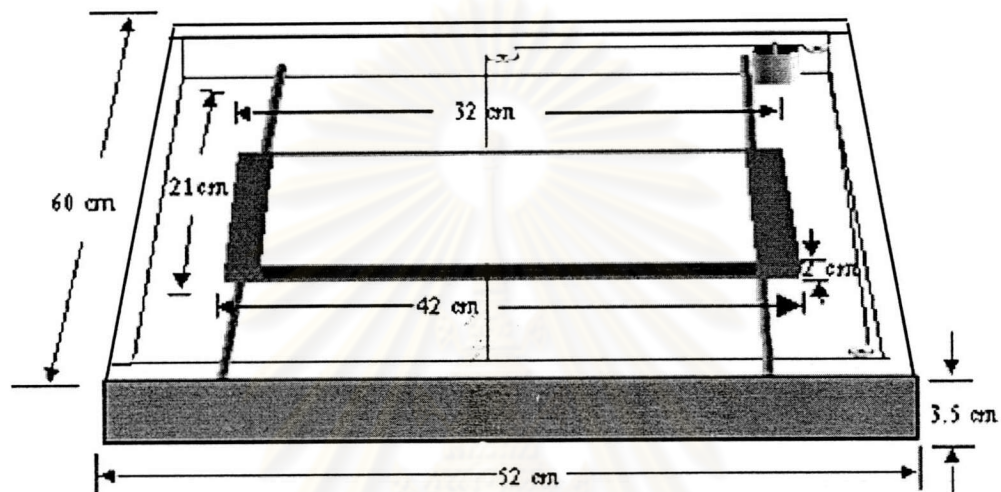


รายการอ้างอิง

1. C.J.Borkowski and M.K.Kopp.New type of position-sensitive detectors of ionizing radiation using risetime measurement. Review of scientific instruments,1968.
2. อติศักดิ์ ปัญญาบุษ.หัววัดรังสีพรอพอร์ชันแนลชนิดไวต่อตำแหน่งราคาประหยัด.วิทยานิพนธ์ปริญญาโทมหาบัณฑิต ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย,2541.
3. Wilhelm Conrad Röntgen.A new kind of ray. Physical Medical Society of Wurzburg ,1895
4. Eugene P.Bertin.Principles and practice of x-ray spectrometric analysis.Plenum Press-New York,1975.
5. สุวิทย์ ปุณณชัยยะ.Industrial x-ray Radiographic Equipments .เอกสารประกอบการสอนการตรวจวัดรังสีนิวเคลียร์และอุปกรณ์วัดนิวเคลียร์. ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, ม.ป.ป..
6. ASTM Committee on Radiographic Inspection.Radiographic Inspection , 1972
7. Ron Jenkins R.W. Gould Dale Gedcke.Quantitative x-ray spectrometry.Macle Dekker,Inc.,1995.
8. V.K.Zworykin,G.A.Morton Television,2nd Edition John Wiley & Sons,New york,1954.
9. นเรศร์ จันทน์ขาว. การถ่ายภาพด้วยรังสีเอกซ์และรังสีแกมมาในงานอุตสาหกรรม.เอกสารการสอนวิชา 2111626 การประยุกต์ใช้รังสีและไอโซโทปในทางอุตสาหกรรม.ภาควิชานิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2540.
10. Knoll G.F.Radiation Detection and Measurement.New York : John Wiley & Sons,Inc,1999.
11. C.J. Borkowski and M.K. Kopp. Some applications and properties of one-and two dimensional position-sensitive proportional counter,1971.
12. Dhananjay V.Gadre.Programming the parallel port.CMP Books,1998.

ภาคผนวก ก.
แบบระบบกลสำหรับสแกนสร้างภาพ



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข.

โปรแกรมควบคุมการทำงานของระบบ

:Line profiles

Dim CX, CY, CH

Dim YValue

Dim a(2000) As Long

Dim b(100, 2000) As Long

Dim k, m As Long

Dim h, z As Integer

Dim HighCount As Integer

Dim colorvalue As Long

Private Sub cmdLineExit_Click()

 LineProfile.Hide

 RemoteMCA.Show

End Sub

Private Sub cmdShowPic_Click()

On Error GoTo Err_cmdShowPic_Click

Dim InputData As String

Dim I, j, MyValue As Integer

Dim GraphFile As String

Dim n As Integer

 LP_Pic.Cls

 LP_Pic.DrawWidth = 1

 m = 0

 n = CInt(txtNumProfile.Text)

 For m = 1 To n

 I = 0

 GraphFile = "c:\lineprofile\data\" & txtFileName & m & ".txt" 'Text1.Text

 Open GraphFile For Input As #1

```

Do While Not EOF(1)
    Line Input #1, InputData
    MyValue = Val(InputData)
    b(m, I) = MyValue
    I = I + 1
Loop
For j = 0 To I
    k = b(m, j)
    h = HS1.Value
    z = HS2.Value
    If k < z Then
        k = 0
    Else
        k = k - z
    End If
    k = k * h
    colorvalue = RGB(k, k, k)
    If j > 300 Then
        LP_Pic.PSet ((j * 3) - 900, 720 - (m * 5)), colorvalue
    End If
Next j
Close #1
Next m
Exit_cmdShowPic_Click:
Exit Sub
Err_cmdShowPic_Click:
MsgBox Err.Description
Resume Exit_cmdShowPic_Click
End Sub
-----
Private Sub cmdShowProfile_Click()
On Error GoTo Err_cmdShowProfile_Click

```

```

Dim InputData As String
Dim j, MyValue As Integer
Dim GraphFile As String

LP_Pic.Cls
LP_Pic.DrawWidth = 2

'=====
I = 0
GraphFile = "c:\lineprofile\data\" & txtFileName & ".txt"
Open GraphFile For Input As #1
Do While Not EOF(1)
    Line Input #1, InputData
    MyValue = Val(InputData)
    a(I) = MyValue
    I = I + 1
Loop
'=====

For j = 0 To I
    k = a(j)
    h = CInt(HS1.Value)
    z = CInt(HS2.Value)
    If k < z Then
        k = 0
    Else
        k = k - z
    End If
    k = k * h
    colorvalue = RGB(k, k, k)
    LP_Pic.PSet (j + 10, 720), colorvalue
Next j
'=====

Max = 0
For j = 0 To I

```

```

k = a(j)
If k > Max Then
    Max = k
End If
Next j
'=====
YValue = Max / 600
For j = 0 To I
    k = a(j)
    If k < z Then
        k = 0
    Else
        k = k - z
    End If
    k = k * h
    LP_Pic.PSet (j + 10, 700 - (k / YValue)), RGB(0, 0, 255)
Next j
'=====
Close #1 ' Close file.
Exit_cmdShowProfile_Click:
    Exit Sub
Err_cmdShowProfile_Click:
    MsgBox Err.Description
    Resume Exit_cmdShowProfile_Click
End Sub
Private Sub Form_Load()
    HighCount = 100
End Sub
Private Sub HS1_Change()
    Label1.Caption = "LEVEL x " & HS1.Value

```

End Sub

Private Sub HS2_Change()

Label9.Caption = HS2.Value

End Sub

Private Sub HScroll1_Change()

LP_Pic.ScaleWidth = HScroll1.Value

Label4.Caption = HScroll1.Value

End Sub

Private Sub txtFileName_Change()

End Sub

Private Sub VScroll1_Change()

LP_Pic.ScaleHeight = VScroll1.Value

Label5.Caption = VScroll1.Value

End Sub

:MCA control

Option Explicit

Dim LineChannel(1153) As String

Dim Channel(1023) As Integer

Dim I, j, k, m As Integer

Dim InputData, StrData As String

Dim Chdata, Chdat, Charline As String

Dim GraphFile

Dim Linelen As Integer

Dim Collectstr(128) As String

Dim CHStart, CHStart2 As Boolean

Dim ABChannel(1153) As String

```

Dim YValue
Dim ScaleH, Max As Integer
Dim GainVal As Integer
Dim CHVal As Integer
Dim DataAddress, StatAddress, CtrlAddress As Integer
Dim ROR, ROL As Boolean
Dim Delaytime As Integer
Dim PCBusy As Boolean
Dim PCTime As Integer
Dim BatchStop As Integer

```

```
Private Sub DrawPic()
```

```
    Ln_Pic1.DrawWidth = 1
```

```
    Max = 0
```

```
    I = GainVal
```

```
    For k = 0 To (I - 1)
```

```
        If Channel(k) > Max Then
```

```
            Max = Channel(k)
```

```
        End If
```

```
    Next k
```

```
    YValue = Max / ScaleH
```

```
    For k = 0 To (I - 1)
```

```
        Ln_Pic1.PSet (k + 50, 869 - (Channel(k) / YValue)), RGB(255, 255, 0)
```

```
    Next k
```

```
    PCBusy = False
```

```
End Sub
```

```
Private Sub DataOut()
```

```
On Error GoTo DataOut_Err
```

```
    j = 0
```

```
    For I = 1 To (CHVal - 1)
```



```

If (I Mod 9) <> 1 Then
    Channel(j) = CInt(ABChannel(I))
    j = j + 1
End If
Next I
'=====
'For I = 0 To 1023
' Text1.Text = Text1.Text & "channel[" & I & "], count =" & Channel(I) & vbCrLf
' Text1.SelStart = Len(Text1.Text)
'Next I
DataOut_Err:
    txtStatus.Text = txtStatus.Text & Err.Description & " - Error number: " & Err.Number &
vbCrLf
    txtStatus.SelStart = Len(txtStatus.Text)
End Sub
-----
Private Sub cmdBatch_Click()
Dim loopbatch As Integer
Dim timetac As Integer
Dim SaveGraph
Dim stepone As Integer

BatchStop = False
loopbatch = 1
Do While Not (BatchStop)
    Timer1.Enabled = True
    PCBusy = True
'===== start collect =====
    Out DataAddress, 176
    Out DataAddress, 240
    LC6.Active = True
'Delay 1

```

```

Out DataAddress, 208
Out DataAddress, 240
LC6.Active = False

'===== timer =====
timetac = CInt(txtSetTime.Text)
Delay timetac

'===== stop collect =====

LC5.Active = True
Out DataAddress, 224
Out DataAddress, 240
Delay 1
cmdStartCollect.Active = False
LC5.Active = False
Timer1.Enabled = False
PCTime = 0

'===== PC Read In =====

MSComm1.Settings = BPS & PAR & DB & SB
MSComm1.CommPort = COMP
txtStatus.Text = ""
txtStatus.Text = txtStatus.Text & "Com" & COMP & "," & BPS & "," & DB & "," & _
    PAR & "," & SB & vbCrLf
txtStatus.Text = txtStatus.Text & "Open Port" & vbCrLf
If Not MSComm1.PortOpen Then
    MSComm1.PortOpen = True
End If
j = 0
If CHStart Then
    CHStart = False
End If
ScaleH = 500
YValue = 1
Ln_Pic1.Cls

```

```

For I = 0 To (GainVal - 1)
    Channel(I) = 0
Next I
For I = 0 To CHVal
    ABChannel(I) = ""
Next
'===== MCA read out =====
LC8.Active = True
Delay 1
Out DataAddress, 112
Out DataAddress, 240
LC8.Active = False
Do While PCBusy
    Delay 1
Loop
'===== save profile =====
SaveGraph = "c:\lineprofile\data\" & txtProfileName & loopbatch & ".txt"
If Not (SaveGraph = "") Then
    Open SaveGraph For Output As #1 ' Open file for output.
    For I = 0 To (GainVal - 1)
        Print #1, Channel(I) 'write datat to file
    Next I
    Close #1
End If
txtStatus.Text = txtStatus.Text & "Profile Name:" & SaveGraph & vbCrLf
txtStatus.SelStart = Len(txtStatus.Text)
'===== clear MCA Memory =====
LC7.Active = True
Delay 1
Out DataAddress, 176
Out DataAddress, 240
LC7.Active = False

```

```

===== rotate right step motor =====
stepone = 0
Do
    LC4.Active = False
    LC1.Active = True
    Out DataAddress, 241
    Delay (Delaytime)

    LC1.Active = False
    LC2.Active = True
    Out DataAddress, 242
    Delay (Delaytime)

    LC2.Active = False
    LC3.Active = True
    Out DataAddress, 244
    Delay (Delaytime)

    LC3.Active = False
    LC4.Active = True
    Out DataAddress, 248
    Delay (Delaytime)

    stepone = stepone + 1
Loop While (stepone < 2)
    Out DataAddress, 240
    LC4.Active = False
=====

    MSComm1.PortOpen = False

    loopbatch = loopbatch + 1
Loop
    Timer1.Enabled = False
    txtStatus.Text = txtStatus.Text & "Finish Batch Process" & vbCrLf

```

```
txtStatus.SelStart = Len(txtStatus.Text)
```

```
End Sub
```

```
Private Sub cmdClear_Click()
```

```
Ln_Pic1.Cls
```

```
PCTime = 0
```

```
sgtX1.Value = 0
```

```
End Sub
```

```
Private Sub cmdClearCollect_Click()
```

```
LC7.Active = True
```

```
'Delay 1
```

```
Out DataAddress, 176
```

```
Out DataAddress, 240
```

```
LC7.Active = False
```

```
PCTime = 0
```

```
sgtX1.Value = 0
```

```
End Sub
```

```
Private Sub cmdReadout_Click()
```

```
LC8.Active = True
```

```
Delay 1
```

```
Out DataAddress, 112
```

```
Out DataAddress, 240
```

```
LC8.Active = False
```

```
End Sub
```

```
Private Sub cmdROL_Click()
```

```
ROL = True
```

```
Do While ROL
```

```
LC3.Active = False
```

```
LC1.Active = True
```

```
LC2.Active = True
Out DataAddress, 243
Delay (Delaytime)
LC2.Active = False
LC4.Active = True
LC1.Active = True
Out DataAddress, 249
Delay (Delaytime)
LC1.Active = False
LC3.Active = True
LC4.Active = True
Out DataAddress, 252
Delay (Delaytime)
LC4.Active = False
LC2.Active = True
LC3.Active = True
Out DataAddress, 246
Delay (Delaytime)
```

Loop

End Sub

Private Sub cmdROR_Click()

ROR = True

Do While ROR

 LC4.Active = False

 LC1.Active = True

 LC2.Active = True

 Out DataAddress, 243

 Delay (Delaytime)

 LC1.Active = False

 LC2.Active = True

 LC3.Active = True

```

Out DataAddress, 246
Delay (Delaytime)
LC2.Active = False
LC3.Active = True
LC4.Active = True
Out DataAddress, 252
Delay (Delaytime)
LC3.Active = False
LC4.Active = True
LC1.Active = True
Out DataAddress, 249
Delay (Delaytime)
Loop
End Sub

```

```

Private Sub cmdStartCollect_OnChange()

```

```

    If cmdStartCollect.Active Then

```

```

        PCTime = 0

```

```

        Timer1.Enabled = True

```

```

        Out DataAddress, 176

```

```

        Out DataAddress, 240

```

```

        LC6.Active = True

```

```

        'Delay 1

```

```

        Out DataAddress, 208

```

```

        Out DataAddress, 240

```

```

        LC6.Active = False

```

```

    End If

```

```

End Sub

```

```

Private Sub cmdStepClear_Click()

```

```

    Out DataAddress, 240

```

```

    LC1.Active = False

```

```
LC2.Active = False
LC3.Active = False
LC4.Active = False
LC5.Active = False
LC6.Active = False
LC7.Active = False
LC8.Active = False
```

```
End Sub
```

```
Private Sub cmdStepStop_Click()
```

```
    ROR = False
```

```
    ROL = False
```

```
End Sub
```

```
Private Sub cmdStop_Click()
```

```
    BatchStop = True
```

```
End Sub
```

```
Private Sub cmdStopCollect_Click()
```

```
    Timer1.Enabled = False
```

```
    LC5.Active = True
```

```
    'Delay 1
```

```
    Out DataAddress, 224
```

```
    Out DataAddress, 240
```

```
    cmdStartCollect.Active = False
```

```
    LC5.Active = False
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    '==== Comm Port Initial Setting =====
```

```
    BPS = "9600"
```



```

PAR = "N"
DB = "8"
SB = "2"
COMP = "1"
MSComm1.Settings = BPS & PAR & DB & SB
MSComm1.CommPort = COMP
MSComm1.InputLen = 1
MSComm1.RThreshold = 1
'===== Constant Value Setting =====
j = 0
ScaleH = 500
YValue = 1
GainVal = 1024
CHVal = 1153
'===== Clear Buffer =====
If CHStart Then
    CHStart = False
End If
For I = 0 To (GainVal - 1)
    Channel(I) = 0
Next I
'===== stepping motor setting =====
DataAddress = &H378
StatAddress = &H379
CtrlAddress = &H37A
Out CtrlAddress, &H0
Delaytime = 0.1
End Sub

Private Sub iSwitchLedX1_OnChange()
On Error GoTo Err_iSwitchLedX1_OnChange
    If iSwitchLedX1.Active Then

```

```

MSComm1.Settings = BPS & PAR & DB & SB
MSComm1.CommPort = COMP
txtStatus.Text = ""
txtStatus.Text = txtStatus.Text & "Com" & COMP & "," & BPS & "," & DB & "," & _
    PAR & "," & SB & vbCrLf
txtStatus.Text = txtStatus.Text & "Open Port" & vbCrLf
If Not MSComm1.PortOpen Then
    MSComm1.PortOpen = True
End If
j = 0
If CHStart Then
    CHStart = False
End If
ScaleH = 500
YValue = 1
Ln_Pic1.Cls
For I = 0 To (GainVal - 1)
    Channel(I) = 0
Next I
For I = 0 To CHVal
    ABChannel(I) = ""
Next
Else
    If MSComm1.PortOpen Then
        MSComm1.PortOpen = False
    End If
    txtStatus.Text = txtStatus.Text & "Close Port" & vbCrLf
End If
Exit_iSwitchLedX1_OnChange:
Exit Sub
Err_iSwitchLedX1_OnChange:

```

```

txtStatus.Text = txtStatus.Text & Err.Description & " - Error number: " & Err.Number &
vbCrLf

```

```

txtStatus.SelStart = Len(txtStatus.Text)

```

```

Resume Exit_iSwitchLedX1_OnChange

```

```

End Sub

```

```

Private Sub Ln_Pic1_Click()

```

```

End Sub

```

```

-----
Private Sub LSComPort_Click()

```

```

CommPort.Show

```

```

RemoteMCA.Hide

```

```

End Sub

```

```

Private Sub LSExit_Click()

```

```

End

```

```

End Sub

```

```

Private Sub LSLineProfile_Click()

```

```

RemoteMCA.Hide

```

```

LineProfile.Show

```

```

End Sub

```

```

Private Sub LSOpen_Click()

```

```

On Error GoTo Err_LSOpen_Click

```

```

Ln_Pic1.Cls

```

```

txtStatus.Text = "Open File"

```

```

For I = 1 To 128

```

```

Collectstr(I) = ""

```

```

Next I

```

```

For I = 1 To CHVal - 1

```

```

LineChannel(I) = ""
Next I
txtStatus.Text = ""
j = 0
'=====

CommonDialog1.ShowOpen
GraphFile = CommonDialog1.filename
Open GraphFile For Input As #1 ' Open file for input.
Do While Not EOF(1) ' Check for end of file.

    Line Input #1, InputData ' Read line of data.
    StrData = InputData
    Chdat = Mid(StrData, 1, 1)
    If Chdat = Chr(27) Then
        j = j + 1
        Linelen = Len(StrData)
        For I = 1 To Linelen
            Chdata = Mid(StrData, I, 1)
            If Asc(Chdata) <> 27 Then
                If (Asc(Chdata) <> 32) Then
                    Select Case Asc(Chdata)
                        Case 48 To 57
                            Collectstr(j) = Collectstr(j) & Chdata
                        Case Else
                            Collectstr(j) = Collectstr(j) & ","
                    End Select
                End If
            End If
        Next I
        Collectstr(j) = Collectstr(j) + ","
    End If

Loop
'=====

```

```

k = 1
For I = 1 To 128
    m = 0
    j = 1
    Linelen = Len(Collectstr(I))
    Do While (m < 9)
        Charline = Mid(Collectstr(I), j, 1)
        If (Charline <> ",") Then
            LineChannel(k) = LineChannel(k) + Charline
        Else
            m = m + 1
            k = k + 1
        End If
        j = j + 1
    Loop
Next I
'=====

j = 0
For I = 1 To (CHVal - 1)
    If (I Mod 9) <> 1 Then
        Channel(j) = CInt(LineChannel(I))
        j = j + 1
    End If
Next I
'=====

'For I = 0 To 1023
'    Text1.Text = Text1.Text & "Channel[" & I & "], Count =" & Channel(I) & vbCrLf
'    Text1.SelStart = Len(Text1.Text)
'Next I
'=====

DrawPic
Close #1 ' Close file.

```

```
Exit_LSOpen_Click:
```

```
Exit Sub
```

```
Err_LSOpen_Click:
```

```
txtStatus.Text = txtStatus.Text & Err.Description & " - Error number: " & Err.Number &
vbCrLf
```

```
txtStatus.SelStart = Len(txtStatus.Text)
```

```
Resume Exit_LSOpen_Click
```

```
End Sub
```

```
Private Sub LSSave_Click()
```

```
On Error GoTo Err_LSSave_Click
```

```
Dim SaveGraph
```

```
CommonDialog1.ShowSave
```

```
SaveGraph = Trim(CommonDialog1.filename)
```

```
'SaveGraph = "c:\lineprofile\data\" & txtProfileName.Text & ".txt"
```

```
If Not (SaveGraph = "") Then
```

```
Open SaveGraph For Output As #1 ' Open file for output.
```

```
For I = 0 To (GainVal - 1)
```

```
Print #1, Channel(I) 'write datat to file
```

```
Next I
```

```
Close #1
```

```
End If
```

```
Exit_LSSave_Click:
```

```
Exit Sub
```

```
Err_LSSave_Click:
```

```
txtStatus.Text = txtStatus.Text & Err.Description & " - Error number: " & Err.Number &
vbCrLf
```

```
txtStatus.SelStart = Len(txtStatus.Text)
```

```
Resume Exit_LSSave_Click
```

```
End Sub
```

```
Private Sub LSSteppingMotor_Click()
```

```

RemoteMCA.Hide
StepMotor.Show
End Sub

```

```

Private Sub MSComm1_OnComm()
Dim CommData As String
If Led1.Active Then
    Led1.Active = False
Else
    Led1.Active = True
End If
If MSComm1.PortOpen Then
    CommData = CStr(MSComm1.Input)
    If CommData <> (Chr(32)) And CommData <> (Chr(0)) And CommData <> (Chr(4)) Then
        If CommData <> (Chr(15)) Then
            If CommData <> (Chr(27)) Then
                If CommData <> (Chr(10)) Then
                    If CommData <> (Chr(13)) And (CHStart) Then
                        ABChannel(j) = ABChannel(j) + CommData
                        If j >= 1152 Then
                            iSwitchLedX1.Active = False
                            DataOut
                            DrawPic
                        End If
                    End If
                Else
                    CHStart = False
                End If
            Else
                CHStart = True
                j = j + 1
            End If
        End If
    End If
End Sub

```

```

Else
    j = j + 1
End If
End If
End If
End Sub

```

```

Private Sub Timer1_Timer()
    PCTime = PCTime + 1
    sgtX1.Value = PCTime
End Sub

```

```

:Stepping motor transition

```

```

Dim Value As Integer
Dim DataAddress, StatAddress, CtrlAddress As Integer
Dim I As Integer
Dim ROR, ROL As Boolean
Dim Delaytime As Integer
Dim T As Integer
Private Sub Command1_Click()
    ROL = True
    Do While ROL

```

```

        C4.Active = False
        C1.Active = True
        C2.Active = True
        Out DataAddress, 243
        Delay (Delaytime)
        C1.Active = False
        C2.Active = True
        C3.Active = True
        Out DataAddress, 246
        Delay (Delaytime)

```



```

C2.Active = False
C3.Active = True
C4.Active = True
Out DataAddress, 252
Delay (Delaytime)
C3.Active = False
C4.Active = True
C1.Active = True
Out DataAddress, 249
Delay (Delaytime)
Loop
End Sub

```

```
Private Sub Command2_Click()
```

```

    Out DataAddress, 240
    C1.Active = False
    C2.Active = False
    C3.Active = False
    C4.Active = False
    C5.Active = False
    C6.Active = False
    C7.Active = False
    C8.Active = False

```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
ROR = True
```

```
Do While ROR
```

```

    C3.Active = False
    C1.Active = True
    C2.Active = True

```

```
Out DataAddress, 243
Delay (Delaytime)
C2.Active = False
C4.Active = True
C1.Active = True
Out DataAddress, 249
Delay (Delaytime)
C1.Active = False
C3.Active = True
C4.Active = True
Out DataAddress, 252
Delay (Delaytime)
C4.Active = False
C2.Active = True
C3.Active = True
Out DataAddress, 246
Delay (Delaytime)
Loop
End Sub
-----
Private Sub Command4_Click()
    ROR = False
    ROL = False
End Sub

Private Sub Command5_Click()
    RemoteMCA.Show
    StepMotor.Hide
End Sub

Private Sub Form_Load()
```

```
DataAddress = &H378  
StatAddress = &H379  
CtrlAddress = &H37A  
Out CtrlAddress, &H0  
Delaytime = 1  
End Sub
```

```
Private Sub HScroll1_Change()  
    Text1.Text = (HScroll1.Value)  
    Delaytime = (HScroll1.Value)  
End Sub
```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

รายละเอียดหัวต่อสาย remote control J112

ตารางที่ ค.1 การจัดขารับส่งสัญญาณที่พอร์ต J112 ของเครื่องวิเคราะห์แบบหลายช่อง

| PIN NUMBER | SIGNAL | NORMAL STATUS | DESCRIPTION |
|------------|----------|---------------|----------------------------|
| 1,2,3 | GND | | GROUND |
| 4 | EXT ADV | LOW | MCS EXTERNAL ADVANCE IN |
| 5 | EXT TRIG | LOW | MCS ADVANCE TRIGGER IN |
| 6 | SCADV | LOW | SAMPLE CHANGER ADVANCE OUT |
| 7 | BSYOUT | HIGH | MCA BUSY OUT |
| 8 | BSYIN | HIGH | DEVICE BUSY IN |
| 9 | COLLECT | HIGH | COLLECT STATUS OUT |
| 10 | I/O | HIGH | READOUT STATUS OUT |
| 11 | STOPCOL | HIGH | STOP COLLECT IN |
| 12 | STARTCOL | HIGH | START COLLECT IN |
| 13 | CLRDAT | HIGH | CLEAR DATA IN |
| 15 | GATE | LOW | GATE IN |
| 21 | AMS | LOW | ABORT MCS SWEEP |
| 22 | AOF | LOW | ADDRESS OVERFLOW |
| 23,24 | GND | | GROUND |
| 25 | STARTIO | HIGH | START READOUT IN |

ALL INPUTS

HIGH +2.4 to 5.0 V or open

LOW 0.0 to +3.0 V at < 2 mA

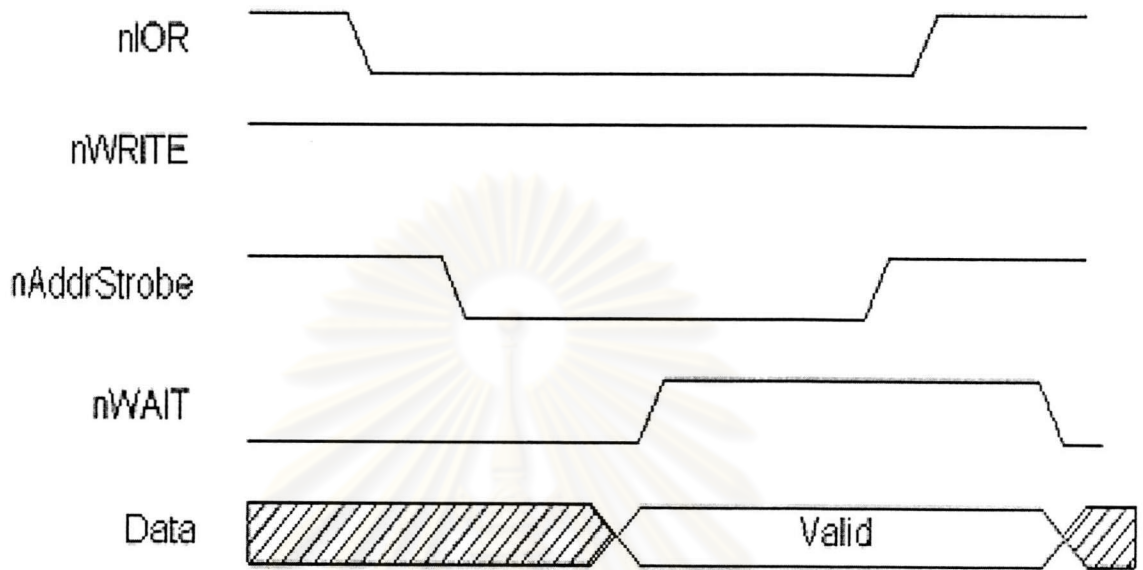
pluse widths > 0.5 μ s , except as noted

ALL OUTPUTS

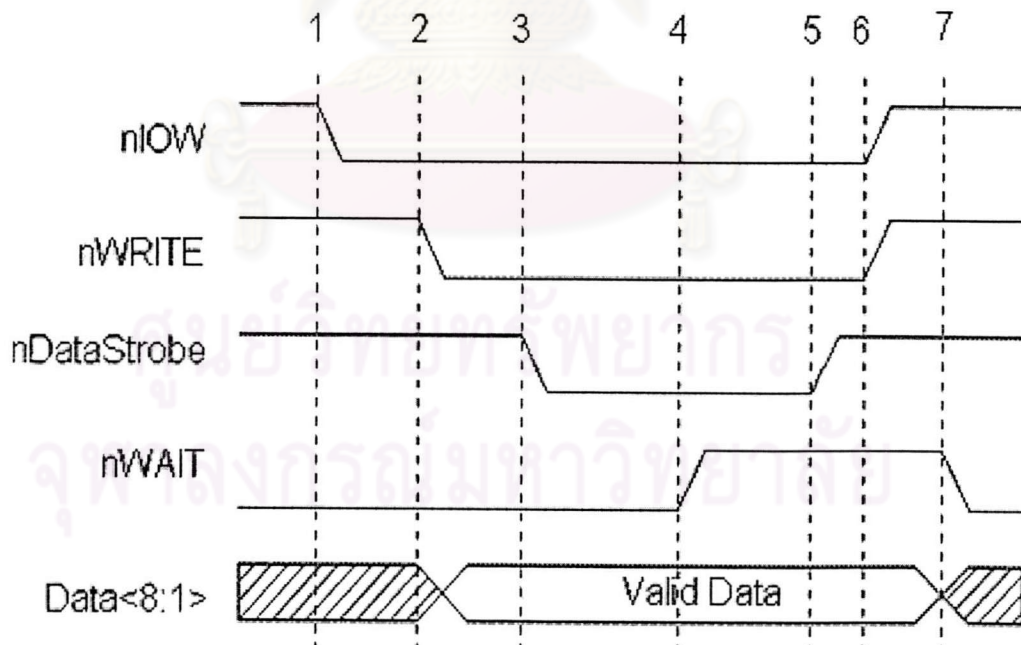
HIGH +2.4 to 5.0 V at < 0.4 mA (2.4 V)

LOW 0.0 to 0.5 V at < 0.4 mA

ภาคผนวก ง
รายละเอียดช่วงสัญญาณของพอร์ตเครื่องพิมพ์



EPP Address_Read Cycle



EPP Address_Write Cycle

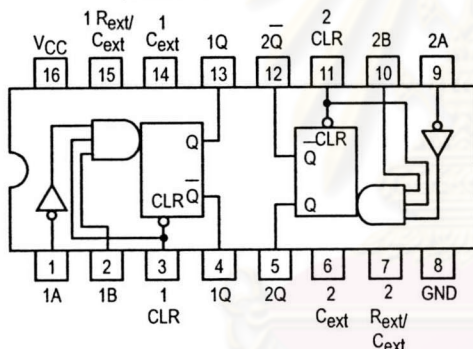


RETRIGGERABLE MONOSTABLE MULTIVIBRATORS

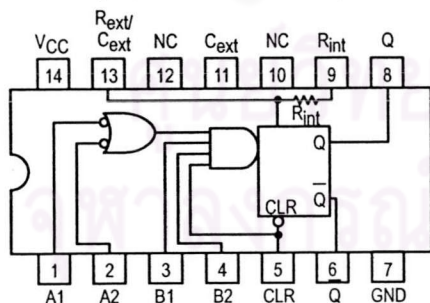
These dc triggered multivibrators feature pulse width control by three methods. The basic pulse width is programmed by selection of external resistance and capacitance values. The LS122 has an internal timing resistor that allows the circuits to be used with only an external capacitor. Once triggered, the basic pulse width may be extended by retriggering the gated low-level-active (A) or high-level-active (B) inputs, or be reduced by use of the overriding clear.

- Overriding Clear Terminates Output Pulse
- Compensated for V_{CC} and Temperature Variations
- DC Triggered from Active-High or Active-Low Gated Logic Inputs
- Retriggerable for Very Long Output Pulses, up to 100% Duty Cycle
- Internal Timing Resistors on LS122

SN54/74LS123 (TOP VIEW)
(SEE NOTES 1 THRU 4)



SN54/74LS122 (TOP VIEW)
(SEE NOTES 1 THRU 4)



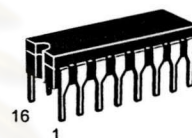
NC — NO INTERNAL CONNECTION.

NOTES:

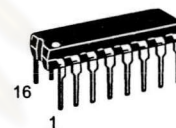
1. An external timing capacitor may be connected between C_{ext} and R_{ext}/C_{ext} (positive).
2. To use the internal timing resistor of the LS122, connect R_{int} to V_{CC} .
3. For improved pulse width accuracy connect an external resistor between R_{ext}/C_{ext} and V_{CC} with R_{int} open-circuited.
4. To obtain variable pulse widths, connect an external variable resistance between R_{int}/C_{ext} and V_{CC} .

SN54/74LS122 SN54/74LS123

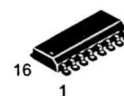
RETRIGGERABLE MONOSTABLE MULTIVIBRATORS LOW POWER SCHOTTKY



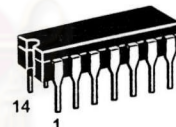
**J SUFFIX
CERAMIC
CASE 620-08**



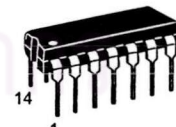
**N SUFFIX
PLASTIC
CASE 648-08**



**D SUFFIX
SOIC
CASE 751B-03**



**J SUFFIX
CERAMIC
CASE 632-08**



**N SUFFIX
PLASTIC
CASE 646-06**



**D SUFFIX
SOIC
CASE 751A-02**

ORDERING INFORMATION

| | |
|------------|---------|
| SN54LSXXXJ | Ceramic |
| SN74LSXXXN | Plastic |
| SN74LSXXXD | SOIC |

SN54/74LS122 • SN54/74LS123

LS122
FUNCTIONAL TABLE

| INPUTS | | | | | OUTPUTS | |
|--------|----|----|----|----|---------|---|
| CLEAR | A1 | A2 | B1 | B2 | Q | Q |
| L | X | X | X | X | L | H |
| X | H | H | X | X | L | H |
| X | X | X | L | X | L | H |
| X | X | X | X | L | L | H |
| H | L | X | ↑ | H | ⌋ | ⌋ |
| H | L | X | H | ↑ | ⌋ | ⌋ |
| H | X | L | ↑ | H | ⌋ | ⌋ |
| H | X | L | H | ↑ | ⌋ | ⌋ |
| H | H | ↓ | H | H | ⌋ | ⌋ |
| H | ↓ | ↓ | H | H | ⌋ | ⌋ |
| H | ↓ | H | H | H | ⌋ | ⌋ |
| ↑ | L | X | H | H | ⌋ | ⌋ |
| ↑ | X | L | H | H | ⌋ | ⌋ |

LS123
FUNCTIONAL TABLE

| INPUTS | | | OUTPUTS | |
|--------|---|---|---------|---|
| CLEAR | A | B | Q | Q |
| L | X | X | L | H |
| X | H | X | L | H |
| X | X | L | L | H |
| H | L | ↑ | ⌋ | ⌋ |
| H | ↓ | H | ⌋ | ⌋ |
| ↑ | L | H | ⌋ | ⌋ |

TYPICAL APPLICATION DATA

The output pulse t_W is a function of the external components, C_{ext} and R_{ext} or C_{ext} and R_{int} on the LS122. For values of $C_{ext} \geq 1000$ pF, the output pulse at $V_{CC} = 5.0$ V and $V_{RC} = 5.0$ V (see Figures 1, 2, and 3) is given by

$$t_W = K R_{ext} C_{ext} \text{ where } K \text{ is nominally } 0.45$$

If C_{ext} is in pF and R_{ext} is in k Ω then t_W is in nanoseconds.

The C_{ext} terminal of the LS122 and LS123 is an internal connection to ground, however for the best system performance C_{ext} should be hard-wired to ground.

Care should be taken to keep R_{ext} and C_{ext} as close to the monostable as possible with a minimum amount of inductance between the R_{ext}/C_{ext} junction and the R_{ext}/C_{ext} pin. Good groundplane and adequate bypassing should be designed into the system for optimum performance to insure that no false triggering occurs.

It should be noted that the C_{ext} pin is internally connected to ground on the LS122 and LS123, but not on the LS221. Therefore, if C_{ext} is hard-wired externally to ground, substitution of a LS221 onto a LS123 socket will cause the LS221 to become non-functional.

The switching diode is not needed for electrolytic capacitance application and should not be used on the LS122 and LS123.

To find the value of K for $C_{ext} \geq 1000$ pF, refer to Figure 4. Variations on V_{CC} or V_{RC} can cause the value of K to change, as can the temperature of the LS123, LS122. Figures 5 and 6 show the behavior of the circuit shown in Figures 1 and 2 if

separate power supplies are used for V_{CC} and V_{RC} . If V_{CC} is tied to V_{RC} , Figure 7 shows how K will vary with V_{CC} and temperature. Remember, the changes in R_{ext} and C_{ext} with temperature are not calculated and included in the graph.

As long as $C_{ext} \geq 1000$ pF and $5K \leq R_{ext} \leq 260K$ (SN74LS122/123) or $5K \leq R_{ext} \leq 160K$ (SN54LS122/123), the change in K with respect to R_{ext} is negligible.

If $C_{ext} \leq 1000$ pF the graph shown on Figure 8 can be used to determine the output pulse width. Figure 9 shows how K will change for $C_{ext} \leq 1000$ pF if V_{CC} and V_{RC} are connected to the same power supply. The pulse width t_W in nanoseconds is approximated by

$$t_W = 6 + 0.05 C_{ext} (\text{pF}) + 0.45 R_{ext} (\text{k}\Omega) C_{ext} + 11.6 R_{ext}$$

In order to trim the output pulse width, it is necessary to include a variable resistor between V_{CC} and the R_{ext}/C_{ext} pin or between V_{CC} and the R_{ext} pin of the LS122. Figure 10, 11, and 12 show how this can be done. R_{ext} remote should be kept as close to the monostable as possible.

Retriggering of the part, as shown in Figure 3, must not occur before C_{ext} is discharged or the retrigger pulse will not have any effect. The discharge time of C_{ext} in nanoseconds is guaranteed to be less than $0.22 C_{ext}$ (pF) and is typically $0.05 C_{ext}$ (pF).

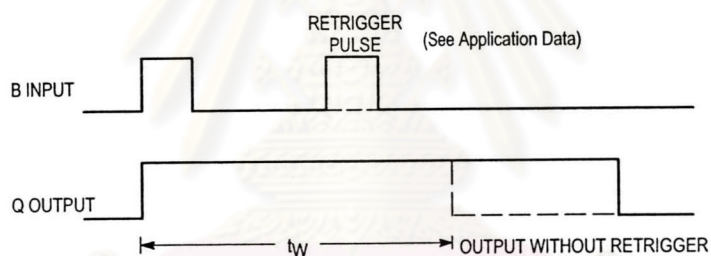
For the smallest possible deviation in output pulse widths from various devices, it is suggested that C_{ext} be kept ≥ 1000 pF.

SN54/74LS122 • SN54/74LS123

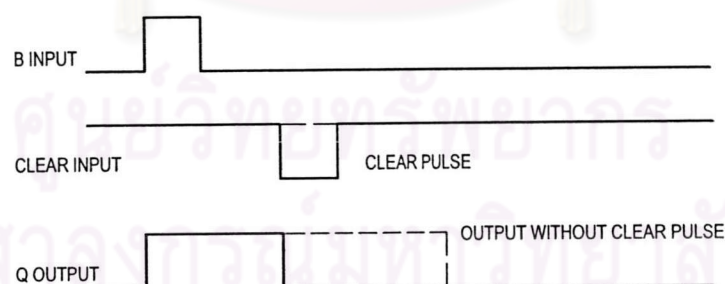
GUARANTEED OPERATING RANGES

| Symbol | Parameter | | Min | Typ | Max | Unit |
|------------------------------------|-------------------------------------------------------------------|----------|----------------|------------|-------------|------|
| V _{CC} | Supply Voltage | 54 74 | 4.5 4.75 | 5.0 5.0 | 5.5 5.25 | V |
| T _A | Operating Ambient Temperature Range | 54 74 | -55 0 | 25 25 | 125 70 | °C |
| I _{OH} | Output Current — High | 54, 74 | | | -0.4 | mA |
| I _{OL} | Output Current — Low | 54 74 | | | 4.0 8.0 | mA |
| R _{ext} | External Timing Resistance | 54 74 | 5.0 5.0 | | 180 260 | kΩ |
| C _{ext} | External Capacitance | 54, 74 | No Restriction | | | |
| R _{ext} /C _{ext} | Wiring Capacitance at R _{ext} /C _{ext} Terminal | 54, 74 | | | 50 | pF |

WAVEFORMS



EXTENDING PULSE WIDTH



OVERRIDING THE OUTPUT PULSE

SN54/74LS122 • SN54/74LS123

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

| Symbol | Parameter | | Limits | | | Unit | Test Conditions | |
|-----------------|--------------------------------|--------|--------|-------|------|------|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| | | | Min | Typ | Max | | | |
| V _{IH} | Input HIGH Voltage | | 2.0 | | | V | Guaranteed Input HIGH Voltage for All Inputs | |
| V _{IL} | Input LOW Voltage | 54 | | | 0.7 | V | Guaranteed Input LOW Voltage for All Inputs | |
| | | 74 | | | 0.8 | | | |
| V _{IK} | Input Clamp Diode Voltage | | | -0.65 | -1.5 | V | V _{CC} = MIN, I _{IN} = -18 mA | |
| V _{OH} | Output HIGH Voltage | 54 | 2.5 | 3.5 | | V | V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table | |
| | | 74 | 2.7 | 3.5 | | V | | |
| V _{OL} | Output LOW Voltage | 54, 74 | | 0.25 | 0.4 | V | I _{OL} = 4.0 mA | V _{CC} = V _{CC} MIN, V _{IN} = V _{IL} or V _{IH} per Truth Table |
| | | 74 | | 0.35 | 0.5 | V | I _{OL} = 8.0 mA | |
| I _{IH} | Input HIGH Current | | | | 20 | μA | V _{CC} = MAX, V _{IN} = 2.7 V | |
| | | | | | 0.1 | mA | V _{CC} = MAX, V _{IN} = 7.0 V | |
| I _{IL} | Input LOW Current | | | | -0.4 | mA | V _{CC} = MAX, V _{IN} = 0.4 V | |
| I _{OS} | Short Circuit Current (Note 1) | | -20 | | -100 | mA | V _{CC} = MAX | |
| I _{CC} | Power Supply Current | LS122 | | | 11 | mA | V _{CC} = MAX | |
| | | LS123 | | | 20 | | | |

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C, V_{CC} = 5.0 V)

| Symbol | Parameter | | Limits | | | Unit | Test Conditions | |
|--------------------|-------------------------------|--|--------|-----|-----|------|------------------------------------------------------------------------------------------------------------|--|
| | | | Min | Typ | Max | | | |
| t _{PLH} | Propagation Delay, A to Q | | | 23 | 33 | ns | C _{ext} = 0 C _L = 15 pF R _{ext} = 5.0 kΩ R _L = 2.0 kΩ | |
| t _{PHL} | Propagation Delay, A to Q | | | 32 | 45 | | | |
| t _{PLH} | Propagation Delay, B to Q | | | 23 | 44 | ns | | |
| t _{PHL} | Propagation Delay, B to Q | | | 34 | 56 | | | |
| t _{PLH} | Propagation Delay, Clear to Q | | | 28 | 45 | ns | | |
| t _{PHL} | Propagation Delay, Clear to Q | | | 20 | 27 | | | |
| t _{W min} | A or B to Q | | | 116 | 200 | ns | C _{ext} = 1000 pF, R _{ext} = 10 kΩ, C _L = 15 pF, R _L = 2.0 kΩ | |
| t _{WQ} | A to B to Q | | 4.0 | 4.5 | 5.0 | μs | | |

AC SETUP REQUIREMENTS (T_A = 25°C, V_{CC} = 5.0 V)

| Symbol | Parameter | | Limits | | | Unit | Test Conditions | |
|----------------|-------------|--|--------|-----|-----|------|-----------------|--|
| | | | Min | Typ | Max | | | |
| t _W | Pulse Width | | 40 | | | ns | | |

SN54/74LS122 • SN54/74LS123

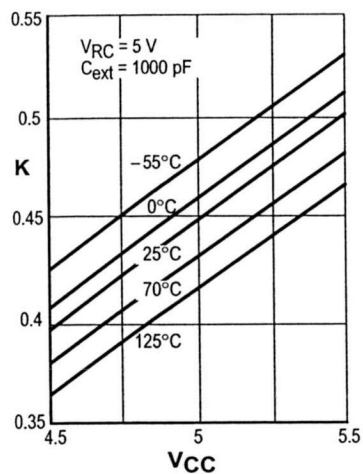


Figure 5. K versus VCC

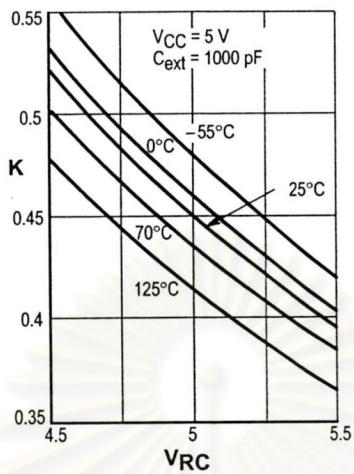


Figure 6. K versus VRC

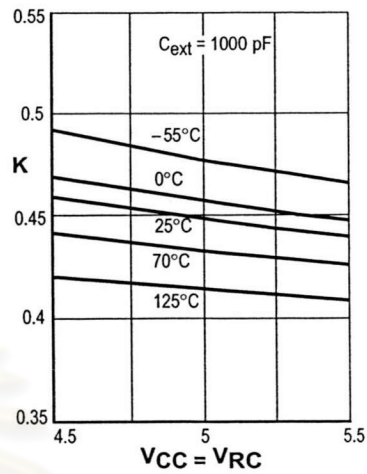


Figure 7. K versus VCC and VRC

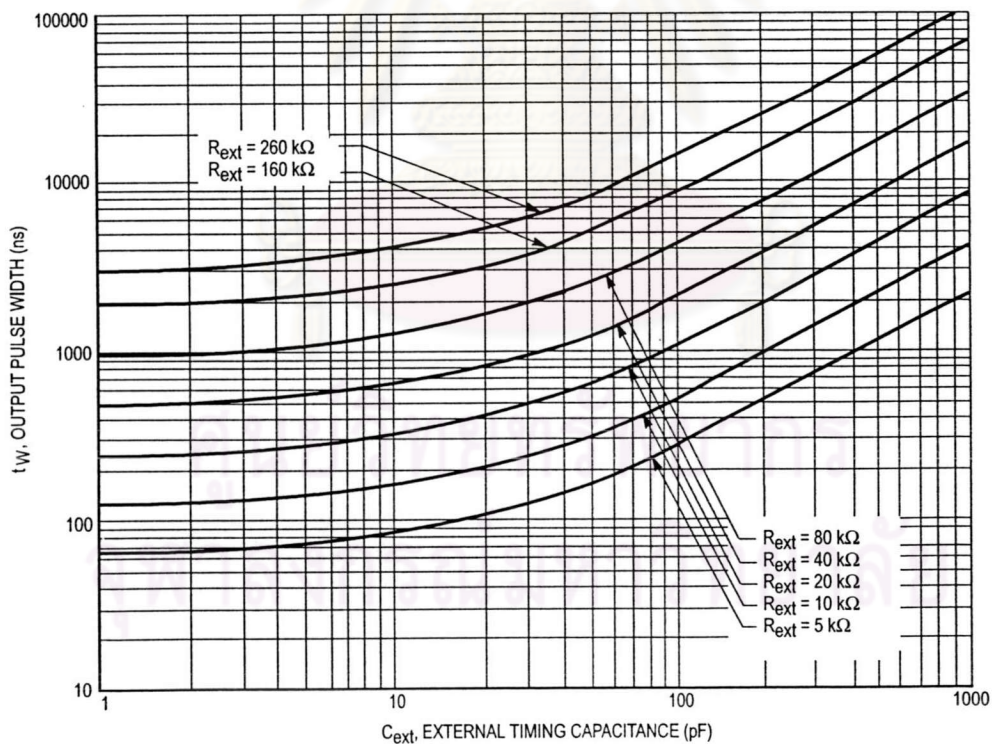


Figure 8

SN54/74LS122 • SN54/74LS123

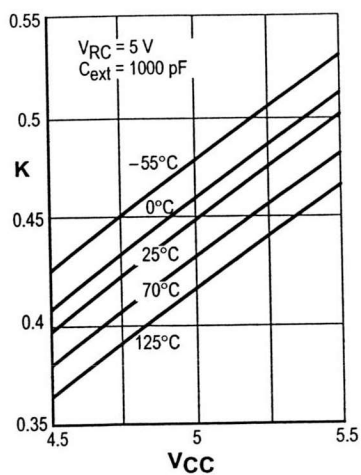


Figure 5. K versus V_{CC}

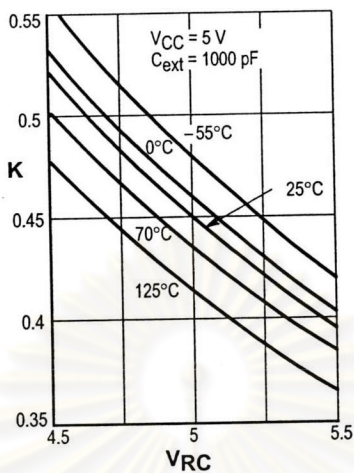


Figure 6. K versus V_{RC}

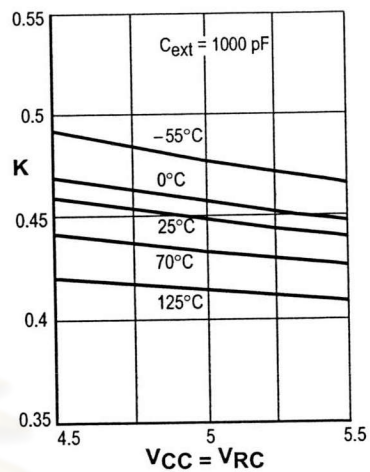


Figure 7. K versus V_{CC} and V_{RC}

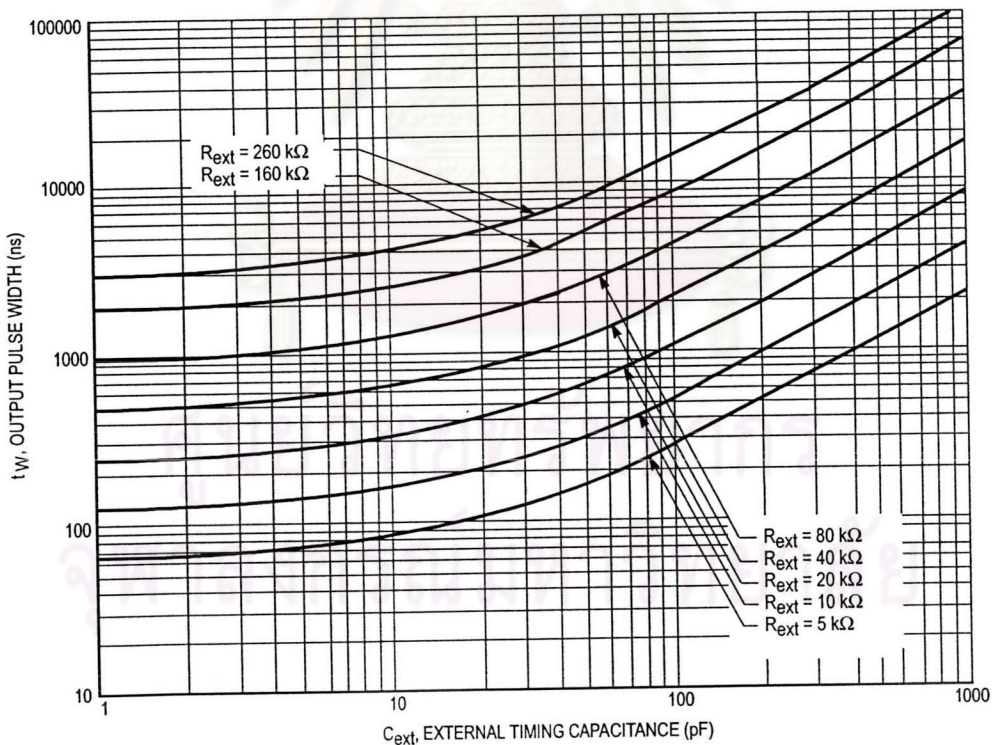


Figure 8

ประวัติผู้เขียนวิทยานิพนธ์

นายอร โณพ บุญพัฒนาภรณ์ เกิดเมื่อวันที่ 10 สิงหาคม 2518 ที่กรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาบัณฑิตจากภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ เมื่อปีการศึกษา 2540 และในปีการศึกษา 2541 ได้เข้าศึกษาระดับปริญญาโทที่ภาควิชาวิศวกรรมเทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย