

## รายการอ้างอิง

1. สมยศ ศรีสถิต, อรรถพร กัทรสุมันต์. การคำนวณสร้างภาพโทโมกราฟีด้วยเทคนิคฟิล์มเพื่อการตรวจสอบแบบไม่ทำลาย. สถาบันวิจัยและพัฒนาคณะวิศวกรรม คณะวิศวกรรม จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพมหานคร , 2538.
2. มงคล วรรณประภา. การพัฒนาระบบสแกนด้วยรังสีแกมมาเพื่อคำนวณการสร้างภาพโทโมกราฟีของเสาคอนกรีตเสริมเหล็ก. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชาวิศวกรรมเทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2536 .
3. ชีรวัฒน์ ประกอบผล. การพัฒนาระบบเก็บข้อมูลด้วยเทคนิคโทรทัศน์สำหรับคำนวณสร้างภาพโทโมกราฟี . วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชาวิศวกรรมเทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2537.
4. มนต์ มงคลสุข. พื้นฐานทางฟิสิกส์ของ CT และ MRI . กรุงเทพมหานคร, สำนักพิมพ์ไพศาลศิลป์การพิมพ์ , 2532
5. อภินันท์ มัชฌานนท์. การสื่อสารเส้นใยแสง . คณะวิศวกรรม สจล. กรุงเทพมหานคร, พิมพ์ครั้งที่ 8 , มกราคม 2540

## บรรณานุกรม

### ภาษาไทย

ไชยันต์ สุวรรณหิระศิริ. สแตมป์มอดูร์และการควบคุม. บริษัทซีเอ็ดยูเคชั่น จำกัด เซมิคอน-ดัคเตอร์อิเล็กทรอนิกส์, ฉบับที่ 116 พฤษภาคม 2535.

ธานินทร์ ถาวรศาสนวงศ์ และ ทินกร คู้ก. การอินเทอร์เฟซ IBM PC. กรุงเทพมหานคร : พิสิกส์เซ็นเตอร์การพิมพ์

ประวิทย์ เรืองไรรัตน์โรจน์. การพัฒนาระบบสร้างภาพตัดขวางด้วยไมโครคอมพิวเตอร์สำหรับงานถ่ายภาพด้วยรังสีในอุตสาหกรรม. วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2535 .

### ภาษาอังกฤษ

Don E.Bray, Roderic K. Stanley . Nondestructive Evaluation (A Tool in design , manufac turing and service) , McGraw Hill, 1989 printed in USA.

Glenn F. Knoll. Radiation Detection and Measurement, 2 nd ed. Singapore : John wiley & Sons, Inc. , 1989.

T.O white, Scintillating Fibres Nuclear Instruments and Methods in Physics Research A273 ,(1988) 820 - 825 North Holland , Amsterdar

Thomas S.Curry , James E.Dowdey, Robert C.Murry .Christensen's Introduction to the Physics of Diagnostic Radiology.. 3 rd Edition , Lea & Febiger , Philadelphia 1984 printed in USA.

ภาคผนวก

## ภาคผนวก ก

โปรแกรมสนับสนุนการทำงานของระบบเก็บข้อมูลสำหรับการคำนวณสร้างภาพโทโมกราฟีโดยใช้เส้นใยนำแสงเคลือบปลายด้วยซินทิลเลเตอร์

โปรแกรมการทำงานของระบบเก็บข้อมูลสำหรับการคำนวณสร้างภาพโทโมกราฟีโดยใช้เส้นใยนำแสงเคลือบปลายด้วยซินทิลเลเตอร์ เขียนด้วยภาษา Quick Basic Version 4 ทั้งหมด

ตัวแปร

di	ขนาดเส้นผ่าศูนย์กลางวัตถุตัวอย่าง
hi	ความสูงของวัตถุตัวอย่าง
FS	ชื่อแฟ้มข้อมูลที่ไใช้บันทึก
MM%(raysum,profile)	ข้อมูลที่อ่านได้แต่ละจุด

```

DECLARE FUNCTION PopUpMennuss% (X1%, Y1%,
Handle%, head$, foot$, item$())
DECLARE FUNCTION KeyCode% ()
DECLARE FUNCTION PopUpMenu% (X1%, Y1%, Handle%,
head$, foot$, item$())
DECLARE FUNCTION PopUpMenuss% (X1%, Y1%,
Handle%, head$, foot$, item$())
DECLARE FUNCTION PopUpMenus% (X1%, Y1%,
Handle%, head$, foot$, item$())
DECLARE SUB DefinedMenu (X1%, Y1%, X2%, Y2%,
Menuflag%)
DECLARE SUB DefinedMenuss (X1%, Y1%, X2%, Y2%,
Menuflag%)
DECLARE SUB DefinedMenus (X1%, Y1%, X2%, Y2%,
Menuflag%)
DECLARE SUB DrawBox2 (X1%, Y1%, X2%, Y2%)
DECLARE SUB GetBox (X1%, Y1%, X2%, Y2%, Handle%)
DECLARE SUB PrintMenu (X1%, Y1%, First%, MaxItem%,
item$())
DECLARE SUB PutBox (X1%, Y1%, X2%, Y2%, Handle%)
OUT &H303, &H89
CLS
DEFSNG M
DIM MM%(200, 140)
DIM P(100, 140)
rr:

COLOR 1, 7
CONST wndfrg = 4, wndbkg = 3
CONST HEADFRG = 1, HEADBKG = 2
CONST FOOTFRG = 2, FOOTBKG = 3

CONST enter = 13, ESC = 27
CONST LEFTARW = 19200, RIGHTARW = 19712
CONST DOWNARW = 20480, UPARW = 18432
CONST HOMEARW = 18176, ENDARW = 20224
CONST PGUPARW = 18688, PGDNARW = 20736
CONST INSERT = 20992, DELETE = 21248

CONST MAXHANDLE = 16
CONST MAXCOLUMN = 80
CONST MAXROW = 25

DIM SHARED MENU$(3)
DIM SHARED MENU$(3)
DIM SHARED MENU$(2)

DIM SHARED Vertptr%
DIM V(7) AS INTEGER

dd:
SCREEN 0

CLS
COLOR wndfrg, wndbkg

PRINT STRING$(2000, 177);

head$ = "    Main Menu    "

DefinedMenu X1%, Y1%, X2%, Y2%, 1
J% = PopUpMenu%(25, 8, 12, head$, foot$, MENU$( ))
LOCATE 24, 1: PRINT " You select choice number"; J%;
SPACE$(81 - POS(0));
IF J% < 1 > 3 THEN BEEP: GOTO dd
IF J% = 1 THEN CLS : GOTO EE
IF J% = 2 THEN CLS : GOTO VIEWFILE
IF J% = 3 THEN SCREEN 0: END
GOTO dd

EE:
SCREEN 0

CLS
COLOR wndfrg, wndbkg

PRINT STRING$(2000, 177);

head$ = "    Step Increment    "

DefinedMenus X1%, Y1%, X2%, Y2%, 1
Z% = PopUpMenuss%(25, 8, 12, head$, foot$, MENU$( ))
LOCATE 24, 1: PRINT " You select choice number"; Z%;
SPACE$(81 - POS(0));
IF Z% < 1 > 2 THEN BEEP: GOTO EE
IF Z% = 1 THEN CLS
        rot = 15
        GOTO FF
END IF
IF Z% = 2 THEN CLS

```

```

        rot = 25
        GOTO FF
END IF
GOTO dd

FF:
SCREEN 0

CLS
COLOR wndfrg, wndbk

PRINT STRING$(2000, 177);

head$ = "      Angle step      "

DefinedMenus X1%, Y1%, X2%, Y2%, 1
L% = PopUpMenus%(25, 8, 12, head$, foot$, MENUSS())
LOCATE 24, 1: PRINT " You select choice number"; L%;
SPACE$(81 - POS(0));
IF L% < 1 > 3 THEN BEEP: GOTO FF
IF L% = 1 THEN CLS : GOTO ANGLE1
IF L% = 2 THEN CLS : GOTO ANGLE2
IF L% = 3 THEN CLS : GOTO ANGLE3
GOTO dd

***** Select Angle for rotate in each step*****
ANGLE1:

        PRINT " Select to rotate 1.8 degree per step"
        GOSUB MATERIAL
        GOSUB HOMEY
        GOSUB MOVEy
        GOSUB HOMEX
        GOSUB STARTPOINT
        GOTO rotate1

ANGLE2:

        PRINT " Select to rotate 3.6 degree per step"
        GOSUB MATERIAL
        GOSUB HOMEY
        GOSUB MOVEy
        GOSUB HOMEX
        GOSUB STARTPOINT

```

```

        GOTO rotate2

ANGLE3:

        PRINT "Select to rotate 7.2 degree per step"
        GOSUB MATERIAL
        GOSUB HOMEY
        GOSUB MOVEy
        GOSUB HOMEX
        GOSUB STARTPOINT
        GOTO rotate3

*****Material size*****

MATERIAL:
INPUT "Fill diameter size (cm.) : ", di
INPUT "Height of material (cm.) : ", hi
IF di > 10 THEN GOTO MATERIAL
IF hi > 10 THEN GOTO MATERIAL
'free = free distance (mm.) move to start point for scan'
free = (6.8 - (di / 2)) * 10
strpoint = 25 * free
PRINT "strpoint", strpoint
' 25 loops equal 1 mm.'
length = (2 + di) * 10
endpoint = 25 * length
PRINT "endpoint", endpoint
height = (hi * 10 * 50) / 2
' 50 loops equal 1 mm.'
RETURN

*****Save data *****

FILESAVE:

        CLS
        INPUT "Please fill a filename : ", F$
        IF F$ = "" THEN F$ = "CTSCAN"
        PRINT "While is saving data"
        OPEN F$ FOR OUTPUT AS #1
        FOR profile = 1 TO row
        FOR raysum = 1 TO length
        WRITE #1, MM%(raysum, profile)
        NEXT raysum
        NEXT profile
        CLOSE 1
        PRINT "Already saved "
        FOR late = 1 TO 20: NEXT late

```

```

GOTO dd

*****Move step-y to sensor position*****
HOMEY:
' 1 lp = 4 step : 1 step = 1.8 degree '
delay = 20: lp = 10000
F = &H30: G = &H60: H = &HC0: P = &H90
FOR I = 1 TO lp
OUT &H300, F: FOR t = 1 TO delay: NEXT t
    op = (INP(770) AND 48)
    IF op = 32 THEN GOTO STP5
OUT &H300, G: FOR t = 1 TO delay: NEXT t
    op = (INP(770) AND 48)
    IF op = 32 THEN GOTO STP6
OUT &H300, H: FOR t = 1 TO delay: NEXT t
    op = (INP(770) AND 48)
    IF op = 32 THEN GOTO STP7
OUT &H300, P: FOR t = 1 TO delay: NEXT t
    op = (INP(770) AND 48)
    IF op = 32 THEN GOTO STP8
NEXT I
'FOR a = 1 TO 10000: NEXT a:
'GOTO start
STP5: PRINT "This is sensor Y-position for setting"
    F = &H90: G = &HC0: H = &H60: P = &H30
    RETURN
STP6: PRINT "This is sensor Y-position for setting"
    F = &H30: G = &H90: H = &HC0: P = &H60
    RETURN
STP7: PRINT "This is sensor Y-position for setting"
    F = &H60: G = &H30: H = &H90: P = &HC0
    RETURN
STP8: PRINT "This is sensor Y-position for setting"
    F = &HC0: G = &H60: H = &H30: P = &H90
    RETURN

*****Move Step-y to center position of y-axis*****

MOVEy:
delay = 20: lp = height
FOR I = 1 TO lp
    OUT &H300, F: FOR t = 1 TO delay: NEXT t
    OUT &H300, G: FOR t = 1 TO delay: NEXT t

OUT &H300, H: FOR t = 1 TO delay: NEXT t
OUT &H300, P: FOR t = 1 TO delay: NEXT t
NEXT I
RETURN

*****Move to Opto checking for step-X *****
HOMEX:
' 1 lp = 4 step : 1 step = 1.8 degree '
delay = 10: lp = 6250
A = &H3: B = &H6: C = &HC: D = &H9

FOR I = 1 TO lp
    op = (INP(770) AND 48)
OUT &H300, A: FOR t = 1 TO delay: NEXT t
    op = (INP(770) AND 48)
    IF op = 16 THEN GOTO STP1
OUT &H300, B: FOR t = 1 TO delay: NEXT t
    op = (INP(770) AND 48)
    IF op = 16 THEN GOTO STP2
OUT &H300, C: FOR t = 1 TO delay: NEXT t
    op = (INP(770) AND 48)
    IF op = 16 THEN GOTO STP3
OUT &H300, D: FOR t = 1 TO delay: NEXT t
    op = (INP(770) AND 48)
    IF op = 16 THEN GOTO STP4
NEXT I
'FOR a = 1 TO 10000: NEXT a:
STP1: PRINT "This is sensor X-position for setting"
    A = &H9: B = &HC: C = &H6: D = &H3
    RETURN
STP2: PRINT "This is sensor X-position for setting"
    A = &H3: B = &H9: C = &HC: D = &H6
    RETURN
STP3: PRINT "This is sensor X-position for setting"
    A = &H6: B = &H3: C = &H9: D = &HC
    RETURN
STP4: PRINT "This is sensor X-position for setting"
    A = &HC: B = &H6: C = &H3: D = &H9
    RETURN

*****Move to starting point for scan*****
STARTPOINT:

```

```

' 1 lp = 4 step : 1 step = 1.8 degree '
delay = 10: lp = strpoint
FOR I = 1 TO lp
OUT &H300, A: FOR t = 1 TO delay: NEXT t
OUT &H300, B: FOR t = 1 TO delay: NEXT t
OUT &H300, C: FOR t = 1 TO delay: NEXT t
OUT &H300, D: FOR t = 1 TO delay: NEXT t
NEXT I
FOR J = 1 TO 100: NEXT J:
RETURN

***** Motor for Rotate Material every 1.8 deg*****

rotate1: FOR k = 1 TO 25: delay = 10
profile = (4 * (k - 1)) + 1
OUT &H301, &H3: FOR t = 1 TO
delay: NEXT t
GOSUB LEFT
profile = (4 * (k - 1)) + 2
OUT &H301, &H9: FOR t = 1 TO
delay: NEXT t
GOSUB RIGHT
profile = (4 * (k - 1)) + 3
OUT &H301, &HC: FOR t = 1 TO
delay: NEXT t
GOSUB LEFT
profile = (4 * (k - 1)) + 4
OUT &H301, &H6: FOR t = 1 TO
delay: NEXT t
GOSUB RIGHT
NEXT k
row = 100
GOTO FILESAVE
GOTO dd

***** Motor for Rotate Material every 3.6 deg*****

rotate2: FOR kk = 1 TO 25: delay = 10
profile = (2 * (kk - 1)) + 1
OUT &H301, &H3: FOR t = 1 TO
delay: NEXT t
OUT &H301, &H9: FOR t = 1 TO
delay: NEXT t
GOSUB LEFT
profile = (2 * (kk - 1)) + 2
OUT &H301, &HC: FOR t = 1 TO
delay: NEXT t
GOSUB RIGHT
NEXT kk
row = 50
GOTO FILESAVE
GOTO dd

***** Motor for Rotate Material every 7.2 deg*****

rotate3: FOR E = 1 TO 25: delay = 20
profile = E
OUT &H301, &H3: FOR t = 1 TO
delay: NEXT t
OUT &H301, &H9: FOR t = 1 TO
delay: NEXT t
OUT &H301, &HC: FOR t = 1 TO
delay: NEXT t
OUT &H301, &H6: FOR t = 1 TO
delay: NEXT t
route = profile MOD 2
IF route = 1 THEN GOSUB LEFT
IF route = 0 THEN GOSUB RIGHT
NEXT E
row = 25
GOTO FILESAVE
GOTO dd

*****Move to left side*****
LEFT:
CLS
FOR Z = 1 TO 50: NEXT Z
' 1 lp = 4 step : 1 step = 1.8 degree '
delay = 20: lp = length
FOR raysum = 1 TO lp
**25 loops equal 1 mm. for collect data 1 time**
FOR I = 1 TO rot
OUT &H300, A: FOR t = 1 TO delay:
NEXT t
OUT &H300, B: FOR t = 1 TO delay:
NEXT t

```



```

                OUT &H300, C: FOR t = 1 TO delay:
NEXT t
                OUT &H300, D: FOR t = 1 TO delay:
NEXT t
                NEXT I
                GOSUB ADCIN
                MM%(raysum, profile) = M
                GOSUB plotgraph
        NEXT raysum
                IF profile MOD 5 = 0 THEN GOSUB
CHECKxray
                RETURN
                FOR relay = 0 TO 100: NEXT relay
'*****Read data from moving Left*****'
ADCIN:   FOR nor = 1 TO 1000: NEXT nor
                M = 0
                Total = 0
                FOR Round = 1 TO 100
                        FOR t = 0 TO 10: NEXT t
                                count = (INP(772))
                                Total = count + Total
                        NEXT Round
                                M = Total / 100
                                PRINT "average("; profile;
";", raysum; ")="; M
                                RETURN
                                'FOR q = 1 TO 100: NEXT q:
'Relay for backward motion'
'*****Plot to graph*****'
plotgraph:
                SCREEN 9 '320 x 200 graphics
                COLOR 5, 8
                LINE (50, 30)-(50, 310)
                LINE (45, 300)-(600, 300)
                LOCATE 23, 30: PRINT "profile no."; profile
'*****Text-X*****'
                'FOR x = 0 TO plot STEP 5
                'LOCATE 23, 6 + 2 * x: PRINT x
                'NEXT x
                *****Y-Scale*****
                LOCATE 19, 2: PRINT 50
                LOCATE 16, 2: PRINT 100
                LOCATE 13, 2: PRINT 150
                LOCATE 10, 2: PRINT 200
                LOCATE 7, 2: PRINT 250
                *****Point-Show*****
                U = 300 - (.84 * M)
                W = 50 + (3.5 + (((140 - length) / 10) *
.5)) * raysum
                CIRCLE (W, U), 1, 14
                'FOR j = 1 TO 500: NEXT j
                RETURN
                *****Move Free to origin start on Right side*****
                RIGHT:
                '1 lp = 4 step : 1 step = 1.8 degree'
                delay = 20: lp = length
                CLS
                FOR raysum = lp TO 1 STEP -1
                        GOSUB ADCIN
                        MM%(raysum, profile) = M
                        GOSUB plotgraph
                        FOR Z = 1 TO 50: NEXT Z
                                ***25 loops equal 1 mm. for collect data 1
                                time**
                                FOR I = 1 TO rot
                                        OUT &H300, C: FOR t = 1 TO delay:
                                        NEXT t
                                        OUT &H300, B: FOR t = 1 TO delay:
                                        NEXT t
                                        OUT &H300, A: FOR t = 1 TO delay:
                                        NEXT t
                                        OUT &H300, D: FOR t = 1 TO delay:
                                        NEXT t
                                        NEXT I
                                        NEXT raysum
                                        IF profile MOD 5 = 0 THEN GOSUB
CHECKxray
                                        RETURN

```

```

*****Check X-ray shooting times*****
CHECKxray:
    PRINT "Press any key to continue"
    WHILE INKEY$ = ""
    WEND
    RETURN

***** VIEW PROFILE*****
VIEWFILE:
LOCATE 12, 25: LINE INPUT "file name : ", SFNS$
LOCATE 13, 25: LINE INPUT "No. of Profile : ", NP$
NP = VAL(NP$)
LOCATE 14, 25: LINE INPUT "No. of ray-sum : ", MAT$
LOCATE 19, 40: PRINT "loading....Please wait"

MAT = VAL(MAT$)
Inth = 20 + MAT
OPEN SFNS FOR INPUT AS #1
    FOR I% = 1 TO NP
    FOR J% = 1 TO MAT
    INPUT #1, A$
    P(I%, J%) = VAL(A$)
    GOSUB plotgraph2
    NEXT J%
    'CLS
    NEXT I%
CLOSE #1
WHILE INKEY$ = ""
WEND
CLS

GOTO dd
plotgraph2:
SCREEN 9
COLOR 5, 8
LINE (50, 30)-(50, 310)
LINE (45, 300)-(600, 300)
UU = 300 - (.84 * P(I%, J%))
WW = 50 + (3.5 + (((140 - Inth) / 10) * .5)) * J%
CIRCLE (WW, UU), 1, 14
    RETURN

SUB DefinedMenu (X1%, Y1%, X2%, Y2%, Menuflag%)
STATIC
    MENU$(1) = " New Scan Profile      "
    MENU$(2) = " View Profile          "
    MENU$(3) = " Exit program          "

    X1% = 2: X2% = X1% + LEN(MENU$(1)) + 1
    Y1% = 2: Y2% = Y1% + UBOUND(MENU$(1)) + 1

END SUB

SUB DefinedMenu (X1%, Y1%, X2%, Y2%, Menuflag%)
STATIC
    MENU$(1) = " 1.8 degree per step      "
    MENU$(2) = " 3.6 degree per step      "
    MENU$(3) = " 7.2 degree per step      "

    X1% = 2: X2% = X1% + LEN(MENU$(1)) + 1
    Y1% = 2: Y2% = Y1% + UBOUND(MENU$(1)) + 1

END SUB

REM $DYNAMIC
'
'FUNCTION:  Draws double-lines box.
'
'PARAMETERS:  X1%, Y1'  Upper left x, y coordinate.
'              X2%, Y2%  Lower right x, y coordinate.
'
SUB DrawBox2 (X1%, Y1%, X2%, Y2%) STATIC
LOCATE Y1%, X1%: PRINT CHR$(201); STRING$(X2% -
X1% - 1, 205); CHR$(187);
FOR I% = Y1% + 1 TO Y2% - 1
    LOCATE I%, X1%: PRINT CHR$(186);
    LOCATE , X2%: PRINT CHR$(186);
NEXT I%

```

```
LOCATE Y2%, X1%: PRINT CHR$(200), STRING$(X2% -
X1% - 1, 205); CHR$(188);
```

```
END SUB
```

```
REM $STATIC
```

```
SUB GetBox (X1%, Y1%, X2%, Y2%, Handle%) STATIC
```

```
  DIM Reg AS Regtype
```

```
  FOR I% = Y1% TO Y2%
```

```
    FOR J% = X1% TO X2%
```

```
      LOCATE I%, J%
```

```
    ' Reg.ax = &H800
```

```
    NEXT J%
```

```
  NEXT I%
```

```
END SUB
```

```
REM $DYNAMIC
```

```
' F_NCTION: Returns a unique integer for any key pressed.
```

```
FUNCTION KeyCode% STATIC
```

```
DO
```

```
  P$ = INKEY$
```

```
LOOP UNTIL P$ <> ""
```

```
KeyCode% = CVAL(P$ + CHR$(0))
```

```
END FUNCTION
```

```
REM $STATIC
```

```
FUNCTION PopUpMennuss% (X1%, Y1%, Handle%, head$,
foot$, item$()) STATIC
```

```
  Arrnum% = UBOUND(item$)
```

```
  HeadLen% = LEN(head$)
```

```
  FootLen% = LEN(foot$)
```

```
  IF HeadLen% THEN
```

```
    FirstRow% = Y1% + 2
```

```
  IF FootLen% THEN
```

```
    Y2% = Y1% + Arrnum% + 3
```

```
  ELSE
```

```
    Y2% = Y1% + Arrnum% + 2
```

```
  END IF
```

```
ELSE
```

```
  FirstRow% = Y1% + 1
```

```
  IF FootLen% THEN
```

```
    Y2% = Y1% + Arrnum% + 2
```

```
  ELSE
```

```
    Y2% = Y1% + Arrnum% + 1
```

```
  END IF
```

```
END IF
```

```
X2% = X1% + LEN(item$(1)) + 1
```

```
GetBox X1%, Y1%, X2%, Y2%, Handle%
```

```
COLOR wndfrg, wndbkg
```

```
DrawBox2 X1%, Y1%, X2%, Y2%
```

```
PrintMenu X1% + 1, FirstRow%, 1, Arrnum%, item$()
```

```
COLOR HEADFRG, HEADBKG
```

```
LOCATE Y1% + 1, (X1% + X2% - HeadLen% + 1) \ 2:
```

```
PRINT head$;
```

```
COLOR FOOTFRG, FOOTBKG
```

```
LOCATE Y2% - 1, (X1% + X2% - FootLen% + 1) \ 2: PRINT
foot$;
```

```
Ptr% = 1
```

```
DO
```

```
  COLOR wndbkg, wndfrg
```

```
  LOCATE FirstRow% + Ptr% - 1, X1% + 1: PRINT
```

```
MENUSS$(Ptr%);
```

```
  k% = KeyCode%
```

```
  COLOR wndfrg, wndbkg
```

```
  LOCATE FirstRow% + Ptr% - 1, X1% + 1: PRINT
```

```
MENUSS$(Ptr%)
```

```
  SELECT CASE k%
```

```
  CASE ESC
```

```
    PopUpMennuss% = 0
```

```
  CASE enter
```

```
    PopUpMennuss% = Ptr%
```

```
  CASE HOMEARW, PGUPARW
```

```
    Ptr% = 1
```

```
  CASE ENDARW, PGDNARW
```

```
    Ptr% = Arrnum%
```

```

CASE UPARW
    IF Ptr% > 1 THEN
        Ptr% = Ptr% - 1
    ELSE
        Ptr% = Arnum%
    END IF
CASE DOWNARW
    IF Ptr% < Arnum% THEN
        Ptr% = Ptr% + 1
    ELSE
        Ptr% = 1
    END IF
CASE ELSE
    SOUND 900, 1
    SOUND 700, 1
END SELECT

LOOP UNTIL k% = enter OR k% = ESC
PutBox X1%, Y1%, X2%, Y2%, Handle%

END FUNCTION

FUNCTION PopUpMenu% (X1%, Y1%, Handle%, head$,
foot$, item$()) STATIC

    Arnum% = UBOUND(item$)

    HeadLen% = LEN(head$)
    FootLen% = LEN(foot$)
    IF HeadLen% THEN
        FirstRow% = Y1% + 2
    IF FootLen% THEN
        Y2% = Y1% + Arnum% + 3
    ELSE
        Y2% = Y1% + Arnum% + 2
    END IF
ELSE
    FirstRow% = Y1% + 1
    IF FootLen% THEN
        Y2% = Y1% + Arnum% + 2
    ELSE
        Y2% = Y1% + Arnum% + 1
    END IF
END IF
X2% = X1% + LEN(item$(1)) + 1

    GetBox X1%, Y1%, X2%, Y2%, Handle%
    COLOR wndfrg, wndbkg
    DrawBox2 X1%, Y1%, X2%, Y2%
    PrintMenu X1% + 1, FirstRow%, 1, Arnum%, item$()

    COLOR HEADFRG, HEADBKG
    LOCATE Y1% + 1, (X1% + X2% - HeadLen% + 1) \ 2:
PRINT head$;
    COLOR FOOTFRG, FOOTBKG
    LOCATE Y2% - 1, (X1% + X2% - FootLen% + 1) \ 2: PRINT
foot$;

    Ptr% = 1
    DO
        COLOR wndbkg, wndfrg
        LOCATE FirstRow% + Ptr% - 1, X1% + 1: PRINT
MENU$(Ptr%);

        k% = KeyCode%

        COLOR wndfrg, wndbkg
        LOCATE FirstRow% + Ptr% - 1, X1% + 1: PRINT
MENU$(Ptr%)

    SELECT CASE k%
    CASE ESC
        PopUpMenu% = 0
    CASE enter
        PopUpMenu% = Ptr%
    CASE HOMEARW, PGUPARW
        Ptr% = 1
    CASE ENDARW, PGDNARW
        Ptr% = Arnum%
    CASE UPARW
        IF Ptr% > 1 THEN
            Ptr% = Ptr% - 1
        ELSE
            Ptr% = Arnum%
        END IF
    CASE DOWNARW
        IF Ptr% < Arnum% THEN
            Ptr% = Ptr% + 1
        ELSE
            Ptr% = 1
        END IF
    END SELECT

```

```

CASE ELSE
    SOUND 900, 1
    SOUND 700, 1
END SELECT

LOOP UNTIL k% = enter OR k% = ESC
PutBox X1%, Y1%, X2%, Y2%, Handle%

END FUNCTION

FUNCTION PopUpMenus% (X1%, Y1%, Handle%, head$,
foot$, item$()) STATIC
    Arrnum% = UBOUND(item$)

    HeadLen% = LEN(head$)
    FootLen% = LEN(foot$)
    IF HeadLen% THEN
        FirstRow% = Y1% + 2
        IF FootLen% THEN
            Y2% = Y1% + Arrnum% + 3
        ELSE
            Y2% = Y1% + Arrnum% + 2
        END IF
    ELSE
        FirstRow% = Y1% + 1
        IF FootLen% THEN
            Y2% = Y1% + Arrnum% + 2
        ELSE
            Y2% = Y1% + Arrnum% + 1
        END IF
    END IF
    X2% = X1% + LEN(item$(1)) + 1

    GetBox X1%, Y1%, X2%, Y2%, Handle%
    COLOR wndfrg, wndbkg
    DrawBox2 X1%, Y1%, X2%, Y2%
    PrintMenu X1% + 1, FirstRow%, 1, Arrnum%, item$()

    COLOR HEADFRG, HEADBKG
    LOCATE Y1% + 1, (X1% + X2% - HeadLen% + 1) \ 2:
PRINT head$:
    COLOR FOOTFRG, FOOTBKG
    LOCATE Y2% - 1, (X1% + X2% - FootLen% + 1) \ 2: PRINT
foot$:

```

```

Ptr% = 1
DO
    COLOR wndbkg, wndfrg
    LOCATE FirstRow% + Ptr% - 1, X1% + 1: PRINT
MENU$(Ptr%);

    k% = KeyCode%

    COLOR wndfrg, wndbkg
    LOCATE FirstRow% + Ptr% - 1, X1% + 1: PRINT
MENU$(Ptr%)

    SELECT CASE k%
    CASE ESC
        PopUpMenus% = 0
    CASE enter
        PopUpMenus% = Ptr%
    CASE HOMEARW, PGUPARW
        Ptr% = 1
    CASE ENDARW, PGDNARW
        Ptr% = Arrnum%
    CASE UPARW
        IF Ptr% > 1 THEN
            Ptr% = Ptr% - 1
        ELSE
            Ptr% = Arrnum%
        END IF
    CASE DOWNARW
        IF Ptr% < Arrnum% THEN
            Ptr% = Ptr% + 1
        ELSE
            Ptr% = 1
        END IF
    CASE ELSE
        SOUND 900, 1
        SOUND 700, 1
    END SELECT

    LOOP UNTIL k% = enter OR k% = ESC
    PutBox X1%, Y1%, X2%, Y2%, Handle%

END FUNCTION

REM $DYNAMIC
'
' FUNCTION: Prints one page menu.

```

```

'
' PARAMETERS: X1%, Y1% Upper left x, y coordinate.
'           First% Number of choice to print.
'           MaxItem% Number of choices per menu list.
'           item$( ) String containing list of word representing
choices.
'

```

```
SUB PrintMenu (X1%, Y1%, First%, MaxItem%, item$( ))
```

```
STATIC
```

```
COLOR wndfg, wndbk
```

```
FOR I% = 0 TO MaxItem% - 1
```

```
    LOCATE Y1% + I%, X1%: PRINT item$(First% + I%);
```

```
NEXT I%
```

```
END SUB
```

```

'
' FUNCTION: PutWindow at a specified place.
'

```

```
' PARAMETERS: X1%, Y1% Upper left x, y coordinate.
```

```
'           X2%, Y2% Lower right x, y coordinate.
```

```
'           Handle% Window number.
```

```
SUB PutBox (X1%, Y1%, X2%, Y2%, Handle%) STATIC
```

```
' DIM Reg AS Regtype
```

```
Reg.cx = 1
```

```
FOR I% = Y1% TO Y2%
```

```
    FOR J% = X1% TO X2%
```

```
        LOCATE I%, J%
```

```
        ' Reg.ax = &H900 + BoxSpC%(i% - Y1% + 1, j% - X1% +
1, 1, Handle%)
```

```
        ' Reg.bx = BoxSpC%(i% - Y1% + 1, j% - X1% + 1, 2,
Handle%)
```

```
        TNERRUPT &H10, Reg, Reg
```

```
    NEXT J%
```

```
NEXT I%
```

```
END SUB
```

## ภาคผนวก ข

วงจร interface ระหว่างระบบสแกนกับไมโครคอมพิวเตอร์ และระบบนิวเคลียร์กับไมโครคอมพิวเตอร์

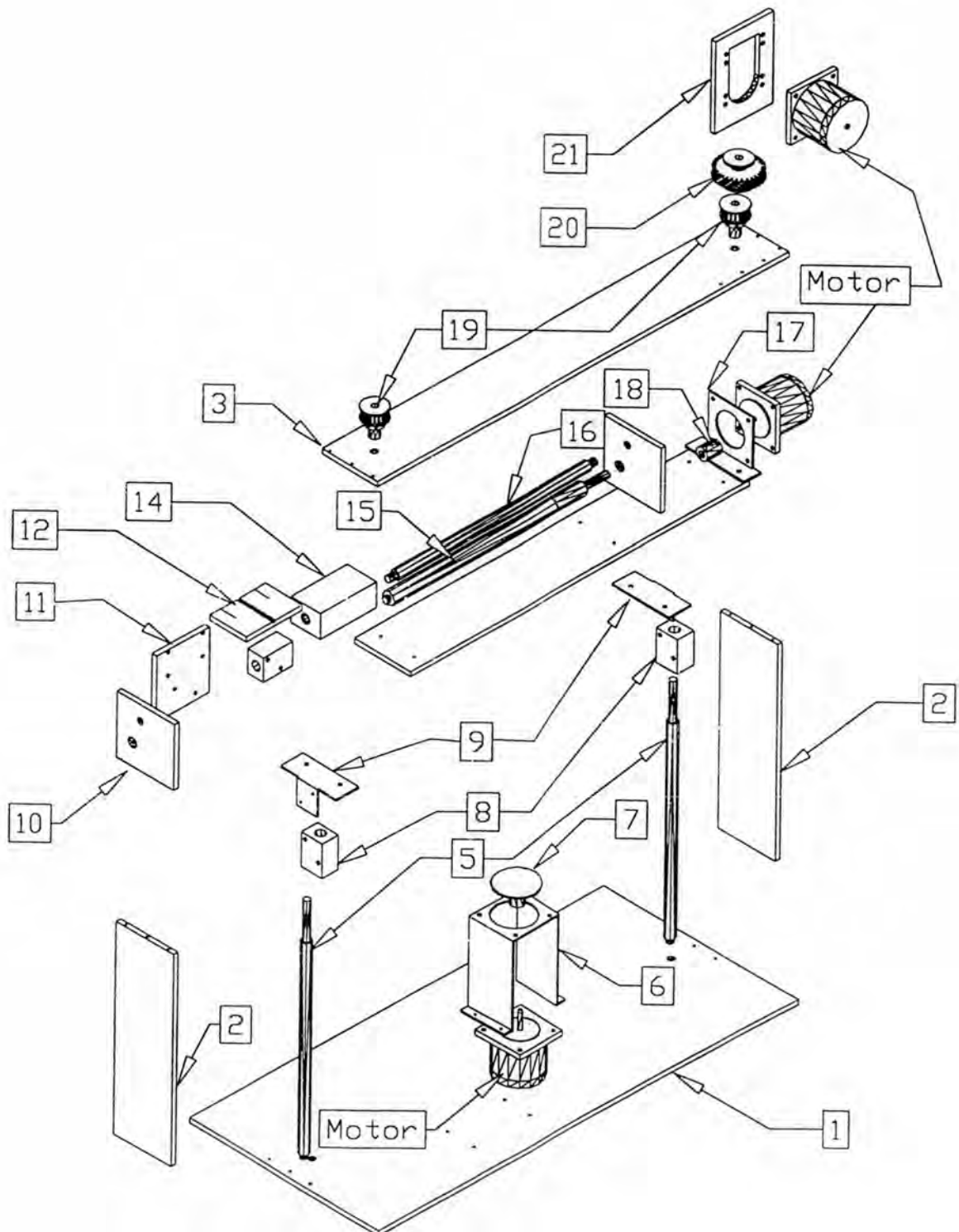




ภาคผนวก ก

แผนภาพแสดงขนาดระบบเก็บข้อมูลโทรไฟล์

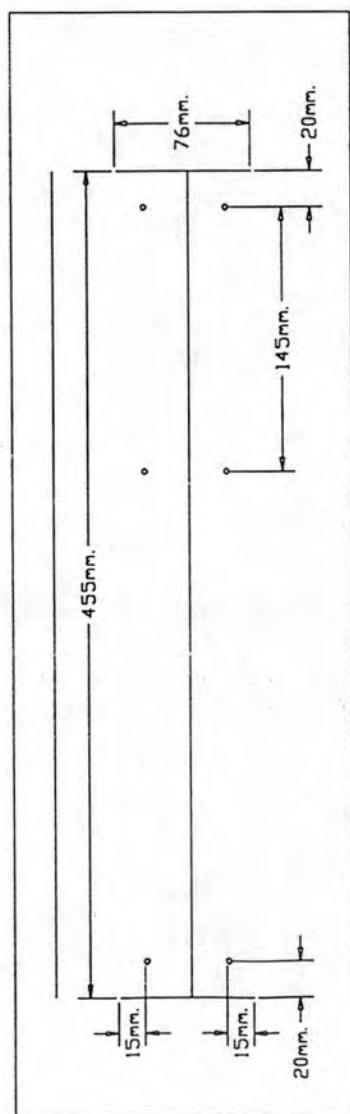




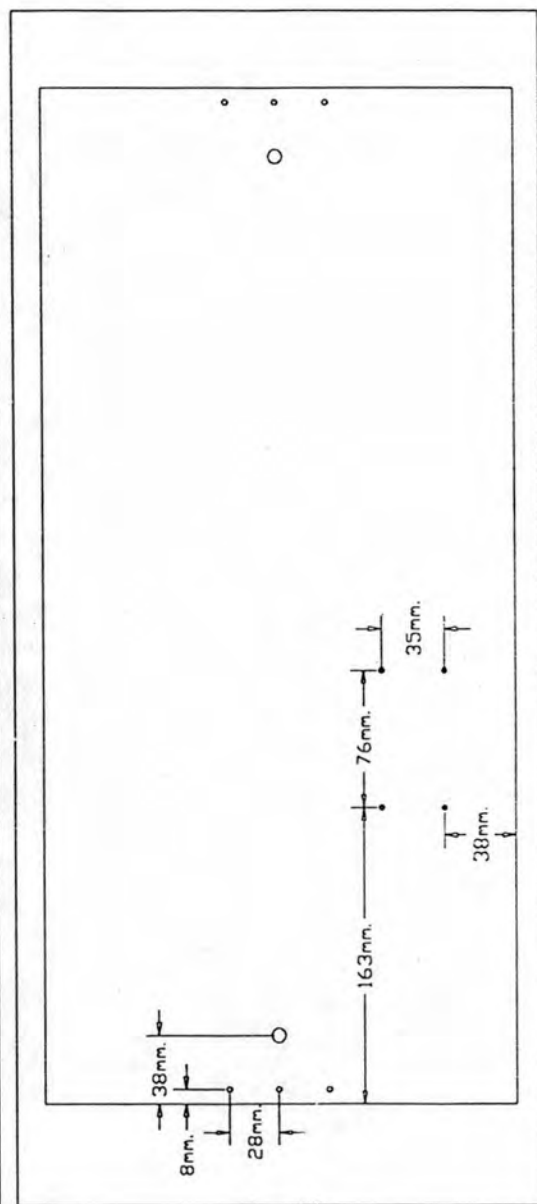
รูปที่ ค.2 ภาพรายละเอียดชิ้นประกอบของข้อมูลโพรไฟล์



3

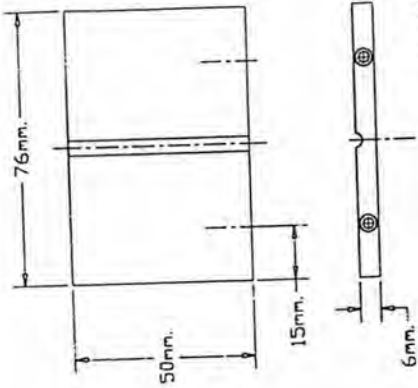
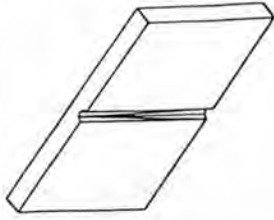


4

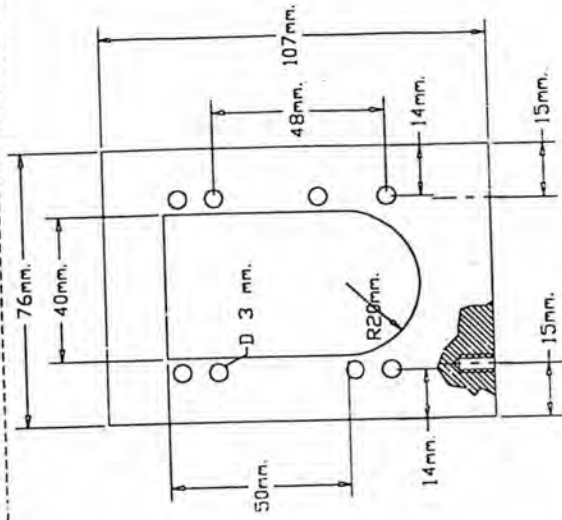
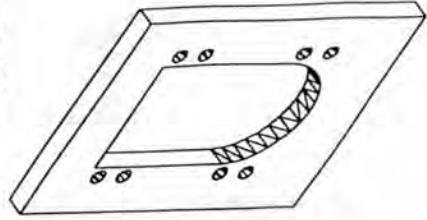


1

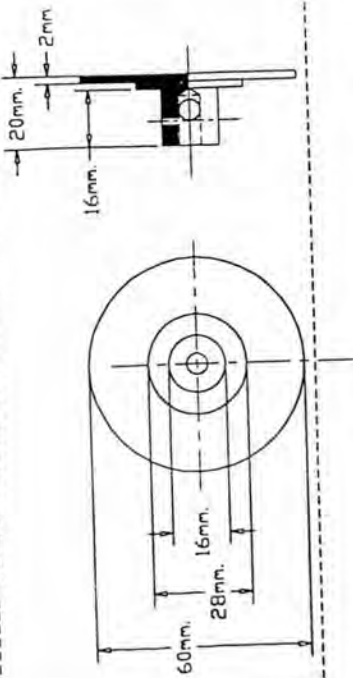
12



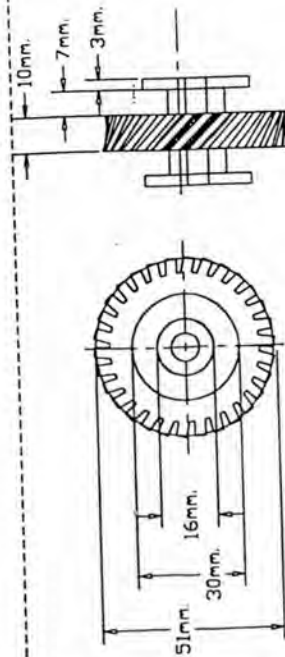
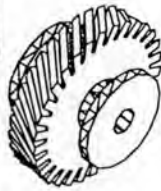
21



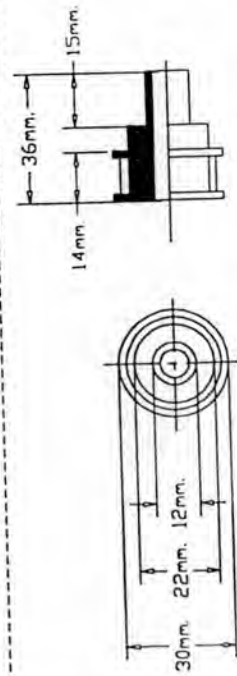
7



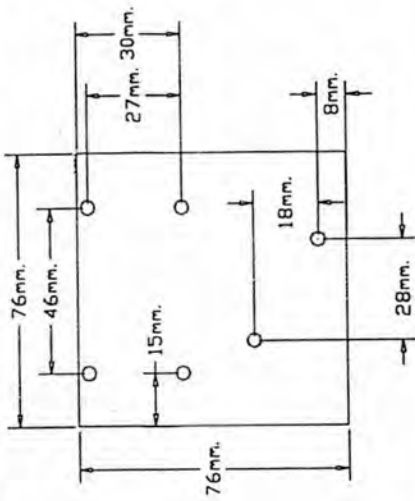
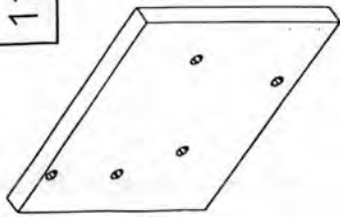
20



19



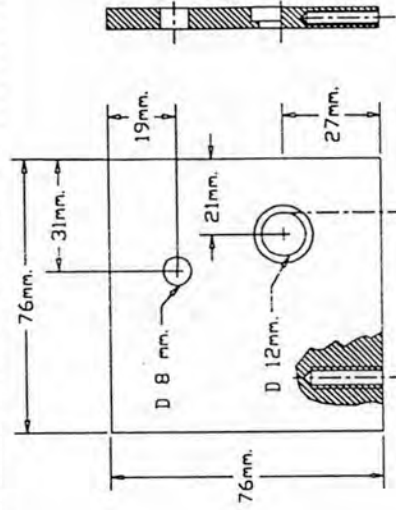
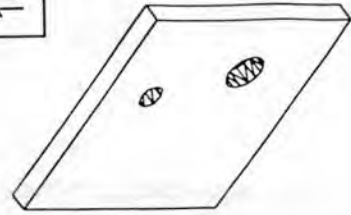
11



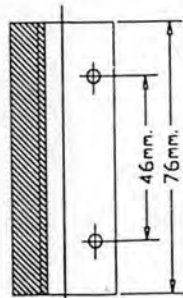
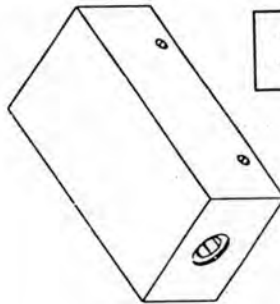
18



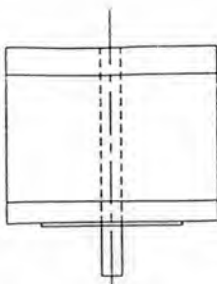
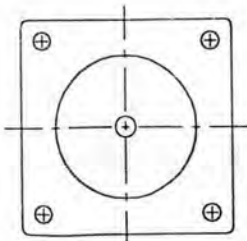
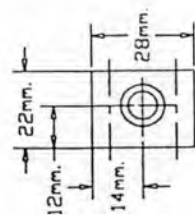
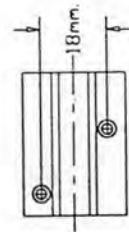
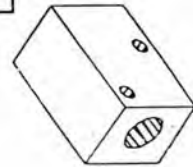
10

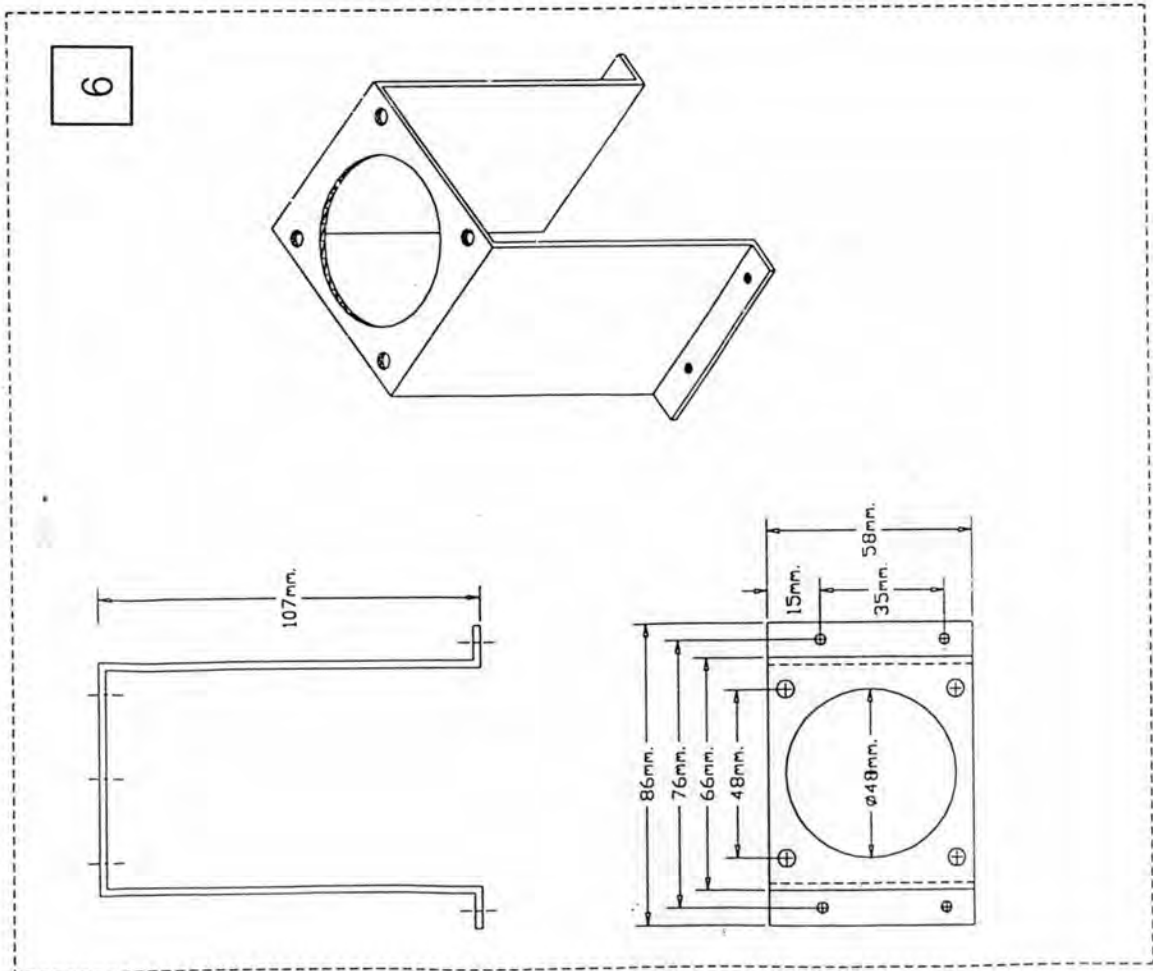
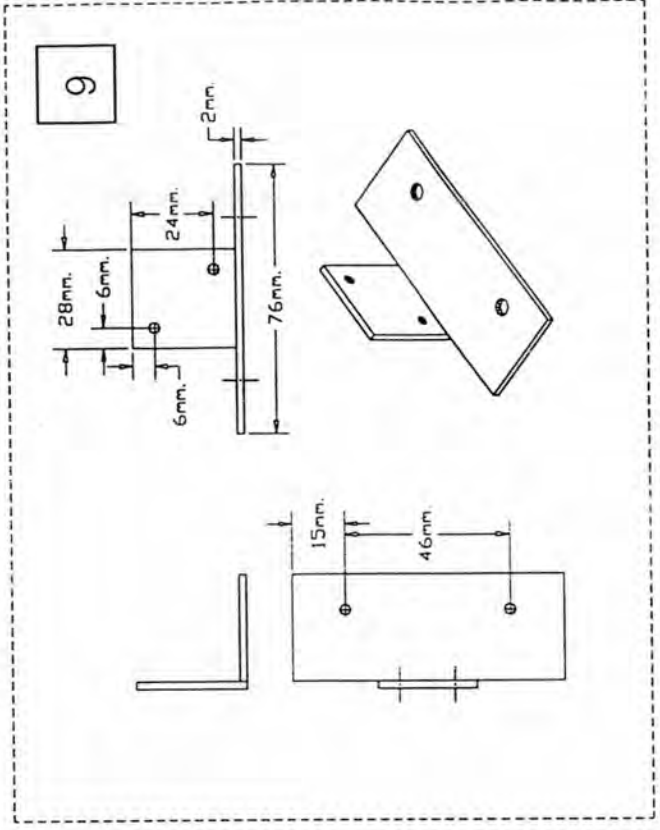
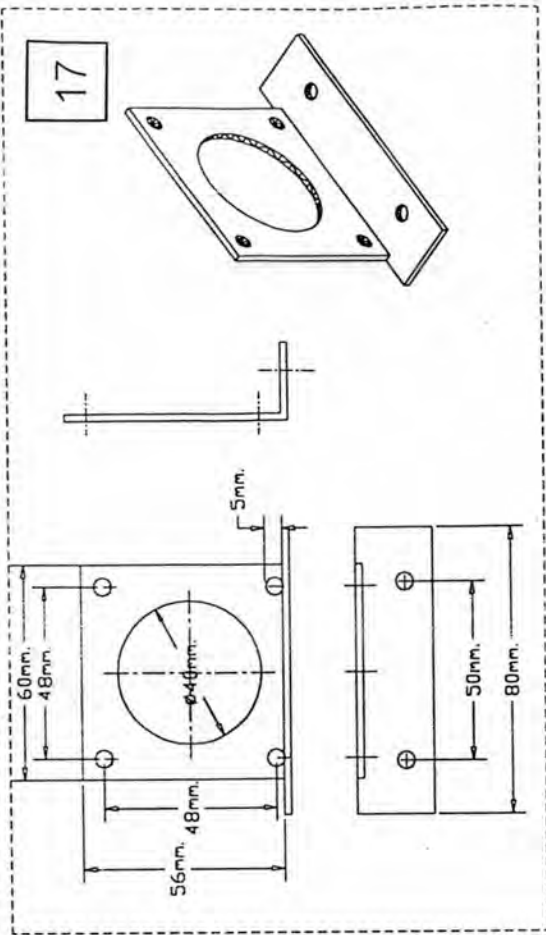


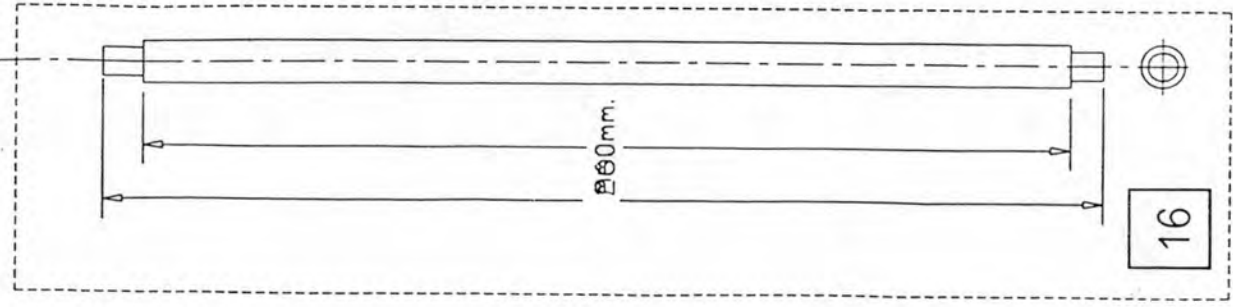
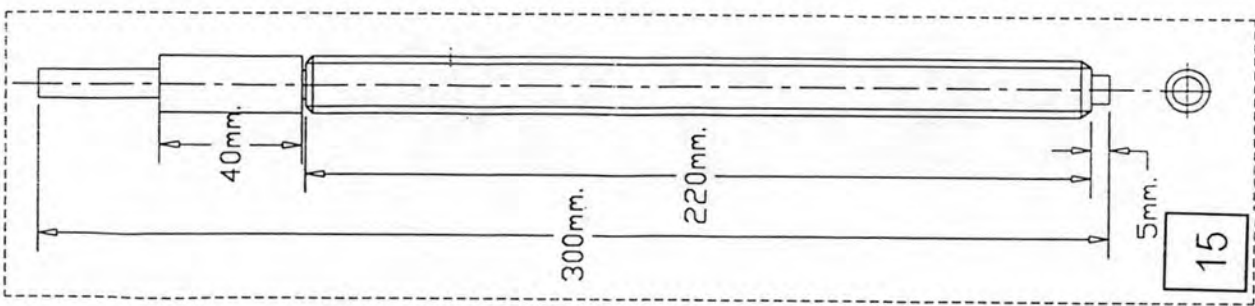
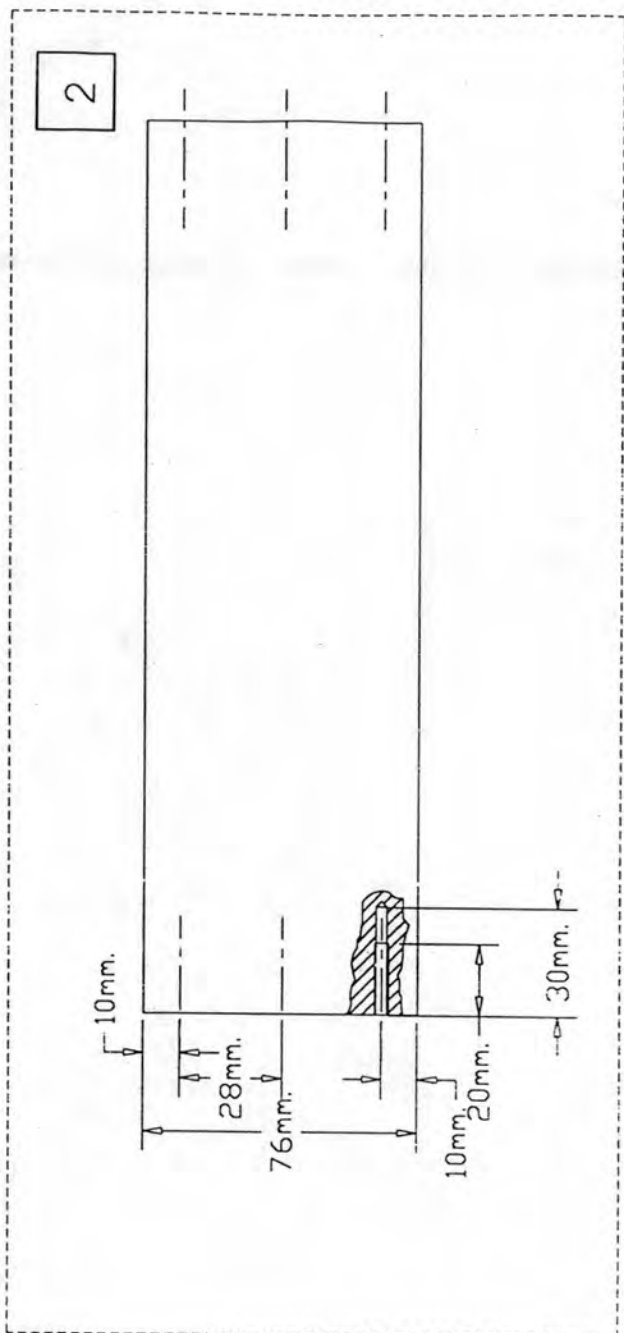
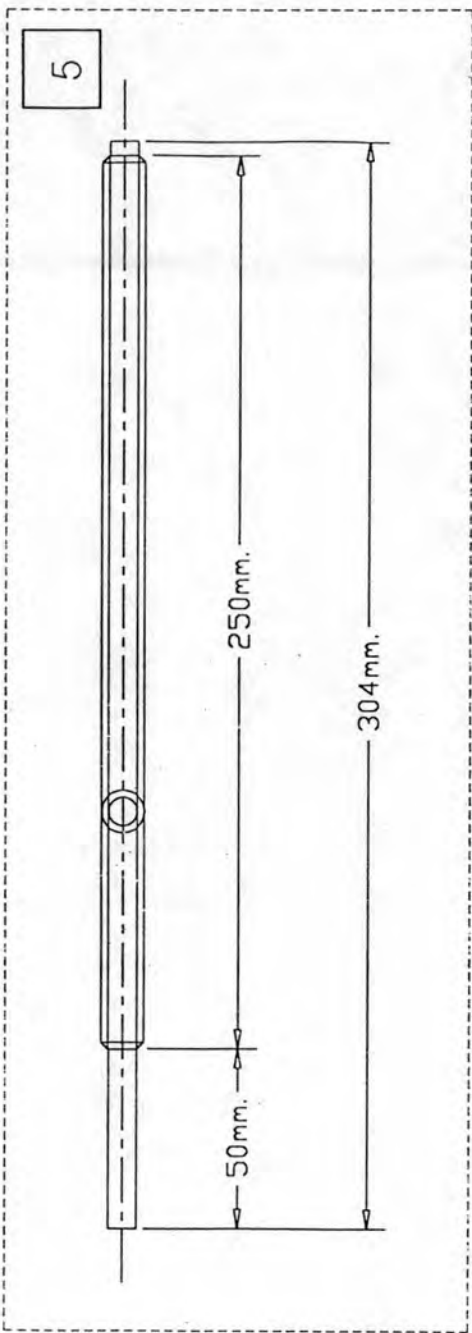
14



8









## ประวัติผู้เขียน

นายวัชรพงษ์ ปถมพานิชย์ เกิดวันที่ 16 ธันวาคม พ.ศ. 2515 ที่เขตบางกอกน้อย กรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรี ภาควิชาฟิสิกส์ประยุกต์(โซลิตอนเทคโนโลยีทรอนิกส์) คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปี พ.ศ.2536 จากนั้นเข้าศึกษาต่อที่ ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2537