```
'File:  frmJS2
'Date:  5/18/02
'Auth:  CWK (adapted from sample Microsoft program)
'Desc:  Joystick control program

Option Explicit
Implements DirectXEvent8


Dim dx As New DirectX8
Dim di As DirectInput8
Dim diDev As DirectInputDevice8
Dim diDevEnum As DirectInputEnumDevices8
Dim EventHandle As Long
Dim joyCaps As DIDEVCAPS
Dim js As DIJOYSTATE
Dim DiProp_Dead As DIPROPLONG
Dim DiProp_Range As DIPROPRANGE
Dim DiProp_Saturation As DIPROPLONG
Dim AxisPresent(1 To 8) As Boolean
Dim running As Boolean

Dim outputString As String


Sub InitDirectInput()

   Set di = dx.DirectInputCreate()
   Set diDevEnum = di.GetDIDevices(DI8DEVCLASS_GAMECTRL,
DIEDFL_ATTACHEDONLY)
   If diDevEnum.GetCount = 0 Then
     MsgBox "No joystick attached."
     Unload Me
   End If

   'Add attached joysticks to the listbox
   Dim i As Integer
   For i = 1 To diDevEnum.GetCount
      Call lstJoySticks.AddItem(diDevEnum.GetItem(i).GetInstanceName)
   Next

   ' Get an event handle to associate with the device
   EventHandle = dx.CreateEvent(Me)
   Exit Sub

Error_Out:
```

```
    MsgBox "Error initializing DirectInput."
    Unload Me

End Sub


Private Sub cmdEnable_Click()
Call lstJoySticks_Click

End Sub

Private Sub DirectXEvent8_DXCallback(ByVal eventid As Long)

' This is called whenever there's a change in the joystick state.
' We check the new state and update the display.


    Dim i As Integer
    Dim ListPos As Integer
    Dim S As String

    If diDev Is Nothing Then Exit Sub

    " Get the device info
    On Local Error Resume Next
    diDev.GetDeviceStateJoystick js
    If Err.Number = DIERR_NOTACQUIRED Or Err.Number = DIERR_INPUTLOST
Then
        diDev.Acquire
        Exit Sub
    End If


    On Error GoTo err_out

    outputString = "x"

    Select Case js.x
        Case 0
            outputString = outputString & "0"
        Case 5000
            outputString = outputString & "1"
        Case 10000
            outputString = outputString & "2"
    End Select
```

```
outputString = outputString & "y"

Select Case js.y
   Case 0
      outputString = outputString & "0"
   Case 5000
      outputString = outputString & "1"
   Case 10000
      outputString = outputString & "2"
End Select

outputString = outputString & "a"

Select Case js.Buttons(0)
   Case 0
      outputString = outputString & "0"
   Case Else
      outputString = outputString & "1"
End Select

outputString = outputString & "d"

Select Case js.Buttons(3)
   Case 0
      outputString = outputString & "0"
   Case Else
      outputString = outputString & "1"
End Select


Call frmMain1.Send_Data(outputString)

' Display axis coordinates
ListPos = 0
For i = 1 To 8
   If AxisPresent(i) Then
      Select Case i
         Case 1
            S = "X: " & js.x
         Case 2
            S = "Y: " & js.y
         Case 3
            S = "Z: " & js.z
         Case 4
            S = "RX: " & js.rx
         Case 5
```

```
                S = "RY: " & js.ry
            Case 6
                S = "RZ: " & js.rz
            Case 7
                S = "Slider0: " & js.slider(0)
            Case 8
                S = "Slider1: " & js.slider(1)

        End Select
        lstJoyAxis.List(ListPos) = S
        ListPos = ListPos + 1

      End If
    Next

  ' Buttons

  For i = 0 To joyCaps.lButtons - 1
     Select Case js.Buttons(i)
     Case 0
        lstButton.List(i) = "Button " + CStr(i + 1) + ": Up"

     Case Else
        lstButton.List(i) = "Button " + CStr(i + 1) + ": Down"

     End Select
  Next

   ' Hats
  For i = 0 To joyCaps.lPOVs - 1
     lstHat.List(i) = "POV " + CStr(i + 1) + ": " + CStr(js.POV(i))
  Next


  Exit Sub

err_out:
   MsgBox Err.Description & " : " & Err.Number, vbApplicationModal
  End

End Sub


Private Sub Form_Load()
  running = True
  InitDirectInput
```

```
End Sub


Private Sub Form_Unload(cancel As Integer)
  On Local Error Resume Next
  If EventHandle <> 0 Then dx.DestroyEvent EventHandle

  running = False

  'Unacquire if we are holding a device
  If Not diDev Is Nothing Then
    diDev.Unacquire
  End If

  DoEvents
  End
End Sub



Private Sub lstJoySticks_Click()

   On Local Error Resume Next

   Call CLRLISTS


   'Unacquire the current device
   'if we are holding a device
   If Not diDev Is Nothing Then
     diDev.Unacquire
   End If

   'Create the joystick device
   Set diDev = Nothing
   Set diDev = di.CreateDevice(diDevEnum.GetItem(lstJoySticks.ListIndex +
1).GetGuidInstance)
   diDev.SetCommonDataFormat DIFORMAT_JOYSTICK
   diDev.SetCooperativeLevel Me.hWnd, DISCL_BACKGROUND Or
DISCL_NONEXCLUSIVE

   ' Find out what device objects it has
   diDev.GetCapabilities joyCaps
   Call IdentifyAxes(diDev)

   ' Ask for notification of events
```

```
    Call diDev.SetEventNotification(EventHandle)

    ' Set deadzone for X and Y axis to 10 percent of the range of travel
    With DiProp_Dead
       .lData = 1000
       .lHow = DIPH_BYOFFSET

       .lObj = DIJOFS_X
       diDev.SetProperty "DIPROP_DEADZONE", DiProp_Dead

       .lObj = DIJOFS_Y
       diDev.SetProperty "DIPROP_DEADZONE", DiProp_Dead

    End With

    ' Set saturation zones for X and Y axis to 5 percent of the range
    With DiProp_Saturation
       .lData = 9500
       .lHow = DIPH_BYOFFSET

       .lObj = DIJOFS_X
        diDev.SetProperty "DIPROP_SATURATION", DiProp_Saturation

       .lObj = DIJOFS_Y
        diDev.SetProperty "DIPROP_SATURATION", DiProp_Saturation

    End With

    SetPropRange


    diDev.Acquire

    ' Get the list of current properties
    ' USB joysticks wont call this callback until you play with the joystick
    ' so we call the callback ourselves the first time
    DirectXEvent8_DXCallback 0

    ' Poll the device so that events are sure to be signaled.
    ' Usually this would be done in Sub Main or in the game rendering loop.

    While running = True
       DoEvents
       diDev.Poll
    Wend
End Sub
```

```
Sub SetPropRange()
    ' NOTE Some devices do not let you set the range
    On Local Error Resume Next

    ' Set range for all axes
    With DiProp_Range
        .lHow = DIPH_DEVICE
        .lMin = 0
        .lMax = 10000
    End With
    diDev.SetProperty "DIPROP_RANGE", DiProp_Range
End Sub

Sub CLRLISTS()
    lstJoyAxis.Clear
    lstButton.Clear
    lstHat.Clear
End Sub

Sub IdentifyAxes(diDev As DirectInputDevice8)

   ' It's not enough to count axes; we need to know which in particular
   ' are present.

   Dim didoEnum As DirectInputEnumDeviceObjects
   Dim dido As DirectInputDeviceObjectInstance
   Dim i As Integer

   For i = 1 To 8
     AxisPresent(i) = False
   Next

   ' Enumerate the axes
   Set didoEnum = diDev.GetDeviceObjectsEnum(DIDFT_AXIS)

   ' Check data offset of each axis to learn what it is
   Dim sGuid As String
   For i = 1 To didoEnum.GetCount

     Set dido = didoEnum.GetItem(i)

        sGuid = dido.GetGuidType
        Select Case sGuid
           Case "GUID_XAxis"
             AxisPresent(1) = True
```

```
            Case "GUID_YAxis"
              AxisPresent(2) = True
            Case "GUID_ZAxis"
              AxisPresent(3) = True
            Case "GUID_RxAxis"
              AxisPresent(4) = True
            Case "GUID_RyAxis"
              AxisPresent(5) = True
            Case "GUID_RzAxis"
              AxisPresent(6) = True
            Case "GUID_Slider"
                AxisPresent(8) = True
                AxisPresent(7) = True
        End Select

    Next
End Sub
```