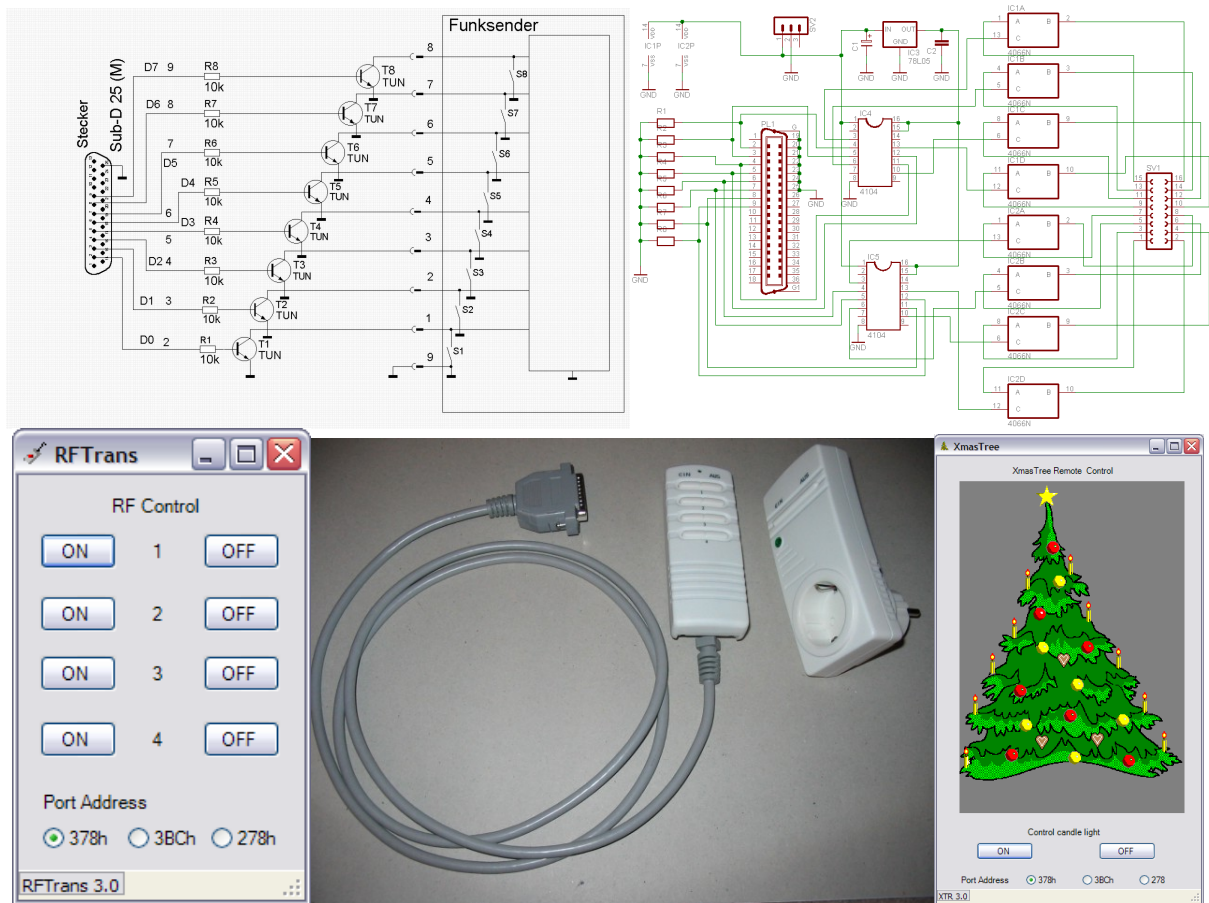


Parallel-Interface für Funkschaltssysteme



Autor:

Dipl. Ing. (FH) Karl Köckeis

Parallel-Interface für Funk- schaltssysteme

Parallel-Interface für Funkschaltssysteme

Funkschaltssysteme erfreuen sich immer größerer Beliebtheit zur Fernsteuerung aller erdenklichen Anwendungen im Haushalt. Funkschaltersets sind heute in jedem Baumarkt günstig zu erwerben. Was liegt daher näher, als solche Systeme für die Steuerung und Automatisierung über den PC einzusetzen. Alles was fehlt ist ein geeignetes Interface, das die Verbindung vom PC zum Funkschaltssystem herstellt. Das hier vorgestellte Interface erfüllt diesen Zweck und ist dabei so einfach gehalten, dass es mit sehr geringem Aufwand und kostengünstig verwirklicht werden kann.



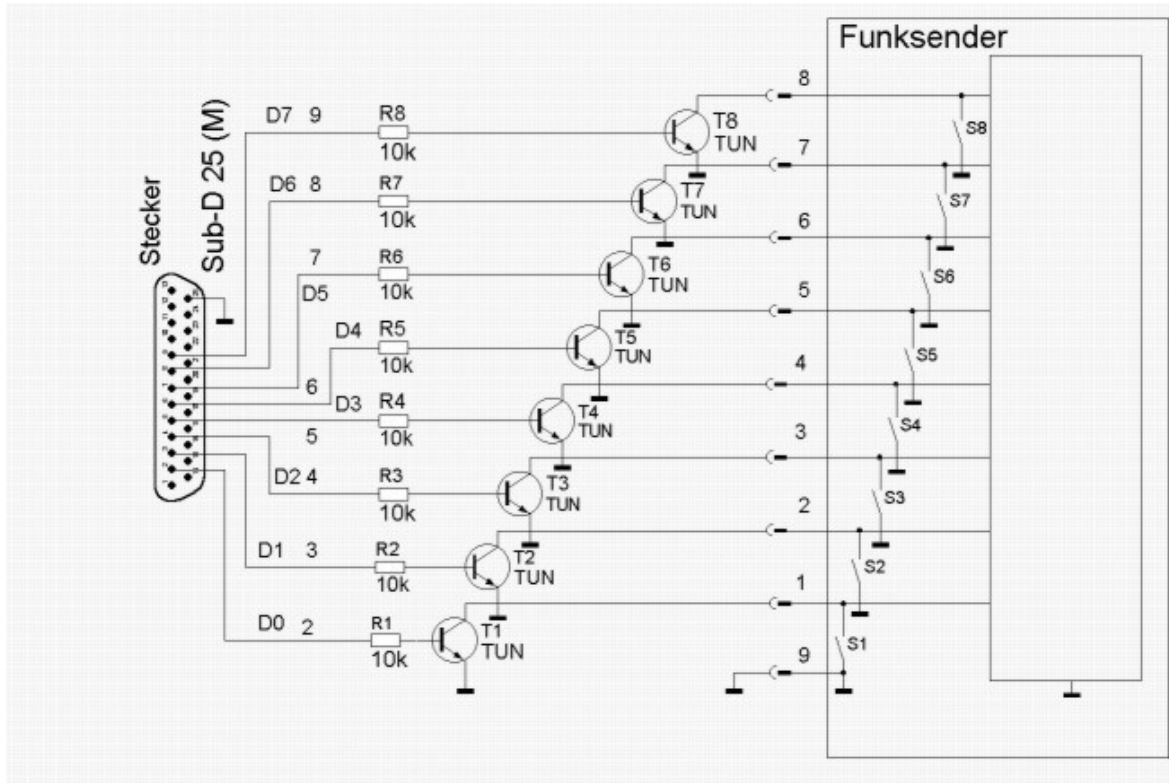
Funktion

Die Parallelschnittstelle bietet 8 Datenbits (D0...D7), die zur Ansteuerung des Interface genutzt werden. Dies sind die Pins 2...9 an der 25-poligen Sub-D Buchse. Ein Transistor im Interface bildet einen elektronischen Schalter, der mit dem

Funksender verbunden ist. Eine Software erzeugt kurze Schaltimpulse, die über die Schaltstufe im Interface so einen Funkbefehl im Funkschaltssystem auslösen und die gewünschte Steuerfunktion ausführen.

Schaltung

Die Schaltung ist denkbar einfach aufgebaut:



Schaltimpulse mit einer Länge von 0,5s werden von den Datenbits D0...D7 der Parallelschnittstelle auf 8 Schaltstufen bestehend aus R1...R8 und T1...T8 gegeben. Diese arbeiten im Open Collector Betrieb und sind über einen Steckverbinder mit den Tasteneingängen des Codierchips im Funksender verbunden. Die Schalttransistoren liegen damit parallel zu den Tasten des Senders. Die von der Ansteuersoftware erzeugten Schaltimpulse lösen so einen Funkbefehl aus, genau wie es auch bei einem Tastendruck am Sender der Fall ist.

Stückliste

Widerstände:

R1...R8 = 10kΩ

Halbleiter:

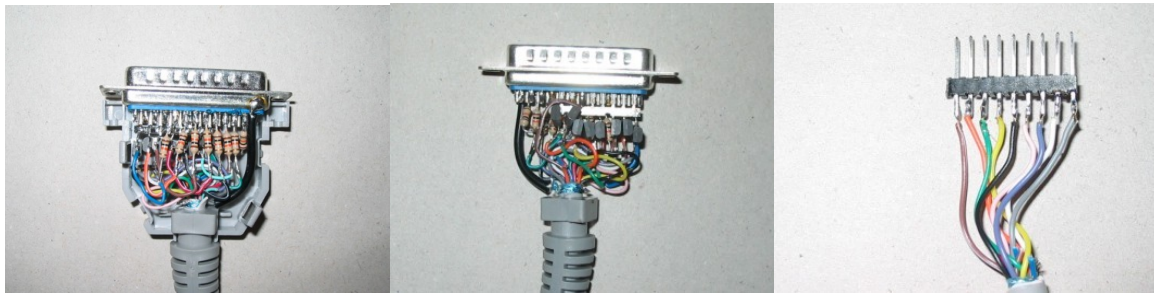
T1...T8 = TUN (kleine Bauform)

Sonstiges:

Centronics-Druckerkabel
9-pol. Steckverbinder

Kabel

Als Kabel kann ein handelsübliches Drucker­kabel verwendet werden, bei dem die Stecker nicht angegossen sind. Durch Auswahl von Bauteilen mit kleiner Bauform kann die gesamte Elektronik im Sub-D Stecker des Kabels untergebracht werden. Hierzu wird das Steckergehäuse geöffnet, das Kabel ausgelötet, die nicht benötigten Leitungen gekürzt um Platz zu schaffen und dann die Widerstände R1...R8 sowie die Transistoren T1...T8 frei verdrahtet. Der Stecker am anderen Ende des Kabels wird ausgelötet oder abgetrennt und der 9-polige Stecker für die Verbindung zum Sender angelötet. Nach Überprüfung der Funktion kann das Steckergehäuse wieder geschlossen werden.



Steckverbindung zum Handsender

Die 9-polige Buchsenleiste wird in den Handsender eingebaut. Das vorliegende Model (Fa. Well-Light) bietet hierzu neben der Batterie ausreichend Platz. Bei nicht gestecktem Interface kann somit der Handsender uneingeschränkt wie üblich weiterverwendet werden.

Die Buchsenleiste wird wie im Schaltbild dargestellt verdrahtet. Der Handsender wird geöffnet, eine Leitung wird an Masse und 8 Leitungen an die Schalter S1...S8 gelegt. Die Kabel werden an geeigneten Stellen, die man sich vorher ausgesucht hat, an Bauteile auf der Platine gelötet. Kabel und Buchsenleiste werden verstaut und das Gehäuse wieder geschlossen. Der Steckverbinder kann dann über das geöffnete Batteriefach erreicht werden. Fertig ist die Hardware.



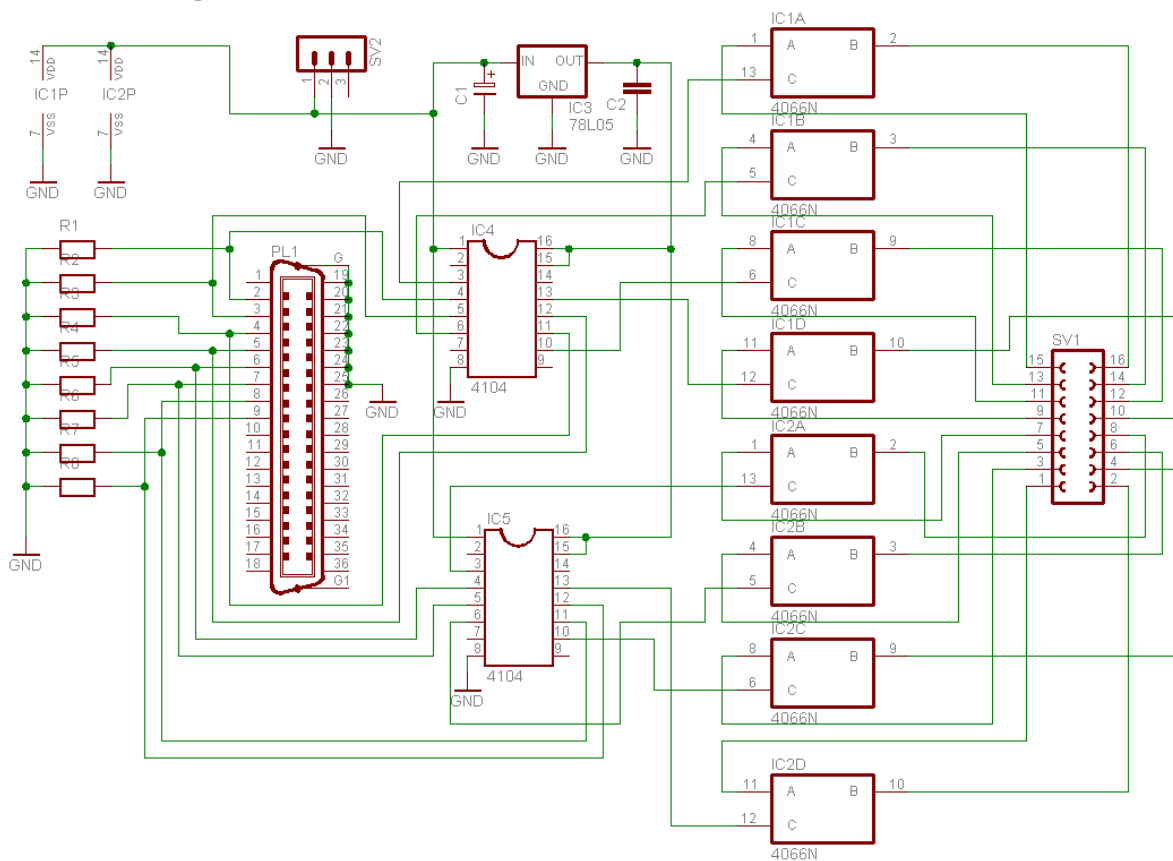
Parallel-Interface mit MOS-Schalter

Das hier vorgestellte Variante des Interface arbeitet mit Pegelanpassung und MOS-Schaltern. Das Interface kann somit auch an eine Matrix angeschlossen werden.

Funktion

Die Parallelschnittstelle bietet 8 Datenbits (D0...D7), die zur Ansteuerung des Interface genutzt werden. Dies sind die Pins 2...9 an der 25-poligen Sud-D Buchse. Ein MOS-Schalter im Interface bildet einen elektronischen Schalter, der mit dem Funksender verbunden ist. Eine Software erzeugt kurze Schaltimpulse, die über die Schaltstufe im Interface so einen Funkbefehl im Funkschaltsystem auslösen und die gewünschte Steuerfunktion ausführen.

Schaltung



Schaltimpulse mit einer Länge von 0,5s werden von den Datenbits D0...D7 der Parallelschnittstelle werden über eine Pegelanpassung (4104) auf MOS-Schalter (4066) gegeben. Ein Spannungsregler erzeugt aus der Versorgungsspannung (8...15V) die für TTL-Einganspegel erforderliche Spannung Vddi von 5V für die Pegelwandler IC4 und IC5. Damit bei offenem Eingang definierte zustände herrschen,

legen die Widerstände R1...R8 die Eingänge auf LOW. Die Ausgänge der Pegelwandler steuern die MOS-Schalter mit dem erforderlichen Pegel an.

Stückliste

Widerstände:

R1...R8 = 47k

Kondensatoren:

C1 = 10 μ F/16V

C2 = 100nF

Halbleiter:

IC1, IC2 = 4066

IC3 = 78L05

IC4, IC5 = 4104

Sonstiges:

PL1 = Centronics-Buchse

SV1 = 16-pol. Steckverbinder

SV2 = 3-pol. Steckverbinder

Software

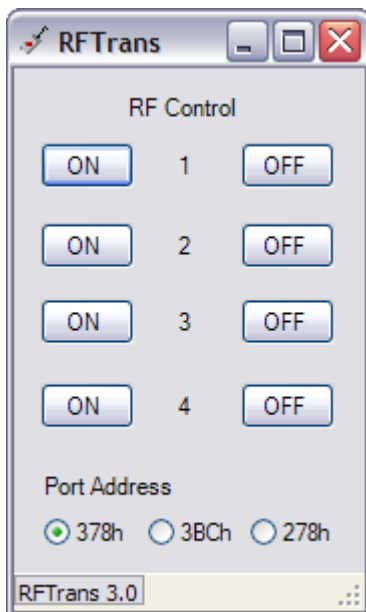
Die Software wurde in VisualBasic.NET geschrieben.

Das Programm RFTrans bildet den Handsender auf dem PC nach Die so erzeugten Schaltimpulse dienen zur Ansteuerung des Interface. Die Portadresse kann über RadioButtons eingestellt werden.

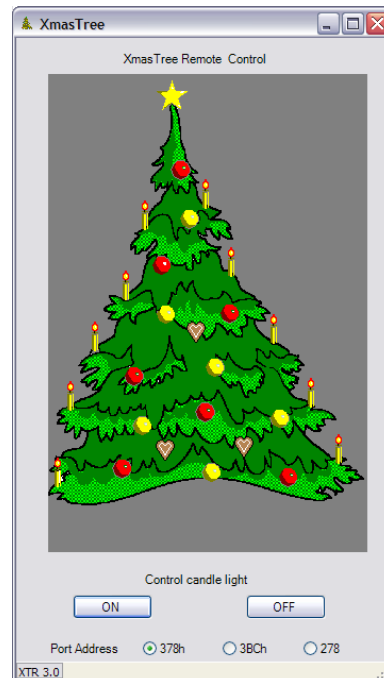
Das Programm CChannel arbeitet ohne Ausgabe auf den Bildschirm. Sie können verwendet werden, wenn die Schaltfunktion von einem anderen Programm aus aufgerufen werden. Beim Aufruf werden Portadresse (dezimal) und Kanal (1-8) in der Kommandozeile angegeben. Z.B.: CChannel &H378,5

Speziell zum Steuern eines Weihnachtsbaums wurde das Programm XmasTree RemoteControl entwickelt. Es ist mit einer ansprechenden grafischen Schaltzustandsanzeige versehen und schaltet Funksteckdose 1.

Screenshots:



RFTrans



XmasTree RemoteControl

Windows XP

Unter dem Betriebssystem Windows XP ist die Ansteuerung der Schnittstelle nicht so einfach ist. Der `out`-Befehl wird hier nicht mehr unterstützt.

Das Problem kann durch eine externe DLL-Datei (`inpout32.dll`) umgangen werden. Hierzu muß die Verwendung der DLL-Datei durch einfügen der Zeile

```
DECLARE SUB out LIB "inpout32.DLL" ALIAS "Out32" (ByVal  
PortAddress AS Integer, ByVal Value AS Integer)
```

am Anfang des Programms deklariert werden. So wird dem Compiler mitgeteilt, den `out`-Befehl durch den Aufruf der Funktion `Out32` aus der externen DLL-Datei zu ersetzen. Die `inpout32.dll` unterstützt die Versionen 98/ME/NT/2000/XP. Damit sollten die Programme unter all diesen Windows-Versionen laufen.

Schalten per SMS

Das Interface kann auch sehr gut mit der SMS-Kommunikations Software aus Elektor 10/2003 kombiniert werden. In Verbindung mit dem Programmen `CChannel` erlaubt es auf einfache Weise die Ansteuerung des Funkschaltsystems per SMS vom Handy aus. Beim Aufruf des Programms wird die als Parameter angegebene Funktion, der auf dem Handsender entsprechenden Taste aktiviert.

Übersicht der Programmfunktionen:

Programm RTrans:

Schalten von Funksteckdose 1
Datenbit 4 EIN
Datenbit 0 AUS

Schalten von Funksteckdose 2
Datenbit 5 EIN
Datenbit 1 AUS

Schalten von Funksteckdose 3
Datenbit 6 EIN
Datenbit 2 AUS

Schalten von Funksteckdose 4
Datenbit 7 EIN
Datenbit 3 AUS

Programm Cchannel:

Schalten von Funksteckdose n über Port Adr (hex)

Aufruf: Cchannel Adr,n z.B.: Cchannel &H378,5

Adr:

LPT1	888 dez	378hex
LPT1	952 dez	3BChex
LPT2	632 dez	278hex

n	0	alle Datenbits	0
n	1	Datenbit 4	Kanal 1 AUS
n	2	Datenbit 5	Kanal 2 AUS
n	3	Datenbit 6	Kanal 3 AUS
n	4	Datenbit 7	Kanal 4 AUS
n	5	Datenbit 0	Kanal 1 EIN
n	6	Datenbit 1	Kanal 2 EIN
n	7	Datenbit 2	Kanal 3 EIN
n	8	Datenbit 3	Kanal 4 EIN

Programm XmasTree RemoteControl:

Schalten von Funksteckdose 1
Datenbit 4 EIN
Datenbit 0 AUS

Quellen

Input32.dll: <http://www.logix4u.net/>

Infos zu Steuerung über PC-Schnittstellen: <http://www.franksteinberg.de>

Grafik: <http://www.bingo-ev.de>

Anhang

Visual Basic.NET Sourcecodes

CChannel

Module Module1

```
Declare Sub out Lib "inout32.DLL" Alias "Out32" (ByVal PortAddress As Integer,  
ByVal Value As Integer)
```

```
Declare Sub Sleep Lib "kernel32" (ByVal milliseconds As Long)
```

```
Dim Wert As Integer
```

```
Dim Channel As Integer
```

```
Dim Address As String
```

```
Dim Chan As String
```

```
Sub Main()
```

```
    Address = Left(Command$, 5)
```

```
    Chan = Right(Command$, 1)
```

```
    Channel = Int(Val(Chan))
```

```
    Select Case (Channel)
```

```
        Case 0 : Wert = 0
```

```
        Case 1 : Wert = 1
```

```
        Case 2 : Wert = 2
```

```
        Case 3 : Wert = 4
```

```
        Case 4 : Wert = 8
```

```
        Case 5 : Wert = 16
```

```
        Case 6 : Wert = 32
```

```
        Case 7 : Wert = 64
```

```
        Case 8 : Wert = 128
```

```
    End Select
```

```
    out(Address, Wert)
```

```
    Sleep(500)
```

```
    out(Address, 0)
```

```
End Sub
```

```
End Module
```

RFTrans

Public Class Form1

Dim Address As String

Declare Sub out Lib "inpout32.DLL" Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

Declare Sub Sleep Lib "kernel32" (ByVal milliseconds As Long)

Private Sub Output(ByVal Port As String, ByVal Value As Integer)

My.Computer.Audio.Play("clack.wav", 1)

out(Port, Value)

Sleep(500)

out(Port, 0)

End Sub

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

Output(Address, 1)

End Sub

Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged

Address = &H3BC

out(Address, 0)

End Sub

Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged

Address = &H278

out(Address, 0)

End Sub

Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged

Address = &H378

out(Address, 0)

End Sub

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

Output(Address, 16)

End Sub

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

Output(Address, 32)

End Sub

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

Output(Address, 64)

End Sub

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Output(Address, 128)
End Sub
```

```
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    Output(Address, 2)
End Sub
```

```
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
    Output(Address, 4)
End Sub
```

```
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
    Output(Address, 8)
End Sub
```

End Class

XmasTree

Public Class XmasTree

```
    Dim Address As String
    Declare Sub out Lib "inpout32.DLL" Alias "Out32" (ByVal PortAddress As Integer,
ByVal Value As Integer)
    Declare Sub Sleep Lib "kernel32" (ByVal milliseconds As Long)
    Dim Treeon As New Bitmap("tree_on.bmp")
    Dim Treeoff As New Bitmap("tree_off.bmp")
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        My.Computer.Audio.Play("clack.wav", 1)
        out(Address, 16)
        Sleep(500)
        out(Address, 0)
        PictureBox1.Image = Treeon
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        My.Computer.Audio.Play("clack.wav", 1)
        out(Address, 1)
        Sleep(500)
        out(Address, 0)
        PictureBox1.Image = Treeoff
    End Sub
```

```
    Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
        Address = &H378
        out(Address, 0)
    End Sub
```

```
    Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton3.CheckedChanged
        Address = &H278
        out(Address, 0)
    End Sub
```

```
    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
        Address = &H3BC
        out(Address, 0)
    End Sub
```

End Class