
Drahtlose Kommunikation mit IrDA

Gunther Zielosko

1. Grundlagen der Infrarot-Kommunikation

Nachdem sich bei elektronischen Geräten die Infrarot-Fernbedienung durchgesetzt hatte, gab es Anfang der 90-er Jahre Bestrebungen, diese Technik auch zur bidirektionalen Kommunikation zwischen Geräten einzusetzen. Es gab schon einzelne Rechner mit Infrarot-Schnittstelle, einen allgemeinen Standard jedoch nicht. 1993 war es dann soweit, ein aus etwa 30 Firmen zusammengesetztes Gremium entstand, die Infrared Data Association (IrDA). Sie arbeitete 1994 einen Standard zur geräteübergreifenden Infrarot-Kommunikation aus, der IrDA 1.0 oder Standard IR (SIR) genannt wurde. 1995 folgte IrDA 1.1, manchmal auch als Fast IrDA oder FIR bezeichnet. Beide Varianten arbeiten bezüglich der Transceiver (Kombination aus IR-Sender und Empfänger) mit derselben Technik, die Kommunikation ist jedoch vollkommen unterschiedlich:

- IrDA 1.0 arbeitet asynchron analog der bekannten RS232-Schnittstelle (COM-Port am PC) mit Datenraten bis zu 115,2 kBit/s. Die Übertragung erfolgt im Halbduplex-Betrieb mit einem Startbit, einem Stopbit und ohne Paritätsbit. Hardware-Grundlage sind UART's, wie sie auch für die Serielle RS232-Schnittstelle verwendet werden.
- IrDA 1.1 arbeitet synchron (Takt erforderlich) und erlaubt 4 Mbit/s. Der Preis für die höhere Geschwindigkeit ist, daß man nicht mehr auf einen UART zurückgreifen kann, sondern einen speziellen Controller benötigt.

Diese grundsätzlichen Festlegungen zur Arbeitsweise, wie Baudraten, Impulslängen usw., bilden sozusagen die physikalische Grundlage (physical layer). Ausführlich wird dieser Aspekt bei:

http://www.vishay.com/docs/irdc_physical_layer.pdf

beschrieben.

Will man jedoch über Infrarot mit anderen Geräten kommunizieren, muß man auch noch in die Welt der verschiedenen Software-Protokolle einsteigen. Ähnlich wie beim Funk-Telefon haben die einzelnen Geräte „Adressen“, gibt es Festlegungen, wer sich zuerst meldet (Master / Slave), wie die Kommunikation abläuft usw. Schließlich können bei IrDA im Gegensatz zum drahtgebundenen Datenaustausch mehrere Partner im Umfeld anwesend sein. Eine einfache Umschaltung der Richtung des Datenverkehrs durch Anlegen eines bestimmten Pegels auf einer der beiden Leitungen ist ebenfalls nicht möglich. Echter IrDA-Betrieb fordert einfach mehr „Disziplin“.

Die Familie der IrDA-fähigen Geräte ist groß, die Anwendung der vorhandenen Technik dagegen ist eher zurückhaltend. Wo finden wir denn IrDA-Schnittstellen?

- An fast allen Notebooks (fertig aufgebaut und Treiber vorhanden)
- An fast allen PDA's (Psion, Palm und Co.)
- An vielen Handies
- An vielen PC's (oft nur vorbereitet als Stiftleisten auf dem Motherboard, es fehlt nur ein IrDA-Transceiver-Baustein)

Ist IrDA nun auch für den BASIC-Tiger® interessant und wie kann man den BASIC-Tiger® IrDA-fähig machen? Wo liegen die Probleme?

Zur ersten Frage – IrDA ist drahtlos, elektrisch kaum zu stören und kann in „gefährlicher“ Umgebung benutzt werden. Wenn ein BASIC-Tiger® z.B. in einer Hochspannungsanlage (hier reicht schon das 220V-Netz!) arbeitet, können Daten nur mit hohem Aufwand (Optokoppler, Berührungsschutz, Potentialtrennung usw.) ausgetauscht werden. Über IrDA geht das problemlos bis weit über 1 m Entfernung durch isolierende Glasfenster oder durchsichtige Vergußmassen sowie elektrisch, chemisch bzw. biologisch kritische Räume hindurch. Ebenso kann man mit hermetisch dichten Systemen kommunizieren – Voraussetzung ist nur ein durchsichtiges Fenster. Denken Sie auch an einen Betrieb unter Wasser! Genauso interessant ist, daß im Gegensatz zu anderen Schnittstellen Verbindungen im laufenden Betrieb hergestellt und getrennt werden können. Obwohl dies auch für neuere drahtgebundene Schnittstellen wie USB immer wieder propagiert wird, zeigt die Praxis, daß es nicht immer funktioniert. Ein weiterer Aspekt ist, daß viele Geräte bereits mit IrDA-Schnittstelle ausgerüstet sind – der BASIC-Tiger® z.B. also sofort loslegen könnte, mit einem Handy über IrDA zu sprechen...

Zur zweiten Frage – Die Verwandtschaft von IrDA 1.0 zur RS232-Schnittstelle ermöglicht recht einfache Lösungen mit dem BASIC-Tiger®. In diesem Applikationsbericht werden wir deshalb ausschließlich IrDA 1.0 anwenden.

Nun zur dritten Frage – wo liegen die Probleme? Die „echte“ IrDA-Kommunikation läuft nach einem strengem Protokoll ab. Ein Standard-IrDA-Gerät kontrolliert vor dem Datenaustausch ihre Umgebung auf ein anderes IrDA-Gerät. Da werden Rollen verteilt (Host / Client), Adressen genannt und vieles andere mehr. Prinzipiell aber geht es – allerdings werden wir uns im Rahmen dieses Applikationsberichtes nicht mit der echten Kommunikation zu beliebigen IrDA-Geräten beschäftigen, sondern lediglich einen Datenverkehr analog zur RS232-Schnittstelle realisieren.

Die Signale von IrDA sind zwar verwandt mit denen der RS232-Schnittstelle, aber nicht identisch! Bei normalen UART-Daten, die zur seriellen Schnittstelle gesendet werden, wird eine „1“ durch einen Impuls dargestellt. Bei IrDA ist das umgekehrt, ein Licht-Impuls entspricht einer „0“. Diese Invertierung spart Strom, da High bei RS232 auch der Ruhepegel ist. Erst mit dem Start-Bit, bei IrDA einem Low Pegel, beginnt die Übertragung eines Bytes. Auf diese Weise verbraucht das IrDA-Interface nur dann Energie, wenn es auch wirklich Daten sendet, wichtig für alle batteriebetriebenen Geräte.

Die Standard-Pulslänge für IrDA 1.0 beträgt 3/16 der „Bit-Länge“ einer normalen RS232-Schnittstelle. Bei 115,2 Kbit/s ist ein Infrarotpuls damit 1,6 µs lang, bei einer

Übertragungsrate von 9600 Bit/s rund 20 μ s. Daraus folgt, daß wir die Signale der SER-Schnittstelle unseres BASIC-Tiger® vor der Umsetzung in Lichtsignale noch umformen müssen.

2. Ein fertiger Transceiver-Baustein, der TFDU4100 von Vishay

Die Firma Vishay stellt verschiedene IrDA-Transceiver her, wir haben den TFDU4100 gewählt, der in einem Gehäuse einen Infrarot-Sender und einen Infrarot-Empfänger vereinigt. Ein Datenblatt bekommt man unter:

<http://www.vishay.com/document/89000/89000.pdf>

Es gibt diesen und ähnliche Bausteine in verschiedenen Gehäusevarianten, letztlich ist es für unsere Anwendung gleichgültig, welche Bauform wir wählen. Der Autor hat eine Variante gemäß Bild 1 bzw. 2 gewählt. Ein Vorteil gegenüber Lösungen aus Einzelkomponenten ist bei diesem Baustein, daß bereits ein Regelverstärker zur Empfindlichkeitssteuerung eingebaut ist.

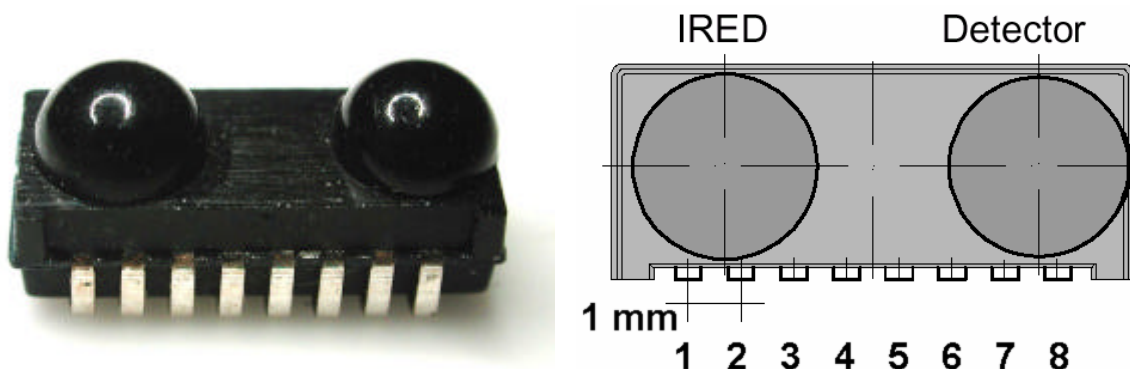


Bild 1 sieht sexy aus: der TFDU4100 von Vishay als IR-Transceiver *Bild 2 Anschlußbelegung des TFDU4100 von vorn auf die Linsen gesehen*

So toll der Baustein TFDU4100 aussieht, die nötige Impulsformung aus einem regulären RS232-Signalspiel kann er nicht allein erledigen, dazu ist noch ein Impuls-Verkürzer notwendig, der auch von Vishay angeboten wird.

3. Das Koppelglied zur SER-Schnittstelle, der TOIM3232 von Vishay

Der TOIM3232 enthält einen programmierbaren Baudraten-Generator, mit dem aus einer Quarzfrequenz von 3,6864 MHz verschiedene genormte Baudraten von 1200 bis 115200 Bd abgeleitet werden können. Außerdem verkürzt er die RS232-Impulse auf entweder die Standardlänge von 3/16 eines RS232-Impulses oder auf den stromsparenden Minimalwert von 1,627 μ s, auch das kann programmiert werden. Leider ist der TOIM3232 nicht mehr so ohne weiteres verfügbar. Der Ersatztyp TOIM4232 ist zwar funktionell und vom Gehäuse her identisch, arbeitet aber mit der „moderneren“ Betriebsspannung von 3,3 V. Eingangsseitig

verträgt der TOIM4232 aber auch 5 V-Pegel und der Tiger versteht rückwärts seine 3,3 V Pegel auch. Das Problem ist lediglich, daß man für den TOIM4232 bei der Zusammenschaltung mit dem BASIC-Tiger® noch einen zusätzlichen Spannungsregler braucht, der stabile 3,3 V erzeugt.

Noch etwas: der TOIM3232 als auch der TOIM4232 arbeiten auf der RS232-Seite trotz entsprechender Pinbezeichnungen in Wahrheit nicht mit echten RS232-Pegeln, sondern mit TTL-Pegeln. Für die Benutzung eines BASIC-Tigers® ohne eingebaute RS232-Pegelwandler eher ein Vorteil. Will man dagegen einen echten RS232-Tiger oder das Plug-and-Play-Lab benutzen, muß wieder mal ein MAX232 dazwischengeschaltet werden. Datenblätter zum TOIM4232 oder TOIM3232 erhält man unter:

<http://www.vishay.com/document/82546/82546.pdf>

<http://www.ida.ing.tu-bs.de/academics/labs/ist2/praktikumWS01/files/comports/toim3232.pdf>

Je nach verfügbarem Typ TOIM4232 oder TOIM3232 müssen Sie im ersteren Falle dessen Betriebsspannung auf 3,3V z.B. mit einem speziellen 3,3V-Regler begrenzen. Wenn Sie noch einen TOIM3232 ergatterten konnten, legen Sie ihn einfach mit an die VCC des BASIC-Tigers®.

Wie schon erwähnt, kann der TOIM3232 programmiert und so an den Anwendungsfall angepaßt werden. Die Programmierung erfolgt über die beiden logischen Anschlüsse RESET und BR/D direkt vom BASIC-Tiger® aus (siehe Komplettschaltbild unseres IrDA-Adapters, Bild 3 und 4). Der Programmiervorgang läuft über normale TTL-Pegel wie folgt ab:

Schritt	RESET (Pin 1)	BR/D (Pin 2)	Beschreibung
1	High	X	Rücksetzen aller Register und Einstellung der Datenrate auf 9600 Bits/s, IrDA-Impulsbreite auf 1,627 µs, Energiesparmodus
2	Low	X	Rücksetzen beenden, danach mindestens 7 µs warten
3	Low	High	Einschalten des Programmier-Modus, mindestens 7 µs warten
4	Low	High	Jetzt Senden des Steuerwortes YZ* (1 Byte) seriell auf Pin 3, RD_232 (Programmier-Daten siehe Tabellen 2 und 3) <ul style="list-style-type: none"> • Nach RESET Datenrate 9600 Bit/s, IRDA-Impulsbreite 1,267 µs • bei Wiederholung der Programmierung zuletzt eingestellte Parameter mindestens 1 µs warten
5	Low	Low	Einschalten des Daten-Kommunikations-Modus. RESET und BR/D bleiben während des Datenverkehrs beide Low. Der folgende Datenverkehr muß ab jetzt mit den eben eingestellten Parametern abgewickelt werden! Eine erneute Umschaltung der Schnittstellen-Parameter kann durch Abarbeiten der Schritte ab 3 erfolgen.

Tabelle 1 Ablauf der Programmierung des TOIM3232

Das Steuerbyte YZ enthält unter Y die 4 höherwertigen Bytes mit der Impulslänge S0 (Bit 4) sowie den Werten für die programmierbaren User-Ausgängen S1 und S2 (Bits 5 und 6) gemäß Tabelle 2. Die 4 niederwertigen Bits (Z) steuern die Datenrate (Tabelle 3).

Bit 7	Bit 6 (S2)*	Bit 5 (S1)*	Bit 4 (S0)	1. Zeichen	IrDA-Pulslänge
X	X	X	0	0	3/16 Bitlänge
X	X	X	1	1	1,627 µs

* Die Ausgänge S1 und S2 (Pins 12 und 13) können im Rahmen der Programmierung vom Nutzer beliebig auf Low oder High eingestellt werden. Wir benutzen in unserer Schaltung S2 zum Umschalten der Empfindlichkeit (SC = Sensitivity Control) des TFDU4100.

Tabelle 2 TOIM3232 - die Interpretation des 1. Programmier-Zeichens Y (Impulslänge)

Bit 3	Bit 2	Bit 1	Bit 0	2. Zeichen	Baud-Rate
0	0	0	0	0	115,200 k
0	0	0	1	1	57,6 k
0	0	1	0	2	38,4 k
0	0	1	1	3	19,2 k
0	1	0	0	4	14,4 k
0	1	0	1	5	12,8 k
0	1	1	0	6	9,6 k
0	1	1	1	7	7,2 k
1	0	0	0	8	4,8 k
1	0	0	1	9	3,6 k
1	0	1	0	A	2,4 k
1	0	1	1	B	1,8 k
1	1	0	0	C	1,2 k

Tabelle 3 TOIM3232 - die Interpretation des 2. Programmier-Zeichens Z (Baudrate)

Beispiel:

Das Datenbyte YZ sei in binärer Form: **11010110b**, also **D6h** hexadezimal oder **214** dezimal.
 Das bedeutet:

- S2 wird statisch auf 1 gesetzt
- S1 wird statisch auf 0 gesetzt
- Die Impulsbreite beträgt ab jetzt 1,627 µs
- Die Datenrate ist ab jetzt 9600 Bit/s (Bd)

Nun wissen wir fast alles über die beiden IC's, mit denen wir unseren BASIC-Tiger® IrDA-fähig machen wollen. Im folgenden Abschnitt werden wir die Zusammenschaltung mit dem BASIC-Tiger® behandeln.

4. Schaltung BASIC-Tiger® / IrDA-Schnittstelle

Hardwareseitig gibt es nicht viel zu tun, wir schalten an die entsprechenden seriellen Anschlüsse des BASIC-Tigers® zunächst den TOIM3232 und danach den TFDU4100 (Bild 3). Die vorgestellte Schaltung gilt für Tiger ohne eingebauten RS232-Pegelwandler. Bei Verwendung von Tigern mit eingebautem RS232- Pegelwandler oder des Plug-and-Play-Labs ist zwischen Tiger und dem Rest der Schaltung noch ein MAX232 einzufügen (Bild 4)!

Die schon beschriebene Besonderheit des TOIM4232 mit seiner Betriebsspannung von 3,3 V benötigt noch einen eigenen Regler, der diese Spannung aus der VCC des BASIC-Tigers® ableitet. Der Einfachheit halber haben wir deshalb die Schaltung für einen TOIM3232 ausgelegt, der das nicht braucht. Trotzdem sollte bezüglich der Spannungsversorgung einiges beachtet werden. Die Betriebsspannung des TFDU4100 sollte über ein Siebglied 47Ω / 100nF , $4,7\mu\text{F}$ gut gepuffert werden. Als Elkos kommen nur Tantalelkos in Frage. Ebenfalls kritisch ist die Versorgung der Sende-Diode am Pin IR-Anode. Da dort erhebliche Impulsströme auftreten ($0,2\text{ A}$!), sollte die Versorgung eigentlich über einen zweiten, unabhängigen 5V-Regler erfolgen. Wenn Sie diese Spannung dennoch aus der „normalen“ VCC abnehmen wollen, muß über eine RC-Kombination einmal der kurzzeitig fließende Impulsstrom gewährleistet sein und zum anderen sichergestellt werden, daß die Versorgung des BASIC-Tigers® dabei nicht zeitweise „in die Knie geht“. Die vorgeschlagene Kombination ist ein Kompromiß (siehe Hinweis am Ende des Berichtes!)

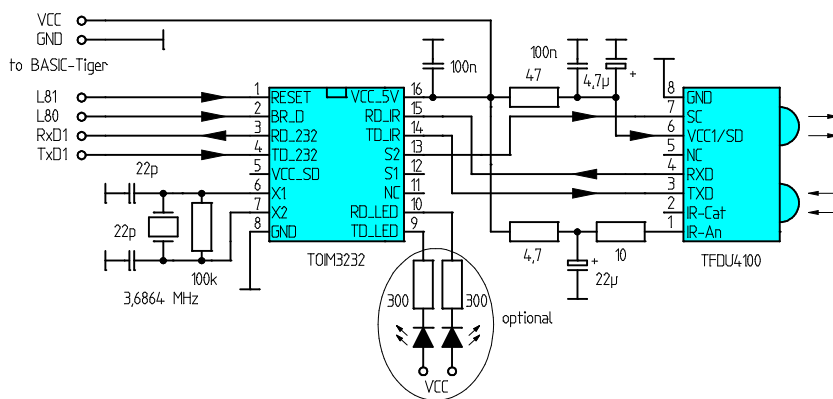


Bild 3 Schaltung des einfachen IrDA-Adapters für BASIC-Tiger® ohne RS232-Schnittstelle

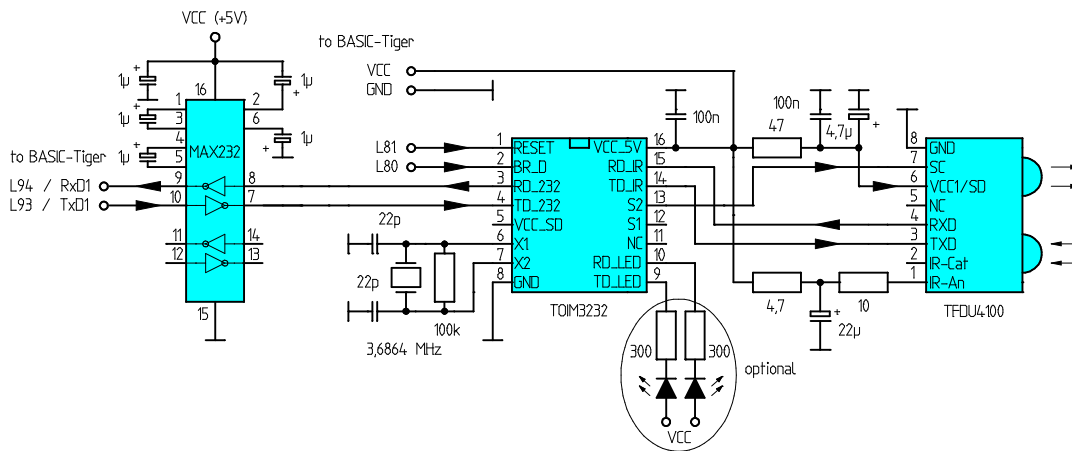


Bild 4 Dasselbe für BASIC-Tiger[®] mit eingebauter RS232-Schnittstelle

Sowohl der TOIM4232/3232 als auch der TFDU4100 haben ein SMD-Gehäuse (der TOIM4232/3232 ein besonders breites, der TFDU4100 ein spezielles mit 1 mm Raster), daher macht der Aufbau der Schaltung etwas Aufwand. Entweder gibt man sich mit einem „Drahtverhau“ zufrieden oder man stellt eine professionelle Leiterplatte (z.B. auf fotografischem Wege) her. Hier als Anregung das Layout der vom Autor hergestellten Leiterplatte (Bild 5) und der fertige Adapter (Bild 6).

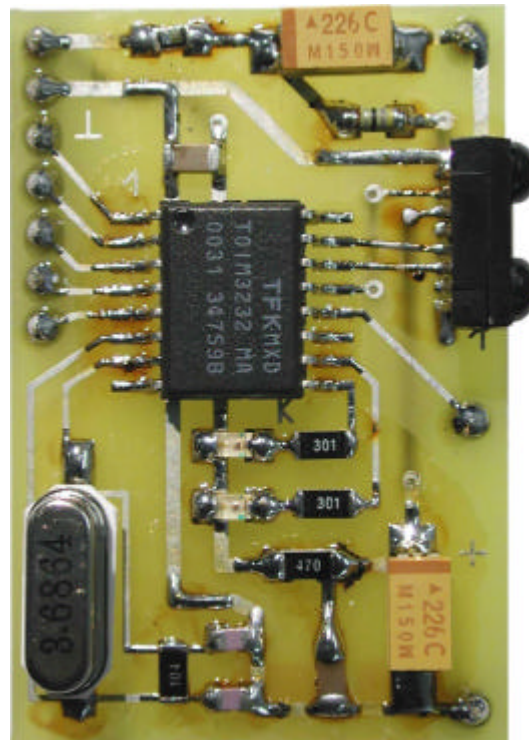
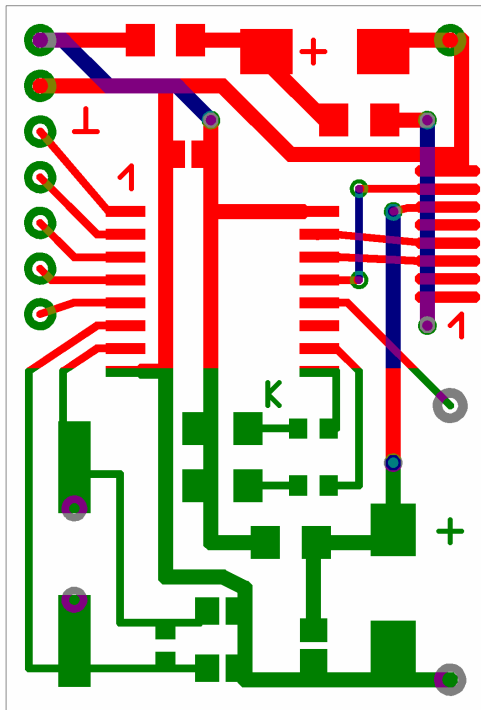


Bild 5 Layout des IrDA-Adapter (rot BE-Seite, blau Leiterseite) Bild 6 fertig bestückter IrDA-Adapter

Für erste Tests des Adapters beschalten Sie das Modul nacheinander wie folgt:

1. Ruhestrom

VCC	an + 5 V	keine LED sollte leuchten, der Stromverbrauch der Schaltung sollte höchstens bei ca. 5 mA liegen
GND	an GND	
RESET	an + 5 V	
BR_D	an GND	
TD_232	an GND	

2. Senden aktiv (nur Änderungen zu 1. dargestellt!)

RESET	an GND	TD_LED sollte jetzt leuchten, Stromverbrauch jetzt ca. 25 mA
-------	--------	--

3. TD_232 wechseln (nur Änderungen zu 2. dargestellt!)

TD_232	an + 5 V	TD_LED aus
TD_232	an GND	TD_LED an

4. Empfang prüfen (nur Änderungen zu 3. Dargestellt!)

TD_232	an + 5 V	TD_LED aus
--------	----------	------------

IrDA-Signal von außen einkoppeln RD_LED blinkt

Wenn es bis hierher geklappt hat, sehen die Chancen für Ihr IrDA-Modul schon recht gut aus.

Wie kann man nun zu anderen IrDA-Geräten Kontakt aufnehmen? Es gibt 2 prinzipielle Wege:

- Man kauft einen handelsüblichen oder baut sich einen eigenen IrDA-Adapter für die serielle Schnittstelle (z.B. für den PC). Damit erfolgt der Datenverkehr fast genauso, wie man das bei RS232 erwartet. Mit einem Terminalprogramm oder anderer passender Software sollte eine Datenübertragung vom bzw. zum BASIC-Tiger[®] möglich sein.
- Man geht den steinigigen Weg der „echten“ IrDA-Kommunikation und schreibt Programme für den BASIC-Tiger[®], die dann zumindest mit IrDA-1.0-Geräten nach Protokoll verkehren.

5. Software

Braucht man eigentlich noch Software für den Datenverkehr auf der „unteren“ Ebene? Der Datenverkehr läuft doch eigentlich vollkommen identisch zur gewohnten RS232-Schnittstelle ab. Ein paar zusätzliche BASIC-Zeilen benötigen wir schon – wir müssen zumindest neben der Installation des seriellen Gerätetreibers (zunächst mit der Standard-Baudrate 9600 Baud) noch den TOIM3232 initialisieren und dann die neue Baudrate einstellen, die auch das „Gegengerät“ versteht (als Beispiel 19200 Baud). Diese Baudrate muß nun noch am Gerätetreiber neu eingerichtet werden.

Als Gegengerät benutzen wir wiederum einem IrDA-fähigen BASIC-Tiger[®], weil wir hier Zugriff auf alle Parameter der Schnittstelle und alle Möglichkeiten der Manipulation haben.

Das Programm IRDA01.TIG zeigt beispielhaft, wie die Anpassung der Baudrate und die Einstellung anderer IrDA-Parameter erfolgen kann. Hier werden ohne Berücksichtigung irgendwelcher Protokolle ASCII Zeichen gesendet bzw. empfangen. Die Unterscheidung erfolgt in den letzten Zeilen (Kommentarzeichen jeweils umsetzen!), so daß man einen BASIC-Tiger® als Sender, einen anderen als Empfänger einrichten kann. Bild 7 zeigt, wie 2 solche Systeme, hier im Minilab und mit aufgesetztem IrDA-Adapter, arbeiten. Bei ersten Versuchen wurden auch mit gering eingestellter Empfindlichkeit (S2 auf „0“) mühelos 3 m und mehr überbrückt.

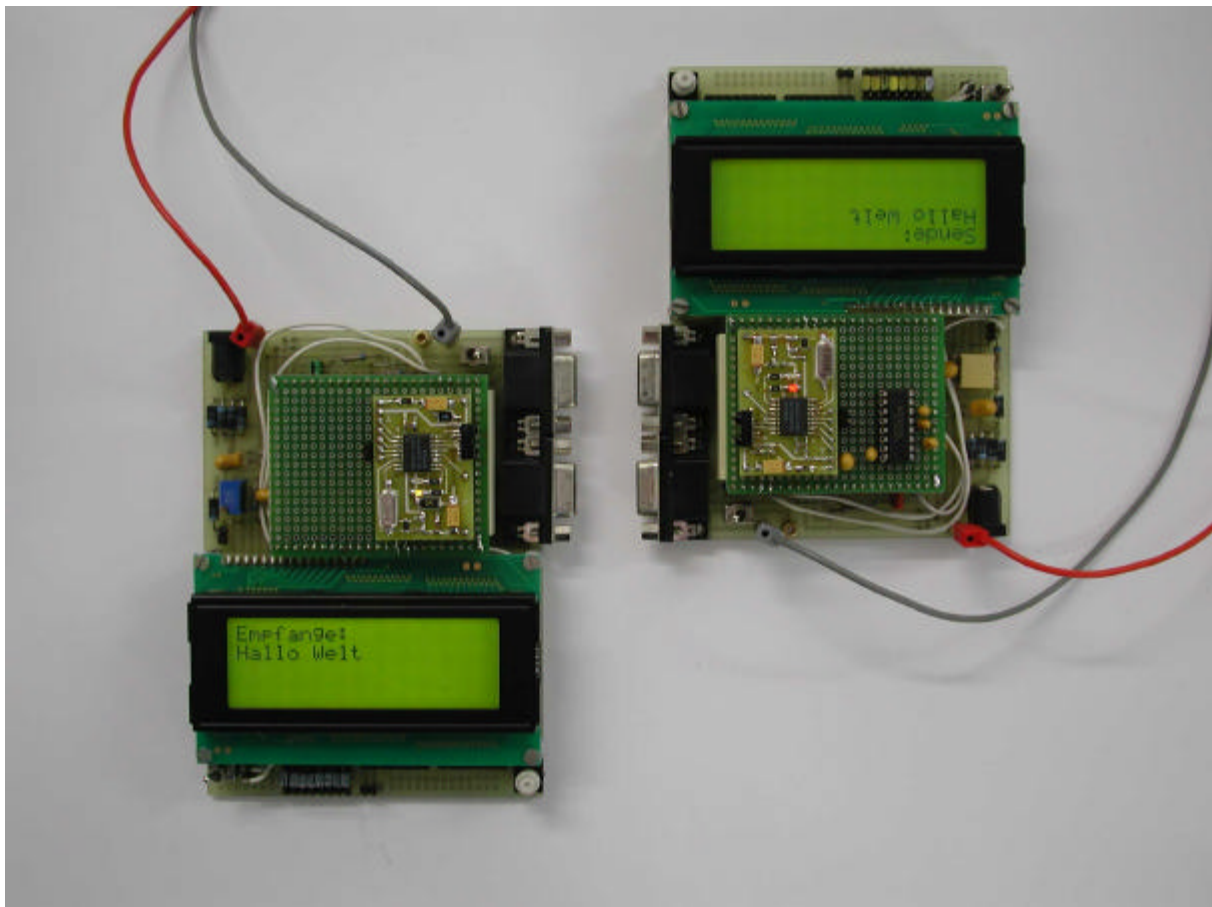


Bild 7 Zwei BASIC-Tiger® jeweils im Minilab kommunizieren über IrDA (links Empfänger – gelbe LED, rechts Sender – rote LED). Das rechte Gerät hat einen BASIC-Tiger® mit eingebauter RS232-Schnittstelle und benötigt daher zur „Pegel-Rückwandlung“ einen MAX232

Noch ein wichtiger Hinweis:

Wir wissen bereits, daß unser IrDA-Sender bei Aktivität einen erheblichen Strom benötigt. Der ungünstigste Fall für den Strombedarf liegt vor, wenn eine extrem geringe Baudrate (z.B. 1,2 kBd) und gleichzeitig die IrDA-Pulslänge auf 3/16 der Bitlänge eingestellt wird. Dann kann es bei der gewählten Schaltung (Spannungsversorgung des IrDA-Adapters aus der VCC des Tigers) vorkommen, daß diese VCC unter ihren

Mindestwert zusammenbricht. Das bewirkt dann einen RESET des BASIC-Tigers, vielleicht auch Schlimmeres! Wenn also wirklich sehr langsame Datenraten gebraucht werden, sollte die Impulslänge auf 1,627 μ s (minimale Impulslänge) eingestellt werden. Geht auch das nicht, muß die Adapterschaltung einen eigenen leistungsfähigen Spannungsregler bekommen!

Na – interessiert?

Wenn ja, warten eine Menge neue Möglichkeiten auf uns...Viel Erfolg!