

```

'=====
' File:                                ULDO01.VB
' Library Call Demonstrated:          MccDaq.MccBoard.DOut()
' Purpose:                            Writes a byte to digital output ports.
' Demonstration:                      Configures FirstPortA for output and writes
'                                     a value to the port.
' Other Library Calls:                MccDaq.MccBoard.DConfigPort()
'                                     MccDaq.MccService.ErrHandling()
' Special Requirements:               Board 0 must have a digital output port.
'                                     Board 0 must have programmable digital ports.
'=====
Option Strict Off
Option Explicit On
Imports System.IO

Public Class frmSetDigOut

    Inherits System.Windows.Forms.Form
    Public Vergleichswert As Single
    Public Koordinaten(2, 2) As Integer
    Public Befehlsnummer As Integer
    Const PortNum As MccDaq.DigitalPortType = MccDaq.DigitalPortType.FirstPortA ' set port A to use
    Const PortNum2 As MccDaq.DigitalPortType = MccDaq.DigitalPortType.FirstPortB ' set port B to use
    Const Direction As MccDaq.DigitalPortDirection = MccDaq.DigitalPortDirection.DigitalOut
    Public WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents Button2 As System.Windows.Forms.Button ' program digital port A for output

    'Create a new MccBoard object for Board 0
    Private DaqBoard As MccDaq.MccBoard = New MccDaq.MccBoard(0)
    Friend WithEvents lblShowData As System.Windows.Forms.Label

    Private RangeSelected As MccDaq.Range
    Friend WithEvents tbpfad As System.Windows.Forms.TextBox
    Public WithEvents lblDataValOut As System.Windows.Forms.Label
    Friend WithEvents Button3 As System.Windows.Forms.Button
    Friend WithEvents Button4 As System.Windows.Forms.Button
    Friend WithEvents Button5 As System.Windows.Forms.Button
    Friend WithEvents Button6 As System.Windows.Forms.Button
    Friend WithEvents Stift As System.Windows.Forms.Button
    Friend WithEvents Button7 As System.Windows.Forms.Button
    Friend WithEvents Button8 As System.Windows.Forms.Button

    Private Property minLeange As Object

    Private Property Leange As Object
    Dim ULStat As MccDaq.ErrorInfo

    Private Sub cmdEndProgram_Click(ByVal eventSender As System.Object, ByVal eventArgs As System.EventArgs) ↵
        Handles cmdEndProgram.Click

        Dim DataValue As UInt16
        DataValue = Convert.ToUInt16(0)

        ULStat = DaqBoard.DOut(PortNum, DataValue)
        If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then Stop

    End

End Sub

#Region "Windows Form Designer generated code "
    Public Sub New()
        MyBase.New()

```

```

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        InitUL()

    End Sub
    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(ByVal Disposing As Boolean)
        If Disposing Then
            If Not components Is Nothing Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(Disposing)
    End Sub
    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
    Public ToolTip1 As System.Windows.Forms.ToolTip
    Public WithEvents cmdEndProgram As System.Windows.Forms.Button
    Public WithEvents lblShowValOut As System.Windows.Forms.Label
    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container()
        Me.ToolTip1 = New System.Windows.Forms.ToolTip(Me.components)
        Me.cmdEndProgram = New System.Windows.Forms.Button()
        Me.lblShowValOut = New System.Windows.Forms.Label()
        Me.Button1 = New System.Windows.Forms.Button()
        Me.Button2 = New System.Windows.Forms.Button()
        Me.lblShowData = New System.Windows.Forms.Label()
        Me.tbpfad = New System.Windows.Forms.TextBox()
        Me.lblDataValOut = New System.Windows.Forms.Label()
        Me.Button3 = New System.Windows.Forms.Button()
        Me.Button4 = New System.Windows.Forms.Button()
        Me.Button5 = New System.Windows.Forms.Button()
        Me.Button6 = New System.Windows.Forms.Button()
        Me.Stift = New System.Windows.Forms.Button()
        Me.Button7 = New System.Windows.Forms.Button()
        Me.Button8 = New System.Windows.Forms.Button()
        Me.SuspendLayout()
        '
        'cmdEndProgram
        '
        Me.cmdEndProgram.BackColor = System.Drawing.SystemColors.Control
        Me.cmdEndProgram.Cursor = System.Windows.Forms.Cursors.Default
        Me.cmdEndProgram.Font = New System.Drawing.Font("Arial", 8.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.cmdEndProgram.ForeColor = System.Drawing.SystemColors.ControlText
        Me.cmdEndProgram.Location = New System.Drawing.Point(404, 167)
        Me.cmdEndProgram.Name = "cmdEndProgram"
        Me.cmdEndProgram.RightToLeft = System.Windows.Forms.RightToLeft.No
        Me.cmdEndProgram.Size = New System.Drawing.Size(57, 33)
        Me.cmdEndProgram.TabIndex = 7
        Me.cmdEndProgram.Text = "Quit"
        Me.cmdEndProgram.UseVisualStyleBackColor = False
        '
        'lblShowValOut
        '
        Me.lblShowValOut.BackColor = System.Drawing.SystemColors.Window
        Me.lblShowValOut.Cursor = System.Windows.Forms.Cursors.Default
        Me.lblShowValOut.Font = New System.Drawing.Font("Arial", 8.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.lblShowValOut.ForeColor = System.Drawing.Color.Blue
        Me.lblShowValOut.Location = New System.Drawing.Point(401, 53)
        Me.lblShowValOut.Name = "lblShowValOut"
        Me.lblShowValOut.RightToLeft = System.Windows.Forms.RightToLeft.No
        Me.lblShowValOut.Size = New System.Drawing.Size(57, 17)
        Me.lblShowValOut.TabIndex = 3
        '
        'Button1
        '
        Me.Button1.Location = New System.Drawing.Point(386, 119)

```

```
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(75, 23)
Me.Button1.TabIndex = 8
Me.Button1.Text = "Drucken"
Me.Button1.UseVisualStyleBackColor = True
'
'Button2
'
Me.Button2.Location = New System.Drawing.Point(386, 73)
Me.Button2.Name = "Button2"
Me.Button2.Size = New System.Drawing.Size(75, 23)
Me.Button2.TabIndex = 9
Me.Button2.Text = "Vorschau"
Me.Button2.UseVisualStyleBackColor = True
'
'lblShowData
'
Me.lblShowData.AutoSize = True
Me.lblShowData.Location = New System.Drawing.Point(32, 46)
Me.lblShowData.Name = "lblShowData"
Me.lblShowData.Size = New System.Drawing.Size(116, 14)
Me.lblShowData.TabIndex = 10
Me.lblShowData.Text = "Dateipfad eingeben:"
'
'tbpfad
'
Me.tbpfad.Location = New System.Drawing.Point(35, 73)
Me.tbpfad.Name = "tbpfad"
Me.tbpfad.Size = New System.Drawing.Size(258, 20)
Me.tbpfad.TabIndex = 11
Me.tbpfad.Text = "C:\Users\student\Desktop\attachment.obj"
'
'lblDataValOut
'
Me.lblDataValOut.BackColor = System.Drawing.SystemColors.Window
Me.lblDataValOut.Cursor = System.Windows.Forms.Cursors.Default
Me.lblDataValOut.Font = New System.Drawing.Font("Arial", 8.0!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.lblDataValOut.ForeColor = System.Drawing.SystemColors.WindowText
Me.lblDataValOut.Location = New System.Drawing.Point(224, 27)
Me.lblDataValOut.Name = "lblDataValOut"
Me.lblDataValOut.RightToLeft = System.Windows.Forms.RightToLeft.No
Me.lblDataValOut.Size = New System.Drawing.Size(185, 17)
Me.lblDataValOut.TabIndex = 2
'
'Button3
'
Me.Button3.Location = New System.Drawing.Point(306, 12)
Me.Button3.Name = "Button3"
Me.Button3.Size = New System.Drawing.Size(88, 24)
Me.Button3.TabIndex = 12
Me.Button3.Text = "Papiereinzug"
Me.Button3.UseVisualStyleBackColor = True
'
'Button4
'
Me.Button4.Location = New System.Drawing.Point(194, 22)
Me.Button4.Name = "Button4"
Me.Button4.Size = New System.Drawing.Size(88, 24)
Me.Button4.TabIndex = 13
Me.Button4.Text = "Papieraussschub"
Me.Button4.UseVisualStyleBackColor = True
'
'Button5
'
Me.Button5.Location = New System.Drawing.Point(344, 222)
Me.Button5.Name = "Button5"
Me.Button5.Size = New System.Drawing.Size(75, 23)
Me.Button5.TabIndex = 14
Me.Button5.Text = "links"
Me.Button5.UseVisualStyleBackColor = True
'
'Button6
'
```

```

    Me.Button6.Location = New System.Drawing.Point(344, 270)
    Me.Button6.Name = "Button6"
    Me.Button6.Size = New System.Drawing.Size(75, 23)
    Me.Button6.TabIndex = 15
    Me.Button6.Text = "rechts"
    Me.Button6.UseVisualStyleBackColor = True
    '
    'Stift
    '
    Me.Stift.Location = New System.Drawing.Point(153, 221)
    Me.Stift.Name = "Stift"
    Me.Stift.Size = New System.Drawing.Size(75, 23)
    Me.Stift.TabIndex = 16
    Me.Stift.Text = "Button7"
    Me.Stift.UseVisualStyleBackColor = True
    '
    'Button7
    '
    Me.Button7.Location = New System.Drawing.Point(153, 221)
    Me.Button7.Name = "Button7"
    Me.Button7.Size = New System.Drawing.Size(75, 23)
    Me.Button7.TabIndex = 16
    Me.Button7.Text = "Stift runter"
    Me.Button7.UseVisualStyleBackColor = True
    '
    'Button8
    '
    Me.Button8.Location = New System.Drawing.Point(171, 289)
    Me.Button8.Name = "Button8"
    Me.Button8.Size = New System.Drawing.Size(75, 23)
    Me.Button8.TabIndex = 17
    Me.Button8.Text = "Stift hoch"
    Me.Button8.UseVisualStyleBackColor = True
    '
    'frmSetDigOut
    '
    Me.AutoScaleBaseSize = New System.Drawing.Size(6, 13)
    Me.BackColor = System.Drawing.SystemColors.Window
    Me.ClientSize = New System.Drawing.Size(473, 404)
    Me.Controls.Add(Me.Button8)
    Me.Controls.Add(Me.Button7)
    Me.Controls.Add(Me.Button6)
    Me.Controls.Add(Me.Button5)
    Me.Controls.Add(Me.Button4)
    Me.Controls.Add(Me.Button3)
    Me.Controls.Add(Me.tbpfad)
    Me.Controls.Add(Me.lblShowData)
    Me.Controls.Add(Me.Button2)
    Me.Controls.Add(Me.Button1)
    Me.Controls.Add(Me.cmdEndProgram)
    Me.Controls.Add(Me.lblShowValOut)
    Me.Controls.Add(Me.lblDataValOut)
    Me.Cursor = System.Windows.Forms.Cursors.Default
    Me.Font = New System.Drawing.Font("Arial", 8.25!, System.Drawing.FontStyle.Bold, System.Drawing.
GraphicsUnit.Point, CType(0, Byte))
    Me.ForeColor = System.Drawing.SystemColors.WindowText
    Me.Location = New System.Drawing.Point(7, 103)
    Me.Name = "frmSetDigOut"
    Me.RightToLeft = System.Windows.Forms.RightToLeft.No
    Me.StartPosition = System.Windows.Forms.FormStartPosition.Manual
    Me.Text = "Druckprogramm"
    Me.ResumeLayout(False)
    Me.PerformLayout()

End Sub
#End Region

#Region "Universal Library Initialization - Expand this region to change error handling, etc."

Private Sub InitUL()
    Dim ULStat As MccDaq.ErrorInfo

    ' declare revision level of Universal Library

```

```

    ULStat = MccDaq.MccService.DeclareRevision(MccDaq.MccService.CurrentRevNum)

    ' Initiate error handling
    ' activating error handling will trap errors like
    ' bad channel numbers and non-configured conditions.
    ' Parameters:
    '     MccDaq.ErrorReporting.PrintAll :all warnings and errors encountered will be printed
    '     MccDaq.ErrorHandling.StopAll  :if any error is encountered, the program will stop

    ULStat = MccDaq.MccService.ErrHandling(MccDaq.ErrorReporting.PrintAll, MccDaq.ErrorHandling.StopAll)
    If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
        Stop
    End If

    ' configure FirstPortA for digital output
    ' Parameters:
    '     PortNum      :the output port
    '     Direction    :sets the port for input or output
    ULStat = DaqBoard.DConfigPort(PortNum, Direction)
    ULStat = DaqBoard.DConfigPort(PortNum2, Direction)
    If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then Stop

End Sub
#End Region

Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2. Click
    Dim dateipfad As String
    Dim inhalt As String

    dateipfad = tbpfad.Text
    '***** Beginn Dateiauslesen *****
    Dim objDateiLeser As StreamReader

    objDateiLeser = New StreamReader(dateipfad)
    inhalt = objDateiLeser.ReadToEnd()

    Dim position1, position2, zaehler As Integer

    zaehler = 0
    position1 = 0
    position2 = InStr(inhalt, vbNewLine)
    If position2 > 0 Then
        Do
            position2 = InStr(position2 + 1, inhalt, vbNewLine)
            zaehler = zaehler + 1
        Loop Until zaehler = 0 Or position2 = 0

    End If
    Dim Zeile(0 To zaehler) As String
    zaehler = 0
    position2 = InStr(inhalt, vbNewLine)

    If position2 > 0 Then
        Zeile(zaehler) = Mid(inhalt, position1 + 1, position2 - 1)

        Do
            position1 = InStr(position1 + 1, inhalt, vbNewLine)
            position2 = InStr(position2 + 1, inhalt, vbNewLine)

            If position1 > 0 AndAlso position2 > 0 Then
                zaehler = zaehler + 1

                Zeile(zaehler) = Mid(inhalt, position1 + 2, position2 - position1 - 2)
            ElseIf position1 > position2 Then

```

```

        zaehler = zaehler + 1

        Zeile(zaehler) = Mid(inhalt, position1 + 2, Len(inhalt) - position1 - 1)

    End If

    Loop Until zaehler = 0 Or position1 = 0
End If
'*****Kreisbefehle zählen*****
Dim anzahlbefehle As Integer
anzahlbefehle = zaehler
Dim Kreisbefehlposition As Integer = 0
Dim Kreisbefehlanzahl As Integer = 0
Dim Kreislinien As Integer = 0
Dim Anzahlkreislinien As Integer = 500
Dim einenmehr As Integer
einenmehr = 1

Do
    Kreisbefehlposition = InStr(Kreisbefehlposition + 1, inhalt, "CS")
    If Kreisbefehlposition > 0 Then
        Kreislinien = Kreislinien + Anzahlkreislinien
        Kreisbefehlanzahl = Kreisbefehlanzahl + 1
    End If
Loop Until Kreisbefehlposition = 0
Dim Kreiskoordinaten(1, 0 To Kreislinien - 1) As Integer

'*****Zähle die Anzahl der PA Befehle*****
Dim PositionPA As Integer = 0
Dim AnzahlPABefehle As Integer = 0
Do
    PositionPA = InStr(PositionPA + 1, inhalt, "PA")

    If PositionPA > 0 Then
        AnzahlPABefehle = AnzahlPABefehle + 1
    End If
Loop Until PositionPA = 0

'*****Zähle Anzahl von PU/PD*****
Dim PositionPU As Integer = 0
Dim PositionPD As Integer = 0
Dim AnzahlPUPDBefehle As Integer = 0
Do
    PositionPU = InStr(PositionPU + 1, inhalt, "PU")

    If PositionPU > 0 Then
        AnzahlPUPDBefehle = AnzahlPUPDBefehle + 1
    End If
Loop Until PositionPU = 0
Do
    PositionPD = InStr(PositionPD + 1, inhalt, "PD")

    If PositionPD > 0 Then
        AnzahlPUPDBefehle = AnzahlPUPDBefehle + 1
    End If
Loop Until PositionPD = 0

objDateiLeser.Close()
objDateiLeser = Nothing

'*****Ende des Ausleseprogrammabschnittes*****

'*****Beginn Befehlsauswertung *****

Dim befehl As String

```

```

Dim n, PU, PA, PD, CS, Koordinatex, Koordinatey, MittelpunktX, MittelpunktY, Winkel As Integer

Dim erstezahl, letztezahl As Integer
Dim ObenUnten, MomentaneAnzahlPUPD, MomentaneAnzahlCS As Integer

MomentaneAnzahlCS = 0
MomentaneAnzahlPUPD = 0
PU = 0
PA = 0
PD = 0
CS = 0
n = 0
Befehlsnummer = 0

Do

    befehl = Zeile(n)

    PU = InStr(befehl, "PU")
    PA = InStr(befehl, "PA")
    PD = InStr(befehl, "PD")
    CS = InStr(befehl, "CS")

    '*****Befehlsauswertung PU/PD *****

    If PU = 1 AndAlso PD = 0 AndAlso PA = 0 AndAlso CS = 0 Then
        ObenUnten = 0

    ElseIf PU = 0 AndAlso PD = 1 AndAlso PA = 0 AndAlso CS = 0 Then
        ObenUnten = 1

    '*****Ende PU / PD Befehlsauswertung*****

    '*****PA Befehlsauswertung*****
    ElseIf PU = 0 AndAlso PD = 0 AndAlso PA = 1 AndAlso CS = 0 Then

        '*****PA X-Koordinate *****
        erstezahl = 1 'Position der Ersten Zahl der X-Koordinate wird von Links (1)
        gesucht

        letztezahl = InStr(befehl, ",") 'Position der Letzten Zahl der X-Koordinate wird vom Komma
        aus gesucht

        If Asc(Mid(befehl, erstezahl, 1)) < 48 Or Asc(Mid(befehl, erstezahl, 1)) > 57 Then
            Do Until Asc(Mid(befehl, erstezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, erstezahl, 1)) <= 57
                57 'Falls Leerzeichen oder Buchstaben zwischen Start und Zahl dann verschiebe Position um eins nach
                rechts
                erstezahl = erstezahl + 1
            Loop
        End If

        'letztezahl ist position des Kommas
        If Asc(Mid(befehl, letztezahl, 1)) < 48 Or Asc(Mid(befehl, letztezahl, 1)) > 57 Then
            Do Until Asc(Mid(befehl, letztezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, letztezahl, 1)) <= 57
                57 'Beginne am Ende und gehe nach links bis x8 eine Zahl wird
                letztezahl = letztezahl - 1
            Loop
        End If

        Koordinatex = CInt(Mid(befehl, erstezahl, letztezahl - erstezahl + 1))

        Befehlsnummer = Befehlsnummer + 1

```

```

ReDim Preserve Koordinaten(2, 0 To Befehlsnummer + 1)

Koordinaten(0, Befehlsnummer) = ObenUnten
Koordinaten(1, Befehlsnummer) = Koordinatex

'*****PA Y-Koordinate *****

    erstezahl = InStr(befehl, ",") 'Position der Ersten Zahl der X-Koordinate ✓
wird vom Komma aus gesucht

    letztezahl = Len(befehl) 'Position der Letzten Zahl der Y-Koordinate ✓
wird vom Ende aus gesucht

    If Asc(Mid(befehl, erstezahl, 1)) < 48 Or Asc(Mid(befehl, erstezahl, 1)) > 57 Then
        Do Until Asc(Mid(befehl, erstezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, erstezahl, 1)) <= 57
57 'Falls Leerzeichen oder Buchstaben zwischen Start und Zahl dann verschiebe Position um eins nach rechts ✓
        erstezahl = erstezahl + 1
    Loop
End If

'letztezahl ist am Ende von befehl positioniert
If Asc(Mid(befehl, letztezahl, 1)) < 48 Or Asc(Mid(befehl, letztezahl, 1)) > 57 Then
    Do Until Asc(Mid(befehl, letztezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, letztezahl, 1)) <= 57
57 'Beginne am Ende und gehe nach links bis linkezahl eine Zahl wird ✓
    letztezahl = letztezahl - 1
    Loop
End If
Koordinatey = CInt(Mid(befehl, erstezahl, letztezahl - erstezahl + 1))
Koordinaten(2, Befehlsnummer) = Koordinatey

'*****Ende PA Befehlsauswertung*****✓
*****

'*****Beginn des Kreisbefehlsauslesen*****✓
*****
    ElseIf PU = 0 AndAlso PD = 0 AndAlso PA = 0 AndAlso CS = 1 Then

        MomentaneAnzahlCS = MomentaneAnzahlCS + 1

        '*****CS X-Koordinate *****
        erstezahl = 1 'Position der Ersten Zahl der X-Koordinate wird von Links (1)✓
gesucht

        letztezahl = InStr(befehl, ",") 'Position der Letzten Zahl der X-Koordinate wird vom Komma ✓
aus gesucht

        If Asc(Mid(befehl, erstezahl, 1)) < 48 Or Asc(Mid(befehl, erstezahl, 1)) > 57 Then
            Do Until Asc(Mid(befehl, erstezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, erstezahl, 1)) <= 57
57 'Falls Leerzeichen oder Buchstaben zwischen Start und Zahl dann verschiebe Position um eins nach rechts ✓
            erstezahl = erstezahl + 1
        Loop
    End If

    'letztezahl ist position des Kommas
    If Asc(Mid(befehl, letztezahl, 1)) < 48 Or Asc(Mid(befehl, letztezahl, 1)) > 57 Then
        Do Until Asc(Mid(befehl, letztezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, letztezahl, 1)) <= 57
57 'Beginne am Ende und gehe nach links bis x8 eine Zahl wird ✓
        letztezahl = letztezahl - 1
        Loop
    End If
    Mittelpunktx = CInt(Mid(befehl, erstezahl, letztezahl - erstezahl + 1))

```



```

'*****CS Y-Koordinate *****
erstezahl = InStr(befehl, ",") 'Position der Ersten Zahl der Y-Koordinate
wird nach dem ersten Komma gesucht

letztezahl = InStr(erstezahl + 1, befehl, ",") 'Mittelpunkt y-koordinate wird zwischen erstem
und zweitem Komma gesucht

If Asc(Mid(befehl, erstezahl, 1)) < 48 Or Asc(Mid(befehl, erstezahl, 1)) > 57 Then
    Do Until Asc(Mid(befehl, erstezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, erstezahl, 1)) <= 57
57 'Falls Leerzeichen oder Buchstaben zwischen Start und Zahl dann verschiebe Position um eins nach
rechts
        erstezahl = erstezahl + 1
    Loop
End If

'letztezahl ist position des Kommas
If Asc(Mid(befehl, letztezahl, 1)) < 48 Or Asc(Mid(befehl, letztezahl, 1)) > 57 Then
    Do Until Asc(Mid(befehl, letztezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, letztezahl, 1)) <= 57
57 'Beginne am Ende und gehe nach links bis x8 eine Zahl wird
        letztezahl = letztezahl - 1
    Loop
End If
Mittelpunkt = CInt(Mid(befehl, erstezahl, letztezahl - erstezahl + 1))

'*****Winkel CS *****

erstezahl = letztezahl + 1 'Position der ersten Zahl wird nach dem zweiten
komma gesucht

letztezahl = Len(befehl) 'Position der Letzten Zahl des Winkels wird
vom Ende aus gesucht

If Asc(Mid(befehl, erstezahl, 1)) < 48 Or Asc(Mid(befehl, erstezahl, 1)) > 57 Then
    Do Until Asc(Mid(befehl, erstezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, erstezahl, 1)) <= 57
57 'Falls Leerzeichen oder Buchstaben zwischen Start und Zahl dann verschiebe Position um eins nach
rechts
        erstezahl = erstezahl + 1
    Loop
End If

'letztezahl ist am Ende von befehl positioniert
If Asc(Mid(befehl, letztezahl, 1)) < 48 Or Asc(Mid(befehl, letztezahl, 1)) > 57 Then
    Do Until Asc(Mid(befehl, letztezahl, 1)) >= 48 AndAlso Asc(Mid(befehl, letztezahl, 1)) <= 57
57 'Beginne am Ende und gehe nach links bis linkezahl eine Zahl wird
        letztezahl = letztezahl - 1
    Loop
End If
Winkel = CInt(Mid(befehl, erstezahl, letztezahl - erstezahl + 1))

'*****Ende des kreisbefehlsauslesen*****
*****

'*****Beginn Berechnung des Kreises*****

Dim APX, APY, MPX, MPY, radius As Integer

APX = Koordinaten(1, Befehlsnummer)
APY = Koordinaten(2, Befehlsnummer)

MPX = MittelpunktX
MPY = MittelpunktY
radius = Math.Sqrt((Math.Abs(APX - MPX)) ^ 2 + (Math.Abs(APY - MPY)) ^ 2)

Dim Anzlinienkreis, Teilwinkel, index As Integer

```

```

Anzlinienkreis = (radius / 100) * (Winkel / 150)

Teilwinkel = Winkel / (Anzlinienkreis * 10)

index = 0

Dim EPX, EPY, NPX, NPY, Anfangswinkel, NeuerWinkel As Integer

NPX = APX - MPX
NPY = APY - MPY
Anfangswinkel = Math.Acos(NPX / radius) * (180 / Math.PI)
If NPY < 0 Or NPX < 0 Then
    NeuerWinkel = -Anfangswinkel

Do
    NeuerWinkel = NeuerWinkel + Teilwinkel

    NPX = (Math.Cos(NeuerWinkel / 180 * Math.PI)) * radius + MPX
    NPY = (Math.Sin(NeuerWinkel / 180 * Math.PI)) * radius + MPY

    index = index + 1

    Befehlsnummer = Befehlsnummer + 1

    ReDim Preserve Koordinaten(2, 0 To Befehlsnummer + 1)

    Koordinaten(0, Befehlsnummer) = ObenUnten
    Koordinaten(1, Befehlsnummer) = NPX
    Koordinaten(2, Befehlsnummer) = NPY

Loop Until index = Anzlinienkreis - 1

EPX = (Math.Cos((Winkel / 10 - Anfangswinkel) / 180 * Math.PI)) * radius + MPX
EPY = (Math.Sin((Winkel / 10 - Anfangswinkel) / 180 * Math.PI)) * radius + MPY
ReDim Preserve Koordinaten(2, 0 To Befehlsnummer + 1)
Koordinaten(0, Befehlsnummer) = ObenUnten
Koordinaten(1, Befehlsnummer) = EPX
Koordinaten(2, Befehlsnummer) = EPY
ElseIf NPY > 0 Or NPX > 0 Then
    NeuerWinkel = Anfangswinkel

Do
    NeuerWinkel = NeuerWinkel + Teilwinkel

    NPX = (Math.Cos(NeuerWinkel / 180 * Math.PI)) * radius + MPX
    NPY = (Math.Sin(NeuerWinkel / 180 * Math.PI)) * radius + MPY

    index = index + 1

    ReDim Preserve Koordinaten(2, 0 To Befehlsnummer + 1)

    Befehlsnummer = Befehlsnummer + 1

    Koordinaten(0, Befehlsnummer) = ObenUnten
    Koordinaten(1, Befehlsnummer) = NPX
    Koordinaten(2, Befehlsnummer) = NPY

Loop Until index = Anzlinienkreis - 1

EPX = (Math.Cos((Winkel / 10 + Anfangswinkel) / 180 * Math.PI)) * radius + MPX
EPY = (Math.Sin((Winkel / 10 + Anfangswinkel) / 180 * Math.PI)) * radius + MPY
Befehlsnummer = Befehlsnummer + 1

```

```

        ReDim Preserve Koordinaten(2, 0 To Befehlsnummer + 1)
        Koordinaten(0, Befehlsnummer) = ObenUnten
        Koordinaten(1, Befehlsnummer) = EPX
        Koordinaten(2, Befehlsnummer) = EPY
    End If

```

```

End If
n = n + 1

```

```

Loop Until n = UBound(Zeile) + 1

```

```

'Hier beginnt die Linienausgabe*****

```

```

Dim stift1 As New Pen(Color.Red, 3)
Dim stift2 As New Pen(Color.Blue, 3)

```

```

Dim g As Graphics
g = Me.CreateGraphics
Dim ko1, ko2, ko3, ko4, stz, m, j As Integer
Dim stift As Object
j = 1

```

```

For m = 0 To Befehlsnummer

```

```

    If m <= Befehlsnummer Then
        stz = Koordinaten(0, m + 1)
        If stz = 0 Then
            stift = stift1
        Else
            stift = stift2
        End If
    End If

```

```

    End If
    ko1 = Koordinaten(1, m) / 4
    ko2 = ((Koordinaten(2, m) + 400) / 4)
    If m <= Befehlsnummer Then
        ko3 = Koordinaten(1, m + 1) / 4
        ko4 = (Koordinaten(2, m + 1) + 400) / 4
    End If

```

```

        g.DrawLine(stift, ko3, ko4, ko1, ko2)
    End If

```

```

Next m

```

```

End Sub

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1. Click

```

```

    Dim ULStat As MccDaq.ErrorInfo
    Dim DataValue As UInt16
    Dim Stiftobenoderunten As Integer

```

```

    If Befehlsnummer < 1 Then
        MsgBox("Zuerst auf Vorschau Klicken, sonst werden keine Linien eingelesen")
        GoTo Allesueberspringen
    End If

```

```

    Dim k, APX, APY, EPX, EPY, Anzlinien As Integer

```

```

    Dim Motorx(4) As Integer
    Dim Motory(4) As Integer

```

```

Motorx(1) = 3
Motorx(2) = 6
Motorx(3) = 12
Motorx(4) = 9
Motory(1) = 48
Motory(2) = 96
Motory(3) = 192
Motory(4) = 144

k = 0

Stiftobenoderunten = 0

Dim AnzzehntelSchritteX, AnzzehntelSchritteY, AnzahlMotorSchritteX, AnzahlMotorSchritteY, Schritte As Integer
Dim UeberlaufX, UeberlaufY, GesamtueberlaufX, GesamtueberlaufY As Double
Dim Verhaeltnisxy As Double
Dim index, z As Integer

Dim LinienueberlaufX, LinienueberlaufY As Double

DataValue = 0

GesamtueberlaufX = 0
k = 4
z = 4
Dim Zwischenspeicher As UInt16

For Anzlinien = 0 To Befehlsnummer

    If Koordinaten(0, Anzlinien + 1) <> Stiftobenoderunten Then
        Zwischenspeicher = DataValue

        If Stiftobenoderunten = 0 Then

            DataValue = 255

            ULStat = DaqBoard.DOut(PortNum2, DataValue)

            If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
                Stop
            End If
            Stiftobenoderunten = 1

        ElseIf Stiftobenoderunten = 1 Then
            DataValue = 0

            ULStat = DaqBoard.DOut(PortNum2, DataValue)

            If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
                Stop
            End If
            Stiftobenoderunten = 0
        End If
        DataValue = Zwischenspeicher
    End If

    APX = Koordinaten(1, Anzlinien)
    EPX = Koordinaten(1, Anzlinien + 1)

    If APX > EPX Then
        AnzzehntelSchritteX = APX - EPX
    ElseIf EPX > APX Then
        AnzzehntelSchritteX = EPX - APX
    End If

```

```

End If
AnzzehntelSchritteX = AnzzehntelSchritteX / 2

AnzahlMotorSchritteX = Int(AnzzehntelSchritteX / 0.113636)
UeberlaufX = (AnzzehntelSchritteX / 0.113636) - AnzahlMotorSchritteX
Gesamtueberlaufx = Gesamtueberlaufx + UeberlaufX

APY = Koordinaten(2, Anzlinien)
EPY = Koordinaten(2, Anzlinien + 1)
If APY > EPY Then
    AnzzehntelSchritteY = APY - EPY
ElseIf EPY > APY Then
    AnzzehntelSchritteY = EPY - APY
End If

AnzzehntelSchritteY = AnzzehntelSchritteY / 2
AnzahlMotorSchritteY = Int(AnzzehntelSchritteY)
Ueberlaufy = (AnzzehntelSchritteY) - AnzahlMotorSchritteY
Gesamtueberlaufy = Gesamtueberlaufy + Ueberlaufy

If AnzahlMotorSchritteX = 0 AndAlso AnzahlMotorSchritteY = 0 Then
    GoTo EndederMotorbewegung
End If
If AnzahlMotorSchritteY <> 0 AndAlso AnzahlMotorSchritteX > AnzahlMotorSchritteY Then
    Verhaeltnisxy = (AnzahlMotorSchritteX / AnzahlMotorSchritteY)
ElseIf AnzahlMotorSchritteX <> 0 AndAlso AnzahlMotorSchritteX < AnzahlMotorSchritteY Then
    Verhaeltnisxy = (AnzahlMotorSchritteY / AnzahlMotorSchritteX)
ElseIf AnzahlMotorSchritteX = 0 AndAlso AnzahlMotorSchritteY > 0 Then
    Verhaeltnisxy = 1

ElseIf AnzahlMotorSchritteY = 0 AndAlso AnzahlMotorSchritteX > 0 Then
    Verhaeltnisxy = 1

End If

If AnzahlMotorSchritteX < AnzahlMotorSchritteY Then
    Schritte = AnzahlMotorSchritteX

    Linienueberlaufy = Gesamtueberlaufy
    Gesamtueberlaufy = 0
    If Schritte = 0 Then
        Schritte = 1
        Linienueberlaufy = Gesamtueberlaufy + AnzahlMotorSchritteY
    End If
ElseIf AnzahlMotorSchritteX > AnzahlMotorSchritteY Then
    Schritte = AnzahlMotorSchritteY
    Linienueberlaufx = Gesamtueberlaufx
    Gesamtueberlaufx = 0
    If Schritte = 0 Then
        Schritte = 1
        Linienueberlaufx = Gesamtueberlaufx + AnzahlMotorSchritteX

    End If
End If

For index = 1 To Schritte
    If EPX > APX AndAlso EPX <> APX Then
        k = k + 1
    End If
    If EPY > APY AndAlso EPY <> APY Then
        z = z + 1
    End If
    If EPX < APX AndAlso EPX <> APX Then
        k = k - 1
    End If
    If EPY < APY AndAlso EPY <> APY Then
        z = z - 1
    End If

```

```

If z = 5 Then
    z = 1
End If
If z = 0 Then
    z = 4
End If
If k = 5 Then
    k = 1
End If
If k = 0 Then
    k = 4
End If
DataValue = Motorx(k) + Motory(z)

ULStat = DaqBoard.DOut(PortNum, DataValue)

If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
    Stop
End If
Sleep(4)

If AnzahlMotorSchritteX > AnzahlMotorSchritteY Then
    Linienueberlaufx = Linienueberlaufx + (Verhaeltnisxy - 1)
End If
If AnzahlMotorSchritteX < AnzahlMotorSchritteY Then
    Linienueberlaufy = Linienueberlaufy + (Verhaeltnisxy - 1)
End If
If Linienueberlaufx > 1 Then

    Do Until Linienueberlaufx < 1

        If EPX > APX Then
            k = k + 1
        End If
        If EPX < APX Then
            k = k - 1
        End If
        If k = 5 Then
            k = 1
        End If
        If k = 0 Then
            k = 4
        End If
        DataValue = Motorx(k)

        Linienueberlaufx = Linienueberlaufx - 1
        ULStat = DaqBoard.DOut(PortNum, DataValue)

        If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
            Stop
        End If
        Sleep(4)
    Loop

End If

If Linienueberlaufy > 1 Then
    Do Until Linienueberlaufy < 1

        If EPY > APY Then
            z = z + 1
        End If
        If EPY < APY Then
            z = z - 1
        End If
        If z = 5 Then
            z = 1

```

```

        End If
        If z = 0 Then
            z = 4
        End If
        DataValue = Motory(z)

        Linienueberlaufy = Linienueberlaufy - 1
        ULStat = DaqBoard.DOut(PortNum, DataValue)

        If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
            Stop
        End If
        Sleep(4)

    Loop
End If

Next index
If AnzahlMotorSchritteX > AnzahlMotorSchritteY Then
    Gesamtueberlaufx = Linienueberlaufx
    Linienueberlaufx = 0
    Gesamtueberlaufy = Gesamtueberlaufy
ElseIf AnzahlMotorSchritteX < AnzahlMotorSchritteY Then
    Gesamtueberlaufy = Linienueberlaufy
    Linienueberlaufy = 0
    Gesamtueberlaufx = Gesamtueberlaufx
End If

```

EndederMotorbewegung:

Next Anzlinien

```

' write the value to FirstPortA
' Parameters:
'   PortNum    :the output port
'   DataValue  :the value written to the port

```

Allesueberspringen:

End Sub

```

Private Sub frmSetDigOut_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.
Load

```

End Sub

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.
Click

```

```

    Dim ULStat As MccDaq.ErrorInfo
    Dim DataValue As UInt16

```

```

    Dim Motorx(4) As Integer
    Dim Motory(4) As Integer
    Motorx(1) = 3
    Motorx(2) = 6
    Motorx(3) = 12
    Motorx(4) = 9
    Motory(1) = 48
    Motory(2) = 96

```

```

    Motory(3) = 192
    Motory(4) = 144
    Dim k As Integer
    k = 1

    Dim i As Integer

    For i = 0 To 250

        k = k - 1

        If k = 5 Then
            k = 1
        End If
        If k = 0 Then
            k = 4
        End If
        DataValue = Motory(k)

        ULStat = DaqBoard.DOut(PortNum, DataValue)

        If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
            Stop
        End If
        Sleep(4)
    Next i

End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.
Click
    Dim ULStat As MccDaq.ErrorInfo
    Dim DataValue As UInt16

    Dim Motorx(4) As Integer
    Dim Motory(4) As Integer
    Motorx(1) = 3
    Motorx(2) = 6
    Motorx(3) = 12
    Motorx(4) = 9
    Motory(1) = 48
    Motory(2) = 96
    Motory(3) = 192
    Motory(4) = 144
    Dim k As Integer
    k = 1

    Dim i As Integer

    For i = 0 To 200

        k = k + 1

        If k = 5 Then
            k = 1
        End If
        If k = 0 Then
            k = 4
        End If
        DataValue = Motory(k)

        ULStat = DaqBoard.DOut(PortNum, DataValue)

        If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
            Stop
        End If
        Sleep(4)
    Next i

End Sub

Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.

```



Click

```
Dim ULStat As MccDaq.ErrorInfo
Dim DataValue As UInt16
```

```
Dim Motorx(4) As Integer
Dim Motory(4) As Integer
Motorx(1) = 3
Motorx(2) = 6
Motorx(3) = 12
Motorx(4) = 9
Motory(1) = 48
Motory(2) = 96
Motory(3) = 192
Motory(4) = 144
Dim k As Integer
k = 1
```

```
Dim i As Integer
```

```
For i = 0 To 2000
```

```
    k = k + 1
```

```
    If k = 5 Then
```

```
        k = 1
```

```
    End If
```

```
    If k = 0 Then
```

```
        k = 4
```

```
    End If
```

```
    DataValue = Motorx(k)
```

```
    ULStat = DaqBoard.DOut(PortNum, DataValue)
```

```
    If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
```

```
        Stop
```

```
    End If
```

```
    Sleep(2)
```

```
Next i
```

End Sub

Private Sub Button6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6. 

Click

```
Dim ULStat As MccDaq.ErrorInfo
Dim DataValue As UInt16
```

```
Dim Motorx(4) As Integer
Dim Motory(4) As Integer
Motorx(1) = 3
Motorx(2) = 6
Motorx(3) = 12
Motorx(4) = 9
Motory(1) = 48
Motory(2) = 96
Motory(3) = 192
Motory(4) = 144
Dim k As Integer
k = 1
```

```
Dim i As Integer
```

```
For i = 0 To 2000
```

```
    k = k - 1
```

```
    If k = 5 Then
```

```
        k = 1
```

```
    End If
```

```
    If k = 0 Then
```

```
        k = 4
```

```
    End If
```

```
    DataValue = Motorx(k)
```

```
        ULStat = DaqBoard.DOut(PortNum, DataValue)

        If ULStat.Value <> MccDaq.ErrorInfo.ErrorCode.NoErrors Then
            Stop
        End If
        Sleep(2)
    Next i

End Sub

End Class
```