

Infrarot-Steuerungssystem IRTXa/IRRXa

H.O.B.S.(\)Helmut Schellong

4. Nov. 2004

Inhaltsverzeichnis

1	Einleitung	3
1.1	Merkmale	3
2	Abbildungen: Sender, Empfänger	6
3	Bedienungselemente Sender	7
3.1	Tasten + Kommandos	8
4	Parameter des Senders	12
5	Parameter des Empfängers	14
6	Datenblatt des Senders	17
7	Datenblatt des Empfängers	18
8	Datenblatt Sender+Empfänger	21
A	Datenlisten des Senders	22
B	Kommando- und Parameterlisten des Empfängers	23
C	Weitere Abbildungen	32
C.1	Sender	32
C.2	Empfänger	33

Tabellenverzeichnis

1	Drehschalter und Tasten	7
2	Taste CMD, Funktion RAM	9
3	Taste CMD, Funktion DISPLAY	9
4	Taste CMD	9
5	Taste SEND, Funktion EEPROM	10
6	Taste SEND, Funktion TX->RX_PROG	10
7	Taste SEND	11

1 Einleitung

Dieses System besteht aus einer Infrarot-Fernbedienung und mindestens einem Empfänger.

Der batteriebetriebene Sender hat 4 Drehschalter und 20 Tasten, wovon 16 Tasten für die Kanäle 1-16 und 4 für feste Sonderfunktionen vorgesehen sind. Beide Geräte sind unabhängig voneinander sehr umfangreich programmierbar. Der Empfänger kann per PC und Terminalprogramm oder mit dem Sender per Infrarot programmiert werden. Alle Programmierungen bleiben dauerhaft auch ohne Spannungsversorgung erhalten - beim Sender auch ohne Batterie.

Es ist selbstverständlich, daß eine Programmierung nicht zwingend erforderlich oder nur geringfügig (und einmalig) notwendig ist. Denn die Geräte haben brauchbare Voreinstellungen für alle Parameter. Programmierungen mit dem Sender sind nämlich offensichtlich zwangsläufig unergonomisch.

Der Sender kann seine Parameter- und andere Daten per Infrarot zum Empfänger übertragen, wobei dieser diese Daten zum PC überträgt. Der Sender kann seine Daten aber auch unabhängig vom Empfänger sichtbar machen, durch Blinken mit seiner Signal-LED. Das funktioniert wie die Impulse bei einem Telefon mit Pulswahleinstellung.

Wahrscheinlich kann mit diesem System jedes beliebige Gerät drahtlos und ohne Konflikte fernbedient werden, dessen Bedienungselemente auf dem Prinzip des Tastschaltens beruhen. Beispielsweise eine relais- oder digital-elektronisch gesteuerte Tonbandmaschine. Besonders günstig ist es, wenn ein Gerät für eine Fernbedienung mit Drahtverbindung bereits vorbereitet ist.

In gewissem Umfang ist die Funktion einer Zeitschaltuhr gegeben. Desweiteren sind viele Anwendungen im Elektro- oder Elektronik-Labor möglich, als Pulsgenerator, Timer, Signalschalter, TestIRsender, etc.

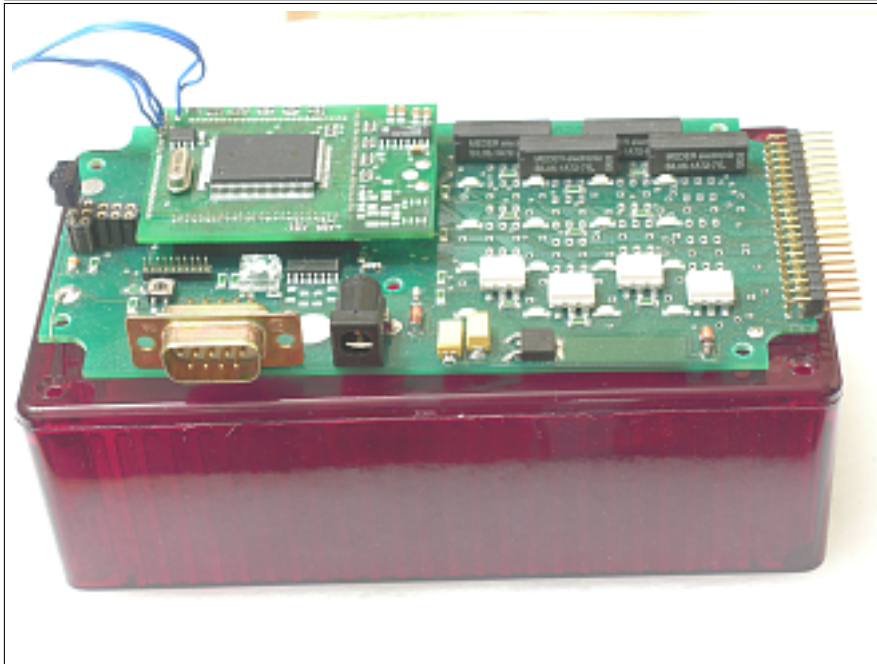
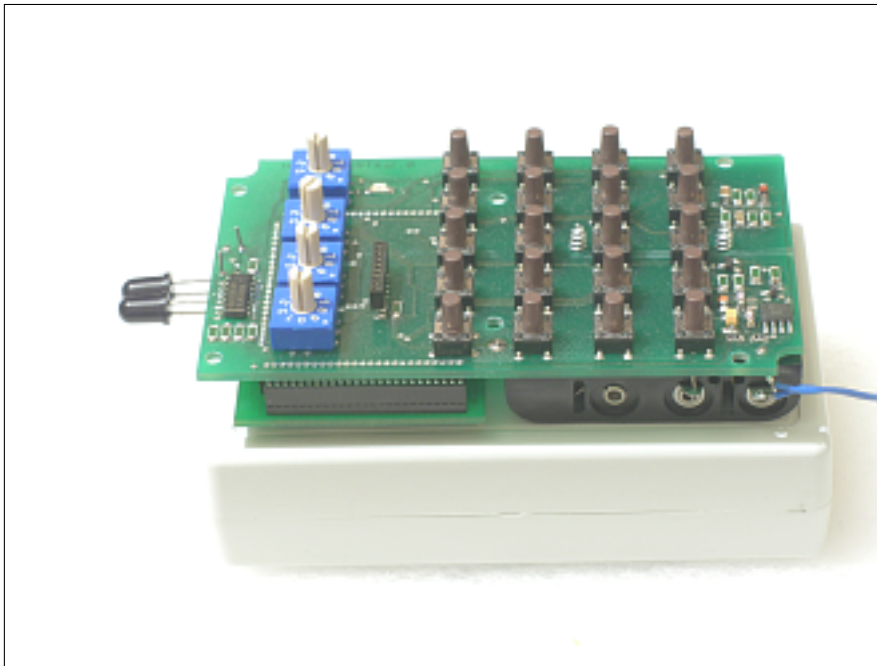
1.1 Merkmale

- Sicheres Infrarot-Protokoll mit CRC-Prüfsumme:
Gültigkeitsprüfung durch Messung der Impuls- und Pausenlängen, Übereinstimmung mit einem festen Identifier, Adresse, CRC und weitere Plausibilitäten. Optionaler Datenblock variabler Länge.
- Geräte-Adressen (000)001-999.
- Broadcast-Adresse 999
- 16 Kanäle pro Empfänger. Folglich bis zu 16000 Kanäle adressenbezogen, zusätzlich mit Ident 4096000 Kanäle.
- Kanal-Schaltkommandos 001-016 und Steuer/Programmier-Kommandos 900-999.
- Schaltmittel: Leistungs-Optokoppler und/oder Reed-Relais. (DC/AC: 60V_s1A)
- Potentialfreie Schließer-Kontakte|MOSFETs.
- Auflösung der Ansteuerung der Ausgänge: 1 Millisekunde.
- Alle Zeiten einstellbar von 0 bis $2^{32} - 1$ ms, also von 0 bis knapp 50 Tage.

- Schaltfunktionen: Anzug- und Abfallverzögerung, Toggle (ElTaKo-Relais), 01-Impulsfolge 1-64 Stufen mit separaten Zeiten für 0 und 1, Öffner/Schließer.
- Programmierbare Matrix 16x16 für Schaltausgang[1-16] <- Kommando[1-16].
- Jede beliebige Kodierung der Schaltausgänge wird durch die 16x16-Matrix ermöglicht. Beispielsweise wenn ein zu steuerndes Gerät nur 4 Kanäle braucht und damit 12 verschiedene Funktionen realisiert.
- Desweiteren sind die Funktionen **Voreilender Kontakt** und **Wechsler** (mit 2 Kanälen) gegeben. Dabei ist unterbrechendes als auch nichtunterbrechendes Schalten einstellbar.
- Ausgänge können mit einer weiteren 16x16-Matrix verriegelt werden.
- Nullsetzen einer noch laufenden Abfallverzögerung (EIN-Verlängerung).
- Mit einem Empfänger können mehrere Geräte ferngesteuert werden. Die universelle Anschlußleiste ohne Wanne und Steckrichtungszwang unterstützt das.
- Zeitschaltuhr: Quarzuhr (RTC) im Empfänger mit Lithium-Zelle.
- Empfänger kann als Relais-Station agieren und Kommandos per Infrarot weiterleiten an andere Empfänger.
- Bis zu drei Adressenbereiche für MultiSenden und MultiEmpfang.
- Betriebsspannung des Empfängers per Steckernetzteil 6-12V und/oder vom zu steuernden Gerät her. Verpolungsschutz; deshalb auch Versorgung mit Wechselspannung prinzipiell möglich.
- Betriebsspannung des Senders mit 4 Micro-Zellen AAA.
- Sender hat 4 Stromaufnahme-Modi: Aus, Standard, Standby, Senden. Zusätzlich kann der treppenförmige Sendestrom 3-stufig manipuliert werden.
- Der Sender hat keinen Ein/Aus-Schalter und nimmt im Zustand **Aus** typisch nur 8 nA (Milliardstel A) Strom auf.
- Der Empfänger hat eine Stromausfallsicherung und rettet längere aktive Verzögerungszeiten in seinen Dauerspeicher, um sie bei Neustart mit Hilfe der Quarzuhr-Zeit zu 'reparieren'.
- Der Empfänger signalisiert durch Flackern der Signal-LED eine Unterspannung.
- Der IR-Empfangsschaltkreis im Empfänger kann in alle 4 Richtungen umgesteckt werden, so daß das Gerätegehäuse nicht gedreht werden muß. Allerdings kann der Empfänger auch reflektierte Signale einwandfrei auswerten, so daß ein Umstecken oft nicht nötig ist.
- Der Empfänger hat ein rot-transparentes Gehäuse. Daher müssen der IR-Empfangsschaltkreis und die 16+1 Signal-LEDs nicht fest in die Gehäusewände montiert werden.

- Beide Geräte sind auch ohne Gehäuse voll funktionsfähig. Die Gehäuse haben keine konstruktive Aufgabe; sie sind nur Hülle.

2 Abbildungen: Sender, Empfänger



Bilder sind vorläufig.

3 Bedienungselemente Sender

Tabelle 1: Drehschalter und Tasten

ADDRESS(100)	ADDRESS(10)	ADDRESS(1)		STATION
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	BLINK	CMD	SEND	STORE

Drehschalter mit Positionen 0-9 ohne Anschlag.

Beispiel für eine Tonbandmaschine:

<<	<	STOP	>	>>
REC	PAUSE	8	9	10

Drehschalter

Adresse (Kommando, Wert)

Mit den drei linken Drehschaltern wird gewöhnlich eine Adresse von 0 bis 999 eingestellt. (Die Empfänger können entsprechend programmiert werden.) Dieser dreistellige Wert kann aber auch eine Kommandonummer 900-999 sein. Desweiteren kann dieser Wert ein zu programmierender Parameterwert sein.

Station

Wenn der Stationsschalter auf 0 steht, wird die Adresse von den drei Adressenschaltern verwendet. Andernfalls wird eine unter 1-9 gespeicherte Adresse verwendet.

3.1 Tasten + Kommandos

Dieser Abschnitt behandelt den Sender, betrifft aber auch den Empfänger, da an den Empfänger gesandte Kommandos im Empfänger die gleiche (korrespondierende) Bedeutung haben.

Tasten CHANNEL1-16

Kanal-Kommando 1-16 wird zum adressierten Empfänger (zu den adressierten Empfängern) gesandt. Bei längerdauerndem Tastendruck beginnt der Sender nach einer programmierbaren Verzögerung, das Kommando wiederholt auszusenden. Nach einer ebenfalls programmierbaren zweiten Verzögerung wird die Senderate noch erhöht.

Taste BLINK

Diese Taste sendet fest eingestellt zum adressierten Empfänger das Kommando 950, welches ein mehrere Sekunden langes Blinken der Signal-LED des Empfängers bewirkt. Diese Funktion kann als *Melde dich!*, *Zeige dich!* beschrieben werden. Adresse=950 -> Station=0 -> Taste STORE -> Adresse=Empf.adr. -> Taste SEND bewirkt das Gleiche.

Durch diese Taste werden auch die StoreFlags gelöscht!

Ein laufendes TestSenden wird abgebrochen.

Taste STORE

Diese Taste ermöglicht es, Zahlenwerte von 000 bis 999 in einen internen RAM-Speicher zu schreiben. Diese Zahlenwerte werden mit den 3 Adressen-Drehschaltern eingestellt. Es sind 10 Speicherplätze vorhanden, gemäß den 10 Positionen 0-9 des Drehschalters STATION. (Speicher[Station]=AAA)

Der erste Speicherplatz[0] ist für eine Kommandonummer 900-999 reserviert. Die weiteren [1-9] können bis zu 9 Zahlenwerte enthalten, die -je nach Kommandonummer in [0]-verschiedenen programmierbaren Parametern zugeordnet sind.

Für jeden der 10 Speicherplätze gibt es ein Flag-Bit, das gesetzt wird bei Betätigung der Taste STORE (StoreFlags). Man muß also nicht von links nach rechts auffüllen bei der Speicherung, sondern kann Lücken lassen.

Nachdem dieser Speicher gefüllt wurde, können mit den Tasten CMD und SEND Aktionen veranlaßt werden, wie Programmierung von Sender oder Empfänger oder Anzeigen von Parameterwerten, und so weiter.

Taste CMD

Bei Verwendung dieser Taste werden die drei Adressenschalter als Kommandoschalter aufgefaßt, für direkte Kommandoangabe: cmd= AAA.

Der Sender sendet quasi mit Adresse 000 an sich selbst: adr= 000.

Tabelle 2: Taste CMD, Funktion RAM

#	1	2	3	4	5	6	7	8	9
900	Stat1	Stat2	Stat3	Stat4	Stat5	Stat6	Stat7	Stat8	Stat9
901	dyStBy	dyOff	dyOffSto	dyRep1	dyRep2	Rate1	Rate2	dySEND	dySend
902	enAdr	Adr1	Adr2	enLed	Ident				
909	Out1	Out2	Out3	Out4	enOut1	enOut2	enOut3		

Tabelle 3: Taste CMD, Funktion DISPLAY

#	1	2	3	4	5	6	7	8	9
940	Stat1	Stat2	Stat3	Stat4	Stat5	Stat6	Stat7	Stat8	Stat9
941	dyStBy	dyOff	dyOffSto	dyRep1	dyRep2	Rate1	Rate2	dySEND	dySend
942	enAdr	Adr1	Adr2	enLed	Ident				
949	Out1	Out2	Out3	Out4	enOut1	enOut2	enOut3		

Tabelle 4: Taste CMD

#	Funktion
950	LedBlink
951	LedTest
952	Vbatt
953	VbattDiff
975	Testsend
997	Standby
998	Reset
999	OFF

RAM Die Parameter im flüchtigen Speicher werden gemäß Store[1-9] gesetzt.

DISPLAY Ein Parameterwert wird angezeigt durch Blinken der Signal-LED. Der Parameter wird ausgewählt gemäß der aktuellen Stellung des Stations-Drehschalters [1-9].

Taste SEND

Bei Verwendung dieser Taste wird die Kommandonummer Speicher 0 entnommen: cmd=Store[0].

Diese Taste hat als einzige eine (einstellbare) Wirkungsverzögerung, damit ungewollte Speicheraktionen nur mit geringer Wahrscheinlichkeit passieren können.

Tabelle 5: Taste SEND, Funktion EEPROM

#	1	2	3	4	5	6	7	8	9
900	Stat1	Stat2	Stat3	Stat4	Stat5	Stat6	Stat7	Stat8	Stat9
901	dyStBy	dyOff	dyOffSto	dyRep1	dyRep2	Rate1	Rate2	dySEND	dySend
902	enAdr	Adr1	Adr2	enLed	Ident				
909	Out1	Out2	Out3	Out4	enOut1	enOut2	enOut3		

Tabelle 6: Taste SEND, Funktion TX->RX_PROG

#	1	2	3	4	5	6	7	8	9
910	i.a1	i.a2	i.ar1	i.ar2	i.at1	i.at2	i.nrx	i.rtcwidth	i.ident
911	.enable	e.flag.breaker	.toggle	.extend	.seq	.rtc	.lock	.tx	.led
919	s.year	s.month	s.day	s.hour	s.minute	s.second	s.weekday		s.baud
920	1-16:	c.delay1	c.delay2	c.seq1	c.seq0	c.seq1	c.nrx		
921	1-16:	e.enable	e.breaker	e.toggle	e.extend	e.seq	e.rtc	e.lock0	
922	1-16:	1-16 (c.rxch)	1-16	1-16	1-16	1-16	1-16	1-16	1-16
922	1-16:	101-116	101-116	101-116	101-116	101-116	101-116	101-116	101-116
923	1-16:	1-16 (c.lock)	1-16	1-16	1-16	1-16	1-16	1-16	1-16
923	1-16:	101-116	101-116	101-116	101-116	101-116	101-116	101-116	101-116
924	1-16:	1-64 (c.seq)	1-64	1-64	1-64	1-64	1-64	1-64	1-64
924	1-16:	101-164	101-164	101-164	101-164	101-164	101-164	101-164	101-164
925	1-16:	c.year	c.month	c.day	c.hour	c.minute	c.second		

EEPROM Programmierung des Senders. Dauerhafte Speicherung plus Speicherung in den flüchtigen Speicher. Werte aus Store[1-9] (Siehe Taste CMD, Funktion RAM). Adresse 000 muß nicht angegeben werden, da die Kommandos 900-909 dem Sender fest zugeordnet sind.

TX->RX_PROG Programmierung des Empfängers, mit dem Sender per Infrarot. Werte aus Store[1-9]. Die Parameternamen wurden teilweise nicht voll ausgeschrieben, da die Tabelle sonst zu lang würde. Der Wert 1-16: gibt den Schaltausgang an. Die Werte 1-64 und 101-164 geben die Bits 1-64 an, wobei bei >100 das jeweilig so adressierte Bit gesetzt wird (1) und anderenfalls gelöscht wird (0).

Tabelle 7: Taste SEND

#	Funktion
940	Parameters -> Receiver -> PC (RS232)
941	Identification -> Receiver -> PC (RS232)
950	LedBlink
951	LedTest
980*	Defaults -> RAM+EEPROM
981*	Defaults -> RAM
982*	Erase EEPROM
983*	Erase Flash-EEPROM
997	Standby
998	Reset
999	OFF

Bei den Kommandos * muß die Adresse 000 eingestellt sein, um diese Aktionen beim Sender zu bewirken.

Die Kommandos 94X senden Datenlisten zum Empfänger zum Zweck ihrer Anzeige, weshalb der Empfänger über die serielle Schnittstelle RS232 dabei mit einem PC verbunden sein sollte und auf dem PC sollte ein Terminalprogramm (wie Hyperterminal) laufen, das mit korrekter Baudrate und Hardware-Handshake rts/cts eingestellt ist. Voreingestellt beim Empfänger ist 57600 baud.

4 Parameter des Senders

Siehe auch die Listen im Anhang (Einheit, Voreinstellung).

Station#	Weitere 9 Schnellzugriff-Adressen. (Durch den Drehschalter ADDRESS(1) sind bereits 9 vorhanden.)
dyStandby	Verzögerung bis zum Eintritt in diesen Modus. Tastendrucke stellen diesen Timer immer wieder auf 0.
dyOff	Verzögerung bis zum völligen Abschalten. Tastendrucke stellen diesen Timer immer wieder auf 0. (Ein Wert von unter 30s wird beim Start unterbunden.)
dyOffStore	Verzögerung bis zum völligen Abschalten. Diese Zeit ist aktiviert, solange StoreFlags gesetzt sind. Tastendrucke stellen diesen Timer immer wieder auf 0. (Ein Wert von unter 60s wird beim Start unterbunden.)
dyRepStart1	Zeit bis der Sender beginnt, Kommandos wiederholt auszusenden. Hinzuaddiert wird stets 0.3s.
dyRepStart2	Zeit bis der Sender beginnt, Kommandos wiederholt auszusenden, in schnellerer Folge. Hinzuaddiert wird stets 1s.
RepRate1	Wiederholgeschwindigkeit. Hinzuaddiert werden stets 120ms.
RepRate2	Wiederholgeschwindigkeit. Hinzuaddiert werden stets 60ms.
dyKeySEND	Wirkungsverzögerung für die Taste SEND. Hinzuaddiert wird stets 0.5s.
dySend	Verzögerung bis zum Senden.
Ident	Identifizierer. Kann als übergeordnete Adresse verstanden werden, die ebenfalls im Empfänger auf Gleichheit geprüft wird.
enAdr	Freigabe des Adressenbereiches zu dessen Benutzung anstelle der Adresse AAA von den Schaltern.
Adr1	Adresse von
Adr2	Adresse bis
enLed	Freigabe der Signal-LED.
Out#	Zeitpunkte für die 3 Treppen des treppenförmigen Sendestromes.
Out4	Zeitpunkt des Endes einer Sendeperiode. Bewirkt mit der Voreinstellung eine Infrarot-Frequenz von exakt 40 kHz. Sollte nicht verändert werden, da der Empfänger einen Bandpaß 40 kHz hat!
enOut#	Freigabe von 3 Ausgängen zur Treppenbildung des Sendestromes. Sperren von Ausgängen spart Sendestrom. Das kann sinnvoll sein bei intensivem Senden und/oder wenn die Reichweite nicht so groß sein muß.

Vbatt1 Gemessene Batteriespannung (Taste CMD).

Vbatt2 Gemessene Batteriespannung (Taste CMD). Zweiter Wert zur Differenzbildung.

5 Parameter des Empfängers

Siehe auch die Listen im Anhang (Einheit, Voreinstellung).

s.year	Quarzuhr (RTC=RealTimeClock)
s.month	RTC
s.day	RTC
s.hour	RTC
s.minute	RTC
s.second	RTC
s.s1970	RTC
s.weekday	RTC
s.baud	Baudrate für die serielle Schnittstelle.
i.ident	Identifizierer. Kann als übergeordnete Adresse verstanden werden, die ebenfalls auf Gleichheit geprüft wird.
i.a1	Adresse von
i.a2	Adresse bis Der Empfänger reagiert auf diesen Adressenbereich. Aber auch auf eine einzelne Adresse von.
i.ar1	Adresse von
i.ar2	Adresse bis Der Empfänger reagiert auf diesen Adressenbereich zum Zwecke der Weiterleitung an einen anderen Empfänger.
i.at1	Adresse von
i.at2	Adresse bis Dies sind die korrespondierenden (anderen) Adressen, die der Empfänger sendet bei Weiterleitung.
i.nrx	Die Zeit, die maximal zwischen zwei Empfangsereignissen vergehen darf bei aktivierter Sicherheitsfunktion, bei der 1 Empfangsereignis nicht ausreicht.
i.rtcwidth	Nachlaufzeit, während der ein Schaltbegehren der Zeitschaltuhr noch erkannt wird, wenn der Gleichheitszeitpunkt durch irgendwelche Störungen verpaßt wurde.
e.flag	Parameter mit globalen Freischalt-Bits:

e.flag.enable	Schaltfreigabe
e.flag.breaker	Öffner-Funktion
e.flag.toggle	EITaKo-Funktion
e.flag.extend	Verlängerung Ein-Zustand
e.flag.seq	Schaltsequenz
e.flag.rtc	Zeitschaltuhr
e.flag.lock	Ausgangssperre
e.flag.tx	Weiterleitung
e.flag.led	Signal-LED
e.flag.safe	Abspeicherung bei PowerFailure/PowerDown. Sollte 0 sein, wenn Unter- spannungsdetektion benutzt wird.
e#.enable	Schaltfreigabe (Einzelfreigabe)
e#.breaker	Öffner-Funktion (Einzelfreigabe)
e#.toggle	EITaKo-Funktion (Einzelfreigabe)
e#.extend	Verlängerung Ein-Zustand (Timer_abfvz=0) (Einzelfreigabe)
e#.seq	Schaltsequenz (Einzelfreigabe)
e#.rtc	Zeitschaltuhr (Einzelfreigabe)
e#.lock0	Ausgangssperre_bei_0
c#.rxch	Schaltverknüpfung 16x16-Matrix
c#.lock	Ausgangssperre/Verriegelung 16x16-Matrix: Wenn der zugehörige Ausgang (nicht) aktiv ist, werden alle Ausgänge, die hierin auf 1 gesetzt sind, inaktiviert. Es ist zu beachten, daß Öffner- Invertierungen logisch später, ganz zum Schluß erfolgen.
c#.delay1	Anzugverzögerung
c#.delay2	Abfallverzögerung
c#.seq	01-Impulsfolge/Sequenz, 64-stufig
c#.seql	Länge der Sequenz. Nach Abarbeitung dieser Länge wird die Sequenz je- weils immer wieder von vorne begonnen. Sequenzlaufzeit während des Ablaufens der Abfallverzögerung. Sequenzlänge wird bei Programmie- rung per Terminalprogramm automatisch gesetzt bei Eingabe der Impuls- folge und kann bei Bedarf nachträglich abgeändert werden.

- c#.seq0 Zeitdauer für ein 0-SequenzBit.
- c#.seq1 Zeitdauer für ein 1-SequenzBit.
- c#.rtc Kanal-Zeitschaltuhrzeit (Schaltbegehren)
- c#.nrx Sicherheitsfunktion. Gibt die Anzahl der erforderlichen Empfangsereignisse an.

6 Datenblatt des Senders

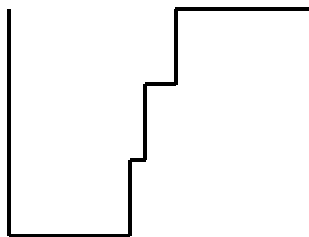
Schutzart	IP40	Kunststoffgehäuse
Drehschalter	4	Dez.0-9, Miniatur
Kurzhub-Taster	20	1.6N
Signal LED	1	rot
Spannungsversorgung	4 Micro-Zellen AAA	1.5V, 800-1500mAh
Umgeb.temperatur	-20..+40°C	
Betriebsspannung	5.0-6.3 V (typ.)	
Microcontroller	Fujitsu F347 16Bit	24 MHz, $T_a \leq 105^\circ\text{C}$
Stromaufnahme Aus	8 nA (typ.)	
Stromaufnahme Standby	2.3 mA (typ.)	16 kHz
Stromaufnahme Normal	8.5 mA (typ.)	2 MHz
Stromaufnahme Senden	55 mA (typ.)	8 MHz
IR Sendediode	1-2	Wellenlänge 950 nm
IR Trägerfrequenz	40 kHz	
Sendedauer	51 ms	1 Kanal-Kommando

Batterien

Die Batterien (1.5V) können nicht bis zur völligen Entleerung (0.9V) verwendet werden. Sie sollten bei etwa $4 \times 1.25\text{V} = 5\text{V}$ in einem anderen Gerät weiterbenutzt werden, das mit einem speziell für Batteriebetrieb geeigneten Schaltkreis arbeitet. Das ist wegen sinkendem Sendestrom und insbesondere dem Spannungsbedarf des Microcontrollers nicht zu vermeiden.

Ausgänge

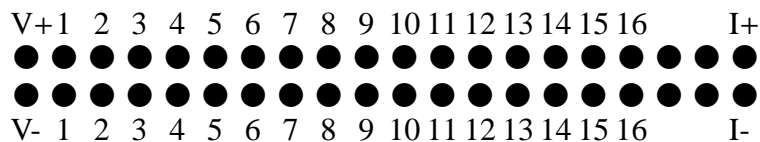
Der Sender hat 3 Ausgänge, die nacheinander einen jeweils weiteren Strom auf die Infrarot-Sendediode schalten. Diese Ausgänge können unabhängig voneinander abgeschaltet werden. Desweiteren können deren Zeitpunkte innerhalb der Sendeperiode eingestellt werden. Die Länge der Sendeperiode selbst kann ebenfalls eingestellt werden, was jedoch die Sendefrequenz ändert.



7 Datenblatt des Empfängers

Schutzart	IP40	Kunstst.geh., rot-transparent
NV-Buchse	5.8/2.1 mm	f. Steckernetzteil
Schnittstelle	DSub Stiftbuchse 9-pol.	RS232, f. Nullmodem-Kabel
Stiftleiste	2x20-pol., RM2.54	vergoldet
Kanal LED	4-16	rot
Signal LED	1	rot, SuperFlux-Typ
Spannungsversorgung	Steckernetzteil über Stiftleiste Verpolungsschutz	6-12V, $\geq 300mA$ 6-12V, $\geq 300mA$
Umgeb.temperatur	-20..+40°C	
Betriebsspannung	7-15V	$V_m \geq 7V$
Microcontroller	Fujitsu F347 16Bit	24 MHz, $T_a \leq 105^\circ C$
Schaltausgänge	16 Kanäle je 1 Kontakt	40V/1A AC/DC (Kleinspannung) Schließer, potentialfrei
Schaltmittel	4-16 4-16	Power-Optokoppler 60V/1A AC/DC Reed-Relais 200V/1A/15VA AC/DC
Schaltauflösung	$1 \pm 0.00005 \text{ ms}$ (typ.)	(vom Controller her)
Schaltverzögerung	0.3-2.5 ms	je nach Last und Schaltrichtung
IR Empfänger	950 nm, Carrier 40 kHz	im Geräte-Gehäuse
Quarzuhr	1	Lithium-Zelle CR2032, 3V

Belegung der Stiftleiste



V+ V- Spannungsversorgung²
 I+ I- Infrarot-Sendediode (Weiterleitung)
 1-16 Schaltausgänge der 16 Kanäle (je 1 Schließer)

Die Stiftleiste ist keinesfalls für Netzspannungen geeignet! Wer mehr als 40V (Schutzkleinspannung) verwendet, muß einen Berührungsschutz anbringen. Die Spannung zwischen den Anschlußpins ein und desselben Kanals ist wegen der verwendeten Schaltmittel begrenzt. Auch wenn die Reed-Relais isoliert betrachtet für wesentlich mehr als 60V geeignet sind, sollten 60V nicht überschritten werden. Die Spannung zwischen verschiedenen Kanälen soll ebenso 60V nicht überschreiten. Das kann vorkommen, wenn in ein und demselben zu steuernden Gerät auf verschiedene Potentiale zugegriffen wird. Obwohl die Optokoppler als auch die Relais eine Isolationsspannung von 4000V besitzen, so begrenzt der enge Aufbau im Empfänger die maximal mögliche Spannung.

Infrarot-Empfangsschaltkreis

Dieser Schaltkreis ist das Teil mit einem schwarz eingefärbten Gehäuse und 3 Anschlußpins, das eine Kalotte (Kugelabschnitt) für den Lichteintritt hat. Der Lichteintritt kann in alle 4 Richtungen zeigen. Dazu kann dieser Schaltkreis umgesteckt werden. Für diese Möglichkeit sind 4 Buchsenleisten mit je 3 Buchsen für die 3 Anschlußpins vorhanden.

Dabei ist unbedingt zu beachten, daß dieser Schaltkreis so eingesteckt wird, daß seine Kalotte in Kalottenrichtung möglichst nahe an einer Platinenkante ist. Also erst in die gewünschte Richtung drehen, dann in Kalottenrichtung bewegen, bis die letztmögliche Steckposition erreicht ist.

Spannungsversorgung

Der Momentanwert der Spannung für den Empfänger soll zu keinem Zeitpunkt unter 7V abfallen. Eine Spannung mit einem Mittelwert von etwa 7.5V ist ideal.

Der Empfänger zeigt Unterspannung durch Flackern seiner Signal-LED an. Er mißt dazu seine eigene Versorgungsspannung nach den Verpolungsschutzdioden, vor seinem internen Spannungsregler, jede Millisekunde. Wegen der beiden Verpolungsschutzdioden können auch beide Spannungsversorgungsanschlüsse gleichzeitig in Betrieb sein. Die höhere Momentanspannung gewinnt.

Bei Steckernetzteilen ist zu beachten, daß der darin befindliche Transformator einen Leerlaufspannungsfaktor hat (bis etwa 1.4) und die Nennspannung sich nur bei Nennstrom ergibt. Das bedeutet, daß manche 6V-Netzteile mit beispielsweise 4VA Leistung (650 mA) unter der Last des Empfängers (<300mA) dennoch eine genügend hohe Spannung abgeben.

Datensicherung bei PowerFail

Der Umgang mit dieser Funktion ist nicht einfach, da einige Voraussetzungen notwendig und viele Abhängigkeiten vorhanden sind. Die korrekte Funktion kann beim Start geprüft werden, wenn dabei der Empfänger mit dem PC und einem Terminalprogramm verbunden ist.

Interne Parameter mit Verzögerungszeiten über 30s werden abgespeichert. Kanäle, deren Ausgänge zu diesem Zeitpunkt geschaltet sind, werden nicht berücksichtigt, da deren Funktion durch den Zusammenbruch der Betriebsspannung ohnehin zerstört ist. Insbesondere lange Anzugverzögerungen sind hier sinnvoll zu speichern.

Beim Start wird geprüft, ob es eine Sicherung im Dauerspeicher gibt. Falls ja, werden diese Werte mit Hilfe der Quarzuhr, die ja wegen ihrer Lithium-Batterie unabhängig ist, korrigiert. Dazu dient die Differenz zwischen dem Zeitpunkt der Datensicherung und dem Zeitpunkt des aktuellen Starts des Gerätes. Die Quarzuhr hat allerdings keine Auflösung von 1ms, sondern eine von 1s, was aber bei Zeiten von über 30s kaum noch eine Rolle spielt. Wenn nun der Ausfall der Betriebsspannung so lang war, daß ein Schaltereignis während des Ausfalls hätte passieren müssen, kann der entsprechende Kanal nicht korrigiert werden, da das Ereignis schlicht verpaßt wurde.

Diese Einrichtung ist schwerpunktmäßig für Fälle gedacht, wo eine lange Anzugverzögerung eingestellt wurde, die in beispielsweise 4 Tagen einen Kanal schalten soll. Ohne diese Datensicherung wäre bei einem Ausfall der Betriebsspannung dieses laufende Kommando sofort gelöscht.

Wenn die Betriebsspannung am Empfänger unterbrochen wird, besteht keine Chance einer Datensicherung, da die Haltezeit des Empfängers mit etwa 2ms viel zu kurz ist. Jedoch wenn ein geeignetes Steckernetzteil aus der Steckdose gezogen wird, kann die Haltezeit bei 300ms und mehr liegen, denn hierbei kommt die höhere Haltezeit des Netzteils zum Tragen. Ein geeignetes Steckernetzteil liegt wahrscheinlich vor, wenn es sich um ein stabilisiertes handelt, mit beispielsweise 17VA Leistung und Ausgangsspannungen 3/4.5/6/7.5/9/12 V, 650mA max. Bei unstabilierten sollte eher 9V statt kleinere Spannungen gewählt werden, weil dann die Spannungsreserve des Ladekondensators höher ist.

8 Datenblatt Sender+Empfänger

Broadcast	999	(Empfangsadresse)
Reichweite	~ 15m	maximal
	~ 10m	praktikabel
Lötzinn	bleifrei (überwiegend)	Ab Mitte 2006 EU-Vorschrift

Reichweite

Die Reichweite wurde unter günstigen Bedingungen, bei geringem Fremdlicht, durch eine leicht verschmutzte Glasscheibe hindurch getestet. Bei maximaler Entfernung muß ziemlich genau gezielt werden, so daß diese Entfernung nicht wirklich praktikabel ist.

Generell ist zu bemerken, daß der Sendestrom durch die IR-Sendediodeen vergleichsweise sehr gering ist. Mit maximal zulässigem Strom könnte die Reichweite erheblich gesteigert werden. Darauf wurde verzichtet, wegen des dann höheren Batterieverbrauches und insbesondere wegen der Einstellbarkeit der 3 Sendeausgänge.

A Datenlisten des Senders

Parameter

{10}	10 []	Station1
{11}	11 []	Station2
{12}	12 []	Station3
{13}	13 []	Station4
{14}	14 []	Station5
{15}	15 []	Station6
{16}	16 []	Station7
{17}	17 []	Station8
{18}	18 []	Station9
{13}	13 [s]	dyStandby
{6}	6 [10s]	dyOff
{18}	18 [10s]	dyOffStore
{40}	40 [10ms]	dyRepStart1
{150}	150 [10ms]	dyRepStart2
{12}	12 [10ms]	RepRate1
{6}	6 [10ms]	RepRate2
{100}	100 [10ms]	dyKeySEND
{0}	0 [s]	dySend
{150}	150 []	Ident
{0}	0 []	Adr1
{0}	0 []	Adr2
{0}	0 []	enAdr
{3}	3 []	enLed
{80}	80 []	Out1
{90}	90 []	Out2
{110}	110 []	Out3
{200}	200 []	Out4
{1}	1 []	enOut1
{1}	1 []	enOut2
{1}	1 []	enOut3
{0}	634 [10mV]	Vbatt1
{0}	634 [10mV]	Vbatt2

Identifikation

H.O.B.S.(\\)IRTXa
Version 1.03_30
Nov 7 2004 02:34:39
Copyright © H. Schellong
Tiefer Grund 12
32108 Bad Salzungen

B Kommando- und Parameterlisten des Empfängers

Kommandos

```
32] cu -l /dev/cuaa0 -57600
Connected.

>?
cmd      command list with text
.        output/format
a[.b[c[d]...]]  output select
1d2h34m56s789 [ms]  altern. format
>cmd
echo     Unit sends back received characters (toggle)
rtc      Display data of Real Time Clock
ver      Version and copyright info
max      Display maximum [ms] value
help     Help text
?        Help text
cmd      Command list
led      Led ON/OFF (toggle)
volt     Display voltage value(s) on regulator input
1ram     Nonvolatile Store1 --> RAM
2ram     Nonvolatile Store2 --> RAM
exch     Exchanges data Store1 <-> Store2
dflt1    Fills RAM and Store1 with default values
dflt2    Fills RAM and Store2 with default values
erase    Erases NV memory all, restart with defaults
>rtc
Tu, 2004.11.09 05:34:22 1099978462
>volt
7.08 V      7.05 Vmin  7.10 Vmax
>max
4294967295 [ms]
49d17h2m47s295ms
1193h2m47s295ms
>ver
H.O.B.S.(\\)IRRXa1.01_71
Nov 9 2004 04:13:55
Copyright (c) 2004
Helmut Schellong, Bad Salzflfen
```

Parameter

```
>.
s.year=2004  ccyy|yy
```

s.month=11
s.day=9
s.hour=5
s.minute=40
s.second=20
s.s1970=1099978820 Seconds since 1970.Jan.1. 00:00:00
s.weekday=2 1(mo)-7
s.baud=576 [100] 48,96,192,384,576,1152
i.ident=150 read only
i.a1=1 Address (from)
i.a2=1 Address to
i.ar1=101 AddressRX from
i.ar2=101 AddressRX to
i.at1=201 AddressTX from
i.at2=201 AddressTX to
i.nrxt=500 [ms] Time between RX events
i.rtcwidth=3 [s] RTC Registration width
e.flag.enable=1 Global enable
e.flag.breaker=1 Global enable
e.flag.toggle=1 Global enable
e.flag.extend=1 Global enable
e.flag.seq=1 Global enable
e.flag.rtc=1 Global enable
e.flag.lock=0 Global enable
e.flag.tx=0 Global enable
e.flag.led=1 Global enable
e.flag.safe=0 Global enable
e1.enable=1 Single enable
e2.enable=1 Single enable
e3.enable=1 Single enable
e4.enable=1 Single enable
e5.enable=1 Single enable
e6.enable=1 Single enable
e7.enable=1 Single enable
e8.enable=1 Single enable
e9.enable=1 Single enable
e10.enable=1 Single enable
e11.enable=1 Single enable
e12.enable=1 Single enable
e13.enable=1 Single enable
e14.enable=1 Single enable
e15.enable=1 Single enable
e16.enable=1 Single enable
e1.breaker=0 Single enable
e2.breaker=0 Single enable
e3.breaker=0 Single enable

e4.breaker=1 Single enable
e5.breaker=0 Single enable
e6.breaker=0 Single enable
e7.breaker=0 Single enable
e8.breaker=0 Single enable
e9.breaker=0 Single enable
e10.breaker=0 Single enable
e11.breaker=0 Single enable
e12.breaker=0 Single enable
e13.breaker=0 Single enable
e14.breaker=0 Single enable
e15.breaker=0 Single enable
e16.breaker=0 Single enable
e1.toggle=0 Single enable
e2.toggle=0 Single enable
e3.toggle=1 Single enable
e4.toggle=0 Single enable
e5.toggle=0 Single enable
e6.toggle=0 Single enable
e7.toggle=0 Single enable
e8.toggle=0 Single enable
e9.toggle=0 Single enable
e10.toggle=0 Single enable
e11.toggle=0 Single enable
e12.toggle=0 Single enable
e13.toggle=0 Single enable
e14.toggle=0 Single enable
e15.toggle=0 Single enable
e16.toggle=0 Single enable
e1.extend=0 Single enable
e2.extend=0 Single enable
e3.extend=0 Single enable
e4.extend=0 Single enable
e5.extend=0 Single enable
e6.extend=0 Single enable
e7.extend=0 Single enable
e8.extend=0 Single enable
e9.extend=0 Single enable
e10.extend=0 Single enable
e11.extend=0 Single enable
e12.extend=0 Single enable
e13.extend=0 Single enable
e14.extend=0 Single enable
e15.extend=0 Single enable
e16.extend=0 Single enable
e1.seq=0 Single enable

e2.seq=1 Single enable
e3.seq=0 Single enable
e4.seq=0 Single enable
e5.seq=0 Single enable
e6.seq=0 Single enable
e7.seq=0 Single enable
e8.seq=0 Single enable
e9.seq=0 Single enable
e10.seq=0 Single enable
e11.seq=0 Single enable
e12.seq=0 Single enable
e13.seq=0 Single enable
e14.seq=0 Single enable
e15.seq=0 Single enable
e16.seq=0 Single enable
e1.rtc=0 Single enable
e2.rtc=0 Single enable
e3.rtc=0 Single enable
e4.rtc=0 Single enable
e5.rtc=0 Single enable
e6.rtc=0 Single enable
e7.rtc=0 Single enable
e8.rtc=0 Single enable
e9.rtc=0 Single enable
e10.rtc=0 Single enable
e11.rtc=0 Single enable
e12.rtc=0 Single enable
e13.rtc=0 Single enable
e14.rtc=0 Single enable
e15.rtc=0 Single enable
e16.rtc=0 Single enable
e1.lock0=0 Lockup reverse
e2.lock0=0 Lockup reverse
e3.lock0=0 Lockup reverse
e4.lock0=0 Lockup reverse
e5.lock0=0 Lockup reverse
e6.lock0=0 Lockup reverse
e7.lock0=0 Lockup reverse
e8.lock0=0 Lockup reverse
e9.lock0=0 Lockup reverse
e10.lock0=0 Lockup reverse
e11.lock0=0 Lockup reverse
e12.lock0=0 Lockup reverse
e13.lock0=0 Lockup reverse
e14.lock0=0 Lockup reverse
e15.lock0=0 Lockup reverse

```

e16.lock0=0    Lockup reverse

c1.rxch=*.*****    hwChannel <- rxChannel
c1.lock=*****    Lockup outputs
c1.delay1=0 [ms]    Start delay
c1.delay2=20 [ms]   Stop delay
c1.seq=*.*****.*****.....*****.....*
c1.seql=64 [1]     Length of sequence
c1.seq0=100 [ms]   Duration of 0bit
c1.seq1=100 [ms]   Duration of 1bit
c1.rtc=1970.01.01.00.00.00  Start time RealTimeClock
c1.nrx=0 [1]       Necessary RX events
c2.rxch=*.*****    hwChannel <- rxChannel
c2.lock=*****    Lockup outputs
c2.delay1=0 [ms]   Start delay
c2.delay2=6400 [ms] Stop delay
c2.seq=*.*****.*****.....*****.....*
c2.seql=64 [1]     Length of sequence
c2.seq0=100 [ms]   Duration of 0bit
c2.seq1=100 [ms]   Duration of 1bit
c2.rtc=1970.01.01.00.00.00  Start time RealTimeClock
c2.nrx=0 [1]       Necessary RX events
c3.rxch=...*.*****    hwChannel <- rxChannel
c3.lock=*****    Lockup outputs
c3.delay1=0 [ms]   Start delay
c3.delay2=20 [ms]  Stop delay
c3.seq=*.*****.*****.....*****.....*
c3.seql=64 [1]     Length of sequence
c3.seq0=100 [ms]   Duration of 0bit
c3.seq1=100 [ms]   Duration of 1bit
c3.rtc=1970.01.01.00.00.00  Start time RealTimeClock
c3.nrx=0 [1]       Necessary RX events
c4.rxch=...*.*****    hwChannel <- rxChannel
c4.lock=*****    Lockup outputs
c4.delay1=0 [ms]   Start delay
c4.delay2=20 [ms]  Stop delay
c4.seq=*.*****.*****.....*****.....*
c4.seql=64 [1]     Length of sequence
c4.seq0=100 [ms]   Duration of 0bit
c4.seq1=100 [ms]   Duration of 1bit
c4.rtc=1970.01.01.00.00.00  Start time RealTimeClock
c4.nrx=0 [1]       Necessary RX events
c5.rxch=...*.*****    hwChannel <- rxChannel
c5.lock=*****    Lockup outputs
c5.delay1=0 [ms]   Start delay
c5.delay2=20 [ms]  Stop delay
c5.seq=*.*****.*****.....*****.....*
c5.seql=64 [1]     Length of sequence
c5.seq0=100 [ms]   Duration of 0bit
c5.seq1=100 [ms]   Duration of 1bit

```

```

c5.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c5.nrx=0 [1] Necessary RX events
c6.rxch=.....*..... hwChannel <- rxChannel
c6.lock=..... Lockup outputs
c6.delay1=0 [ms] Start delay
c6.delay2=20 [ms] Stop delay
c6.seq=*.***.*.*.*.*.*.*.***.****.....*****.....*****.....*.
c6.seql=64 [1] Length of sequence
c6.seq0=100 [ms] Duration of 0bit
c6.seq1=100 [ms] Duration of 1bit
c6.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c6.nrx=0 [1] Necessary RX events
c7.rxch=.....*..... hwChannel <- rxChannel
c7.lock=..... Lockup outputs
c7.delay1=0 [ms] Start delay
c7.delay2=20 [ms] Stop delay
c7.seq=*.***.*.*.*.*.*.*.***.****.....*****.....*****.....*.
c7.seql=64 [1] Length of sequence
c7.seq0=100 [ms] Duration of 0bit
c7.seq1=100 [ms] Duration of 1bit
c7.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c7.nrx=0 [1] Necessary RX events
c8.rxch=.....*..... hwChannel <- rxChannel
c8.lock=..... Lockup outputs
c8.delay1=0 [ms] Start delay
c8.delay2=20 [ms] Stop delay
c8.seq=*.***.*.*.*.*.*.*.***.****.....*****.....*****.....*.
c8.seql=64 [1] Length of sequence
c8.seq0=100 [ms] Duration of 0bit
c8.seq1=100 [ms] Duration of 1bit
c8.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c8.nrx=0 [1] Necessary RX events
c9.rxch=.....*..... hwChannel <- rxChannel
c9.lock=..... Lockup outputs
c9.delay1=0 [ms] Start delay
c9.delay2=20 [ms] Stop delay
c9.seq=*.***.*.*.*.*.*.*.***.****.....*****.....*****.....*.
c9.seql=64 [1] Length of sequence
c9.seq0=100 [ms] Duration of 0bit
c9.seq1=100 [ms] Duration of 1bit
c9.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c9.nrx=0 [1] Necessary RX events
c10.rxch=.....*..... hwChannel <- rxChannel
c10.lock=..... Lockup outputs
c10.delay1=0 [ms] Start delay
c10.delay2=20 [ms] Stop delay
c10.seq=*.***.*.*.*.*.*.*.***.****.....*****.....*****.....*.
c10.seql=64 [1] Length of sequence
c10.seq0=100 [ms] Duration of 0bit

```

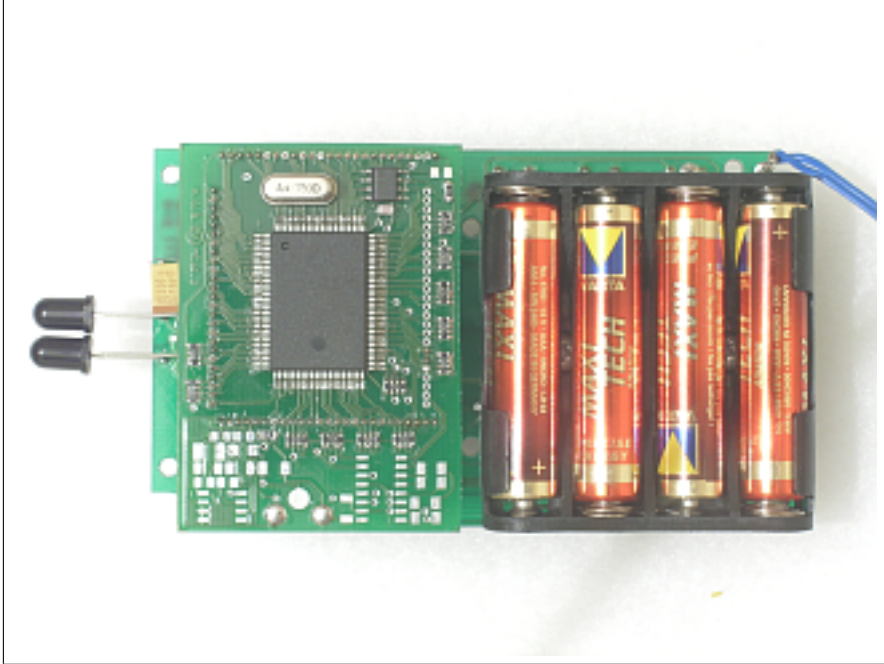
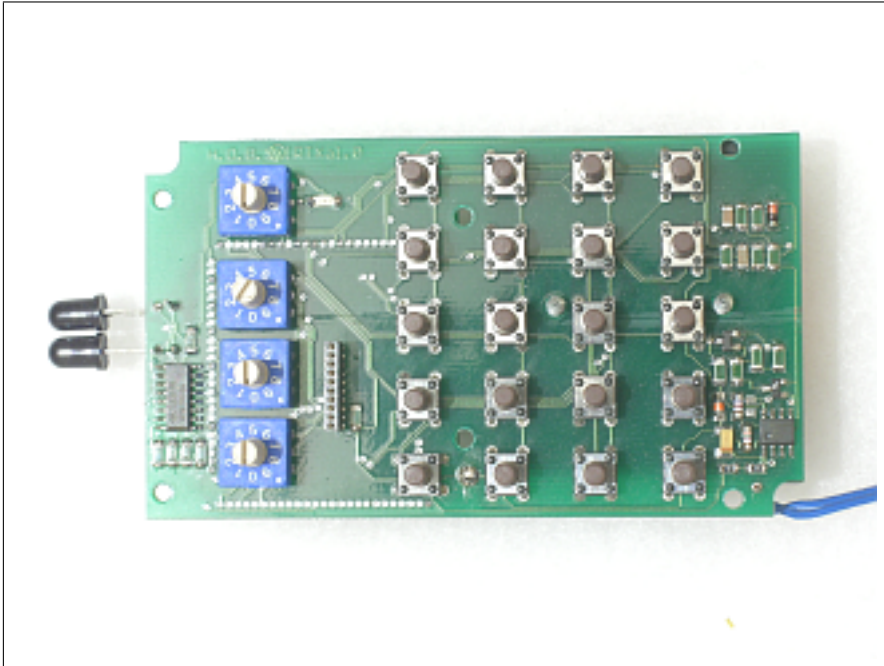
```
c10.seq1=100 [ms] Duration of 1bit
c10.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c10.nrx=0 [1] Neccessary RX events
c11.rxch=.....*..... hwChannel <- rxChannel
c11.lock=..... Lockup outputs
c11.delay1=0 [ms] Start delay
c11.delay2=20 [ms] Stop delay
c11.seq=*.~*.~*.~*.~*.~*.~*.~*.~*..*****.....*****.*.~*..
c11.seq1=64 [1] Length of sequence
c11.seq0=100 [ms] Duration of 0bit
c11.seq1=100 [ms] Duration of 1bit
c11.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c11.nrx=0 [1] Neccessary RX events
c12.rxch=.....*..... hwChannel <- rxChannel
c12.lock=..... Lockup outputs
c12.delay1=0 [ms] Start delay
c12.delay2=20 [ms] Stop delay
c12.seq=*.~*.~*.~*.~*.~*.~*.~*.~*..*****.....*****.*.~*..
c12.seq1=64 [1] Length of sequence
c12.seq0=100 [ms] Duration of 0bit
c12.seq1=100 [ms] Duration of 1bit
c12.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c12.nrx=0 [1] Neccessary RX events
c13.rxch=.....*... hwChannel <- rxChannel
c13.lock=..... Lockup outputs
c13.delay1=0 [ms] Start delay
c13.delay2=20 [ms] Stop delay
c13.seq=*.~*.~*.~*.~*.~*.~*.~*.~*..*****.....*****.*.~*..
c13.seq1=64 [1] Length of sequence
c13.seq0=100 [ms] Duration of 0bit
c13.seq1=100 [ms] Duration of 1bit
c13.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c13.nrx=0 [1] Neccessary RX events
c14.rxch=.....*.. hwChannel <- rxChannel
c14.lock=..... Lockup outputs
c14.delay1=0 [ms] Start delay
c14.delay2=20 [ms] Stop delay
c14.seq=*.~*.~*.~*.~*.~*.~*.~*.~*..*****.....*****.*.~*..
c14.seq1=64 [1] Length of sequence
c14.seq0=100 [ms] Duration of 0bit
c14.seq1=100 [ms] Duration of 1bit
c14.rtc=1970.01.01.00.00.00 Start time RealTimeClock
c14.nrx=0 [1] Neccessary RX events
c15.rxch=.....*. hwChannel <- rxChannel
c15.lock=..... Lockup outputs
c15.delay1=0 [ms] Start delay
c15.delay2=20 [ms] Stop delay
c15.seq=*.~*.~*.~*.~*.~*.~*.~*.~*..*****.....*****.*.~*..
c15.seq1=64 [1] Length of sequence
```


Parameter (Selektionsbeispiel)

```
>c.rx
c1.rxch=*..... hwChannel <- rxChannel
c2.rxch=.*..... hwChannel <- rxChannel
c3.rxch=..*..... hwChannel <- rxChannel
c4.rxch=...*..... hwChannel <- rxChannel
c5.rxch=....*..... hwChannel <- rxChannel
c6.rxch=.....*..... hwChannel <- rxChannel
c7.rxch=.....*..... hwChannel <- rxChannel
c8.rxch=.....*..... hwChannel <- rxChannel
c9.rxch=.....*..... hwChannel <- rxChannel
c10.rxch=.....*..... hwChannel <- rxChannel
c11.rxch=.....*..... hwChannel <- rxChannel
c12.rxch=.....*..... hwChannel <- rxChannel
c13.rxch=.....*..... hwChannel <- rxChannel
c14.rxch=.....*..... hwChannel <- rxChannel
c15.rxch=.....*..... hwChannel <- rxChannel
c16.rxch=.....*..... hwChannel <- rxChannel
>
```

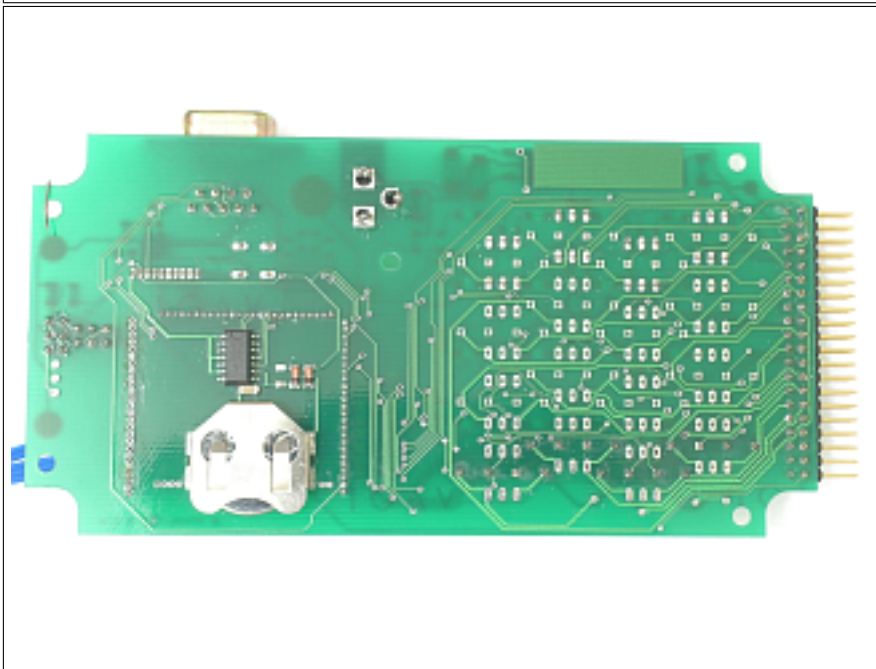
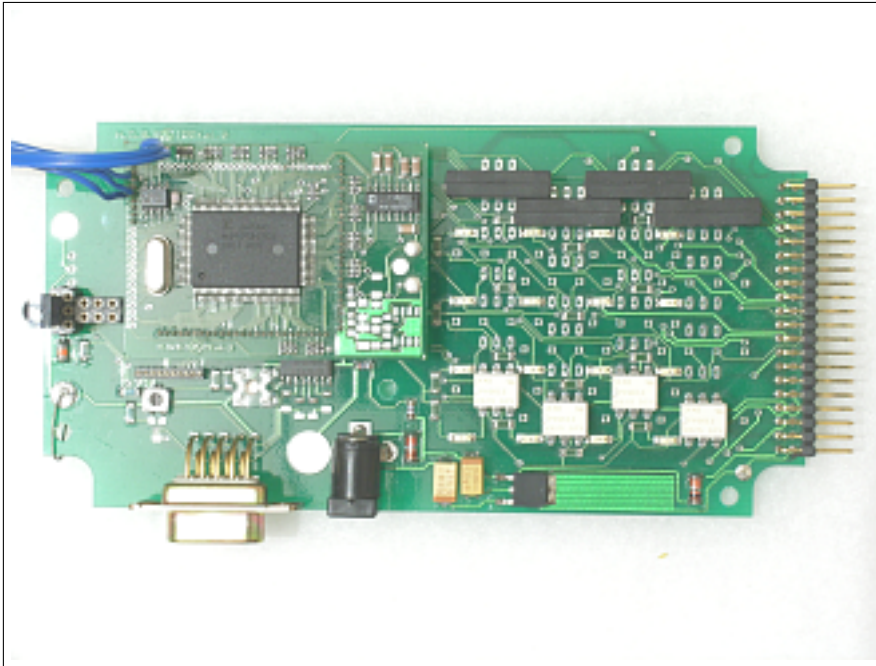
C Weitere Abbildungen

C.1 Sender



Bilder sind vorläufig.

C.2 Empfänger



Bilder sind vorläufig.

Copyright © 2004 Helmut Schellong, Bad Salzufen
www.schellong.de

L^AT_EX 2_ε