));
```

```
            if(dat==0x1B)
               goto film_0;
            setcolor(0);
            outtextxy(10,25,text2);
            outtextxy(10,40,text3);
            text2="Press <Spacebar> show data <Enter> accept";
            setcolor(14);
            outtextxy(10,25,text2);

            rec_no=0;
            INI_DMA();
            SETSCAN(2);
            offset=0;
            do
              { ENA_INT();
                outp(c_port+1,0xFF);
                do
                  buff=inp(8);
                while((buff&0x8)!=0x8);
                gotoxy(74,1);
                printf(" ON  ");
                DIS_DMA();
                SET_DMA();
                PLOT_GRAPH(offset,15);
                do
                  buff=getch();
                while((buff!=0x20)&&(buff!=0xD));
                clearviewport();
              } while(buff!=0xD);
            goto film_1;
film_0: clearviewport();
            SETSCAN(0);
   }

void GET_FILE()
   {        do
```

```
            { gotoxy(54,11);
              printf("Input file name");
              gotoxy(54,12);
              printf("->");
              printf("                    ");
              gotoxy(56,12);
              gets(file_name);
              fd=_creat(file_name,0);
              if(fd!=-1)
                { gotoxy(54,13);
                  printf("                    ");
                }
              else
                { gotoxy(54,13);
                  printf("Change File name");
                }

            } while(fd==-1);
          close(fd);
    }

void GET_DATA()
    {   dat=0;no=0;
          do
           { buff=getch();
             if(z==1)
               { if((buff>=0x30)&&(buff<=0x39))
                          { buff=buff-0x30;
                            printf("%d",buff);
                            dat=dat+buff;
                            no++;
                          }
                        if(buff==0x0D)
                          { dat=dat/10;
                            no++;
                          }
```

```
                    if(buff==8)
                     { dat=0;
                       no--;
                       printf("\b \b");
                     }
                  }
                 else
                  { if(buff==0x0D)
                     { dat=0xFF;
                       no=2;
                       continue;
                     }
                    if((buff>0x30)&&(buff<=0x39))
                     { buff=buff-0x30;
                       printf("%d",buff);
                       buff=buff*10;
                       no++;
                       dat=dat+buff;
                     }
                  }
                } while(no<2);
   }

void GET_LINE()
   {  line_max=0x7FFF/buff_no;
          do
           { gotoxy(54,13);
             printf("Input No. line scan");
             gotoxy(54,14);
             printf("[<=%d default 5]:  \b\b",line_max);
             GET_DATA();
             if(dat==0xFF)
               line_no=5;
             else
               line_no=dat;
           } while((line_no>line_max)||(line_no==0));
```

```
}

void GET_PROFILE()
{  do
        { gotoxy(54,15);
          printf("Input No. profile");
          gotoxy(54,16);
          printf("[<=%d default 13]:  \b\b",line_max);
          GET_DATA();
          if(dat==0xFF)
            profile=13;
          else
            profile=dat;
        } while((profile>line_max)||(profile==0));
}

void INI_DATA()
{  line_max=0x7FFF/buff_no;
   setcolor(9);
   setviewport(0,422,417,479,0);
   clearviewport();
   rectangle(0,0,417,57);
   text1="Program Initial Data";
   setcolor(14);
   outtextxy(100,35,text1);
   setcolor(14);
   setviewport(420,150,639,479,0);
   clearviewport();
   rectangle(0,0,219,329);
   GET_FILE();
   GET_LINE();
   GET_PROFILE();
   clearviewport();
}

int  WRITE_FILE(unsigned sc)                    /* Ret fn= -1 Err. */
```

```
{   buff=(sc)/100;
        if(buff>0)
         { buff=buff+0x30;
           z=write(fd,&buff,1);
           if(z==-1)
             goto wri_0;
         }

        buff=((sc)%100-(sc)%10)/10;
        if((buff>0)||((sc)/100>0)) /* &&(buff==0))) */
         { buff=buff+0x30;
           z=write(fd,&buff,1);
           if(z==-1)
             goto wri_0;
         }

        buff=(sc)%10;
        buff=buff+0x30;
        z=write(fd,&buff,1);
        if(z==-1)
          goto wri_0;
        buff=0x0D;
        z=write(fd,&buff,1);
        if(z==-1)
          goto wri_0;
        buff=0x0A;
        z=write(fd,&buff,1);
        if(z==-1)
          goto wri_0;
        goto wri_1;

wri_0:  gotoxy(1,22);
        printf("Rec Data fail %d",z);
wri_1:  return(-1);

  }
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
void PLOT_GRAPH_ALL()
   {   setviewport(2,152,415,413,0);
            clearviewport();
            offset=0x8000;
            for(l=1;l<=profile;l++)
             { for(x=1;x<=410;x++)
                { z=0;
                  for(j=1;j<=4;j++)
                   { buff=peekb(EMS_SEG,offset);
                     z=z+buff;
                     offset++;
                    }
                  z=z/4;
                  dat=z;
                  j=WRITE_FILE(dat);
                  putpixel(x,261-dat,l);
                }
             }
   }

void CORRECT()
   {   setcolor(9);
            setviewport(0,422,417,479,0);
            clearviewport();
            rectangle(0,0,417,57);
            if(profile==0)
             { text1="Not Initial data";
               setcolor(12);
               outtextxy(100,35,text1);
               getch();
               goto cor_0;
             }
            text1="Program Correct Data";
            setcolor(14);
            outtextxy(100,35,text1);
```

```
setcolor(14);
setviewport(420,150,639,479,0);
clearviewport();
rectangle(0,0,219,329);
setviewport(2,152,415,413,0);
rec_no=0;m_loop=50;
INI_DMA();
SETSCAN(2);
offset=0x8000;
for(l=1;l<=profile;l++)
  { rec_no=0;
    gotoxy(54,11);
    printf("Profile %d",l);
    for(j=1;j<=line_no;j++)
      { ENA_INT();
        outp(c_port+1,0xFF);
        do
          buff=inp(8);
        while((buff&0x8)!=0x8);
        gotoxy(74,1);
        printf(" ON  ");
        rec_no=rec_no+buff_no+1;
        DIS_DMA();
        SET_DMA();
      }
   for(no=0;no<=buff_no;no++)
      { k=0;
        for(j=0;j<=line_no-1;j++)
          { buff=peekb(EMS_SEG,(j*0x678)+no);
            k=k+buff;
          }
        buff=k/line_no;
        pokeb(EMS_SEG,offset+no,buff);
      }
    clearviewport();
```

```
            PLOT_GRAPH(offset,15);
            offset=offset+buff_no+1;
            for(j=1;j<=5;j++)
              STEP_MOTOR(0xFE,m_loop);
           }
         SETSCAN(0);
         fd=_creat(file_name,0);
         if(fd=-1)
          { gotoxy(54,12);
            printf("Write File err.");
            getch();
            close(fd);
            goto cor_0;
           }
         PLOT_GRAPH_ALL();
         close(fd);
         gotoxy(54,12);
         printf("Write file complete");
         for(l=1;l<=profile;l++)
           { for(j=1;j<=5;j++)
               STEP_MOTOR(1,m_loop);
           }

cor_0:   clearviewport();
   }

void LOGO()
   {  int userfont;
         setcolor(11);
         setviewport(80,0,560,35,0);
         rectangle(0,0,480,35);
         setcolor(12);
         rectangle(5,5,475,30);
         setcolor(14);
         userfont=installuserfont("simp.chr");
         settextstyle(userfont,HORIZ_DIR,1);
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
            outtextxy(70,15,menu);
            setcolor(11);
            setviewport(0,0,639,479,0);
            rectangle(0,150,417,415);
    }

void interrupt INTROUT()                    /* Interupt Program */
    {
asm         CLI                             /* Disable Maskable Interupt */
asm         PUSH DS
asm         PUSH ES
asm         PUSH SI
asm         PUSH AX
asm         PUSH BX
asm         PUSH CX
asm         PUSH DX
asm         PUSHF

            ENA_DMA();
            DIS_INT();
            gotoxy(74,1);
            printf(" NMI ");
asm         MOV  AL,EOI
asm         OUT  P_ICWA,AL

asm         POPF
asm         POP  DX
asm         POP  CX
asm         POP  BX
asm         POP  AX
asm         POP  SI
asm         POP  ES
asm         POP  DS
asm         STI                             /* Enable Maskable Interupt */
    }
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
/* detects EGA or VGA cards */
int huge detectEGA(void)
{
  int driver, mode, sugmode = 0;
  detectgraph(&driver, &mode);
  if ((driver == EGA) || (driver == VGA))
    /* return suggested video mode number */
    return sugmode;
  else
    /* return an error code */
    return grError;
}

void main()
   {        clrscr();
            offset=0;
            INI_SCAN();
            SET_NMI();
            INI_EMS();
            if((map_page==0xFF)||(free_p<=2))
              goto chk_0;
            al_page=4;
            SET_EMS();
            if((map_page==0xFF)||(buff_no==0))
              goto chk_0;
            c_port=0x108;profile=0;line_no=0;
            gdriver=installuserdriver("SVGA256",detectEGA);
            gmode=3;
            initgraph(&gdriver,&gmode,"");
            y_max=getmaxy();
            x_max=getmaxx();
            for(i=0; i<63; i++)          /* create gray scale */
              setrgbpalette(i+192, i, i, i);
            if((x_max<639)||(y_max<349))
              { err="Initial Graphic EVGA error";
                dat=gmode;
```

```
                case 0x50   : no=20;
                              break;
                default     : no=0;
              }
            setcolor(0);
            outtextxy(20,y,pointer);
            y=y+no;
            if(y>80)
              y=20;
            if(y<20)
              y=80;
          } while(buff!=0x1C&&dat!=0x1B);
        if(dat==0x1B)
          goto chk_1;
        switch(y)
          { case 20     : INI_FILM();
                          break;
            case 40     : INI_DATA();
                          break;
            case 60     : CORRECT();
                          break;
            default     : goto chk_1;
          }
        if(err!='')
          goto chk_0;
        goto chk_2;

chk_1:  err=" Bye-Bye ";dat=0;
chk_0:  SETSCAN(0);
        RST_EMS();
asm     OUT  MAST_REG,AL
        RST_NMI();
        closegraph();
        clrscr();
        printf("%s %d \n",err,dat);
    }
```

```
            goto chk_0;
          }
        cleardevice();
        setbkcolor(0);
        LOGO();

chk_2:  setcolor(9);
        setviewport(0,422,417,479,0);
        clearviewport();
        rectangle(0,0,417,57);
        text1="Press <\030> or <\031> choice item and <Enter> accept";
        text2="File name : ";
        setcolor(14);
        outtextxy(10,15,text1);
        outtextxy(10,30,text2);
        outtextxy(100,30,file_name);
        setcolor(13);
        setviewport(100,40,545,140,0);
        clearviewport();
        rectangle(0,0,435,100);
        setcolor(10);
        outtextxy(40,20,choice1);
        outtextxy(40,40,choice2);
        outtextxy(40,60,choice3);
        outtextxy(40,80,choice4);
        y=20;err='';
        do
          { setcolor(64);
            outtextxy(20,y,pointer);
            _AH=0;
            geninterrupt(0x16);
            buff=_AH;
            dat=_AL;
            switch(buff)
              { case 0x48     : no=-20;
                                break;
```

ภาคผนวก ข

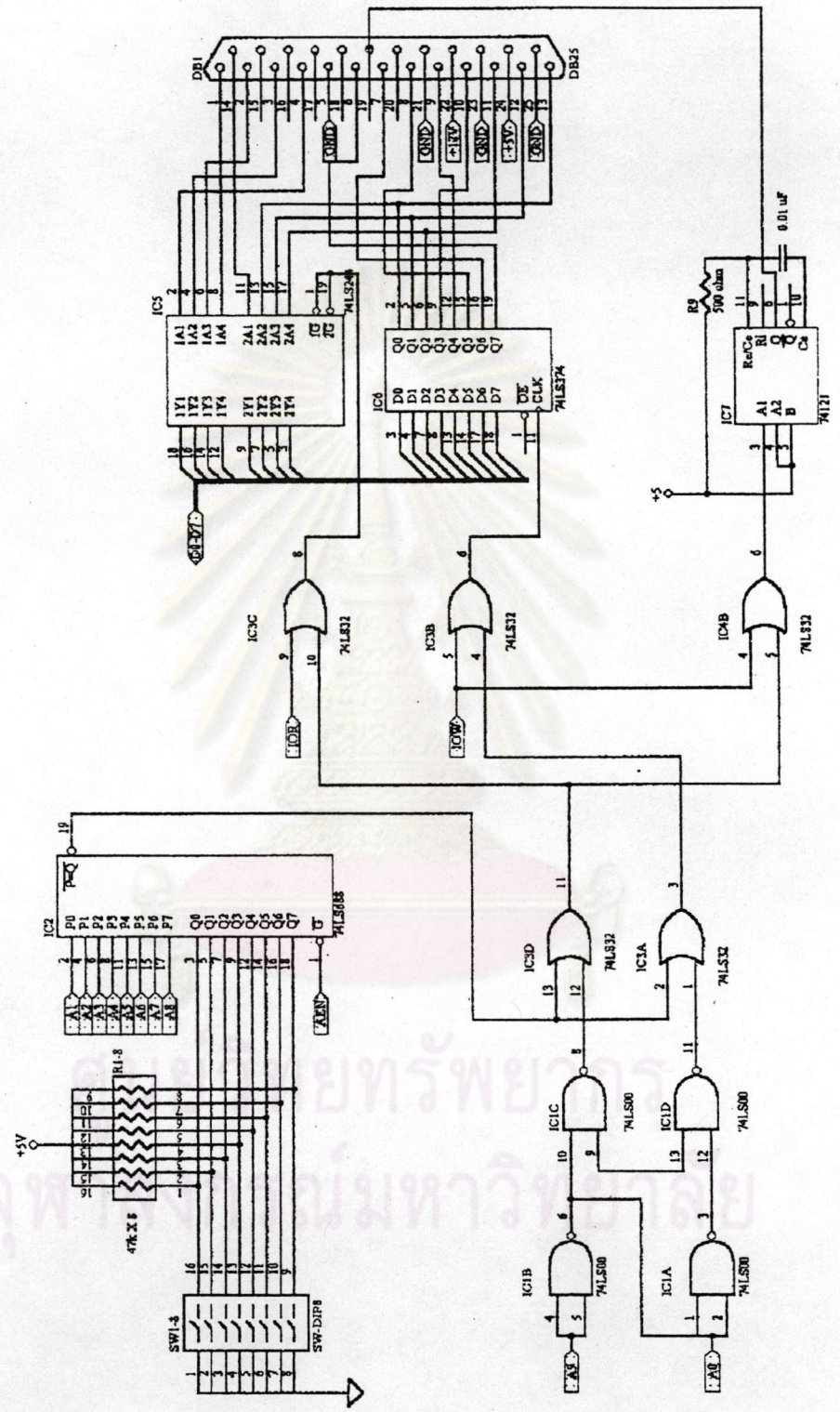
<u>วงจรที่ออกแบบใช้ในระบบเก็บข้อมูลโพรไฟล์</u>

ข.1 วงจรเชื่อมโยงระบบขับเคลื่อนฟิล์มกับไมโครคอมพิวเตอร์

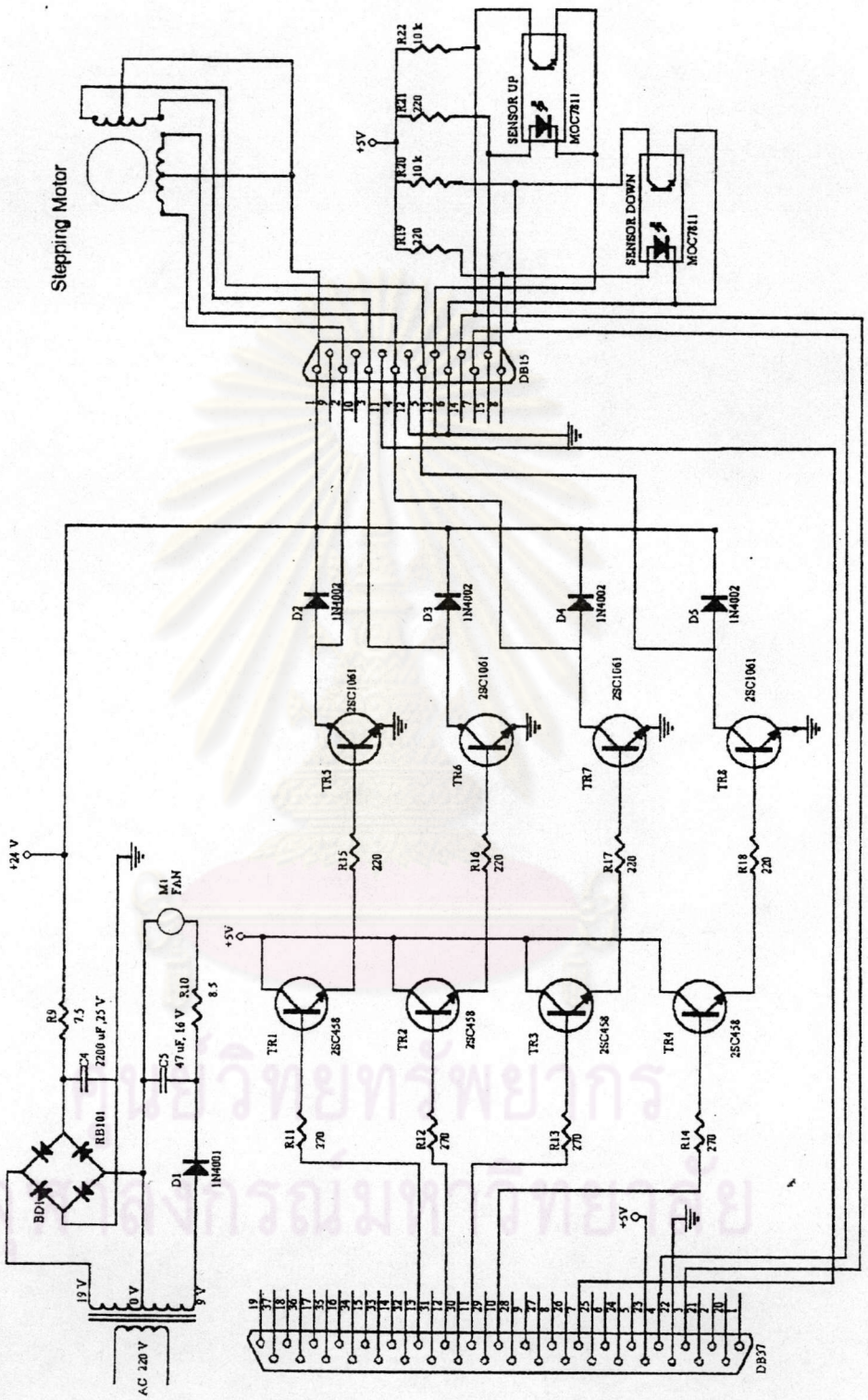
ข.2 วงจรขับสเต็ปปิงมอเตอร์

ข.3 วงจรควบคุมความเข้มแสง

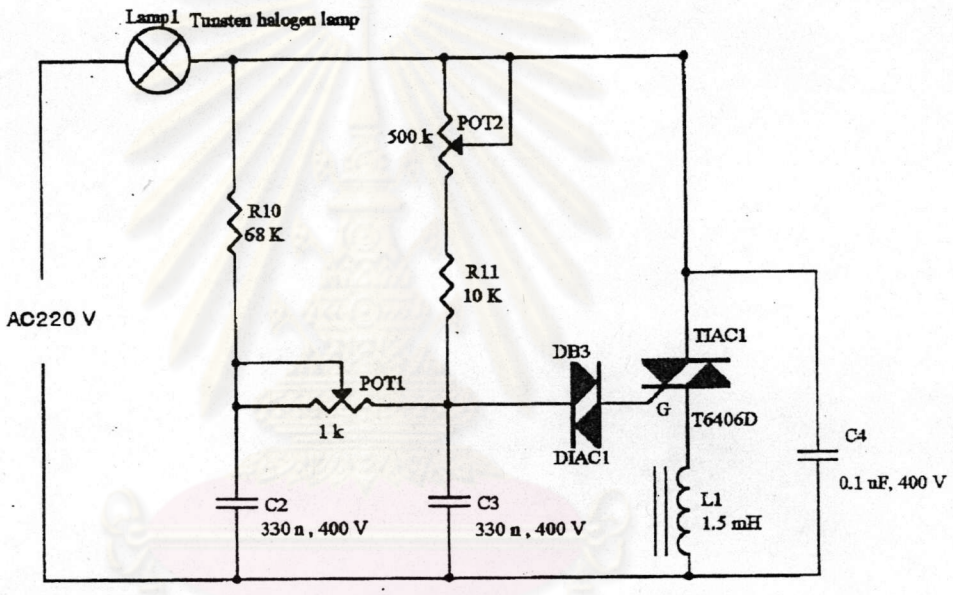
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป ข.1 วงจรเชื่อมโยงระบบเก็บข้อมูลโพรไฟล์



รูปที่ ข.2 แสดงวงจรขับสเต็ปปิงมอเตอร์



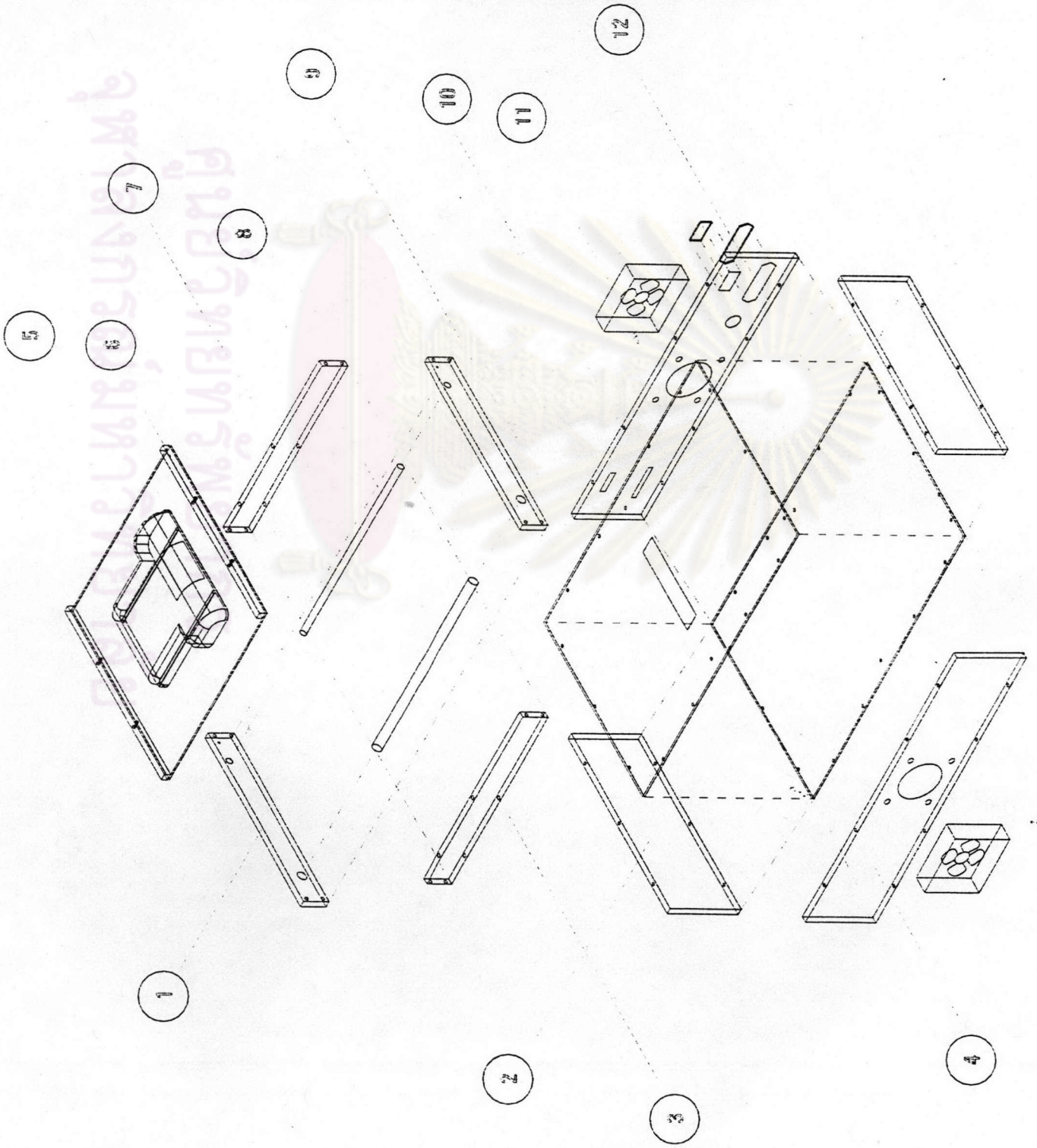
รูปที่ ข.3 แสดงวงจรควบคุมความสว่าง

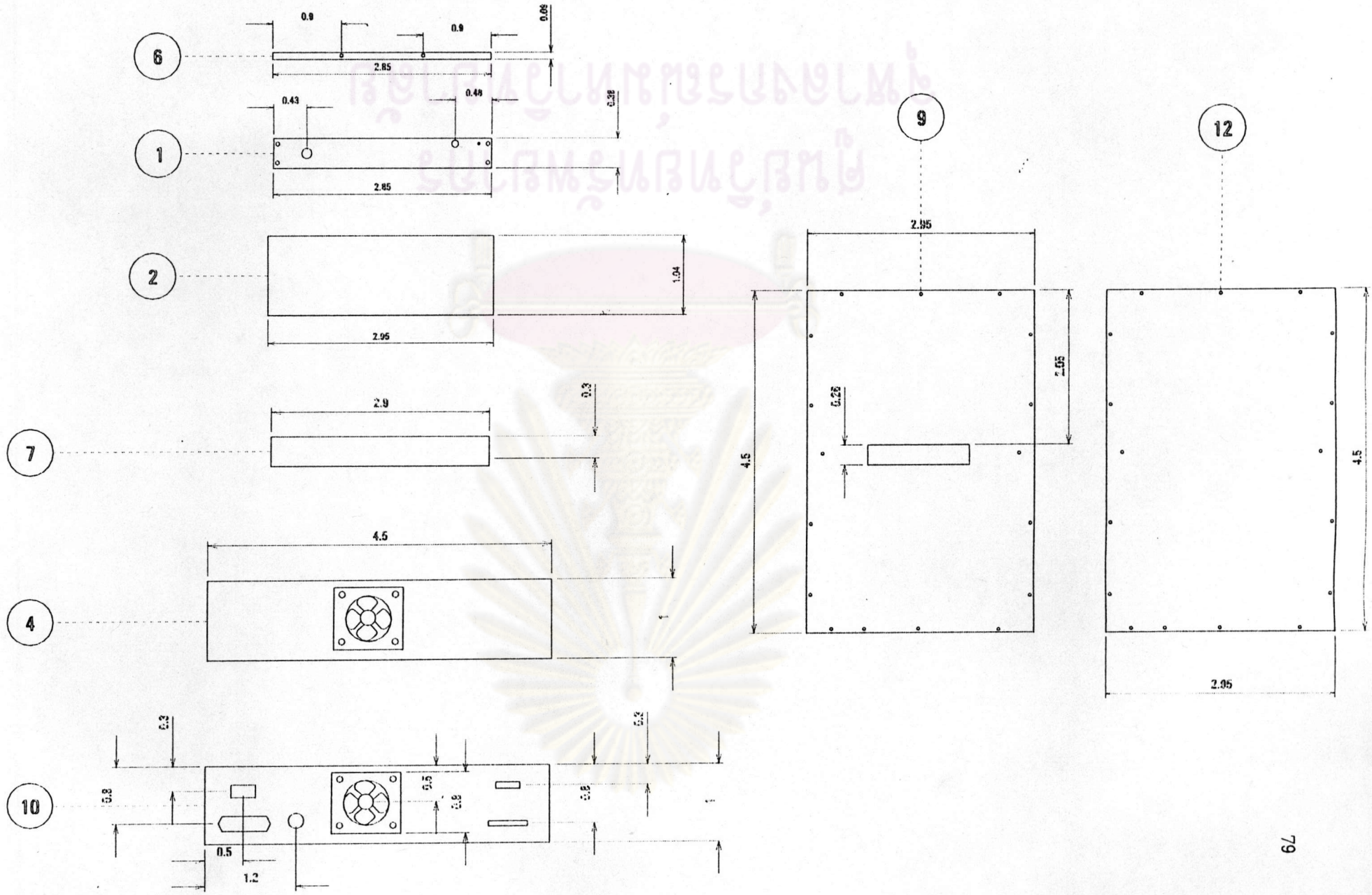
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ค

แผนภาพแสดงขนาดระบบเก็บข้อมูลโพรไฟล์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย


1
2
3
4
5
6
7
8
9
10
11
12


6
0.9
0.9
0.09
2.85
1
0.43
0.48
0.38
2.85
2
1.04
2.95
7
2.9
0.3
4
4.5
10
0.3
0.8
0.5
0.9
0.2
0.8
0.5
1.2
9
2.95
4.5
2.95
0.26
12
4.5
2.95

ประวัติผู้เขียน

นายนิวร ศรีคุณ เกิดวันที่ 25 มีนาคม พ.ศ. 2513 ที่เขตพระโขนง กรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตร์บัณฑิต สาขาอิเล็กโทรนิคส์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ในปีการศึกษา 2535 และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต ที่ ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2536

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย