

Handbuch AMB8426-M

Version 1.6

SW-V1.6

AMBER wireless GmbH

Albin-Köbis-Straße 18

51147 Köln

Tel. +49 2203 98019 0

Fax +49 2203 98019 25

E-Mail info@amber-wireless.de

Internet <http://www.amber-wireless.de>

Inhaltsverzeichnis

1 Kurzbeschreibung	5
2 Kommunikation mit der Kundenapplikation	5
2.1 Transparente, gepufferte Datenübertragung	5
2.1.1 Versand von Daten über Funk.....	5
2.1.2 Empfang von Daten über Funk.....	6
2.2 Kommandobetrieb.....	7
3 Funktelegramm	8
4 Technische Daten	9
4.1 Eingangsspannung	9
4.2 Stromaufnahme AMB8426-M.....	9
4.3 Abmessung und Gewicht	9
4.4 Pin-Belegung	9
5 Serielle Benutzerschnittstelle	9
5.1 UART.....	9
5.1.1 Unterstützte Datenraten	9
5.1.2 Unterstützte Datenformate.....	9
5.2 SPI-Schnittstelle.....	10
6 Die Kommando-Schnittstelle	10
6.1 Datenübertragung im Kommando-Modus	10
6.1.1 CMD_DATA_REQ	10
6.1.2 CMD_DATARETRY_REQ.....	10
6.1.3 CMD_DATA_IND	10
6.1.4 CMD_RESET_REQ	11
6.1.5 CMD_SET_CHANNEL_REQ.....	11
6.1.6 CMD_SET_REQ.....	11
6.1.7 CMD_GET_REQ.....	12
6.1.8 CMD_SERIALNO_REQ.....	12
6.1.9 CMD_RSSI_REQ	13
6.1.10 CMD_ERRORFLAGS_REQ.....	14
6.1.11 CMD_SETUARTSPEED_REQ.....	14
6.1.12 CMD_FACTORYRESET_REQ.....	15
7 Factory-Reset	15
8 Betriebsartumschaltung per Pin	16
9 Konfigurationsparameter	16
9.1 Nichtflüchtige Konfigurationsparameter.....	16
9.1.1 UART_CTL0.....	19
9.1.2 UART_CTL1.....	20
9.1.3 UART_BR0.....	20
9.1.4 UART_BR1.....	20
9.1.5 UART_MCTL.....	20
9.1.6 UART_CMD_Out_Enable.....	20
9.1.7 UART_DIDelay	20
9.1.8 APP_MAXPacketLength.....	20
9.1.9 APP_AES_Enable (in Vorbereitung)	20
9.1.10 APP_AESX (in Vorbereitung)	21

9.1.11	APP_WOR_PeriodH.....	21
9.1.12	APP_WOR_PeriodL.....	21
9.1.13	APP_WOR_MultiplierH.....	21
9.1.14	APP_WOR_MultiplierL.....	21
9.1.15	APP_RX_Time.....	21
9.1.16	APP_RX_After_RX (in Vorbereitung)	21
9.1.17	MBUS_Coding.....	21
9.1.18	MBUS_HeaderLengthH.....	22
9.1.19	MBUS_HeaderLengthL.....	22
9.1.20	MBUS_Synch.....	22
9.1.21	MBUS_RXTIMEOUT	22
9.1.22	MBUS_Bl_ADD_Disable.....	22
9.1.23	MBUS_Bl_Control	23
9.1.24	MBUS_Bl_ManID1	23
9.1.25	MBUS_Bl_ManID2	23
9.1.26	MBUS_Bl_IDNr1	24
9.1.27	MBUS_Bl_IDNr2	24
9.1.28	MBUS_Bl_IDNr3	24
9.1.29	MBUS_Bl_IDNr4	24
9.1.30	MBUS_Bl_Version.....	24
9.1.31	MBUS_Bl_DevType	24
9.1.32	RF_Channel.....	24
9.1.33	RF_Power.....	25
9.1.34	RF_DataRate.....	25
9.1.35	RF_AutoSleep	25
9.1.36	RSSI_Enable.....	27
9.1.37	Mode_Preselect	27
9.1.38	Net_Mode.....	29
9.1.39	Config_CRC_Disable.....	29
9.1.40	CFG_Flags.....	29
10	Inbetriebnahme.....	31
10.1	Minimalkonfiguration	31
10.2	Übertragung großer Datenmengen	31
10.3	Verwendung der Low-Power Funktionalität	31
11	Firmware-Update	31
11.1	Firmware Versionshistorie.....	31
12	Referenzen.....	32
13	Informationen zur regulatorischen Konformität.....	33
13.1	Wichtige Hinweise.....	33
13.2	Konformitätserklärung.....	34
14	Wichtige Hinweise	35
14.1	Haftungsausschluss.....	35
14.2	Warenzeichen	35
14.3	Gebrauchsbeschränkung	35

Abkürzungen

PS Prüfsumme

PS Pin Select Betriebsartauswahl über Pins

BS Benutzerschnittstelle Serielle Schnittstelle zum Host

1 Kurzbeschreibung

Das Wireless M-Bus Funkmodul AMB8426-M wurde als Unterbaugruppe zur drahtlosen Ablesung von Zählern konzipiert. Es erfüllt die Anforderungen gemäß EN13757-4:2005 (Wireless M-Bus, siehe [1]) und kann überall dort zum Einsatz kommen, wo eine entsprechende Funkkommunikation benötigt wird.

Folgende Betriebsarten werden unterstützt:

- S1 / S1-m / S2
- T1 / T2 (Zähler / Andere)
- R2

Zur Kommunikation mit dem Host-System steht eine serielle Schnittstelle (UART, optional SPI) zur Verfügung, deren Datenrate und -format sich flexibel einstellen lässt.

Alle im Standard aufgeführten Übertragungsparameter (Kodierungsverfahren, Chiprate, Trägerfrequenz, Präambellänge etc.) lassen sich flexibel über die Benutzerschnittstelle einstellen.

Das Modul ist sowohl zum Einsatz im Zähler geeignet, kann aber auch in einem Repeater oder Konzentrator eingesetzt werden.

2 Kommunikation mit der Kundenapplikation

Über die serielle Benutzerschnittstelle erfolgen sowohl die transparente, gepufferte Datenübertragung (mit oder ohne Ergänzung und Auswertung des Block1) als auch die Gerätekonfiguration.

2.1 Transparente, gepufferte Datenübertragung

2.1.1 Versand von Daten über Funk

In diesem Modus werden bis zu 128 Byte an Nutzdaten über die serielle Schnittstelle zum Funkmodul übertragen und zunächst zwischengespeichert. Ein Längenbyte muss hierbei den eigentlichen Nutzdaten voran gestellt werden (siehe Abbildung 2-1).

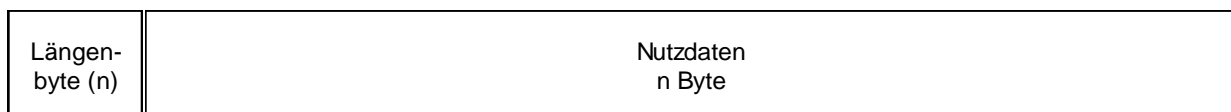


Abbildung 2-1 Empfang per Funk zu sender Daten mit Kontrolle von Block1

Nach Empfang aller Daten erfolgt der Versand über Funk entsprechend der gemäß [1] konfigurierten Betriebsart. Der Inhalt des ersten Blocks (C-Feld, M-Feld, A-Feld) kann optional aus dem nichtflüchtigen Speicher übernommen und vom AMB8426-M in das Funktelegramm eingefügt werden (entspricht Abbildung 2-1), ansonsten ist das Format gemäß Abbildung 2-2 zu verwenden (Auslieferungszustand, siehe Parameter MBUS_B1_ADD_Disable).

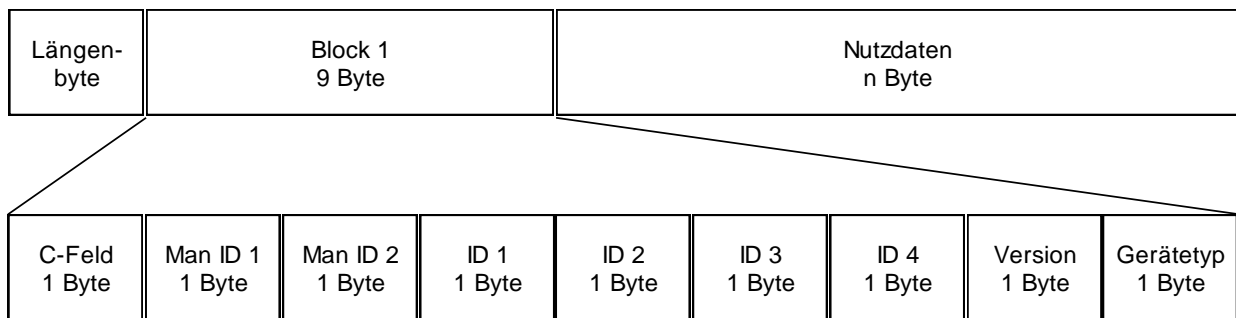


Abbildung 2-2 Empfang per Funk zu sender Daten inklusive Block1

2.1.2 Empfang von Daten über Funk

Die Ausgabe von Daten erfolgt ebenfalls mit voran gestellter Längeninformation. Zusätzlich kann ein Maß für die Verbindungsqualität in Form eines Feldstärkenwertes zur Verfügung gestellt werden (siehe `RSSI_Enable`).

Optional ist auch der Vergleich des ersten Blocks (M- und A-Feld) mit den Parametern im nichtflüchtigen Speicher möglich (siehe `MBUS_B1_ADD_Disable`).

Ist die Überprüfung aktiviert, so werden die empfangenen Daten nur dann wie in Abbildung 2-3 gezeigt ausgegeben, wenn M- und A-Feld (siehe Abschnitt `MBUS_B1_ManID1` und folgende) mit den im Funkmodul gespeicherten Werten übereinstimmen.

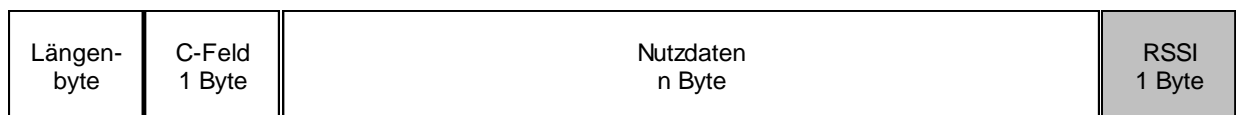


Abbildung 2-3 Ausgabe per Funk empfangener Daten mit Überprüfung von Block1

Ist die Überprüfung deaktiviert, so werden die empfangenen Daten immer komplett ausgegeben (siehe Abbildung 2-4).

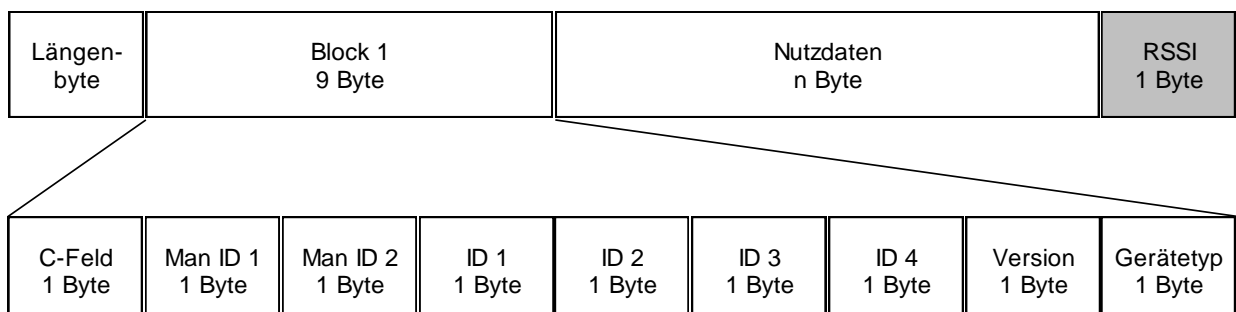


Abbildung 2-4 Ausgabe per Funk empfangener Daten ohne Überprüfung von Block1

2.2 Kommandobetrieb

In dieser Betriebsart erfolgt die Kommunikation mit dem Modul unter Verwendung vorgegebener Kommandos. Diese dienen z.B. dem Einstellen und Lesen von Betriebsparametern oder dem Ausführen spezieller Funktionen (wie z.B. einem Software-Reset oder dem Zurücksetzen der Betriebsparameter in den Auslieferungszustand).

Das Telegramm sieht nach dem Startzeichen (0xFFh) ein Kommando byte und eine Längeninformation der folgenden Daten vor. Eine Prüfsumme (XOR-Verknüpfung aller vorangegangenen Zeichen) muss als letztes Byte übermittelt werden.

Startzeichen 0xFF 1 Byte	Kommando 1 Byte	Längeninformation (n) 1 Byte	Daten n Byte	Prüfsumme 1 Byte
--------------------------------	--------------------	---------------------------------	-----------------	---------------------

Abbildung 2-5 serielle Schnittstelle im Kommandomodus

Eine Übersicht der gültigen Kommandos liefert Tabelle 1.

Kommando	Wert [hex]	Beschreibung
CMD_DATA_REQ 6.1.1	0x01	Daten werden übertragen
CMD_DATA_RETRY_REQ 6.1.2	0x02	Die zuvor an das Modul gesendeten Daten erneut senden
CMD_DATA_IND 6.1.3	0x03	Über wM-Bus empfangene Daten werden ausgegeben. (Data Indication)
CMD_RESET_REQ 0	0x05	Software-Reset
CMD_SET_CHANNEL_REQ 6.1.5	0x06	Kanal wählen
CMD_SET_REQ 6.1.6	0x09	Parameter des nichtflüchtigen Speichers schreiben
CMD_GET_REQ 6.1.7	0x0A	Parameter des nichtflüchtigen Speichers lesen
CMD_SERIALNO_REQ 6.1.8	0x0B	Seriennummer auslesen
CMD_RSSI_REQ 6.1.9	0x0D	Momentanen RSSI-Wert auslesen
CMD_SETUARTSPEED_REQ 6.1.11	0x10	Übertragungsgeschwindigkeit der Benutzerschnittstelle wählen
CMD_FACTORYRESET_REQ 6.1.12	0x11	Modul auf Fabrikeinstellungen zurücksetzen

Tabelle 1 Übersicht der Kommandos

Auf jedes Kommando vom Type „Request“ erfolgt eine Quittierung des Moduls in Form einer Nachricht vom Type „Confirm“. Hierzu wird das entsprechende Kommando byte um das Bit 0x80 ergänzt (so wird z.B. „0x0D“ zu „0x8D“, „0x11“ wird zu „0x91“) und von Modul zusammen mit entsprechenden Daten oder Statusinformationen zurückgegeben.

3 Funktelegramm

Wie in Absatz 2.1.1 erwähnt, wird zu den zu sendenden Nutzdaten der Inhalt des ersten Blocks aus dem nichtflüchtigen Speicher ergänzt (abschaltbar). Außerdem wird die vom Standard geforderte Prüfsumme für jeden Block berechnet und ergänzt.

Die grün gekennzeichneten Felder werden vom Modul ergänzt bzw. neu berechnet.

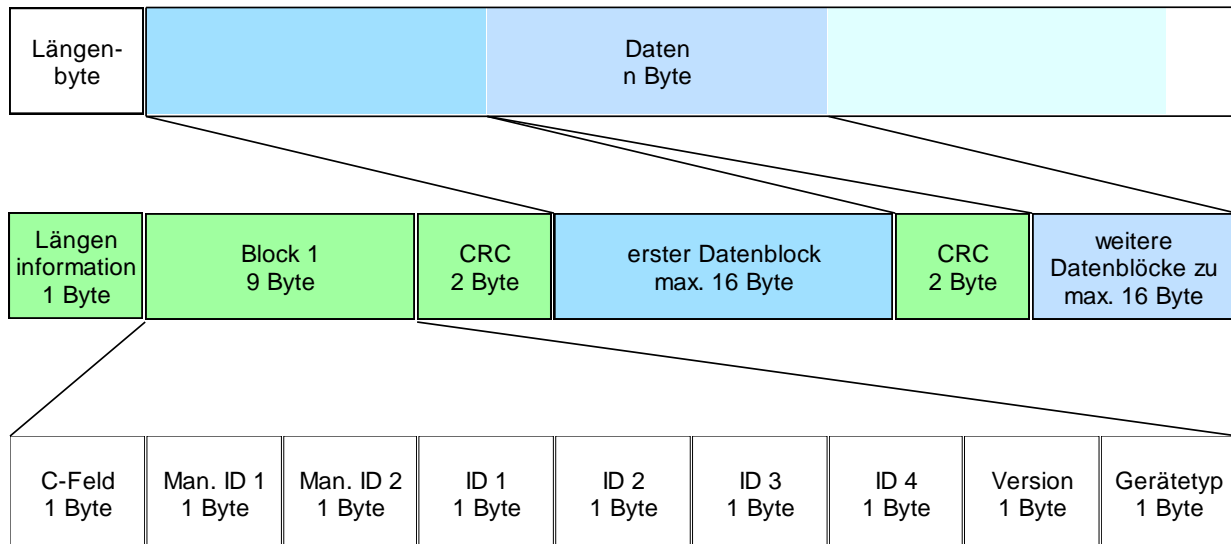


Abbildung 3-1 Funktelegramm mit Übernahme des Inhalts für Block1 aus dem nichtflüchtigen Speicher

Inhalt von Block 1	Parameter im nichtflüchtigen Speicher
C-Feld	MBUS_B1_Control 9.1.23
Man ID 1	MBUS_B1_ManID1 9.1.24
Man ID 2	MBUS_B1_ManID2 9.1.25
ID 1	MBUS_B1_ID1 9.1.26
ID 2	MBUS_B1_ID2 9.1.27
ID 3	MBUS_B1_ID3 9.1.28
ID 4	MBUS_B1_ID4 9.1.29
Version	MBUS_B1_Version 9.1.30
Gerätetyp	MBUS_B1_DevType 9.1.31

Tabelle 2 Einstellungen Funktelegramm Block1

4 Technische Daten

4.1 Eingangsspannung

Der Eingangsspannungsbereich des Moduls liegt bei 2,2 bis 3,6 Volt.

4.2 Stromaufnahme AMB8426-M

Vergleiche Datenblatt [3].

4.3 Abmessung und Gewicht

Vergleiche Datenblatt [3].

4.4 Pin-Belegung

Vergleiche Datenblatt [3].

5 Serielle Benutzerschnittstelle

5.1 UART

5.1.1 Unterstützte Datenraten

Die Einstellung der Datenrate erfolgt durch direkte Konfiguration der entsprechenden Parameter im nichtflüchtigen Speicher des Moduls (siehe `UART_BR0`, `UART_BR1` und `UART_MCTL` ab Seite 19) oder durch das Kommando `CMD_SETUARTSPEED_REQ` (6.1.11).

Aufgrund der Tatsache, dass die Geschwindigkeit der UART aus dem Takt des verwendeten Uhrenquarzes abgeleitet wird, kann es hierbei zu Abweichungen in einer Größenordnung < 0.5% kommen.

Bei Verwendung des PC-Programms „ACC“ ist per Drop-Down Menü die direkte Auswahl folgender Datenraten möglich:

110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 56000, 57600 und 115200 Baud.

Mit dieser Auswahl werden die oben genannten Register automatisch auf den optimalen Wert gesetzt.

Des Weiteren bietet das Programm „ACC“ auch einen Dialog zur Berechnung beliebiger Baudraten.

Die Default-Baudrate des Moduls beträgt 9600.

Die Ausgabe von Zeichen an der seriellen Schnittstelle erfolgt mit untergeordneter Priorität. Aus diesem Grunde kann es vorkommen, dass *zwischen* der Ausgabe einzelner Zeichen (z. B. bei Auftreten eines Interrupts) kurze Unterbrechungen auftreten.

5.1.2 Unterstützte Datenformate

Es werden die folgenden Datenformate unterstützt:

- 8 Bit
- Keine, gerade oder ungerade Parity
- 1 oder 2 Stopbits

In ACC ist per Drop Down Menü die direkte Auswahl folgender Datenformate möglich:

8n1, 8o1, 8e1, 8n2, 8o2, 8e2.

Auch in diesem Fall erfolgt die Einstellung durch direkte Konfiguration der entsprechenden Register des Mikroprozessors.

Das Default-Datenformat lautet: 8 Datenbits, keine Parity, 1 Stopbit („8n1“).

5.2 SPI-Schnittstelle

Alternativ zum UART-Interface steht auch eine SPI-Schnittstelle zur Verfügung (separate Firmware, in Vorbereitung).

6 Die Kommando-Schnittstelle

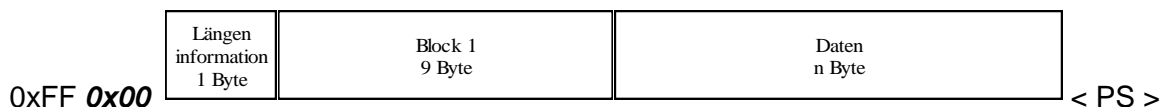
Die in 2.2 bereits erwähnten Kommandos sollen nachfolgend näher beschrieben werden. Über diese Schnittstelle lassen sich alle im Dokument EN 13757-4:2005 spezifizierten Einstellungen am Modul vornehmen.

6.1 Datenübertragung im Kommando-Modus

6.1.1 CMD_DATA_REQ

Dieses Kommando dient zur einfachen Datenübertragung. Die Übertragung erfolgt genau wie bei der transparenten Übertragung. Nach Abschluss des Sendevorgangs erfolgt die Rückgabe.

Format:



Rückgabe:

0xFF 0x80 0x01 < Status > < PS >

Status:

0x00: Daten versendet

6.1.2 CMD_DATA_RETRY_REQ

Dieses Kommando löst die erneute Sendung der zuvor an das Modul übergebenen Daten aus. Eine nochmalige Übertragung über die serielle Schnittstelle kann somit entfallen.

Die gepufferten Daten gehen verloren, sobald neue Daten per UART versendet oder Daten per Funk empfangen werden.

Format:

0xFF 0x02 0x00 0xFD

Rückgabe:

0xFF 0x82 0x01 < Status > < PS >

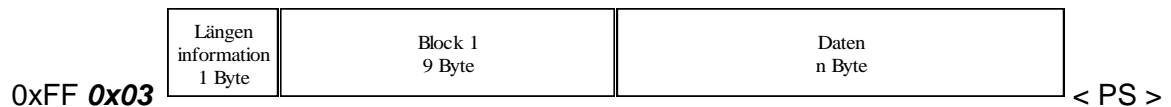
Status:

0x00: Daten versendet

6.1.3 CMD_DATA_IND

Dieses Telegramm zeigt den Empfang von Datenbytes an und stellt das Pendant zum Kommando CMD_DATA_REQ dar.

Format:



6.1.4 CMD_RESET_REQ

Dieses Kommando löst einen Software-Reset des Moduls aus. Der Reset wird nach Ausgabe der Quittung durchgeführt.

Format:

0xFF **0x05** 0x00 0xFA

Rückgabe:

0xFF **0x85** 0x01 < Status > < PS >

Status:

0x00: Erfolg

6.1.5 CMD_SET_CHANNEL_REQ

Dieses Kommando dient der Umschaltung des Funkkanals. Eine Kanalübersicht ist im Abschnitt 0 zu finden.

Format:

0xFF **0x06** 0x01 < 1 Byte Kanal > < PS >

Beispiel (Auswahl des Kanals 11):

0xFF 0x06 0x01 0x0B 0xF3

Rückgabe:

0xFF **0x86** 0x01 < Neuer Kanal > < PS >

Rückgabe für obiges Beispiel:

0xFF 0x86 0x01 0x0B 0x73

6.1.6 CMD_SET_REQ

Dieses Kommando erlaubt die direkte Manipulation der Parameter im nichtflüchtigen Speicher des Moduls. Hierbei wird auf die entsprechenden Parameter mittels der in Tabelle 6 beschriebenen Speicherstelle zugegriffen.

Es können sowohl einzelne als auch mehrere, hintereinander im Speicher liegende Parameter gleichzeitig verändert werden.



Achtung: Es erfolgt keine Gültigkeitsprüfung der übergebenen Parameter. Falsche Werte können zur Fehlfunktion des Gerätes führen!



Achtung: Das Abspeichern der Parameter im Flash-Speicher des Moduls erfordert es, dass zunächst das entsprechende Speichersegment komplett gelöscht und sodann aus dem RAM wiederhergestellt wird. Erfolgt während dieses Vorgangs ein Reset (z. B. durch Schwankungen der Versorgungsspannung), so kann der komplette

Speicherbereich zerstört werden. Das Modul ist in diesem Fall evtl. nicht mehr verwendbar, die Firmware muss per „ACC“ neu eingespielt werden. Empfehlung: erst die Konfiguration des Moduls mittels CMD_GET_REQ verifizieren, nur bei Bedarf schreiben.

Format:

0xFF **0x09** < Anzahl Bytes + 2 > < Speicherstelle > < Anzahl Bytes > < Parameter > < PS >

Rückgabe:

0xFF **0x89** 0x01 < Status > < PS >

Status:

0x00: Erfolg

0x01: Verifizierung fehlgeschlagen

0x02: ungültige Speicherstelle oder ungültige Anzahl zu schreibender Bytes (Schreibzugriff auf unerlaubten Bereich > 127 / 0x7F)

Beispiel 1: Hinzufügen von Block1 ändern (MBUS_B1_ADD_Disable, gemäß Tabelle 6 Speicherstelle 0x2Dh):

0xFF **0x09** 0x03 0x2D 0x01 < Parameterwert > < PS >

Beispiel 2: Setzen der 3 Register zur Baudrateneinstellung (UART_MCTL, UART_BR0 und UART_BR1). UART_BR0 hat gemäß Tabelle 6 die Speicherstelle 0x02h:

0xFF **0x09** 0x05 0x02 0x03 < UART_BR0 > < UART_BR1 > < UART_MCTL > < PS >

6.1.7 CMD_GET_REQ

Dieses Kommando erlaubt die Abfrage von einzelnen oder mehreren nichtflüchtigen Parametern (siehe 9.1). Es wird die angeforderte Anzahl an Bytes ab der angegebenen Speicherstelle zurückgegeben.

Es können sowohl einzelne als auch mehrere, hintereinander im Speicher liegende Parameter gleichzeitig abgerufen werden.

Format:

0xFF **0x0A** 0x02 < Speicherstelle > < Anzahl Bytes > < PS >

Beispiel (Abfrage aller Parameter):

0xFF **0x0A** 0x02 0x00 0x80 0x77

Rückgabe:

0xFF **0x8A** < Anzahl Bytes + 2 [0x82]> < Speicherstelle[0x00] > < Anzahl Bytes[0x80] > < Parameter > < PS >

Ein Lesezugriff auf den Speicherbereich hinter den in Tabelle 6 dokumentierten Parametern wird unterbunden. Speicherstelle und Anzahl der Bytes werden entsprechend limitiert. Somit ist die letzte Speicherstelle, die ausgelesen werden kann 127 (0x7F).

6.1.8 CMD_SERIALNO_REQ

Dieses Kommando dient der Abfrage der individuellen Seriennummer des Moduls.

Format:

0xFF **0x0B** 0x00 0xF4

Rückgabe:

0xFF **0x8B** 0x04 < 4 Byte Seriennummer > < PS >

Die Rückgabe erfolgt mit dem höchstwertigen Byte voran (MSB first), wobei dieses Byte das Produkt identifiziert („Produkt-ID“).

6.1.9 CMD_RSSI_REQ

Dieses Kommando liefert den aktuellen, vom Transceiver-IC ermittelten Empfangspegel als vorzeichenbehaftete Zahl (Zweier-Komplement) zurück.

Format:

0xFF **0x0D** 0x00 0xF2

Rückgabe:

0xFF **0x8D** 0x01 < Empfangspegel > < PS >

Der Empfangspegel $RSSI_{dBm}$ in dBm ergibt sich aus dem so erhaltenen Wert gemäß folgender Anleitung:

1. Konvertieren des Hexadezimalen Wertes in einen dezimalen $RSSI_{dez}$
2. Falls $RSSI_{dez} \geq 128$: $RSSI_{dBm} = (RSSI_{dez} - 256) / 2 - RSSI_{Offset}$
3. Ansonsten ($RSSI_{dez} < 128$): $RSSI_{dBm} = RSSI_{dez} / 2 - RSSI_{Offset}$

Bei $RSSI_{Offset}$ handelt es sich dabei um einen von der Datenrate abhängigen Korrekturfaktor gemäß Tabelle 3 (AMB8426).

Der Zusammenhang zwischen dem so errechneten Wert und dem physikalischen Empfangspegel in dBm ist nicht im gesamten Betriebsbereich linear und wird in Abbildung 6-1 dargestellt.

Datenrate	RSSI-Offset
1.2 kbps	74
38.4 kbps	74
250 kbps	78

Tabelle 3 Datenratenabhängiger RSSI-Offset für das AMB8426 (aus[2])

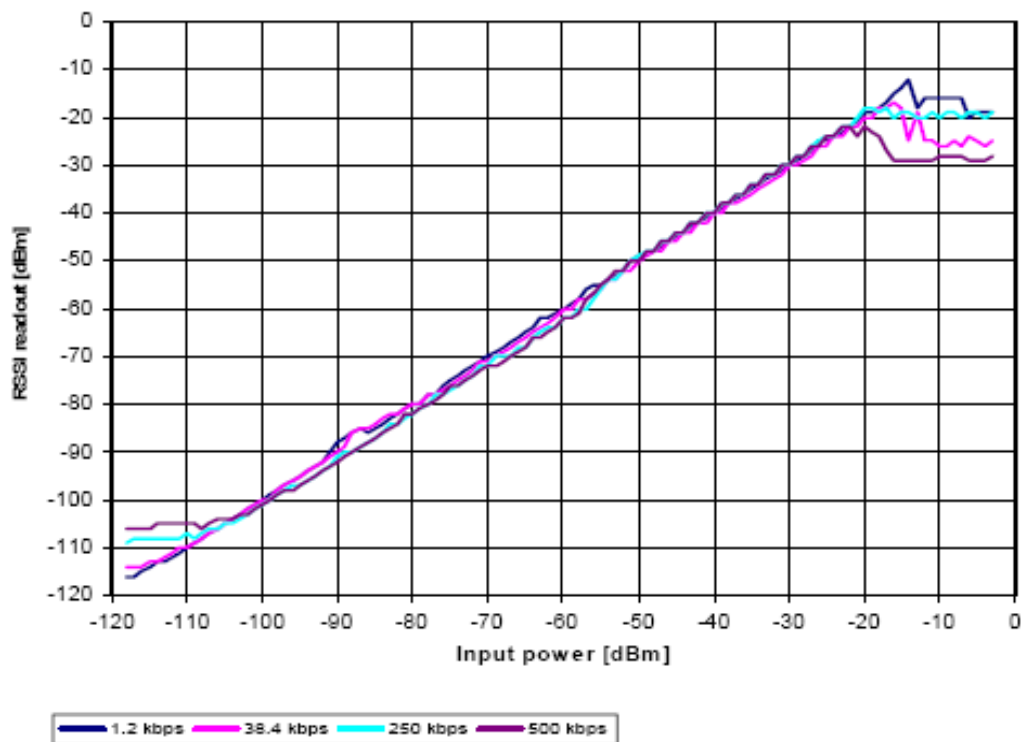


Abbildung 6-1

Zusammenhang zwischen Empfangspegel und ausgelesenem RSSI-Wert für das AMB8426 (aus[2])

6.1.10 CMD_ERRORFLAGS_REQ

Dieses Kommando liefert interne Fehlerzustände zurück.

Format:

0xFF **0x0E** 0x00 0xF1

Rückgabe:

0xFF **0x8E** 0x02 < Fehlerflags MSB > < Fehlerflags LSB >< PS >

Ein Rückgabewert der Fehlerflags von „0“ bedeutet, dass kein Fehler vorliegt. Nach der Abfrage und bei einem Reset wird der Wert zurückgesetzt.

Die Bedeutung der Fehlerflags wird hier nicht weiter beschrieben.

6.1.11 CMD_SETUARTSPEED_REQ

Dieses Kommando ändert die Parameter der seriellen Benutzerschnittstelle (UART_CTL0, UART_CTL1, UART_BR0, UART_BR1, UART_MCTL) auf voreingestellte Werte für feste Übertragungsraten ab. Der Index der Übertragungsrate ist Tabelle 4 zu entnehmen.

Übertragungsgeschwindigkeit [baud]	Index [hex]
1200	0x00

Übertragungsgeschwindigkeit [baud]	Index [hex]
2400	0x01
4800	0x02
9600	0x03
19200	0x04
38400	0x05
56000	0x06
115200	0x07

Tabelle 4 Fest vorkonfigurierte Übertragungsrate der seriellen Schnittstelle

Format:

0xFF **0x10** 0x00 < Index > < PS >

Rückgabe:

0xFF **0x90** 0x01 < Status > < PS >

Status:

0x00: Erfolg

6.1.12 CMD_FACTORYRESET_REQ

Dieses Kommando setzt alle Parameter auf Fabrikeinstellungen zurück.

Format:

0xFF **0x11** 0x00 0xEE

Rückgabe:

0xFF **0x90** 0x01 < Status > < PS >

Status: 0x00: Erfolg



Achtung: Nach dem Factoryreset sind auch die Settings für die Benutzerschnittstelle wieder auf 9600 Baud 8n1 zurückgesetzt!!!

7 Factory-Reset

Einmal ist ein Factory-Reset über das Kommando `CMD_FACTORYRESET_REQ` möglich. Ein anderer Weg ist der Boot-Up Factory-Reset.

Dazu muss nach einem Reset des Moduls Pin4.3 mit Pin4.4 verbunden sein.

Ist der Reset erfolgreich, Toggelt P21 (RX_INDICATE) 3 mal. War das Beschreiben des Flashs nicht erfolgreich, Toggelt P22 (TX_INDICATE) 3 mal. Anschließend wird automatisch ein Reset durchgeführt, um das Zurücksetzen der Parameter nochmals zu testen. In diesem Fall sollten Sie auf ihre Betriebsspannung achten.



Achtung: Nach dem Factory-Reset sind auch die Settings für die Benutzerschnittstelle wieder auf 9600 Baud 8n1 zurückgesetzt!!!

8 Betriebsartumschaltung per Pin

Des Weiteren steht dem Host eine Pin-Schnittstelle zur Umschaltung der voreingestellten Betriebsarten zur Verfügung. Ist diese aktiviert, wird zwischen den voreingestellten Wireless M-Bus Betriebsarten umgeschaltet. Zur Aktivierung dieser Funktionalität wird der Parameter

`Mode_Preselct` auf den Wert 0x0F eingestellt und die Pins der Schnittstelle je nach gewünschter Betriebsart beschaltet.

Pinbelegung				Ausgewählte Betriebsart
PinSelect Bit3	PinSelect Bit2	PinSelect Bit1	PinSelect Bit0	
0	0	0	0	Keine
0	0	0	1	S1
0	0	1	0	S1-m
0	0	1	1	S2
0	1	0	0	Reserved
0	1	0	1	T1 Zähler
0	1	1	0	T1 Andere
0	1	1	1	T2 Zähler
1	0	0	0	T2 Andere
1	0	0	1	Reserved
1	0	1	0	R2 Zähler
1	0	1	1	R2 Andere
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Unzulässig

Tabelle 5 Signale bei Modeumschaltung per Pin – Schnittstelle



Achtung: Bei periodischer Umschaltung sollte die Zustandsdauer für eine Betriebsart mindestens 2,5ms betragen, da ansonsten die Erkennung einer Präambel nicht gewährleistet ist!

9 Konfigurationsparameter

9.1 Nichtflüchtige Konfigurationsparameter

Die in der folgenden Tabelle aufgeführten nichtflüchtigen Parameter können mittels der entsprechenden Befehls im Konfigurationsmodus (`CMD_SET_REQ`, siehe 6.1.6) des Moduls bzw. durch Verwendung der Windows-Software „ACC“ modifiziert werden. Diese Parameter sind dauerhaft im Flash des Moduls abgelegt.



Achtung: Es erfolgt keine Gültigkeitsprüfung der übergebenen Parameter. Falsche Werte können zur Fehlfunktion des Gerätes führen!

Parameter	Beschreibung	Adr. [hex]	Zulässige Werte [hex]	Default [hex]
UART_CTL0	Kontroll-Register für Datenformat UART	0x00	Vgl. Tabelle 7	
UART_CTL1	Kontroll-Register für UART	0x01	0x80	
UART_BR0	Prescaler zur Baudrateneinstellung (LSB), zur Berechnung	0x02	Mit ACC berechnen	
UART_BR1	Prescaler zur Baudrateneinstellung (MSB), zur	0x03	Mit ACC berechnen	
UART_MCTL	Modulations- kontrollreg. UART	0x04	Mit ACC berechnen	
UART_CMD_Out_Enable	Ausgabe von empfangenen Frames als CMD	0x05	0x00, 0x01	0x00
UART_DIDelay	Konfigurierbare Ausgabeverzögerung	0x06	0x0000 – 0xFFFF	0x0000
UART_Reserved3	Reserve	0x08		0xFF
UART_Reserved4	Reserve	0x09		0xFF
APP_MAXPacketLength	Maximale Anzahl der zu sendenden / empfangenden Bytes	0x0A	0x0A – 0xFF	0x80
APP_AES_Enable (in Vorbereitung)	Verschlüsselung einschalten	0x0B	Vgl. Tabelle 8	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB	0x0C	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 1	0x0D	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 2	0x0E	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 3	0x0F	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 4	0x10	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 5	0x11	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 6	0x12	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 7	0x13	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 8	0x14	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 9	0x15	0x00 – 0xFF	0x00

APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 10	0x16	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 11	0x17	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 12	0x18	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 13	0x19	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 14	0x1A	0x00 – 0xFF	0x00
APP_AESX (in Vorbereitung)	Schlüssel AES128 MSB – 15 oder LSB	0x1B	0x00 – 0xFF	0x00
APP_WOR_PeriodH	Periodendauer in ms High-Byte	0x1C	0x00 – 0xFF	0x00
APP_WOR_PeriodL	Periodendauer in ms Low-Byte	0x1D	0x00 – 0xFF	0x32
APP_WOR_MultiplierH	Multiplikator der Periodendauer High-Byte	0x1E	0x00 – 0xFF	0x00
APP_WOR_MultiplierL	Multiplikator der Periodendauer Low-Byte	0x1F	0x00 – 0xFF	0x02
APP_RX_Time	WOR-Zeit die Modul in RX bleibt in ms	0x20	0x00 – 0xFF	0x14
APP_RX_After_RX (in Vorbereitung)	Zeit die das Modul nach Empfang auf Empfang bleibt	0x21	0x00 – 0xFF	0x00
APP_Reserved0	Reserve	0x22		0xFF
APP_Reserved1	Reserve	0x23		0xFF
APP_Reserved2	Reserve	0x24		0xFF
APP_Reserved3	Reserve	0x25		0xFF
APP_Reserved4	Reserve	0x26		0xFF
MBUS_Coding	Kodierungs-verfahren	0x27	Vgl. Tabelle 9	0x01
MBUS_HeaderLengthH	Headerlänge in Byte (High Byte)	0x28	0x00 – 0xFF	0x00
MBUS_HeaderLengthL	Headerlänge in Byte (Low Byte)	0x29	0x00 – 0xFF	0x04
MBUS_Synch	Synchronisationszeichen	0x2A	Vgl. Tabelle 10	0x00
MBUS_RXTIMEOUT	Zeit in ms die das Modul nach dem Versenden von Daten noch auf Empfang bleibt.	0x2B	0x00 – 0xFF	0x00
MBUS_Reserved1	Reserve	0x2C		0xFF
MBUS_Reserved2	Reserve	0x2D		0xFF
MBUS_Reserved3	Reserve	0x2E		0xFF
MBUS_Reserved4	Reserve	0x2F		0xFF
MBUS_B1_ADD_Disable	Hinzufügen von Block1 ausschalten	0x30	Vgl. Tabelle 12	0x01
MBUS_B1_Control	C-Feld von Block1	0x31	0x00 – 0xFF	0x44
MBUS_B1_ManID1	M-Feld von Block1	0x32	0x00 – 0xFF	0x00
MBUS_B1_ManID2	M-Feld von Block1	0x33	0x00 – 0xFF	0x00

MBUS_B1_IDNr1	A-Feld von Block1	0x34	0x00 – 0xFF	0x00
MBUS_B1_IDNr2	A-Feld von Block1	0x35	0x00 – 0xFF	0x00
MBUS_B1_IDNr3	A-Feld von Block1	0x36	0x00 – 0xFF	0x00
MBUS_B1_IDNr4	A-Feld von Block1	0x37	0x00 – 0xFF	0x00
MBUS_B1_Version	A-Feld von Block1	0x38	0x00 – 0xFF	0x00
MBUS_B1_DevType	A-Feld von Block1	0x39	0x00 – 0xFF	0x00
MBUS_Reserved5	Reserve	0x3A		0xFF
MBUS_Reserved6	Reserve	0x3B		0xFF
RF_Channel	Funkkanal	0x3C	Vgl. Tabelle 13	0x0B
RF_Power	Sendeleistung	0x3D	Vgl. Tabelle 14	0x05
RF_DataRate	Chiprate	0x3E	Vgl. Tabelle 15	0x01
RF_AutoSleep	Sleep Modus	0x3F	Vgl. Tabelle 16	0x00
RF_Reserved0	Reserve	0x40		0xFF
RF_Reserved1	Reserve	0x41		0xFF
RF_Reserved2	Reserve	0x42		0xFF
RF_Reserved3	Reserve	0x43		0xFF
RF_Reserved4	Reserve	0x44		0xFF
RSSI_Enable	RSSI Wert Ausgabe einschalten	0x45	Vgl. Tabelle 17	0x00
Mode_Preselect	Voreinstellungen für die M-Bus Betriebsarten	0x46	Vgl. Tabelle 18	0x03
Net_Mode	Modem, Konzentrator oder Repeatermodus	0x47	In Vorbereitung	0x00
Config_CRC_Disable	Prüfsumme im Konfigurations-modus beachten oder nicht	0x48	Vgl. Tabelle 19	0x00
Reserved0	Reserve	0x49		0xFF
Reserved1	Reserve	0x4A		0xFF
Reserved2	Reserve	0x4B		0xFF
Reserved3	Reserve	0x4C		0xFF
Reserved4	Reserve	0x4D		0xFF
Reserved5	Reserve	0x4E		0xFF
Reserved6	Reserve	0x4F		0xFF
CFG_Flags	Konfigurationsflags	0x50	0x0000 – 0xFFFF	0x0000

Tabelle 6 Nichtflüchtige Parameter

9.1.1 UART_CTL0

Mit Hilfe dieses Registers kann das Datenformat der seriellen Schnittstelle angepasst werden.

Bit Nr.	Beschreibung
0 (0x01)	Umschaltung zwischen UART (0x00h) und USART (0x01h)
1 bis 2 (0x02 – 0x04)	Reserviert, müssen immer auf 0x00 stehen
3 (0x08)	Dieses Bit wählt die Anzahl der Stopbits. Ist das Bit gesetzt, werden 2 Stop Bits verwendet, sonst 1.

4 (0x10)	Ist dieses Bit gesetzt, beträgt die Zeichenlänge 7 Bit, sonst 8 Bit.
5 (0x20)	Soll das MSB eines Bytes als erstes gesendet werden, muss dieses Bit gesetzt werden. Ansonsten bleibt es 0x00.
6 (0x40)	Bei gesetztem Bit gerade Parität ansonsten ungerade
7 (0x80)	Dieses Bits schaltet die Verwendung von Parität ein (wenn gesetzt).

Tabelle 7 Parameter UART_CTL0

9.1.2 UART_CTL1

Dieses Register wählt die Quelle zur Erzeugung der UART-Taktrate aus. Im Moment darf hierfür nur der Wert 0x80h verwendet werden.

9.1.3 UART_BR0

Zur Berechnung dieses Parameters sollte das Programm ACC verwendet werden.

9.1.4 UART_BR1

Zur Berechnung dieses Parameters sollte das Programm ACC verwendet werden.

9.1.5 UART_MCTL

Zur Berechnung dieses Parameters sollte das Programm ACC verwendet werden.

9.1.6 UART_CMD_Out_Enable

Über diesen Parameter lässt sich die Ausgabe von empfangenen Nachrichten umstellen. Ist diese Option aktiviert (0x01) werden empfangene Nachrichten im CMD-Format mit Startzeichen und Prüfsumme ausgegeben.

9.1.7 UART_DIDelay

Dieser Parameter bestimmt die Dauer in Millisekunden zwischen der Signalisierung ankommender Funkdaten über den Pin /DATA_INDICATION und der Ausgabe der Daten über die UART. Diese Verzögerung kann genutzt werden, um z. B. ein „schlafendes“ Host-System auf den Empfang der Daten vorzubereiten. Ab SW-Version 1.6 über CFG-Flags aktivierbar.

9.1.8 APP_MAXPacketLength

Über diesen Parameter kann die maximale Paketlänge eingestellt werden. Dies limitiert die Anzahl der über die serielle Schnittstelle ausgegebene Bytes für den Fall, dass das auf das Modul folgende Gerät nur über begrenzten Pufferplatz verfügt.

9.1.9 APP_AES_Enable (in Vorbereitung)

Beinhaltet dieser Parameter den Wert 0x00h so ist die AES128 Verschlüsselung ausgeschaltet. Ist der Wert 0x01h kommt der in den folgenden 16Byte eingetragene Schlüssel der AES128 Verschlüsselung zum Einsatz.

Wert	Beschreibung
0x00	Verschlüsselung ausgeschaltet
0x01	Verschlüsselung eingeschaltet

Tabelle 8 Parameter APP_AES_ENABLE

9.1.10 APP_AESX (in Vorbereitung)

In diesen Parametern wird der 128 Bit lange Schlüssel für die AES128 Verschlüsselung eingetragen. APP_AES0 beinhaltet dabei das MS Byte und APP_AESF das LS Byte,

9.1.11 APP_WOR_PeriodH

Dieser Parameter beinhaltet das High Byte für die Periodendauer des Aufwachens im WOR-Betrieb. Zusammen mit dem Parameter

9.1.12 APP_WOR_PeriodL

ergibt sich ein Parameter, welcher die Periodendauer des Aufwachens in ms vorgibt. Damit auch WOR – Zeiten von mehr als 65535ms erreicht werden können, sind die Parameter APP_WOR_MultiplierH und APP_WOR_MultiplierL vorgesehen.

9.1.13 APP_WOR_MultiplierH

und

9.1.14 APP_WOR_MultiplierL

sind ein Multiplikator für die Periodendauer im WOR –Betrieb des Moduls. Der Gesamtabstand zwischen den Aufwachzeiten beträgt somit:

$$\begin{aligned} \text{Aufwachzeit[ms]} \\ = ((\text{APP_WOR_PeriodH} \cdot 255) + \text{APP_WOR_PeriodL}) \cdot ((\text{APP_WOR_MultiplierH} \cdot 255) + \text{APP_WOR_MultiplierL}) \end{aligned}$$

d.h. immer wenn die Aufwachzeit abgelaufen ist, wacht das Modul auf und „Hört“, ob eine Präambel empfangen wird. Ist das der Fall, wird nach einem Synchronisationszeichen gesucht.

Achtung: Diesen Parameter nicht auf 0 stellen.

9.1.15 APP_RX_Time

Über diesen Parameter kann der Benutzer den Zeitraum einstellen, während dem das Modul im WOR-Betrieb auf Empfang sein soll. Es können bis zu 255ms eingestellt werden. Mindestens sollten jedoch 3 ms eingestellt sein um sicherzustellen, dass der Funk-Chip bereit ist. Wird in dieser Zeit eine Präambel empfangen, bleibt das Modul solange auf Empfang, bis ein Datensatz empfangen wurde bzw. bis die Präambel abbricht.

9.1.16 APP_RX_After_RX (in Vorbereitung)

Sollte es gewünscht sein, dass das Modul im WOR-Betrieb nach einem empfangenen Datensatz immer noch eine bestimmte Zeit auf Empfang bleibt, muss dieser Parameter diese Dauer in ms beinhalten.

9.1.17 MBUS_Coding

Über diesen Parameter kann die Codierung eingestellt werden.

Wert [hex]	Beschreibung
0x00	Kodierung 3 aus 6
0x01	Kodierung Manchester

Tabelle 9 Parameter MBUS_Coding

9.1.18 MBUS_HeaderLengthH

In diesem Parameter wird das High Byte der gesamten Anzahl der Bytes vor dem Synchronisationszeichen hinterlegt. Zusammen mit dem Parameter

9.1.19 MBUS_HeaderLengthL

ergibt sich dann eine 16 Bit große Zahl, die die Anzahl der Headerbytes beinhaltet. Die Berechnung der Anzahl der (01) Sequenzen ist dann:

$$\text{Anzahl(01)Sequenzen} = ((\text{MBUS_HeaderLengthH} \cdot 255) + \text{MBUS_HeaderLengthL}) \cdot 4$$

Bsp.: Ist der im High Byte enthaltene Wert 0x00h und der im Low Byte enthaltene Wert 0x46, so ist die Gesamtanzahl an Bytes 0x0046h (70). Jedes Byte Besteht aus 4 (01) Sequenzen. Nach der Formel ergeben sich somit 280 (01) Sequenzen. Somit reicht es aus, wenn das High Byte auf 0x00h und das Low Byte auf 0x46h gesetzt werden um eine gültige Headerlänge in der Betriebsart S1 (Langer Nachrichtenkopf) zu senden.

Die Dauer des Headers kann dann über die Formel

$$\text{Headerdauer}[s] = \frac{8 \cdot ((\text{HeaderLengthH} \cdot 255) + \text{HeaderLengthL})}{\text{Chiprate}[Hz]}$$

bestimmt werden.

9.1.20 MBUS_Synch

Der Wert in diesem Parameter ist der Index für die Wahl des Synchronisationszeichens.

Wert	Synchronisationszeichen
0x00	000111011010010110
0x01	0000111101

Tabelle 10 Parameter MBUS_Synch

Die Wahl des Synchronisationszeichens ist für Sende- und Empfangsvorgang wirksam.

9.1.21 MBUS_RXTimeout

Über diesen Parameter kann die Zeit in ms eingestellt werden, die das Modul nach dem Versenden von Daten noch auf Empfang bleibt bevor es in den Sleep Modus wechselt. Dieser Parameter ist in allen Betriebsarten wirksam. Beinhaltet der Parameter den Wert 0x00h, so ist keine Timeout aktiviert.

Wert	Beschreibung
0x00	RX-Timeout deaktiviert
> 0x00	RX Timeout in ms

Tabelle 11 Parameter MBUS_RXTimeout

9.1.22 MBUS_B1_ADD_Disable

Über diesen Parameter lässt sich das Einfügen des ersten zu übertragenden Blocks ausschalten. Der Inhalt für das C-, M-, und A-Feld muss somit über die serielle Schnittstelle vom Host (Abbildung 9-1) übertragen werden.

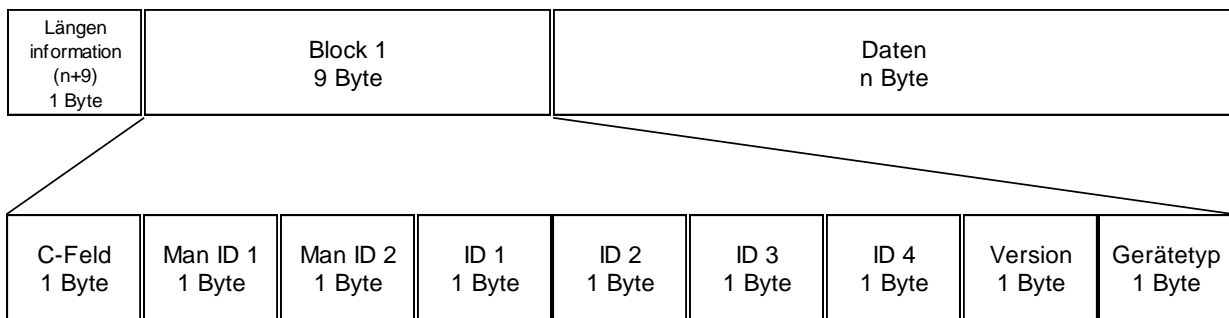


Abbildung 9-1 Einfügen von Block 1 Ausgeschaltet

Es findet dann ebenfalls kein Vergleich zwischen den im nichtflüchtigen Speicher abgelegten und den empfangenen Werten des M- und A-Feldes in Block1 statt. Die Daten werden mit optionalem Feldstärkewert (RSSI) ausgegeben (siehe Abbildung 9-2).

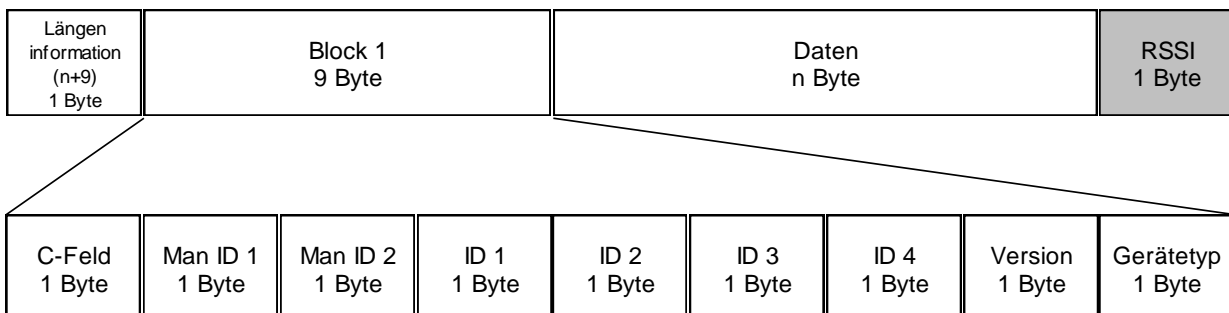


Abbildung 9-2 Ausgabe empfangener Daten bei abgeschaltetem hinzufügen von Block1

Wert	Beschreibung
0x00	Adressauswertung findet im Modul statt, Block1 wird im Modul hinzugefügt
0x01	Es findet keine Auswertung und kein Hinzufügen von Block1 im Modul statt

Tabelle 12 Parameter MBUS_B1_ADD_Disable

9.1.23 MBUS_B1_Control

Dieses Byte beinhaltet den Wert des C-Feldes im ersten Block der zu sendenden Daten.

9.1.24 MBUS_B1_ManID1

Dieses Byte beinhaltet den Wert des ersten Bytes des M-Feldes im ersten Block der zu sendenden Daten.

9.1.25 MBUS_B1_ManID2

Dieses Byte beinhaltet den Wert des zweiten Bytes des M-Feldes im ersten Block der zu sendenden Daten.

9.1.26 MBUS_B1_IDNr1

Dieses Byte beinhaltet den Wert des ersten Bytes des A-Feldes im ersten Block der zu sendenden Daten.

9.1.27 MBUS_B1_IDNr2

Dieses Byte beinhaltet den Wert des zweiten Bytes des A-Feldes im ersten Block der zu sendenden Daten.

9.1.28 MBUS_B1_IDNr3

Dieses Byte beinhaltet den Wert des dritten Bytes des A-Feldes im ersten Block der zu sendenden Daten.

9.1.29 MBUS_B1_IDNr4

Dieses Byte beinhaltet den Wert des vierten Bytes des A-Feldes im ersten Block der zu sendenden Daten.

9.1.30 MBUS_B1_Version

Dieses Byte beinhaltet den Wert des fünften Bytes des A-Feldes im ersten Block der zu sendenden Daten. (Version)

9.1.31 MBUS_B1_DevType

Dieses Byte beinhaltet den Wert des sechsten Bytes des A-Feldes im ersten Block der zu sendenden Daten. (Gerätetyp)

9.1.32 RF_Channel

Über diesen Parameter kann der gewünschte Sende / Empfangskanal ausgewählt werden.

Kanal	Wert	Frequenz [MHz]	Betriebsart
1	0x01	868,03	R2
2	0x02	868,09	R2
3	0x03	868,15	R2
4	0x04	868,21	R2
5	0x05	868,27	R2
6	0x06	868,33	R2
7	0x07	868,39	R2
8	0x08	868,45	R2
9	0x09	868,51	R2
10	0x0A	868,57	R2
11	0x0B	868,30	S1, S1-m, S2, T2
12	0x0C	868,95	T1,T2

Tabelle 13 Parameter RF_Channel

Alle Änderungen an diesen Parametern werden *ohne* Reset übernommen.

9.1.33 RF_Power

Erlaubt die Einstellung der Sendeleistung.

Leistung [dBm]	Wert
-5	0x01
0	0x02
+5	0x03
+7	0x04
+10	0x05

Tabelle 14 Parameter RF_Power

9.1.34 RF_DataRate

Die verschiedenen Chipraten sind über diesen Parameter auszuwählen.

Chiprate [kcps]	Wert	Betriebsart
4,8	0x00	R2
32,768	0x01	S1, S1-m, S2, T2
100,0	0x02	T1, T2

Tabelle 15 Parameter RF_DataRate

9.1.35 RF_AutoSleep

Im Parameter RF_AutoSleep werden 3 verschiedene Einstellungen unterstützt. Zum einen kann AutoSleep ausgeschaltet sein. In diesem Fall ist das Modul im durchgehenden RX Betrieb. Wenn der Parameter auf WOR eingestellt ist, wacht der Funk-Chip in den eingestellten Abständen auf und wechselt in den Empfangsbetrieb. Ist der Parameter auf Sleep eingestellt, legt sich das komplette Modul schlafen. Es kann nur noch über die Benutzerschnittstelle aufgeweckt werden (Daten oder Kommandomodus).

Wert [hex]	Beschreibung
0x00	Das Modul steht zwischen den Sendevorgängen permanent auf Empfang
0x01	Modul wacht nach den Einstellungen der Parameter APP WOR PeriodH , APP WOR PeriodL , APP WOR MultiplierH , APP WOR MultiplierL auf und hört für die eingestellte Zeit im Parameter APP RX Time ob es eine Präambel empfängt
0x02	Modul ist zwischen den Sendevorgängen immer im Sleep Modus (Es kann nichts empfangen werden)

Tabelle 16 Parameter RF_AutoSleep

- RX:

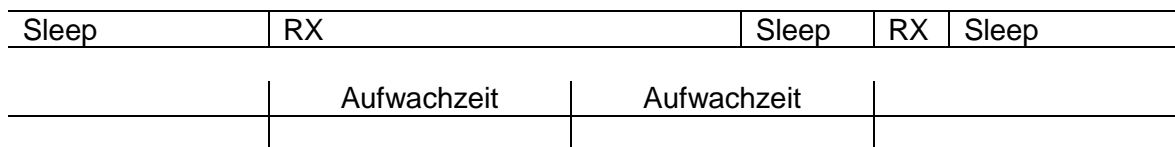
Das Modul ist ständig auf Empfang. Es können jeder Zeit Daten empfangen werden.

- WOR :

Wird das Modul in den WOR-Betrieb geschaltet (Parameter RF_AutoSleep = 0x01), wird ein Timer gestartet, der die über die Parameter APP_WOR_PeriodH, APP_WOR_PeriodL, APP_WOR_MultiplierH und APP_WOR_MultiplierL definierte Zeit abzählt um das Modul aufzuwecken.

Sleep	RX	Sleep	RX	Sleep	RX	Sleep
	Aufwachzeit		Aufwachzeit			
	Aufwachzeit		Aufwachzeit			
	RX Zeit		RX Zeit		RX Zeit	

Die oben gezeigte Grafik veranschaulicht das periodische Umschalten zwischen RX und Sleep, wenn das Modul in der RX Zeit keine Präambel empfängt. Wird eine Präambel und ein Datensatz empfangen, läuft die Aufwachzeit periodisch weiter.



- Sleep:

In diesem Modus hat das Modul die kleinste Stromaufnahme, da Funk-Chip und μC sich im Low Power Zustand befinden. Die Stromaufnahme beträgt in diesem Modus $<1\mu A$. Über die UART kann das Modul zum Versenden von Daten oder zur Konfiguration jeder Zeit geweckt werden.

9.1.36 RSSI_Enable

Beinhaltet dieser Parameter den Wert 1, so wird wie in Abbildung 2-3 bzw. Abbildung 2-4 gezeigt, der RSSI-Wert den Nutzdaten nachgestellt.

RSSI	Wert
Ausgeschaltet	0x00
Eingeschaltet	0x01

Tabelle 17 Parameter RSSI_Enable

9.1.37 Mode_Preselect

Über diesen Parameter können Voreinstellungen der in [1] spezifizierten Betriebsarten ausgewählt werden.

Betriebsart	Wert	Richtung(en)	Headerlänge [(01) Wechsel]	Synchronisationszeichen [Index]	Kanal [Index]	Chiprate [Index]	Codierung	Auto-Sleep [Index]
US	0x00	2	US	US	US	US	US	US
S1	0x01	TX	280	0	11	1	1	2
S1-m	0x02	TX	16	0	11	1	1	2
S2	0x03	2	US	0	11	1	1	US
Reserved	0x04							
T1 Zähler	0x05	TX	20	1	12	2	0	2
T1 Andere	0x06	TX	16	0	11	1	1	2
T2 Zähler	0x07	TX	20	1	12	2	0	US
		RX		0	11	1	1	US
T2 Ander	0x08	TX	16	0	11	1	1	US
		RX		1	12	2	0	US
Reserved	0x09							
R2 Zähler	0x0A	TX	40	0	US	0	1	US

Betriebsart	Wert	Richtung(en)	Headerlänge [(01) Wechsel]	Synchronisationszeichen [Index]	Kanal [Index]	Chiprate [Index]	Codierung	Auto-Sleep [Index]
		RX		0	6	0	1	US
R2 Andere	0x0B	TX	40	0	6	0	1	US
		RX		0	US	0	1	US
Reserved	0x0C							
Reserved	0x0D							
Reserved	0x0E							
Modeumschaltung	0x0F	PS	PS	PS	PS	PS	PS	US

Tabelle 18 Parameter Mode_Preselect

Die Sendeleistung wird immer aus den Wert des Parameter `RF_Power` übernommen. Sie muss also auch für die voreingestellten Modi konfiguriert werden.

- Beinhaltet der Parameter den Wert 0x00h (also keine Voreinstellungen), so werden alle Parameter aus dem nichtflüchtigen Speicher übernommen.
- Wenn das Modul an einem Zähler betrieben wird, der an einen ortsfesten Konzentrador in gewissen Abständen den Zählerstand übermitteln soll und dabei keinerlei Rückmeldung erwartet, kann die Betriebsart S1 ausgewählt und damit der Parameter `Mode_Preselect` auf den Wert 0x01h gestellt werden.
- Für den Fall, dass der Konzentrador nicht batteriebetrieben ist, und daher keine verlängerte Präambel erforderlich ist, kann auch die Betriebsart S1-m ausgewählt werden. Dazu kann dem Parameter `Mode_Preselect` der Wert 0x02h zugewiesen werden.
- Der Konzentrador der die Daten der Zähler sammelt, sollte für die vorigen beiden Fälle in der Betriebsart S2 (0x03h) arbeiten, da diese auch das Empfangen von Daten beinhaltet. Die Präambellänge ist dann über die Parameter `MBUS_HeaderLengthH` und `MBUS_HeaderLengthL` einzustellen. Genau wie das Verhalten des Moduls im Leerlauf (`Auto_Sleep`).
- Wenn ein Zähler immer in kurzen Datenblöcken seine Daten versenden soll, damit diese z.B. per Drive by ausgelesen werden können, kann das Modul im Modus T1 Zähler (`Mode_Preselect` = 0x05h) betrieben werden.
- Die Einstellung 0x06h die die Betriebsart T1 Andere auswählt stellt die gleichen Parameter wie die Betriebsart S1-m ein.
- Sollte der Zähler auch noch Daten empfangen können (Auslesen per Drive by und Konfiguration des Zählers per Funk) sollte das Modul in der Betriebsart T2 Zähler (0x07h) arbeiten.
- Im empfangenden Gerät welches den Zähler auch konfigurieren kann, müsste das Modul in der Betriebsart T2 Andere (0x08h) arbeiten, um mit dem Zähler in der Betriebsart T1 Zähler bzw. T2 Zähler kommunizieren zu können.
- Wenn Daten sehr häufig übermittelt werden sollen und dies über eine lange Funkstrecke geschehen soll, kann der Parameter `Mode_Preselect` auf den Wert 0x0Ah (Zähler) bzw. 0x0Bh (Andere) eingestellt werden und das Modul arbeitet in der Wireless M-Bus [1]

Betriebsart R2. Auch hier ist das Verhalten im Leerlauf gleich dem eingestellten Parameter RF_AutoSleep.

- Wenn zwischen den Betriebsarten schnell umgeschaltet werden muss, sollte der Parameter auf dem Wert 0x0Fh eingestellt sein. Dadurch wird die Umschaltung der voreingestellten Wireless M-Bus [1] Betriebsarten per Pins (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.**) aktiviert.

9.1.38 Net_Mode

Reserviert, muss auf 0x00h stehen.

9.1.39 Config_CRC_Disable

Über diesen Parameter lässt sich die Prüfsummenkontrolle beim Übertragen von Kommandos im Konfigurationsmodus ausschalten. Ist der Wert des Parameters 0x00h wird die Prüfsumme beachtet. Ansonsten wird die Prüfsumme nicht kontrolliert und ohne Überprüfung auf eine korrekte Übertragung verarbeitet.

Wert [hex]	Beschreibung
0x00	CRC ist erforderlich
0x01	CRC ist nicht erforderlich

Tabelle 19 Parameter Config_CRC_Disable

9.1.40 CFG_Flags

Dieser 2 Byte Große Parameter liefert die Möglichkeit verschiedene Konfigurationen durchzuführen. Er dient z.B. zum Aktivieren von verschiedenen Pin-Funktionen. Alle Pin-Funktionen sind im Auslieferungszustand deaktiviert, so dass die normale Pin-Konfiguration kompatibel zu den Vorgängerversionen ist. Eine Beschreibung der Flags finden Sie in Tabelle 20. Bei der Konfiguration dieses Parameters ist darauf zu achten, dass das LSB immer zuerst übertragen wird. Alle Einstellungen werden erst nach einem HW-Reset des Moduls übernommen.

Bit Nr.	Beschreibung
0 (0x0001)	Ist dieses Bit gesetzt, wird die Funktion des Pins /RTS aktiviert. Nach einem Reset des Moduls steht die /RTS Funktion dann an Pin P4.3 zur Verfügung. Dieser Pin symbolisiert in diesem Fall, ob der interne Puffer belegt ist (belegt = High).
1 (0x0002)	Ist dieses Bit gesetzt, wird die Funktion des Pins /CTS aktiviert. Nach dem Reset des Moduls steht die /CTS Funktion dann an Pin P4.4 zur Verfügung. Dieser Pin wird dann vor jedem auszugebenden Byte über die UART abgefragt (Low = es werden Daten ausgegeben).
2 (0x0004)	Ist dieses Bit gesetzt, wird die Funktion des Pins /DATA_INDICATE aktiviert. Nach dem Reset des Moduls steht die /DATA_INDICATE Funktion dann an Pin P4.5 zur Verfügung. Dieser Pin zeigt dann an, ob Daten zur Ausgabe über die UART bereitstehen. Er kann zum Beispiel zum Aufwecken eines nachgeschalteten Prozessors genutzt werden. Eine Verzögerung zwischen Anzeige am Pin und Ausgabe über

	die UART kann über den Parameter UART_DIDelay erfolgen.
3 bis 15 (0xFFFF8)	Reserviert

Tabelle 20 Konfigurationsflags

Beispiele:

Aktivierte Pins: /RTS + /CTS + /DataIndication

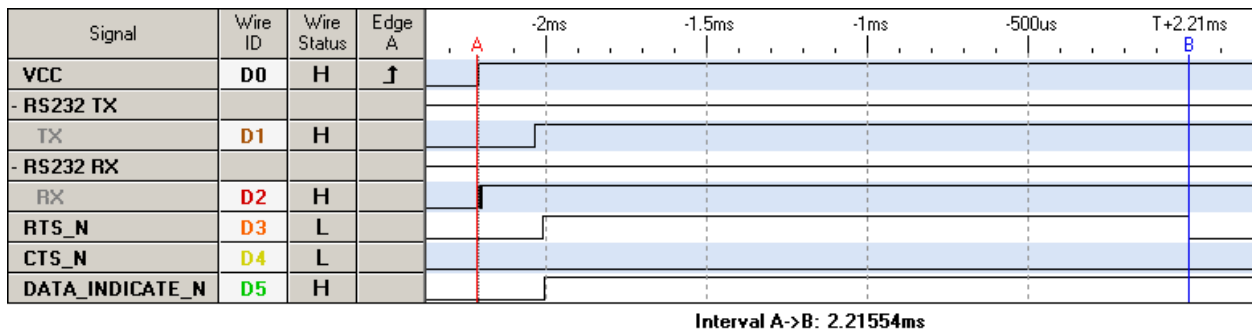


Abbildung 9-3 Einschaltverhalten

Beim einschaltverhalten signalisiert der /RTS-Pin, ab wann das Modul Daten über die UART empfangen kann (B). Diese Zeit kann variieren.

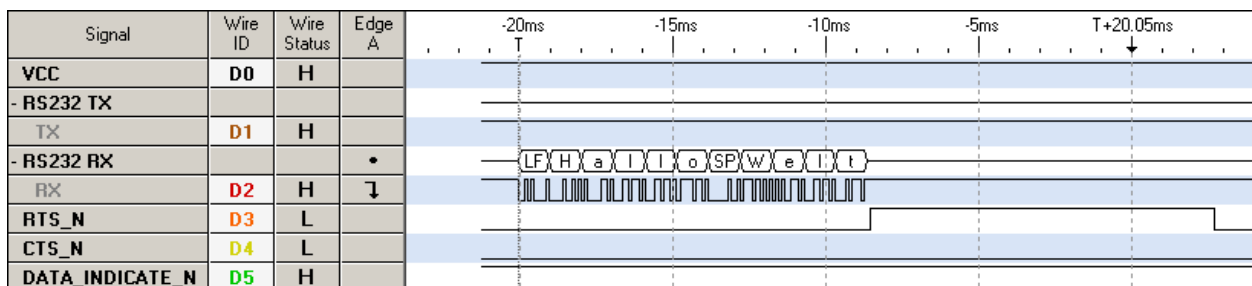


Abbildung 9-4 Versenden von Daten (Transparenter Modus)

Beim Versenden von Daten zeigt der /RTS Pin an, ab wann der Puffer für den Sendevorgang belegt ist (steigende Flanke /RTS) und wann die UART wieder auf empfang ist (fallende Flanke /RTS).

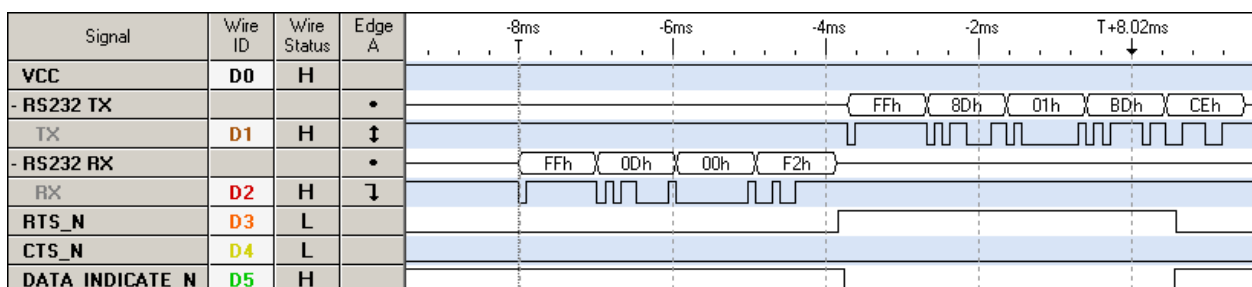


Abbildung 9-5 Kommando RSSI Request (UART_DIDelay = 0x0000)

Bei einem Kommando zeigt der /RTS Pin an, von wann bis wann der Puffer belegt ist. /DataIndication Zeigt an, dass Daten zum versenden über die UART bereit stehen.

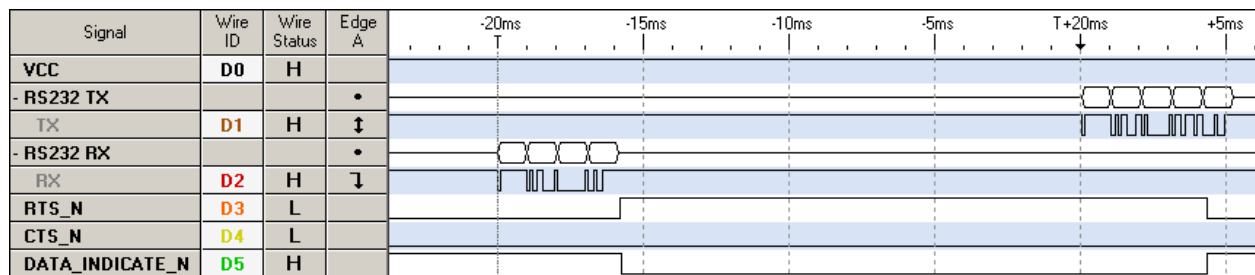


Abbildung 9-6 Kommando RSSI Request (UART_DIDelay = 0x0010)

10 Inbetriebnahme

10.1 Minimalkonfiguration

Die Module sind im Auslieferungszustand direkt betriebsbereit, folgende Pins sind in der Minimalkonfiguration erforderlich: VCC, GND, UTXD und URXD.

Soll das Modul an einem PC betrieben werden, ist die Verwendung eines Pegelwandlers (TTL auf RS232) erforderlich.

10.2 Übertragung großer Datenmengen

Bei der Übertragung größerer Datenmengen ist die beschränkte Puffergröße im Modul zu beachten, Daten können nur paketweise übertragen werden.

10.3 Verwendung der Low-Power Funktionalität

In diesem Fall empfiehlt sich die Benutzung des Parameters `RF_AutoSleep`. Wird dieser auf Sleep (0x02h) gestellt, so befindet sich das Modul außer während des Sendevorgangs im Schlafmodus. Während dieser Zeit ist das Empfangen von Daten nicht möglich. Wenn der Empfang bzw. das Hören vom Host aus gesteuert wird, kann das Modul durch Umkonfiguration dieses Parameters auf RX (0x00h) auf Empfang gesetzt werden. Wird eine Nachricht in dieser Zeit empfangen, leitet das Modul diese an den Host weiter und kann danach wieder „schlafen“ gelegt werden.

Soll das Modul selbständig in einem solchen WOR (Wake on Radio) Betrieb arbeiten, kann dies über den Parameter `RF_AutoSleep = 0x01h` aktiviert und über die Parameter `APP_WOR_PeriodH`, `APP_WOR_PeriodL`, `APP_WOR_MultiplierH`, `APP_WOR_MultiplierL` und `APP_RX_Time` konfiguriert werden.

11 Firmware-Update

Die Firmware des Moduls kann unter Verwendung des PC-Programms „ACC“ ab Version 2.5 über die serielle Schnittstelle erneuert werden. Falls das Modul nicht an einem PC betrieben wird, sollte für diesen Fall die UART des Moduls z. B. mittels geeigneter Steckkontakte zugänglich gemacht werden. Für diesen Vorgang sind lediglich die Signale UTXD und URXD erforderlich.

Zum Anschluss an den PC ist ein Pegelwandler (TTL auf RS232) erforderlich.

11.1 Firmware Versionshistorie

Version 1.0

- Erstes Release

Version 1.1

- Error Flags

Version 1.2

- Bugfix

Version 1.3

- Ergänzung des Befehls CMD_DataReq

Version 1.4

- Ergänzung des Befehls CMD_Data_IND
- Bugfix
- Boot Up Factory Reset

Version 1.5

- Bugfix

Version 1.6

- Aktivierbares Hardware Handshake (Config Flags; UART_DIDelay)

12 Referenzen

[1] Spezifikation Wireless M-Bus EN 13757-4:2005

[2] "CC1101 Single-Chip Low Cost Low Power RF-Transceiver (Rev. B)", Texas Instruments

[3] "AMB8426-M Datenblatt", AMBER wireless GmbH

13 Informationen zur regulatorischen Konformität

13.1 Wichtige Hinweise

Die Nutzung von Funkfrequenzen unterliegt nationalen Bestimmungen. Das AMB8426-M wurde unter Berücksichtigung der Anforderungen der R&TTE-Richtlinie 1999/5/EG der Europäischen Union (EU) entwickelt.

Das AMB8426-M kann ohne Notifizierung lizenzfrei innerhalb der EU verwendet werden. Gemäß der R&TTE-Richtlinie können aber beispielsweise Beschränkungen des Duty-Cycles oder der maximal zulässigen Sendeleistung gelten.

Konformitätsbewertung des Endprodukts

Die AMB8426-M ist eine Baugruppe und ist bestimmt für die Integration in andere Produkte (ein Produkt, in das ein AMB8426-M integriert wird, wird fortan als „Endprodukt“ bezeichnet).

Es liegt in der Verantwortung des Herstellers des Endproduktes sicherzustellen, dass das Endprodukt den Anforderungen der R&TTE-Richtlinie der Europäischen Union entspricht.

Die von AMBER wireless GmbH durchgeführte Konformitätsbewertung der Baugruppe AMB8426-M ersetzt nicht die Notwendigkeit der Konformitätsbewertung des Endprodukts bezüglich der R&TTE Richtlinie!

Haftungsausschlussklausel

Die relevanten regulatorischen Anforderungen können sich ändern. AMBER wireless GmbH übernimmt keine Garantie für die Richtigkeit und Vollständigkeit der oben aufgeführten Informationen. Richtlinien, technische Standards, Verfahrensanweisungen und ähnliches können von den jeweiligen nationalen Behörden unterschiedlich interpretiert werden. Ebenso können nationale Gesetze und Bestimmungen je nach Land variieren. Im Zweifelsfall oder bei Unsicherheit empfehlen wir, die jeweilig zuständigen Landesbehörden bzw. offiziellen Organisationen zu konsultieren. AMBER wireless GmbH ist von jeglicher Verantwortung und Haftung bezüglich der Einhaltung der regulatorischen Konformität freigestellt.

13.2 Konformitätserklärung



DECLARATION OF CONFORMITY Directive 1999/5/EC (R&TTE)

The manufacturer: AMBER wireless GmbH
Albin-Köbis-Straße 18
51147 Köln
Tel. +49-2203-699195-0

declares on its sole responsibility, that the following product:

Type-designation: **AMB8426**

Intended purpose: 868MHz transceiver module
Transfer of digital messages

satisfies all the technical regulations applicable to the product within the scope of council directives 1999/5/EC, 2004/108/EC and 2006/95/EC if used for its intended purpose and in accordance with the manufacturers operating instructions and that the following norms, standards or documents have been applied:

EN 300 220-1 V2.1.1 (2006-04)
EN 300 220-2 V2.1.2 (2007-06)
EN 301 489-1 V1.8.1 (2008-04)
EN 301 489-3 V1.4.1 (2002-08)
EN 60950-1 (2006 + A11:2009 + A1:2010)
EN 50371 (2002)

Köln, 11th of April 2011
Place and date of issue



Manufacturer/Authorized representative
Ulf Knoblich

14 Wichtige Hinweise

14.1 Haftungsausschluss

AMBER wireless GmbH geht davon aus, dass die hierin befindlichen Angaben zum Zeitpunkt der Veröffentlichung zutreffend sind. AMBER wireless GmbH behält sich jedoch das Recht vor, technische Spezifikationen oder Funktionen der eigenen Produkte zu ändern, die Herstellung dieser Produkte oder den Support für eines dieser Produkte einzustellen, ohne dass es einer schriftlichen Ankündigung oder Mitteilung der Kunden bedarf. Der Kunde hat sicherzustellen, dass die ihm zur Verfügung stehenden Informationen gültig sind. AMBER wireless GmbH übernimmt keinerlei Haftung für den Gebrauch ihrer Produkte. Amber wireless GmbH erteilt weder Lizenzen an ihren Patentrechten, noch an anderen Rechten an ihrem geistigen Eigentum oder an Rechten Dritter. Der Kunde ist dafür verantwortlich, dass sein System oder seine Einheit, in das die AMBER wireless Produkte integriert wurden, den entsprechenden gesetzlichen Bestimmungen entspricht.

14.2 Warenzeichen

- AMBER wireless® ist ein eingetragenes Warenzeichen der AMBER wireless GmbH.

Alle anderen Warenzeichen, eingetragene Warenzeichen und Produktnamen sind das ausschließliche Eigentum der jeweils Berechtigten.

14.3 Gebrauchsbeschränkung

AMBER wireless Produkte sind nicht freigegeben für den Gebrauch in lebensunterstützenden oder lebenserhaltenden Systemen oder Einheiten, oder anderen Systemen, bei den davon ausgegangen werden kann, dass eine Fehlfunktion zu einem wesentlichen Personenschaden beim Nutzer führt. AMBER wireless Produkte sind weiterhin nicht freigegeben für den Gebrauch als wesentlicher Bestandteil jeglichen(r) lebensunterstützenden(r) oder lebenserhaltenden(r) Systems oder Einheit, dessen/deren Fehlfunktion zum Ausfall des/der lebensunterstützenden oder lebenserhaltenden Systems oder Einheit führen kann, oder dessen/deren Sicherheit oder Effektivität beeinflusst werden kann. AMBER wireless Kunden, die diese Produkte in solchen Applikationen verwenden oder sie für solche Verwendungen verkaufen, handeln auf eigenes Risiko und müssen AMBER wireless GmbH von allen Schäden freistellen, die durch den Verkauf zu ungeeigneten Zwecken oder die ungeeignete Verwendung entstehen.

Durch die Verwendung von AMBER wireless Produkten ist der Nutzer an diese Bedingungen gebunden.

Copyright © 2011, AMBER wireless GmbH. Alle Rechte vorbehalten.

AMBER wireless GmbH

Albin-Köbis-Straße 18
51147 Köln
Tel. +49 2203 98019 0
Fax +49 2203 98019 25
E-Mail info@amber-wireless.de
Internet <http://www.amber-wireless.de>