

## รายการอ้างอิง

### ภาษาไทย

ทวีชัย ภูมิทิพย์. ไขปัญหา RS 232. กรุงเทพมหานคร : บริษัท ซี เอ็ด ยูเคชั่น จำกัด (มหาชน), 2538.

พิภพ ลลิตาภรณ์. ระบบการวางแผนและควบคุมการผลิต. กรุงเทพมหานคร : บริษัท ส.เอเชียเพรส จำกัด, 2541.

ราบินเดอร์ ศรีกิจจาภรณ์. คู่มือการใช้งาน Visual Basic สำหรับวินโดวส์. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2538.

### ภาษาอังกฤษ

Dennis K., Joe G., Bill H. and Jason T.R. Visual Basic 5 Databas3e How To. USA : The Waite Group, 1997.

Kenneth J. Ayala. The 8051 Microcontroller Architecture Programming and Applications. USA: West Publishing Company, 1991.

Microsoft Press. Microsoft Visual Basic 6 Programmer's Guide. USA : Microsoft., 1998.

Paul Bates. Practical Digital and Data Communication with LSI Applications. USA : Prentice – Hall International Inc., 1987.

Scott, N.R. Analog and Digital Computer Technology. USA : McGraw-Hill, 1980.

Spencer B. Smith . Computer-Based Production and Inventory Control. USA : Prentice- Hall International Inc., 1989.

ภาคผนวก  
รายละเอียดของโปรแกรม

## รายละเอียดของโปรแกรม

```
Attribute VB_Name = "adjust_delay_time"  
Attribute VB_GlobalNameSpace = False  
Attribute VB_Creatable = False  
Attribute VB_PredeclaredId = True  
Attribute VB_Exposed = False  
Private Sub Command1_Click() 'exit  
    'Data1.Recordset.edit  
    'Data1.Recordset.Fields(0).Value = Val(Text1.Text)  
    'Data1.Recordset.Update  
    Unload adjust_delay_time  
End Sub  
  
Private Sub Form_Load()  
    Left = (Screen.Width - Width) / 2 'make form to center  
    Top = (Screen.Height - Height) / 2 - 200  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    delaytime = Data1.Recordset.Fields(0).Value  
    product.Enabled = True      'enable main menu  
    adjust_delay_time.Enabled = False  
    adjust_delay_time.WindowState = 1 'minimize  
End Sub  
  
Attribute VB_Name = "define_product"  
Attribute VB_GlobalNameSpace = False  
Attribute VB_Creatable = False  
Attribute VB_PredeclaredId = True  
Attribute VB_Exposed = False
```

```
Option Explicit
Dim MyDB As Database
Dim MyQuery As QueryDef, MyParameter As Parameter
Dim MyRecordset As Recordset
Dim index As Integer
Dim con_create As Boolean
Dim mo_flag As Boolean
Dim first As Boolean
Dim readercount As Integer
```

```
Private Sub lock_text()
    Text1.Locked = True    'product name
    DBCombo1.Locked = True 'selective process name
End Sub
```

```
Private Sub unlock_text()
    Text1.Locked = False
    DBCombo1.Locked = False
End Sub
```

```
Private Sub red_text()
    Text1.BackColor = &HC0C0FF
    DBCombo1.BackColor = &HC0C0FF
End Sub
```

```
Private Sub yellow_text()
    Text1.BackColor = &HC0FFFF
    DBCombo1.BackColor = &HC0FFFF
End Sub
```

```
Private Sub green_text()  
    Text1.BackColor = &HC0FFC0 'light green  
    DBCombo1.BackColor = &HC0FFC0  
End Sub  
  
Private Sub Command1_Click() 'create  
    Data1.Enabled = False  
    Command1.Enabled = False 'create  
    Command2.Enabled = False 'delete  
    Command3.Enabled = False 'modify  
    Command4.Enabled = False 'exit  
    Command5.Enabled = True 'cancel  
    Command6.Enabled = True 'ok.  
  
    If con_create = False Then 'after update con_create=false,but during create=true  
        If Data1.Recordset.RecordCount > 0 Then  
            Data1.Recordset.MoveLast  
        End If  
        Data1.Recordset.AddNew  
    End If  
    unlock_text  
    Text1.SetFocus  
    green_text  
End Sub  
  
Private Sub Command2_Click() 'delete  
    Dim msg$  
    Dim response  
  
    If Data1.Recordset.RecordCount = 0 Then  
        MsgBox "no record", 0 + 64  
        Exit Sub  
    Else  
        msg$ = "Do you really want to delete?" ' Define message.
```

```
response = MsgBox(msg$, vbYesNo + vbQuestion + vbDefaultButton1) ' help, ctxt)
If response = vbNo Then ' User chose Yes.
    Exit Sub 'exit main program
End If
Data1.Recordset.Delete
Data1.Recordset.MoveNext
If Data1.Recordset.EOF Then
    If Data1.Recordset.RecordCount = 0 Then
        Exit Sub
    Else
        Data1.Recordset.MoveLast 'Previous
    End If
End If
End If
End Sub
```

```
Private Sub Command3_Click() 'modify
If Data1.Recordset.RecordCount = 0 Then
    MsgBox "no record", 0 + 64
    Exit Sub
End If
Data1.Enabled = False
Command1.Enabled = False
Command2.Enabled = False
Command3.Enabled = False
Command4.Enabled = False
Command5.Enabled = True
Command6.Enabled = True
mo_flag = True
con_create = False
unlock_text
```

```
yellow_text
Text1.SetFocus
Data1.Recordset.edit 'is in edit state
End Sub

Private Sub Command4_Click() 'exit
    Unload define_product
End Sub

Private Sub Command5_Click() 'cancel
    If con_create = True Then
        Text1.SetFocus 'cancel for create
        Data1.UpdateControls
        If Data1.Recordset.RecordCount > 0 Then
            Data1.Recordset.MoveLast 'goto prev. record (last rec.)
        End If
        Data1.Enabled = True
        Command1.Enabled = True
        Command2.Enabled = True
        Command3.Enabled = True
        Command4.Enabled = True
        Command5.Enabled = False
        Command6.Enabled = False
        con_create = False
        lock_text
        red_text
    Else
        Text1.SetFocus 'cancel for modify
        Data1.UpdateControls
        Data1.Enabled = True
        Command1.Enabled = True
```

```

Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
lock_text
red_text
Command5.Enabled = False
Command6.Enabled = False
mo_flag = False
End If
End Sub

Private Sub Command6_Click() 'ok.
Dim MyDB As Database
Dim MyRecordset As Recordset
Dim i As Integer
Dim MyCheck As Boolean
Dim zone_error As Boolean
If con_create Or mo_flag = True Then 'first create not check textbox
Text1.Text = RTnm(Text1.Text)
If Len(Text1.Text) = 0 Then 'check for empty text
MsgBox "you must fill in product name", 0 + 48
Text1.SetFocus
Exit Sub 'only ok. and cancel command active
Else
Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyRecordset = MyDB.OpenRecordset("product", dbOpenDynaset)
If MyRecordset.RecordCount > 0 Then
MyRecordset.MoveFirst
End If
If mo_flag = True Then
Do While Not MyRecordset.EOF 'for modify

```



```

    If Data1.Recordset.AbsolutePosition <>
MyRecordset.AbsolutePosition Then
    If Text1.Text = MyRecordset.Fields(0) Then 'timer tag field
        MsgBox "duplicate product name", 0 + 48
        Text1.SetFocus
        MyRecordset.Close
        Exit Sub
    End If
End If
MyRecordset.MoveNext
Loop
Else      'for create
Do While Not MyRecordset.EOF
    If Text1.Text = MyRecordset.Fields(0) Then
        MsgBox "duplicate product name", 0 + 48
        Text1.SetFocus
        MyRecordset.Close
        Exit Sub
    End If
    MyRecordset.MoveNext
Loop
End If
MyRecordset.Close

Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyRecordset = MyDB.OpenRecordset("process", dbOpenDynaset)
' For index = 0 To Data2.Recordset.RecordCount - 1 'readrcount - 1
    'MyRecordset.MoveFirst
    If MyRecordset.RecordCount > 0 Then
        MyRecordset.MoveFirst
    Else

```

```
        MsgBox "no process defined", 0 + 48
        MyRecordset.Close
    Exit Sub
End If
Do While Not MyRecordset.EOF '
    If DBCombo1.Text <> MyRecordset.Fields(0) Then
        MyRecordset.MoveNext
    Else
        zone_error = False
        Exit Do
    End If
    zone_error = True
Loop
If zone_error = True Then
    MsgBox "process is not in the list". 0 + 48
    DBCombo1.SetFocus
    MyRecordset.Close
    Exit Sub
End If
' Next index
MyRecordset.Close

End If
Data1.Recordset.Update 'make con_create=false
Data1.Enabled = True
Command1.Enabled = True
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = False
Command6.Enabled = False
```

```
lock_text
red_text
Text1.SetFocus
If mo_flag = False Then
    Data1.Recordset.MoveLast
Else
    mo_flag = False
End If
Else
End If
End Sub

Private Sub Data1_Reposition() 'update Data1.Caption
    Data1.Caption = "product " & Data1.Recordset.AbsolutePosition + _
    1 & " of " & Data1.Recordset.RecordCount
End Sub

Private Sub Data1_Validate(Action As Integer, Save As Integer)
    Select Case Action
        Case 5 'addnew action
            con_create = True
        Case Else
            con_create = False
    End Select
End Sub

Private Sub DBCombo1_change()
Dim MyDB As Database
Dim MyQuery As QueryDef, MyParameter As Parameter
Dim MyRecordset As Recordset
Dim i As Integer
```

```

Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyQuery = MyDB.QueryDefs("process_operator")
'PARAMETERS process_want Text;
'SELECT process.*, *
'From process
'WHERE (((process.process)=[process_want]));
MyQuery.Parameters("process_want").Value = DBCombo1.Text ' Set parameters.
Set MyRecordset = MyQuery.OpenRecordset(, dbOpenDynaset) ' Open Recordset.
List2.Clear
If MyRecordset.RecordCount > 0 Then      'field 0 is process name
    For i = 1 To (MyRecordset.Fields.Count) - 1 'begin at field 1
        If MyRecordset.Fields(i) <> "no" Then
            List2.AddItem MyRecordset.Fields(i) 'add to station list
        End If
    Next i
End If
MyRecordset.Close
End Sub

Private Sub Form_Activate()
    If Data1.Recordset.RecordCount > 0 Then
        Data1.Recordset.MoveLast 'update data1 caption
        Data1.Recordset.MoveFirst
    End If
    lock_text
    red_text
    mo_flag = False
    con_create = False
    Command5.Enabled = False 'cancel
    Command6.Enabled = False 'ok.
    Text1.SetFocus

```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Left = (Screen.Width - Width) / 2
```

```
    Top = (Screen.Height - Height) / 2 - 200
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    product.Enabled = True
```

```
    define_product.Enabled = False
```

```
    define_product.WindowState = 1
```

```
End Sub
```

```
Attribute VB_Name = "define_station"
```

```
Attribute VB_GlobalNameSpace = False
```

```
Attribute VB_Creatable = False
```

```
Attribute VB_PredeclaredId = True
```

```
Attribute VB_Exposed = False
```

```
Dim myDB As Database
```

```
Dim MyRecordset As Recordset
```

```
Dim con_create As Boolean
```

```
Dim mo_flag As Boolean
```

```
Dim a As Integer
```

```
Private Sub lock_text()
```

```
    Text4.Locked = True 'station name
```

```
    Text1.Locked = True 'unit number
```

```
    Text2.Locked = True 'operator name
```

```
    Text3.Locked = True 'location
```

```
End Sub
```

```
Private Sub unlock_text()
```

```
    Text4.Locked = False
```

```
    Text1.Locked = False
```

```
    Text2.Locked = False
```

```
    Text3.Locked = False
```

```
End Sub
```

```
Private Sub red_text()
```

```
    Text4.BackColor = &HC0C0FF 'light red
```

```
    Text1.BackColor = &HC0C0FF
```

```
    Text2.BackColor = &HC0C0FF
```

```
    Text3.BackColor = &HC0C0FF
```

```
End Sub
```

```
Private Sub yellow_text()
```

```
    Text4.BackColor = &HC0FFFF 'light yellow
```

```
    Text1.BackColor = &HC0FFFF
```

```
    Text2.BackColor = &HC0FFFF
```

```
    Text3.BackColor = &HC0FFFF
```

```
End Sub
```

```
Private Sub green_text()
```

```
    Text4.BackColor = &HC0FFC0 'light green
```

```
    Text1.BackColor = &HC0FFC0
```

```
    Text2.BackColor = &HC0FFC0
```

```
    Text3.BackColor = &HC0FFC0
```

```
End Sub
```

```
Private Sub Command1_Click() 'exit
```

```
    Unload define_station
```

```

End Sub

Private Sub Command2_Click() 'create
    Data1.Enabled = False
    Command1.Enabled = False 'create
    Command2.Enabled = False 'delete
    Command3.Enabled = False 'modify
    Command4.Enabled = False 'exit
    Command5.Enabled = True 'cancel
    Command6.Enabled = True 'ok.
    If con_create = False Then 'after update con_create=false
        If Data1.Recordset.RecordCount > 0 Then
            Data1.Recordset.MoveLast
        End If
        Data1.Recordset.AddNew
    End If
    unlock_text
    Text4.SetFocus 'station name
    green_text
End Sub

Private Sub Command3_Click() 'delete
    Dim msg$
    Dim response
    If Data1.Recordset.RecordCount = 0 Then
        MsgBox "no record", 0 + 64
        Exit Sub
    Else
        msg$ = "Do you really want to delete?" ' Define message.
        response = MsgBox(msg$, vbYesNo + vbQuestion + vbDefaultButton1) ' help, ctxt)
        If response = vbNo Then ' User chose no
            Exit Sub 'exit main program
        End If
    End If
End Sub

```

```
End If
Data1.Recordset.Delete
Data1.Recordset.MoveNext
If Data1.Recordset.EOF Then
    If Data1.Recordset.RecordCount = 0 Then
        Exit Sub
    Else
        Data1.Recordset.MoveLast
    End If
End If
End If
End Sub

Private Sub Command4_Click() 'modify
    If Data1.Recordset.RecordCount = 0 Then
        MsgBox "no record", 0 + 64
        Exit Sub
    End If
    Data1.Enabled = False
    Command1.Enabled = False
    Command2.Enabled = False
    Command3.Enabled = False
    Command4.Enabled = False
    Command5.Enabled = True
    Command6.Enabled = True
    mo_flag = True
    unlock_text
    yellow_text
    Text4.SetFocus
    Data1.Recordset.edit 'in edit mode con_create=false
End Sub
```



```
Private Sub Command5_Click() 'cancel
    If con_create = True Then
        Text4.SetFocus      'cancel for create
        Data1.UpdateControls 'restore old value
        If Data1.Recordset.RecordCount > 0 Then
            Data1.Recordset.MoveLast
        End If
        Data1.Enabled = True
        Command1.Enabled = True
        Command2.Enabled = True
        Command3.Enabled = True
        Command4.Enabled = True
        Command5.Enabled = False
        Command6.Enabled = False
        con_create = False
        lock_text
        red_text
    Else
        Text4.SetFocus      'cancel for modify
        Data1.UpdateControls 'restore old value
        Data1.Enabled = True
        Command1.Enabled = True
        Command2.Enabled = True
        Command3.Enabled = True
        Command4.Enabled = True
        lock_text
        red_text
        Command5.Enabled = False
        Command6.Enabled = False
        mo_flag = False
    End If
End Sub
```

```

End If
End Sub

Private Sub Command6_Click() 'ok.
Dim MyDB As Database
Dim MyRecordset As Recordset
Dim i As Integer
Dim time1 As Date
Dim MyCheck As Boolean
If con_create Or mo_flag = True Then 'first create not check station name
    Text4.Text = RTrim(Text4.Text) 'space adjustment
    If Len(Text4.Text) = 0 Then
        MsgBox "you must fill in station name", 0 + 48
        Text4.SetFocus
        Exit Sub
    Else 'check for dup. of station name
        Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
        Set MyRecordset = MyDB.OpenRecordset("station name", dbOpenDynaset)
        If MyRecordset.RecordCount > 0 Then
            MyRecordset.MoveLast
            MyRecordset.MoveFirst
        End If
        If mo_flag = True Then 'in modify state
            Do While Not MyRecordset.EOF 'not check current record
                If Data1.Recordset.AbsolutePosition <> MyRecordset.AbsolutePosition Then
                    If Text4.Text = MyRecordset.Fields(0) Then 'station name field
                        MsgBox "duplicate station name", 0 + 48
                        Text4.SetFocus
                        MyRecordset.Close
                    End If
                End If
            Loop
        End If
        Exit Sub
    End If
End If

```

```

End If
MyRecordset.MoveNext
Loop
Else          'ok. for create
Do While Not MyRecordset.EOF
  If Text4.Text = MyRecordset.Fields(0) Then
    MsgBox "duplicate station name", 0 + 48
    Text4.SetFocus
    MyRecordset.Close
    Exit Sub
  End If
  MyRecordset.MoveNext
Loop
End If
MyRecordset.Close

Text1.Text = RTrim(Text1.Text) 'check unit number
If Text1.Text = "" Then
  MsgBox "you must fill in unit number", 0 + 48
  Text1.SetFocus
  Exit Sub
Else
  MyCheck = IsNumeric(Text1.Text)
  If (MyCheck = False) Or (Len(Text1.Text) < 3) Then 'check for unit num. len.
    MsgBox "unit number must be 3 digit number", 0 + 48
    Text1.SetFocus
    Exit Sub
  End If
End If

Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyRecordset = MyDB.OpenRecordset("station name", dbOpenDynaset)

```

```

If MyRecordset.RecordCount > 0 Then
    MyRecordset.MoveLast
    MyRecordset.MoveFirst
End If

If mo_flag = True Then      'for modify state
Do While Not MyRecordset.EOF 'check for dup. unit number
If Data1.Recordset.AbsolutePosition <> MyRecordset.AbsolutePosition Then
    If Text1.Text = MyRecordset.Fields(1) Then 'unit number field
        MsgBox "duplicate unit number", 0 + 48
        Text1.SetFocus
        MyRecordset.Close
        Exit Sub
    End If
End If

MyRecordset.MoveNext

Loop

Else      'for create state
Do While Not MyRecordset.EOF '
If Text1.Text = MyRecordset.Fields(1) Then
    MsgBox "duplicate unit number", 0 + 48
    Text1.SetFocus
    MyRecordset.Close
    Exit Sub
End If

MyRecordset.MoveNext

Loop

End If

MyRecordset.Close

If Text2.Text = "" Then      'check for operator name
    MsgBox "you must fill in operator name", 0 + 48
    Text2.SetFocus

```

```
Exit Sub
End If
If Text3.Text = "" Then      'check for location name
    MsgBox "you must fill in location name", 0 + 48
    Text3.SetFocus
    Exit Sub
End If
End If
Data1.Recordset.Update
Data1.Enabled = True
Command1.Enabled = True
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = False
Command6.Enabled = False
lock_text
red_text
Text4.SetFocus
if mo_flag = False Then    'if in create state then move to last record
    Data1.Recordset.MoveLast 'move to the create record
Else
    mo_flag = False      'if in modify state
End If
End If
End Sub

Private Sub Data1_Reposition() 'update data1 caption
    Data1.Caption = "Station " & Data1.Recordset.AbsolutePosition + _
    1 & " of " & Data1.Recordset.RecordCount
End Sub
```

```
Private Sub Data1_Validate(Action As Integer, Save As Integer)
```

```
    Select Case Action
```

```
        Case 5          'addnew action
```

```
            con_create = True
```

```
        Case Else      'other action
```

```
            con_create = False
```

```
    End Select
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
    If Data1.Recordset.RecordCount > 0 Then
```

```
        Data1.Recordset.MoveLast 'update data1 caption
```

```
        Data1.Recordset.MoveFirst
```

```
    End If
```

```
    lock_text
```

```
    red_text
```

```
    mo_flag = False
```

```
    Command5.Enabled = False
```

```
    Command6.Enabled = False
```

```
    Text4.SetFocus
```

```
    con_create = False
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    ' mo_flag = False
```

```
    Left = (Screen.Width - Width) / 2
```

```
    Top = (Screen.Height - Height) / 2 - 200
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
product.Enabled = True
define_station.Enabled = False
define_station.WindowState = 1
End Sub
```

```
Private Sub Text4_KeyPress(KeyAscii As Integer) 'station name
If con_create Or mo_flag = True Then
    char = Chr(KeyAscii)
    If char = Chr$(13) Then
        Text1.SetFocus      'set focus at unit number
    End If
End If
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer) 'unit number
Dim MyCheck As Boolean
If con_create Or mo_flag = True Then
    char = Chr(KeyAscii)
    If char = Chr$(13) Then
        Text2.SetFocus      'set focus at unit number
    End If
End If
End Sub
```

```
Private Sub Text2_KeyPress(KeyAscii As Integer)
If con_create Or mo_flag = True Then
```

```
char = Chr(KeyAscii)
If char = Chr$(13) Then
    Text3.SetFocus
End If
End If
End Sub
Attribute VB_Name = "mon_Module"
Global delaytime As Long 'Integer

Sub delay() 'depen on computer speed
'Dim i As Integer
'Dim MyDB As Database
'Dim MyRecordset As Recordset
'Dim x As Integer
'Dim delaytime As Integer

'Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
'Set MyRecordset = MyDB.OpenRecordset("station name", dbOpenDynaset)
'MyRecordset.MoveFirst
'x = 20000 'Val(MyRecordset.Fields(0))
For i = 1 To delaytime 'x 'MyRecordset.Fields(0).Value 'pentium 100mhz use 25000,30000
Next i '486 33mhz use 2000
End Sub
```



```
Attribute VB_Name = "monitor"  
Attribute VB_Creatable = False  
Attribute VB_Exposed = False  
Private Sub Command1_Click()  
    monitor.WindowState = 1 'minimize  
    monitor.Enabled = False  
    run.Enabled = True  
    run.Show  
    run.WindowState = 2 'maximize  
End Sub  
  
Private Sub Command2_Click()  
    monitor.WindowState = 1 'minimize  
    monitor.Enabled = False  
    Edit.Enabled = True  
    Edit.Show  
    Edit.WindowState = 2 'maximize  
End Sub  
  
Private Sub Command3_Click()  
    monitor.WindowState = 1 'minimize  
    monitor.Enabled = False  
    report.Enabled = True  
    report.Show  
    report.WindowState = 2 'maximize  
End Sub  
  
Private Sub Command4_Click() 'exit  
    product.Enabled = True  
    product.WindowState = 2 'maximize  
    monitor.WindowState = 1 'minimize
```

```
monitor.Enabled = False
run.Picture1.Visible = False
run.Picture2.Visible = True
End Sub
```

```
Attribute VB_Name = "edit"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Dim fromchild$, dummy
Dim timeout_flag As Boolean
Dim stopflag As Boolean
Dim dummy1
Dim con_create As Boolean 'after addnew command con_create=true
```

```
Private Sub lock_text()
    Text1.Locked = True 'work order
    Text2.Locked = True 'pcs.
    Text3.Locked = True 'lot size
    Text4.Locked = True 'last lot size
    Text5.Locked = True 'lot quantity
    Text8.Locked = True 'lot remain
    Text9.Locked = True 'product
End Sub
```

```
Private Sub unlock_text()
    Text1.Locked = False
    Text2.Locked = False
    Text3.Locked = False
    Text4.Locked = False
```

```
Text5.Locked = False
Text8.Locked = False
Text9.Locked = False
End Sub
```

```
Private Sub red_text() 'display red text
Text1.BackColor = &HC0C0FF
Text2.BackColor = &HC0C0FF
Text3.BackColor = &HC0C0FF
Text4.BackColor = &HC0C0FF
Text5.BackColor = &HC0C0FF
Text8.BackColor = &HC0C0FF
Text9.BackColor = &HC0C0FF
End Sub
```

```
Private Sub yellow_text() 'display yellow text
Text1.BackColor = &HC0FFFF
Text2.BackColor = &HC0FFFF
Text3.BackColor = &HC0FFFF
Text4.BackColor = &HC0FFFF
Text5.BackColor = &HC0FFFF
Text8.BackColor = &HC0FFFF
Text9.BackColor = &HC0FFFF
End Sub
```

```
Private Sub green_text()
Text1.BackColor = &HC0FFC0 'display green text
Text2.BackColor = &HC0FFC0
Text3.BackColor = &HC0FFC0
' Text4.BackColor = &HC0FFC0 'light green
' Text5.BackColor = &HC0FFC0
' Text8.BackColor = &HC0FFC0
```

End Sub

```
'Private Sub delay() 'depend on computer speed
'Dim i As Integer
' For i = 1 To 25000 'pentium 100mhz use 25000,30000
' Next i      '486 33mhz use 2000
'End Sub
```

```
Private Sub init_comport() 'init. comport use port 2
If (MSComm1.PortOpen = False) Then
    MSComm1.PortOpen = True
' MSComm1.CommPort = 2
    MSComm1.Settings = "2400,N,8,1"
    On Error Resume Next 'if error exec. the next line
    If MSComm1.PortOpen = False Then
        MSComm1.PortOpen = True
        If Err Then
            MsgBox "COM1: not available. Change the CommPort property to another port."
            Exit Sub
        End If
    End If
End If
End Sub
```

```
'Private Sub trigger() 'send turn on reader command
' Comm1.DTREnable = False
' MSComm1.Output = "Z" 'start reading coil #1,(Y or Z)
' delay      'period of pc. occupation time
' MSComm1.DTREnable = True
'End Sub
```

```

Private Sub Read_buffer() 'receive char. from serial port
Dim a$, fromchild$, dummy 'by pooling method
Timer1.Enabled = True 'on timeout timer
fromchild$ = "" 'receive buffer
' Wait for receive char. to come back from the reader
' MSComm1.InBufferCount=0
Do
dummy = DoEvents() 'pooling for os.
' If there is data in the buffer, then read it.
If MSComm1.InBufferCount Then
fromchild$ = fromchild$ + MSComm1.Input
' Check for "return" char.
If InStr(fromchild$, Chr$(13)) Then 'if ok then
Timer1.Enabled = False 'off timer
timeout_flag = False
a$ = Time$
Text6.Text = Left(fromchild$, 10) 'text6 contain id. code
Exit Do 'find id then exit do
Else
End If
Else 'if not has incoming char.
If timeout_flag = True Then 'check time out
Timer1.Enabled = False 'if time cut then turn off timer
timeout_flag = False 'prepare for the next time
MsgBox "reader not response!,please check the connection wire and try again!", 0
Exit Do 'time out then exit do
Else
' status.Text = "searching" 'if not time out then show searching
End If
End If

```

```

    Loop
End Sub

Private Sub process_query() 'show operation list according to process
Dim MyDB As Database
Dim MyQuery As QueryDef, MyParameter As Parameter
Dim MyRecordset As Recordset

Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyQuery = MyDB.QueryDefs("productquerydef")
'PARAMETERS product_want Text;
'SELECT *
'From station
'WHERE ((station.product=[product_want]));
MyQuery.Parameters("product_want").Value = Text7.Text 'text7=process name
Set MyRecordset = MyQuery.OpenRecordset(, dbOpenDynaset) 'Open Recordset.
List2.Clear
Do While Not MyRecordset.EOF
    List2.AddItem MyRecordset.Fields(1) 'display station list
    MyRecordset.MoveNext
Loop
MyRecordset.Close
MyQuery.Close
End Sub

Private Sub query_delwork() 'delete w/o of granddb use para. from workorder
Dim MyDB As Database 'in text1.text
Dim MyQuery As QueryDef, MyParameter As Parameter
Dim MyRecordset As Recordset

Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyQuery = MyDB.QueryDefs("delworkorder") 'Open gate1 QueryDef.
'PARAMETERS work_want Text;

```

```
'DELETE granddb.workorder, *
'From granddb
'WHERE (((granddb.workorder)=[work_want]));
MyQuery.Parameters("work_want").Value = Text1.Text 'parameters.
MyQuery.Execute
MyQuery.Close
Data3.DatabaseName = ("c:\program files\microsoft visual basic\db5.MDB")
Data3.RecordSource = "granddb"
Data3.Refresh      'update data of granddb in dbgrid
End Sub
```

```
Private Sub Command5_Click()      'cancel creat new w/o command
'If con_create = True Then      'after addnew command
    Text1.SetFocus              'w/o
    Data1.UpdateControls        'update old value ignore new value
    If Data1.Recordset.RecordCount > 0 Then
        Data1.Recordset.MoveLast '
    End If
    Data1.Enabled = True
    Command1.Enabled = True      'create
    Command2.Enabled = True      'del.
    ' Command3.Enabled = True      'modify
    Command4.Enabled = True      'exit
    Command5.Enabled = False      'cancel
    Command6.Enabled = False      'ok. (cal. para.)
    lock_text
    red_text
'Else                            'cancel for modify
'    Text1.SetFocus
'    Data1.UpdateControls
'    Data1.Enabled = True
```

```

' Command1.Enabled = True
' Command2.Enabled = True
'' Command3.Enabled = True
' Command4.Enabled = True
' lock_text
' red_text
' Command5.Enabled = False
' Command6.Enabled = False
'End If
End Sub

```

```

Private Sub Command1_Click() 'create w/o
    Data1.Enabled = False
    Command1.Enabled = False 'create
    Command2.Enabled = False 'delete
' Command3.Enabled = False 'modify
    Command4.Enabled = False 'exit
    Command5.Enabled = True 'cancel
    Command6.Enabled = True 'ok.
    If con_create = False Then 'first create
        If Data1.Recordset.RecordCount > 0 Then
            Data1.Recordset.MoveLast
        End If
        Data1.Recordset.AddNew 'after this command con_create=true
    End If
    unlock_text
    Text1.SetFocus 'w/o
    green_text
'Data1.Recordset.AddNew
'Text1.SetFocus 'Text = "*****"
'Command1.Enabled = True 'cal. para

```



```
End Sub
```

```
Private Sub Command3_Click() 'on reader
    init_comport
    MSComm1.DTREnable = False
    MSComm1.Output = "Z" "T" "Z" 'start reader coil(Y, Z)
    delay
    MSComm1.DTREnable = True
    Picture1.Visible = True
    Picture2.Visible = False
End Sub
```

```
Private Sub Command40_Click() 'off reader
    init_comport
    MSComm1.DTREnable = False
    MSComm1.Output = "X" 'stop reader coil #1
    delay
    MSComm1.DTREnable = True
    Picture2.Visible = True
    Picture1.Visible = False
    'stopflag = True
End Sub
```

```
Private Sub Command2_Click() 'delete w/o
    Dim MyTableDef As TableDef, MyField As Field
    Dim MyDatabase As Database
    Dim i As Integer
    Dim msg$
    Dim response
    msg$ = "Do you really want to delete ?" 'Define message.
    response = MsgBox(msg$, vbYesNo + vbCritical + vbDefaultButton1) 'help, ctxt)
```

```
If response = vbNo Then          'User chose no
    Exit Sub                    'do not del.
Else                            'User chose del.
    query_delwork              'del. w/o from granddb
    If Data1.Recordset.RecordCount = 0 Then
        Exit Sub
    Else                        'RecordCount > 0
        Data1.Recordset.Delete    'del workorder in data1
        Data1.Recordset.MoveNext
    If Data1.Recordset.EOF Then
        If Data1.Recordset.RecordCount = 0 Then
            Exit Sub
        Else
            Data1.Recordset.MoveLast
        End If
    End If
End If
End If
End If
End Sub

Private Sub Command7_Click()    'load id code from reader
    Dim msg$, response
    Dim MyDatabase As Database
    Dim MyRecordset As Recordset
    Dim MyTableDef As TableDef, MyField As Field
    Dim str As String
    Dim i As Integer
    Dim ii As Integer
    Dim iii As Integer
    Dim firstfill_flag As Boolean
    If Val(Text8.Text) > 0 Then    'lot remain still > 0
```

```

If List2.ListCount = 0 Then      'lot remain still > 0
    MsgBox "you must select a product name in the list", 0 + 64
    Exit Sub
End If

' If DBCombo1.Text = "" Then      'check for select product
'   MsgBox "you must select a product name", 0 + 64
'   Exit Sub
' End If

If MSComm1.PortOpen = True Then  'ckeck for open port
    MSComm1.DTREnable = False
    MSComm1.Output = "V"        'dump id code command
    delay
    MSComm1.DTREnable = True
    Text6.Text = "0000000000"    'dimmy value
    Read_buffer                  'receive id code from reader
    If Text6.Text <> "0000000000" And Text6.Text <> "" Then
        For iii = 1 To 10
            ' MyCheck = "F" Like "[A-Z]" ' Returns True.
            ' MyCheck = "F" Like "[!A-Z]" ' Returns False.
            ' isNumeric(MyVar)
            '
            '   If (IsNumeric(Mid(Text6.Text, iii, 1)) = False) And ((Mid(Text6.Text, iii, 1) Like "[!A-F]") = True)
Then '
            '   '((Mid(Text6.Text, iii, 1) Like "[!A-F]")
            '       TEXT9.Text = iii
            '       MsgBox "data buffer of readaer is not valid,try load card again", 0 + 64
            '       Exit Sub
            '   End If
        Next iii
    End If

```

```

Set MyDatabase = Workspaces(0).OpenDatabase("c:\program files\microsoft visual
basic\db5.MDB")

Set MyRecordset = MyDatabase.OpenRecordset("granddb", dbOpenDynaset) ' Open
Recordset.

Do While Not MyRecordset.EOF
  If Text6.Text = MyRecordset.Fields(1) Then 'id code field
    MsgBox "duplicate lot id,try read new card", 0 + 48
    MyRecordset.Close
    Exit Sub
  End If
  MyRecordset.MoveNext
Loop

firstfill_flag = True 'add record to granddb
For ii = 0 To List2.ListCount - 1 'station list
  MyRecordset.AddNew 'create new record
  MyRecordset.Fields(0).Value = Text1.Text 'w/o
  MyRecordset.Fields(1).Value = Text6.Text 'lot id
  If firstfill_flag = True Then 'first load
    If Text8.Text = 1 Then 'has only 1 lot remain
      MyRecordset.Fields(3).Value = Text4.Text 'inbuf=last lot size
      Data1.Enabled = True
      lock_text
      red_text
    Else
      MyRecordset.Fields(3).Value = Text3.Text 'inbuf field=lot size
    End If
  Else
    MyRecordset.Fields(3).Value = 0 'if not first load from reader
  End If
  MyRecordset.Fields(2).Value = List2.List(ii) 'station

```

```

MyRecordset.Fields(4).Value = 0      'input
MyRecordset.Fields(5).Value = 0      'output
MyRecordset.Fields(6).Value = 0      'reject
MyRecordset.Fields(7).Value = 0      'defect1
MyRecordset.Fields(8).Value = 0
MyRecordset.Fields(9).Value = 0
MyRecordset.Fields(10).Value = 0
MyRecordset.Fields(11).Value = 0     'defect5
MyRecordset.Fields(12).Value = DBCombo1.Text 'product
MyRecordset.Fields(13).Value = Text7.Text 'process
MyRecordset.Update
firstfill_flag = False              'indicate for next load from reader
Next ii                             'next station fill
Text8.Text = Val(Text8.Text) - 1    'dec. lot remain
MyDatabase.Close
Else
MsgBox "no card found,try to read a card again", 0 + 64
Exit Sub
End If
Data3.DatabaseName = ("c:\program files\microsoft visual basic\db5.MDB")
Data3.RecordSource = "granddb"
Data3.Refresh
' Data3.Recordset.MoveLast
' run.Data2.DatabaseName = ("\\mor95\vb\db5.mdb") '
' run.Data2.RecordSource = "granddb"
' run.Data2.Refresh
Else
msg$ = "you must tum on reader first,do you wani to tum cn now? " ' Define message.
response = MsgBox(msg$, vbYesNo + vbCritical + vbDefaultButton1) ' help, ctxt)
If response = vbNo Then ' User chose Yes.
Exit Sub                'exit main program

```

```
Else ' User chose No.
    Command3_Click 'on reader
End If
End If
Else
    MsgBox "end of w/o", 0 + 64
End If
End Sub

Private Sub Command4_Click() 'exit
Unload edit
' If Data1.Recordset.RecordCount > 0 Then
'   If Data1.Recordset.EOF Then
'     Data1.Recordset.MoveLast
'   Else
'     Data1.Recordset.MoveFirst
'   End If
' End If
" monitor.Enabled = True
" monitor.WindowState = 2 'maximize
' edit.WindowState = 1 'minimize
' edit.Enabled = False
' If MSComm1.PortOpen = True Then
'   MSComm1.DTREnable = False
'   MSComm1.Output = "X" 'stop reader#1
'   delay
'   MSComm1.DTREnable = True
'   MSComm1.PortOpen = False
' End If
' Picture2.Visible = True 'red light
' Picture1.Visible = False
```

End Sub

```

Private Sub Command6_Click() 'ok. cal. para
Dim MyTableDef As TableDef '
Dim MyDB As Database
Dim MyRecordset As Recordset
Dim i As Integer
Dim MyCheck As Boolean
If DBCombo1.Text = "" Then 'check for select product
    MsgBox "you must select a product name", 0 + 64
    Exit Sub
End If
If Val(Text2.Text) = 0 Then 'quantity
    MsgBox "quantity must more than 0"
    Exit Sub
End If
If Val(Text3.Text) = 0 Then 'lot size
    MsgBox "lot size must more than 0"
    Exit Sub
End If
If Val(Text3.Text) > Val(Text2.Text) Then 'check for valid lot size
    MsgBox "quantity must more than lot size"
    Exit Sub
End If

Text5.Text = Val(Text2.Text) \ Val(Text3.Text) 'lot quantity=pcs.\lotsize
Text9.Text = DBCombo1.Text 'product
Text8.Text = Text5.Text 'lot remain=lot quantity

If Val(Text3.Text) > 0 Then 'check for last lot size
    ' Text5.Text = Val(Text2.Text) \ Val(Text3.Text)

```

```

a = Val(Text2.Text) Mod Val(Text3.Text)
If a > 0 Then
    Text4.Text = Val(Text3.Text) + a 'last lot size=lot size + remainder
Else 'a=0
    Text4.Text = Text3          'last lot size=lot size
End If
End If

If con_create = True Then          'is in create status not check textbox
    Text1.Text = RTrim(Text1.Text)
    If Len(Text1.Text) = 0 Then
        MsgBox "you must fill in w/o", 0 + 48
        Text1.SetFocus
        Exit Sub
    Else 'w/o text not empty
        Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
        Set MyRecordset = MyDB.OpenRecordset("workorder", dbOpenDynaset)
        If MyRecordset.RecordCount > 0 Then
            MyRecordset.MoveFirst
        End If
        Do While Not MyRecordset.EOF 'check for dup. w/o
            If Text1.Text = MyRecordset.Fields(0) Then
                MsgBox "duplicate w/o name", 0 + 48
                Text1.SetFocus
                MyRecordset.Close
                Exit Sub
            End If

            MyRecordset.MoveNext
        Loop
        MyRecordset.Close

```



```

End If
Data1.Recordset.Update 'after this command con_create=false
' Data1.Recordset.MoveLast
Data1.Enabled = True
Command1.Enabled = True
Command2.Enabled = True
' Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = False
Command6.Enabled = False
lock_text
red_text
Text1.SetFocus
' sound
End If
If Data1.Recordset.RecordCount <> 0 Then
Data1.Recordset.MoveFirst
Data1.Recordset.MoveLast 'move to lasi create record
Data1.Caption = "Workorder " & Data1.Recordset.AbsolutePosition + _
1 & " of " & Data1.Recordset.RecordCount
End If
End Sub

Private Sub list_operation() 'display list of station in list2
Dim MyDB As Database
Dim MyQuery As QueryDef, MyParameter As Parameter
Dim MyRecordset As Recordset
Dim X As Integer
Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyQuery = MyDB.QueryDefs("product_operation")
'PARAMETERS product_want Text;

```

```

'SELECT process.*
'FROM product INNER JOIN process ON product.process = process.process
'WHERE (((product.product)=[product_want]));
MyQuery.Parameters("product_want").Value = DBCombo1.Text 'Set parameters.
Set MyRecordset = MyQuery.OpenRecordset(, dbOpenDynaset) 'Open Recordset.
List2.Clear
If MyRecordset.RecordCount > 0 Then
  For i = 1 To (MyRecordset.Fields.Count) - 1
    If MyRecordset.Fields(i) <> "no" Then
      List2.AddItem MyRecordset.Fields(i) '& "' & MyRecordset.Fields(2)
    End If
  Next i
  Text7.Text = MyRecordset.Fields(0).Value 'process
End If

' Text7.Text = MyRecordset.Fields(0).Value 'process
MyRecordset.Close
End Sub

Private Sub Data1_Reposition()
  Data1.Caption = "Workorder " & Data1.Recordset.AbsolutePosition + _
  1 & " of " & Data1.Recordset.RecordCount
  If Val(Text8.Text) = 0 Then 'lot remain = 0 , display yellow text
    red_text
  Else 'lot remain <> 0 , display yellow text
    yellow_text
  End If
End Sub

Private Sub Data1_Validate(Action As Integer, Save As Integer)
  Select Case Action

```

```

Case 5          'addnew action
    con_create = True      'after addnew con_create=true
Case Else
    con_create = False    'other action like update con_create=false
End Select
' Text2.Text = con_create 'save
' Text3.Text = Save
End Sub

Private Sub Data3_Reposition() 'update granddb to run menu
    run.Data2.DatabaseName = ("c:\program files\microsoft visual basic\db5.MDB")
    run.Data2.RecordSource = "granddb"
    run.Data2.Refresh
End Sub

Private Sub DBCombo1_change()
    list_operation 'display list station
End Sub

Private Sub Form_Activate()
    If Data1.Recordset.RecordCount > 0 Then
        Data1.Recordset.MoveLast 'update data1 caption
        Data1.Recordset.MoveFirst 'in case of call this form not in first time
    End If
    lock_text
    red_text
    Command5.Enabled = False
    Command6.Enabled = False
    Text1.SetFocus
    Data1.Enabled = True
    If Val(Text8.Text) = 0 Then 'display file status

```

```
    red_text          'lot remain = 0 , display yellow text
Else
    yellow_text      'lot remain <> 0 , display yellow text
End If
product.Enabled = True
End Sub

Private Sub Form_Load()
    Timer1.Enabled = False
    Picture1.Visible = False
    Picture2.Visible = True
    Data3.DatabaseName = ("c:\program files\microsoft visual basic\db5.MDB")
    Data3.RecordSource = "granddb"
    Data3.Refresh
    ccn_create = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
    If Data1.Recordset.RecordCount > 0 Then
        If Data1.Recordset.EOF Then
            Data1.Recordset.MoveLast
        Else
            Data1.Recordset.MoveFirst
        End If
    End If
    ' monitor.Enabled = True
    ' monitor.WindowState = 2 'maximize
    edit.WindowState = 1 'minimize
    edit.Enabled = False
    if MSComm1.PortOpen = True Then
        MSComm1.DTREnable = False
```

```
MSComm1.Output = "X"    'stop reader#1
delay
MSComm1.DTREnable = True
MSComm1.PortOpen = False
End If
Picture2.Visible = True  'red light
Picture1.Visible = False
End Sub

Private Sub Timer1_Timer()
    timeout_flag = True
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer) 'station name
    Dim char As String
    'If con_create Or mo_flag = True Then
        char = Chr(KeyAscii)
        If char = Chr$(13) Then
            Text2.SetFocus    'set focus at unit number
        End If
    'End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer) 'unit number
    Dim char As String
    'If con_create Or mo_flag = True Then
        char = Chr(KeyAscii)
```

```
If char = Chr$(13) Then
    Text3.SetFocus      'set focus at unit number
End If
'End If
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
Dim char As String
' If con_create Or mo_flag = True Then
    char = Chr(KeyAscii)
    If char = Chr$(13) Then
        Text1.SetFocus
    End If
' End If
End Sub

End

Attribute VB_Name = "process"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim MyDB As Database
Dim MyQuery As QueryDef, MyParameter As Parameter
Dim MyRecordset As Recordset
Dim index As Integer
Dim con_create As Boolean
Dim mo_flag As Boolean
Dim first As Boolean
```

```
Private Sub lock_text()  
    Text1.Locked = True      'process name  
    For index = 0 To 15      'max. operation=16  
        DBCombo1(index).Locked = True 'operation  
    Next index  
End Sub
```

```
Private Sub unlock_text()  
    Text1.Locked = False  
    For index = 0 To 15  
        DBCombo1(index).Locked = False  
    Next index  
End Sub
```

```
Private Sub red_text()  
    Text1.BackColor = &HC0C0FF  'red bgcolor  
    For index = 0 To 15  
        DBCombo1(index).BackColor = &HC0C0FF  
    Next index  
End Sub
```

```
Private Sub yellow_text()  
    Text1.BackColor = &HC0FFFF  
    For index = 0 To 15  
        DBCombo1(index).BackColor = &HC0FFFF  
    Next index  
End Sub
```

```
Private Sub green_text()  
    Text1.BackColor = &HC0FFC0  'green  
    For index = 0 To 15
```

```

        DBCombo1(index).BackColor = &HC0FFC0
        DBCombo1(index).Text = "no" 'default value of add new record
    Next index
End Sub

Private Sub Command1_Click() 'create
Dim i As Integer
    Data1.Enabled = False
    Command1.Enabled = False 'create
    Command2.Enabled = False 'delete
    Command3.Enabled = False 'modify
    Command4.Enabled = False 'exit
    Command5.Enabled = True 'cancel
    Command6.Enabled = True 'ok.
    If con_create = False Then 'false=first create record
        If Data1.Recordset.RecordCount > 0 Then
            Data1.Recordset.MoveLast
        End If
        Data1.Recordset.AddNew 'add new 1 record,con_create=true
    End If
    unlock_text
    Text1.SetFocus
    green_text
End Sub

Private Sub Command2_Click() 'delete
Dim msg$
Dim response
    If Data1.Recordset.RecordCount = 0 Then
        MsgBox "no record", 0 + 64
    Exit Sub

```



```
Else
    msg$ = "Do you really want to delete?" ' Define message.
    response = MsgBox(msg$, vbYesNo + vbQuestion + vbDefaultButton1) ' help, ctxt)
    If response = vbNo Then 'User chose yes
        Exit Sub          'exit and do nothing
    End If
    Data1.Recordset.Delete 'del. current record
    Data1.Recordset.MoveNext
    If Data1.Recordset.EOF Then
        If Data1.Recordset.RecordCount = 0 Then
            Exit Sub
        Else
            Data1.Recordset.MoveLast
        End If
    End If
End If
End Sub

Private Sub Command3_Click() 'modify
    If Data1.Recordset.RecordCount = 0 Then
        MsgBox "no record", 0 + 64
        Exit Sub
    End If
    Data1.Enabled = False
    Command1.Enabled = False
    Command2.Enabled = False
    Command3.Enabled = False
    Command4.Enabled = False
    Command5.Enabled = True
    Command6.Enabled = True
```

```
mo_flag = True
con_create = False
unlock_text
yellow_text
Text1.SetFocus
Data1.Recordset.edit    'edit state
End Sub

Private Sub Command4_Click() 'exit
    Unload process
End Sub

Private Sub Command5_Click() 'cancel
If con_create = True Then
    Text1.SetFocus    'cancel for create
    Data1.UpdateControls    'update old value
    If Data1.Recordset.RecordCount > 0 Then
        Data1.Recordset.MoveLast
    End If
    Data1.Enabled = True
    Command1.Enabled = True
    Command2.Enabled = True
    Command3.Enabled = True
    Command4.Enabled = True
    Command5.Enabled = False
    Command6.Enabled = False
    lock_text
    red_text
Else
    Text1.SetFocus    'cancel for modify
    Data1.UpdateControls
```

```

Data1.Enabled = True
Command1.Enabled = True
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
lock_text
red_text
Command5.Enabled = False
Command6.Enabled = False
mo_flag = False
End If
End Sub

Private Sub Command6_Click() 'ok.
Dim MyDB As Database
Dim MyRecordset As Recordset
Dim i As Integer
Dim MyCheck As Boolean
Dim zone_error As Boolean
If con_create Or mo_flag = True Then 'check process name in textbox
Text1.Text = RTrim(Text1.Text) 'process name
If Len(Text1.Text) = 0 Then
MsgBox "you must fill in process name", 0 + 48
Text1.SetFocus
Exit Sub
Else 'has chr. fill in process
Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyRecordset = MyDB.OpenRecordset("process", dbOpenDynaset)
If MyRecordset.RecordCount > 0 Then
MyRecordset.MoveFirst 'movefirst for compare purpose
End If

```

```

If mo_flag = True Then      'compare process name exclude itself
Do While Not MyRecordset.EOF '
    If Data1.Recordset.AbsolutePosition <> MyRecordset.AbsolutePosition Then
        If Text1.Text = MyRecordset.Fields(0) Then 'process name field
            MsgBox "duplicate process name", 0 + 48
            Text1.SetFocus      'set focus at process name
            MyRecordset.Close
            Exit Sub
        End If
    End If
    MyRecordset.MoveNext
Loop
Else          'con_create=true
Do While Not MyRecordset.EOF 'compare process name
    If Text1.Text = MyRecordset.Fields(0) Then
        MsgBox "duplicate process name", 0 + 48
        Text1.SetFocus
        MyRecordset.Close
        Exit Sub
    End If
    MyRecordset.MoveNext
Loop
End If
MyRecordset.Close
Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyRecordset = MyDB.OpenRecordset("station name", dbOpenDynaset)
For index = 0 To 15

MyRecordset.MoveFirst
Do While Not MyRecordset.EOF '
    If DBCombo1(index).Text <> MyRecordset.Fields(0) Then

```

```
        MyRecordset.MoveNext
    Else
        zone_error = False 'if equal 1 in list
        Exit Do
    End If
    zone_error = True 'if not same in all list
Loop
If zone_error = True Then
    MsgBox "some operation is not in the list", 0 + 48
    DBCombo1(index).SetFocus
    MyRecordset.Close
    Exit Sub
End If
Next index
MyRecordset.Close
End If
Data1.Recordset.Update 'update text to record,con_create=false
Data1.Enabled = True
Command1.Enabled = True
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = False
Command6.Enabled = False
lock_text
red_text
Text1.SetFocus
If mo_flag = False Then 'for create
    Data1.Recordset.MoveLast
Else
    mo_flag = False 'for modify
```

```
End If
End If
End Sub

Private Sub Data1_Reposition()
Data1.Caption = "process " & Data1.Recordset.AbsolutePosition + _
1 & " of " & Data1.Recordset.RecordCount
End Sub

Private Sub Data1_Validate(Action As Integer, Save As Integer)
Select Case Action
Case 5          'addnew
con_create = True
Case Else
con_create = False
End Select
End Sub

Private Sub Form_Activate()
If Data1.Recordset.RecordCount > 0 Then
Data1.Recordset.MoveLast 'update data1 caption
Data1.Recordset.MoveFirst
End If
lock_text
red_text
mo_flag = False
Command5.Enabled = False
Command6.Enabled = False
Text1.SetFocus
```

```
Data1.Caption = "process " & Data1.Recordset.AbsolutePosition + _  
1 & " of " & Data1.Recordset.RecordCount
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Left = (Screen.Width - Width) / 2
```

```
Top = (Screen.Height - Height) / 2 - 200
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
product.Enabled = True
```

```
process.Enabled = False
```

```
process.WindowState = 1
```

```
End Sub
```

```
Attribute VB_Name = "producti"
```

```
Attribute VB_GlobalNameSpace = False
```

```
Attribute VB_Creatable = False
```

```
Attribute VB_PredeclaredId = True
```

```
Attribute VB_Exposed = False
```

```
Dim dummy, dummy1
```

```
Dim index As Integer
```

```
Dim nextframe_flag As Boolean
```

```
Private Sub adjust_delay_cmd_Click()
```

```
product.Enabled = False
```

```
adjust_delay_time.Show
```

```
adjust_delay_time.WindowState = 0
```

```
End Sub
```

```
Private Sub define_process_cmd_Click()  
    product.Enabled = False 'disable mainmenu  
    process.Show          'display define process  
    process.WindowState = 0 'process window size  
End Sub
```

```
Private Sub define_product_cmd_Click()  
    product.Enabled = False  
    define_product.Show    'display define product  
    define_product.WindowState = 0  
End Sub
```

```
Private Sub define_station_cmd_Click()  
    product.Enabled = False  
    define_station.Show  
    define_station.WindowState = 0  
End Sub
```

```
Private Sub exit_cmd_Click() 'exit  
    Unload product  
End Sub
```

```
Private Sub Form_Activate() 'this sub routine will be called every time form active  
    Timer1.Enabled = False 'but it will not exit because do loop  
    nextframe_flag = False  
    index = 0  
    Do  
        dummy1 = DoEvents() 'for os. service  
        Picture1(index).Visible = True  
        Timer1.Enabled = True  
    Do          'wait until time out
```



```
dummy = DoEvents() 'return to operating system for service
If nextframe_flag = True Then 'if timeout
    nextframe_flag = False
    Timer1.Enabled = False
    Exit Do      'exit wait loop
End If

Loop

If index = 13 Then
    For index = 0 To 13
        Picture1(index).Visible = False
    Next index
    Timer1.Enabled = True
    Do          'wait until time out
        dummy = DoEvents() 'return to operating system for service
        If nextframe_flag = True Then 'if timeout
            nextframe_flag = False
            Timer1.Enabled = False
            Exit Do      'exit wait loop
        End If
    Loop
    index = 0
Else
    index = index + 1
    If index = 7 Then
        End If
    If index = 13 Then
        End If
    End If
End If

Loop

End Sub
```

```

Private Sub Form_Load()
'Dim i As Integer
Dim MyDB As Database
Dim MyRecordset As Recordset
'Dim x As Integer
'Dim delaytime As Integer
Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyRecordset = MyDB.OpenRecordset("adjust delay time", dbOpenDynaset)
MyRecordset.MoveFirst
' x = 20000 'Val(MyRecordset.Fields(0))
delaytime = Val(MyRecordset.Fields(0))
product.WindowState = 2 'maximize
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
Dim msg$
Dim response
msg$ = "Do you really want to exit ?" ' Define message.
response = MsgBox(msg$, vbYesNo + vbCritical + vbDefaultButton1) ' help, cxt)
If response = vbNo Then ' User chose no
Cancel = 1 'still open form even unload
Exit Sub 'exit main program
Else ' User chose yes
End
End If
End Sub

```

```

Private Sub generate_work_order_cmd_Click()
product.Enabled = False
edit.Enabled = True
edit.Show

```

```
edit.WindowState = 2 'maximize  
End Sub
```

```
Private Sub on_line_monitoring_cmd_Click()  
product.Enabled = False  
run.Enabled = True  
run.Show  
run.WindowState = 2 'maximize  
End Sub
```

```
Private Sub production_report_cmd_Click()  
product.Enabled = False  
report.Enabled = True  
report.Show  
report.WindowState = 2 'maximize  
End Sub
```

```
Private Sub select_pro_cmd_Click()  
product.Enabled = False  
select_process.Show  
select_process.WindowState = 0  
End Sub
```

```
Private Sub Timer1_Timer()  
nextframe_flag = True  
Timer1.Enabled = False  
End Sub
```

```
'Private Sub Toolbar1_ButtonClick(ByVal Button As Button)  
' Select Case Button.Key  
' Case Is = "database" ' Open file.
```

```

' Command1_Click
' Case Is = "run"      ' run
' Command4_Click
' Beep
' Case Is = "exit"    ' exit
' Command3_Click
' End Select
'End Sub

Attribute VB_Name = "report"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub Command1_Click()      'delete workurder
Dim msg$
Dim response
Dim MyDB As Database
Dim MyQuery As QueryDef, MyParameter As Parameter
Dim MyRecordset As Recordset
msg$ = "Do you really want to delete ?" 'Define message.
response = MsgBox(msg$, vbYesNo + vbCritical + vbDefaultButton1) ' help, cxt)
If response = vbNo Then 'User chose no
Exit Sub      'exit sub program
Else          'User chose yes
If Data1.Recordset.RecordCount = 0 Then 'if no record
Exit Sub
Else
Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
Set MyQuery = MyDB.QueryDefs("del1worklist")
' PARAMETERS work_want Text;

```

```

' DELETE *
' From workorder
' WHERE ((workorder.wo=[work_want]));
MyQuery.Parameters("work_want").Value = Data1.Recordset.Fields("workorder").Value
MyQuery.Execute
MyQuery.Close
edit.Data3.DatabaseName = ("c:\program files\microsoft visual basic\db5.MDB")
edit.Data3.RecordSource = "granddb" 'update granddb of edit form
edit.Data1.RecordSource = "workorder" 'update workorder of edit form
edit.Data3.Refresh
edit.Data1.Refresh
'check is if del. last record
If Data1.Recordset.AbsolutePosition + 1 = Data1.Recordset.RecordCount Then 'del last rec.
    Data1.Recordset.Delete
    If Data1.Recordset.RecordCount = 0 Then
        Data1_Reposition 'show 0 value of graph
        Exit Sub
    Else
        Data1.Recordset.MoveLast 'if use movenext error occure(no record)
    End If 'data1.recordset.recordcount=-1
Else
    Data1.Recordset.Delete
    Data1.Recordset.MoveNext
End If
' If Data1.Recordset.EOF Then
' If Data1.Recordset.RecordCount = 0 Then
' Exit Sub
' Else
' Data1.Recordset.MoveLast 'Previous
' End If
' End If

```

```

End If
End If
End Sub

Private Sub Data1_Reposition() 'display graph
Dim MyDB As Database
Dim MyQuery As QueryDef, MyParameter As Parameter
Dim MyQuery1 As QueryDef, MyParameter1 As Parameter
Dim MyRecordset As Recordset
Dim MyRecordset1 As Recordset
Dim X As Integer
If Data1.Recordset.RecordCount > 0 Then 'if has record
    Data1.Caption = "workorder" & Data1.Recordset.AbsolutePosition + 1 & " of " &
Data1.Recordset.RecordCount
    Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
    Set MyRecordset = MyDB.TableDefs("worksum").OpenRecordset(, dbOpenDynaset)
    If Option1.Value = True Then 'select individual station
' For x = 0 To 2      '3 graphs
    X = 0
    For i = 2 To MyRecordset.Fields.Count - 1 Step 4
        If Data1.Recordset.Fields(i).Value <> 0 Then 'input field
            Graph1(X).NumPoints = 3
            Graph1(X).LabelText = "input"
            Graph1(X).LabelText = "output"
            Graph1(X).LabelText = "reject"
            Graph1(X).ColorData = 9
            Graph1(X).GraphData = Data1.Recordset.Fields(i).Value 'input
            Graph1(X).GraphData = Data1.Recordset.Fields(i + 1).Value 'output
            Graph1(X).GraphData = Data1.Recordset.Fields(i + 2).Value 'reject
            Graph1(X).LegendText = Data1.Recordset.Fields(i).Value 'input Value
            Graph1(X).LegendText = Data1.Recordset.Fields(i + 1).Value 'output Value

```

```

Graph1(X).LegendText = Data1.Recordset.Fields(i + 2).Value 'reject Value
Graph1(X).GraphTitle = Data1.Recordset.Fields(i + 3).Name & "=" _
& Format(Data1.Recordset.Fields(i + 3).Value, "##.00") & "%"
Graph1(X).DrawMode = 2
X = X + 1
End If
Next i
' Next x
For X = X To 2      'fill 0 for empty station
    Graph1(X).NumPoints = 3
    Graph1(X).LabelText = "input"
    Graph1(X).LabelText = "output"
    Graph1(X).LabelText = "reject"
    Graph1(X).ColorData = 9
    Graph1(X).GraphData = 0
    Graph1(X).GraphData = 0
    Graph1(X).GraphData = 0
    Graph1(X).LegendText = 0
    Graph1(X).LegendText = 0
    Graph1(X).LegendText = 0
    Graph1(X).GraphTitle = ""
    Graph1(X).DrawMode = 2
Next X
Else      'select all station (sum of each station)
X = 0
For i = 2 To MyRecordset.Fields.Count - 1 Step 4
    If (Data1.Recordset.Fields(i).Value <> 0) Then
        a = 0: B = 0: c = 0:
        MyRecordset.MoveFirst
        Do While Not MyRecordset.EOF
            a = a + MyRecordset.Fields(i).Value      'input

```

```

    B = B + MyRecordset.Fields(i + 1).Value    'output
    c = c + MyRecordset.Fields(i + 2).Value    'reject
    MyRecordset.MoveNext

Loop
Graph1(X).NumPoints = 3
Graph1(X).LabelText = "input"
Graph1(X).LabelText = "output"
Graph1(X).LabelText = "reject"
Graph1(X).ColorData = 9
Graph1(X).GraphData = a    'input
Graph1(X).GraphData = B    'output
Graph1(X).GraphData = c    'reject
Graph1(X).LegendText = a
Graph1(X).LegendText = B
Graph1(X).LegendText = c
Graph1(X).GraphTitle = Data1.Recordset.Fields(i + 3).Name & "=" _
& Format(str((B / a) * 100), "##.00") & "%" 'yield in percent format
Graph1(X).DrawMode = 2
X = X + 1
End If
Next i
For X = X To 2    'fill 0 for empty station
    Graph1(X).NumPoints = 3
    Graph1(X).LabelText = "input"
    Graph1(X).LabelText = "output"
    Graph1(X).LabelText = "reject"
    Graph1(X).ColorData = 9
    Graph1(X).GraphData = 0
    Graph1(X).GraphData = 0
    Graph1(X).GraphData = 0
    Graph1(X).LegendText = 0

```



```
Graph1(X).LegendText = 0
Graph1(X).LegendText = 0
Graph1(X).GraphTitle = ""
Graph1(X).DrawMode = 2
Next X
End If          'end select option
Else           'no record to display graph
Data1.Caption = Data1.Recordset.AbsolutePosition + 1 & " of " _
& Data1.Recordset.RecordCount
For X = X To 2      'fill 0 for empty station
Graph1(X).NumPoints = 3
Graph1(X).LabelText = "input"
Graph1(X).LabelText = "output"
Graph1(X).LabelText = "reject"
Graph1(X).ColorData = 9
Graph1(X).GraphData = 0
Graph1(X).GraphData = 0
Graph1(X).GraphData = 0
Graph1(X).LegendText = 0
Graph1(X).LegendText = 0
Graph1(X).LegendText = 0
Graph1(X).GraphTitle = ""
Graph1(X).DrawMode = 2
Next X
End If
End Sub

Private Sub Command3_Click() 'exit
Unload report
End Sub
```

```
Private Sub Form_Activate()  
    Option1.Value = True 'select individual  
End Sub  
  
Private Sub Form_Load()  
    Dim MyDB As Database  
    Dim MyRecordset As Recordset  
    Dim i As Integer  
    Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")  
    Set MyRecordset = MyDB.TableDefs("worksum").OpenRecordset(, dbOpenDynaset)  
    If MyRecordset.RecordCount > 0 Then  
        MyRecordset.MoveFirst  
        For i = 2 To MyRecordset.Fields.Count - 1  
            DBGrid2.Columns.Add (i)          'add column to dbgrid2  
            DBGrid2.Columns(i).Width = 900  
            DBGrid2.Columns(i).Visible = True  
            DBGrid2.Columns(i).Caption = MyRecordset.Fields(i).Name  
            DBGrid2.Columns(i).DataField = MyRecordset.Fields(i).Name 'bound data field  
        Next i  
    End If  
    MyRecordset.Close  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    product.Enabled = True  
End Sub  
  
Private Sub Option1_Click()  
    Data1_Reposition  
End Sub
```

```
Private Sub Option2_Click()
```

```
    Data1_Reposition
```

```
End Sub
```

```
Attribute VB_Name = "select_process"
```

```
Attribute VB_GlobalNameSpace = False
```

```
Attribute VB_Creatable = False
```

```
Attribute VB_PredeclaredId = True
```

```
Attribute VB_Exposed = False
```

```
Private Sub Command1_Click() 'exit
```

```
    Unload select_process
```

```
End Sub
```

```
Private Sub DBCombo1_change() 'change current process
```

```
Dim MyDB As Database
```

```
Dim MyQuery As QueryDef, MyParameter As Parameter
```

```
Dim MyRecordset As Recordset
```

```
Dim i As Integer
```

```
Set MyDB = Workspaces(0).OpenDatabase("c:\program files\microsoft visual basic\db5.MDB")
```

```
Set MyQuery = MyDB.QueryDefs("process_operator")
```

```
'PARAMETERS process_want Text;
```

```
'SELECT process.*, *
```

```
'From process
```

```
'WHERE (((process.process)=[process_want]));
```

```
MyQuery.Parameters("process_want").Value = DBCombo1.Text 'Set parameters.
```

```
Set MyRecordset = MyQuery.OpenRecordset(, dbOpenDynaset) 'Open Recordset.
```

```
List2.Clear 'clear station list
```

```
If MyRecordset.RecordCount > 0 Then
```

```
    For i = 1 To (MyRecordset.Fields.Count) - 1
```

```
        If MyRecordset.Fields(i) <> "no" Then "no"=end of station
```

```
List2.AddItem MyRecordset.Fields(i) 'add to station list
End If
Next i
End If
MyRecordset.Close
End Sub
```

```
Private Sub Form_Load()
    Left = (Screen.Width - Width) / 2 'make form to center
    Top = (Screen.Height - Height) / 2 ' - 200
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    product.Enabled = True 'enable main menu
    select_process.Enabled = False
    select_process.WindowState = 1 'minimize
End Sub
```

1

2

105

105

## รายละเอียดของโปรแกรม Single sift key unit

```

menuchoice    equ 07fh ;config,lotid,in,out.reject,def1,def2..
list_pointer  equ 07dh ;
sum           equ 07eh ;communication check sum for check error
rs_status     equ 07dh ;communication state
submenu1     equ 07ch ;
offset        equ 07bh ;
offset_rejhi  equ 07ah ;high byte of reject offset compare with reject list
offset_rejlo  equ 079h ;low byte of reject offset compare with reject list
def_pointer   equ 078h ;
root1         equ 077h ;config submenu
root5         equ 076h ;reject submenu
template     equ 075h ;temporary buffer of defect type
offset_inhi   equ 074h ;high byte of input value
offset_inlo   equ 073h ;low byte of input value
offset_outhi  equ 072h ;high byte of output value
offset_outlo  equ 071h ;low byte of output value

lot_buf       equ 0c000h ;
id_buf        equ 0c008h ;external memory id buffer
input_buf     equ 0c018h ;external memory input buffer
output_buf    equ 0c020h ;external memory output buffer
reject_buf    equ 0c028h ;total reject count
reject1_buf   equ 0c030h ;reject type1 amount
reject2_buf   equ 0c038h ;reject type2 amount
reject3_buf   equ 0c040h ;reject type3 amount
reject4_buf   equ 0c048h ;reject type4 amount
reject5_buf   equ 0c050h ;reject type5 amount
password_buf  equ 0c058h ;buffer for user key in
password_store equ 0c060h ;buffer for station in compare purpose

```

```
mc_buf      equ 0c068h ;station number

reject_list equ 0c100h ;begin of reject list
rs_buf      equ 0c200h ;

key0      equ 01h ;00h
key1      equ 02h ;01h
key2      equ 00h ;02h
key3      equ 04h ;03h

key4      equ 06h ;05h
key5      equ 07h ;06h
key6      equ 05h ;07h
key7      equ 09h ;08h

key8      equ 0bh ;0ah
key9      equ 0ch ;0bh
clr_key   equ 010h ;0fh
;tab_key   equ 010h ;bh

wr_key    equ 03h ;04h
up_key    equ 0ah ;0ch
down_key  equ 0fh ;011h
load_key  equ 0eh ;0dh

enter_key equ 012h ;0eh
rej_inc   equ 0dh ;013h
rej_dec   equ 08h ;09h
right_key equ 013h ;012h
left_key  equ 011h ;010h
```

```

first_key_bit equ 07fh ;clear for skiping identify key purpose
rs_get_flag equ 07eh ;set for receive a character
load_or_save equ 07dh
commu_bit equ 07ch ;set for is in communication status
already_save_bit equ 07bh
sound_bit equ 07ah ;set for sound,clear for read tag logic(timer0)
read_bit equ 079h ;set for is in reading tag status
in_oroutbit equ 078h ;identify between reader1 and reader2
edit_enablebit equ 077h ;set for enable edit mode
wrong_passbit equ 076h ;set for show access deny
add_bit equ 075h
empty_bit equ 074h
mcerror_bit equ 073h ;set for show station number error
welcome_bit equ 072h ;set for show welcome password
rej_error equ 071h ;set for show reject>input
success_readbit equ 070h
virvisual_keybit equ 06fh ;simulate a key by clr carry flag
zrej_bit equ 06eh ;set for show [ ] ***** Q

```

```

START: ORG 0000H
        LJMP MAIN
        ORG 0003H
        LJMP SWREAD
        ORG 000BH
        jmp rdtag
        ORG 0023H
        LJMP RS232
SOUND: CPL P1.2
        RETI

```

```
RDTAG:  jb sound_bit,on_sound
        CLR IE.7
        PUSH ACC
        jb in_oroutbit,rd_out
        MOV C,P3.4
rd_out:  mov c,p3.3
rd_com:  MOV A,066H
        RLC A
        INC R0
        CJNE R0,#08H,LOOP1
        MOV R0,#00H
        MOV @R1,A
        CJNE R1,#5FH,LOOP1
        MOV R1,#30H
LOOP1:  POP ACC
        POP PSW
        SETB IE.7
        reti
on_sound: cpl p1.2
        RETI
```

```
SWREAD: MOV R5,#0FFH
        RETI
RS232:  JNB RI,LOOP2
        CLR RI
LOOP2:  JNB TI,LOOP3
        SETB 01H
LOOP3:  RETI
COMMAND: push dph
```



```
push dpl
CLR P1.0
nop
nop
nop
nop
nop
nop
nop
mov dptr,#and to lcd
pop dpl
pop dph
RET
```

SD\_DATA: push dph

```
push dpl
SETB P1.0
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
mov dptr,#8000h
movx @dptr,a
clr p1.0
pop dpl
pop dph
RET
```

```
SET_LCD: MOV A,#005H
          LCALL DELAY
          MOV R4,#001H
LOOP4:   MOV A,#00aH
          LCALL DELAY
          MOV A,#038H
          MOV A,#005H
          LCALL DELAY
          MOV A,#038H
          LCALL COMMAND
          MOV A,#032H
LOOP5:   DEC A
          MOV A,#038H
          LCALL COMMAND
          DEC R4
          CJNE R4,#00,LOOP4
          MOV A,#005H
          LCALL DELAY
          MOV A,#038H
          LCALL COMMAND
          MOV A,#010H
          LCALL DELAY
          MOV A,#008H
          LCALL COMMAND
          MOV A,#010H
          LCALL DELAY
          MOV A,#001H
          MOV A,#010H
          LCALL DELAY
          MOV A,#006H
          LCALL DELAY
```

```
    MOV A,#002H
    LCALL COMMAND
    MOV A,#010H
    LCALL DELAY
    MOV A,#0dh ;CH
    LCALL COMMAND
; MOV R4,#000H
; RET

newchar: mov r2,#00h
         mov r3,#040h;
lop25:  mov a,#02h
         lcall delay
         mov a,r3
         lcall command
         mov a,#02h
         lcall delay
         mov a,r2
         inc r3
         inc r2
         cjne r2,#040h,lop25
         MOV A.#002H
         LCALL COMMAND
         ret

SD_OUT:  PUSH 006H
         MOV R6,A
         LCALL RS_SND
         CJNE R4,#00H,LOOP6
         MOV A,#001H
         LCALL COMMAND
```

```
        MOV A,#002H
        LCALL COMMAND
        LJMP LOOP7
LOOP6:  CJNE R4,#008H,LOOP8
        MOV A,#0C0H
        LCALL COMMAND
LOOP7:  MOV A,#01H
        LCALL DELAY
LOOP8:  MOV A,R6
        LCALL SD_DATA
        CJNE R4,#010H,LOOP9
LOOP9:  POP 06
        RET

SD_OUT1: PUSH 006H
        MOV R6,A
        ; LCALL RS_SND
        CJNE R4,#00H,LCOP611
        MOV A,#001H
        MOV A,#002H
        LJMP LOOP711
LOOP611:CJNE R4,#008H,LOOP811
        MOV A,#0C0H
LOOP711:MOV A,#01H
        LCALL DELAY
LOOP811: MOV A,R6
        MOV A,#01H
        LCALL DELAY
        INC R4
        MOV R4,#00H
LOOP911: POP 06
```

RET

```
LOOP24: MOV tl0,#0B5H
        MOV th0,#0B5H
LOOP25: SETB et0
        SETB tr0
        JB 08H,LOOP26
        MOV A,#01BH
        LJMP LOOP27
LOOP26: MOV A,#015H
LOOP27: LCALL DELAY
        CLR tr0
        CLR et0
        JB 08H,LOOP28
        MOV A,#02H
        LJMP LOOP29
LOOP28: MOV A,#02H
LOOP29: LCALL DELAY
        MOV TMOD,#00100010B
        JB 08H,LOOP30
        MOV tl0,#088H
        MOV th0,#088H
        LJMP LOOP31
LOOP30: MOV tl0,#0B5H
        MOV th0,#0B5H
LOOP31: SETB et0
        SETB tr0
        JB 08H,LOOP32
        MOV A,#01BH
        LJMP LOOP33
```

```
LOOP32: MOV A,#015H
LOOP33: LCALL DELAY
        CLR tr0
        CLR et0
        ; LCALL RS_ON
        RET

BAD_BEEP: JB 08,LOOP34
          MOV A,#02H
          LJMP LOOP35
LOOP34:  MOV A,#02H
LOOP35:  LCALL DELAY
          MOV TMOD,#00100010B
          JB 08H,LOOP36
          MOV tl0,#022H
          MOV th0,#022H
          LJMP LOOP37
LOOP36:  MOV tl0,#091H
          MOV th0,#091H
LOOP37:  SETB et0
          SETB tr0
          JB 08H,LOOP38
          MOV A,#025H
          LJMP LOOP39
LOOP38:  MOV A,#01CH
LOOP39:  LCALL DELAY
          CLR tr0
          CLR et0
          ; LCALL RS_ON
          RET
```

```
DELAY:  PUSH 03
        PUSH 02
        MOV R3,A
DELAY2:  MOV R2,#0FFH
DELAY3:  DEC R2
        NOP
        CJNE R2,#00,DELAY3
        DEC R3
        CJNE R3,#00,DELAY2
        POP 02
        POP 03
        RET
```

```
PREPARE: MOV TMOD,#022H
        SETB EA
        clr ex1
        CLR TR0
        ;SETB ET0
        MOV 08AH,#038H
        MOV 08CH,#038H
        ret
```

```
say_tat equ $
db ' TN. Co.,Ltd. '
di 00h
```

```
say_config equ $
db '1.Config.[Enier]'
db 00h
```

```
say_config1 equ $  
db '1:Config' :'  
db 00h
```

```
say_id EQU $  
db '2.Id: ' :'  
db 00h
```

```
SAY_input EQU $  
DB '3.Input : ' :'  
db 00h
```

```
SAY_output EQU $  
DB '4.Output: ' :'  
db 00h
```

```
say_reject equ $  
db '5.Reject: ' :'  
db 00h
```

```
say_reject1 equ $  
db '6. Def[1]: ' :'  
db 00h
```

```
say_reject2 equ $  
db '7. Def[2]: ' :'  
db 00h
```

```
say_reject3 equ $  
db '8. Def[3]: ' :'  
db 00h
```



```
say_reject4 equ $  
db '9. Def[4]:  '  '  
db 00h
```

```
say_reject5 equ $  
db '10. Def[5]:  '  '  
db 00h
```

```
say_balance_error equ $  
db ' Balance Error '  '  
db 00h
```

```
say_looking equ $  
db ' Looking '  '  
db 00h
```

```
say_se_reject equ $  
db '[ ] 12345 Q '  '  
db 00h
```

```
say_zero_reject equ $  
db '[ ] ***** Q '  '  
db 00h
```

```
say_no_reject equ $  
db ' No Reject '  '  
db 00h
```

```
say_password equ $  
db '*Password*:  '  '  
db 00h
```

```
db 00h
```

```
say_mc equ $
```

```
db '1.Station ID: '
```

```
db 00h
```

```
say_newpass equ $
```

```
db '2.Password: '
```

```
db 00h
```

```
say_exit equ $
```

```
db '4.Exit [Enter]'
```

```
db 00h
```

```
say_wrongpass equ $
```

```
db ' Access Deny '
```

```
db 00h
```

```
say_edit equ $
```

```
db '3.Edit [ ] '
```

```
db 00h
```

```
say_mc_error equ $
```

```
db ' ID. Error '
```

```
db 00h
```

```
say_welcome equ $
```

```
db ' * Welcome * '
```

```
db 00h
```

```
say_load equ $
```

```
db '11.PC<-->Station'  
db 00h
```

```
say_reject_error equ $  
db ' Reject > Input '  
db 00h
```

```
say_notexist equ $  
db ' Lot Not Exist '  
db 00h
```

```
say_inputerror equ $  
db ' Input Sum Error'  
db 00h
```

```
say_pcsumerror equ $  
db ' Pc. Sum Error '  
db 00h
```

```
say_commu_error equ $  
db ' Comu. Error '  
db 00h
```

```
say_commu equ $  
db ' Communicating..' '  
db 00h
```

```
;FOR GET CHARACTER FROM TABLE TO DISPLAY  
;EXAMPLE FOR DIS."LOW BATTERY"
```

```
;
SENTENCE_LCD:MOV A,#000H
LOOP300:  PUSH ACC
          MOVC A,@A+DPTR
          JZ LOOP301
          LCALL SD_OUT
          POP ACC
          INC A
          SJMP LOOP300
LOOP301:  POP ACC
          ; MOV A,#00DH
          ; MOV A,#00AH
          RET
```

```
clr_all_buf:call clr_lot_buf
            ; call clr_id_buf
            ; call clr_in_buf
            ; call clr_out_buf
            ; call clr_rej_buf
; call clr_rej1_buf
; call clr_rej2_buf
; call clr_rej3_buf
; call clr_rej4_buf
; call clr_rej5_buf
            call clr_password_buf
            ret
```

```
clr_lot_buf:mov dptr,#lot_buf
            mov r0,#06h
            lcall all_clr
            ret
```

```
clr_id_buf:mov dptr,#id_buf
            mov r0,#0bh
            lcall all_clr
            ret

clr_in_buf:mov dptr,#input_buf
            mov r0,#06h
            lcall all_clr
            ret

clr_out_buf:mov dptr,#output_buf
            mov r0,#06h
            lcall all_clr
            ret

clr_rej_buf:mov dptr,#reject_buf
            mov r0,#06h
            lcall all_clr
            ret

clr_rej1_buf:mov dptr,#reject1_buf
            mov r0,#06h
            lcall all_clr
            ret

clr_rej2_buf:mov dptr,#reject2_buf
            mov r0,#06h
            lcall all_clr
            ret

clr_rej3_buf:mov dptr,#reject3_buf
            mov r0,#06h
            lcall ail_clr
            ret

clr_rej4_buf:mov dptr,#reject4_buf
            mov r0,#06h ;6h
            lcall all_clr
```

```
        ret
clr_rej5_buf:mov dptr,#reject5_buf
        mov r0,#06h ;6h
        lcall all_clr
        ret
clr_password_buf:mov dptr,#password_buf
        mov r0,#06h
        lcall all_clr
        ret
clr_password_store:mov dptr,#password_store
        mov r0,#06h
        lcall all_clr
        ret
clr_mc_buf:mov dptr,#mc_buf
        mov r0,#06h
        lcall all_clr
        ret

all_clr: mov a,#0f0h
clr_next: movx @dptr,a
        djnz r0,clr_next
        ret

MAIN:   mov submenu1,#00
        mov root1,#00h
        mov root5,#00h
        mov def_pointer,#05h
        clr zrej_bit
        clr virsual_keybit
        clr p3.
        clr rej_error
```

```
clr welcome_bit
setb add_bit
clr empty_bit
clr wrong_passbit
clr edit_enablebit
clr read_bit
clr 08h
mov rs_status,#00h
clr first_key_bit
clr already_save_bit
lcall clr_all_buf
LCALL SET_LCD
LCALL PREPARE
MOV DPTR,#say_tat
MOV R4,#00H
LCALL SENTENCE_LCD
lcall good_beep
main1: lcall jb first_key_bit,normal_key
clr wrong_passbit
clr merror_bit
clr welcome_bit
clr rej_error
jmp skip_iden
normal_key:mov a,b
lcall idenkey
skip_iden:jb read_bit,main1
lcall
jmp main1

scankey: mov r7,#010h
```

```
        jnc scankey
        djnz r7,scnx
scloop: lcall
        push b
        push acc
        push psw
        jnb read_bit,skip_read
        LCALL PREPARE
        MOV DPTR,#say_looking
        MOV R4,#00H
        LCALL SENTENCE_LCD
        lcall read_tag
        setb sound_bit
skip_read:jnb commu_bit,skip_rs_service
        lcall rs_service
skip_rs_service:pop psw
        pop acc
        pop b
        jb virsual_keybit,vkey_press
        jc scloop
        ret
vkey_press:clr virsual_keybit
        mov b,#00
        ret

dis_commu:mov dptr,#say_commu
commu_dis:MOV R4,#00H
        LCALL SENTENCE_LCD
        ret
```



```
dis_next1_1: jmp dis_next1
display: ; mov a,mainmenu
        ; jb readerr_bit,dis_err
        jb commu_bit,dis_commu
        mov a,menuchoice
        cjne a,#00h,dis_next1_1
        mov a,root1
        cjne a,#00h,next_root1
        jb wrong_passbit,dis_wrongpass
        mov dptr,#say_config
        MOV R4,#00H
        LCALL SENTENCE_LCD
        mov a,#0c2h
        lcall command
        ret
dis_wrongpass: mov dptr,#say_wrongpass
               MOV R4,#00H
               LCALL SENTENCE_LCD
               ret
next_root1: cjne a,#01h,next_root2
            mov dptr,#say_password
            MOV R4,#00H
            LCALL SENTENCE_LCD
            mov dptr,#password_buf
            mov a,#0c3h
            lcall command
            mov r1,#05h
            lcall dis_digit
            ret
next_root2: cjne a,#02h,next_root3
            jnb welcome_bit,not_welcome
```

```
    LCALL good_bEEP
    mov dptr,#say_welcome
    MOV R4,#00H
    LCALL SENTENCE_LCD
    ret
not_welcome:mov a,submenu1
    cjne a,#00h,show_next1
    jnb mCerror_bit,mc_not_error
    mov dptr,#say_mc_error
    MOV R4,#00H
    LCALL SENTENCE_LCD
    ret
mc_not_error:mov dptr,#say_mc
    MOV R4,#00H
    LCALL SENTENCE_LCD
    mov dptr,#mc_buf
    mov a,#0c5h
    lcall command
    mov r1,#03h
    lcall dis_digit
    ret
show_next1:cjne a,#01h,show_next2
    mov dptr,#say_newpass
    MOV R4,#00H
    LCALL SENTENCE_LCD
    mov dptr,#password_store
    mov a,#0c3h
    lcall command
    mov r1,#05h
    lcall dis_digit
    ret
```

```

show_next2:cjne a,#02h,show_next3
    mov dptr,#say_edit
    MOV R4,#00H
    LCALL SENTENCE_LCD
    mov a,#0c4h
    lcall command
    jnb edit_enablebit,skip_mark
    mov a,#078h
    lcall sd_data
    mov a,#0c4h
    lcall command
skip_mark:ret
show_next3:cjne a,#03h,show_next4
    mov dptr,#say_exit
    MOV R4,#00H
    LCALL SENTENCE_LCD
    mov a,#0c1h
    lcall command
    ret
show_next4:ret
next_root3: ret
show_char:mov dptr,#say_config
show_char1:MOV R4,#00H
    LCALL SENTENCE_LCD
    ret

dis_next1:cjne a,#01h,dis_next2
; jnb already_save_bit,show_char2
    mov dptr,#say_id
; jmp show_char3
;show_char2:mov dptr,#say_id

```

```
show_char3:MOV R4,#00H
            LCALL SENTENCE_LCD
            mov dptr,#id_buf
            lcall dis_digit1
            ret

dis_next2:cjne a,#02h,dis_next3
            mov dptr,#say_input
            MOV R4,#00H
            LCALL SENTENCE_LCD
            mov a,#0c2h
            lcall command
            mov r1,#06h
            lcall dis_digit
            ret

dis_next7: cjne a,#07h,dis_next8
            mov dptr,#say_reject3
            MOV R4,#00H
            LCALL SENTENCE_LCD
            mov dptr,#reject3_buf
            mov a,#0c2h
            lcall command
            mov r1,#06h
            lcall dis_digit
            ret

dis_next8: cjne a,#08h,dis_next9
            mov dptr,#say_reject4
            MOV R4,#00H
            LCALL SENTENCE_LCD
```

```

    mov dptr,#reject4_buf
    mov a,#0c2h
    lcall command
    mov r1,#06h      ;
    lcall dis_digit  ;
    ret

```

```
dis_next9: cjne a,#09h,dis_next10
```

```

    mov dptr,#say_reject5
    MOV R4,#00H
    LCALL SENTENCE_LCD
    mov dptr,#reject5_buf
    mov a,#0c2h
    lcall command
    mov r1,#06h
    lcall dis_digit
    ret

```

```
dis_next10: cjne a,#0ah,dis_next11
```

```

    mov dptr,#say_load      MOV R4,#00H
    LCALL SENTENCE_LCD
    mov a,#0c7h
    lcall command
    ret

```

```
dis_next11:ret
```

```
entry_dis:jb zrej_bit,dis_aster ;
```

```

    mov dptr,#say_se_reject ;[ ] 12345 q
    jmp sel_discon

```

```

dis_aster:mov dptr,#say_zero_reject ;[ ] ***** q
sel_discon:MOV R4,#00H
            LCALL SENTENCE_LCD
            mov a,#082h
            lcall command
            mov a,list_pointer
            mov 061h,a
            mov 060h,#00
            lcall hex_bcd
            mov r1,#04h
            mov r0,#031h
next_charlist: mov a,@r0
            MOV A,@r0
            add a,#030h
            lcall sd_data
            mov a,#02h
            lcall delay
            inc r0
            djnz r1,next_charlist
            jnb zrej_bit,se_discon
; mov def_pointer,#05h
; mov template,#01h
            mov a,#0c6h
            lcall command
            ret
se_discon:mov a,list_pointer
            clr c
            add a,dpl
            mov dpl,a
            movx a,@dptr
            mov template,a

```

```
        mov a,template
first_sel:jnb acc.0,sec_sel
        mov a,#0c0h
        lcall command
        mov a,#01h
        lcall sd_data
sec_sel: mov a,template
        jnb acc.1,third_sel
        mov a,#0c1h
        lcall command
        mov a,#02h
        lcall sd_data
third_sel:mov a,template
        jnb acc.2,fourth_sel
        mov a,#0c2h
        lcall command
        mov a,#03h
fourth_sel:mov a,template
        jnb acc.3,fifth_sel
        mov a,#0c3h
        lcall command
        mov a,#04h
        lcall sd_data
fifth_sel:mov a,template
        jnb acc.4,sixth_sel
        mov a,#0c4h
        lcall command
        mov a,#05h
        lcall sd_data
sixth_sel:mov a,def_pointer
        cjne a,#00,next_point1
```

```
        mov a,#0c0h
        lcall command
        ret
        ret
dis_con:  cjne r1,#03h,next_char1
        mov a,#0c0h
        lcall command
        jmp next_char1

down_arrow1: jmp down_arrow
idenkey:  mov a,b
next_key10:  cjne a,#clr_key,down_arrow1
        mov a,menuchoice
        cjne a,#00,next_clr1
        mov a,root1
        cjne a,#00,next_clr_root1
        ret
next_clr_root1:  cjne a,#01h,next_clr_root2
        lcall clr_password_buf
        ret
next_clr_root2:  cjne a,#02h,next_clr_root3
        mov a,submenu1
        cjne a,#00h,next_clr_root2_1
        lcall clr_mc_buf
        ret
next_clr_root2_1:  cjne a,#01h,next_clr_root2_2
        lcall clr_password_store
        ret
next_clr_root2_2:  cjne a,#02h,next_clr_root3
        clr edit_enablebit
        ret
```



```
next_clr_root3:ret

next_clr1:cjne a,#01,next_clr2
    jb edit_enablebit,edit_clr_id
    ret
edit_clr_id:lcall clr_id_buf
    ret

next_clr2:cjne a,#02,next_clr3
    jb edit_enablebit,edit_clr_in
    ret
edit_clr_in:lcall clr_in_buf
    ret

next_clr3:cjne a,#03,next_clr4
    jb edit_enablebit,edit_clr_out ;
    ret
edit_clr_out:lcall clr_out_buf
    ret

next_clr4:cjne a,#04,next_clr5
    mov a,root5
    cjne a,#00h,clr4_1
    jb edit_enablebit,edit_clr_rej
    ret
edit_clr_rej:lcall clr_rej_buf
clr4_1: mov template,#00h
    ret

next_clr5:cjne a,#05,next_clr6
    jb edit_enablebit,edit_clr_rej1
```

```
        ret
edit_clr_rej1:lcall clr_rej1_buf
        ret

next_clr6:cjne a,#06,next_clr7
        jb edit_enablebit,edit_clr_rej2
        ret
edit_clr_rej2:lcall clr_rej2_buf
        ret

next_clr7:cjne a,#07,next_clr8
        jb edit_enablebit,edit_clr_rej3
        ret
edit_clr_rej3:lcall clr_rej3_buf
        ret

next_clr8:cjne a,#08,next_clr9
        jb edit_enablebit,edit_clr_rej4
        ret
edit_clr_rej4:lcall clr_rej4_buf
        ret

next_clr9:cjne a,#09,next_clr10
        jb edit_enablebit,edit_clr_rej5
        ret
edit_clr_rej5:lcall clr_rej5_buf
        ret

next_clr10:cjne a,#0ah,next_clr11
        clr read_bit
        clr commu_bit
```

```
        clr visual_keybit
        mov rs_status,#00h
        ret
next_clr11:ret

down_arrow:cjne a,#down_key,loop70
        mov a,menuchoice
        cjne a,#00h,check_nextmenu1
        mov a,root1
        cjne a,#00h,inc_menuroot1
        inc menuchoice
        ret
inc_menuroot1:cjne a,#01h,inc_menuroot2
        rei
inc_menuroot2:cjne a,#02h,inc_menuroot2_1
        mov a,submenu1
        cjne a,#00h,inc_menuroot2_1
        mov dptr,#mc_buf+2
        movx a,@dptr
        anl a,#0fh
        cjne a,#00,inc_menuroot2_2
        mov dptr,#mc_buf+1
        movx a,@dptr
        anl a,#0fh
        cjne a,#00,inc_menuroot2_2
        mov dptr,#mc_buf
        movx a,@dptr
        anl a,#0fh
        cjne a,#00,inc_menuroot2_2
        clr first_key_bit
        setb merror_bit
```

```
        ret
inc_menuroot2_1:cjne a,#03h,inc_menuroot2_2
        ret
inc_menuroot2_2:
        inc submenu1
        ret
check_nextmenu1:cjne a,#04h,check_nextmenu2
        mov a,root5
        cjne a,#00h,inc_root5_2
        inc menuchoice
        ret
inc_root5_2:
        mov a,list_pointer
        cjne a,#01h,dec_list
        ret
dec_list:dec list_pointer
        ret
check_nextmenu2:cjne a,#0ah,check_nextmenu3
        ret
check_nextmenu3:inc menuchoice
        ret

loop70: cjne a,#up_key,loop72
        mov a,menuchoice
        cjne a,#00h,check_nextdec1
        mov a,root1
        cjne a,#00h,dec_menuroot1
        ret
dec_menuroot1:cjne a,#01h,dec_menuroot2
        ret
```

```
dec_menuroot2:cjne a,#02h,dec_menuroot2_2
    mov a,submenu1
    cjne a,#00h,dec_menuroot2_1
    ret

dec_menuroot2_1:dec submenu1
dec_menuroot2_2:ret

check_nextdec1:cjne a,#04h,check_nextdec2
    mov a,root5
    cjne a,#00h,dec_root5_2
    dec: menuchoice
    ret

dec_root5_2:
    mov dptr,#reject_buf+2
    lcall receive_lcd
    clr c
    subb a,list_pointer
    jnz check_carry
    ret

check_carry:jnc inc_list
    ret

inc_list:inc list_pointer
    ret

    ret

editmode9:mov dptr,#reject5_buf
    mov r0,#05h
    lcall fill_in1
    ret

next_menu10:ret
```

```
fill_in1: ;mov r0,#05h
fill_in6_1:inc dptr
    movx a,@dptr
    dec dpl
    movx @dptr,a
    inc dpl
    djnz r0,fill_in6_1
    mov a,b
    movx @dptr,a
    ret

mode4_1: cjne a,#01h,mode4_2
    mov a,b
    cjne a,#01,next_mode4_1
    mov a,template
    cpl acc.0
    jmp end_mode4

next_mode4_1:cjne a,#02,next_mode4_2
    mov a,template
    cpl acc.1
    jmp end_mode4

next_mode4_2:cjne a,#03,next_mode4_3
    mov a,template
    cpl acc.2
    jmp end_mode4

next_mode4_3:cjne a,#04,next_mode4_4
    mov a,template
    cpl acc.3
    jmp end_mode4

next_mode4_4:cjne a,#05,mode4_2
    mov a,tempiate
```

```
        cpl acc.4
end_mode4:mov template,a
        lcall updatetoggle
mode4_2: ret

next_key11:cjne a,#wr_key,next_key13
        ; setb commu_bit
        ; setb load_or_save
        ; mov rs_status,#00h
        ret

next_key13:cjne a,#load_key,next_key14
        ; mov a,menuchoice
        ; cjne a,#01,set_readerr
        ; cpl read_bit
        ret

enter10: jmp enter1
next_key150:jmp next_key15
next_key14:cjne a,#enter_key,next_key150
        mov a,menuchoice
        cjne a,#00,enter10
        mov a,root1
        cjne a,#00h,enter_root1
        inc root1
        ret

enter_root1:cjne a,#01h,enter_root2
        mov dptr,#password_buf+4
        movx a,@dptr
        mov r0,a
```

```
mov dptr,#password_store+4
movx a,@dptr
cjne a,00,pre_exit
mov dptr,#password_buf+3
movx a,@dptr
mov r0,a
mov dptr,#password_store+3
movx a,@dptr
cjne a,00,pre_exit
mov dptr,#password_buf+2
movx a,@dptr
mov r0,a
mov dptr,#password_store+2
movx a,@dptr
cjne a,00,pre_exit
mov dptr,#password_buf+1
movx a,@dptr
mov r0,a
mov dptr,#password_store+1
movx a,@dptr
cjne a,00,pre_exit
mov dptr,#password_buf
movx a,@dptr
mov r0,a
mov dptr,#password_store
movx a,@dptr
cjne a,00,pre_exit
inc root1
lcall clr_password_buf
clr first_key_bit
setb welcome_bit
```



```
ret
```

```
dec_1list: mov a,offset_rejlo
           dec offset_rejlo
           mov 061h,offset_rejlo
           mov 060h,#00
           lcall hex_bcd
           mov a,034h
           mov dptr,#reject_buf+5
           movx @dptr,a
           mov a,033h
           mov dptr,#reject_buf+4
           movx @dptr,a
           mov a,032h
           mov dptr,#reject_buf+3
           movx @dptr,a
           mov a,031h
           mov dptr,#reject_buf+2
           movx @dptr,a
           lcall find_output
           ret
```

```
find_output:mov dptr,#input_buf+2
            lcall receive_lcd
            mov offset_inlo,a
            clr c
            mov r0,offset_rejlo
            subb a,r0
            jnc good_con1
            lcall bad_beep
```

```
    clr c
    clr first_key_bit
    setb rej_error
    ret
good_con1: mov 061h,a
           mov 060h,#00
           mov dptr,#output_buf
           lcall write_back
update2:  mov dptr,#reject_buf+2
           lcall update_rej_type
           ret

toggle:   mov r0,template
           mov a,def_pointer
toggle0:  cjne a,#00h,toggle1
           mov a,r0
           cpl acc.0
           mov template,a
           lcall updatetoggle
           ret
toggle1:  cjne a,#01h,toggle2
           mov a,r0
           cpl acc.1
           mov template,a
           lcall updatetoggle
           ret
toggle2:  cjne a,#02h,toggle3
           mov a,r0
           cpl acc.2
           mov template,a
           lcall updatetoggle
```

```
    ret
toggle3: cjne a,#03h,toggle4
    mov a,r0
    cpl acc.3
    mov template,a
    lcall updatetoggle
    ret
toggle4: cjne a,#04h,toggle5
    mov a,r0
    cpl acc.4
    mov template,a
    lcall updatetoggle
    ret
toggle5: ret
do_nothing1:ret

updatetoggle:mov a,list_pointer
    mov dptr,#reject_list
    clr c
    add a,dpl
    mov dpl,a
    mov a,template
    movx @dptr,a
    ret

enter2:  cjne a,#0ah,enter3
    lcall good_beep
    cpl read_bit
    jb read_bit,enter4
    clr p3.5
    clr commu_bit
```

```
        ret
enter4:
        MOV DPTR,#say_looking
        MOV R4,#00H
        LCALL SENTENCE_LCD

enter3: ret

next_key16_1: jmp next_key16
next_key15:cjne a,#rej_inc,next_key16_1
        mov a,menuchoice
        cjne a,#04,do_nothing
        mov a,root5
        cjne a,#00h,rejinc_root5
dc_nothing:ret
rejinc_root5:cjne a,#01h,rejinc_root6
        mov a,template
        clr zrej_bit
        cjne a,#00h,rejinc_con
        ret
rejinc_con:mov dptr,#reject_buf+2
        lcall receive_lcd
        mov offset_rejlo,a

        mov dptr,#input_buf+2
        lcall receive_lcd
        mov offset_inlo,a
        clr c
        mov r0,offset_rejlo
        inc r0
```

```
    subb a,r0
    jnc good_con
    lcall bad_bEEP
    clr c
    clr first_key_bit
    setb rej_error
    ret
good_con: mov 061h,a
          mov 060h,#00
          mov dptr,#output_buf
          lcall write_back
          mov a,offset_rejlo
          mov template,#00h
          lcall addto_list
          mov list_pointer,offset_rejlo
          ret
rejinc_root6:ret      ;

addto_list:mov a,offset_rejlo
            inc offset_rejlo
            mov 061h,offset_rejlo
            mov 060h,#00
            lcall hex_bcd
            mov a,034h
            mov dptr,#reject_buf+5
            movx @dptr,a
            mov a,033h
            mov dptr,#reject_buf+4
            movx @dptr,a
            mov a,032h
            mov dptr,#reject_buf+3
```

```
movx @dptr,a
mov a,031h
mov dptr,#reject_buf+2
movx @dptr,a
mov a,offset_rejlo
mov dptr,#reject_list
clr c
add a,dpi
mov dpl,a
mov a,template
movx @dptr,a
ret
```

```
next_key16:cjne a,#rej_dec,next_key17
```

```
mov a,menuchoice
cjne a,#04,do_nothing2
mov a,root5
cjne a,#00h,rejdec_root5
ret
```

```
rejdec_root5:cjne a,#01h,rejdec_roct6
```

```
mov dptr,#reject_buf+2
lcall receive_lcd
mov offset_rejlo,a
cjne a,#01h,del_shift1
lcall dec_1list
setb zrej_bit
mov def_pointer,#05h
mov template,#01h
mov list_pointer,offset_rejlo
; mov template,#00h
ret
```

```
del_shift1:cjne a,#00h,del_shift
    ret
del_shift:mov a,offset_rejlo
    mov r0,list_pointer
    clr c
    subb a,r0
    jnz del_shift_con1
    lcall dec_1list
    mov list_pointer,offset_rejlo
    ret
del_shift_con1:mov r0,a
    push list_pointer
shift_con2:mov a,list_pointer
    inc a
    mov dptr,#reject_list
    clr c
    add a,dpl
    mov dpl,a
    movx a,@dptr
    mov template,a
    mov a,list_pointer
    mov dptr,#reject_list
    clr c
    add a,dpl
    mov dpl,a
    mov a,template
    movx @dptr,a
    inc list_pointer
    djnz r0,shift_con2
    lcall dec_1list
    pop list_pointer
```

```
rejdec_root6:ret
do_nothing2:ret

next_key17:cjne a,#right_key,next_key18
    mov a,menuchoice
    cjne a,#04,do_nothing2
    jb zrej_bit,skip_right
    mov a,def_pointer
    cjne a,#05h,inc_defpoint
skip_right:ret
inc_defpoint:inc def_pointer
    ret

next_key18:cjne a,#left_key,next_key19
    mov a,menuchoice
    cjne a,#04,do_nothing2
    jb zrej_bit,skip_left
    mov a,def_pointer
    cjne a,#00h,dec_defpoint
skip_left: ret
dec_defpoint:dec def_pointer
next_key19:ret

hex_bcd: push dph
    push dpl
    mov r0,#030h
    MOV 065H,#00H
LOP14: MOV DPTR,#TABLE1
    MOV A,065H
    ADD A,065H
    PUSH ACC
```



```
    MOVC A,@A+DPTR
    MOV 062H,A
    POP ACC
    INC A
    MOVC A,@A+DPTR
    MOV 063H
    LCALL DEVIDE
LOP16: MOV A,064H
        mov @r0,a
        inc r0
LOP17: INC 065H
        MOV A,065H
        CJNE A,#05H,LOP14
        pop dpl
        pop dph
        RET

receive_lcd:;mov dptr,#reject_buf+2
            movx a,@dptr
            anl a,#00fh
            swap a
            mov r0,a
            inc dptr
; mov dptr,#reject_buf+3
            movx a,@dptr
            anl a,#00fh
            orl a,r0
            mov r0,a
            inc dptr
; mov dptr,#reject_buf+4
            movx a,@dptr
```

```
    anl a,#00fh
    swap a
    mov r1,a
    inc dptr
; mov dptr,#reject_buf+5
    movx a,@dptr
    anl a,#00fh
    orl a,r1
    mov r1,a
    mov 060h,r0
    mov 061h,r1
bcd_hex: mov a,061h
    anl a,#0f0h
    rrc a
    mov r0,a
    rrc a
    rrc a
    add a,r0
    mov r0,a
    mov a,061h
    anl a,#0fh
    add a,r0
; mov offset_rejlo,a
    ret

update_rej_type:
    lcall receive_lcd
    cjne a,#00h,update_con
    jmp update_con1
update_con:mov r0,a
    mov r1,#00
```

```
        mov r2,#00
        mov r3,#00
        mov r4,#00
        mov r5,#00
extrac: mov a,r0
        mov dptr,#reject_list
        clr c
        add a,dpl
        mov dpl,a
        movx a,@dptr
extrac0: jnb acc.0,extrac1
        inc r1
extrac1: jnb acc.1,extrac2
        inc r2
extrac2: jnb acc.2,extrac3
        inc r3
extrac3: jnb acc.3,extrac4
        inc r4
extrac4: jnb acc.4,extrac_con
        inc r5
extrac_con:djnz r0,extrac

        jmp rd_com7
rd_out7:mov rs_status,#020h
rd_com7:ret

not_recei2:ret

id_rs6: cjne a,#06h,id_rs7
        lcall rs_get1
        jb rs_get_flag,not_recei2
```

```
        cjne a,#019h,rs6_con1
        mov dptr,#say_notexist
        jmp rs6_con2
rs6_con1:cjne a,#021h,rs6_con
        mov dptr,#say_pcsumerror
rs6_con2:clr commu_bit
        MOV R4,#00H
        LCALL SENTENCE_LCD
        lcall bad_beep
        clr first_key_bit
        mov rs_status,#00h
        ret
rs6_con:mov b,a
        mov sum,#00h
        clr c
        subb a,#030h
        mov 030h,a
        mov sum,a
        mov rs_status,#07h
        ret

id_rs7: cjne a,#07h,id_rs8
        lcall rs_get1
        jb rs_get_flag,not_recei2
;      mov dptr,#input_buf+4
        clr c
        subb a,#030h
        mov 031h,a
        clr c
        add a,sum
        mov sum,a
```

```
    mov rs_status,#08h
```

```
    ret
```

```
id_rs8: cjne a,#08h,id_rs9
```

```
    lcall rs_get1
```

```
    jb rs_get_flag,not_recei2
```

```
;    mov dptr,#input_buf+3
```

```
    clr c
```

```
    subb a,#030h
```

```
    mov 032h,a
```

```
    clr c
```

```
    add a,sum
```

```
    mov sum,a
```

```
    mov rs_status,#09h
```

```
    ret
```

```
id_rs9: cjne a,#09h,id_rs9_1
```

```
    lcall rs_get1
```

```
    jb rs_get_flag,not_recei2
```

```
    clr c
```

```
    subb a,#030h
```

```
    mov 033h,a
```

```
    clr c
```

```
    add a,sum
```

```
    mov sum,a
```

```
    mov rs_status,#01eh
```

```
    ret
```

```
id_rs9_1: cjne a,#01eh,id_rs10
```

```
    lcall rs_get1
```

```
    jb rs_get_flag,not_recei5
```

```
cjne a,sum,inputerror
mov a,#03h
lcall delay
setb p1.7
mov a,#01h
lcall delay
mov a,#01eh
lcall rs_snd
mov a,#02h
lcall delay
mov a,#0dh
lcall rs_snd
mov a,#02h
lcall delay
clr p1.7
```

```
mov dptr,#input_buf+2
mov r0,#033h
mov r1,#04h
```

```
transfer:mov a,@r0
movx @dptr,a
inc dptr
dec r0
djnz r1,transfer
clr commu_bit
clr first_key_bit
setb virsual_keybit
mov rs_status,#00h
lcall good_beep
ret
```

```
inputerror:
```

```
mov a,#03h
lcall delay
setb p1.7
mov a,#01h
lcall delay
mov a,#01fh
lcall rs_snd
mov a,#02h
lcall delay
mov a,#0dh
lcall rs_snd
mov a,#02h
lcall delay
clr p1.7
```

```
clr commu_bit
mov dptr,#say_inputerror
MOV R4,#00H
LCALL SENTENCE_LCD
lcall bad_beep
clr first_key_bit
mov rs_status,#00h ;06h
ret
```

```
not_recei5:ret
```

```
id_rs11: cjne a,#021h,id_rs12
lcall rs_get1
jb rs_get_flag,not_recei3
cjne a,#019h,rs11_con1
mov dptr,#say_notexist
```

```
        jmp rs11_con2
rs11_con1:cjne a,#021h,rs11_con3
        mov dptr,#say_pcsumerror
        jmp rs11_con2

rs11_con3:cjne a,#0fh,rs11_con4
        mov dptr,#say_balance_error
        jmp rs11_con2

rs11_con4:cjne a,#01eh,rs11_con5
        clr commu_bit
        setb virsual_keybit
        mov rs_status,#00h
        lcall good_beep
        ret

rs11_con5:mov dptr,#say_commu_error
rs11_con2:clr commu_bit
        MOV R4,#00H
        LCALL SENTENCE_LCD
        lcall bad_beep
        clr first_key_bit
        mov rs_status,#00h ;06h
        ret

id_rs12:ret

not_recei3:ret

snd_con10:mov r0,#04h
        mov r1,#00h
```



```
snd_con8:movx a,@dptr
          anl a,#0fh
          mov r2,a
          clr c
          add a,sum
          mov sum,a
          mov a,r2
          clr c
          add a,#030h
          lcall rs_snd
          inc dptr
          djnz r0,snd_con8
          ret
```

```
err_recei:mov rs_status,#00h
not_recei:ret
```

```
scan2:  push 00h
        push 01h
        push 02h
        push 03h
        mov r0,#00
        mov r1,#11110111b
        mov r2,#04h
kcol:  mov r3,#05h
        setb p1.3
        setb p1.4
        setb p1.5
```

```
    setb p1.6
    mov a,r1
    anl p1,a
    lcall coldel
    push dph
    push dpl
    mov dptr,#08000h
    movx a,@dptr
    pop dpl
    pop dph
    anl a,#00011111b
krow: rrc a
      jc nokey
      mov b,r0
      clr c
      jmp end
nokey: inc r0
      djnz r3,krow
      mov a,r1
      rl a
      mov r1,a
      djnz r2,kcol
      setb c
end:  pop 03h
      pop 02h
      pop 01h
      pop 00h
      ret

coldel: nop
       nop
```

```
nop
nop
nop
nop
nop
nop
nop
ret
```

```
read_tag: cpl in_oroutbit
          setb p3.5
          clr sound_bit
          setb tr0
          setb et0
          setb ea
          LCALL SYNC_IDI
          JC LOOP57
HERE1:   CJNE R1,#03BH,HERE1
          CLR TR0
          LCALL MANCHESTER
          JC LOOP57
          LCALL PARITY
          jc loop57
          LCALL SHOW_IDI
          setb sound_bit
          lcall good_beep
          clr read_bit
          setb success_readbit
          setb commu_bit
          mov rs_status,#00h
          clr p3.5
```

```

        ret
loop57: ret

SYNC_IDI: MOV R0,#00H
          MOV R1,#030H
          MOV A,#035H
LOOP64:  DEC
          NOP
          JNZ LOOP64
;        JB 08H,LOOP65
          MOV A,#09EH
;        LJMP LOOP66
;LOOP65:  MOV A,#078H
LOOP66:  JNZ LOOP67
          LJMP LOOP86
LOOP67:  DEC A
          jb in_oroutbit,rd_out1
          JB P3.4,LOOP66
          jmp rd_com1
rd_out1: jb p3.3,loop66
rd_com1: MOV A,#078h
          ; JB P3.4,LOOP66
          ; MOV A,#078H
LOOP77:  JNZ LOOP78
          LJMP LOOP86
LOOP78:  DEC A

          jb in_oroutbit,rd_out2
          JnB P3.4,LOOP77
          jmp rd_com2
rd_out2: jnb p3.3,loop77

```

```
RET
RET

SHOW_data: MOV R4,#00H
           MOV R1,#040h ;70H
           ; MOV A,#049H ;SEND IDI*
           ; LCALL SD_OUT
           ; MOV A,#044H
           ; LCALL SD_OUT
           ; MOV A,#049H
           ; LCALL SD_OUT
           ; MOV A,#02AH
           ; LCALL SD_OUT
LOOP191:   MOV A,@R1
           mov @r1,#00h
           CLR 0E7H
           INC R1
           LCALL BI_ASCII
LOOP192:   CJNE R1,#045H,LOOP191
           setb p1.7
           mov a,#03
           lcall delay
           clr p1.7

           ; MOV A,#00AH
           setb tr1
           ; LCALL RS_SND
           ; CLR C
           RET
           END
```

## ประวัติผู้เขียน

นาย วรพจน์ ไพจิตร เกิดวันที่ 8 มกราคม พ.ศ. 2505 ที่จังหวัดแพร่ สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง ในปีการศึกษา 2527 และ สำเร็จการศึกษาระดับปริญญาโทบริหารธุรกิจมหาบัณฑิต สาขาการเงิน จากมหาวิทยาลัยเกษตรศาสตร์ ในปีการศึกษา 2535 ได้เข้าศึกษาต่อในหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2539

