

# Infrarot-Transceiver für den PC

Geräte-Fernbedienung und Datenlink in einem

Von Burkhard Kainka

Dieser Beitrag beschreibt nicht nur ein IR-Transceiver-Projekt, mit dem Sie vom PC aus fernsteuern und Daten übertragen können. Er baut auf der erfolgreichen Elektor-Serie „Elektronik am PC“ auf und informiert Sie auch über die Grundlagen der verwendeten Infrarotübertragung und Schnittstellentechnik, die Sie dann für eigene Entwicklungen einsetzen können.



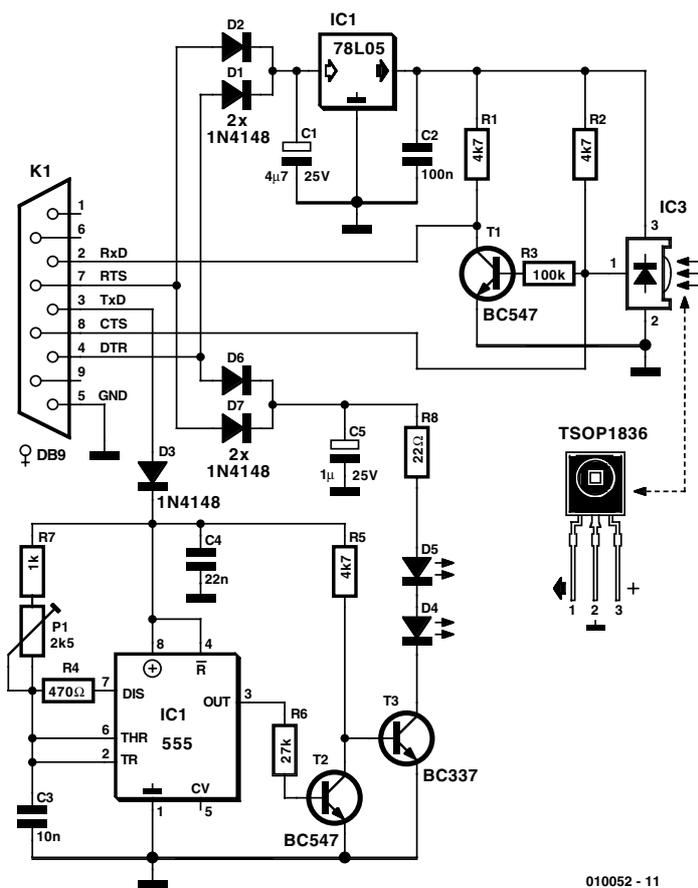
Datenübertragung mit Infrarot wird immer wichtiger. Nicht nur Fernbedienungen arbeiten mit unsichtbarem Licht, sondern auch PC-Mäuse, Tastaturen, Drucker und andere Zusatzgeräte. Auch in Glasfaserkabeln ver-

wendet man Infrarotlicht. Der hier vorgestellte „Infrarot-Transceiver am PC“ stellt als praxisbezogene Anwendung von PC-Schnittstellen und Visual-BASIC in gewissem

Sinne eine Fortsetzung der Serie „Basiskurs Elektronik am PC“ (Elektor 9/99 bis 3/00) dar, zu der inzwischen auch das neue Elektor-Buch *„Elektronik am PC - Visual Basic in der Praxis“* erschienen ist.

Infrarot-Fernbedienungen für Fernsehgeräte, Videorecorder und andere Geräte der Unterhaltungselektronik arbeiten oft nach dem von Philips definierten RC5-Standard. Ob eine bestimmte Fernbedienung RC5 verwendet, kann mit den im Folgenden vorgestellten Programmen leicht festgestellt werden. Man verwendet bei diesem Standard modulierte Lichtsignale im Bereich von 30 kHz bis etwa 40 kHz. Die Fernbedienung sendet so genannte Bursts aus, das sind einzelne Impulspakete, die hier eine Länge von 0,888 ms oder 1,776 ms haben. Bei einer Modulationsfrequenz von 36 kHz enthält ein kurzer Burst 32 Einzelimpulse, ein langer 64. Das gesamte Datenpaket dauert etwa 25 ms und wird alle 100 ms wiederholt, solange eine Taste gedrückt wird.

Eine Infrarot-Fernbedienung kann man leicht für andere Zwecke mit



010052 - 11

Bild 1. Die Schaltung des IR-Transceivers zum Anschluss an die serielle Schnittstelle eines PCs.

verwenden. Es ist zum Beispiel möglich, bestimmte Funktionen eines eigenen Programms zu steuern. Eine typische Anwendung wäre die Steuerung einer PC-Diashow. Nachdem das Programm gestartet ist, lehnt man sich zurück und bedient alles bequem vom Sitzplatz aus.

## Transceiver-Hardware

Der Empfang üblicher Infrarot-Signale ist dank integrierter Empfänger relativ einfach. Der bekannte IR-Empfänger SFH506 von Siemens wird für feste Modulationsfrequenzen von 30kHz, 33 kHz, 36 kHz und so weiter geliefert. Die Filterkurve ist relativ breitbandig, so dass Abweichungen von einigen Kilohertz nur zu einer relativ geringen Verschlechterung der Empfindlichkeit führen. Ein vergleichbares IC ist der TSOP1836 von Vishai/Telefunken. Beide ICs benötigen nur eine Betriebsspannung von 5 V und kommen mit weniger als 2 mA aus. Sie

können daher direkt von der seriellen Schnittstelle des PCs versorgt werden.

Der hier vorgestellte IR-Transceiver besitzt neben dem Empfänger auch einen modulierten IR-Sender für Hilfsträgerfrequenzen zwischen 30 kHz und 40 kHz. Der IR-Lichtsender kann eingesetzt werden, um Geräte wie den Videorecorder oder das Fernsehgerät fernzusteuern. Man kann ihn aber auch zur seriellen Datenübertragung zwischen zwei PCs einsetzen.

Das Schaltbild nach **Bild 1** zeigt den Empfängerbaustein (IC3) mit einem Spannungsregler 78L05 (IC1). Die Betriebsspannung wird den RS232-Ausgängen DTR und RTS entnommen, die zur Entkopplung über die Dioden D1 und D2 zusammengeschaltet sind. Hier kann per PC-Programm eine Spannung von ca. 10 V eingeschaltet werden. Beide Leitungen versorgen über D6 und D7 auch den Sendeteil der Schaltung. Da beim Senden hohe Impulsströme

## IR-Transceiver

### Technische Daten

- Empfangsfrequenz: 30, 33, 36 kHz je nach Bestückung
- Sendefrequenz: 30...40 kHz stufenlos einstellbar
- Stromversorgung: aus der seriellen Schnittstelle
- Reichweite: ca. 10 m
- IR-Empfänger: für Fernbedienungen nach dem RC5-Standard
- IR-Sender: RC5-kompatibel
- IR-Datentransceiver: serielle Daten, max. 2400 Bd.

benötigt werden, muss ein relativ großer Elko von 4,7 µF (C1) eingesetzt werden, um die Eingangsspannung des Spannungsreglers zu glätten.

Der IR-Empfängerbaustein für 36-kHz-modulierte IR-Signale erzeugt an seinem mittleren Anschluss ein Ausgangssignal mit aktivem Low-Pegel. Diese Ausgangsimpulse werden direkt an die Leitung CTS gegeben, wo sie per Software dekodiert werden sollen. Ein zusätzlicher Pull-up-Widerstand ist erforderlich, da die CTS-Leitung einen relativ geringen Eingangswiderstand hat. Zusätzlich wird das Signal aber auch noch in einer Transistorstufe (T1) invertiert und an den Eingang RxD der seriellen Schnittstelle gelegt. Dieser Anschluss dient zum Empfang serieller Daten, zum Beispiel bei der Datenverbindung zwischen zwei PCs.

Der Infrarotsender besteht aus einer Modulatorstufe mit IC2 und einem Impulsverstärker mit T2 und T3 für zwei Infrarot-LEDs (D4 und D5). IC2 ist ein Timerbaustein 555, der als Oszillator geschaltet ist und negative Nadelimpulse von etwa 2 µs Dauer erzeugt. Die Frequenz ist mit P1 zwischen etwa 30 kHz und 40 kHz einstellbar. Je nach Anwendung kann mit dem Trimpotentiometer auf die passende Frequenz und damit auf beste Reichweite abgeglichen werden. Der 555 erhält seine Betriebsspannung von der TXD-Leitung, die den Sender ein- und ausschaltet und damit moduliert („tastet“).

Die Energie für die beiden IR-Sendediode kommt ebenfalls aus der seriellen Schnittstelle. Die Leitungen DTR und RTS laden über D6 und D7 einen Elko von 1 µF (C5). Die kurzen Ausgangsimpulse am Ausgang (Pin 3) des Timerbausteins schalten die LED-Treiberstufe mit T2 und T3 durch. Es entstehen Impulsströme von ca. 200 mA bis 300 mA. Die damit erzielte Reichweite beträgt etwa 10 Meter. Der Ladestrom aus der seriellen Schnittstelle ist mit maximal 40 mA zwar

relativ gering, zwischen den Impulsen bleibt aber genügend Zeit, um den Ladeelko neu zu füllen.

Für den Aufbau dieser einfachen Hardware ist die Platine in **Bild 2** vorgesehen. Bei der Bestückung muss man lediglich darauf achten, dass alle Dioden, Elkos und ICs „richtig herum“ eingelötet beziehungsweise eingesetzt werden. Ein sehr häufiger Fehler ist auch, dass anstelle der gewinkelten Sub-D-Buchse (9 Löcher) ein Sub-D-Stecker (9 Stifte) bestückt wird.

### RC5-Software-Dekoder

Der Datenausgang des Empfänger-ICs ist direkt mit der Leitung CTS verbunden. Hier erscheinen also Signale einer Fernbedienung, die auf den Empfänger gerichtet wird. Ein Programm zur Dekodierung des Signals muss nun die ankommenden Impulse auswerten, um die gedrückte Taste der Fernbedienung zu erkennen.

**Bild 3** zeigt das von einer RC5-Fernbedienung empfangene Signal. Das Diagramm wurde übrigens mit einem Logik-Analysator eingefangen, der in dem eingangs erwähnten neuen Elektor-Buch beschrieben wird. Ein Programm zeichnet dabei direkt Signaländerungen am CTS-Eingang auf. Das RC5-Protokoll verwendet ein so genanntes Biphasensignal. Die eigentliche Information steckt in den Phasenwechseln. Spätestens nach 1,776 ms wechselt der Signalpegel. Der Empfänger kann sich durch diese Pegelwechsel immer wieder neu auf das Signal synchronisieren. Das Signal wird mit einer immer gleich bleibenden Startsequenz eingeleitet. Dann folgen drei Datenbereiche, wobei Pegelwechsel im Abstand von 1,776 ms die eigentlichen Datenbits repräsentieren. Nach jedem dieser Pegelwechsel wartet der Empfänger zunächst etwas mehr als 0,888 ms und überliest den nächsten Oder-auch-nicht-Pegelwechsel. Der dann folgende Pegelwechsel ist sowohl ein Synchronpunkt als auch eine Bitinformation. Im Prinzip lassen sich auf diese Weise beliebig lange Datenworte übertragen. Beim RC5-Signal sind es genau 12 Bit:

Das Control-Bit (Ctl) wechselt bei jeder Tastenbetätigung zwischen 0 und 1. Der Empfänger kann auf diese Weise unterscheiden, ob eine Taste nur einmal und lange oder mehrmals gedrückt wird.

Die Geräte-Adresse (Adr) enthält 5 Bits, wobei höherwertige Bits zuerst übertragen werden. Übliche Geräte-Adressen sind 1 für Fernsehgeräte und 5 für Videorecorder. Auf diese Weise können mehrere Fernbedienungen im selben Raum eingesetzt werden. Der Datenbereich (Dat) enthält 6 Bits für bis

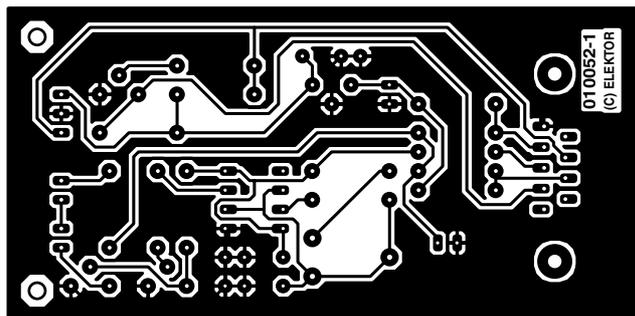
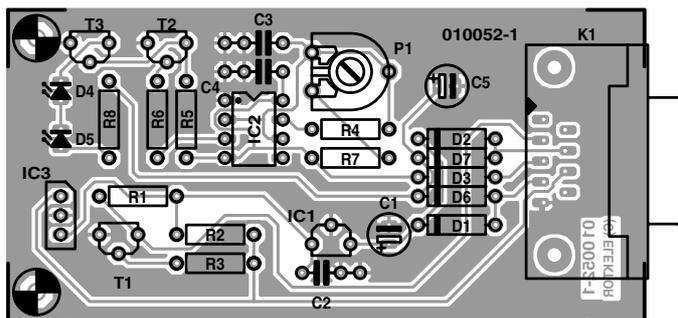


Bild 2. Platinenlayout und Bestückungsplan des IR-Transceivers.

#### Stückliste

##### Widerstände:

- R1, R2, R5 = 4k7
- R3 = 100 k
- R4 = 470 Ω
- R6 = 27 k
- R7 = 1 k
- R8 = 22 Ω
- P1 = 2k5-Trimpotentiometer

##### Kondensatoren:

- C1 = 4µ7/25 V stehend
- C2 = 100 n
- C3 = 10 n
- C4 = 22 n
- C5 = 1 µ/25 V stehend

##### Halbleiter:

- D1...D3, D6, D7 = 1N4148
- D4, D5 = IR-LED, z.B. LD271
- T1, T2 = BC547
- T3 = BC337
- IC1 = 78L05
- IC2 = 555
- IC3 = TSOP1836, SFH506-36

##### Außerdem:

- K1 = 9-polige Sub-D-Buchse (female!), gewinkelt, für Platinenmontage
- Platine 010052-1 (Layout-Download bei [www.elektor.de](http://www.elektor.de))

zu 64 unterschiedliche Tasten. Die Nummerntasten 0 bis 9 erzeugen Codes von 0 bis 9. Auch hier werden die höchstwertigen Bits zuerst gesendet.

**Listing 1** zeigt den eigentlichen Software-Dekoder in Visual Basic. Die Prozedur RC5 empfängt die Daten. Hier wird die PORT.DLL aus dem Buch "PC-Schnittstellen unter Windows" (Gratis-Download bei [www.elektor.de](http://www.elektor.de), siehe Hinweis am Artikelende) für alle Zugriffe auf die serielle Schnittstelle und für die Zeitsteuerung eingesetzt. Die Aufgabe

ist relativ zeitkritisch und erfordert die Verwendung von REALTIME True. Die Prozedur wartet zunächst auf einen Low-Pegel als Startimpuls. Damit der PC bei ausbleibendem Signal nicht endlos hängt, wurde eine Timeout-Bedingung eingebaut. Wenn nach 500 ms noch kein Signal empfangen wurde, bricht das Programm mit einer Fehlermeldung ab. Infrarot-Steuerungen können immer durch andere Lichtquellen gestört werden. Typische Störer sind Leuchtstofflampen, die impulsartige Lichtstörungen verursachen. Die Empfangsroutine RC5 überprüft

## Listing 1

### Empfang und Anzeige von RC5-Daten

```

Dim Ctr
Dim Adr
Dim Dat

Sub RC5Error()
  Ctr = -1
  Adr = -1
  Dat = -1
End Sub

Function RC5Bit() As Integer
  TIMEINITUS
  If CTS = 0 Then
    While ((TIMEREADUS < 500) And (CTS = 0))
      Wend
      If TIMEREADUS > 499 Then RC5Error
      DELAYUS 444
      If CTS = 0 Then RC5Error
      RC5Bit = 0
    Else
      While ((TIMEREADUS < 500) And (CTS = 1))
        Wend
        If TIMEREADUS > 499 Then RC5Error
        DELAYUS 444
        If CTS = 1 Then RC5Error
        RC5Bit = 1
      End If
    End Function

Sub RC5()
  Ctr = 0
  Adr = 0
  Dat = 0
  Startbit = False
  REALTIME True
  TIMEINIT
  While Not Startbit
    While (CTS = 1) And (TIMEREAD < 500)
      Startbit = True
      DELAYUS 444
      If CTS = 1 Then Startbit = False
      DELAYUS 888
      If CTS = 0 Then Startbit = False
      DELAYUS 888
      If CTS = 1 Then Startbit = False
      Adr = Startbit
      If TIMEREAD > 499 Then Startbit = True
    Wend
    DELAYUS 888
    Ctr = RC5Bit
    Adr = 0
    For N = 1 To 5
      DELAYUS 888
      Adr = Adr * 2
      Adr = Adr + RC5Bit
    Next N
    Dat = 0
    For N = 1 To 6
      DELAYUS 888
      Dat = Dat * 2
      Dat = Dat + RC5Bit
    Next N
    REALTIME (False)
  End Sub

Private Sub Form_Load()
  OPENCOM "COM2"
  DTR 1
  RTS 1
End Sub

Private Sub Form_Unload(Cancel As Integer)
  CLOSECOM
End Sub

Private Sub Timer1_Timer()
  RC5
  Text1.Text = Str$(Ctr) + " " + Str$(Adr) + " " +
  Str$(Dat)
End Sub

```

daher die Startsequenz eines empfangenen Signals auf den korrekten Verlauf. Falls ein Signal von der erwarteten Impulsfolge abweicht,

muss es sich um einen Störimpuls handeln. Die Prozedur wartet dann auf einen neuen Anfang. So ist es möglich, auch in relativ gestörter

Umgebung noch sicher RC5-Signale zu empfangen.

Nach der Startsequenz werden einzelne Bits in der Prozedur RC5bit empfangen. Wenn

## Listing 2

### Laden von Bildern für eine Diashow

```

Private Sub Timer1_Timer()
  RC5
  If Dat <> Dat_old Then
    If Dat = 1 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast11.jpg")
    If Dat = 2 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast21.jpg")
    If Dat = 3 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast31.jpg")
    If Dat = 4 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast41.jpg")
    If Dat = 5 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast51.jpg")
    If Dat = 6 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast61.jpg")
    If Dat = 7 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast71.jpg")
    If Dat = 8 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast81.jpg")
    If Dat = 9 Then Picture1.Picture = LoadPicture("D:\Homepage\Bast91.jpg")
  End If
  Dat_old = Dat
End Sub

```

beim Start der Prozedur ein 0-Pegel gelesen wird, sollte es sich um ein 0-Bit handeln. Nun wird der Pegelwechsel abgewartet. Nach einer halben Impulslänge ( $444 \mu\text{s}$ ) wird noch einmal abgefragt, ob immer noch der gleiche Pegel anliegt. Wenn nicht, wurde ein Fehler erkannt. Alle bisher gelesenen Informationen werden dann mit dem Wert -1 überschrieben. Die Prozedur RC5bit kehrt nach etwas mehr als  $444 \mu\text{s}$  mit dem Ergebnis des gelesenen Bits zurück. Die aufrufende Prozedur RC5 wertet das Bit aus und wartet dann noch einmal  $444 \mu\text{s}$  bis zum nächsten Aufruf von RC5bit. Auf diese Weise wird ein eventueller nächster Pegelwechsel überlesen. Das soll auch so sein, denn erst  $1,776 \text{ ms}$  nach dem letzten ausgewerteten Pegelwechsel wird der nächste gültige Wechsel erwartet.

Der Zeichenempfang wird von einem Timer des VB-Programms gesteuert. Alle empfan-

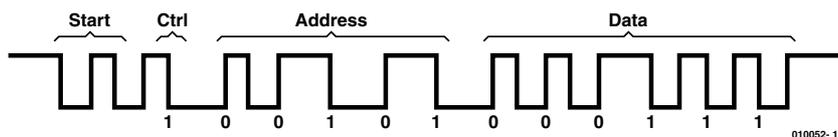


Bild 3. Ein RC5-Beispielsignal mit Adresse 5 und Taste 7.



Bild 4. Ausgabe von RC5-Daten.



Bild 5. Die Diashow am Bildschirm.

genen Informationen werden in einem Textfenster dargestellt. **Bild 4** zeigt die Bildschirmausgabe der empfangenen Informationen für eine Fernbedienung mit der Geräteadresse 5 (Videorecorder). Es wurde die Zifferntaste 7 gedrückt. Wenn kein Signal empfangen wird, sieht man mit  $\text{Ctr} = -1$  und  $\text{Adr} = -1$  eine entsprechende Fehlermeldung. In einer fertigen Entwicklung würde die Fehlerinformation nicht ausgegeben, sondern verwendet, um die brauchbaren von unbrauchbaren Tasteninformatio-

nen zu unterscheiden. Man kann das Programm verwenden, um sich mit den einzelnen Codes der eigenen Fernbedienung vertraut zu machen. Dies kann die Grundlage für spätere Steuerungen sein, bei denen der PC eine Fernbedienung ersetzt. Vollständige Listen der üblichen Codes findet man auch im Internet.

## PC-Fernbedienung

Eine andere sinnvolle Anwendung ist die Fernbedienung des PCs. Man kann zum Beispiel eine private Diashow mit der Fernbedienung steuern. **Listing 2** zeigt einen Programmausschnitt mit der veränderten Timerprozedur. Hier werden neun verschiedene Tastencodes ausgewertet, um Bilder von der Festplatte zu laden. Dass es hier nur neun Bilder sind, mag viele leidige Opfer ausufernder Diabende erfreuen. Aber einer Erweiterung steht nichts im Wege. Man

könnte zum Beispiel auch die Plus- und Minus-Tasten für die Steuerung einsetzen.

(100052-1e)

*In der nächsten Ausgabe wird die Software für einen RC5-Sender mit dem PC und der IR-Transceiverplatine vorgestellt. Damit kann man dann Videorecorder und andere Geräte vom PC aus steuern und serielle Daten zwischen zwei PCs über die Infrarotverbindung übertragen.*

## Literatur:

Burkhard Kainka:  
"Elektronik am PC - Visual Basic in der Praxis",  
Elektor Verlag, Aachen (ISBN 3-89576-113-3)

Burkhard Kainka:  
"PC-Schnittstellen unter Windows",  
Elektor Verlag, Aachen (ISBN 3-89576-086-2)

## Download-Hinweis:

Die in diesem Beitrag verwendete PORT.DLL sowie die Programm listings und das Platinenlayout stehen auf dem Elektor-Website [www.elektor.de](http://www.elektor.de) zum kostenlosen Download zur Verfügung. Sie finden die Download-Dateien wie immer auf der Homepage, indem Sie den Menüpunkt „Download“ und danach den Monat der Veröffentlichung des Artikels auswählen.

Die Software-Dateien sind auch auf der Homepage des Autors verfügbar:  
<http://home.t-online.de/home/B.Kainka>

# Infrarot-Transceiver für den PC (2)

## RC5-Sender und Datenlink

Von Burkhard Kainka

Die Schaltung des IR-Transceivers wurde bereits in der letzten Ausgabe zusammen mit Anwendungsbeispielen für den Betrieb als RC5-Empfänger am PC vorgestellt. Im zweiten und letzten Teil wird der Sendebetrieb aufgenommen: Neben dem Fernsteuern mit dem PC wird auch die Infrarot-Datenübertragung von PC zu PC anhand von Beispielen verständlich und nachvollziehbar beschrieben.

Nicht nur der Empfang von Fernbedienungssignalen, sondern auch ihre Erzeugung kann sinnvoll sein, um zum Beispiel den Videorecorder automatisch zu steuern, Aufnahmen einer Videokamera zu starten oder das Fernsehgerät zu einer bestimmten Uhrzeit auszuschalten. Im Prinzip kann jede Steuerungsfunktion einer RC5-IR-Fernbedienung automatisch vom PC aus ausgeführt werden.

### RC-5-Sender

In üblichen Fernbedienungen verwendet man kurze Impulse von weniger als 10% Einschaltdauer, aber mit Stromstärken bis etwa 1 A. Dadurch ist die Signalstärke beim Empfänger auch dann noch ausreichend, wenn er das gesendete Signal nur durch Reflexionen an weißen Wänden erhält. Eine derart große Stromstärke ist allerdings mit der seriellen Schnittstelle nicht zu erreichen. Dafür hat man aber genügend Spannung, um mehrere IR-LEDs in Reihe zu betreiben. Der IR-Transceiver verwendet zwei Sendedioden bei

einem Impulsstrom von ca. 200 mA bis 300 mA und erzeugt so kurze und kräftige Lichtimpulse, die eine Reichweite von mehreren Metern sichern.

Das erste Sende-Programm (**Listing 1**) verwendet eine Prozedur RC5tx zur Erzeugung der Steuerimpulse an TXD. Die drei Datenbereiche Ctr, Adr und Dat werden beim Aufruf übergeben. Der Anwender kann in entsprechende Textfenster die Geräteadresse und den Tastencode eingeben. Der Steuerungscode wird insgesamt 5 mal wiederholt, so dass 0,5 Sekunden lang Daten gesendet werden.

Die Prozedur RC5tx arbeitet im REALTIME-Modus. Sie erzeugt zunächst die Startsequenz und verarbeitet dann die übergebenen Steuerdaten. Jedes Bit wird durch eine Art Schiebeoperation durch eine Division durch 2 an die Stelle Null

gebracht und so isoliert. Mit dem einzelnen Bit wird die Prozedur RC5bit angesteuert.

Die DLL-Funktion REALTIME aus der Port.DLL bewirkt eine hohe Priorität des eigenen Programms. Die entscheidende Prozedur kann daher nicht durch andere Windows-Prozesse unterbrochen werden. Dieses Vorgehen mindert das allgemeine Problem der mangelnden Echtzeitfähigkeit eines Windows-Programms. Unterbrechungen zum Beispiel durch Mausereignisse sind für die Zeit eines RC5-Pakets unterbunden. Wichtig ist aber die Zeile "REALTIME false" am Ende der Prozedur. Damit wird der Normalzustand wieder hergestellt.

Der RC5-Sender kann für eine automatische Steuerung verwendet werden. Man muss dazu die Geräteadresse (siehe **Tabelle 1**) und den

### Redaktioneller Hinweis:

Die Listings 4 und 5 konnten wegen ihres Umfangs nicht in diesem Beitrag untergebracht werden. Alle besprochenen Listings stehen auf dem Elektor-Website [www.elektor.de](http://www.elektor.de) zusammen mit den Programmdateien zum kostenlosen Download zur Verfügung und sind in der Downloadliste unter der Nummer 010052-2-11 und auf der Homepage des Autors (Adresse am Artikelende) zu finden. Leser ohne Internetzugang können die Listings auch per Post, Fax oder telefonisch bei der Redaktion anfordern.

Tastencode (siehe **Tabelle 2**) kennen. Der Sender kann dann zum Beispiel zu einer bestimmten Urzeit oder auf ein anderes Ereignis hin ein geeignetes Kommando erzeugen. Hierzu ein Beispiel:

Ein Videorecorder soll zu einer bestimmten Urzeit auf Aufnahme geschaltet werden und zu einer anderen Uhrzeit gestoppt werden. Das Programm VCRtimer (**Listing 2**) erlaubt die Eingabe der Schaltzeiten. Außerdem gibt es noch zwei Schaltflächen für die direkte Bedienung. Ein weiteres Programmbeispiel beschäftigt sich mit dem so genannten Zapperfinger. Dieses gefürchtete Leiden ereilt angesichts der heuti-

gen Programmvielfalt so manchen Fernsehfreund genauso wie den Sportsfreund der Tennisarm und den Computerfreak die Maushand. Der RC5-Transceiver kann hier zusammen mit dem kleinen Programm in **Listing 3** Abhilfe schaffen.

## Serielle Datenübertragung

Bisher war nur von Infrarotsignalen nach dem RC5-Standard die Rede. Im Folgenden sollen nun auch serielle Daten zwischen PCs über moduliertes Infrarotlicht übertragen werden. Mit dem IR-Transceiver kann man zum Beispiel in einem Computerraum chatten, also Textbotschaf-



Bild 1. Ein Testprogramm für RC5-Signale.

## Listing 1

### Das Programm des RC5-Senders (RC5TxTest.vbp)

```
Dim Control
Private Sub Form_Load()
    i = OPENCOM("COM2,1200,N,8,1")
    If i = 0 Then
        i = OPENCOM("COM1,1200,N,8,1")
        Option1.Value = True
    End If
    If i = 0 Then MsgBox("COM Interface Error")
    RTS 1
    DTR 1
    Control = 1
    Chan = 0
End Sub
Private Sub Form_Unload(Cancel As Integer)
    CLOSECOM
End Sub
Private Sub Option1_Click()
    i = OPENCOM("COM1,1200,N,8,1")
    If i = 0 Then MsgBox("COM1 not available")
    RTS 1
    DTR 1
End Sub

Private Sub Option2_Click()
    i = OPENCOM("COM2,1200,N,8,1")
    If i = 0 Then MsgBox("COM2 not available")
    RTS 1
    DTR 1
End Sub

Private Sub TxBit(ONOff!)
    If ONOff = 1 Then
        TXD 0
        DELAYUS 888
        TXD 1
        DELAYUS 888
    Else:
        TXD 1
        DELAYUS 888
        TXD 0
        DELAYUS 888
    End If
End Sub

Private Sub RC5tx(Ctr, Adr, Dat)
    BitTime = 888
    REALTIME True
    TIMEINITUS
    TXD 1
    While TIMEREADUS < (BitTime * 1)
    Wend
    TXD 0
    While TIMEREADUS < (BitTime * 2)
    Wend
    TXD 1
    While TIMEREADUS < (BitTime * 3)
    Wend
    TxBit (Ctr)
    BitVal = 16
    For n = 1 To 5
        b = Int((Adr And BitVal) / BitVal)
        TxBit (b)
        BitVal = BitVal / 2
    Next n
    BitVal = 32
    For n = 1 To 6
        b = Int((Dat And BitVal) / BitVal)
        TxBit (b)
        BitVal = BitVal / 2
    Next n
    TXD 0
    REALTIME False
End Sub

Private Sub Chan1()
    For n = 1 To 3
        RC5tx Control, 5, 1
        DELAY 100
    Next n
    If Control = 1 Then
        Control = 0
    Else
        Control = 1
    End If
    Chan = 1
End Sub

Private Sub Command1_Click()
    Adr = Val(Text1.Text)
    Dat = Val(Text2.Text)
    For n = 1 To 5
        RC5tx Control, Adr, Dat
        DELAY 100
    Next n
    If Control = 1 Then
        Control = 0
    Else
        Control = 1
    End If
End Sub
```



Bild 2. Der Videotimer in Betrieb.

ten hin und her senden. Einige Vorversuche erklären die Vorgänge beim Senden eines seriellen Signals.

Im ersten Teil dieses Aufsatzes wurde die serielle Schnittstelle durchweg anders eingesetzt als von ihren Entwicklern ursprünglich vorgesehen.

Der eigentliche Zweck dieser Schnittstelle war die Datenübertragung über Modems. Bezeichnungen wie "Ring Indikator" oder "Data Terminal Ready" zeugen noch davon. Im Laufe der Zeit hat sich die Schnittstelle immer weiter auch in anderen Bereichen nützlich gemacht. Die größte Verbreitung fand der Anschluss der Computermaus an die RS232. Bei der RS232-Einführung hat wohl niemand daran gedacht, dass einmal Geräte mit eigenem Mikrocontroller an der Schnittstelle arbeiten würden, die sogar die Betriebsspannung aus der Schnittstelle beziehen. Im Gegensatz dazu war das dann (viele) Jahre später einer der Ausgangs-



Bild 3. Das Automatik-Zapperprogramm.

punkte bei der Konzeption einer neuen seriellen Schnittstelle namens USB.

Doch zurück zum guten alten RS232-Port, der für die Übertragung von Daten auf zwei Leitungen (TXD und RXD) entwickelt wurde. Die Daten werden dazu von der Hardware der Schnittstelle in einen seriellen Bitstrom umgewandelt und auf die Reise geschickt. Beim Empfänger erzeugt eine zweite Schnittstelle aus dem seriellen Datenstrom wieder parallele Datenbytes, die vom PC weiterverarbeitet werden können. Das Ganze ist ein Beispiel gelungener Arbeitsteilung zwischen Software und Hardware. Die Aussendung eines Zeichens über die serielle Schnittstelle benötigt viel Zeit im Vergleich zu anderen Operationen. Hier geht es um Millisekunden, während man sonst eher um Mikrosekunden kämpft. Die Schnittstelle im PC verfügt daher über einen speziellen Hardware-Baustein, den universellen seriellen Sendeempfänger (Universal Serial Receiver Transmitter, UART), eine Art Telegrafienstation des PC. Nachrichten werden hier in Auftrag gegeben und selbstständig abgesandt. Umgekehrt empfängt der UART eigenständig Nachrichten und legt sie so ab, dass sie von einem Programm in einer Aktion abgeholt werden können.

Eine typische Aufgabe für die serielle Schnittstelle ist die Übertragung von Daten von einem PC zum andern. Das können beliebige Daten sein, wie Textdaten, Programme, Bilder und so weiter. Letztlich werden einzelne Bytes versandt, also Abschnitte von acht Bits. Es gibt aber auch Einstellungen der Schnittstelle, bei denen ein Zeichen nur sieben Bits oder sogar nur 5 Bits hat. Die 5-Bit-Zeichen sind ein Erbe aus der Zeit der mechanischen Fernschreiber, die diese Zeichen im Prinzip genauso seriell über die Leitung schickten wie eine RS232-Schnittstelle, nur sehr viel langsamer und sehr laut.

### Die Nullmodem-Verbindung

In einem ersten Versuch sollen zwei PCs verbunden werden, um gegenseitig Zeichen auszutauschen. Nötig ist dazu eine kreuzweise Verbindung

### Tabelle 1 Geräteadressen

Geräteadresse	Gerät
0	Fernseher
2	Teletext
5	Videorecorder
7	Experimentell
16	Vorverstärker
17	Receiver/Tuner
18	Kassettenrecorder
19	Experimentell

### Tabelle 2 Einige wichtige Tastencodes für TV und VCR

Code	Funktion
0...9	Zifferntasten 0..9
12	Standby
13	Stummschaltung
16	Lautstärke +
17	Lautstärke -
32	Programm +
33	Programm -
48	Video Pause
53	Video Play, TV Ton an
54	Video Stopp
55	Video Aufnahme

der Leitungen TXD und RXD. Man bezeichnet eine solche Verbindung auch als Nullmodem-Kabel, weil hier kein Modem verwendet wird. Während über große Entfernungen die Telefonleitungen und Modems verwendet werden, kommt man bei Entfernungen bis zu einigen Metern allein mit normalen Kabeln aus (**Bild 4a**). Man kann den Versuch auch mit einem einzigen PC ausführen, der sich auf dem Umweg über die serielle Schnittstelle selbst Daten schickt (**Bild 4b**).

Für die serielle Übertragung von Daten hält die bereits vom vorherigen Artikel bekannte PORT.DLL die Prozedur SENDBYTE und die Funktion READBYTE bereit. Eine weitere Voraussetzung für einen gelungenen Datenaustausch ist, dass die Schnittstellen beider Partner mit den selben Parametern initialisiert wurden. Mit OPENCOM ("COM2,1200,N,8,1") einigt man sich auf eine Übertragungsgeschwindigkeit von 1200 Bits pro Sekunde (1200 Baud), kein Paritätsbit (Prüfbit), acht Bits pro Zei-

## Listing 2

### Die wichtigsten Prozeduren des Videotimers (VCRtimer.vbp)

```

Private Sub RecordOn()
  For n = 1 To 10
    RC5tx Control, 5, 55
    DELAY 100
  Next n
  If Control = 1 Then
    Control = 0
  Else
    Control = 1
  End If
  ONOff = 1
End Sub

Private Sub RecordOff()
  For n = 1 To 10
    RC5tx Control, 5, 54
    DELAY 100
  Next n
  If Control = 1 Then
    Control = 0
  Else
    Control = 1
  End If
  ONOff = 0
End Sub

Private Sub Timer1_Timer()
  Text1.Text = Time$
  If (Time$ > Text2.Text) And (Time$ < Text3.Text) And (ONOff = 0) Then RecordOn
  If (Time$ > Text3.Text) And (ONOff = 1) Then RecordOff
End Sub

Private Sub Command1_Click()
  RecordOn
End Sub

Private Sub Command2_Click()
  RecordOff
End Sub

```

chen und ein Stoppbit. Diese Parameter werden weiter unten noch genauer betrachtet.

**Bild 5** zeigt das Oszillogramm eines seriellen Zeichens "1", das mit 1200 Baud gesendet wird. Bei einer Übertragungsrate von 1200 Baud benötigt ein Bit eine Zeit von  $1s/1200 = 833\mu s$ . Links im Bild erkennt man das Startbit mit der Länge von 0,833 ms. Es folgen acht Abschnitte mit ebenfalls jeweils 0,833ms. Nur das erste Bit nach dem Startbit hebt sich mit einem Nullpegel deutlich aus dem Datenstrom ab. Da hier eine Eins gesendet wurde, erkennt man, dass die niederwertigen Bits zuerst übertragen werden und dass Datenbits mit invertiertem Pegel gesendet werden. Es folgen noch sieben High-Zustände mit ebenfalls jeweils 0,833 ms. Sie stehen für die sieben noch folgenden Nullbits. Das gelesene Byte lautet

also 00000001 = 1 Danach folgt, wenn auch unsichtbar, ein Stoppbit als Nullpegel. Das Stoppbit ist nichts anderes als die kleinste mögliche Pause zu einem eventuell folgenden Zeichen. Die tatsächliche Pause kann beliebig lang sein.

Aus diesem Aufbau eines seriellen

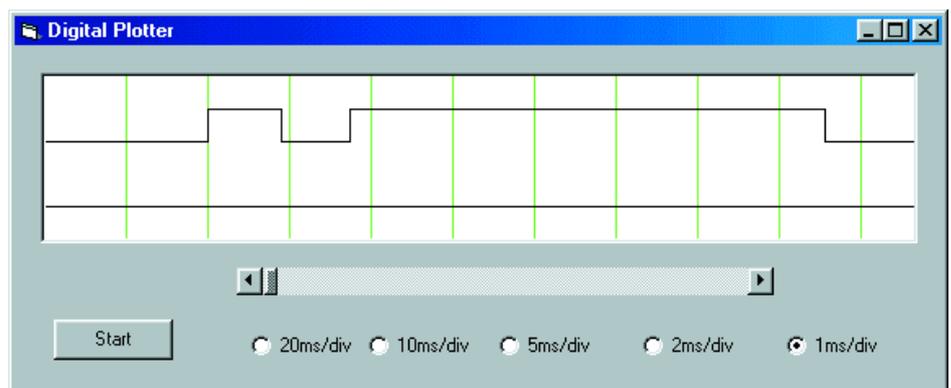
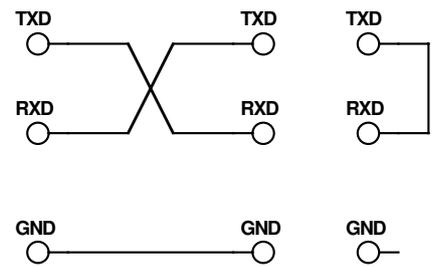


Bild 5. Senden eines Byte 1 bei 1200 Baud .



010052 - 18

Bild 4. Nullmodem-Verbindung (a) und Test mit einer einzigen Schnittstelle (b).

Zeichens folgt die erforderliche Ablauf beim Empfang eines Zeichens im UART. Der Baustein muss seine Eingangsleitung laufend beobachten, bis eine Low-High-Flanke auftaucht. Das ist dann der Beginn eines Startbits. Nun muss der UART einen inneren Zeitmesser starten und dieses Startbit abwarten. Darauf muss noch einmal eine halbe Bitlänge gewartet werden, um dann den Pegel abzufragen. So wird das erste Bit gelesen. Jedes folgende Bit wird jeweils genau eine Bitlänge später eingelesen. Damit ist der Lesevorgang theoretisch eine halbe Bitlänge vor dem Erreichen des Stoppbits abgeschlossen.

Man sieht, dass eine Synchronisation auf den Datenrahmen nur einmal, nämlich am Anfang des Startbits stattfindet. Der übrige Zeitrahmen muss genau bekannt sein und auch relativ exakt eingehalten werden. Falls ein Sender mit einer anderen Baudrate als der Empfänger arbeitet, muss es notwendig zu Lesefehlern kommen. Moderne UARTS begnügen sich übrigens nicht mit einer einzelnen Abtastung jedes Datenbits, sondern lesen etwa in der Mitte des Bits zum Beispiel drei Zustände ein, die im Normalfall gleich sein sollten. Falls es jedoch auf Grund von Störungen zu ungleichen Ergebnissen kommen sollte, wird eine Mehrheitsentscheidung gefällt. Auf diese Weise erreicht man eine

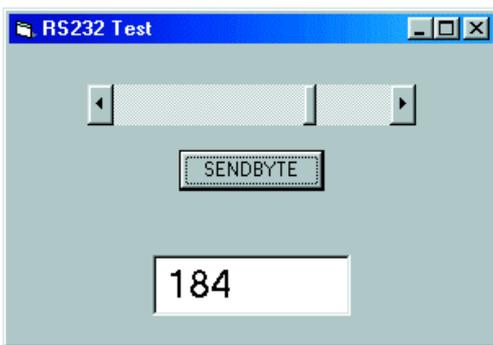


Bild 6. Senden und Empfangen einzelner Bytes.

größere Datensicherheit. Das erste Programmbeispiel (**Listing 4**, siehe Downloads) zur Datenübertragung überträgt bei jedem Kommando nur ein Byte. Und zwar wird in diesem Fall die Position eines Schiebereglers als Zahlenwert zwischen 0 und 255 an den Partner übermittelt. Umgekehrt wird jedes empfangene Byte in einem Textfenster angezeigt (**Bild 6**). Mit einem weiteren Listing (**Listing 5**) und einer Beschreibung in der Download-Datei erhalten Sie ein universelles Terminal-Programm, das sich speziell für eigene Entwicklungen und Versuche eignet.

### Licht-Datenübertragung

Ein einfacher Versuch zur optischen Signalübertragung benötigt nur eine IR-Sendediode und einen Fototransistor (siehe **Bild 7**). Während die Sende-LED direkt an TXD betrieben wird, benötigt der Empfänger eine Hilfsspannung, die man zum Beispiel von der hochgesetzten DTR-Leitung bekommen kann. Sobald das Licht der Sendediode auf den Fototransistor fällt, fließt Strom, und TXD wird hochgezogen. Damit erhält man das originalgetreue Signal von TXD auch an RXD. Der Versuch kann zum Beispiel mit dem Terminalprogramm aus der Download-Datei durchgeführt werden. Dabei spielt es keine Rolle, ob Daten zwischen zwei PCs ausgetauscht werden oder ob man den Versuch mit nur einem PC und Rückkopplung analog zu Bild 4b durchführt. Man muss nur daran denken, die DTR-Leitung beim Empfänger einzuschalten. Die einfache Übertragungsstrecke hat nur eine geringe Reichweite von wenigen Millimetern. Man kann jedoch Lichtwellenleiter einsetzen und kommt dann schon auf einige Meter. Es gibt spezielle Lichtsender und -empfänger, die sich einfach mit üblichen Lichtwellenleitern verbinden lassen. Beispiele sind die Sendediode SFH250 und der Fototransistor SFH350, die Bohrungen zur Aufnahme von Lichtwellenleitern haben. Mit

## Listing 3

### Automatischer Programmwechsel (Zapper.vbp)

```
Private Sub Chan1()
    For n = 1 To 3
        RC5tx Control, 5, 1
        DELAY 100
    Next n
    If Control = 1 Then
        Control = 0
    Else
        Control = 1
    End If
    Chan = 1
End Sub
Private Sub ChanPlus()
    For n = 1 To 3
        RC5tx Control, 5, 32
        DELAY 100
    Next n
    If Control = 1 Then
        Control = 0
    Else
        Control = 1
    End If
    Chan = Chan + 1
End Sub

Private Sub Timer1_Timer()
    If Chan = 30 Then Chan1
    If (Chan > 0) And (Chan < 30) Then ChanPlus
    If Chan = 0 Then Chan1
    Text1.Text = "Chan " + Str$(Chan)
End Sub

Private Sub Command1_Click()
    Timer1.Enabled = True
End Sub

Private Sub Command2_Click()
    Timer1.Enabled = False
End Sub
```

einer solchen Lichtleiterstrecke können Daten mit beliebigen Übertragungsraten bis 115200 Baud übertragen werden.

Will man größere Reichweiten auch im freien Raum erreichen, dann kann man modulierte Signale einsetzen, wie sie zum Beispiel bei IR-Fernbedienungen üblich sind. Dafür können wir als Sender und Empfänger den IR-Transceiver (aus Teil 1 des Artikels) verwenden, der das Empfangssignal durch eine Transistorstufe invertiert, damit es den Ruhepegel Null hat. Allerdings sind die speziell für IR-Fernbedienungen entwickelten Lichtempfänger wie der SFH506 oder TSOP1836 für die Übertragung von RS232-Signalen nur dann geeignet, wenn man einige Besonderheiten beachtet. Der Grund: Beim Empfang von Fernbedienungssignalen ist auch bei großer Umgebungshelligkeit eine möglichst große Reichweite

gefordert. Dies gelingt durch ein Bandpassfilter und eine automatische Verstärkungsregelung (AGC, siehe **Bild 8**), wie man sie auch aus Radio- und Funkempfängern kennt. Eine solche Regelschaltung hat immer eine Zeitkonstante, die für den gegebenen Einsatzzweck optimiert sein muss.

Wegen dieser für Fernbedienungssignale dimensionierten Zeitkonstante dürfen die modulierten Sendepulse nicht zu kurz und nicht zu lang sein. Die Schaltung eignet sich nicht für Dauerpegel, weil dabei die automatische Verstärkungsregelung zu weit zurückregeln würde. Das Datenblatt nennt folgende Einschränkungen:

- Ein Burst muss mindestens aus 6 Impulsen bestehen, also bei 36 kHz mindestens 167 µs lang sein.
- Nach jedem Burst ist eine Pause von mindestens 9 Impulsen nötig

**Downloads:**

Auf dem Elektor-Website ([www.elektor.de](http://www.elektor.de)) steht in der Downloadliste zu dieser Ausgabe unter der Nummer 010052-2-11 eine Zip-Datei zum Download bereit. Nach dem Entzippen erhält man einen Ordner mit sämtlichen Programmdateien und eine Datei mit den Programmlistings und der Beschreibung des Terminalprogramms.

Die Homepage des Autors hat die Adresse:

<http://home.t-online.de/home/B.Kainka>

**Weiterführende Literatur:**

Das Mitte des Jahres im Elektor-Verlag erschienene Buch "Elektronik am PC" von Burkhard Kainka zeigt zahlreiche weitere Anwendungen vom digitalen Logikanalysator über die Klangerzeugung mit der seriellen Schnittstelle bis zu einfachen Oszilloskopen. Das Buch gibt auch einen Einblick in die inneren Abläufe einer DLL zur Ansteuerung der RS232-Schnittstelle. Der Leser erfährt einiges über Win32-API-Aufrufe, die direkte Verwendung mit Visual Basic und über den Aufbau einer eigenen DLL mit Delphi. Die für das Buch geschriebene DLL ermöglicht unter anderem die gleichzeitige Verwendung mehrerer COM-Schnittstellen in einem Programm.

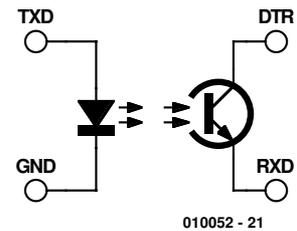


Bild 7. Die einfachste Licht-Übertragungsstrecke besteht nur aus LED und Fototransistor.

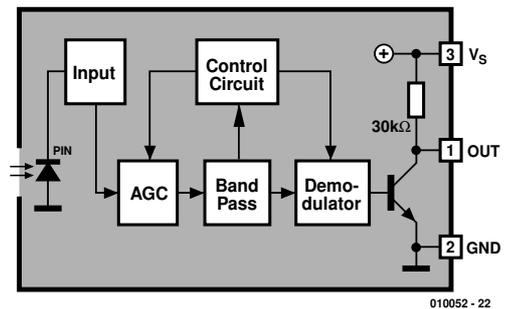


Bild 8. Blockschaltbild eines IR-Empfängers TSOP18xx.

(250µs).

– Spätestens alle 90 ms muss eine Pause von mindestens 15 ms eingehalten werden.

Wenn man vor allem die letzte Forderung immer erfüllen will, müssen die geforderten 15 ms durch die Stoppbits eines seriellen Signals sichergestellt werden. Der ungünstigste Fall liegt vor, wenn man eine kontinuierliche Folge von Nullbytes sendet. Dann nämlich sind das Startbit und acht Datenbits eingeschaltet. Nur das Stoppbit bildet eine Pause. Das bedeutet streng genommen, dass die maximale Baudrate bei einem kontinuierlichen Datenstrom 130 Baud betragen darf, wenn man zwei Stoppbits verwendet. Günstiger liegen die Verhältnisse, wenn kein kontinuierlicher Datenstrom, sondern sporadische Datenbytes anfallen, wie es zum Beispiel beim Tippen auf einer Tastatur der Fall ist. Aber auch die zufällige Mischung unterschiedlicher Bytes in einem anhaltenden Datenstrom bringt im Durchschnitt genügend Pausen.

Mit dem Terminalprogramm aus der Download-Datei lassen sich leicht verschiedene Übertragungsparameter ausprobieren (Bild 9). Ein ausführlicher Test mit von Hand eingetippten Zeichen ergab, dass alle Baudraten zwischen 50 Baud und 2400 Baud eine fehlerfreie Übertragung ermöglichen. Bei 2400 Baud beträgt die kleinste Impuls- und Pausenlänge 417 µs, was von den Daten des Empfängers abgedeckt wird. Die erreichbare Entfernung zwischen

Sender und Empfänger hängt vor allem von der verwendeten Sendediode und ihrem Diodenstrom ab. Das Datenblatt des TSOP18xx gibt eine typische Reichweite von immerhin 35 Metern an.

010052-2e

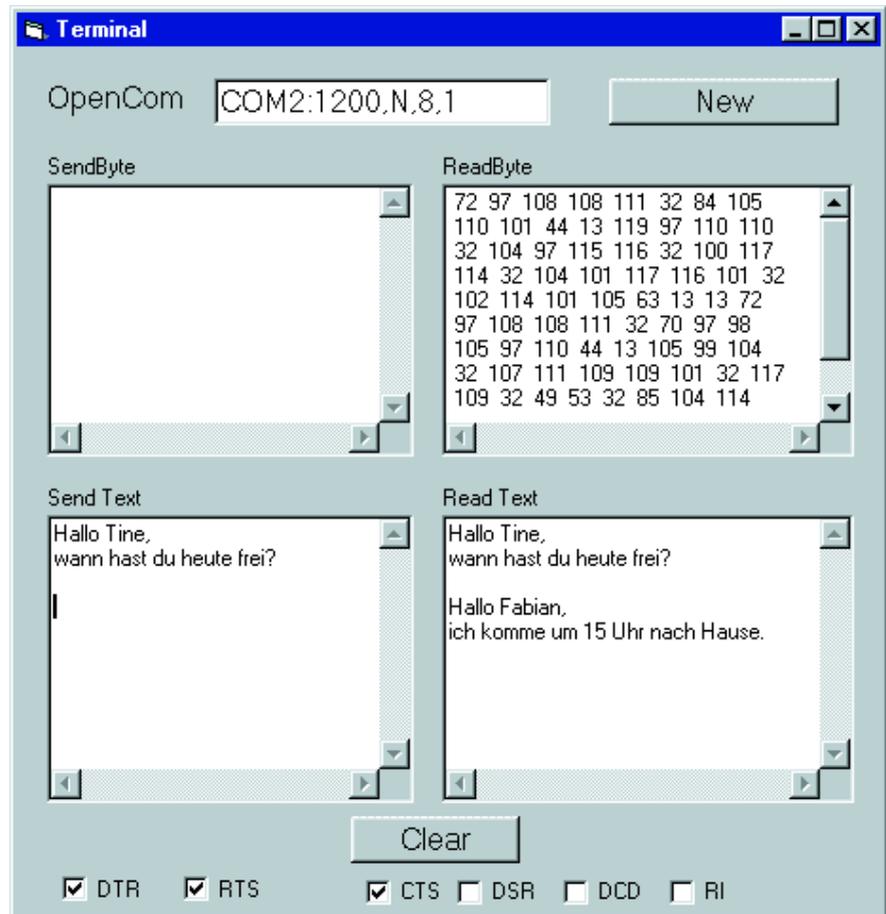


Bild 9. Der mit dem universellen Terminal-Programm eingerichtete Infrarot-Chatroom.