

# Funktransceiver

## TRX433-10-C2/12-C2

## TRX868-10-C2/12-C2

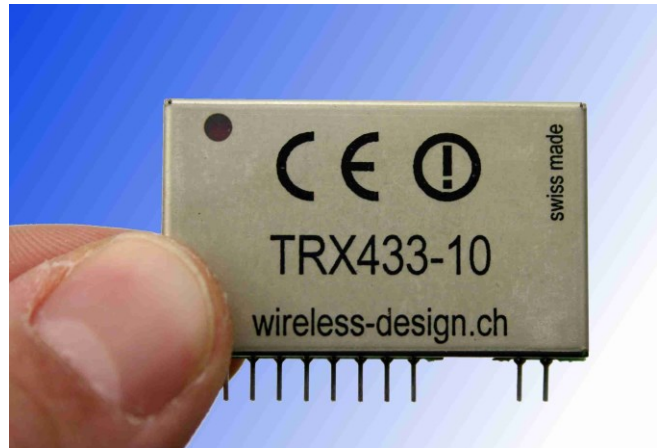


### Versionenliste

Datum	Version	Beschreibung
03.03.2009	0.2	Entwurf
14.04.2009	1.0	Überarbeitet zur Freigabe

## Inhaltsverzeichnis

Kurzbeschreibung .....	3
Eigenschaften .....	3
Anwendungen .....	3
Familienübersicht Funktransceiver .....	4
Dimensions .....	5
Pin description.....	5
Allgemeine Informationen zur Funktionalität.....	6
Serielle Schnittstelle (RS232).....	6
Datenmodus (MODE Pin = Low) .....	6
Konfigurationsmodus (MODE Pin = High) .....	6
Funkübertragung .....	6
Frequenzwahl .....	6
Funkdatenrate.....	6
Aufbau Funktelegramm .....	7
Sende- und Empfangsbuffer.....	7
Sendebuffer (TX-BUF) .....	7
Empfangsbuffer (RX-BUF) .....	7
Bufferüberlauf .....	8
Handshake mit RTS/CTS .....	8
Funktion RTS .....	8
Funktion CTS .....	8
CRC16 .....	9
CRC16 aus.....	9
CRC16 ein.....	9
Fastsend .....	9
Fastsend aus.....	9
Fastsend ein.....	9
Datenflussbeispiele.....	10
Beispiele für verschiedene Einstellungen von Fastsend und CRC16.....	10
Beispiel für kontinuierlichen Funkdatenstrom (Funktelegramme mit mehr als 61 Bytes Nutzdaten).....	12
Beispiel für kontinuierlichen Datenstrom über RS232 .....	12
RSSI des Telegramms und Ruhe-RSSI .....	13
Rekalibration.....	14
Stromsparmodi und Verhalten bei Reset, Powerup, Wakeup und sleep .....	14
RSSI (Received Signal Strength Indicator) .....	17
Speisespannung.....	18
Dateninterface mit 5V Systemen.....	18
MCLR\ Reset .....	18
Status LED .....	18
Temperatursensor .....	19
Technische Daten TRX433-10C .....	20
Technische Daten TRX868-10C .....	21
Funkreichweite .....	22
Vereinfachtes Schema TRXnnn-10 .....	23
Anwendungsbeispiel für TRXnnn-10C.....	24
Frequenztafel TRX433-10C .....	25
Frequenztafel TRX868-10C .....	26
Befehlssatz zur Konfiguration (Version C2).....	27
Befehlsstruktur.....	27
Konfigurationen im RAM oder EEPROM.....	27
Befehlsübersicht Version C2 .....	28
Initialisierungsbeispiel in C .....	29
Funktionsbeschreibungen im Detail .....	31
Funktionsgruppe READ .....	31
Funktionsgruppe WRITE .....	40
Funktionsgruppe REPORT .....	50
Funktionsgruppe ERROR .....	51
Werkseinstellung.....	52
Codierung Typenschild .....	53
CE Konformitätserklärung.....	54



## Kurzbeschreibung

Das Transceivermodul TRXnnn-10C mit integriertem Mikroprozessor wird über die serielle Schnittstelle mit einfachen Kommandos parametrierbar (Frequenz, Funkdatenrate usw.). Die Datenkommunikation verläuft ebenfalls über die serielle Schnittstelle. Dabei übernimmt das Transceivermodul die Codier- und Decodieraufgaben für die Funkübertragung. Je nach Anforderung kann die automatische Funkübertragung den aktuellen Bedürfnissen angepasst werden (z.B. minimale Verzögerung, maximale Datensicherheit usw.). Durch die einfache Ansteuerung über die serielle Schnittstelle bei gleichzeitig hoher Kontrolle über den Datenfluss steht dem Anwender ein mächtiges Werkzeug zur Realisation von einfachen Punkt zu Punkt Funkverbindungen bis hin zu komplexen Funknetzwerken zur Verfügung, ohne sich dabei um HF-spezifische Probleme kümmern zu müssen. Mit diesen Transceivern lassen sich hervorragende Funkreichweiten erzielen.

## Eigenschaften

- einfache Datenkommunikation über serielle Schnittstelle. Codier- und Decodieraufgaben für die Funkübertragung werden vom Transceivermodul übernommen.
- flexibel und einfach konfigurierbar mittels einfacher RS232 Kommandos. Leistungsfähiger Befehlssatz für viele Kommunikationsparameter und Stromsparfunktionen.
- Schmalbandbetrieb mit max. 139 bzw. 159 Frequenzen im 12.5kHz Raster. Einstellbare Funkdatenraten von 1.2kbits/s bis 19.2kbits/s (mit Bestückungsvariante bis max. 38.4kbits/s möglich)
- geringer Stromverbrauch und schnelles Setup. Stromsparfunktionen, optimiert für Batteriebetrieb
- kompakt und leicht, ideal für portable Geräte
- LED Statusanzeige, per Kommando abschaltbar
- Version ohne Spannungsregler lieferbar für Batteriespannungen von 2.4 ... 3.6V
- drei pin-kompatible Versionen A, B, C decken unterschiedliche Anforderungen ab. Der Befehlssatz ist auf Kundenwunsch erweiterbar.

## Anwendungen

- hochwertige Fernsteuerungen mit Rückmeldung
- Industrie, Gewerbe, Gebäudeautomation, Sicherheitstechnik

Für die komfortable Evaluation oder für den leichten Einstieg wird der Demokit3 empfohlen. Damit kann innerhalb von wenigen Minuten eine bidirektionale serielle PC-PC oder PC-Drucker Verbindung aufgebaut oder die Reichweite im Feld (ohne PC) getestet werden.

## Familienübersicht Funktransceiver

Die Funktransceiver der Familie TRX433 und TRX868 werden in mehreren Ausführungen angeboten, welche sich in der Software und / oder in der Hardware voneinander unterscheiden.

Zurzeit gibt es die drei Versionen A, B und C, welche sich wie folgt unterscheiden

	Version A	Version B	Version C
Datenschnittstelle	RS232	beliebig, transparent	RS232
Konfiguration über RS232	Nur wichtigste Parameter, nur bei Powerup möglich	Umfassender Befehlssatz, jederzeit während Betrieb konfigurierbar	Umfassender Befehlssatz, jederzeit während Betrieb konfigurierbar
Sleep, Wakeupfunktionen	nein	ja	ja
delay TX-RX <sup>1</sup>	$t_{\text{DATA}} + t_{\text{RADIO}}$ , siehe Timingdiagramm	$6 t_{\text{BIT-RADIO}}$ Jitter $\pm 1/8 t_{\text{BIT-RADIO}}$	$t_{\text{DATA}} + t_{\text{RADIO}}$ siehe Timingdiagramm
Fehlerprüfung	CRC16	keine	mit / ohne CRC16
Retransmit nach Fehler	Solange, bis Daten vom Empfänger korrekt quittiert	nein	nein
Buffergrösse TX	2 x 31 Bytes		Ringbuffer, 61 Bytes <sup>2</sup>
Buffergrösse RX	1 x 31 Bytes, Ringbuffer		Ringbuffer, 61 Bytes <sup>2</sup>
Datenhandshake	RTS-CTS, XON-XOFF		RTS-CTS

Die drei Versionen decken die Bedürfnisse verschiedener Anwendung folgendermassen ab:

**Version A** ist für einfache Anwendungen als Kabelersatz z.B. zwischen PC und Peripheriegeräten gedacht, wo die Zeitverzögerung durch die Funkstrecke keine Rolle spielt. Die Daten werden vom Transceiver selbständig auf Fehler geprüft, quittiert und im Fehlerfall wiederholt. Die Daten werden über Funk erst gesendet, wenn der Buffer voll ist oder wenn während der Zeitdauer von 3ms keine weiteren Daten mehr anliegen. Version A ist zusammen mit dem Evalkit3 ideal geeignet, um die Funkreichweite zu testen.

**Version B** wird eingesetzt, wenn die maximale Transparenz und Kontrolle über den Funkkanal notwendig ist oder gewünscht wird. Es wird absolut keine Codierung oder Fehlerprüfung gemacht, das senderseitig angelegte Signal wird transparent und mit minimalster Verzögerung 1:1 beim Empfänger ausgegeben, dadurch optimale Kompatibilität zu beliebigen Codierungsarten und Funkmodulen anderer Hersteller.

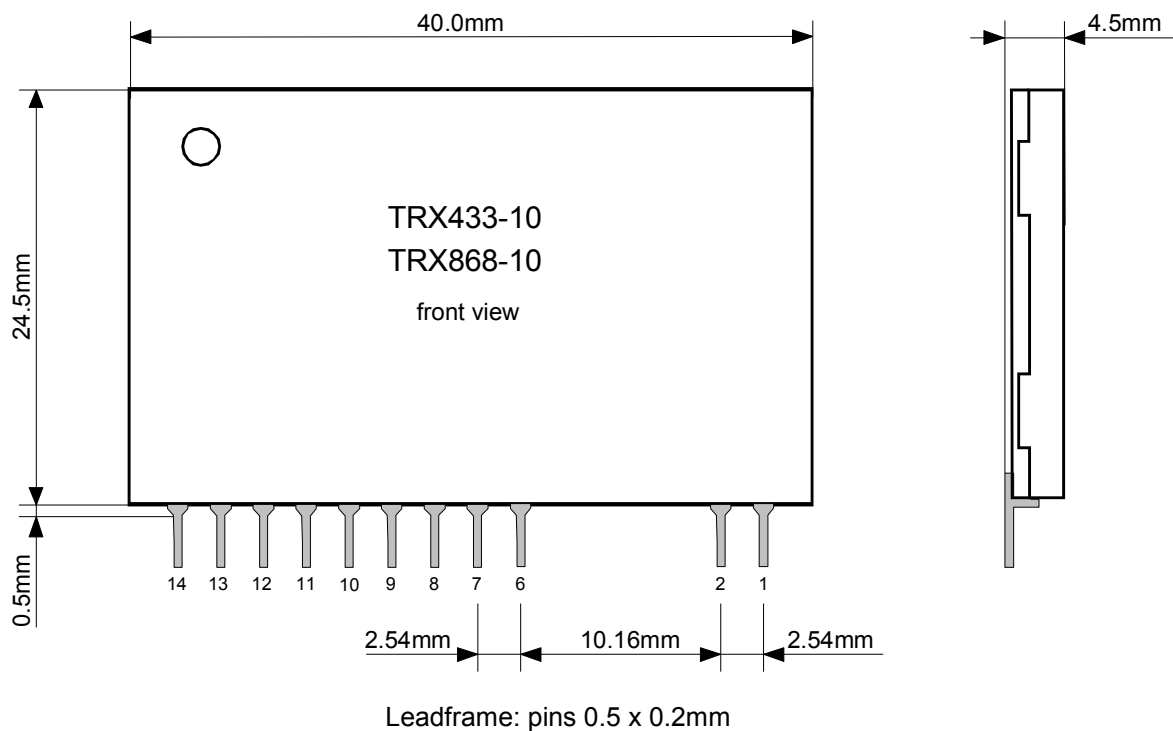
**Version C** mit transparentem Bytemodus wird eingesetzt, wenn kurze Reaktionszeiten bei gleichzeitig einfachster Ansteuerung gewünscht sind. Die byteweise seriell eingehenden Daten werden mit einer kurzen Verzögerung über Funk gesendet, sodass der Empfänger bereits wenige Millisekunden nach der Übertragung des 1. Datenbytes mit der seriellen Datenausgabe beginnen kann.

Wenn die Fehlerprüfung aktiviert ist, werden empfängerseitig nur korrekt übermittelte Daten ausgegeben, ohne die Fehlerprüfung muss diese Aufgabe durch die übergeordnete Applikation erfolgen.

<sup>1</sup>  $t_{\text{BIT\_RADIO}}$ : Zeitdauer für 1 Bit der eingestellten Funkdatenrate

<sup>2</sup> ist die Funkdatenrate im Verhältnis zur RS232 Baudrate genügend hoch eingestellt, entsteht ein kontinuierlicher Datenfluss ohne Unterbrechung durch den Handshake

## Dimensions



## Pin description

Pin	Name	I/O	Description	Level	Condition
1	RF	I/O	RF- in/out for lambda / 4 antenna (~ 50 Ω)	0 V	DC-path to GND
2	GND		RF Signal Ground		
6	V_P	O	Programming voltage (do not connect)	VCC	
7	MCLR\	I	Reset input, active low. <sup>3</sup>	VCC	Normal operation
8	RSSI	O	Received Signal Strength Indicator (analog out)	0 V	-128 dBm
9	RTS	O	RTS Handshake	0 V	Ready to send
10	TxD	O	UART Transmit Data (digital out)	VCC	Stop bit or no data
11	RxD	I	UART Receive Data (digital in)	VCC	Stop bit or no data
12	CTS MODE WKUP	I I I	CTS Handshake Interface mode (data or configuration) Sleep mode: Pin change terminates sleep	0 V 0 V	Clear to send Data mode
13	V+		Positive supply voltage		3.5 to 6VDC <sup>4</sup> or 2.4 to 3.6VDC <sup>5</sup>
14	GND		Ground		

<sup>3</sup> connect an external reset controller, if supply voltage is not always within specified limits or if voltage ramp is slower than 50ms from 0V to 3.5V.

<sup>4</sup> standard version with internal 3.3V supply voltage regulator (output logic high levels are limited to the internal 3.3V supply voltage)

<sup>5</sup> version without internal voltage regulator (internal supply voltage = V+)

## Allgemeine Informationen zur Funktionalität

### Serielle Schnittstelle (RS232)

Zur Kommunikation mit dem Transceivermodul steht eine serielle, bidirektionale Vollduplexschnittstelle zur Verfügung. Über dieselbe Schnittstelle werden Daten übertragen sowie das Transceivermodul konfiguriert. Mit dem MODE Pin wird zwischen Datenmodus und Konfigurationsmodus gewählt.

#### Datenmodus (MODE Pin = Low)

Empfängt das Transceivermodul Daten über die serielle Schnittstelle, so werden diese automatisch über Funk an alle Transceiver mit den gleichen Einstellungen (Frequenz, Funkdatenrate, usw.) übertragen und bei diesen nach wenigen Millisekunden wieder über die serielle Schnittstellen ausgegeben. Die Baudrate für die RS232-Übertragung kann von 1.2kbaud bis 115.2kbaud<sup>6</sup> eingestellt werden.

#### Konfigurationsmodus (MODE Pin = High)

Das Transceivermodul wird über die serielle Schnittstelle konfiguriert. Es steht ein umfassender Befehlssatz zur Verfügung. Die letzte Konfiguration bleibt im internen EEPROM gespeichert, falls im Befehl das entsprechende Bit für die Speicherung im EEPROM gesetzt ist. Die Konfiguration kann in beliebiger Baudrate von 1.2kbaud bis 115.2kbaud<sup>6</sup> erfolgen (im Konfigurationsmodus wird die Baudrate automatisch detektiert).

### Funkübertragung

Im Datenmodus werden über die serielle Schnittstelle empfangene Daten vom Transceivermodul selbstständig über Funk übertragen. Das Transceivermodul übernimmt dabei vollständig die Codier- und Decodieraufgaben der Funkübertragung. Der Anwender kann aber die Funkübertragung mit Hilfe von wenigen einfachen Befehlen auf seine Bedürfnisse anpassen. Folgende Einstellungen können vorgenommen werden:

#### Frequenzwahl

Es stehen 139 Frequenzen im 433 MHz – Band bzw. 159 Frequenzen im 868 MHz – Band zur Verfügung, jeweils im 12.5kHz – Kanalraster. Die Frequenz kann unabhängig von der eingestellten Funkdatenrate gewählt werden. Es ist aber darauf zu achten, dass bei Systemen mit mehreren verwendeten Frequenzen der Frequenzabstand mindestens dem Abstand der belegten Kanäle der eingestellten Funkdatenrate entspricht (siehe Tabelle 1). Für bessere Reichweiten wird dieser Abstand so weit wie möglich erhöht.

#### Funkdatenrate

Die Funkdatenrate kann je nach Anforderung von 1.2kbit/s bis 38.4kbit/s<sup>7</sup> gewählt werden. Je kleiner die Funkdatenrate gewählt wird, desto höhere Funkreichweiten können mit den Transceivermodulen erreicht werden. Funkdatenraten >19.2kbits/s sind möglich, erfordern jedoch ein entsprechend dimensioniertes Loopfilter bzw. eine Bestückungsänderung. Durch diese Änderung wird die Empfängerempfindlichkeit bei kleinen Funkdatenraten reduziert. Die Standardversion TRXnnn-10C2 ist mit einem Loopfilter für max. 19.2kbits/s bestückt.

Zur gewählten Funkdatenrate werden weitere Einstellungen für die Funkübertragung automatisch vorgenommen. Die weiteren Einstellungen für die verschiedenen möglichen Funkdatenraten können der Tabelle 1 entnommen werden.

Funkdatenrate	Codierung	Modulation	FM-Deviation	ZF-Bandbreite	Belegte Kanäle	Bemerkung
1.2 kbit/s	Manchester	GFSK	+ - 2.025 kHz	9.6 kHz	1 x 12.5kHz	433 MHz
	Manchester	GFSK	+ - 2.025 kHz	12.3 kHz	1 x 12.5kHz	868 MHz
2.4 kbit/s	NRZ	GFSK	+ - 2.025 kHz	9.6 kHz	1 x 12.5kHz	433 MHz
	NRZ	GFSK	+ - 2.025 kHz	12.3 kHz	1 x 12.5kHz	868 MHz
4.8 kbit/s	NRZ	GFSK	+ - 4.050 kHz	19.2 kHz	2 x 12.5kHz	
9.6 kbit/s	NRZ	GFSK	+ - 4.950 kHz	25.6 kHz	4 x 12.5kHz	
19.2 kbit/s	NRZ	GFSK	+ - 9.900 kHz	51.2 kHz	8 x 12.5kHz	
38.4 kbit/s <sup>7</sup>	NRZ	GFSK	+ - 19.80 kHz	102.4 kHz	12 x 12.5kHz	

**Tabelle 1 Wählbare Funkdatenraten mit zugehörigen weiteren Einstellungen**

<sup>6</sup> Die höchste Baudrate von 115.2 kbaud ist nur mit einer 16MHz  $\mu$ P Clockfrequenz möglich

<sup>7</sup> Die höchste Funkdatenrate von 38.4 kbit/s ist nur mit einer 16MHz  $\mu$ P Clockfrequenz und dem entsprechenden Loopfilter möglich

## Aufbau Funktelegramm

Zur Funkübertragung werden die eigentlichen Nutzdaten in ein Funktelegramm eingebunden. Die Bildung dieses Funktelegramms und die dazugehörigen Codier- und Decodieraufgaben übernimmt das Transceivermodul und entlastet somit den Anwender. Ein komplettes Funktelegramm ist gemäss Abbildung 1 aufgebaut.

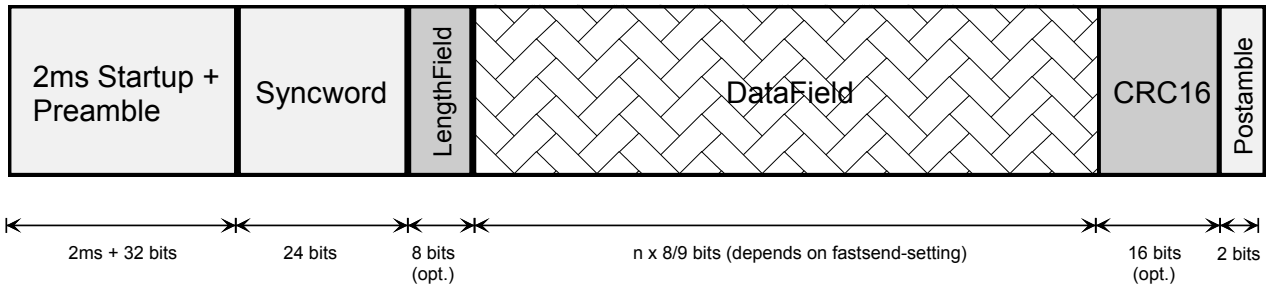


Abbildung 1 Aufbau des Funktelegramms

Funktion der einzelnen Blöcke im Funktelegramm:

- 2ms Startup: Einschwingzeit für den Sender zur Reduktion des Störspektrums
- Preamble: Erlaubt dem Empfänger auf das Trägersignal des Senders einzupegeln
- Syncword: Zeigt Start der Datenübertragung an und dient somit dem Empfänger zur Synchronisation
- LengthField: Gibt Anzahl Bytes Nutzdaten im DataField an (optional)
- DataField: Effektive Nutzdaten
- CRC16: Prüfsumme für Nutzdaten im DataField (optional)
- Postamble: Nachsenden für korrekten Empfang der letzten Datenbits des Funktelegramms

Jedes Funktelegramm besteht zwingend aus den Blöcken 2ms Startup + Preamble, Syncword, DataField und Postamble. Die Blöcke LengthField und CRC16 werden optional, je nach gewählter Einstellung ebenfalls zum Funktelegramm hinzugefügt. Die Zeitdauer der einzelnen Blöcke ist abhängig von der eingestellten Funkdatenrate und berechnet sich aus der Anzahl Bits mal die Bitdauer bei der eingestellten Funkdatenrate. Einzig die Zeitdauer des 2ms Startup bleibt bei jeder Funkdatenrate konstant 2ms.

## Sende- und Empfangsbuffer

Das Transceivermodul verfügt über je einen 61Byte grossen Sendebuffer und Empfangsbuffer für die serielle Schnittstelle.

### Sendebuffer (TX-BUF)

Im Sendebuffer werden die über RS232 empfangen Daten bis zur Funkübertragung zwischengespeichert. Der Platz im Sendebuffer wird Byteweise wieder freigegeben, sobald ein Byte im Funktelegramm an der entsprechenden Stelle im DataField eingefügt werden kann.

### Empfangsbuffer (RX-BUF)

Im Empfangsbuffer werden die über Funk empfangen Daten bis zur Ausgabe über RS232 zwischengespeichert. Ist die Funktion „Ausgabe des Telegramm-RSSI“ bzw. „Ausgabe des Ruhe-RSSI“ eingeschaltet, so werden diese RSSI-Werte ebenfalls im Empfangsbuffer im Anschluss an die Nutzdaten des Funktelegramms zwischengespeichert. Pro Funktelegramm werden also je nach Einstellung noch maximal 2 zusätzliche Bytes im Empfangsbuffer benötigt. Der Platz im Empfangsbuffer für den Funktelegrammempfang muss erst zu dem Zeitpunkt frei sein, zu dem die effektiven Nutzdaten aus dem DataField des Funktelegramms auch wirklich empfangen werden. Der Platz im Empfangsbuffer wird byteweise wieder freigegeben, sobald ein Byte über RS232 ausgegeben wird.



**Bufferüberlauf**

Es ist darauf zu achten, dass die beiden Buffer (Sende- und Empfangsbuffer) nicht überfüllt werden. Dies kann durch Einhalten des richtigen Timings oder mit Hilfe des Handshakes sichergestellt werden (siehe Abschnitte Datenflussbeispiele und Handshake mit RTS/CTS).

**Überlauf des Sendebuffers:**

Der Sendebuffer verfügt zusätzlich zu seinen 61 Bytes noch über einen 2-Byte Überlaufbuffer. Werden also trotz vollem Sendebuffer (61Bytes belegt) noch weitere Daten über RS232 übertragen, so können noch maximal 2 Bytes im Überlaufbuffer gespeichert werden. Diese werden in den Sendebuffer übertragen, sobald in diesem wieder Platz vorhanden ist. Ist auch der Überlaufbuffer belegt, so werden weitere über RS232 empfangene Daten verworfen.

**Überlauf des Empfangsbuffers:**

Der Empfangsbuffer verfügt zusätzlich zu seinen 61 Bytes noch über einen 2-Byte Überlaufbuffer. Dieser dient ausschliesslich dazu, bei eingeschalteter Funktion „Ausgabe des Telegramm-RSSI“ bzw. „Ausgabe des Ruhe-RSSI“ noch die RSSI-Werte im Anschluss an die Nutzdaten zu speichern, falls mit dem Funktelegramm schon alle 61 Bytes des Empfangsbuffers belegt sind.

Werden Daten über Funk empfangen, obwohl der Buffer bereits voll ist (61 Bytes belegt), so wird bei eingeschalteter CRC16 das ganze eingehende Funktelegramm verworfen. Bei ausgeschalteter CRC16 bleiben die bereits empfangenen Daten gespeichert, der Rest des eingehenden Funktelegramms wird verworfen.

**Handshake mit RTS/CTS**

Die Kommunikation über die RS232-Schnittstelle kann mit Hilfe von RTS/CTS gesteuert werden, um Bufferüberläufe zu verhindern. RTS und CTS werden nur für den Handshake verwendet, wenn dieser auch eingeschaltet ist. Ist der Handshake ausgeschaltet, werden RTS und CTS ignoriert.

**Funktion RTS**

Sobald 61 Bytes im Sendebuffer gespeichert sind, stoppt das Transceivermodul den Empfang von Daten über RS232, indem der RTS-Pin auf High gesetzt wird. Der Anwender darf nun keine weiteren Daten mehr über die RS232-Schnittstelle an das Transceivermodul senden, bis der RTS-Pin wieder auf Low geht. Die RS232-Schnittstelle wird wieder freigegeben (RTS-Pin zurückgesetzt auf Low), sobald wieder ≤58 Bytes im Sendebuffer gespeichert sind.

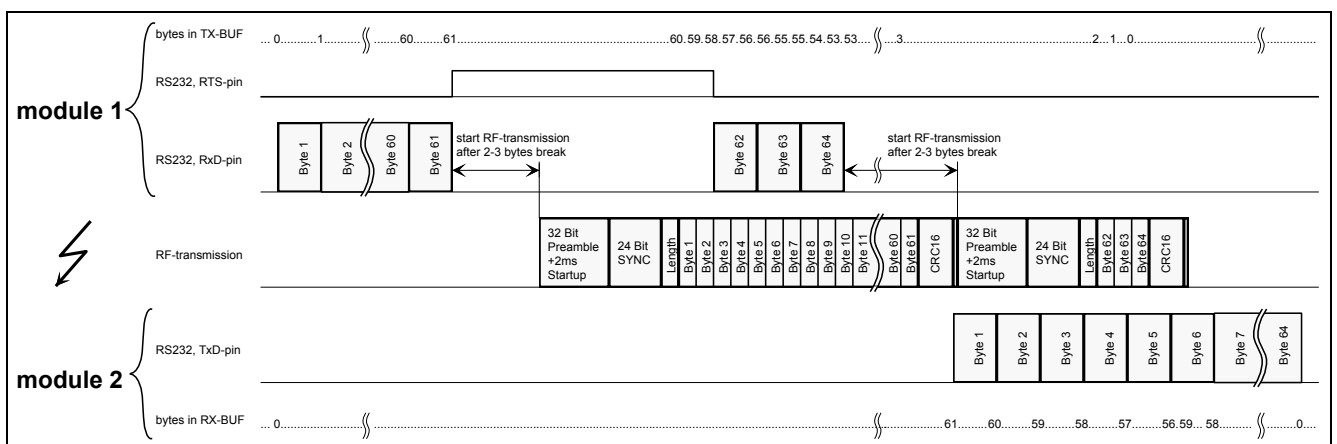
**Funktion CTS**

Bei eingeschaltetem Handshake kann der Anwender die Ausgabe von Daten durch das Transceivermodul stoppen, indem er den CTS-Pin auf High setzt. Das Transceivermodul gibt nun keine weiteren Daten mehr über den TxD-Pin aus, bis der CTS-Pin vom Anwender wieder zurück auf Low gesetzt wird.

**Hinweis**

Falls der CTS-Pin als Handshake verwendet wird, dürfen während der High-Zeit des CTS-Pins keine Daten an das Transceivermodul gesendet werden, da diese Daten als Konfigurationsdaten interpretiert würden. Der CTS-Pin besitzt also die Doppelfunktion Handshake (CTS) und Interfacemode-Wahl (MODE).

In Abbildung 2 ist als Beispiel der Ablauf einer Datenübertragung von 64 Bytes mit Hilfe des Handshakes dargestellt. Dabei wird Modul 1 als Sender und Modul 2 als Empfänger verwendet. Ausserdem ist zu erkennen, wie viele Bytes im Sendebuffer (TX-BUF) des sendenden Transceivermoduls und im Empfangsbuffer (RX-BUF) des empfangenden Transceivermoduls gespeichert sind.



**Abbildung 2 Datenübertragung mit Handshake**



## **CRC16**

Bei der CRC16 handelt es sich um eine 16-Bit Checksumme, mit der die Nutzdaten im DataField auf Korrektheit überprüft werden. Falls die übergeordnete Anwendung keine Prüfung der Daten übernimmt, sollte auf jeden Fall die CRC16 verwendet werden.

### **CRC16 aus**

Die Funkdaten werden nicht auf Korrektheit geprüft. Sobald das erste Datenbyte aus dem DataField des Funktelegramms empfangen wird, wird es über RS232 ausgegeben. Die übergeordnete Anwendung muss die Prüfung der Daten übernehmen.

### **CRC16 ein**

Die Nutzdaten im DataField werden CRC16 checksummengepüft und die errechnete Checksumme wird am Ende des Funktelegramms mitgesendet. Dies bedeutet, dass die Daten beim empfangenden Transceivermodul erst am Ende der Funkübertragung und nur bei übereinstimmender CRC16 über RS232 ausgegeben werden. Bei falscher CRC16 wird das ganze Funktelegramm verworfen.

## **Fastsend**

Die Einstellung Fastsend bestimmt, wann die Funkübertragung der über RS232 empfangen Nutzdaten beginnt und auf welche Art die Funktelegrammlänge übermittelt wird.

### **Fastsend aus**

Die an einem Stück über RS232 empfangen Daten werden im Sendebuffer zwischengespeichert. Sobald für die Zeitdauer von 2-3 RS232-Bytes keine Daten mehr über RS232 empfangen werden, wird die Funkübertragung gestartet und die aktuell gespeicherten Nutzdaten im Sendebuffer werden übertragen. Die Funktelegrammlänge ist somit schon zu Beginn der Funkübertragung bestimmt (im LengthField wird die Anzahl Bytes Nutzdaten im DataField eingetragen). Werden während der Funkübertragung weitere Daten über RS232 empfangen, so werden diese wieder im Sendebuffer zwischengespeichert und bilden ein neues Funktelegramm, sobald die Funkübertragung des aktuellen Funktelegramms abgeschlossen ist.

### **Fastsend ein**

Die Funkübertragung wird gestartet, sobald das erste Byte über RS232 empfangen wurde und wird erst beendet, wenn sich kein Byte mehr im Sendebuffer befindet. Bei dieser Funktion ist somit die Funktelegrammlänge zu Beginn der Funkübertragung noch nicht bekannt. Deshalb wird auch kein LengthField zum Funktelegramm hinzugefügt. Zur Kennzeichnung der Funktelegrammlänge wird jedem Nutzdatenbyte im DataField ein zusätzliches Bit angefügt, welches das Ende der Nutzdaten anzeigt.

## **Hinweise**

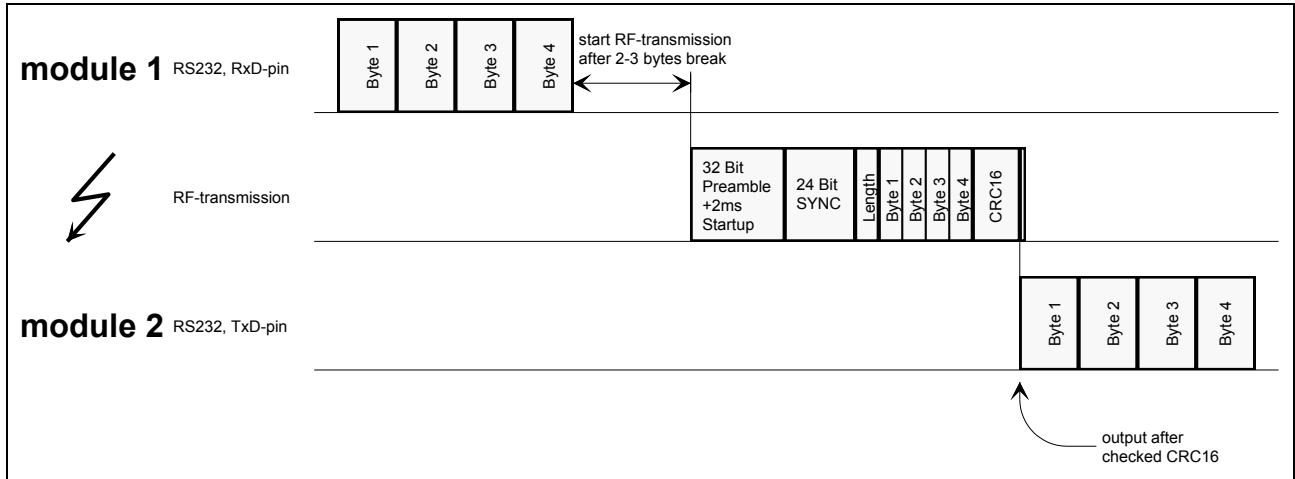
- Mit Fastsend beginnt die Funkübertragung zwar früher als ohne Fastsend, es wird aber jedem Nutzdatenbyte im DataField ein zusätzliches Bit zur Bestimmung der Funktelegrammlänge hinzugefügt. Dadurch wird das Funktelegramm bei mehr als 8 Bytes Nutzdaten länger als ein Funktelegramm ohne Fastsend.
- Da Fastsend für das Transceivermodul zusätzlichen Rechenaufwand bedeutet, kann Fastsend nur bis zu einer Funkdatenrate von 19.2kbit/s (bzw. 9.6kbit/s bei 8MHz  $\mu$ P Clockfrequenz) verwendet werden.
- Mit eingeschalteter CRC16 ist die Funktelegrammlänge auch mit Fastsend auf 61Bytes beschränkt, da der Empfänger das ganze Funktelegramm zwischenspeichern muss, bis die CRC16 geprüft ist. Der Anwender muss also sicherstellen, dass die Funktelegrammlänge 61Bytes nicht überschreitet.
- Mit ausgeschalteter CRC16 und Fastsend sind Funktelegrammlängen mit mehr als 61Bytes Nutzdaten möglich.
- Die Pause mit der Zeitdauer von 2-3 RS232-Bytes für den Start der Funkübertragung ohne Fastsend variiert je nach eingestellter Konfiguration. Um sicherzustellen, dass die Funkübertragung auch wirklich gestartet wird, sollte also in jedem Fall mindestens eine Pause mit der Zeitdauer von 3 RS232-Bytes eingehalten werden.

**Datenflussbeispiele**

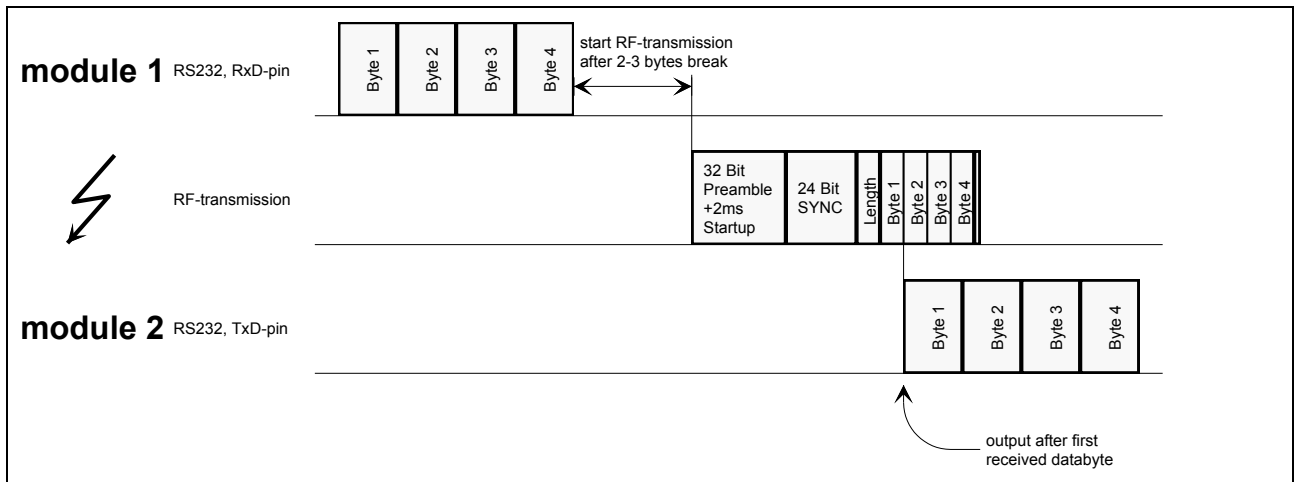
Im Folgenden werden einige Beispiele aufgeführt, denen der Datenfluss bei verschiedenen Einstellungen von Fastsend, CRC16, Handshake usw. entnommen werden kann.

**Beispiele für verschiedene Einstellungen von Fastsend und CRC16**

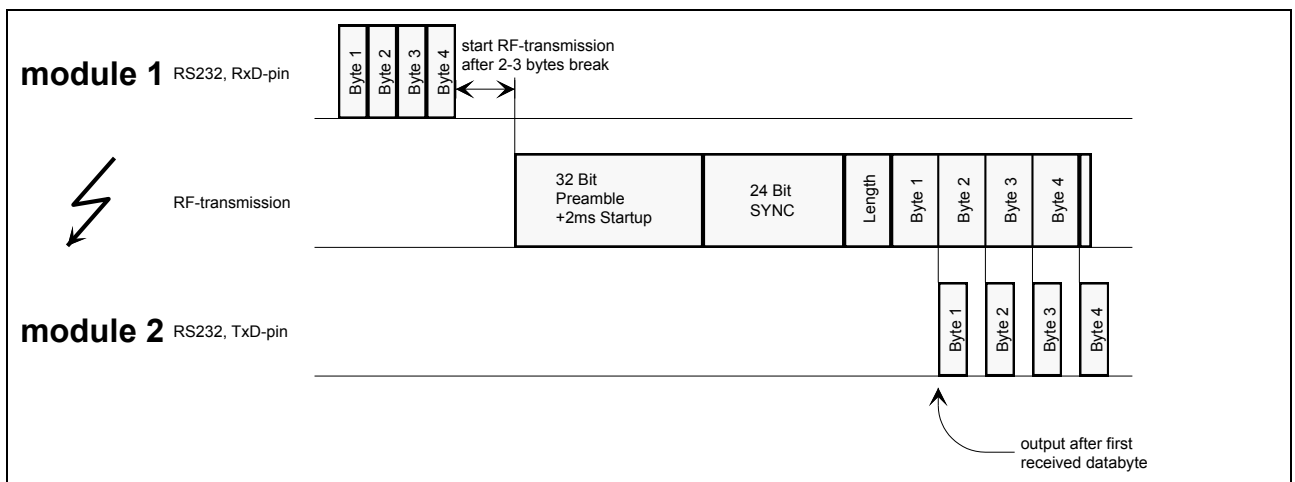
Den folgenden 6 Abbildungen kann der Datenfluss, abhängig von den Einstellungen Fastsend und CRC16 entnommen werden. Dabei wird Modul 1 als Sender und Modul 2 als Empfänger verwendet. Natürlich können auch mehrere Module mit den gleichen Einstellungen wie der Sender das Datensignal empfangen.



**Abbildung 3 Datenfluss ohne Fastsend mit CRC16, Funkdatenrate > RS232-Baudrate**



**Abbildung 4 Datenfluss ohne Fastsend ohne CRC16, Funkdatenrate > RS232-Baudrate**



**Abbildung 5 Datenfluss ohne Fastsend ohne CRC16, Funkdatenrate < RS232-Baudrate**

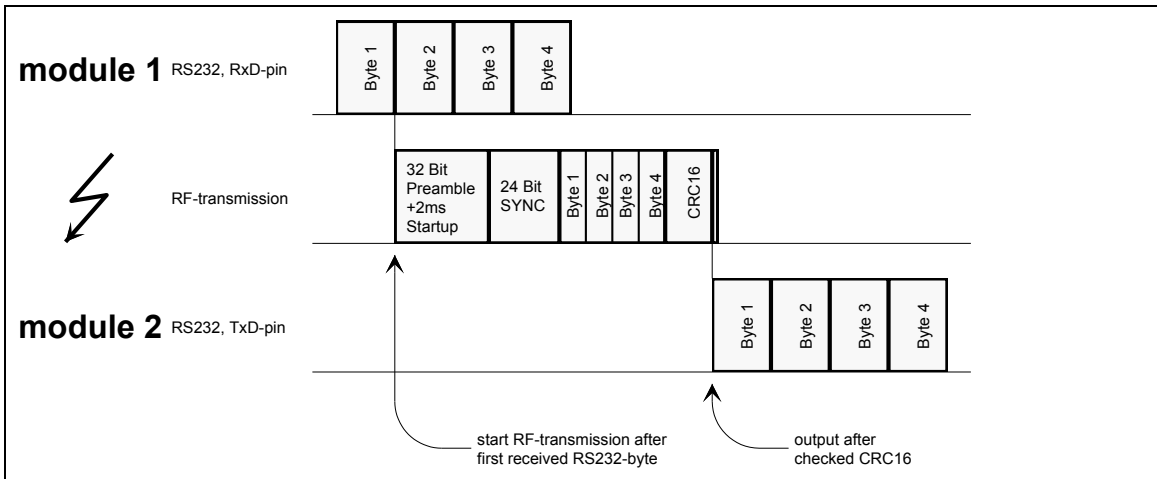


Abbildung 6 Datenfluss mit Fastsend mit CRC16, Funkdatenrate > RS232-Baudrate

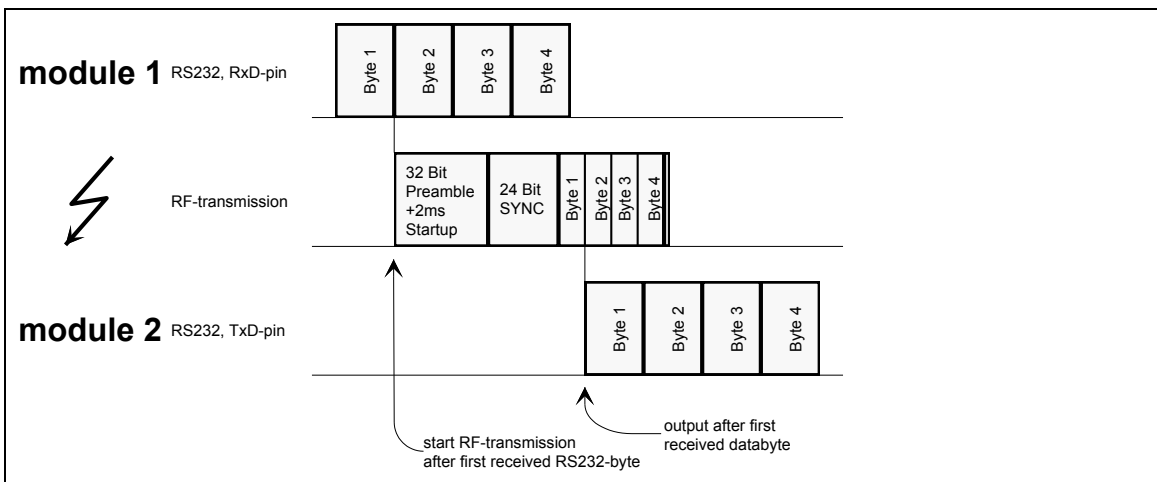


Abbildung 7 Datenfluss mit Fastsend ohne CRC16, Funkdatenrate > RS232-Baudrate

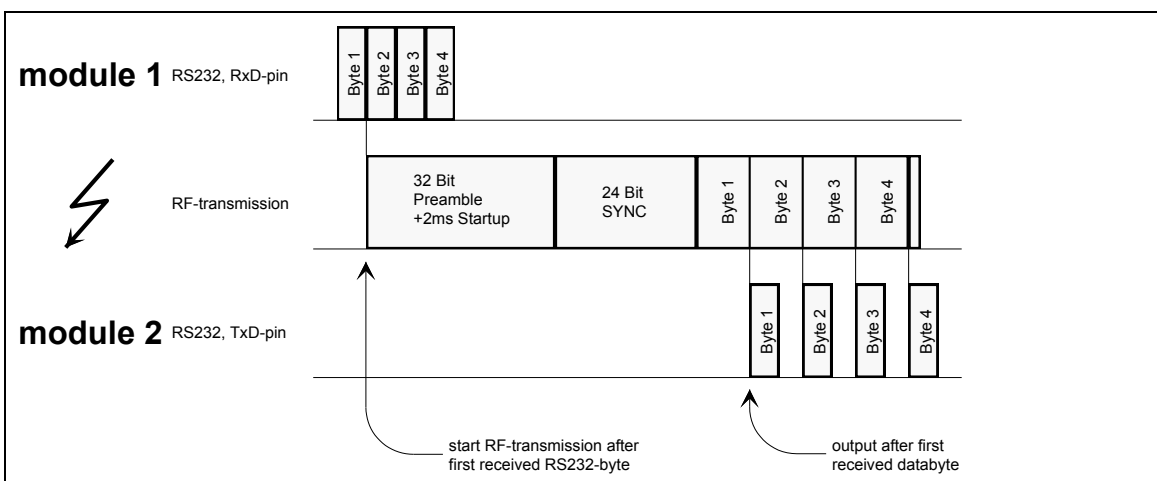
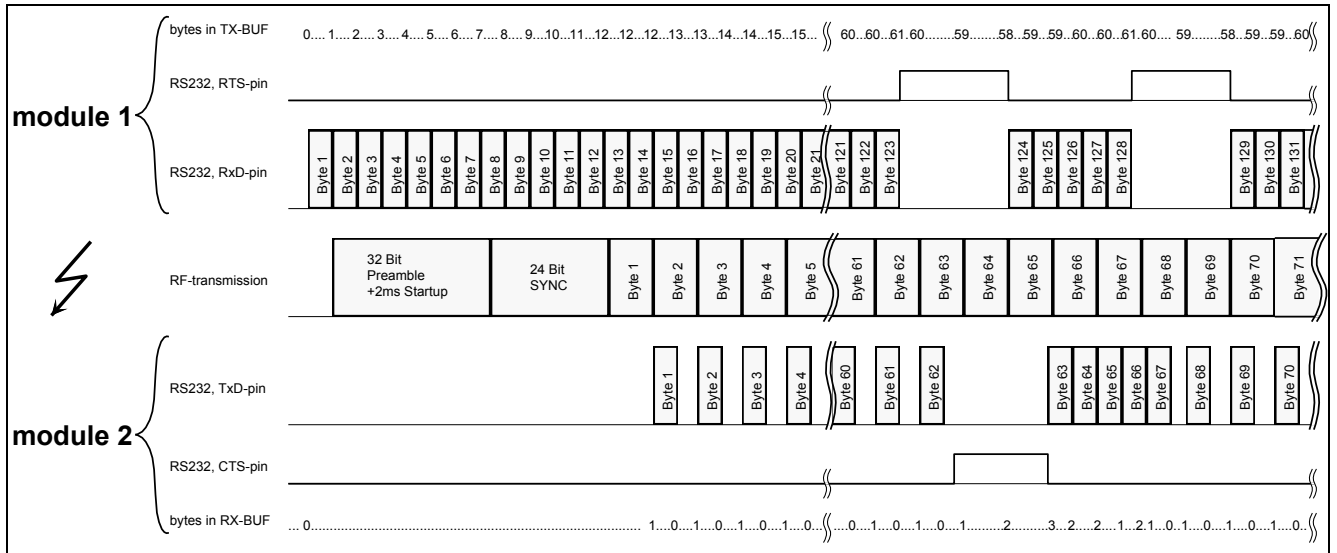


Abbildung 8 Datenfluss mit Fastsend ohne CRC16, Funkdatenrate < RS232-Baudrate

**Beispiel für kontinuierlichen Funkdatenstrom (Funktelegramme mit mehr als 61 Bytes Nutzdaten)**

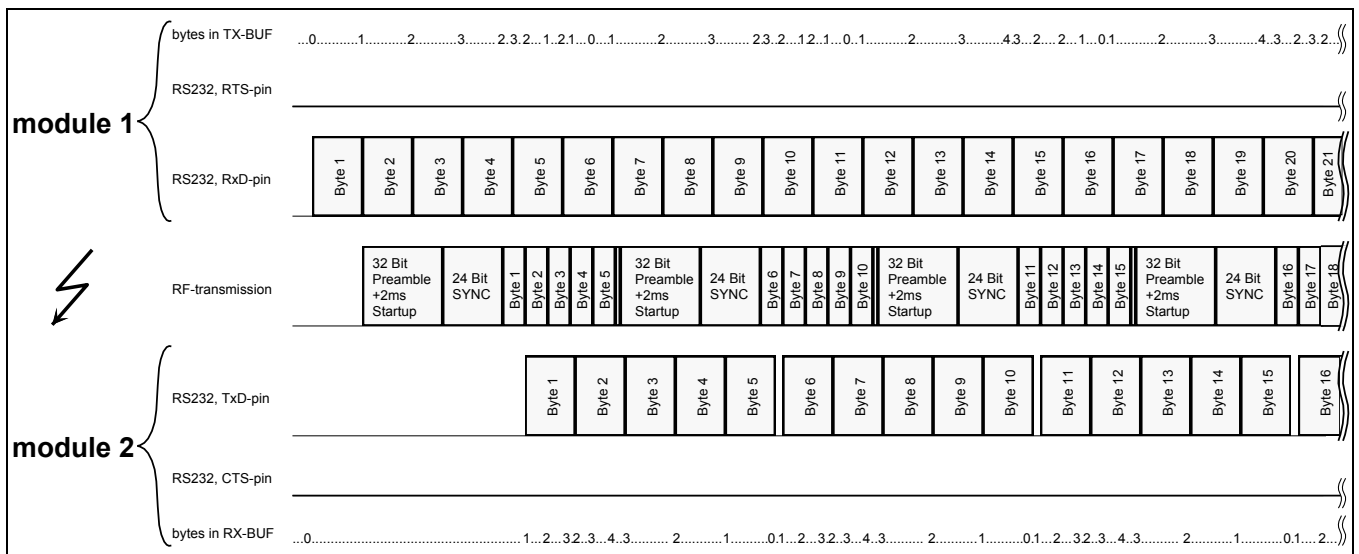
Ein kontinuierlicher Funkdatenstrom (also ein Telegramm, das nicht auf 61 Bytes Nutzdaten beschränkt ist) kann nur mit FastSend und ohne CRC16 realisiert werden. Ebenfalls muss die RS232-Baudrate mindestens so schnell gewählt werden wie die Funkdatenrate. Mit diesen Einstellungen kann ein kontinuierlicher Funkdatenstrom gemäss Abbildung 9 realisiert werden. In Abbildung 9 ist ausserdem zu sehen, wie mit dem CTS-Pin das Transceivermodul daran gehindert werden kann, seine über Funk empfangenen Daten über RS232 auszugeben. Ebenfalls ist die Anzahl Bytes im Sendebuffer (TX-BUF) des sendenden Transceivermoduls und im Empfangsbuffer (RX-BUF) des empfangenden Transceivermoduls abgebildet.



**Abbildung 9 Datenstrom mit FastSend ohne CRC16, RS232-Baudrate = 2x Funkdatenrate**

**Beispiel für kontinuierlichen Datenstrom über RS232**

Bei eingeschaltetem FastSend und ausgeschalteter CRC16 kann ein kontinuierlicher Datenstrom über die RS232-Schnittstelle realisiert werden. Dazu muss die Funkdatenrate höher gewählt werden als die RS232-Baudrate. In Abbildung 10 ist ein kontinuierlicher Datenstrom über RS232 dargestellt, bei dem die Funkdatenrate doppelt so schnell eingestellt ist wie die RS232-Baudrate. Ebenfalls ist zu erkennen, wie viele Bytes im Sendebuffer (TX-BUF) des sendenden Transceivermoduls und im Empfangsbuffer (RX-BUF) des empfangenden Transceivermoduls gespeichert sind.



**Abbildung 10 Datenstrom mit FastSend ohne CRC16, Funkdatenrate = 2x RS232-Baudrate**

### RSSI des Telegramms und Ruhe-RSSI

Um die Qualität einer Funkverbindung zu bestimmen, kann nach jedem empfangenen Funktelegramm die Signalstärke (RSSI) des eben empfangenen Funktelegramms über RS232 ausgegeben werden. Die Signalstärke wird im Anschluss an die Nutzdaten des Funktelegramms im 2er-Komplement in dBm ausgegeben.

Zusätzlich zum RSSI des Telegramms kann optional auch noch das Ruhe-RSSI ausgegeben werden. Als Ruhe-RSSI wird der unmittelbar nach dem Funktelegramm gemessene RSSI-Wert ausgegeben. Aus diesen beiden RSSI-Werten kann der Signal-Rausch-Abstand berechnet werden. Das Ruhe-RSSI kann natürlich auch mit dem Befehl „READ RSSI aktuell“ ausgelesen werden. In Abbildung 11 ist die Ausgabe des Telegramm-RSSI (RSSI TLG) und des Ruhe-RSSI (RSSI Ruhe) im Anschluss an die Nutzdaten zu sehen.

#### Hinweis

Ist die Funktion „Ausgabe des Telegramm-RSSI“ bzw. „Ausgabe des Ruhe-RSSI“ eingeschaltet, so werden diese RSSI-Werte im Empfangsbuffer des Transceivermoduls im Anschluss an die Nutzdaten des Funktelegramms zwischengespeichert, bis sie über RS232 ausgegeben werden können. Pro Funktelegramm werden also je nach Einstellung noch maximal 2 zusätzliche Bytes im Empfangsbuffer benötigt.

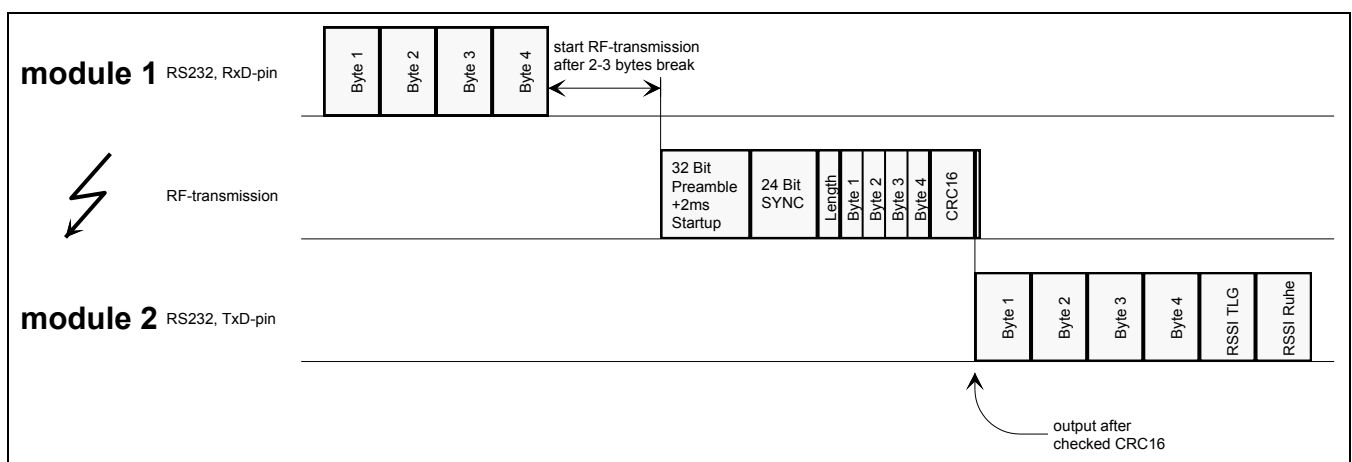


Abbildung 11 Ausgabe des Telegramm-RSSI und des Ruhe-RSSI

## Rekalibration

Bei Temperaturschwankungen > 40°C oder Schwankungen der Speisespannung von > 0.25 Volt (für Version ohne Spannungsregler) muss der VCO nachkalibriert werden. Die Kalibration kann automatisch in definierten Intervallen von 5 bis 150 Minuten, automatisch bei Temperaturunterschieden von mehr als 20°C oder jeweils nur einmalig manuell beim Kalibrierbefehl durchgeführt werden. Während der Kalibration, die typ. 50ms dauert, ist kein Funkempfang möglich. Beginn, Ende und Ergebnis der Kalibration werden rückgemeldet, falls sich das Modul im Konfigurationsmodus befindet.

Bei jedem powerup wird automatisch eine Kalibration gemacht. Während dem sleep bleibt die letzte Kalibration im RAM gespeichert. Der Transceiver ist somit nach dem wakeup schnell (bei 19.2kbit/s Funkdatenrate nach 3ms, bei 2.4kbit/s Funkdatenrate nach 5ms) wieder betriebsbereit. Wenn sich während dem sleep die Temperatur oder Speisespannung wesentlich ändern können, muss nach dem wakeup zuerst rekaliert werden.

## Stromsparmodi und Verhalten bei Reset, Powerup, Wakeup und sleep

Mit dem Steuerbefehl **Power-control** kann der HF-Teil des Transceivers ein- und ausgeschaltet werden, ebenso die LED. Im abgeschalteten Zustand wird der Stromverbrauch stark reduziert, der Prozessor bleibt jedoch weiterhin im aktiven Betrieb und benötigt Strom.

Für batteriebetriebene Geräte, welche die meiste Zeit inaktiv sind und nur während der Funkkommunikation Strom verbrauchen dürfen, wird der **sleep** Befehl verwendet. Dadurch kann der Verbrauch während des sleeps bis auf typ. 3µA (Version ohne internen Spannungsregler) reduziert werden. Die Version mit Spannungsregler benötigt ca. 100µA im sleep. Während dem sleep ist der Transceiver inaktiv, alle Einstellungen und die letzte Kalibration bleiben jedoch im RAM erhalten.

Infolge des geringen Standbyverbrauches im sleep und der kurzen setup-Zeit nach einem wakeup von 3ms bei 19.2kbit/s Funkdatenrate bzw. 5ms bei 2.4kbit/s Funkdatenrate, wird eine sehr hohe Batterielebensdauer erzielt, da der mittlere Stromverbrauch praktisch nur vom Verhältnis Sendezeit / Sendeintervall abhängt.

Der sleep- Befehl deaktiviert den HF-Teil und die LED und stoppt danach den clock des Mikrokontrollers. Mit dem Befehlsparameter WKUP\_mode des sleep-Befehls wird das Verhalten während des sleeps sowie das Verhalten nach einem wakeup definiert. Während dem sleep ist der Timer für automatische Rekalibration eingefroren.

Vor dem sleep Befehl, spätestens jedoch 50µs nach der Antwort auf den sleep Befehl muss der WKUP-Pin auf seinem sleep-Pegel sein (high oder low). Danach löst ein Pegelwechsel sofort einen wakeup aus.

Der Befehl **powerup mode** ist verwandt mit dem sleep-Befehl. Der sleep-Befehl wird sofort wirksam, der Befehl powerup mode jedoch wirkt sich erst beim nächsten Powerup aus. Der Transceiver kann mit PWUP\_mode so konfiguriert werden, dass er nach einem Powerup automatisch in den Sleepmode wechselt, bis er von aussen aufgeweckt und neu konfiguriert wird. Dies ist wichtig bei Batteriegeräten, damit nach einem Batteriewechsel das Funkmodul nicht selbst aktiv wird und unbemerkt die Batterie entleert.

Der Sleepmode wird durch einen Flankenwechsel am WKUP-Pin beendet. Dabei wird entweder das Programm dort weitergeführt, wo die Ausführung beim Sleep-Kommando unterbrochen wurde oder es erfolgt ein interner Reset, wobei ein kompletter Neustart erfolgt. Befindet sich der Transceiver im Konfigurationsmodus (MODE-Pin = High), so wird durch die Ausgabe des „REPORT READY“ über RS232 angezeigt, dass der Mikrocontroller auf dem Transceiver wieder betriebsbereit ist. Alternativ dazu kann auch der RTS-Pin überwacht werden. Nach dem Sleep-Befehl geht der RTS-Pin auf High. Sobald der Transceiver nach einem wakeup wieder bereit ist, wird der RTS-Pin wieder auf Low gesetzt (siehe Abbildung 12 und Abbildung 13).

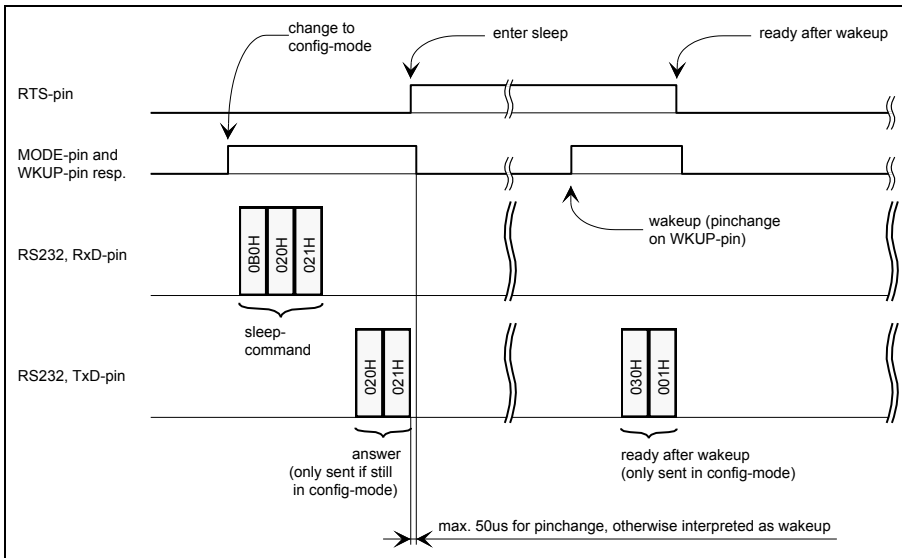


Abbildung 12 Sleep und Wakeup mit Konfigurationsmodus nach Wakeup

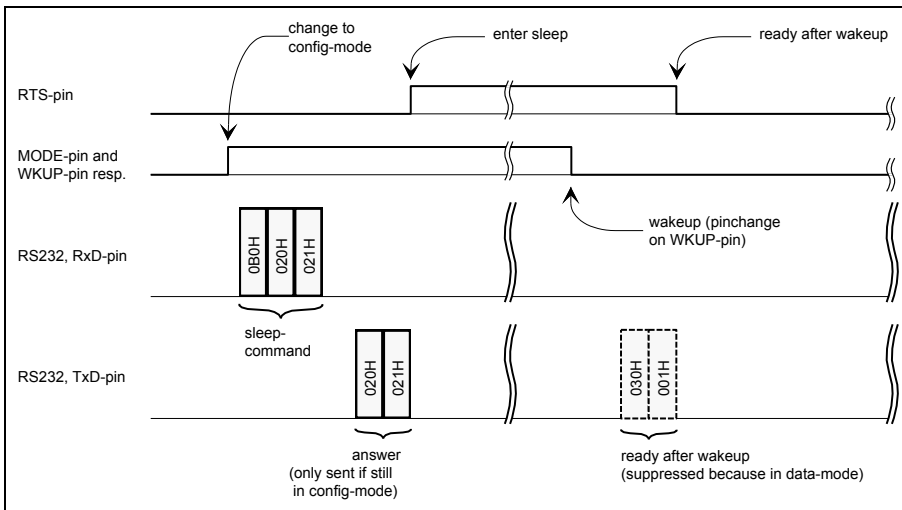


Abbildung 13 Sleep und Wakeup mit Datenmodus nach Wakeup

Nach einem powerup meldet sich das Transceivermodul nach ca. 175ms mit dem Breitzichen über RS232, falls sich das Transceivermodul im Konfigurationsmodus befindet. Alternativ dazu kann auch eine negative Flanke am RTS-Pin als Bereitzeichen verwendet werden (siehe Abbildung 14 bzw. Abbildung 15).

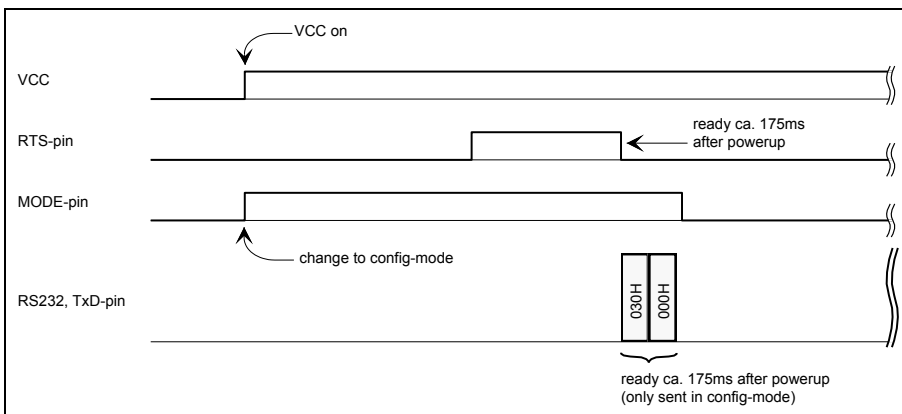
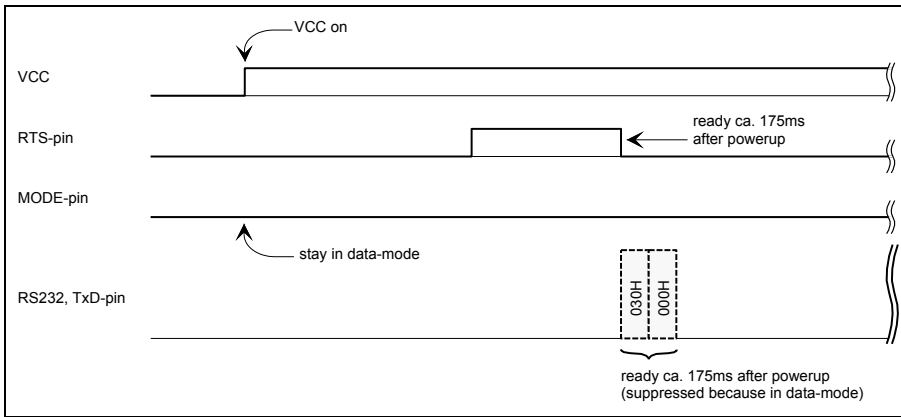


Abbildung 14 Powerup mit Transceivermodul im Konfigurationsmodus





**Abbildung 15 Powerup mit Transceivermodul im Datenmodus**

**Hinweise:**

- Wenn die Speisung geschaltet wird, speziell den Abschnitt Speisespannung und MCLR\ Reset beachten!
- Mit der sleep- Funktion kann in der Regel am meisten Batteriekapazität gespart werden. Es wird empfohlen, anstelle einer geschalteten Speisung die sleep- Funktion zu verwenden.

## RSSI (Received Signal Strength Indicator)

An Pin 8 (RSSI) liegt eine Spannung, welche direkt proportional zur HF-Empfangsfeldstärke ist. Der Transceiver tastet die Feldstärke alle 3ms ab und gibt den RSSI-Wert als Analogspannung am Pin 8 aus.

Die Spannung am RSSI-Ausgang folgt der Gleichung:

$$P_{RF} [\text{dBm}] = -128 + \frac{U_{RSSI}}{1.5 * 3.3V} * 127$$

Nullpunkt bei -128 dBm, Steigung: 25.7dB/V, Messbereich -120...-60 dBm  
Der RSSI-Ausgang weist einen Innenwiderstand von ca. 10kΩ auf.

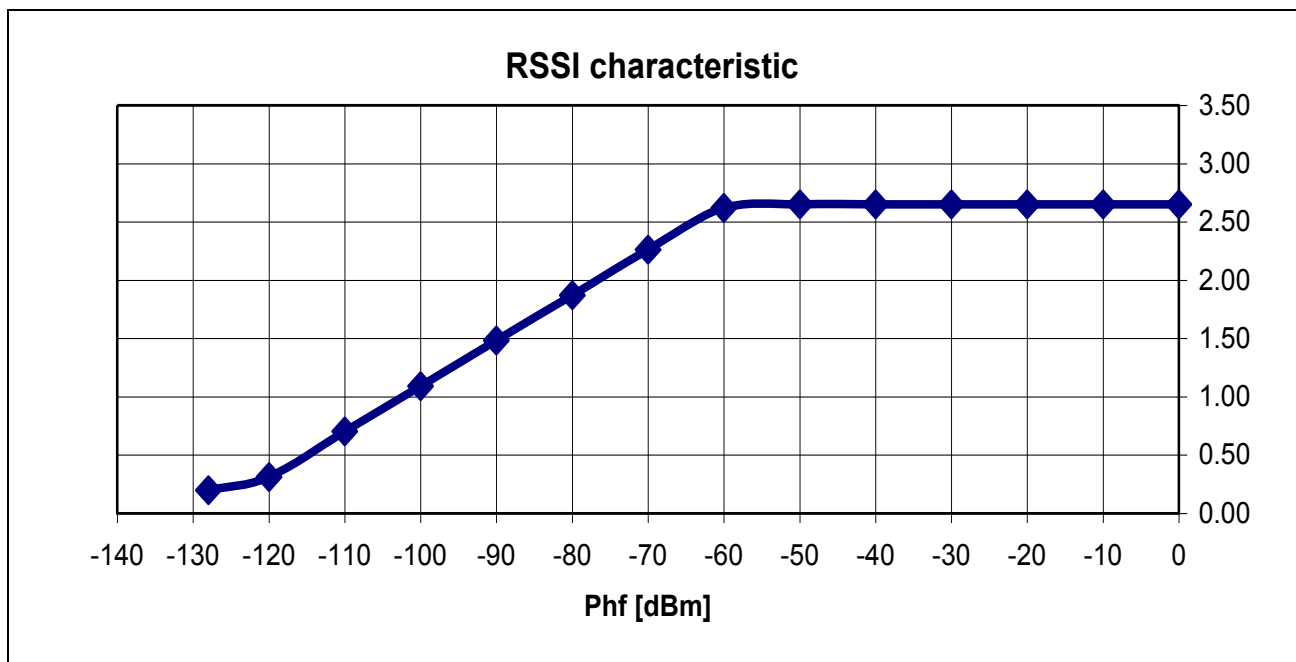


Abbildung 16 Spannung am RSSI-Pin

$P_{RF}$  kann als RSSI Wert auch digital ausgelesen werden mit dem Befehl „READ RSSI aktuell“, wobei der Wert im Zweierkomplement direkt in der Einheit dBm ausgegeben wird.

Beispiele:  $P_{RF} = -128$  dBm ergibt ein RSSI von 128dec bzw. 080H (unter der Rauschgrenze)  
 $P_{RF} = -120$  dBm ergibt ein RSSI von 136dec bzw. 088H  
 $P_{RF} = -110$  dBm ergibt ein RSSI von 146dec bzw. 092H  
 $P_{RF} = -100$  dBm ergibt ein RSSI von 156dec bzw. 09CH  
 $P_{RF} = -60$  dBm ergibt ein RSSI von 196dec bzw. 0C4H (Sättigungsgrenze)

Der Maximalwert des RSSI wird vom Transceiver gespeichert (peak hold) und kann ebenfalls ausgelesen werden mit dem Befehl „READ RSSI peak“. Jeder Lesevorgang des RSSI peak initialisiert dabei den Maximalwert, sodass der Maximalwert die Zeitspanne zwischen zwei Lesevorgängen repräsentiert. Unmittelbar nach einem sleep wird RSSI peak auch initialisiert.

Während dem Sendebetrieb geht RSSI aktuell und RSSI peak auf -128dBm zurück.

## Speisespannung

Der Funktransceiver mit Spannungsregler benötigt eine saubere Speisespannung mit min. 3.5V und einem Ripple von < 10mVpp. Der Sendeverstärker liegt direkt an der eingangsseitigen Speisespannung, die Sendeleistung ist deshalb in geringem Mass von der Speisespannung abhängig. Wenn die Speisung geschaltet wird, so muss die Spannung nach dem Ausschalten immer auf 0V fallen, bevor erneut eingeschaltet wird. Die Speisespannung muss innerhalb von max. 50ms von 0V auf 3.5V ansteigen, damit der integrierte Mikrokontroller korrekt startet. Die Spannung darf zu keinem Zeitpunkt, auch nicht kurzzeitig, unterhalb 3.3 Volt fallen. Wenn dies nicht gewährleistet werden kann, so muss die Speisung mit einem Spannungsdetektor überwacht werden. Sobald diese unterhalb von ca. 3.3V sinkt, muss der Eingang MCLR\ auf low gezogen werden. Siehe auch Applikationsbeispiel mit Demokit3.

Der Funktransceiver ohne Spannungsregler kann von 2.4V bis 3.6V betrieben werden. Hier muss ebenfalls ein Ripple von <10mVpp eingehalten werden. Damit der integrierte Mikrokontroller korrekt startet gelten für die Spannung die Angaben gemäss Datenblatt von Microchip für den 16LF648A.

Bei Betrieb ohne internen Spannungsregler direkt an einer Batterie ist der Innenwiderstand der Batterie zu beachten. Als minimale Batteriegrösse wird die CR2032 empfohlen. Je grösser die Batterie und je tiefer deren Innenwiderstand, desto besser kann Kapazität der Batterie ausgeschöpft werden. Über den Einsatz von Batteriespeisungen ist eine Applikationsschrift geplant.

Anstelle einer geschalteten Speisung sollte die sleep- Funktion verwendet werden, siehe Abschnitt Stromsparmodi.

## Dateninterface mit 5V Systemen

Die Logikpegel des Transceivers entsprechen dem int. Spannungsregler und liegen bei der Standardversion bei typ. 3.3Volt. Der Transceiver kann direkt an 5Volt Systemen arbeiten, dabei fliesst jedoch während dem Highpegel des 5V Systems ein zusätzlicher kleiner Querstrom über die internen Seriewiderstände im Transceiver. Die Eingangs- Highpegel beim 5V System müssen kompatibel sein zur 3.3V Logik. Siehe auch vereinfachtes Schema des Transceivers. Die genauen Pegelgrenzen des internen Mikrokontrollers können dem Datenblatt von Microchip für den 16LF648A entnommen werden. Eventuell ist eine Pegelanpassung nötig, wenn die Pegel der beiden Systeme nicht 100% kompatibel sind.

## MCLR\ Reset

Der Mikrokontroller des Funktransceivers verfügt über keinen internen Resetbaustein, der Reset ist deshalb nach aussen geführt. Für das Verhalten bei Speisespannungen ausserhalb der spezifizierten Werte (d.h. < 3.5V) gelten die Angaben gemäss Datenblatt von Microchip für den 16LF648A. Spezielle Bedingungen wie „brown-out“, d.h. ein Zusammenbrechen der Speisung unterhalb 3.5 Volt, jedoch nicht komplett auf null oder aber langsam ansteigende Betriebsspannungen erfordern einen externen Spannungsdetektor. Die Resetschwelle des Voltagedetectors soll bei einer Spannung von > 3.3V liegen (Standardversion mit int. Spannungsregler). Bei Versionen ohne Spannungsregler und reduzierter  $\mu$ P Clockfrequenz kann die Resetschwelle bis auf 2.3V reduziert werden.

Wenn die unter dem Punkt Speisespannung definierten Bedingungen eingehalten werden, ist kein externer Resetbaustein bzw. Voltagedetector notwendig.

Nach einem Reset am MCLR\ Resetinput benötigt der Transceiver typ. 75ms, bis das Bereitzeichen ausgegeben wird, da nach einem Reset automatisch der Transceiver kalibriert und konfiguriert wird. Das Bereitzeichen wird nur ausgegeben, falls sich das Modul im Konfigurationsmodus befindet.

## Status LED

Der Zustand der Transceiver- LED wird mit dem Befehl **power-control** definiert. Wird dieser Befehl mit gesetztem Bit7 ausgeführt, so wird die LED bereits beim Powerup entsprechend ein- oder ausgeschaltet.

Nach einem **sleep**- Befehl wird die LED immer ausgeschaltet, nach dem wakeup wird die LED auf den Zustand vor dem sleep zurückgesetzt.

## Temperatursensor

Das Transceivermodul verfügt über einen internen Temperatursensor. Die aktuelle Temperatur kann über den Befehl „READ Temperatur“ ausgelesen werden. Der Wertebereich liegt bei  $-40^{\circ}\text{C}$  bis  $+115^{\circ}\text{C}$ . Die Temperatur wird alle 5s aktualisiert, egal wie oft sie ausgelesen wird.

### Hinweis

Der 5s-Timer für die Aktualisierung der Temperatur wird während dem Sleep eingefroren. Nach einem Sleep kann also im Extremfall noch 5s die Temperatur die vor dem Sleep gemessen wurde, angezeigt werden. Diese kann bei langer Sleepdauer von der aktuellen Temperatur abweichen.

**Technische Daten TRX433-10C**

Frequenzbereich	433.0625 .... 434.7875 MHz (12.5 kHz Kanalaraster)				
Mögliche Parametrierung der Funkdatenrate mit zugehörigen fixen Einstellungen (Standardversion bis 19.2kbits/s <sup>8</sup> )	Funkdatenrate	Codierung	Belegte Kanäle	ZF-Bandbreite	FM-Deviation
	1.2 kbit/s	Manchester	1 x 12.5 kHz	9.6 kHz	+/- 2.025 kHz
	2.4 kbit/s	NRZ	1 x 12.5 kHz	9.6 kHz	+/- 2.025 kHz
	4.8 kbit/s	NRZ	2 x 12.5 kHz	19.2 kHz	+/- 4.050 kHz
	9.6 kbit/s	NRZ	4 x 12.5 kHz	25.6 kHz	+/- 4.950 kHz
	19.2 kbit/s	NRZ	8 x 12.5 kHz	51.2 kHz	+/- 9.900 kHz
Empfangsempfindlichkeit bei verschiedener Parametrierung (50 Ohm / BER = 1E-3)	38.4 kbit/s <sup>8</sup>	NRZ	12 x 12.5 kHz	102.4 kHz	+/- 19.80 kHz
Empfangsempfindlichkeit bei verschiedener Parametrierung (50 Ohm / BER = 1E-3)	-120 dBm -118 dBm -115 dBm -113 dBm -110 dBm -106 dBm	Bei Funkdatenrate = 1.2kbit/s Bei Funkdatenrate = 2.4kbit/s Bei Funkdatenrate = 4.8kbit/s Bei Funkdatenrate = 9.6kbit/s Bei Funkdatenrate = 19.2kbit/s Bei Funkdatenrate = 38.4kbit/s <sup>8</sup>			
Frequenzfehler	± 5ppm standard (Temp. -10°C ... +60°C) ± 2ppm optional (Temp. -10°C ... +60°C)				
Sendeleistung	+12 dBm, Speisung 5.0 V +10 dBm, Speisung 3.5 V				
Modulation	GFSK				
Funkreichweite bei freier Sicht <sup>9</sup>	2000m	Bei Funkdatenrate = 1.2kbit/s			
Leistungsfähiger Befehlssatz unterstützt Funktionen wie:	Power-control, definiert Verhalten bei Powerup, sleep, standby Kalibration, RSSI-Ausgabe, Modultemperatur auslesen, Register lesen und schreiben, direkter EEPROM-Zugriff (lesen, schreiben), Seriennummer, Hardware, Softwareversion usw.				
Datenschnittstelle	seriell über RS232 mit TTL Pegel, 8 Datenbits, 1 Stopbit, no parity Baudraten: 1.2 / 2.4 / 4.8 / 9.6 / 19.2 / 38.4 / 57.6 / 115.2 <sup>10</sup> kbaud automatische Baudratenerkennung bei Konfiguration Handshake mit RTS/CTS 61 Bytes Sende- und Empfangsbuffer				
Speisung	3.5 bis 6 V DC unstabilisiert, Ripple <10mVpp (Version mit Spannungsregler) 2.4 bis 3.6V ab Batterie, Ripple <10mVpp (Version ohne Spannungsregler)				
Stromverbrauch	25mA Empfang und Kalibration (typ.) 35mA Senden, (typ.) Speisung 3.5V 50mA Senden, (max.) Speisung 5.0V 0.1mA sleep (Version mit Spannungsregler, 2ms Wakeupzeit nach sleep) 3uA sleep (Version ohne Spannungsregler, 2ms Wakeupzeit nach sleep)				
Mikrokontroller	PIC 16LF648A von Microchip, Flash, incircuit-programmierbar				
Funkchipsatz	Chipcon CC1020				
Spezielles	Statusanzeige mit LED, integrierter Temperatursensor				
Montage	horizontal oder vertikal bestückbar Leadframe im 2.54mm Raster, (SMD bestückbar auf Anfrage)				
Modulabmessungen	40.0 x 25.0 x 4.5mm (ohne Anschlusspins)				
Modulgewicht	7.0 g 10.0 g vergossene Version auf Anfrage				

<sup>8</sup> Funkdatenraten > 19.2kbits/s erfordern eine Modifikation des Loopfilters, also eine Bestückungsänderung und eine 16MHz µP Clockfrequenz

<sup>9</sup> Siehe auch Abschnitt Funkreichweite

<sup>10</sup> 115.2kbaud erfordern eine 16MHz µP Clockfrequenz

## Technische Daten TRX868-10C

Frequenzbereich	868.0125 .... 869.9875 MHz (12.5 kHz Kanalaraster)				
Mögliche Parametrierung der Funkdatenrate mit zugehörigen fixen Einstellungen  (Standardversion bis 19.2kbits/s <sup>11</sup> )	Funkdatenrate	Codierung	Belegte Kanäle	ZF-Bandbreite	FM-Deviation
	1.2 kbit/s	Manchester	1 x 12.5 kHz	9.6 kHz	+/- 2.025 kHz
	2.4 kbit/s	NRZ	1 x 12.5 kHz	9.6 kHz	+/- 2.025 kHz
	4.8 kbit/s	NRZ	2 x 12.5 kHz	19.2 kHz	+/- 4.050 kHz
	9.6 kbit/s	NRZ	4 x 12.5 kHz	25.6 kHz	+/- 4.950 kHz
	19.2 kbit/s	NRZ	8 x 12.5 kHz	51.2 kHz	+/- 9.900 kHz
	38.4 kbit/s <sup>11</sup>	NRZ	12 x 12.5 kHz	102.4 kHz	+/- 19.80 kHz
Empfangsempfindlichkeit (50 Ohm / BER = 1E-3)	-115 dBm	Bei Funkdatenrate = 1.2kbit/s			
Frequenzfehler	± 5ppm standard (Temp. -10°C ... +60°C) ± 2ppm optional (Temp. -10°C ... +60°C)				
Sendeleistung	+10 dBm, Speisung 5.0 V + 8 dBm, Speisung 3.5 V				
Modulation	GFSK				
Funkreichweite bei freier Sicht <sup>12</sup>	1200m	Bei Funkdatenrate = 1.2kbit/s			
Leistungsfähiger Befehlssatz unterstützt Funktionen wie:	Power-control, definiert Verhalten bei Powerup, sleep, standby Kalibration, RSSI-Ausgabe, Modultemperatur auslesen, Register lesen und schreiben, direkter EEPROM-Zugriff (lesen, schreiben) Seriennummer, Hardware, Softwareversion usw.				
Datenschnittstelle	seriell über RS232 mit TTL Pegel, 8 Datenbits, 1 Stopbit, no parity Baudraten: 1.2 / 2.4 / 4.8 / 9.6 / 19.2 / 38.4 / 57.6 / 115.2 <sup>13</sup> kbaud automatische Baudratenerkennung bei Konfiguration Handshake mit RTS/CTS 61 Bytes Sende- und Empfangsbuffer				
Speisung	3.5 bis 6 V DC unstabilisiert, Ripple <10mVpp (Version mit Spannungsregler) 2.4 bis 3.6V ab Batterie, Ripple <10mVpp (Version ohne Spannungsregler)				
Stromverbrauch	25mA Empfang und Kalibration (typ.) 35mA Senden, (typ.) Speisung 3.5V 50mA Senden, (max.) Speisung 5.0V 0.1mA sleep (Version mit Spannungsregler, 2ms Wakeupzeit nach sleep) 3uA sleep (Version ohne Spannungsregler, 2ms Wakeupzeit nach sleep)				
Mikrokontroller	PIC 16LF648A von Microchip, Flash, incircuit-programmierbar				
Funkchipsatz	Chipcon CC1020				
Spezielles	Statusanzeige mit LED, integrierter Temperatursensor				
Montage	horizontal oder vertikal bestückbar Leadframe im 2.54mm Raster, (SMD bestückbar auf Anfrage)				
Modulabmessungen	40.0 x 25.0 x 4.5mm (ohne Anschlusspins)				
Modulgewicht	7.0 g 10.0 g vergossene Version auf Anfrage				

<sup>11</sup> Funkdatenraten > 19.2kbits/s erfordern eine Modifikation des Loopfilters, also eine Bestückungsänderung und eine 16MHz µP Clockfrequenz

<sup>12</sup> Siehe auch Abschnitt Funkreichweite

<sup>13</sup> 115.2kbaud erfordern eine 16MHz µP Clockfrequenz

## **Funkreichweite**

Die Funkreichweite bei einer bestimmten Funkdatenrate hängt von vielen Faktoren ab, welche vom Einsatzstandort abhängen.

Wesentliche Faktoren sind: Höhe der Antenne über Boden, Störquellen am Empfängerstandort (PC's, Monitore usw.), andere Funksender oder Funktelefone, die Geländeart, Funksender im gleichen Band auf Nachbarkanälen.

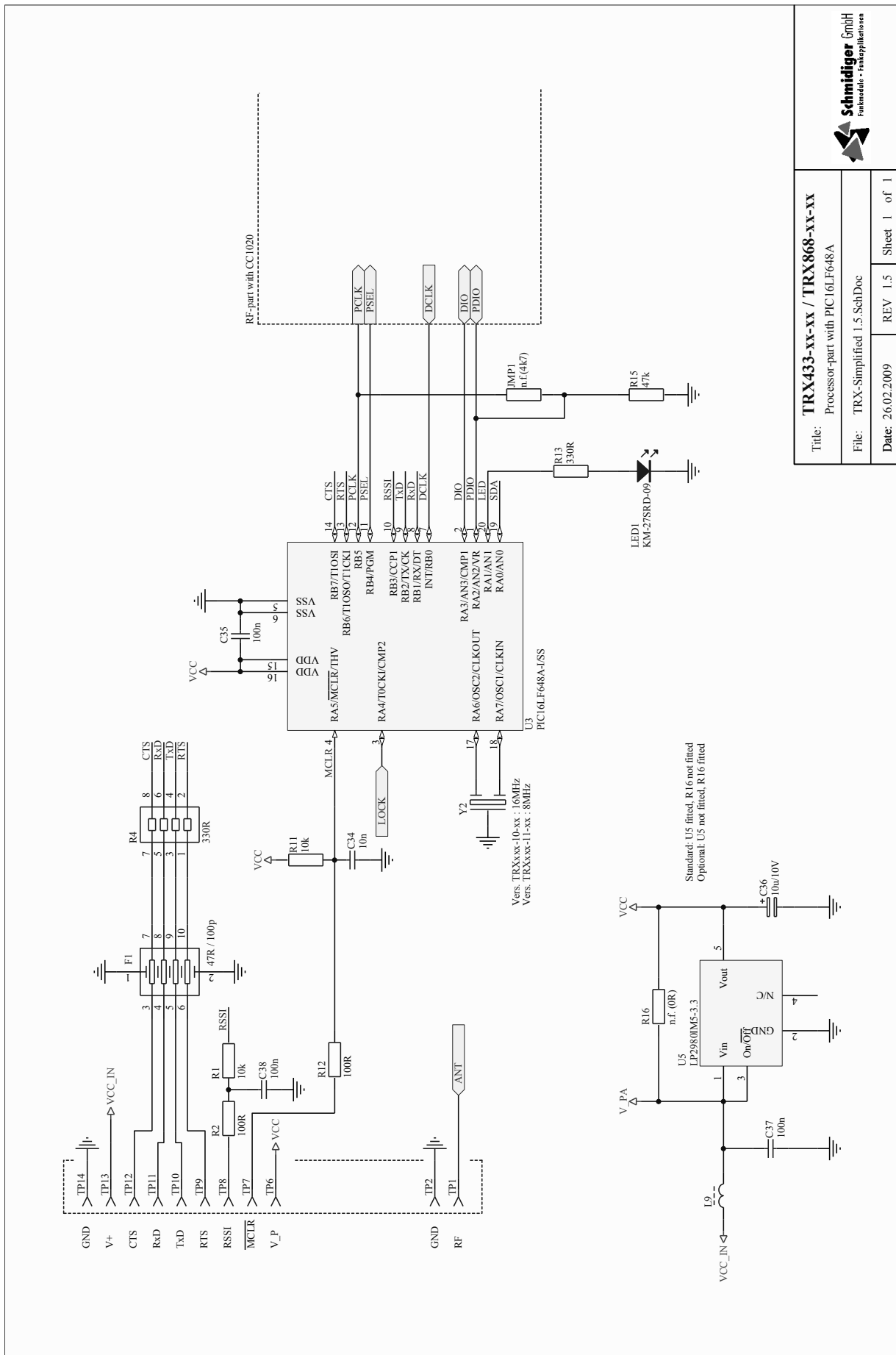
Unter normalen Bedingungen wurden mit dem Demokit3 bei 433 MHz und der kleinsten Funkdatenrate Reichweiten von über 4km im hügeligen Gelände gemessen, die angegebenen Werte sind also eher konservativ angegeben.

Wenn maximale Reichweiten im Gelände wichtig sind, wird das 433 MHz Frequenzband empfohlen. Wenn am Einsatzort bereits Systeme mit 433 MHz im Betrieb sind, können bei 868MHz an diesem Standort eventuell höhere Reichweiten erzielt werden als bei 433 MHz.

Die in den technischen Daten angegebene Funkreichweite ist konservativ angegeben und mit dem Demokit3 im hügeligen Gelände ermittelt.



### Vereinfachtes Schema TRXnnn-10



**Schmidiger GmbH**  
Fernstudie • Fernkopplösungen

---

**Title:** TRX433-xx-xx / TRX868-xx-xx  
Processor-part with PIC16LF648A

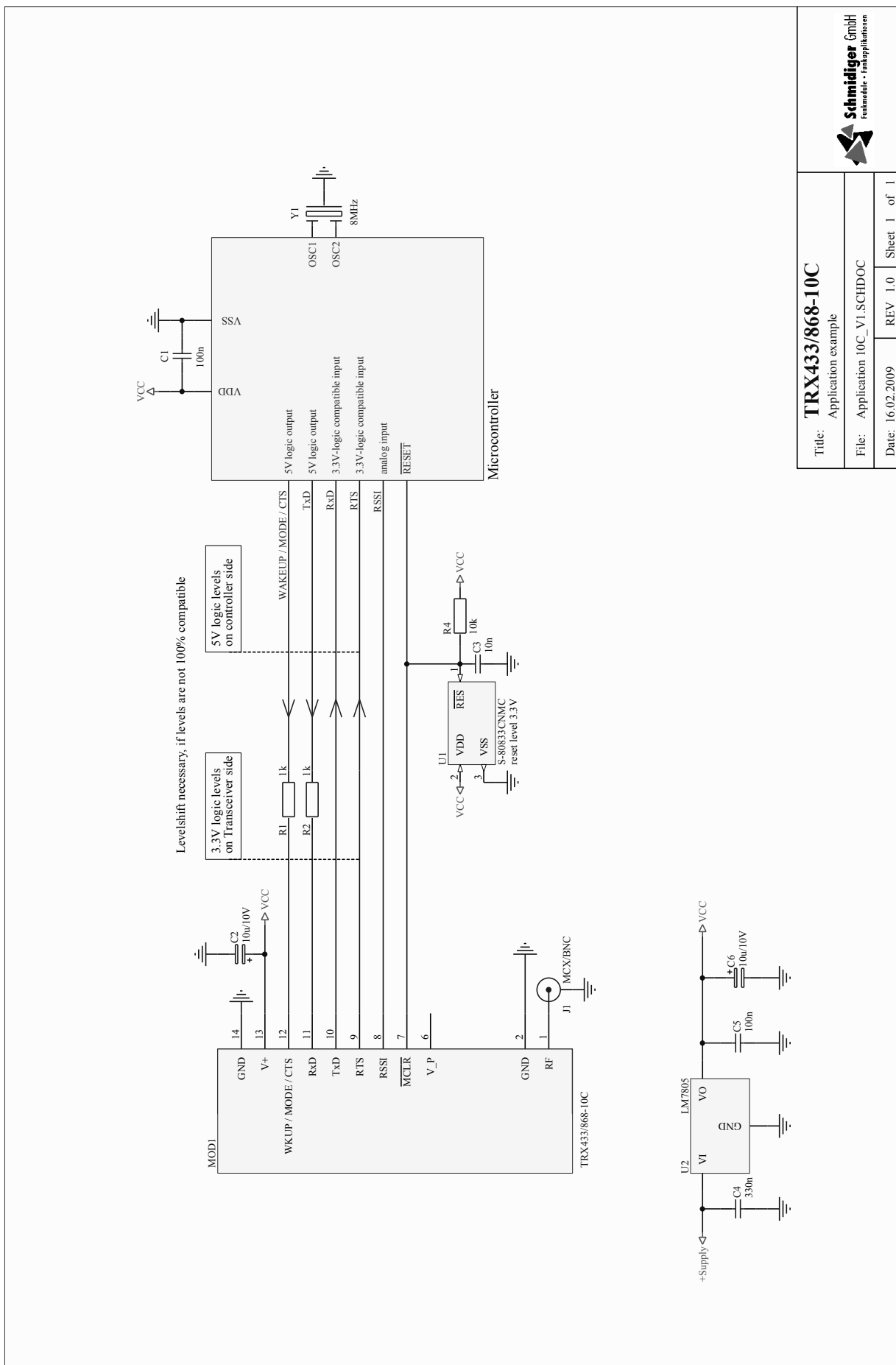
---

**File:** TRX-Simplified 1.5.SchDoc

---

**Date:** 26.02.2009      **REV** 1.5      **Sheet** 1 of 1

## Anwendungsbeispiel für TRXnnn-10C



Title: TRX433/868-10C

Application example

File: Application 10C\_V1.SCHDOC

Date: 16.02.2009

REV 1.0

Sheet 1 of 1



Schmidiger GmbH  
Fernstudie • Funkapplikationen

### Frequenztabelle TRX433-10C

<b>1</b>	433.0625 MHz	<b>36</b>	433.5000 MHz	<b>71</b>	433.9375 MHz	<b>106</b>	434.3750 MHz
<b>2</b>	433.0750 MHz	<b>37</b>	433.5125 MHz	<b>72</b>	433.9500 MHz	<b>107</b>	434.3875 MHz
<b>3</b>	433.0875 MHz	<b>38</b>	433.5250 MHz	<b>73</b>	433.9625 MHz	<b>108</b>	434.4000 MHz
<b>4</b>	433.1000 MHz	<b>39</b>	433.5375 MHz	<b>74</b>	433.9750 MHz	<b>109</b>	434.4125 MHz
<b>5</b>	433.1125 MHz	<b>40</b>	433.5500 MHz	<b>75</b>	433.9875 MHz	<b>110</b>	434.4250 MHz
<b>6</b>	433.1250 MHz	<b>41</b>	433.5625 MHz	<b>76</b>	434.0000 MHz	<b>111</b>	434.4375 MHz
<b>7</b>	433.1375 MHz	<b>42</b>	433.5750 MHz	<b>77</b>	434.0125 MHz	<b>112</b>	434.4500 MHz
<b>8</b>	433.1500 MHz	<b>43</b>	433.5875 MHz	<b>78</b>	434.0250 MHz	<b>113</b>	434.4625 MHz
<b>9</b>	433.1625 MHz	<b>44</b>	433.6000 MHz	<b>79</b>	434.0375 MHz	<b>114</b>	434.4750 MHz
<b>10</b>	433.1750 MHz	<b>45</b>	433.6125 MHz	<b>80</b>	434.0500 MHz	<b>115</b>	434.4875 MHz
<b>11</b>	433.1875 MHz	<b>46</b>	433.6250 MHz	<b>81</b>	434.0625 MHz	<b>116</b>	434.5000 MHz
<b>12</b>	433.2000 MHz	<b>47</b>	433.6375 MHz	<b>82</b>	434.0750 MHz	<b>117</b>	434.5125 MHz
<b>13</b>	433.2125 MHz	<b>48</b>	433.6500 MHz	<b>83</b>	434.0875 MHz	<b>118</b>	434.5250 MHz
<b>14</b>	433.2250 MHz	<b>49</b>	433.6625 MHz	<b>84</b>	434.1000 MHz	<b>119</b>	434.5375 MHz
<b>15</b>	433.2375 MHz	<b>50</b>	433.6750 MHz	<b>85</b>	434.1125 MHz	<b>120</b>	434.5500 MHz
<b>16</b>	433.2500 MHz	<b>51</b>	433.6875 MHz	<b>86</b>	434.1250 MHz	<b>121</b>	434.5625 MHz
<b>17</b>	433.2625 MHz	<b>52</b>	433.7000 MHz	<b>87</b>	434.1375 MHz	<b>122</b>	434.5750 MHz
<b>18</b>	433.2750 MHz	<b>53</b>	433.7125 MHz	<b>88</b>	434.1500 MHz	<b>123</b>	434.5875 MHz
<b>19</b>	433.2875 MHz	<b>54</b>	433.7250 MHz	<b>89</b>	434.1625 MHz	<b>124</b>	434.6000 MHz
<b>20</b>	433.3000 MHz	<b>55</b>	433.7375 MHz	<b>90</b>	434.1750 MHz	<b>125</b>	434.6125 MHz
<b>21</b>	433.3125 MHz	<b>56</b>	433.7500 MHz	<b>91</b>	434.1875 MHz	<b>126</b>	434.6250 MHz
<b>22</b>	433.3250 MHz	<b>57</b>	433.7625 MHz	<b>92</b>	434.2000 MHz	<b>127</b>	434.6375 MHz
<b>23</b>	433.3375 MHz	<b>58</b>	433.7750 MHz	<b>93</b>	434.2125 MHz	<b>128</b>	434.6500 MHz
<b>24</b>	433.3500 MHz	<b>59</b>	433.7875 MHz	<b>94</b>	434.2250 MHz	<b>129</b>	434.6625 MHz
<b>25</b>	433.3625 MHz	<b>60</b>	433.8000 MHz	<b>95</b>	434.2375 MHz	<b>130</b>	434.6750 MHz
<b>26</b>	433.3750 MHz	<b>61</b>	433.8125 MHz	<b>96</b>	434.2500 MHz	<b>131</b>	434.6875 MHz
<b>27</b>	433.3875 MHz	<b>62</b>	433.8250 MHz	<b>97</b>	434.2625 MHz	<b>132</b>	434.7000 MHz
<b>28</b>	433.4000 MHz	<b>63</b>	433.8375 MHz	<b>98</b>	434.2750 MHz	<b>133</b>	434.7125 MHz
<b>29</b>	433.4125 MHz	<b>64</b>	433.8500 MHz	<b>99</b>	434.2875 MHz	<b>134</b>	434.7250 MHz
<b>30</b>	433.4250 MHz	<b>65</b>	433.8625 MHz	<b>100</b>	434.3000 MHz	<b>135</b>	434.7375 MHz
<b>31</b>	433.4375 MHz	<b>66</b>	433.8750 MHz	<b>101</b>	434.3125 MHz	<b>136</b>	434.7500 MHz
<b>32</b>	433.4500 MHz	<b>67</b>	433.8875 MHz	<b>102</b>	434.3250 MHz	<b>137</b>	434.7625 MHz
<b>33</b>	433.4625 MHz	<b>68</b>	433.9000 MHz	<b>103</b>	434.3375 MHz	<b>138</b>	434.7750 MHz
<b>34</b>	433.4750 MHz	<b>69</b>	433.9125 MHz	<b>104</b>	434.3500 MHz	<b>139</b>	434.7875 MHz
<b>35</b>	433.4875 MHz	<b>70</b>	433.9250 MHz	<b>105</b>	434.3625 MHz		

#### Hinweis

Zwei benachbarte Transceiver, die unabhängig voneinander arbeiten, müssen sich im Kanal mindestens um die Anzahl belegter Kanäle gemäss Frequenzraster unterscheiden, damit sich die Frequenzspektren nicht überlappen.

Beispiel: Bei der Einstellung mit Funkdatenrate 9.6kbit/s werden 4 Kanäle belegt (siehe Abschnitt Funkdatenrate). Somit müssen benachbarte Transceiver mindestens 4 Kanäle Differenz aufweisen, wenn diese mit identischen Kommunikationseinstellungen arbeiten.

Je grösser der Kanalabstand gewählt wird, desto besser ist die Reichweite bei gleichzeitig mehreren aktiven Transceivern.

## Frequenztabelle TRX868-10C

<b>1</b>	868.0125 MHz
<b>2</b>	868.0250 MHz
<b>3</b>	868.0375 MHz
<b>4</b>	868.0500 MHz
<b>5</b>	868.0625 MHz
<b>6</b>	868.0750 MHz
<b>7</b>	868.0875 MHz
<b>8</b>	868.1000 MHz
<b>9</b>	868.1125 MHz
<b>10</b>	868.1250 MHz
<b>11</b>	868.1375 MHz
<b>12</b>	868.1500 MHz
<b>13</b>	868.1625 MHz
<b>14</b>	868.1750 MHz
<b>15</b>	868.1875 MHz
<b>16</b>	868.2000 MHz
<b>17</b>	868.2125 MHz
<b>18</b>	868.2250 MHz
<b>19</b>	868.2375 MHz
<b>20</b>	868.2500 MHz
<b>21</b>	868.2625 MHz
<b>22</b>	868.2750 MHz
<b>23</b>	868.2875 MHz
<b>24</b>	868.3000 MHz
<b>25</b>	868.3125 MHz
<b>26</b>	868.3250 MHz
<b>27</b>	868.3375 MHz
<b>28</b>	868.3500 MHz
<b>29</b>	868.3625 MHz
<b>30</b>	868.3750 MHz
<b>31</b>	868.3875 MHz
<b>32</b>	868.4000 MHz
<b>33</b>	868.4125 MHz
<b>34</b>	868.4250 MHz
<b>35</b>	868.4375 MHz
<b>36</b>	868.4500 MHz
<b>37</b>	868.4625 MHz
<b>38</b>	868.4750 MHz
<b>39</b>	868.4875 MHz
<b>40</b>	868.5000 MHz

<b>41</b>	868.5125 MHz
<b>42</b>	868.5250 MHz
<b>43</b>	868.5375 MHz
<b>44</b>	868.5500 MHz
<b>45</b>	868.5625 MHz
<b>46</b>	868.5750 MHz
<b>47</b>	868.5875 MHz
<b>48</b>	868.6000 MHz
<b>49</b>	868.6125 MHz
<b>50</b>	868.6250 MHz
<b>51</b>	868.6375 MHz
<b>52</b>	868.6500 MHz
<b>53</b>	868.6625 MHz
<b>54</b>	868.6750 MHz
<b>55</b>	868.6875 MHz
<b>56</b>	868.7000 MHz
<b>57</b>	868.7125 MHz
<b>58</b>	868.7250 MHz
<b>59</b>	868.7375 MHz
<b>60</b>	868.7500 MHz
<b>61</b>	868.7625 MHz
<b>62</b>	868.7750 MHz
<b>63</b>	868.7875 MHz
<b>64</b>	868.8000 MHz
<b>65</b>	868.8125 MHz
<b>66</b>	868.8250 MHz
<b>67</b>	868.8375 MHz
<b>68</b>	868.8500 MHz
<b>69</b>	868.8625 MHz
<b>70</b>	868.8750 MHz
<b>71</b>	868.8875 MHz
<b>72</b>	868.9000 MHz
<b>73</b>	868.9125 MHz
<b>74</b>	868.9250 MHz
<b>75</b>	868.9375 MHz
<b>76</b>	868.9500 MHz
<b>77</b>	868.9625 MHz
<b>78</b>	868.9750 MHz
<b>79</b>	868.9875 MHz
<b>80</b>	869.0000 MHz

<b>81</b>	869.0125 MHz
<b>82</b>	869.0250 MHz
<b>83</b>	869.0375 MHz
<b>84</b>	869.0500 MHz
<b>85</b>	869.0625 MHz
<b>86</b>	869.0750 MHz
<b>87</b>	869.0875 MHz
<b>88</b>	869.1000 MHz
<b>89</b>	869.1125 MHz
<b>90</b>	869.1250 MHz
<b>91</b>	869.1375 MHz
<b>92</b>	869.1500 MHz
<b>93</b>	869.1625 MHz
<b>94</b>	869.1750 MHz
<b>95</b>	869.1875 MHz
<b>96</b>	869.2000 MHz
<b>97</b>	869.2125 MHz
<b>98</b>	869.2250 MHz
<b>99</b>	869.2375 MHz
<b>100</b>	869.2500 MHz
<b>101</b>	869.2625 MHz
<b>102</b>	869.2750 MHz
<b>103</b>	869.2875 MHz
<b>104</b>	869.3000 MHz
<b>105</b>	869.3125 MHz
<b>106</b>	869.3250 MHz
<b>107</b>	869.3375 MHz
<b>108</b>	869.3500 MHz
<b>109</b>	869.3625 MHz
<b>110</b>	869.3750 MHz
<b>111</b>	869.3875 MHz
<b>112</b>	869.4000 MHz
<b>113</b>	869.4125 MHz
<b>114</b>	869.4250 MHz
<b>115</b>	869.4375 MHz
<b>116</b>	869.4500 MHz
<b>117</b>	869.4625 MHz
<b>118</b>	869.4750 MHz
<b>119</b>	869.4875 MHz
<b>120</b>	869.5000 MHz

<b>121</b>	869.5125 MHz
<b>122</b>	869.5250 MHz
<b>123</b>	869.5375 MHz
<b>124</b>	869.5500 MHz
<b>125</b>	869.5625 MHz
<b>126</b>	869.5750 MHz
<b>127</b>	869.5875 MHz
<b>128</b>	869.6000 MHz
<b>129</b>	869.6125 MHz
<b>130</b>	869.6250 MHz
<b>131</b>	869.6375 MHz
<b>132</b>	869.6500 MHz
<b>133</b>	869.6625 MHz
<b>134</b>	869.6750 MHz
<b>135</b>	869.6875 MHz
<b>136</b>	869.7000 MHz
<b>137</b>	869.7125 MHz
<b>138</b>	869.7250 MHz
<b>139</b>	869.7375 MHz
<b>140</b>	869.7500 MHz
<b>141</b>	869.7625 MHz
<b>142</b>	869.7750 MHz
<b>143</b>	869.7875 MHz
<b>144</b>	869.8000 MHz
<b>145</b>	869.8125 MHz
<b>146</b>	869.8250 MHz
<b>147</b>	869.8375 MHz
<b>148</b>	869.8500 MHz
<b>149</b>	869.8625 MHz
<b>150</b>	869.8750 MHz
<b>151</b>	869.8875 MHz
<b>152</b>	869.9000 MHz
<b>153</b>	869.9125 MHz
<b>154</b>	869.9250 MHz
<b>155</b>	869.9375 MHz
<b>156</b>	869.9500 MHz
<b>157</b>	869.9625 MHz
<b>158</b>	869.9750 MHz
<b>159</b>	869.9875 MHz

### Hinweis

Zwei benachbarte Transceiver, die unabhängig voneinander arbeiten, müssen sich im Kanal mindestens um die Anzahl belegter Kanäle gemäss Frequenzraster unterscheiden, damit sich die Frequenzspektren nicht überlappen.

Beispiel: Bei der Einstellung mit Funkdatenrate 9.6kbit/s werden 4 Kanäle belegt (siehe Abschnitt Funkdatenrate). Somit müssen benachbarte Transceiver mindestens 4 Kanäle Differenz aufweisen, wenn diese mit identischen Kommunikationseinstellungen arbeiten.

Je grösser der Kanalabstand gewählt wird, desto besser ist die Reichweite bei gleichzeitig mehreren aktiven Transceivern.

## Befehlssatz zur Konfiguration (Version C2)

Um den Transceiver zu konfigurieren, muss der MODE-Pin auf High gesetzt werden. Dadurch wird der Transceiver in den Konfigurationsmodus versetzt und kann über die serielle Schnittstelle (Pin 10, TxD und Pin 11, RxD) konfiguriert werden.

Das serielle Datenformat ist N,8,1 (no Parity, 8 Databits, 1 Stopbit). Im Konfigurationsmodus kann die Baudrate beliebig gewählt werden, da der Transceiver folgende Baudraten automatisch erkennt: 1.2kBaud, 2.4kBaud, 4.8kBaud, 9.6kBaud, 19.2kBaud, 38.4kBaud, 57.6kBaud und 115.2kBaud (115.2kBaud erfordert eine 16MHz  $\mu$ P Clockfrequenz).

### Befehlsstruktur

Ein Befehl besteht immer aus 3 Bytes:

Befehl Byte 1	Befehl Byte 2	Befehl Byte 3
Startzeichen [0B0H]	Funktion [00H...0FFH]	Parameter oder Wert [00H...0FFH]

Ein korrekt empfangener Befehl wird vom Transceiver innerhalb von 2ms (bzw. 10ms bei Befehlen, die ins EEPROM gespeichert werden) mit einer Antwort quittiert, falls sich der Transceiver zum Zeitpunkt der Antwort immer noch im Konfigurationsmodus befindet (MODE-Pin auf High). Die Antwort besteht meistens aus zwei Bytes, kann aber je nach Befehl auch mehr Daten enthalten:

Antwort Byte 1	Antwort Byte 2	Antwort Byte 3.....n
Funktion bzw. Echo von Befehl Byte 2	Wert [00H...0FFH]	Wert [00H...0FFH] Anzahl Bytes je nach Funktion

### Startzeichen

Das Startzeichen kennzeichnet den Beginn einer Befehlssequenz und muss den Wert 0B0H aufweisen.

### Hinweise:

- Damit ein Befehl als solcher erkannt wird, müssen die drei Bytes innerhalb von max. 200ms zum Transceiver übertragen werden.
- Erkennt der Transceiver keinen gültigen Befehl oder einen gültigen Befehl mit falschem Parameter so gibt er einen Fehlercode in der verwendeten Baudrate zurück. Siehe auch Abschnitt Fehlercodes.

### Konfigurationen im RAM oder EEPROM

Bei folgenden Ereignissen startet der Transceiver mit der im EEPROM gespeicherten Konfiguration (RS232 Baudrate, Funkfrequenz, Funkdatenrate usw.).

- Powerup
- MCLR\ Reset
- Wakeup nach sleep, wenn Reset nach sleep konfiguriert ist

Danach können mit der WRITE-Funktion beliebige Parameter geändert werden. Die Änderung wird nach dem Befehl sofort aktiv.

Wenn bei den WRITE- Befehlen 08H..1FH das Bit7 der Funktion (Byte 2) gesetzt ist, wird die Konfiguration sowohl im RAM wie auch zusätzlich im EEPROM dauerhaft gespeichert. Ist das Bit7 gelöscht, wird die Konfiguration nur im RAM gespeichert und steht nach einem Powerup nicht mehr zur Verfügung. Bei allen anderen Befehlen (ausserhalb 08H..1FH) hat das Bit7 keine Bedeutung und wird deshalb ignoriert.

Die Antwort auf einen Befehl wird immer mit gelöschtem Bit7 zurückgesendet, egal ob beim gesendeten Befehl das Bit7 gesetzt war oder nicht.

**Wichtiger Hinweis:** Das EEPROM hat eine begrenzte Anzahl von 100'000 Schreibzyklen pro Parameter. Deshalb dürfen häufig wechselnde Parameter (z.B. Frequenzwechsel bei Frequency- hopping Systemen) nur ins RAM gespeichert werden.

## Befehlsübersicht Version C2

Die Funktionen sind gegliedert nach den Gruppen READ, WRITE, REPORT und ERROR.

Funktionsgruppe	Byte 2	Befehl oder Funktion	Byte3	Beschreibung	Antwort (Beispiel)	Anzahl Bytes	
	Bit0..6						
READ	00H	Interne Konstanten	00H	Softwareversion	00H, 02H, 28H, 07H	4	
			01H	Softwaretyp	00H, 03H	2	
			02H	Befehlssatzversion	00H, 02H	2	
			03H	Hardwareversion	00H, 0AH	2	
			04H	Frequenzversion	00H, 01H	2	
			05H	Seriennummer 32bit	00H, 12H, 34H, 56H, 78H	5	
	01H	Messwerte	00H	RSSI aktuell	01H, 90H	2	
			01H	RSSI peak	01H, A3H	2	
			02H	Temperatur	01H, 18H	2	
	02H	Konfiguration (aus EEPROM)	09H	Empfangsfrequenz	02H, 01H	2	
			0AH	Sendefrequenz	02H, 01H	2	
			0BH	Funkdatenrate	02H, 04H	2	
			0CH	RS232-Baudrate	02H, 04H	2	
			0DH	Powerup mode	02H, 20H	2	
			0EH	Power control	02H, 03H	2	
			0FH	Automat. Rekalibration	02H, 3CH	2	
10H			Data Format	02H, 03H	2		
03H	EEPROM Register	00H- FFH	EEPROM Register lesen an dieser Adresse	03H, 55H	2		
04H – 07H	Reserviert für künftige Befehlsweiterungen						
WRITE	Nur ins RAM, falls Bit7=0 in Byte2  Ins RAM und ins EEPROM, falls Bit7=1 von Byte2  Byte2 ist identisch mit Byte3 der ent- sprechen den READ Funktion	08H	Frequenz RX+TX	01H- 9FH	Frequenzkanal TX und RX	08H, 01H	2
		09H	Frequenz RX	01H- 9FH	Frequenzkanal nur RX	09H, 01H	2
		0AH	Frequenz TX	01H- 9FH	Frequenzkanal nur TX	0AH, 01H	2
		0BH	Funkdatenrate	00H- 05H	Funkdatenrate	0BH, 04H	2
		0CH	RS232-Baudrate	00H- 07H	RS232 Baudrate	0CH, 04H	2
		0DH	Powerup mode	00H- 25H	Konfiguration für powerup, sleep und wakeup	0DH, 20H	2
		0EH	Power control	00H- 03H	LED und HF ein/ausschalten	0EH, 03H	2
		0FH	Automat. Rekalibration	00H- 96H	Temperatur- und zeitgesteuerte Kalibration	0FH, 3CH	2
		10H	Data Format	00H- 1FH	Daten- und Funktelegrammformat	10H, 03H	2
11H – 1FH	Reserviert für künftige Befehlsweiterungen						

Funktions- gruppe	Byte2 Bit0..6	Befehl oder Funktion	Byte3	Beschreibung	Antwort (Beispiel)	Anzahl Bytes
	WRITE  Nur ins RAM möglich, nicht ins EEPROM	20H	Sleep	00H- 21H	Definiert sleep und wakeup	20H, 21H
21H		Jetzt rekalisieren	00H	Start VCO-Kalibration wenn Kalibrat. erfolgreich wenn Kalibrat. fehlerhaft	21H, 00H 21H, 01H (typ. 50ms) 21H, 02H (typ. 250ms)	2 2 2
22H- 2CH		Reserviert für künftige Befehlsweiterungen				
2DH		EEPROM WR-enable	00H- FFH	Einzelne Adresse zum schreiben freigeben	2DH, 10H	2
2EH		EEPROM Data	00H- FFH	Dieses Datenbyte ins EEPROM schreiben	2E, 43H	2
2FH		EEPROM Adresse	00H- FFH	EEPROM Adresse, die beschrieben wird	2EH, 10H	2
REPORT  Automat. Ausgabe von Werten		READY. Bereitzeichen des Transceivers nach powerup, MCLR\ - Reset oder wakeup		Kalibrationswerte OK, nach powerup oder MCLR\ - Reset	30H, 00H	2
				Kalibrationswerte Fehler, nach powerup oder MCLR\ - Reset	30H, 80H	2
				Bereit nach wakeup	30H, 01H	2
ERROR		Antworten auf ungültige Funktionen oder Funktionsparameter		Ungültige Funktion (Byte2)	38H, 00H	2
				Ungültiger Funktionswert (Byte3)	38H, 01H	2

## Initialisierungsbeispiel in C

Das folgende Beispiel zeigt die Anwendung des Befehlsatzes in der Programmiersprache C. Es wird eine Initialisierungssequenz aufgezeigt, welche den Transceiver in die gewünschte Betriebsart konfiguriert. Die Konfiguration wird im EEPROM des Transceivers gespeichert, da Bit7 des Befehls = 1 ist. Beim nächsten Powerup steht dann diese Einstellung sofort ohne erneute Konfiguration zur Verfügung.

Beim Beschreiben des EEPROM's muss darauf geachtet werden, dass die begrenzte Anzahl von 100'000 Schreibzyklen pro Parameter nicht überschritten wird. Wird allerdings versucht einen Wert ins EEPROM zu schreiben, der schon im EEPROM steht, wird der Schreibbefehl nicht ausgeführt. Somit kann also eine sich nicht ändernde Konfiguration beliebig oft ins EEPROM geschrieben werden. Der effektive Schreibbefehl wird dabei nur bei der ersten Konfiguration ausgeführt. Wird die Konfiguration aber häufig gewechselt, so soll diese ausschliesslich ins RAM erfolgen, damit die maximale Anzahl Schreibzyklen des EEPROM's nicht überschritten wird. Dazu ist im nachfolgenden Code jeweils Bit7 des Befehlsbytes zu löschen.

Am Ende der Initialisierung wird nochmals eine VCO-Kalibration gemacht, um das Ergebnis der Kalibration prüfen zu können. Bei der VCO-Kalibration, welche automatisch bei jedem Powerup gemacht wird, würde ein fünfmaliger Kalibrationsfehler keine Fehlermeldung ergeben.



```

int error_code; // Globale Variable für Fehler-Code

// -----
// Initialisierung eines TRX433-10-C2 direkt ins EEPROM (beschränkte Speicherzyklen beachten).
// Für Initialisierung nur ins RAM Bit7 des Befehlsbytes auf null setzen.
// Die Speisung des Transceivermoduls wird kurz vor oder zu Beginn der Initialisierung eingeschaltet.
//
// input: -

// output: error_code: 0 = Komplette Initialisierung erfolgreich
//                  1 = Kein Ready-Report empfangen oder Kalibrationskonstanten fehlerhaft
//                  2 = Fehler in RS232 Kommunikation
//                  3 = Fehler bei VCO-Kalibration

int init_trx()
{
    error_code = 0; // Init Fehler-Code auf erfolgreich

    // hier wird die Speisung des Transceivers eingeschaltet
    // danach muss sich das Modul mit dem REPORT READY melden

    if(!echo(0x30, 0x00)){ // Ready-Report empfangen (Ready-Report wird 170-180ms nach
                          // Powerup des Transceivermoduls gesendet). Die Funktion echo() liest zwei
                          // Bytes über RS232 ein und prüft, ob diese mit den gewünschten zwei Bytes
                          // übereinstimmen. Sie beinhaltet eine Timeoutfunktion, damit bei
                          // ausbleibender Antwort vom Modul die Funktion nicht hängen bleibt.
                          error_code = 1; // Falls kein Ready-Report empfangen wurde oder Kalibrations-
                          return error_code; // konstanten fehlerhaft => Fehler_Code zurückgeben
    } // und Initialisierung abbrechen

    else{ // Ready-Report empfangen. Modul bereit für Initialisierung

        output_rs232(0x88, 0x46); // Frequenz RX+TX : 433.925MHz
        output_rs232(0x8B, 0x02); // Funkdatenrate : 4.8kbit
        output_rs232(0x8C, 0x04); // RS232 Baudrate : 19.2kbaud
        output_rs232(0x8D, 0x25); // Powerup mode : -Continue nach Wakeup
        // -Run nach Powerup
        // -Alle Pins bleiben unverändert

        output_rs232(0x8E, 0x03); // Power control : LED on, HF on
        output_rs232(0x8F, 0x01); // automat. Rekalibration control: -automatisch bei dT>20°C
        output_rs232(0x90, 0x03); // Data Format: : -CRC16 ein
        // : -Handshake ein
        // : -Fastsend aus
        // : -RSSI-TLG nicht anhängen
        // : -RSSI-RUHE nicht anhängen

        if(error_code){ // War Kommunikation immer erfolgreich?
            return error_code; // Nein. => Fehler_Code zurückgeben und Initialisierung abbrechen
        } // **** Initialisierung war bis hierher erfolgreich ****

        output_rs232(0x21, 0x00); // VCO-Kalibration starten
        if(!echo(0x21, 0x01)){ // Antwort von Kalibration abwarten (typ. 50ms, maximal 250ms)
            error_code = 3; // Fehler! Fehler-Code setzen
        }

        return error_code; // Fehler-Code zurückgeben
    } // error_code == 0 bedeutet Initialisierung erfolgreich
} // -----
// Gibt einen Befehl gemäss Befehlsliste aus (3 Bytes: Startzeichen == 0xB0, Funktion, Wert).
// Die Antwort wird geprüft. Falls das Transceivermodul keine oder eine falsche Antwort zurückgibt,
// so wird der Befehl maximal 3x wiederholt.
// Danach wird im Fehlerfall der Fehler-Code "Fehler in RS232 Kommunikation" gesetzt.

// input: funktion = Funktion gemäss Befehlssatz
//        wert = Wert gemäss Befehlssatz

// output: -

void output_rs232(int funktion, int wert)
{
    int i = 3; // maximal 3 Wiederholungen

    while(i){ // Solange noch nicht 3 Versuche gemacht...

        putchar(0xB0); // Startzeichen über RS232 ausgeben (0xB0)
        putchar(funktion); // Funktion über RS232 ausgeben
        putchar(wert); // Wert über RS232 ausgeben

        if(echo((funktion&0x7F), wert)){ // Stimmt Antwort (EEPROM-Bit der Funktion nicht prüfen)?
            i = 0; // Ja. Kommunikation erfolgreich -> While-Schleife beenden
        }
        else{ // Nein, falsche oder keine Antwort
            if(--i){ // Befehl schon dreimal wiederholt?
                error_code = 2; // Ja, Fehler-Code setzen
            }
        }
    }
} // -----

```

## Funktionsbeschreibungen im Detail

### **Funktionsgruppe READ**

**Hinweis:** Die aktuelle Einstellung im RAM kann von der Konfiguration im EEPROM abweichen, wenn eine Konfiguration mit dem WRITE-Befehl nur ins RAM gespeichert worden ist! Ausgelesen werden aber immer die Einstellungen, die im EEPROM gespeichert sind.

---

Funktionsname	<b>READ Softwareversion</b>	<b>Funktion 00H</b>
Funktionsaufruf	0B0H, 00H, 00H	
Antwort	00H, VERS, WEEK, YEAR	

aktuelle Softwareversion im Flash-Speicher des Transceivers. Siehe auch READ Softwaretyp

---

Funktionsname	<b>READ Softwaretyp</b>	<b>Funktion 00H</b>
Funktionsaufruf	0B0H, 00H, 01H	
Antwort	00H, 03H	

Softwaretyp 1 entspricht der Bezeichnung „A“, Softwaretyp 2 der Bezeichnung „B“ und Softwaretyp 3 der Bezeichnung „C“. Diese Bezeichnung ist Bestandteil der Modulbezeichnung. Beim TRX433-10C2 kennzeichnet der Buchstabe „C“ den Softwaretyp.

Typ A = Funkmodem, Typ B = Directmode, Typ C = Bytemode  
Nur Softwaretyp und Softwareversion zusammen identifizieren die Firmware des Transceivers eindeutig. Siehe auch READ Softwareversion und Codierung Typenschild.

---

Funktionsname	<b>READ Befehlssatzversion</b>	<b>Funktion 00H</b>
Funktionsaufruf	0B0H, 00H, 02H	
Antwort	00H, 02H	

Befehlssatz-Version des Transceivers. Diese ist Bestandteil der Modulbezeichnung. Beim TRX433-10C2 kennzeichnet die letzte Ziffer die Befehlssatzversion 2.

Die Funktionen des Transceivers können in Zukunft erweitert oder Kundenanforderungen angepasst werden. Mit dieser Funktion kann bei Bedarf die Rückwärtskompatibilität mit früheren Versionen sichergestellt werden. Siehe auch Codierung Typenschild.

---

---

Funktionsname	<b>READ Hardwareversion</b>	<b>Funktion 00H</b>
Funktionsaufruf	0B0H, 00H, 03H	
Antwort	00H, HVERS	

Hardware-Version des Transceivers. Diese ist Bestandteil der Modulbezeichnung und setzt sich zusammen aus Bestückungsart und Modulart.

Beim TRX433-10C2 kennzeichnen die Ziffern 10 die Hardwareversion 10.

Der Transceiver wird in mehreren Bestückungsvarianten hergestellt.

HVERS = 0AH: 16MHz  $\mu$ P Clockfrequenz, interner Spannungsregler 3.3V, Loopfilter 19.2kbaud  
 HVERS = 0BH: 8MHz  $\mu$ P Clockfrequenz, interner Spannungsregler 3.0V, Loopfilter 19.2kbaud  
 HVERS = 0CH: 16MHz  $\mu$ P Clockfrequenz, ohne internen Spannungsregler, Loopfilter 19.2kbaud  
 HVERS = 0DH: 8MHz  $\mu$ P Clockfrequenz, ohne internen Spannungsregler, Loopfilter 19.2kbaud  
 HVERS = 0EH: 8MHz  $\mu$ P Clockfrequenz, interner Spannungsregler 3.3V, Loopfilter 19.2kbaud

In Zukunft können weitere Hardwareversionen dazukommen. Siehe auch Codierung Typenschild.

---

Funktionsname	<b>READ Frequenzversion</b>	<b>Funktion 00H</b>
Funktionsaufruf	0B0H, 00H, 04H	
Antwort	00H, FVERS	

Frequenz- Version des Transceivers. Diese ist Bestandteil der Modulbezeichnung.  
 Beim TRX433-10C2 kennzeichnet die Ziffernfolge 433 die Frequenzversion.

FVERS = 0: 433 MHz Band, 433.0625 - 434.7875MHz  
 FVERS = 1: 868 MHz Band, 868.0125 - 869.9875MHz  
 FVERS = 2: 915 MHz Band, 914.0125 - 915.9875MHz

Siehe auch Codierung Typenschild.

---

Funktionsname	<b>READ Seriennummer</b>	<b>Funktion 00H</b>
Funktionsaufruf	0B0H, 00H, 05H	
Antwort	00H, SER0, SER1, SER2, SER3	
	SER0 = LSB, SER3 = MSB der 32Bit-Seriennummer	

32bit Unikats Seriennummer des Transceivers.

Siehe auch Codierung Typenschild.

---

---

Funktionsname      **READ RSSI aktuell**      **Funktion 01H**

Funktionsaufruf      0B0H, 01H, 00H

Antwort      01H, RSSI

RSSI im 2er Komplement, Bereich -128 dBm ... -60 dBm, Auflösung 1dBm

Aktueller, momentaner RSSI Wert (**R**eceived **S**ignal **S**trength Indicator = Empfangsfeldstärke) in dBm. Der RSSI-Wert wird alle 3ms ermittelt, unabhängig vom Funktionsaufruf. RSSI stellt also die Feldstärke zu einem Zeitpunkt dar, der max. 3ms zurückliegen kann. Während dem Sendebetrieb ist RSSI = -128dBm.

Der max. messbare Wert liegt bei -60 dBm. Die untere Rauschgrenze hängt vor allem von der eingestellten Funkdatenrate ab.

Siehe auch Abschnitt RSSI und Funktion READ RSSI peak.

---

Funktionsname      **READ RSSI peak**      **Funktion 01H**

Funktionsaufruf      0B0H, 01H, 01H

Antwort      01H, RSSI\_peak

RSSI\_peak im 2er Komplement, Bereich -128 dBm ... -60 dBm, Auflösung 1dBm

Maximalwert (peak hold) des RSSI seit dem letzten Aufruf dieser Funktion.

(**R**eceived **S**ignal **S**trength Indicator = Empfangsfeldstärke) in dBm.

Nach jedem Aufruf dieser Funktion wird RSSI\_peak initialisiert.

Der max. messbare Wert liegt bei -60 dBm. Die untere Rauschgrenze hängt vor allem von der eingestellten Funkdatenrate ab. Während dem Sendebetrieb ist RSSI = -128dBm. Unmittelbar nach einem sleep wird RSSI peak initialisiert.

Siehe auch Abschnitt RSSI und Funktion READ RSSI aktuell.

---

Funktionsname      **READ Temperatur**      **Funktion 01H**

Funktionsaufruf      0B0H, 01H, 02H

Antwort      01H, TEMP

TEMP im 2er Komplement, Bereich -40°C ... +115°C, Auflösung 1°C

Temperatur Transceivermodul in °C.

Die Temperatur wird nur alle 5 Sekunden aktualisiert, egal wie oft diese Funktion aufgerufen wird.

Da während dem Sleep alle Timer angehalten werden, wird auch die Temperatur erst aktualisiert, nachdem der Mikrokontroller des Transceivers 5 Sekunden lang aktiv war. Der HF-Teil des Transceivers muss nicht aktiv sein, um die Temperatur messen zu können.

---

Funktionsname      **READ Empfangsfrequenz**                      **(aus EEPROM)**                      **Funktion 02H**

Funktionsaufruf      0B0H, 02H, 09H

Antwort                      02H, FREQ

FREQ = Kanalnummer gemäss Frequenztafel  
 Bereich 1...139 für 433 MHz Band  
 Bereich 1...159 für 868 MHz Band

Frequenzkanal für den Funkempfang gemäss Frequenztafel.  
 Das Kanalraster beträgt fix 12.5 kHz, unabhängig von der eingestellten Funkdatenrate.

Hinweis: die aktuelle Einstellung (RAM) kann sich von der Einstellung im EEPROM unterscheiden!

Funktionsname      **READ Sendefrequenz**                      **(aus EEPROM)**                      **Funktion 02H**

Funktionsaufruf      0B0H, 02H, 0AH

Antwort                      02H, FREQ

FREQ = Kanalnummer gemäss Frequenztafel  
 Bereich 1...139 für 433 MHz Band  
 Bereich 1...159 für 868 MHz Band

Frequenzkanal für den Sender gemäss Frequenztafel.  
 Das Kanalraster beträgt fix 12.5 kHz, unabhängig von der eingestellten Funkdatenrate.

Hinweis: die aktuelle Einstellung (RAM) kann sich von der Einstellung im EEPROM unterscheiden!

Funktionsname      **READ Funkdatenrate**                      **(aus EEPROM)**                      **Funktion 02H**

Funktionsaufruf      0B0H, 02H, 0BH

Antwort                      02H, RF\_bitrate

RF_bitrate = 00H:	1.2 kbit/s;	Manchester, GFSK, 9.6kHz RF-bandwidth
RF_bitrate = 01H:	2.4 kbit/s;	NRZ, GFSK, 9.6kHz RF-bandwidth
RF_bitrate = 02H:	4.8 kbit/s;	NRZ, GFSK, 19.2kHz RF-bandwidth
RF_bitrate = 03H:	9.6 kbit/s;	NRZ, GFSK, 25.6kHz RF-bandwidth
RF_bitrate = 04H:	19.2kbit/s;	NRZ, GFSK, 51.2kHz RF-bandwidth
RF_bitrate = 05H:	38.4kbit/s;	NRZ, GFSK, 102.4kHz RF-bandwidth

Funkdatenrate für den Sender und Empfänger.  
 Mit der Funkdatenrate sind die Datencodierungsart, die Modulationsart und die RF-bandwidth fest verknüpft.

Funkdatenraten >19.2kbits/s sind möglich, erfordern jedoch eine 16MHz µP Clockfrequenz und ein entsprechend dimensioniertes Loopfilter bzw. eine Bestückungsänderung. Dadurch wird die Empfängerempfindlichkeit bei kleinen Funkdatenraten reduziert. Die Standardversion TRXnnn-10C2 ist mit einem Loopfilter für max. 19.2kbits/s bestückt.

Hinweis: die aktuelle Einstellung (RAM) kann sich von der Einstellung im EEPROM unterscheiden!

Funktionsname	<b>READ RS232 Baudrate</b>	<b>(aus EEPROM)</b>	<b>Funktion 02H</b>
Funktionsaufruf	0B0H, 02H, 0CH		
Antwort	02H, BAUD		
	BAUD = 00H:	1.2 kBaud, Toleranz +- 1%	
	BAUD = 01H:	2.4 kBaud, Toleranz +- 1%	
	BAUD = 02H:	4.8 kBaud, Toleranz +- 1%	
	BAUD = 03H:	9.6 kBaud, Toleranz +- 1%	
	BAUD = 04H:	19.2 kBaud, Toleranz +- 1%	
	BAUD = 05H:	38.4 kBaud, Toleranz +- 1%	
	BAUD = 06H:	57.6 kBaud, Toleranz +- 1%	mit 16 MHz clock
		57.6 kBaud, Toleranz -3.5% +-1%	mit 8 MHz clock
	BAUD = 07H:	115.2 kBaud, Toleranz -3.5% +-1%	

Die höchste Baudrate von 115 kbaud ist nur mit einer 16MHz  $\mu$ P Clockfrequenz möglich.

Die Baudraten haben die angegebene typ. Toleranz vom Idealwert. Die jeweils höchste einstellbare

Baudrate ist 3.5% tiefer als der Nennwert, d.h. 111.1 kBaud bzw. 55.5kBaud (bei 8MHz  $\mu$ P Clockfrequenz).

Um Kommunikationsprobleme zu vermeiden, soll die Summe der Toleranzen beider Kommunikationspartner kleiner als ca. 6% (Richtwert) sein.

### Hinweise Konfigurations-Baudrate

Die eingestellte RS232 Baudrate gilt nur für den normalen Datenverkehr, nicht aber während der Konfiguration (MODE-Pin auf High). Im Konfigurationsmodus ist eine automatische Baudratenerkennung implementiert, welche jede der obigen Baudraten erkennt (für 115.2kBaud ist eine 16MHz  $\mu$ P Clockfrequenz erforderlich). Das heisst, es kann immer mit einer beliebigen Baudrate konfiguriert werden. Automatische Ausgaben wie z.B. das READY nach einem Powerup werden nur ausgegeben, wenn sich das Modul im Konfigurationsmodus befindet (MODE-Pin auf High). Die automatischen Ausgaben werden in der Baudrate ausgegeben, mit der zuletzt konfiguriert wurde. Im normalen Betriebsmodus (MODE-Pin auf Low) werden diese automatischen Ausgaben unterdrückt.

Hinweis: die aktuelle Einstellung (RAM) kann sich von der Einstellung im EEPROM unterscheiden!

---

Funktionsname	<b>READ powerup mode</b>	<b>(aus EEPROM)</b>	<b>Funktion 02H</b>
Funktionsaufruf	0B0H, 02H, 0DH		
Antwort	02H, PWUP_mode	(Bitregister)	

Die einzelnen Bits von PWUP\_mode definieren das Verhalten des Transceivers nach einem Powerup. Der Befehl ist z.B. für Batteriebetrieb oder bei geschalteter Speisung wichtig, damit der Transceiver nach einem Batteriewechsel keinen Batteriestrom verbraucht und nicht zuerst vom externen Controller konfiguriert werden muss.

Die Einstellungen von PWUP\_mode werden nur wirksam, wenn Bit2 = 0 ist, ansonsten startet der Transceiver bei powerup ganz normal auf. Bit2 ist somit der „Hauptschalter“.

Dieser Befehl ist verwandt mit dem sleep- Befehl, im Unterschied dazu wird er jedoch ausschliesslich wirksam bei **Powerup** sowie beim automatischen sleep **unmittelbar nach** einem Powerup.

Bit0 und Bit1 sind ausschliesslich nur wirksam für einen wakeup aus einem automatischen sleep nach powerup (also nicht bei einem normalen wakeup nach einem sleep- Befehl). Bei einem continue nach Wakeup ist der Transceiver sehr schnell (bei 19.2kbit/s Funkdatenrate nach 3ms, bei 2.4kbit/s Funkdatenrate nach 5ms) nach dem wakeup wieder bereit und macht nicht zuerst noch eine Kalibration und Konfiguration wie beim Soft-Reset, welche typ. 75ms dauert.

PWUP\_mode, bit0 = 0: Wakeup erzeugt einen internen Soft-Reset, d.h. Programm wird neu gestartet  
 PWUP\_mode, bit0 = 1: Continue nach Wakeup, d.h. Programm macht dort weiter, wo es vor Sleep war  
 PWUP\_mode, bit1: unbenützt, reserviert für Wakeup

PWUP\_mode, bit2 = 0: automatischer sleep nach Powerup. Transceiver muss danach geweckt werden.  
 PWUP\_mode, bit2 = 1: Run nach Powerup, d.h. Transceiver startet nach Powerup mit Konfig. aus EEPROM  
 PWUP\_mode, bit3: unbenützt, reserviert für Powerup

PWUP\_mode, bit4: unbenützt, reserviert für Sleep-State  
 PWUP\_mode, bit5 = 0: alle Pins ausser WKUP-Pin während sleep auf Ausgang mit Low-Pegel setzen  
 PWUP\_mode, bit5 = 1: alle Pins bleiben unverändert

Mit dem WKUP-Pin kann der automatische sleep nach Powerup beendet werden. Dazu muss am WKUP-Pin ein Pegelwechsel auftreten. Nach dem wakeup meldet sich der Transceiver mit der READY Meldung 30H, 01H „bereit nach wakeup“ und die Pins CTS und RxD werden wieder zu Eingängen, falls diese während dem sleep auf Ausgang gesetzt wurden (die READY Meldung wird nur ausgegeben, wenn sich der Transceiver im Konfigurationsmodus befindet). Alternativ dazu kann auch der RTS-Pin überwacht werden. Nach einem Powerup mit automatischem Sleep geht der RTS-Pin nach ca. 110ms auf High (Sleep wird eingeleitet). Wird der Transceiver danach wieder geweckt (mit einem Flankenwechsel am WKUP-Pin), wird der RTS-Pin wieder auf Low gesetzt. Als Bereitzeichen kann also auch eine negative Flanke am RTS-Pin verwendet werden.

**Hinweis:** nur die Einstellung im EEPROM ist von Interesse, der Wert im RAM geht bei einem powerdown-powerup Zyklus verloren!

Siehe auch Abschnitt Stromsparmodi und WRITE sleep.

---



---

Funktionsname	<b>READ power control</b>	<b>(aus EEPROM)</b>	<b>Funktion 02H</b>
Funktionsaufruf	0B0H, 02H, 0EH		
Antwort	02H, PWR_ctrl	(Bitregister)	
PWR_ctrl, bit0 = 0:	LED Transceiver = OFF		
PWR_ctrl, bit0 = 1	LED Transceiver = ON		
PWR_ctrl, bit1 = 0:	HF-Teil Transceiver = OFF		
PWR_ctrl, bit1 = 1:	HF-Teil Transceiver = ON		

Die einzelnen Bits von PWR\_ctrl schalten Blöcke des Transceivers ein und aus, um Strom zu sparen. Die Kalibrationsdaten bleiben erhalten, wenn der Transceiver mit diesem Befehl aus- und dann wieder eingeschaltet wird.

Dieser Befehl kann dazu verwendet werden, um z.B. die Temperatur auszulesen oder die LED zu schalten, wozu der Funk nicht benötigt wird. Bei abgeschaltetem Funkteil beträgt der Stromverbrauch nur einen Bruchteil des normalen Bedarfs.

Hinweis: die aktuelle Einstellung (RAM) kann sich von der Einstellung im EEPROM unterscheiden!

---

Funktionsname	<b>READ automat. Rekalibration control</b>	<b>(aus EEPROM)</b>	<b>Funktion 02H</b>
Funktionsaufruf	0B0H, 02H, 0FH		
Antwort	02H, ACAL		
	ACAL = 0:	keine automatische VCO-Kalibration	
	ACAL = 1:	automatisch bei $\Delta T = 20^\circ\text{C}$ neu kalibrieren	
	ACAL = mm:	automatisch alle mm Minuten neu kalibrieren	
		mm = 5d ... 150d (Intervall 5...150 Minuten)	

Zeit- oder temperaturgesteuerte automatische VCO-Kalibration.

Der VCO muss neu kalibriert werden, wenn:

- sich die Speisespannung um mehr als 0.25V ändert. Dies ist bei der Version ohne internen Spannungsregler bei Batteriebetrieb zu beachten.
- Die Temperatur sich um mehr als  $40^\circ\text{C}$  verändert.

Während der Dauer der Kalibration ist kein Funkkontakt möglich und der Stromverbrauch entspricht dem Empfangsmodus. Normalerweise ist die Kalibration nach typ. 50ms, d.h. auf Anhieb erfolgreich beendet. Wenn der erste Versuch nicht erfolgreich war, wiederholt der Transceiver selber bis zu fünfmal die Kalibration. Somit kann die Kalibration im Maximum bis zu 250ms dauern. Beginn, Ende und Status der automatischen Rekalibration werden nicht rückgemeldet, auch nicht, wenn sich das Modul im Konfigurationsmodus befindet (im Gegensatz zur Funktion „jetzt kalibrieren“ (21H, 00H), bei der der Status der Kalibration rückgemeldet wird, falls sich der Transceiver im Konfigurationsmodus befindet).

Da während dem Sleep alle Timer angehalten werden, führen zeitgesteuerte Ausgaben nur zum erwarteten Resultat, wenn der sleep- Befehl nicht verwendet wird. Dasselbe gilt für die temperaturgesteuerte Kalibration, da nur alle 5 Sekunden eine neue Temperatur ermittelt wird.

Hinweis: die aktuelle Einstellung (RAM) kann sich von der Einstellung im EEPROM unterscheiden!

---

Funktionsname	<b>READ Data Format</b>	<b>(aus EEPROM)</b>	<b>Funktion 02H</b>
Funktionsaufruf	0B0H, 02H, 10H		
Antwort	02H, FORMAT	(Bitregister)	

Die einzelnen Bits von FORMAT definieren die verschiedenen Einstellmöglichkeiten im Zusammenhang mit der seriellen Datenübertragung über RS232 und der Funkdatenübertragung. Über die genauen Auswirkungen auf das Timingverhalten der einzelnen Einstellungen gibt der Abschnitt Funkübertragung Auskunft.

FORMAT, bit0 = 0:	CRC16 aus. Keine Checksummenprüfung der Funkdaten
FORMAT, bit0 = 1:	CRC16 ein. Checksummenprüfung der Funkdaten mit CRC16
FORMAT, bit1 = 0:	Handshake aus. Keine Flusssteuerung der RS232-Schnittstelle
FORMAT, bit1 = 1:	Handshake ein. Flusssteuerung der RS232-Schnittstelle mit RTS/CTS
FORMAT, bit2 = 0:	Fastsend aus. Start der Funkübertragung nach 2-3-Byte-Pause über RS232
FORMAT, bit2 = 1:	Fastsend ein. Start der Funkübertragung sofort nach erstem Byte über RS232
FORMAT, bit3 = 0:	Nach Nutzdaten wird das RSSI des Funktelegramms nicht ausgegeben
FORMAT, bit3 = 1:	Nach Nutzdaten wird auch noch das RSSI des Funktelegramms ausgegeben
FORMAT, bit4 = 0:	Nach dem RSSI des Funktelegramms wird das Ruhe - RSSI nicht ausgegeben
FORMAT, bit4 = 1:	Nach dem RSSI des Funktelegramms wird auch noch das Ruhe - RSSI ausgegeben

Bei der CRC16 – Checksumme handelt es sich um eine 16-Bit Checksumme, mit deren Hilfe die Funkübertragungstrecke gesichert wird. Stimmt beim Empfänger die berechnete Checksumme nicht mit der über Funk übertragenen Checksumme überein, werden die Daten verworfen.

Bei eingeschaltetem Handshake dürfen keine Daten mehr über RS232 an das Transceivermodul gesendet werden, sobald der RTS-Pin vom Transceivermodul auf High gesetzt wird. Der Anwender kann die Ausgabe von Daten durch das Transceivermodul stoppen, indem er den CTS-Pin auf high setzt. Ist der Handshake ausgeschaltet, so ist der Anwender selbst dafür verantwortlich, dass der interne Sende- und Empfangsbuffer des Transceivermoduls nicht überfüllt wird. RTS- und CTS-Leitung werden in diesem Fall nicht beachtet.

Da eingeschalteter Fastsend für den Transceiver zusätzlichen Rechenaufwand bedeutet, kann Fastsend nur bis zu einer Funkdatenrate von 19.2kbit/s (bzw. 9.6kbit/s bei 8MHz  $\mu$ P Clockfrequenz) verwendet werden.

Das Ruhe - RSSI kann nur ausgegeben werden, wenn auch das RSSI des Funktelegramms ausgegeben wird. Das RSSI des Telegramms und das Ruhe - RSSI werden im Anschluss an die eigentlichen über Funk übertragenen Nutzdaten im 2er-Komplement in dBm ausgegeben.

#### Hinweise:

- Die aktuelle Einstellung (RAM) kann sich von der Einstellung im EEPROM unterscheiden!
  - Wird der CTS-Pin als Handshake verwendet, so dürfen keine weiteren Daten an das Transceivermodul gesendet werden solange der CTS-Pin auf High gesetzt ist, da diese Daten sonst vom Transceivermodul als Konfigurationsdaten interpretiert würden.
-

Funktionsname **READ EEPROM** **Funktion 03H**

Funktionsaufruf 0B0H, 03H, EE\_ADR

Antwort 03H, EE\_REG

EE\_REG: EEPROM Registerinhalt an der Adresse EE\_ADR  
 EE\_ADR = 0 ... FFH

Ein beliebiges EEPROM Registerbyte auslesen.

Der vom Transceiver unbenutzte Teil des EEPROM's (Adresse 0D0H...0FEH) kann von der Applikation frei verwendet werden. Die Kalibrationswerte des Transceivers, div. Konstanten sowie die Konfiguration sind ebenfalls in diesem EEPROM abgelegt und sind der Applikation zugänglich.

An der EEPROM- Adresse FFH befindet sich der Schreibschutz EE\_WRPROT für das EEPROM. Jedes der 8 Bits von EE\_WRPROT steuert einen EEPROM Bereich wie folgt:

Bitn = 0: Adressbereich ist beschreibbar und lesbar (Werkseinstellung)  
 Bitn = 1: Adressbereich ist zum Schreiben gesperrt und nur noch lesbar

EE_WRPROT, Bit0	Adresse 00H..1FH	Kalibrationswerte Frequenz und RSSI
EE_WRPROT, Bit1	Adresse 20H..6FH	Kalibrationswerte Temperaturkompensation
EE_WRPROT, Bit2	Adresse 70H..8FH	Kalibrationswerte Reserve
EE_WRPROT, Bit3	Adresse 90H..9FH	Hardwarekonstanten
EE_WRPROT, Bit4	Adresse A0H..AFH	Softwarekonstanten
EE_WRPROT, Bit5	Adresse B0H..CFH	aktuelle Transceiverkonfiguration
EE_WRPROT, Bit6	Adresse D0H..FEH	<b>frei verwendbar für Applikation</b>
EE_WRPROT, Bit7	Adresse FFH..FFH	EE_WRPROT (dieses Byte)

Der Schreibschutz EE\_WRPROT gilt nur für den Zugriff mit den Befehlen WRITE EEPROM und nicht für die anderen WRITE Befehle, welche auch ins EEPROM schreiben.

**Hinweise:**

- Obwohl das ganze EEPROM zum Beschreiben freigegeben ist, darf nur der für die Applikation zugeteilte Block (Adresse 0D0H...0FEH) beschrieben werden.
- Wenn das EEPROM von der Applikation als Speicher benutzt wird, sollte EE\_WRPROT auf b'10111111' = BFH gesetzt werden, um das EEPROM vor ungewollter Veränderung zu schützen.
- Nachdem Bit7 von EE\_WRPROT im EEPROM auf 1 gesetzt worden ist, kann der Schreibschutz als Ganzes nicht mehr verändert werden. Dieser Befehl muss also gut überlegt angewendet werden!
- Werden andere EEPROM –Daten wie z.B. Kalibrationswerte verändert, wird der Transceiver unter Umständen unbrauchbar und muss vom Hersteller neu kalibriert werden.

Siehe auch EEPROM Funktionen 2DH, 2EH, 2FH zum Beschreiben eines beliebigen EEPROM Registers.

**Funktionsgruppe WRITE**

Wenn die Parameter ins RAM und ins EEPROM gespeichert werden, muss darauf geachtet werden, dass die begrenzte Anzahl von 100'000 EEPROM-Schreibzyklen pro Parameter nicht überschritten wird. Wird allerdings versucht einen Wert ins EEPROM zu schreiben, der schon im EEPROM steht, wird der Schreibbefehl nicht ausgeführt. Somit kann also eine sich nicht ändernde Konfiguration beliebig oft ins EEPROM geschrieben werden. Der effektive Schreibbefehl wird dabei nur bei der ersten Konfiguration ausgeführt.

---

Funktionsname	<b>WRITE Frequenz RX+TX</b>	<b>Funktion 08H</b>
---------------	-----------------------------	---------------------

Funktionsaufruf	0B0H, 08H, FREQ
-----------------	-----------------

Antwort	08H, FREQ
---------	-----------

FREQ = Kanalnummer gemäss Frequenztafel  
 Bereich 1...139 für 433 MHz Band  
 Bereich 1...159 für 868 MHz Band

Frequenzkanal für den Funkempfang und Senden gemäss Frequenztafel. Sende- und Empfangskanal werden beide gemeinsam auf die gleiche Frequenz eingestellt.

Das Kanalraster beträgt fix 12.5 kHz, unabhängig von der eingestellten Funkdatenrate.

Wenn Bit7 von Byte2 high ist, wird der Parameter auch im EEPROM abgelegt, die Einstellung wird dann nach einem Powerup sofort aktiviert. Die Antwort erfolgt immer mit Bit7 = 0.

Ein Frequenzwechsel benötigt bei 19.2kbit/s Funkdatenrate 2ms, bei 2.4kbit/s Funkdatenrate 5ms, d.h. nach Ablauf dieser Zeit ist die neue Frequenz eingestellt und kann verwendet werden.

---

Funktionsname	<b>WRITE Frequenz RX</b>	<b>Funktion 09H</b>
---------------	--------------------------	---------------------

Funktionsaufruf	0B0H, 09H, FREQ
-----------------	-----------------

Antwort	09H, FREQ
---------	-----------

FREQ = Kanalnummer gemäss Frequenztafel  
 Bereich 1...139 für 433 MHz Band  
 Bereich 1...159 für 868 MHz Band

Frequenzkanal für den Funkempfang gemäss Frequenztafel. Die Sendefrequenz kann sich von der Empfangsfrequenz unterscheiden.

Das Kanalraster beträgt fix 12.5 kHz, unabhängig von der eingestellten Funkdatenrate.

Wenn Bit7 von Byte2 high ist, wird der Parameter auch im EEPROM abgelegt, die Einstellung wird dann nach einem Powerup sofort aktiviert. Die Antwort erfolgt immer mit Bit7 = 0.

Ein Frequenzwechsel benötigt bei 19.2kbit/s Funkdatenrate 2ms, bei 2.4kbit/s Funkdatenrate 5ms, d.h. nach Ablauf dieser Zeit ist die neue Frequenz eingestellt und kann verwendet werden.

---

---

Funktionsname	<b>WRITE Frequenz TX</b>	<b>Funktion 0AH</b>
Funktionsaufruf	0B0H, 0AH, FREQ	
Antwort	0AH, FREQ	

FREQ = Kanalnummer gemäss Frequenztafel  
 Bereich 1...139 für 433 MHz Band  
 Bereich 1...159 für 868 MHz Band

Frequenzkanal für den Sendemodus gemäss Frequenztafel. Die Empfangsfrequenz kann sich von der Sendefrequenz unterscheiden.

Das Kanalraster beträgt fix 12.5 kHz, unabhängig von der eingestellten Funkdatenrate.

Wenn Bit7 von Byte2 high ist, wird der Parameter auch im EEPROM abgelegt, die Einstellung wird dann nach einem Powerup sofort aktiviert. Die Antwort erfolgt immer mit Bit7 = 0.

Ein Frequenzwechsel benötigt bei 19.2kbit/s Funkdatenrate 2ms, bei 2.4kbit/s Funkdatenrate 5ms, d.h. nach Ablauf dieser Zeit ist die neue Frequenz eingestellt und kann verwendet werden.

---

Funktionsname	<b>WRITE Funkdatenrate</b>	<b>Funktion 0BH</b>
Funktionsaufruf	0B0H, 0BH, RF_bitrate	
Antwort	0BH, RF_bitrate	

RF_bitrate = 00H:	1.2 kbit/s;	Manchester, GFSK, 9.6kHz RF-bandwidth
RF_bitrate = 01H:	2.4 kbit/s;	NRZ, GFSK, 9.6kHz RF-bandwidth
RF_bitrate = 02H:	4.8 kbit/s;	NRZ, GFSK, 19.2kHz RF-bandwidth
RF_bitrate = 03H:	9.6 kbit/s;	NRZ, GFSK, 25.6kHz RF-bandwidth
RF_bitrate = 04H:	19.2kbit/s;	NRZ, GFSK, 51.2kHz RF-bandwidth
RF_bitrate = 05H:	38.4kbit/s;	NRZ, GFSK, 102.4kHz RF-bandwidth

Funkdatenrate für den Sender und Empfänger.

Mit der Funkdatenrate sind die Datencodierungsart, die Modulationsart und die RF-bandwidth fest verknüpft.

Funkdatenraten >19.2kbits/s sind möglich, erfordern jedoch eine 16MHz  $\mu$ P Clockfrequenz und ein entsprechend dimensioniertes Loopfilter bzw. eine Bestückungsänderung. Dadurch wird die Empfängerempfindlichkeit bei kleinen Funkdatenraten reduziert. Die Standardversion TRXnnn-10C2 ist mit einem Loopfilter für max. 19.2kbits/s bestückt.

Wenn Bit7 von Byte2 high ist, wird der Parameter auch im EEPROM abgelegt, die Einstellung wird dann nach einem Powerup sofort aktiviert. Die Antwort erfolgt immer mit Bit7 = 0.

---

Funktionsname	<b>WRITE RS232 Baudrate</b>	<b>Funktion 0CH</b>
Funktionsaufruf	0B0H, 0CH, BAUD	
Antwort	0CH, BAUD	
	BAUD = 00H: 1.2 kBaud, Toleranz +- 1%	
	BAUD = 01H: 2.4 kBaud, Toleranz +- 1%	
	BAUD = 02H: 4.8 kBaud, Toleranz +- 1%	
	BAUD = 03H: 9.6 kBaud, Toleranz +- 1%	
	BAUD = 04H: 19.2 kBaud, Toleranz +- 1%	
	BAUD = 05H: 38.4 kBaud, Toleranz +- 1%	
	BAUD = 06H: 57.6 kBaud, Toleranz +- 1%	mit 16 MHz clock
	57.6 kBaud, Toleranz -3.5% +-1%	mit 8 MHz clock
	BAUD = 07H: 115.2 kBaud, Toleranz -3.5% +-1%	

Die höchste Baudrate von 115 kbaud ist nur mit einer 16MHz  $\mu$ P Clockfrequenz möglich.

Die Baudraten haben die angegebene typ. Toleranz vom Idealwert. Die jeweils höchste einstellbare

Baudrate ist 3.5% tiefer als der Nennwert, d.h. 111.1 kBaud bzw. 55.5kBaud (bei 8MHz  $\mu$ P Clockfrequenz).

Um Kommunikationsprobleme zu vermeiden, soll die Summe der Toleranzen beider Kommunikationspartner kleiner als ca. 6% (Richtwert) sein.

#### **Hinweise Konfigurations-Baudrate**

Die eingestellte RS232 Baudrate gilt nur für den normalen Datenverkehr, nicht aber während der Konfiguration (MODE-Pin auf High). Im Konfigurationsmodus ist eine automatische Baudratenerkennung implementiert, welche jede der obigen Baudraten erkennt (für 115.2kBaud ist eine 16MHz  $\mu$ P Clockfrequenz erforderlich). Das heisst, es kann immer mit einer beliebigen Baudrate konfiguriert werden. Automatische Ausgaben wie z.B. das READY nach einem Powerup werden nur ausgegeben, wenn sich das Modul im Konfigurationsmodus befindet (MODE-Pin auf High). Die automatischen Ausgaben werden in der Baudrate ausgegeben, mit der zuletzt konfiguriert wurde. Im normalen Betriebsmodus (MODE-Pin auf Low) werden diese automatischen Ausgaben unterdrückt.

Wenn Bit7 von Byte2 high ist, wird der Parameter auch im EEPROM abgelegt, die Einstellung wird dann nach einem Powerup sofort aktiviert. Die Antwort erfolgt immer mit Bit7 = 0.

Funktionsname	<b>WRITE powerup mode</b>	<b>Funktion 0DH</b>
Funktionsaufruf	0B0H, 0DH, PWUP_mode	
Antwort	0DH, PWUP_mode (Bitregister)	

Die einzelnen Bits von PWUP\_mode definieren das Verhalten des Transceivers nach einem Powerup. Der Befehl ist z.B. für Batteriebetrieb oder bei geschalteter Speisung wichtig, damit der Transceiver nach einem Batteriewechsel keinen Batteriestrom verbraucht und nicht zuerst vom externen Controller konfiguriert werden muss.

Die Einstellungen von PWUP\_mode werden nur wirksam, wenn Bit2 = 0 ist, ansonsten startet der Transceiver bei powerup ganz normal auf. Bit2 ist somit der „Hauptschalter“.

Dieser Befehl ist verwandt mit dem sleep- Befehl, im Unterschied dazu wird er jedoch ausschliesslich wirksam bei **Powerup** sowie beim automatischen sleep **unmittelbar nach** einem Powerup.

Bit0 und Bit1 sind ausschliesslich nur wirksam für einen wakeup aus einem automatischen sleep nach powerup (also nicht bei einem normalen wakeup nach einem sleep-Befehl). Bei einem continue nach Wakeup ist der Transceiver sehr schnell (bei 19.2kbit/s Funkdatenrate nach 3ms, bei 2.4kbit/s Funkdatenrate nach 5ms) nach dem wakeup wieder bereit und macht nicht zuerst noch eine Kalibration und Konfiguration wie beim Soft-Reset, welche typ. 75ms dauert.

PWUP\_mode, bit0 = 0: Wakeup erzeugt einen internen Soft-Reset, d.h. Programm wird neu gestartet  
 PWUP\_mode, bit0 = 1: Continue nach Wakeup, d.h. Programm macht dort weiter, wo es vor Sleep war  
 PWUP\_mode, bit1: unbenützt, reserviert für Wakeup

PWUP\_mode, bit2 = 0: automatischer sleep nach Powerup. Transceiver muss danach geweckt werden.  
 PWUP\_mode, bit2 = 1: Run nach Powerup, d.h. Transceiver startet nach Powerup mit Konfig. aus EEPROM  
 PWUP\_mode, bit3: unbenützt, reserviert für Powerup

PWUP\_mode, bit4: unbenützt, reserviert für Sleep-State  
 PWUP\_mode, bit5 = 0: alle Pins ausser WKUP-Pin während sleep auf Ausgang mit Low-Pegel setzen  
 PWUP\_mode, bit5 = 1: alle Pins bleiben unverändert

Mit dem WKUP-Pin kann der automatische sleep nach Powerup beendet werden. Dazu muss am WKUP-Pin ein Pegelwechsel auftreten. Nach dem wakeup meldet sich der Transceiver mit der READY Meldung 30H, 01H „bereit nach wakeup“ und die Pins CTS und RxD werden wieder zu Eingängen, falls diese während dem sleep auf Ausgang gesetzt wurden (die READY Meldung wird nur ausgegeben, wenn sich der Transceiver im Konfigurationsmodus befindet). Alternativ dazu kann auch der RTS-Pin überwacht werden. Nach einem Powerup mit automatischem Sleep geht der RTS-Pin nach ca. 110ms auf High (Sleep wird eingeleitet). Wird der Transceiver danach wieder geweckt (mit einem Flankenwechsel am WKUP-Pin), wird der RTS-Pin wieder auf Low gesetzt. Als Bereitzeichen kann also auch eine negative Flanke am RTS-Pin verwendet werden.

#### Hinweis:

nur die Einstellung im EEPROM ist von Interesse, der Wert im RAM geht bei einem powerdown- powerup Zyklus verloren! Deshalb muss Bit7 von Byte2 high sein, der Parameter also im EEPROM abgelegt werden, damit die Einstellung nach einem Powerup aktiviert wird. Die Antwort erfolgt immer mit Bit7 = 0.

Siehe auch Abschnitt Stromsparmodi und WRITE sleep.



---

Funktionsname	<b>WRITE power control</b>	<b>Funktion 0EH</b>
Funktionsaufruf	0B0H, 0EH, PWR_ctrl	
Antwort	0EH, PWR_ctrl	(Bitregister)
PWR_ctrl, bit0 = 0:	LED Transceiver = OFF	
PWR_ctrl, bit0 = 1	LED Transceiver = ON	
PWR_ctrl, bit1 = 0:	HF-Teil Transceiver = OFF	
PWR_ctrl, bit1 = 1:	HF-Teil Transceiver = ON	

Die einzelnen Bits von PWR\_ctrl schalten Blöcke des Transceivers ein und aus, um Strom zu sparen. Die Kalibrationsdaten bleiben erhalten, wenn der Transceiver mit diesem Befehl aus- und dann wieder eingeschaltet wird.

Der HF-Teil ist bei 19.2kbit/s Funkdatenrate nach 3ms, bei 2.4kbit/s Funkdatenrate nach 5ms nach Einschalten durch diesen Befehl wieder Betriebsbereit.

Dieser Befehl kann dazu verwendet werden, um z.B. die Temperatur auszulesen oder die LED zu schalten, wozu der Funk nicht benötigt wird. Bei abgeschaltetem Funkteil beträgt der Stromverbrauch nur einen Bruchteil des normalen Bedarfs.

Wenn Bit7 von Byte2 high ist, wird der Parameter auch im EEPROM abgelegt, die Einstellung wird dann nach einem Powerup sofort aktiviert. Die Antwort erfolgt immer mit Bit7 = 0.

---

Funktionsname	<b>WRITE automat. Rekalibration control</b>	<b>Funktion 0FH</b>
Funktionsaufruf	0B0H, 0FH, ACAL	
Antwort	0FH, ACAL	
	ACAL = 0:	keine automatische VCO-Kalibration
	ACAL = 1:	automatisch bei $\Delta T = 20^{\circ}\text{C}$ neu kalibrieren
	ACAL = mm:	automatisch alle mm Minuten neu kalibrieren mm = 5d ... 150d (Intervall 5...150 Minuten)

Definiert die zeit- oder temperaturgesteuerte automatische VCO-Kalibration.

Der VCO muss neu kalibriert werden, wenn:

- sich die Speisespannung um mehr als 0.25V ändert. Dies ist bei der Version ohne internen Spannungsregler bei Batteriebetrieb zu beachten.
- Die Temperatur sich um mehr als  $40^{\circ}\text{C}$  verändert.

Während der Dauer der Kalibration ist kein Funkkontakt möglich und der Stromverbrauch entspricht dem Empfangsmodus. Normalerweise ist die Kalibration nach typ. 50ms, d.h. auf Anhieb erfolgreich beendet. Wenn der erste Versuch nicht erfolgreich war, wiederholt der Transceiver selber bis zu fünfmal die Kalibration. Somit kann die Kalibration im Maximum bis zu 250ms dauern. Beginn, Ende und Status der automatischen Rekalibration werden nicht rückgemeldet, auch nicht, wenn sich das Modul im Konfigurationsmodus befindet (im Gegensatz zur Funktion „jetzt kalibrieren“ (21H, 00H), bei der der Status der Kalibration rückgemeldet wird, falls sich der Transceiver im Konfigurationsmodus befindet).

Da während dem Sleep alle Timer angehalten werden, führen zeitgesteuerte Ausgaben nur zum erwarteten Resultat, wenn der sleep- Befehl nicht verwendet wird. Dasselbe gilt für die temperaturgesteuerte Kalibration, da nur alle 5 Sekunden eine neue Temperatur ermittelt wird.

Wenn Bit7 von Byte2 high ist, wird der Parameter auch im EEPROM abgelegt, die Einstellung wird dann nach einem Powerup sofort aktiviert. Die Antwort erfolgt immer mit Bit7 = 0.

---



Funktionsname	<b>WRITE Data Format</b>	<b>Funktion 10H</b>
Funktionsaufruf	0B0H, 10H, FORMAT	
Antwort	10H, FORMAT	(Bitregister)

Die einzelnen Bits von FORMAT definieren die verschiedenen Einstellmöglichkeiten im Zusammenhang mit der seriellen Datenübertragung über RS232 und der Funkdatenübertragung. Über die genauen Auswirkungen auf das Timingverhalten der einzelnen Einstellungen gibt der Abschnitt Funkübertragung Auskunft.

FORMAT, bit0 = 0:	CRC16 aus. Keine Checksummenprüfung der Funkdaten
FORMAT, bit0 = 1:	CRC16 ein. Checksummenprüfung der Funkdaten mit CRC16
FORMAT, bit1 = 0:	Handshake aus. Keine Flusssteuerung der RS232-Schnittstelle
FORMAT, bit1 = 1:	Handshake ein. Flusssteuerung der RS232-Schnittstelle mit RTS/CTS
FORMAT, bit2 = 0:	Fastsend aus. Start der Funkübertragung nach 2-3-Byte-Pause über RS232
FORMAT, bit2 = 1:	Fastsend ein. Start der Funkübertragung sofort nach erstem Byte über RS232
FORMAT, bit3 = 0:	Nach Nutzdaten wird das RSSI des Funktelegramms nicht ausgegeben
FORMAT, bit3 = 1:	Nach Nutzdaten wird auch noch das RSSI des Funktelegramms ausgegeben
FORMAT, bit4 = 0:	Nach dem RSSI des Funktelegramms wird das Ruhe - RSSI nicht ausgegeben
FORMAT, bit4 = 1:	Nach dem RSSI des Funktelegramms wird auch noch das Ruhe - RSSI ausgegeben

Bei der CRC16 – Checksumme handelt es sich um eine 16-Bit Checksumme, mit deren Hilfe die Funkübertragungstrecke gesichert wird. Stimmt beim Empfänger die berechnete Checksumme nicht mit der über Funk übertragenen Checksumme überein, werden die Daten verworfen.

Bei eingeschaltetem Handshake dürfen keine Daten mehr über RS232 an das Transceivermodul gesendet werden, sobald der RTS-Pin vom Transceivermodul auf High gesetzt wird. Der Anwender kann die Ausgabe von Daten durch das Transceivermodul stoppen, indem er den CTS-Pin auf high setzt. Ist der Handshake ausgeschaltet, so ist der Anwender selbst dafür verantwortlich, dass der interne Sende- und Empfangsbuffer des Transceivermoduls nicht überfüllt wird. RTS- und CTS-Leitung werden in diesem Fall nicht beachtet.

Da eingeschalteter Fastsend für den Transceiver zusätzlichen Rechenaufwand bedeutet, kann Fastsend nur bis zu einer Funkdatenrate von 19.2kbit/s (bzw. 9.6kbit/s bei 8MHz  $\mu$ P Clockfrequenz) verwendet werden.

Das Ruhe - RSSI kann nur ausgegeben werden, wenn auch das RSSI des Funktelegramms ausgegeben wird. Das RSSI des Telegramms und das Ruhe - RSSI werden im Anschluss an die eigentlichen über Funk übertragenen Nutzdaten im 2er-Komplement in dBm ausgegeben.

Wenn Bit7 von Byte2 high ist, wird der Parameter auch im EEPROM abgelegt, die Einstellung wird dann nach einem Powerup sofort aktiviert. Die Antwort erfolgt immer mit Bit7 = 0.

**Hinweis:** Wird der CTS-Pin als Handshake verwendet, so dürfen keine weiteren Daten an das Transceivermodul gesendet werden solange der CTS-Pin auf High gesetzt ist, da diese Daten sonst vom Transceivermodul als Konfigurationsdaten interpretiert würden.

---

Funktionsname	<b>WRITE sleep</b>	<b>Funktion 20H</b>
Funktionsaufruf	0B0H, 20H, WKUP_mode	
Antwort	20H, WKUP_mode	(Bitregister)

Versetzt den ganzen Transceiver in den Sleep-mode um Strom zu sparen. Transceiver und LED werden abgeschaltet und der clock des Mikrokontrollers wird gestoppt.

Vor dem sleep Befehl, spätestens jedoch 50µs nach der Antwort auf den sleep Befehl muss der WKUP-Pin auf seinem sleep-Pegel sein (high oder low). Danach löst ein Pegelwechsel sofort einen wakeup aus.

Die einzelnen Bits von WKUP\_mode definieren das Verhalten des Transceivers während dem Sleep sowie nach dem Wakeup. Diese Konfigurationen sind für Batteriebetrieb wichtig.

WKUP\_mode, bit0 = 0: Wakeup erzeugt einen Soft-Reset, d.h. Programm wird neu gestartet

WKUP\_mode, bit0 = 1: Continue nach Wakeup, d.h. Programm macht dort weiter, wo es vor Sleep war.

WKUP\_mode, bit1: unbenützt, reserviert für Wakeup

WKUP\_mode, bit2: unbenützt

WKUP\_mode, bit3: unbenützt

WKUP\_mode, bit4: unbenützt, reserviert für Sleep-State

WKUP\_mode, bit5 = 0: alle Pins ausser WKUP-Pin während sleep auf Ausgang mit Low-Pegel setzen.

WKUP\_mode, bit5 = 1: alle Pins bleiben unverändert

Mit dem WKUP-Pin kann der sleep beendet werden. Dazu muss am WKUP-Pin ein Pegelwechsel auftreten. Nach dem wakeup meldet sich der Transceiver mit der READY Meldung 30H, 01H (die READY Meldung wird nur ausgegeben, wenn sich der Transceiver im Konfigurationsmodus befindet).

Die Pins CTS und RxD werden wieder zu Eingängen, falls diese während dem sleep auf Ausgang gesetzt wurden.

Alternativ zur READY Meldung 30H, 01H kann auch der RTS-Pin überwacht werden. Nach dem Sleep-Befehl geht der RTS-Pin auf High. Sobald der Transceiver nach einem wakeup wieder bereit ist, wird der RTS-Pin wieder auf Low gesetzt. Als Bereitzeichen kann also auch eine negative Flanke am RTS-Pin verwendet werden.

Nach einem wakeup sind immer noch die letzten Kalibrierdaten vor dem sleep aktiv. Falls diese nach dem wakeup nicht mehr gültig sind (z.B. wegen grosser Temperaturänderung), muss nach dem wakeup als erstes neu kalibriert werden. Beim Soft-Reset wird automatisch neu kalibriert und danach die Konfiguration aus dem EEPROM eingestellt. Deshalb ist nach einem Soft-Reset der Transceiver erst nach typ. 75ms bereit, im Gegensatz zum Continue nach Wakeup, das keine automatische Kalibration durchführt und somit bereits nach 3ms (bei 19.2kbit/s Funkdatenrate) bzw. 5ms (bei 2.4kbit/s Funkdatenrate) wieder bereit ist.

Während dem sleep werden alle Timer für automatische Ausgaben usw. angehalten und laufen erst nach einem wakeup weiter. Zeitgesteuerte Ausgaben führen daher nur zum erwarteten Resultat, wenn der sleep-Befehl nicht verwendet wird.

Siehe auch Funktion WRITE powerup mode sowie den Abschnitt Stromsparmodi.

---

---

Funktionsname	<b>WRITE jetzt recalibrieren</b>	<b>Funktion 21H</b>
Funktionsaufruf	0B0H, 21H, 00H	
Antwort	21H, 00H	sofort, wenn Kalibration startet
	21H, 01H	nach typ. 50ms, wenn Kalibration erfolgreich beendet
bzw.	21H, 00H	sofort, wenn Kalibration startet
	21H, 02H	nach typ. 250ms, falls Kalibration nicht erfolgreich war.

Startet unmittelbar eine VCO- Kalibration. Falls sich das Transceivermodul im Konfigurationsmodus befindet, werden Beginn und Ende der Kalibration rückgemeldet, ebenso das Ergebnis der Kalibration. Normalerweise ist die Kalibration nach typ. 50ms, d.h. auf Anhieb erfolgreich beendet. Wenn der erste Versuch nicht erfolgreich war, wiederholt der Transceiver selber bis zu fünfmal die Kalibration und meldet dann das Ergebnis. Somit kann die zweite Antwort bereits nach 50ms, spätestens aber nach 250ms eintreffen. Wenn eine Kalibration nicht erfolgreich war, kann der Aufruf wiederholt werden.

---

Funktionsname	<b>WRITE EEPROM WR-enable</b>	<b>Funktion 2DH</b>
Funktionsaufruf	0B0H, 2DH, ADR	
	ADR	EEPROM Adresse, welche zum Schreiben freigegeben wird Adressbereich 0...FFH
Antwort	2DH, ADR	

Das EEPROM-Register an der Adresse ADDR wird für genau einen Schreibbefehl freigegeben.

Es ist erforderlich, die drei Befehle WRITE EEPROM WR-enable, WRITE EEPROM Data und WRITE EEPROM Adresse in dieser Reihenfolge unmittelbar nacheinander auszuführen. Wenn die Zeit vom ersten Befehl an bis zum Ende des dritten Befehls länger als 90ms dauert, wird das EEPROM nicht beschrieben. Dieser Ablauf schützt vor einer unbeabsichtigten Veränderung des EEPROM's.

Der vom Transceiver unbenützte Teil des EEPROM's (Adresse 0D0H...0FEH) kann von der Applikation frei verwendet werden. Die Kalibrationswerte des Transceivers, div. Konstanten sowie die Konfiguration sind ebenfalls in diesem EEPROM abgelegt und sind der Applikation zugänglich.

An der EEPROM- Adresse FFH befindet sich der Schreibschutz EE\_WRPROT für das EEPROM. Jedes der 8 Bits von EE\_WRPROT steuert einen EEPROM Bereich wie folgt:

Bitn = 0:	Adressbereich ist beschreibbar und lesbar (Werkseinstellung)	
Bitn = 1:	Adressbereich ist zum Schreiben gesperrt und nur noch lesbar	
EE_WRPROT, Bit0	Adresse 00H..1FH	Kalibrationswerte Frequenz und RSSI
EE_WRPROT, Bit1	Adresse 20H..6FH	Kalibrationswerte Temperaturkompensation
EE_WRPROT, Bit2	Adresse 70H..8FH	Kalibrationswerte Reserve
EE_WRPROT, Bit3	Adresse 90H..9FH	Hardwarekonstanten
EE_WRPROT, Bit4	Adresse A0H..AFH	Softwarekonstanten
EE_WRPROT, Bit5	Adresse B0H..CFH	aktuelle Transceiverkonfiguration
EE_WRPROT, Bit6	Adresse D0H..FEH	<b>frei verwendbar für Applikation</b>
EE_WRPROT, Bit7	Adresse FFH..FFH	EE_WRPROT (dieses Byte)

Der Schreibschutz EE\_WRPROT gilt nur für den Zugriff mit den Befehlen WRITE EEPROM und nicht für die anderen WRITE Befehle, welche auch ins EEPROM schreiben.

#### Hinweise:

- Obwohl das ganze EEPROM zum Beschreiben freigegeben ist, darf nur der für die Applikation zugeteilte Block (Adresse 0D0H...0FEH) beschrieben werden.
- Wenn das EEPROM von der Applikation als Speicher benutzt wird, sollte EE\_WRPROT auf b'10111111' = BFH gesetzt werden, um das EEPROM vor ungewollter Veränderung zu schützen.
- Nachdem Bit7 von EE\_WRPROT im EEPROM auf 1 gesetzt worden ist, kann der Schreibschutz als Ganzes nicht mehr verändert werden. Dieser Befehl muss also gut überlegt angewendet werden!
- Werden andere EEPROM –Daten wie z.B. Kalibrationswerte verändert, wird der Transceiver unter Umständen unbrauchbar und muss vom Hersteller neu kalibriert werden. Falls jedoch die Kalibrierdaten vom Anwender gespeichert worden sind, können diese erneut ins EEPROM geschrieben werden.

Mit der EEPROM Schreibfunktion muss also vorsichtig umgegangen werden und die Werkskalibrierung darf nicht verändert werden! Es wird sehr empfohlen, ausschliesslich nur den EEPROM-Bereich an der Adresse D0H...FFH zu beschreiben

Siehe auch WRITE EEPROM Data und WRITE EEPROM Adresse.

---

Funktionsname	<b>WRITE EEPROM Data</b>	<b>Funktion 2EH</b>
Funktionsaufruf	0B0H, 2EH, DATA	
	DATA	EEPROM Datenbyte, welches zum Schreiben an die zuvor freigegebene Adresse gespeichert werden soll.
Antwort	2EH, DATA	

Datenbyte für das EEPROM-Register an der vorher bestimmten Adresse. Das EEPROM wird erst nach dem Erhalt des dritten Befehles mit DATA beschrieben.

Siehe auch WRITE EEPROM WR-enable und WRITE EEPROM Adresse.

---

Funktionsname	<b>WRITE EEPROM Adresse</b>	<b>Funktion 2FH</b>
Funktionsaufruf	0B0H, 2FH, ADR	
	ADR	Bestätigung der EEPROM Adresse, welche zum Schreiben freigegeben wurde. Adressbereich 0...FFH
Antwort	2FH, ADR	

Das DATA-Byte, das zuvor übertragen worden ist, wird nun ins EEPROM gespeichert, falls ADR mit der im ersten Befehl übertragenen Adresse übereinstimmt und die drei Befehle innerhalb max. 90ms eingetroffen sind.

Siehe auch WRITE EEPROM WR-enable und WRITE EEPROM Adresse.

---

## **Funktionsgruppe REPORT**

Die automatischen Ausgaben erfolgen in der Baudrate, mit der zuletzt eine Konfiguration gemacht wurde. Sie werden nur ausgegeben, wenn sich der Transceiver im Konfigurationsmodus befindet (MODE-Pin auf High).

---

Funktionsname	<b>REPORT READY</b>	<b>(automatische Datenausgabe)</b>
Funktionsaufruf	automatisch nach Powerup, MCLR\ - Reset oder Wakeup	
Automat. Ausgabe:	30H, RDY_status	
RDY_status	00H	Mikrokontroller nach Powerup oder MCLR\ - Reset bereit, Prüfsumme Kalibrationswerte OK (Dauer typ. 170ms bei 16MHz bzw. 180ms bei 8MHz $\mu$ P Clockfrequenz).
	01H	Mikrokontroller nach Wakeup bereit.
	80H	Mikrokontroller nach Powerup oder MCLR\ - Reset bereit, Prüfsumme Kalibrationswerte falsch!

Zeigt an, dass der Mikrokontroller des Transceivers bereit ist. Je nach Konfiguration des Transceivers dauert es nach dem wakeup eine gewisse Zeit, bis auch der HF-Teil wirklich bereit ist (3ms bei 19.2kbit/s Funkdatenrate, 5ms bei 2.4kbit/s Funkdatenrate).

Wenn die Kalibrationswerte im EEPROM verändert wurden, wird dies nach einem powerup oder MCLR\ - Reset im RDY\_status angezeigt.

Die automatische Ausgabe des REPORT READY erfolgt in der Baudrate, mit der zuletzt eine Konfiguration gemacht wurde und wird nur ausgegeben, wenn sich der Transceiver im Konfigurationsmodus befindet (MODE-Pin auf High).

**Hinweis:** Beim powerup und MCLR\ - Reset wird selbständig eine Kalibration durchgeführt und im Fehlerfall 5x wiederholt. Kann die Kalibration nicht erfolgreich beendet werden, wird trotzdem RDY\_status = 00H (bzw. 80H) ausgegeben.

Das Ergebnis der Kalibration kann nur mit dem Kalibrationsbefehl ermittelt werden.

---

---

**Funktionsgruppe ERROR**

---

Funktionsname        **ERROR**

Funktionsaufruf        Antwort auf eine ungültige Funktion oder Funktionswerte.

Automat. Ausgabe:    38H, ERR\_code

ERR\_code            00H    ungültige Funktion (Byte2)  
                      01H    ungültiger Funktionswert (Byte3)

Zeigt an, dass der Transceiver den Befehl nicht ausführen kann.

Der Transceiver gibt auf jede Konfiguration eine Antwort zurück. Kann kein gültiger Befehl erkannt werden, so wird ein Fehlercode ausgegeben, sodass der Befehl wiederholt werden kann.

---

## Werkseinstellung

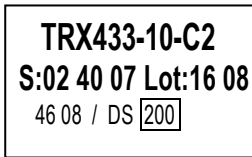
Ausgeliefert werden die Transceivermodule mit folgender Einstellung im EEPROM:

Funktion	Wert	Beschreibung
Frequenz RX	01H	433.0625 MHz bzw. 868.0125 MHz
Frequenz TX	01H	433.0625 MHz bzw. 868.0125 MHz
Funkdatenrate	04H	19.2 kbit/s
RS232 Baudrate	04H	19.2 kbaud
Powerup mode	25H	Continue nach Wakeup Run nach Powerup Alle Pins bleiben unverändert
Power control	03H	LED Transceiver ON HF-Teil Transceiver ON
Automat. Rekalibration control	01H	automatisch bei $\Delta T = 20^{\circ}\text{C}$ neu kalibrieren
Data Format	03H	CRC16 eingeschaltet Handshake eingeschaltet Fastsend ausgeschaltet RSSI des Telegramms nicht ausgeben Ruhe-RSSI nicht ausgeben

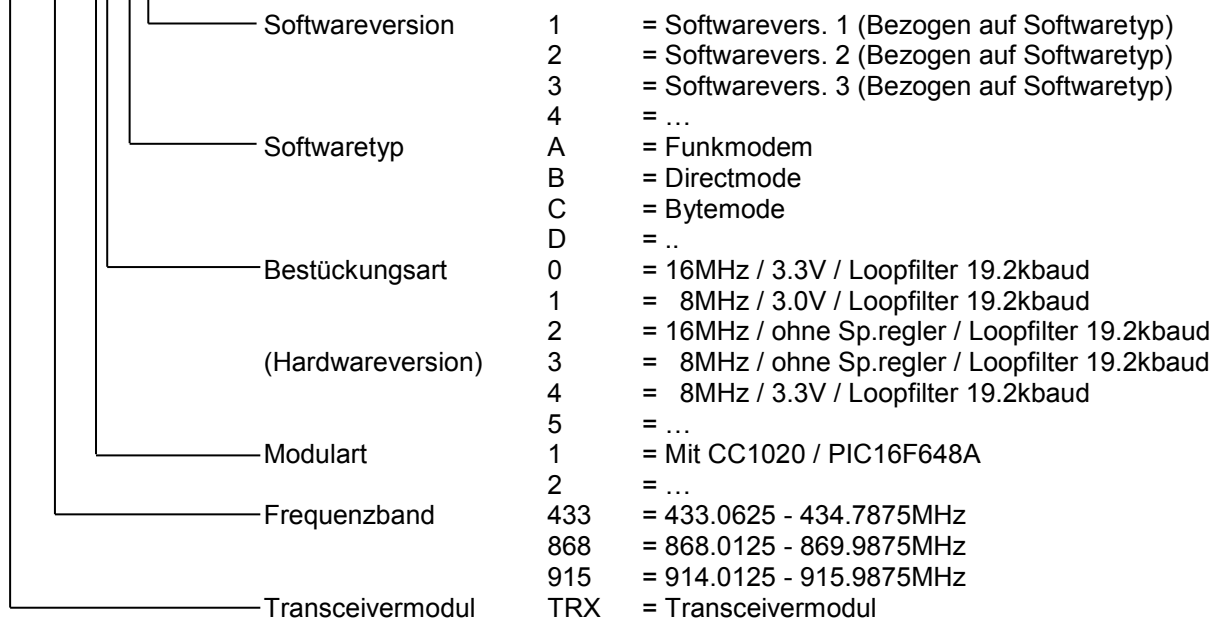


## Codierung Typenschild

Beispielkleber:

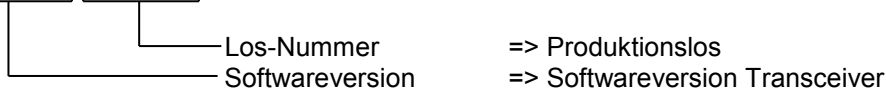


TRX433-10-C2:

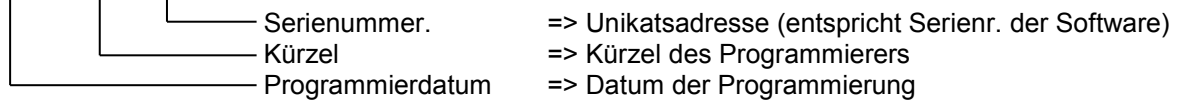


Hardwareversion = Bestückungsart + Modulart

S:02 40 07, Lot:16 08,



46 08 / DS 200



**CE Konformitätserklärung****DECLARATION OF CONFORMITY***KONFORMITÄT SERKLÄRUNG*

**Equipment** : Radio Transceiver Module (narrowband)  
*Geräteart*

**Brand** : wireless-design.ch  
*Handelsmarke*

**Type** : TRX433-10A, TRX868-10A, TRX433-10B, TRX868-10B  
*Typ*

**Further Description** : See technical data sheet at: [www.wireless-design.ch](http://www.wireless-design.ch)  
*Weitere Angaben* : *Siehe Technisches Datenblatt unter: [www.wireless-design.ch](http://www.wireless-design.ch)*

The signing legal authorities state, that the above mentioned equipment meets the requirements for EMC, Electrical Safety and for radio equipment and telecommunications terminal equipment and the mutual recognition of their conformity according to

- DIRECTIVE OF COUNCIL 89 / 336 / EEC, Electromagnetic Compatibility EMC
- DIRECTIVE OF THE COUNCIL 1999/5/EC of 9. March 1999 (R&TTE)

*Die Unterzeichnenden erklären als rechtsverbindliche Bevollmächtigte, dass das oben erwähnte Gerät den folgenden EMV-, Elektrischen Sicherheits-Anforderungen und Funkanlagen und Telekommunikationseinrichtungen und die gegenseitige Anerkennung ihrer Konformität entspricht, gemäss*

- *RICHTLINIE DES RATES 89 / 336 / EWG, Elektromagnetische Verträglichkeit EMV*
- *RICHTLINIE 1999/5/EG DES RATES vom 9. März 1999 (R&TTE)*

**The following harmonized standards have been used:**

*Folgende harmonisierte Normen wurden angewandt:*

**ETSI EN 300 220-3 V1.1.1 : 2000**

**ETSI EN 301 489-1 V1.4.1 : 2002**

**ETSI EN 301 489-3 V1.1.1 : 2002**

**Test report / Prüfbericht** : EMCKP585.1A, EMCKP585.2A, EMCKP585.2A, EMCKP585.3A

**Test Laboratory** : EMC-Testcenter Zurich AG, Postfach 268, CH-8052 Zurich  
**accredited according to EN 45001**

*Prüfstelle* : *EMC-Testcenter Zurich AG, Postfach 268, CH-8052 Zürich  
akkreditiert gemäss EN 45001*

**Manufacturer** **Name:** Willi Schmidiger GmbH, Gutenegg, CH-6125 Menzberg  
*Hersteller*

**Authority**  
*Bevollmächtigter*

Menzberg,  
*City*  
*Ort*

12.01.2006  
*Date*  
*Datum*

  
*Signature*  
*Unterschrift*