

# **Spezifikation BF-Bus Version 1.3**

**Februar 1994**

**Spezifikation erstellt von  
R. Bichler  
Tel.: +49-89-35099-372**

**Spezifikation überarbeitet von  
H. Stenger  
Tel.: +49-89-35099-490**

**Siemens AG  
80807 München 45  
Ingolstädterstr.16**

**Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patenterteilung oder GM-Eintragung vorbehalten.**

## 1. Inhaltsverzeichnis

|   |    |
|---|----|
| 1. Inhaltsverzeichnis .....                   | 2  |
| 2. Prinzip und physikalische Verbindung ..... | 3  |
| 3. Layer1 .....                               | 4  |
| 3.1. Übertragung eines Bytes .....            | 4  |
| 3.2. L1-Fehler .....                          | 5  |
| 3.3. Verhalten bei Layer1-Fehlern.....        | 5  |
| 3.4. Übertragung eines Layer2-Frames .....    | 5  |
| 3.5. Der Zustand SLEEPING .....               | 5  |
| 3.6. Prioritäten und Buszustände .....        | 6  |
| 3.7. Zeitkonstanten .....                     | 7  |
| 3.8. Beispiele mit Zeitdiagrammen .....       | 8  |
| 3.9. Erlaubte Toleranzwerte.....              | 10 |
| 3.10. Zustandsdiagramm für Layer1 .....       | 11 |
| 4. Layer 2.....                               | 12 |
| 4.1. UI-Frames.....                           | 12 |
| 4.2. Der Acknowledged-Mode .....              | 13 |
| 4.3. Beispiele .....                          | 14 |
| 4.4. Der (7,4)-Hamming-Code.....              | 17 |
| 4.4.1. Codierung.....                         | 17 |
| 4.4.2. Decodierung.....                       | 17 |
| 4.4.3. Beispiel.....                          | 18 |
| 4.5. Fehlerbehandlung .....                   | 19 |
| 5. Rundruf der Mobile .....                   | 19 |
| 6. Weitere Daten.....                         | 21 |

## 2. Prinzip und physikalische Verbindung

Der BF-Bus dient zum Austausch digitaler Nachrichten zwischen einem Funkgerät und Peripheriegeräten, wie zum Beispiel einem Hörer oder einer Testbox. Jeder Busteilnehmer soll mit jedem anderen Teilnehmer ohne Einschränkung kommunizieren können. Da einerseits manche Nachrichten viel Information enthalten und/oder zeitkritisch sind, andererseits aber auch Geräte am Bus hängen, die billig herstellbar und stromsparend sein sollen, wollen wir zunächst die Anforderungen auflisten, die zu dem von uns gewählten Design geführt haben (die einzelnen Punkte bedingen sich teilweise gegenseitig).

- Als Bustreiber muß die serielle Schnittstelle eines Standardcontrollers ohne "viel" Zusatzhardware einsetzbar sein.
- Nicht jeder Teilnehmer muß alle Features unterstützen, insbesondere gibt es Teilnehmer, die nur mit dem Funkgerät kommunizieren.
- Unterstützung von Teilnehmern, die den IDLE- oder POWERDOWN-Modus des Controllers benutzen.
- Datenübertragung mit 19200 Baud
- Priorisierung der Teilnehmer.
- Unterstützung eines "acknowledged mode".
- Feste maximale Wartezeit für den höchstpriorisierten Teilnehmer.
- Rundrufe sollen möglich sein.

Diese Anforderungen führten uns zu folgendem Prinzip:

Der BF-Bus ist ein 1-Leiter-Bus, an dem maximal 15 Teilnehmer per "wired-AND" angeschlossen werden können (Pegel 0V = logisch 0; 5V = logisch 1). Jeder Teilnehmer muß einen Pull-Up von 2.2 kOhm treiben können. Es gibt nur einen Pull-Up-Widerstand, der sich im Funkgerät befindet. Jedem Teilnehmer ist eine Adresse und damit eine Priorität zugeordnet, mit deren Hilfe Konflikte auf dem Bus vermieden bzw. aufgelöst werden.

Nachrichten werden in Paketen (den Layer2-Frames) übertragen, die von keinem Teilnehmer unterbrochen werden dürfen. Diese Partitionierung ist so gewählt, daß auf der einen Seite eine möglichst große Netto-Datenübertragungsrate gewährleistet ist, und andererseits die Übertragung wichtigerer Nachrichten nicht allzu lange verzögert wird.

*Ruhepos. 5V  
=> 0V = log. 1!*

Das Protokoll ist an das ISO/OSI-Layer-Modell angelehnt, wobei an dieser Stelle nur der Layer1, der Layer2 und der geräteunabhängige Teil vom Layer3 beschrieben wird.

Die einzelnen Layer lassen sich mit folgenden Stichworten umreißen:

**Layer1:** Priorität, Konfliktauflösung, Senden und Empfangen der Layer2-Frames. Der Layer1 ist für alle Teilnehmer identisch.

**Layer2:** Partitionierung und Datensicherung (Checksum, Acknowledge). Nicht jeder Teilnehmer muß alle Features vom Layer2 bereitstellen.

**Layer3:** Nachrichtenaustausch (geräteabhängige Protokolle).

Oben war von Teilnehmern und Geräten die Rede. Diese beiden Begriffe können jetzt definiert werden. Die Teilnehmer hängen am Bus und können miteinander kommunizieren. Ein Teilnehmer ist durch seine Adresse eindeutig bestimmt. Ein Gerät ist ein potentieller Teilnehmer, der durch die von ihm bereitgestellten Layer2-Features und durch das Layer3-Protokoll bestimmt ist.

In dem nachfolgendem Text, werden mehrmals die Konstanten  $N_1$ ,  $N_2$ ,  $N_3$  und  $N_4$  verwendet. Die Werte für diese Konstanten werden im Kapitel 3.7 näher erläutert.

### 3. Layer1

Der Layer1 hat die Aufgabe Layer2-Frames zu senden und zu empfangen. Ein Layer2-Frame ist aus der Sicht von Layer1 eine Folge von bis zu 29 Bytes, die "am Stück" übertragen wird, d.h. von keinem Teilnehmer unterbrochen werden darf. Der Layer1 wertet diese Bytes nicht aus; sie sind für ihn lediglich inhaltlose Bitfolgen. Dadurch hat der Layer1 auch keine Möglichkeit die Empfängeradresse auszuwerten. Das bedeutet, daß zunächst jeder Teilnehmer die Nachricht mithört, und erst im Layer2 entscheidet, ob die Nachricht an ihn adressiert ist.

#### 3.1. Übertragung eines Bytes

Die Übertragung eines Bytes soll im wesentlichen durch die serielle Schnittstelle eines Contollers realisierbar sein. Die Datenübertragung ist asynchron und erfolgt mit 19200 Baud, d.h. die Übertragung eines Bits dauert  $1/19200 \text{ s} \approx 52 \mu\text{s}$ . Gesendet wird ein Startbit, acht Datenbits (LSB zuerst) und ein Stopbit. Da ein 8051-Controller bei einer 9-Bit-Datenübertragung nicht in der Lage ist Stopbit-Fehler zu erkennen, wurde auf ein Paritybit verzichtet.

Ein übertragenes Byte setzt sich folgendermaßen zusammen:

- 1 Startbit = 0
- 8 Datenbits, LSB zuerst
- 1 Stopbit = 1

### 3.2. L1-Fehler

Der sendende Teilnehmer muß das von ihm gesendete Byte über den Bus empfangen, und mit dem Original-Byte vergleichen. Dies ist einerseits zur Konfliktauflösung wichtig und dient andererseits zur Datensicherung. Alle Teilnehmer empfangen das gesendete Byte und testen ob das Stopbit korrekt ist. Ein Layer1-Fehler liegt vor, wenn der Sender feststellt, daß das gesendete Byte nicht mit dem empfangenen übereinstimmt, oder wenn irgendein Teilnehmer einen Stopbit-Fehler feststellt. Was zu *ZUVOR wurde festgestellt, daß 8051 keine Stop-Bit-Fehler erkennen kann* tun ist, wenn ein Teilnehmer einen Layer1-Fehler feststellt, wird in Kapitel 3.3 behandelt. Ein Byte gilt als korrekt übertragen, wenn kein Teilnehmer einen Layer1-Fehler feststellt.

### 3.3. Verhalten bei Layer1-Fehlern

Wird von einem Teilnehmer ein Layer1-Fehler *5ms* erkannt, so schaltet er den Empfänger aus und sendet ein *5ms* d.h. er zieht den Bus für *5ms* auf logisch 0. Dadurch wird bei allen empfangenden Teilnehmern ein Layer1-Fehler erkannt (Stopbit-Fehler), die dann ihrerseits wieder ein BREAK-Signal erzeugen. Erfolgt das BREAK-Signal innerhalb einer Nachricht, so wird bei den Empfängern der bisher empfangene Teil der Nachricht verworfen und der Sender muß nach der Konfliktauflösung seine Nachricht wiederholen.

Da nicht unbedingt alle Teilnehmer gleichzeitig einen Layer1-Fehler erkennen, kann die Zeit in der der Bus logisch 0 ist auch länger als  $N_2$  ms werden. Ein BREAK-Signal mit einer Dauer von bis zu 32ms (kann unter Umständen von der Mobile gesendet werden) darf bei den Teilnehmergeräten zu keinem Fehlverhalten führen.

Die Teilnehmer müssen also nach dem Senden eines BREAKS warten, bis der Bus wieder in der Lage ist eine neue Nachricht aufzunehmen, d.h. im IDLE Zustand ist. Dazu setzen wir fest:

Nach einem BREAK beginnt der *2ms* wenn der Bus *2ms* logisch 1 war. Dadurch ist eine Synchronisation der Teilnehmer nach dem Auftreten eines Layer1-Fehlers gewährleistet. Die so unterbrochene Nachricht muß dann vom Sender wiederholt werden.

### 3.4. Übertragung eines Layer2-Frames

Wir definieren: Zwei (korrekt übertragene) Bytes, die zum selben Layer2-Frame gehören, dürfen zwischen dem Stopbit des ersten und dem Startbit des zweiten Bytes nicht mehr als 1,5 ms auseinander liegen. Gemessen wird dabei ab Mitte des Stopbits bis zur fallenden Flanke des Startbits.

Wenn von keinem Teilnehmer ein Layer1-Fehler festgestellt wurde und damit der Bus nach der Übertragung eines Bytes für  $N_1$  ms auf logisch 1 liegt, so wird der Layer2-Frame von den Empfängern als vollständig und korrekt übertragen (aus Sicht von Layer1) gewertet.

Der Sender darf jedoch erst nachdem der Bus *2,5ms* auf logisch 1 liegt seine gesendete Nachricht als akzeptiert und abgeschlossen werten. Detektiert der Sender zuvor ein Breaksignal auf dem Bus, so muß er die Nachricht wiederholen. Anschließend ist der Bus wieder bereit eine neue Nachricht aufzunehmen.

### 3.5. Der Zustand SLEEPING

Um Teilnehmer zu unterstützen, die sich immer wieder in einen Stromsparmodus (power down, idle, etc.) versetzen, muß eine Möglichkeit bestehen, diese rechtzeitig vor der Übertragung einer Nachricht zu wecken. Dazu definieren wir einen weiteren Buszustand SLEEPING. Die Idee ist folgende:

War der Bus in den letzten *20ms* logisch 1, so befindet er sich im Zustand *20ms*, und die Teilnehmer können sich in einen Stromsparmodus versetzen. Dieser Zustand kann nur durch ein BREAK-Signal verlassen werden (auch von den Geräten, die nicht in einen Stromsparmodus gehen). Wie bei der Konfliktauflösung müssen alle Teilnehmer, die dieses BREAK-Signal empfangen ihrerseits mit einem BREAK antworten. Danach (s.o.) befindet sich der Bus wieder im Zustand IDLE und eine Nachricht kann übermittelt werden.

Ein Teilnehmer darf sich nur in den Stromsparmodus begeben,

- während der Bus im Zustand SLEEPING ist.
- wenn er innerhalb ( $N_2 - 0,25$ ) ms nach dem "Wecken" in der Lage ist, ein BREAK-Signal zu senden.

### 3.6. Prioritäten und Buszustände

Jedem Teilnehmer  $T_i$  ist eindeutig eine Priorität  $p_i \geq 1$  zugeordnet. Diese Größe gibt an, wieviele Millisekunden sich der Bus im IDLE-Zustand befunden haben muß, bevor der Teilnehmer  $T_i$  senden darf. Aufgrund der erlaubten Anzahl von Busteilnehmer und aufgrund der definierten Zeiten ist gewährleistet, daß jeder Teilnehmer senden darf, bevor der Bus in den Zustand SLEEPING übergeht. Die Priorität  $p_i$  errechnet sich aus der  $N_i$

Damit kann man jetzt folgende Buszustände definieren:

| Name              | Definition   | Bedeutung   |
|-------------------|--|---|
| Sleeping          | Der Bus war in den vergangenen $N_3$ ms logisch 1.<br><i>20ms</i>                                    | Kein Teilnehmer will unmittelbar auf die gerade übertragene Nachricht senden, die Teilnehmer können sich in einen Stromsparmodus begeben. |
| IDLE              | Der Bus war in den vergangenen $t$ ms logisch 1, mit $N_1 \leq t \leq N_3$ .<br><i>2ms 20ms</i>      | Der Bus ist bereit eine Nachricht aufzunehmen, alle bis dato gesendeten Nachrichten sind von den Empfängern als abgeschlossen gewertet.   |
| IDLE <sub>p</sub> | Der Bus war in den vergangenen $t$ ms logisch 1, mit $N_1 + p \leq t \leq N_1 + 1 + p$<br><i>2ms</i> | Der Sender wertet seine Nachricht als akzeptiert.   |
| ACTIVE            | Der Bus war in den vergangenen $N_1$ ms nicht immer logisch 1  | Jeder Teilnehmer einer Priorität $\leq p$ darf senden. ( $p = \text{Adresse} + 1$ )   |
|                   |  | Der Bus ist nicht in der Lage eine (weitere) Nachricht aufzunehmen. Sendewillige Teilnehmer müssen auf den IDLE-Zustand warten.           |

Die Zeitmessung läuft dabei:

- ab Mitte des Stopbits, nach einer Startbedingung.
- ab steigende Flanke, nach einem BREAK.

Beachte: Die Zustände IDLE und IDLE<sub>p</sub> können gemeinsam auftreten !

Nach dieser Spezifikation ist einem Teilnehmer mit der Priorität  $p$  nicht verboten, stets mit dem Beginn des Zustandes  $IDLE_{p+1}$  zu senden. Da dies zu sich ständig wiederholenden Konflikten führen könnte, fordern wir:

Erhält der Layer1 im Zustand ACTIVE einen Sendeauftrag vom Layer2, so muß er prinzipiell in der Lage sein, im Zustande  $IDLE_p$  mit dem Senden zu beginnen. Das bedeutet, daß man bei der Implementierung konstante Verzögerungen zwischen den Layern von den Zeitkonstanten abziehen sollte. Selbstverständlich darf der Layer1 auch im Zustand  $IDLE_x$  mit  $x > p$  mit dem Senden beginnen, wenn er zu diesem Zeitpunkt einen Sendeauftrag erhält.

Wenn der Bus in Betrieb ist, überprüft niemand mehr, ob sich die Teilnehmer an die vorgeschriebenen Zeiten halten. Zum Beispiel könnte ein Teilnehmer nach einem BREAK zu senden beginnen, obwohl die vorgeschriebenen  $N_1$  ms Wartezeit bis zum Zustand IDLE noch nicht abgelaufen sind. Ein solches fehlerhaftes Verhalten muß mit Hilfe eines Simulators (BF-Bus-Tester) festgestellt werden, so daß nur korrekt arbeitende Geräte an den Bus angeschlossen werden.

### 3.7. Zeitkonstanten

Wir definieren:

$$\begin{aligned} N_1 &:= 2 \\ N_2 &:= 5 \\ N_3 &:= 20 \\ N_4 &:= 2,5 \end{aligned}$$

(Zeiten in ms)

Anforderungen:

- Maximale Zeitdifferenz zwischen dem Senden von zwei aufeinanderfolgenden Bytes in einer Nachricht:  
< 1,5 ms
- Maximale Zeitdifferenz zwischen fehlerhaftem Empfang eines Bytes und dem Senden eines BREAK-Signals:  
< 1,5 ms

### 3.8. Beispiele mit Zeitdiagrammen

#### 1. Aus dem Zustand SLEEPING eine Nachricht senden:

Die Sendeberechtigung (Erreichen des IDLE<sub>p</sub>-Statuses) eines Peripheriegeräts hängt mit dessen Sendepriorität zusammen. Laut Definition besteht zwischen Priorität p und Teilnehmeradresse folgender Zusammenhang:

$$p(\text{in ms}) = \text{Adresse} + 1$$

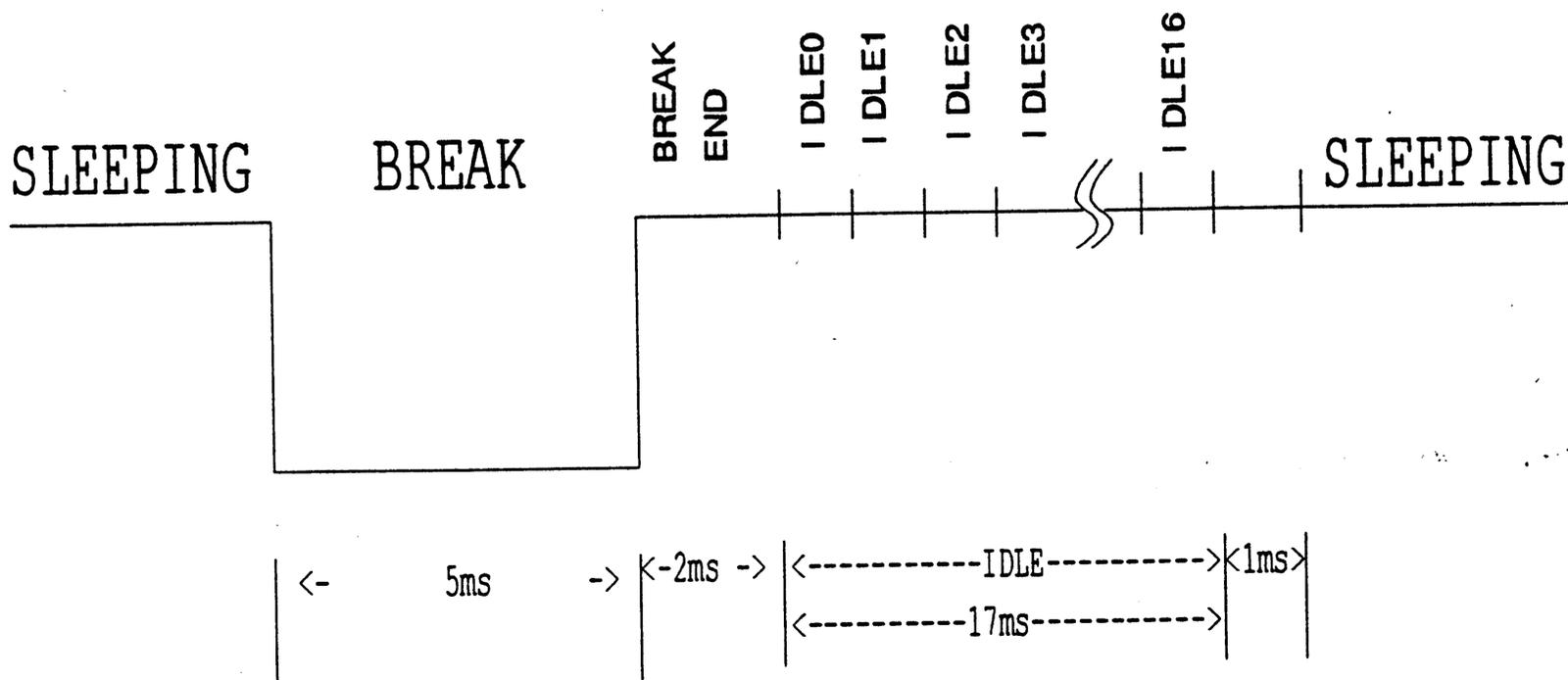


Bild 3.1 Senden aus dem Zustand SLEEPING

zu Bild 3.1: Die Mobile mit der Geräteadresse 0 und damit mit der Priorität 1 (Geräteadresse +1) darf frühestens zum Zeitpunkt IDLE<sub>1</sub> mit dem Senden beginnen. Zwei Millisekunden nach einem Break erreicht der Bus den Zustand IDLE<sub>0</sub>. Die Mobile muß eine weitere Millisekunde warten, ehe sie sendeberechtigt ist.

#### 2. Senden zweier aufeinanderfolgende Nachrichten

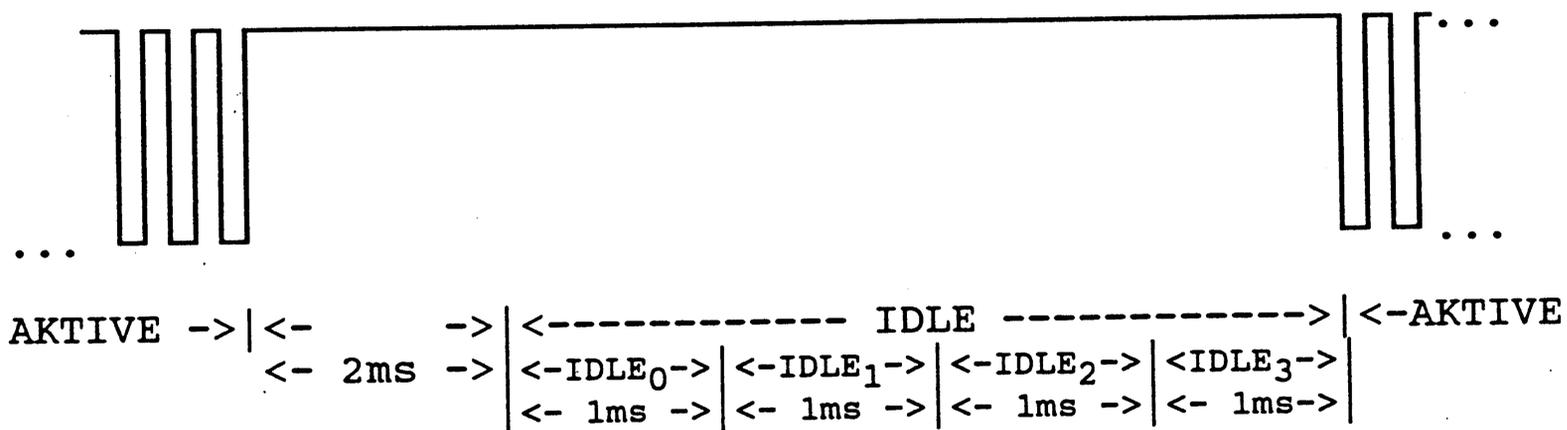
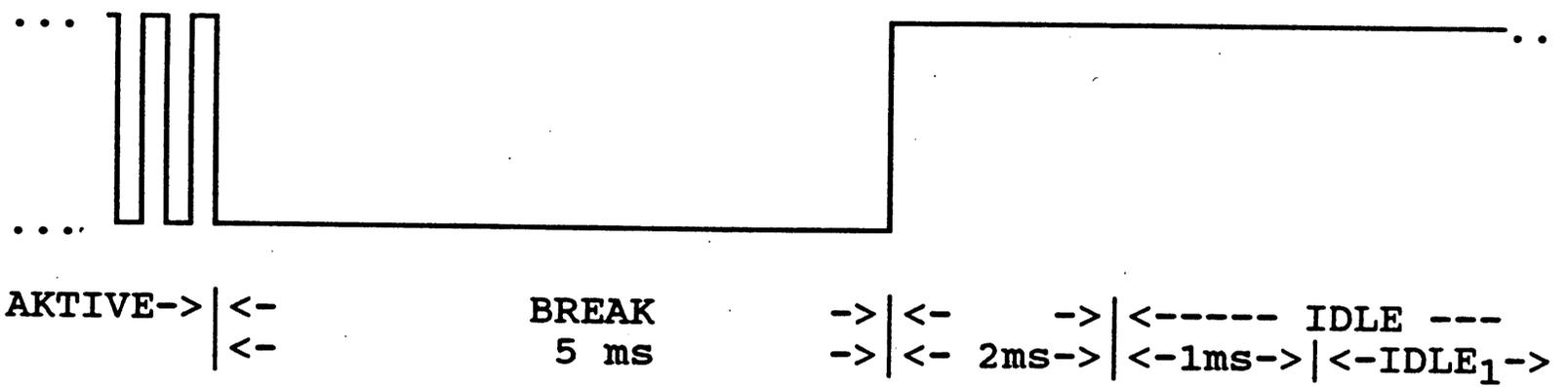


Bild 3.2 Senden zweier aufeinanderfolgende Nachrichten

zu Bild 3.2: Ein Gerät mit der Sendeadresse 3 darf nach Beendigung einer zuvor gesendeten Nachricht frühestens mit dem Beginn des Zustandes IDLE<sub>4</sub> mit dem Senden beginnen.

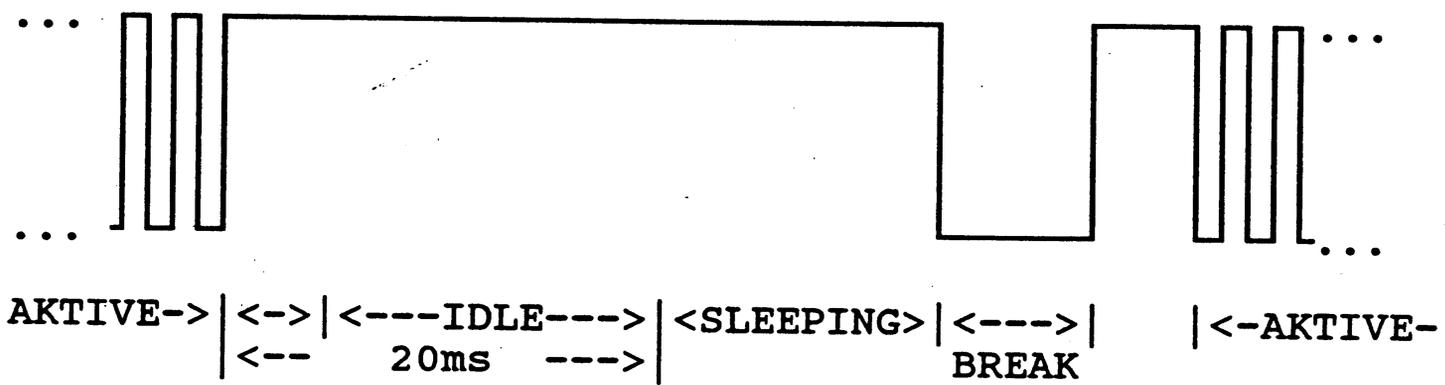
**3. Reaktion auf einen L1-Fehler**



**Bild 3.3** Reaktion auf einen Layer1-Fehler

zu Bild 3.3: Wird von einem Teilnehmergerät ein Layer1-Fehler festgestellt, so sendet dieses ein BREAK-Signal.

**4. Sleeping-Zustand**



**Bild 3.4** Der SLEEPING-Zustand

zu Bild 3.4: Befindet sich der Bus in dem Zustand SLEEPING, so kann dieser Zustand nur durch ein BREAK verlassen werden.

### 3.9. Erlaubte Toleranzwerte

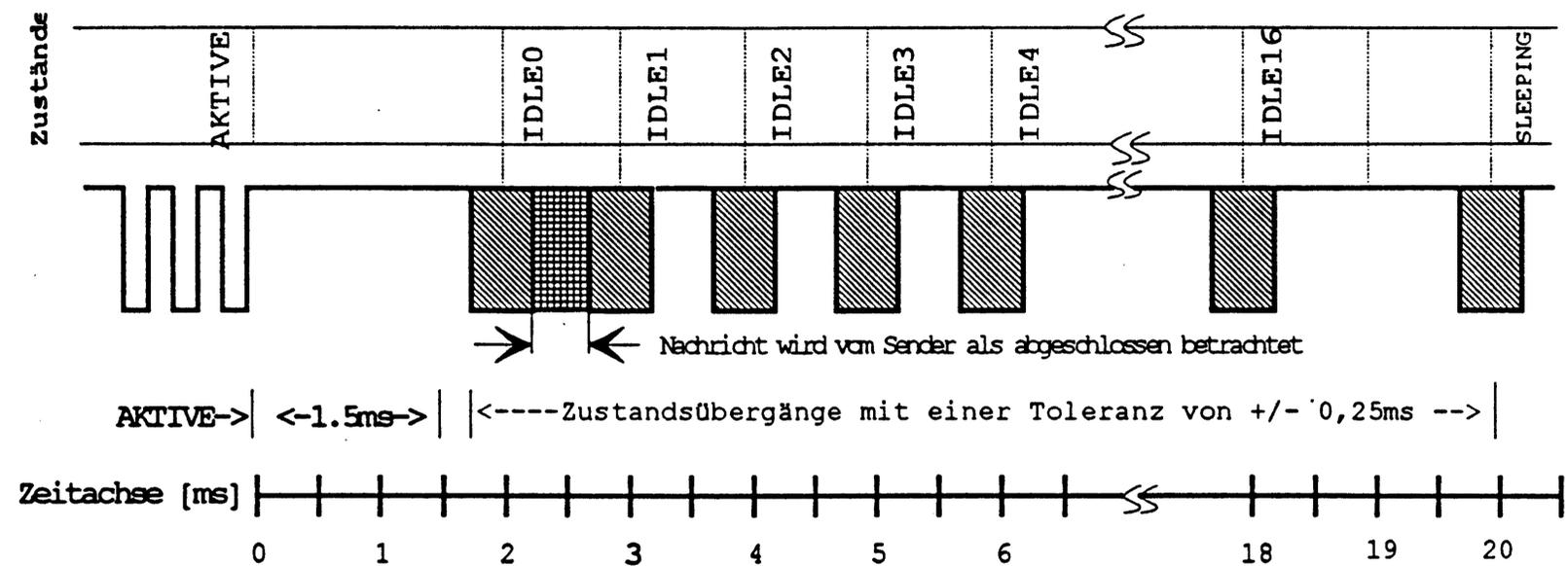


Bild 3.5 Toleranzen für das Senden einer Nachricht

zu Bild 3.5:

Wenn ein Teilnehmer eine Nachricht nicht akzeptiert (Layer1-Fehler erkannt), so muß das BREAK-Signal spätestens nach 1,5ms nach dem Empfang des letzten Stopbits kommen.

Sobald der Zustand IDLE0 erreicht wird, wird die Nachricht von den Empfängern als akzeptiert gewertet. Der Sender darf jedoch frühestens nach 2,25ms und spätestens nach 2,75ms (kariertes Toleranzfenster) nach dem letzten gesendeten Stopbit seine auf dem BF-Bus gesendete Nachricht als abgeschlossen betrachten. Sollte ein Teilnehmer entgegen der Spezifikation bei Erreichen des IDLE0-Zustandes ein Breaksignal senden, so entsteht für keinen Teilnehmer ein Nachrichtenverlust; unter Umständen erhält ein Teilnehmer die gesendete Nachricht doppelt.

Sämtliche IDLE<sub>p</sub> Zustände müssen innerhalb der oben angegebenen Toleranzfenster liegen:

IDLE<sub>p</sub>-Zustand wird erreicht, wenn der Bus in den vergangenen  $t$  ms logisch 1 war, mit  $t = N_1 + p$  (+/- 0,25ms).

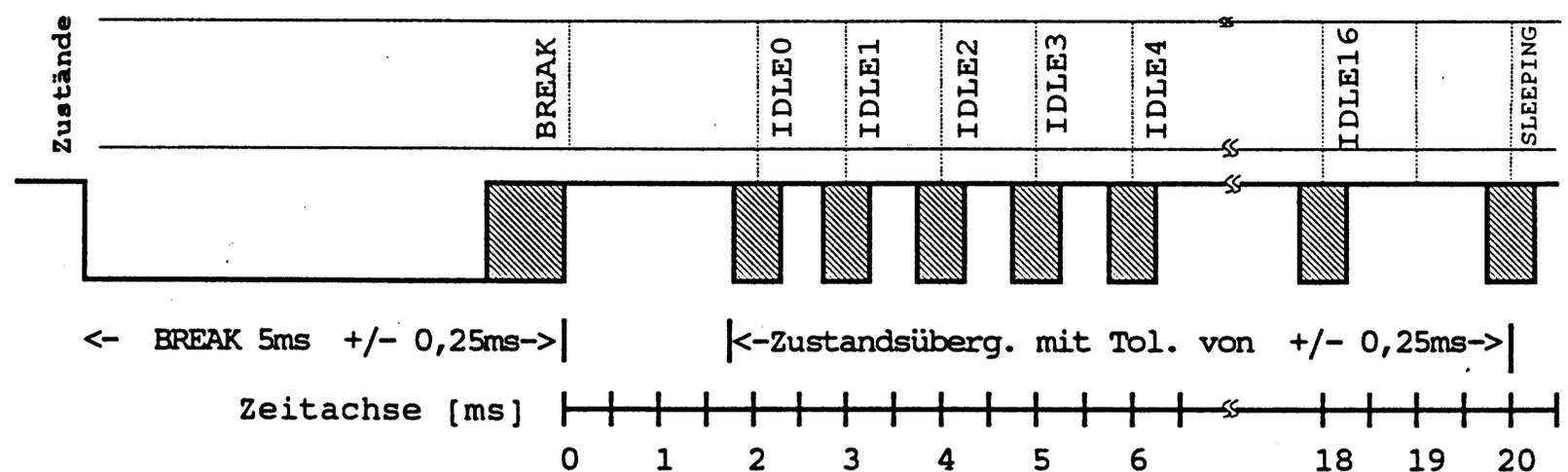


Bild 3.6 Toleranz für das BREAK-Signal

zu Bild 3.6:

Erlaubte Toleranz für die Länge des gesendeten BREAK-Signals:  $N_2$  (+/- 0,25 ms).

Jeder Teilnehmer muß nachdem er ein Break auf dem Bus erkannt hat (Stopbitfehler), innerhalb von  $(N_2 - 0,25)$  ms in der Lage sein, ein Breaksignal zu senden. Dadurch kann ein Breaksignal auch länger als  $N_2$  ms auf dem Bus andauern.

3.10. Zustandsdiagramm für Layer1

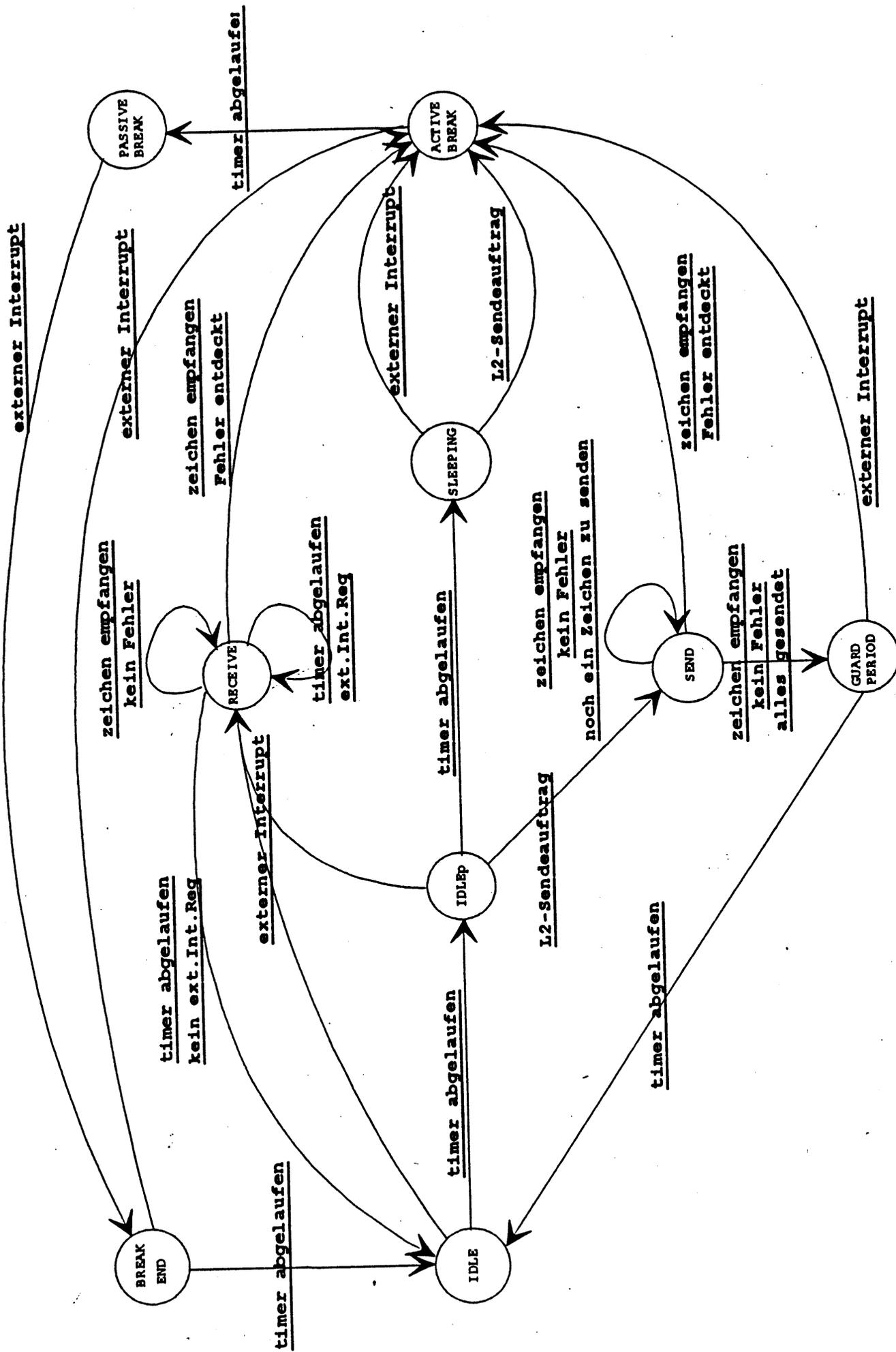


Bild 3.6 Zustandsdiagramm für den Layer1

## 4. Layer 2

Der Layer2 bereitet Layer3-Nachrichten für den Layer1 auf und sorgt für eine korrekte Datenübertragung. Es gibt prinzipiell 2 Arten von Layer3-Nachrichten:

1. Nachrichten, die in der Länge auf 24 Bytes beschränkt sind, und auf Layer2-Ebene nicht bestätigt werden.
2. Nachrichten, die in der Länge auf 24 Byte beschränkt sind, aber vom Layer2 des Empfängers bestätigt werden müssen.

Die in der Länge auf 24 Byte beschränkten Layer3-Nachrichten werden vom Layer2 in einem UI-Frame (unnumbered information) übertragen.

Jeder Layer2-Frame besteht also aus Layer3-Datenbytes (L3MU = Layer3 Message Unit), die zusätzlich vom Layer2 mit gewisser Rahmeninformation versehen sind. Da auf dem BF-Bus nur ein Kanal zur Verfügung steht (sowohl physikalisch als auch logisch), muß diese Rahmeninformation auf jeden Fall den Absender und den Empfänger der Nachricht spezifizieren. Außerdem enthält ein Layer2-Frame noch weitere (redundante) Informationen, die für das Layer2-Protokoll notwendig sind (siehe Kapitel 4.1).

Zur Identifizierung eines Teilnehmers  $T_i$  ist ihm eindeutig eine Adresse  $A_i$  ( $0 \leq A_i \leq 14$ ) zugeordnet.

### 4.1. UI-Frames

UI-Frames haben folgendes Format <sup>1</sup>.

|    |    |     |    |        |    |
|----|----|-----|----|--------|----|
| SA | EA | LAE | AF | {L3MU} | CS |
|----|----|-----|----|--------|----|

SA: Adresse des Senders, durch (7,4)-Hamming-Code (siehe Kapitel 4.4) verschlüsselt.

EA: Adresse des Empfängers, durch (7,4)-Hamming-Code (siehe Kapitel 4.4) verschlüsselt.

LAE: Anzahl der Bytes im Layer2-Frame (stets  $\geq 5$ ).

AF: Adressfeld.

Falls der Frame eine L3MU enthält, hat das Adressfeld folgende Bedeutung:

|           |   |   |      |
|-----------|---|---|------|
| Frame-Art | M | A | SAPI |
|-----------|---|---|------|

Enthält der Frame keine L3MU (d.h. LAE=5), so sieht das Adressfeld wie folgt aus:

|           |   |   |                 |
|-----------|---|---|-----------------|
| Frame-Art | 0 | A | Layer2-Funktion |
|-----------|---|---|-----------------|

Frame-Art (2Bit) = 0 für UI-Frames (eine andere Frame-Art existiert nicht).

<sup>1</sup>Das am weitesten links stehende Byte wird stets zuerst gesendet.

Das Mode-Bit M legt fest, ob der Frame zu bestätigen ist.

M = 0, falls Frame im unacknowledged mode gesendet wird, d.h. nicht bestätigt werden muß.  
 = 1, falls Frame im acknowledged mode gesendet wird, d.h. bestätigt werden muß.

Das Acknowledge-Bit A dient zur Bestätigung eines im Acknowledged-Mode empfangenen Frames.

A = 0, falls keine empfangene Meldung bestätigt werden muß.  
 = 1, um die letzte empfangene Meldung zu bestätigen.

Sapi (4Bit): Durch die SAPI (Service Access Point Identifier) wird eine Untereinheit im Empfänger angesprochen

Layer2-Funktion (4Bit): Als Layer2-Funktion existiert nur eine NULL-Funktion (Layer2-Funktion = 0) zum Bestätigen eines Frames, falls nichts anderes zu senden ist

CS: Checksum. Exclusive-Oder von SA, EA, LAE, AF und den Bytes der L3MU.

## 4.2. Der Acknowledged-Mode

Protokoll: Sendet ein Teilnehmer T<sub>1</sub> einen Layer2-Frame mit M=1 an einen Teilnehmer T<sub>2</sub>, so muß T<sub>2</sub> diesen Frame mit einem Layer2-Frame bestätigen in dem das A-Bit gesetzt ist. Diese Bestätigung muß erfolgen, bevor der Bus den Zustand SLEEPING annimmt. Falls T<sub>2</sub> keine Layer3-Nachricht zu senden hat, sendet er einen NULL-Frame mit gesetztem A-Bit, also:

|    |    |   |                  |    |
|----|----|---|------------------|----|
| SA | EA | 5 | 10 <sub>16</sub> | CS |
|----|----|---|------------------|----|

Erhält Teilnehmer T<sub>1</sub> keine Bestätigung von T<sub>2</sub>, bis der Bus den Zustand SLEEPING erreicht, oder empfängt T<sub>1</sub> einen Frame von T<sub>2</sub> mit A=0, so gilt die Nachricht als unbestätigt und wird von T<sub>1</sub> wiederholt. Empfängt T<sub>1</sub> allerdings die Bestätigung bevor er selbst in der Lage ist, den Frame zu wiederholen, so gilt der Frame als bestätigt.

Bei der Wiederholung gilt: Falls kein Frame von T<sub>2</sub> empfangen wurde bis die Wiederholung gesendet wird, wird das A-Bit auf 0 gesetzt. Ansonsten wird das A-Bit so gesetzt wie es der letzte von T<sub>2</sub> empfangene Frame erfordert. Für eine Implementierung bietet sich folgende Vorgehensweise an: T<sub>1</sub> schreibt den zu sendenden Frame in einen Puffer. Nach dem Senden wird das A-Bit auf 0 gesetzt und anschließend jeweils durch das M-Bit eines von T<sub>2</sub> empfangenen Frame ersetzt. Damit ist gewährleistet, daß das A-Bit stets korrekt gesetzt ist.

Bleibt die Nachricht dreimal unbestätigt, so geht T<sub>1</sub> davon aus, daß sich T<sub>2</sub> in einem Fehler-Zustand befindet, und muß entsprechend reagieren (teilnehmerabhängig).

Falls T<sub>1</sub> einen weiteren Frame sendet, bevor T<sub>2</sub> antworten konnte (und demnach auch bevor der Bus den Zustand SLEEPING angenommen hat), so muß der zuerst gesendete Frame nicht mehr bestätigt werden - der zweite gesendete Frame nur, falls in ihm das M-Bit gesetzt ist.

Empfängt T<sub>1</sub> einen Frame, in dem das A-Bit gesetzt ist, obwohl die zuletzt von T<sub>1</sub> an T<sub>2</sub> gesendete Nachricht mit M=0 gesendet wurde, so gilt der Frame als korrekt empfangen, und muß ausgewertet werden. Das gesetzte A-Bit ist zu ignorieren.

BEACHTEN: Das Senden eines Frames mit A=1 als Reaktion auf den Empfang eines Frames mit M=0 ist nach obiger Definition verboten!! Die geschilderte Situation kann eintreten, wenn T<sub>2</sub> das M-Bit, oder T<sub>1</sub> das A-Bit falsch empfängt.

Bemerkung: Durch den acknowledged mode verringert sich zwar die Wahrscheinlichkeit von Informationsverlusten, allerdings werden dadurch auch Nachrichtenverdopplungen möglich. Dies kommt vor, wenn das Acknowledge nicht korrekt empfangen wird. Man sollte also Nachrichten, bei denen doppeltes Senden unerwünscht ist im unacknowledged mode senden.

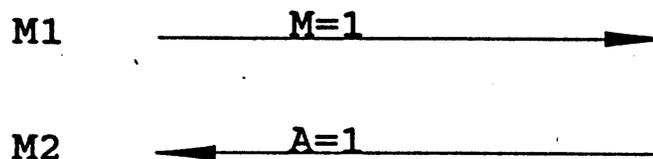
### 4.3. Beispiele

Die gestrichelte Linie bedeutet, daß der Bus in den Zustand SLEEPING übergegangen ist. Eine Linie, die kurz nach der Mitte aufhört, soll darstellen, daß der eine Teilnehmer den Frame als ausgesendet betrachtet, der andere diesen Frame aber nicht oder fehlerhaft empfangen hat. Man beachte, daß nach einem SLEEPING je nach Priorität der eine oder der andere Teilnehmer zu senden beginnt.

Nach der Übertragung des jeweils letzten Frames ist keine weitere Bestätigung nötig. L und R stehen für linker und rechter Teilnehmer.

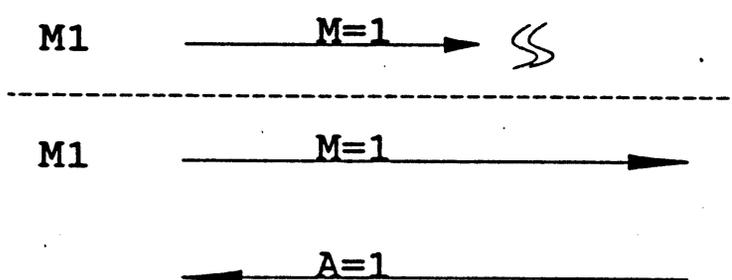
#### 1. Beispiel

Der Teilnehmer L schickt die Meldung  $M_1$  mit Acknowledge-Anforderung. Der Teilnehmer R bestätigt mit einem Acknowledge-Frame.



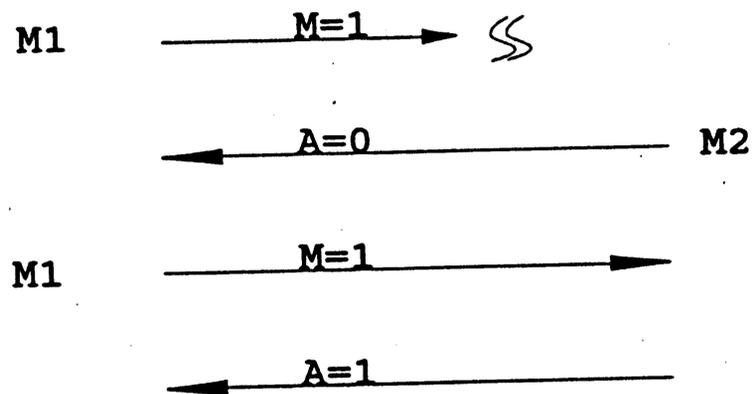
#### 2. Beispiel

L schickt die Meldung  $M_1$  mit Acknowledge-Anforderung. R bestätigt nicht, so daß L die Meldung wiederholt. Diese Wiederholung wird von R bestätigt.



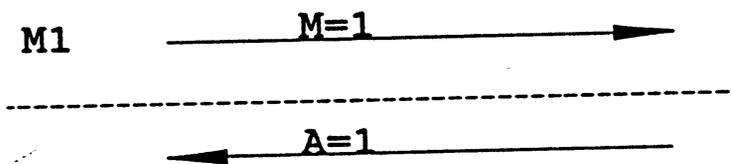
## 3. Beispiel

L sendet  $M_1$  mit Acknowledge-Anforderung. R schickt  $M_2$  mit  $A=0$  zurück. L wiederholt daraufhin die Meldung  $M_1$ . Diese Wiederholung wird von R bestätigt.



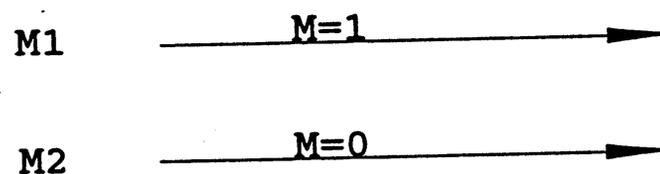
## 4. Beispiel

L sendet  $M_1$  mit Acknowledge-Anforderung. R ist aus irgendwelchen Gründen nicht in der Lage die Bestätigung sofort zu schicken. Da R eine höhere Priorität als L hat, sendet er das Acknowledge, bevor L eine Möglichkeit hat, die Nachricht zu wiederholen.



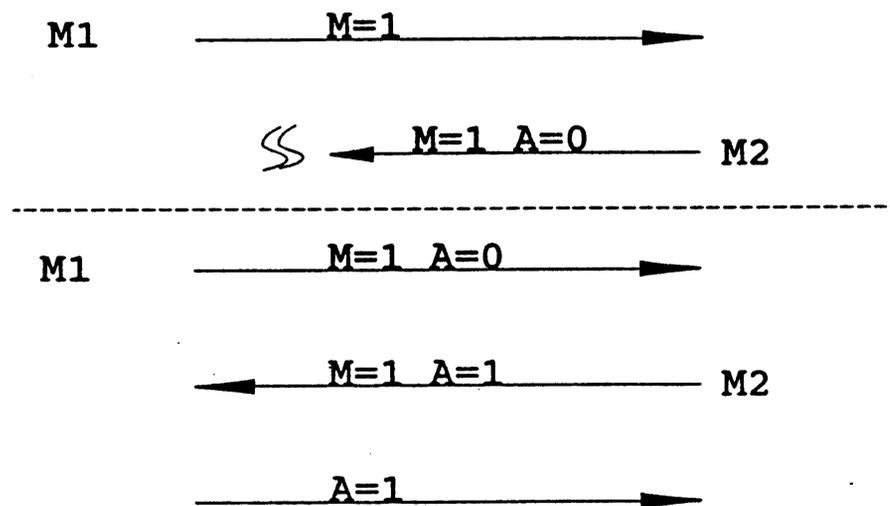
## 5. Beispiel

L sendet  $M_1$  mit Acknowledge-Anforderung. Bevor R bestätigt, sendet L eine weitere Nachricht ohne Acknowledge-Anforderung. R braucht dann nichts mehr zu bestätigen.



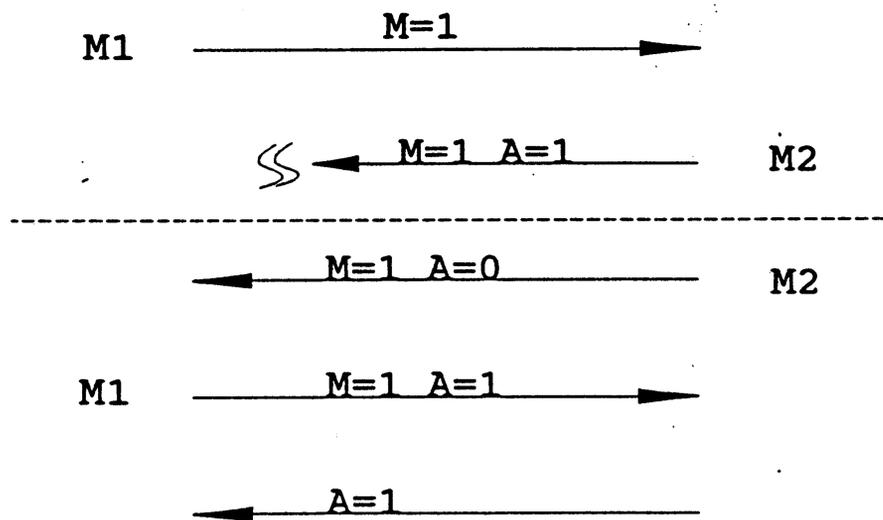
## 6. Beispiel

L sendet  $M_1$  mit Acknowledge-Anforderung. R bestätigt, indem er das A-Bit in der Antwort  $M_2$  setzt, außerdem fordert R selbst ein Acknowledge an. Dieser Frame geht verloren. Jetzt warten beide Teilnehmer auf eine Bestätigung des anderen. In diesem Beispiel habe L die höhere Priorität, wiederholt also  $M_1$  mit  $A=0$ . Da  $M_2$  damit unbestätigt ist, wird diese Nachricht von R erneut gesendet. Dieser Frame wird dann von L mit einem Acknowledge-Frame bestätigt. Beachte: R erhält  $M_1$  zweimal.



## 7. Beispiel

L sendet  $M_1$  mit Acknowledge-Anforderung. R bestätigt, indem er das A-Bit in der Antwort  $M_2$  setzt; außerdem fordert R selbst ein Acknowledge an. Dieser Frame geht verloren. Jetzt warten beide Teilnehmer auf eine Bestätigung des anderen. Im Gegensatz zum 6. Beispiel habe jetzt R die höhere Priorität, wiederholt also  $M_2$  mit  $A=0$ . Da  $M_1$  damit unbestätigt ist, wird diese Nachricht von L erneut gesendet. Dieser Frame wird dann von R mit einem Acknowledge-Frame bestätigt.



#### 4.4. Der (7,4)-Hamming-Code

4 Bit Information (die Sender- und Empfängeradressen) werden mit 3 Bit Redundanz versehen und somit auf 7 Bit erweitert. Dadurch ist es möglich, 1-Bitfehler in den Adressen zu korrigieren. Das höchstwertige Bit ist nicht geschützt und muß ignoriert werden; gesendet wird stets 0.

##### 4.4.1. Codierung

Codiert wird nach folgender Tabelle:

| Adresse | SA bzw. EA                |
|---------|---------------------------|
| 0       | 0 = 000000 <sub>2</sub>   |
| 1       | 81 = 101000 <sub>2</sub>  |
| 2       | 114 = 111001 <sub>2</sub> |
| 3       | 35 = 010001 <sub>2</sub>  |
| 4       | 52 = 011010 <sub>2</sub>  |
| 5       | 101 = 110010 <sub>2</sub> |
| 6       | 70 = 100011 <sub>2</sub>  |
| 7       | 23 = 001011 <sub>2</sub>  |
| 8       | 104 = 110100 <sub>2</sub> |
| 9       | 57 = 011100 <sub>2</sub>  |
| 10      | 26 = 001101 <sub>2</sub>  |
| 11      | 75 = 100101 <sub>2</sub>  |
| 12      | 92 = 101110 <sub>2</sub>  |
| 13      | 13 = 000110 <sub>2</sub>  |
| 14      | 46 = 010111 <sub>2</sub>  |
| 15      | 127 = 111111 <sub>2</sub> |

##### 4.4.2. Decodierung

###### 1. Möglichkeit:

Berechnen des Syndroms  $s$  mit anschließender Auswertung.

$x$ : empfangenes Byte

$$s = 4 * p(x \text{ AND } 75) + 2 * p(x \text{ AND } 46) + 1 * p(x \text{ AND } 23).$$

Dabei ist  $p$  die gerade Paritätsfunktion, d.h. die Exklusiv-Oder-Verknüpfung der einzelnen Bits.

In folgender Tabelle kann man dann den Korrekturwert ablesen:

| Syndrom | Korrekturwert            |
|---------|--------------------------|
| 0       | 0 = 000000 <sub>2</sub>  |
| 1       | 16 = 001000 <sub>2</sub> |
| 2       | 32 = 010000 <sub>2</sub> |
| 3       | 4 = 000010 <sub>2</sub>  |
| 4       | 64 = 100000 <sub>2</sub> |
| 5       | 1 = 000001 <sub>2</sub>  |
| 6       | 8 = 000100 <sub>2</sub>  |
| 7       | 2 = 000010 <sub>2</sub>  |

Die gewünschte Adresse errechnet sich dann durch

$$(x \text{ EOR Korrekturwert}) \text{ AND } 15.$$

2. Möglichkeit:

Suchen der kodierten Adresse aus der Tabelle von Kapitel 4.4.1, die sich um höchstens ein Bit von dem empfangenen Byte unterscheidet.

x: empfangenes Byte  
y: Byte aus der Tabelle von Kap. 4.4.1

Wenn  $(x \text{ EOR } y) \& ((x \text{ EOR } y) - 1) = 0 \Rightarrow$  richtiges Originalbyte aus der Tabelle ermittelt

**4.4.3. Beispiel**

Der Teilnehmer mit der Adresse 5 möchte dem Teilnehmer mit der Adresse 13 über SAPI 1 den String "HALLO" im unacknowledged mode senden. Dann sieht die Layer1-Nachricht wie folgt aus:

| SA                | EA               | LAE              | AF               | L3-Msg.             | CS               |
|-------------------|------------------|------------------|------------------|---------------------|------------------|
| 101 <sub>10</sub> | 13 <sub>10</sub> | 10 <sub>10</sub> | 01 <sub>10</sub> | 'H' 'A' 'L' 'L' 'O' | 37 <sub>10</sub> |

Dabei ist:

101 kodierte Adresse 5  
13 kodierte Adresse 13  
10 die Anzahl der Zeichen im Frame  
1 SAPI  
37 Exklusiv-Oder-Verknüpfung der Bytes 101,...,'O'.

Aufgrund eines Übertragungsfehlers wird die Sendeadresse um ein Bit verfälscht, so daß anstatt 101 = 1100101<sub>2</sub> (Adresse 5) 1100100<sub>2</sub> = 100 empfangen wird.

**Ermitteln der dekodierten Adresse:**

nach 1. Möglichkeit:

Berechnen des Syndroms s

$$\begin{aligned} s &= 4 \cdot p(1000000_2) + 2 \cdot p(0100100_2) + 1 \cdot p(0000100_2) \\ s &= 4 \cdot 1 \quad \quad \quad + 2 \cdot 0 \quad \quad \quad + 1 \cdot 1 \\ &= 5_{10} = 101_2 \end{aligned}$$

Aus obiger Tabelle liest man für das Syndrom 5 den Korrekturwert 1 ab und damit erhält man das Originalbyte

$$(1100100_2 \text{ EOR } 0000001_2) \text{ AND } 1111_2 = 101_2 = 5$$

nach 2. Möglichkeit:

$$(x \text{ EOR } y) \& ((x \text{ EOR } y) - 1) = 0 \Rightarrow \text{richtiges Originalbyte aus der Tabelle ermittelt}$$

Für die Adresse 5 = 1100101<sub>2</sub> ergibt sich

$$(1100100_2 \text{ EOR } 1100101_2) \& ((1100100_2 \text{ EOR } 1100101_2) - 1) = 0$$

## 4.5. Fehlerbehandlung

Wird ein Layer2-Fehler erkannt, so wird kein BREAK gesendet (!!!), sondern die Nachricht einfach ignoriert (verworfen).

Die Fehlerkorrektur sollte nach folgendem Schema ablaufen:

- Decodierung der empfangenen Sende- bzw. Empfangsadresse mit anschließender Auswertung (Empfangsadresse = eigene Sendeadresse ?)
- Vergleich der errechneten Checksumme (falls bei der Decodierung der Adressen ein Bitfehler erkannt wurde, so muß mit den berechtigten hammingcodierten Adressen die Checksummenberechnung durchgeführt werden) mit der empfangenen Checksumme.
- Falls sich die beiden Checksummen unterscheiden  $\Rightarrow$  Verwerfen der Nachricht

## 5. Rundruf der Mobile

Die Adresse 15 ist eine Dummyadresse und wird für Rundrufe verwendet, d.h. jeder Teilnehmer muß Nachrichten an die Adresse 15 auswerten.

Einen Rundruf darf jedoch nur die Mobile veranlassen (Master  $\Leftrightarrow$  Slave). Durch das Senden eines Rundrufes werden die angeschlossenen Teilnehmer aufgefordert, ihre Kennungsdaten an die Mobile zu schicken.

Die Mobile wird durch den Erhalt der Kennungsdaten über das Vorhandensein eines BF-Busteilnehmers informiert.

### Zuordnung von Teilnehmergeräten in der Mobile

Alle Geräte müssen sich beim Systemstart beim Funkgerät anmelden (Layer3-Master-Funktion des Funkgerätes).

Wird das Funkgerät eingeschaltet, so sendet es einen Rundruf aus. Ein Rundruf besitzt folgendes Aussehen:

|                  |  |
|------------------|--|
| Sendeadresse:    | 00 <sub>H</sub> (Mobile)   |
| Empfangsadresse: | 0F <sub>H</sub> (Rundruf)  |
| Länge:           | 06 <sub>H</sub>  |
| AF               | 0F <sub>H</sub> (SAPI: 0F <sub>H</sub> , Modebit: 0 (kein Acknowledge-Mode)) |
| Layer3-Msg:      | 01 <sub>H</sub>  |

Die angeschlossenen Geräte müssen auf den Rundruf hin ihre eigene Geräteerkennung senden. Ein angeschlossener Teilnehmer darf, wenn er noch keinen Rundruf empfangen hat und sich damit bei der Mobile noch nicht eingetragen hat, niemals von sich aus mit dem Senden beginnen (Master  $\Leftrightarrow$  Slave).

Für Teilnehmergeräte die, während das Funkgerät eingeschaltet ist, am BF-Bus kontaktiert bzw. wieder entfernt werden, muß von Fall zu Fall eine Möglichkeit gefunden werden (z.B. Hardwareerkennung), damit das Teilnehmergerät von der Mobile einen Rundruf erhält, bzw. damit die Mobile das Entfernen des Zusatzgerätes mitgeteilt bekommt.

Die Geräteerkennung besitzt folgendes Aussehen:

|                  |  |
|------------------|--|
| Sendeadresse:    | XX <sub>H</sub> (Adresse, des Teilnehmergerätes)   |
| Empfangsadresse: | 00 <sub>H</sub> (Mobile)   |
| Länge:           | Länge der L3-Msg. + 5Byte Rahmeninfo von Layer2  |
| SAPI:            | 0F <sub>H</sub>  |
| Modebit:         | 0 oder 1   |
| Layer3-Msg.      | 1.Byte:   Gerätenummer<br>2.Byte:   Geräteklasse<br>3.Byte:   Hard-/Softwarestand<br>weitere 7 optionale Bytes für weitere Informationen |

Durch den Empfang der Geräteerkennung wird die Mobile über das Vorhandensein eines bestimmten Teilnehmergerätes unter der angegebenen Adresse informiert.

## 6. Weitere Daten

Baudrate: 19200 +/- 1%

minimaler Gesamtlastwiderstand am BF-Bus:  $R_L \geq 25\text{K}\Omega$

maximale Gesamtlastkapazität am BF-Bus:  $C_L \leq 1,6\text{nF}$

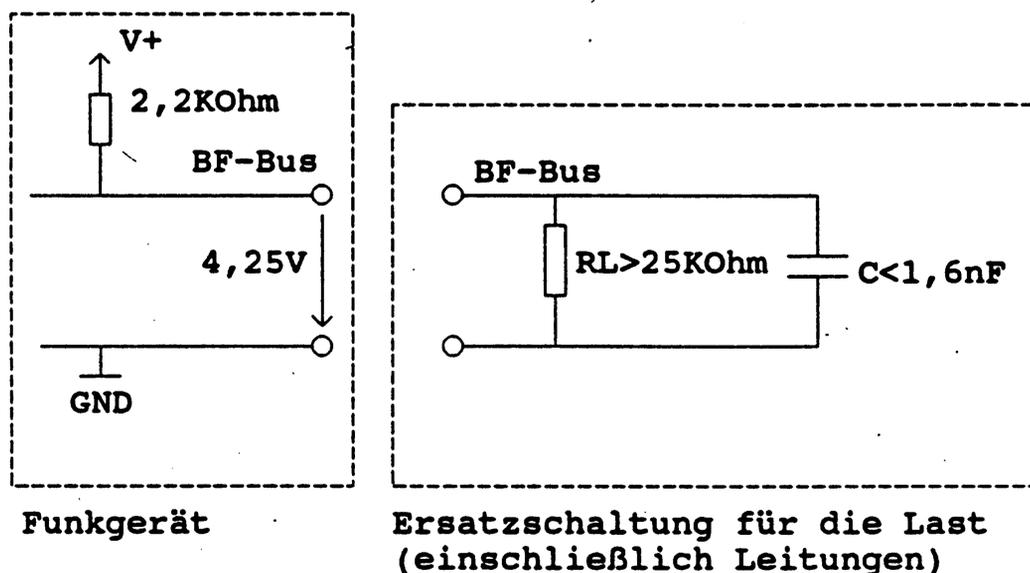


Bild 6.1 Ersatzschaltung Funkgerät  $\leftrightarrow$  Teilnehmer

**Pull-Down-Widerstand:** Falls Teilnehmergeräte an den BF-Bus angeschlossen werden, deren Verbindung zu der Mobile nicht dauerhaft existiert, so sollte in dem Teilnehmergerät ein Pull-Down-Widerstand vorhanden sein, um dessen BF-Bus-Eingang auf einen definierten Pegel zu ziehen.

$$R_{\text{Pull Down}} \geq 100\text{K}\Omega$$

Der Gesamtlastwiderstand aller Teilnehmergeräte muß jedoch größer als  $25\text{k}\Omega$  sein (siehe oben).

### Elektr. Pegel:

#### low-Pegel:

Ein Eingangspegel von  $< 0,8\text{V}$  an dem BF-Bus-Eingang eines Teilnehmergerätes muß als low erkannt werden.

Zieht ein BF-Busteilnehmer den Bus auf low, so muß der BF-Bus-Eingang an diesem Teilnehmergerät  $< 0,6\text{V}$  sein. Das Gesamtsystem der BF-Busteilnehmer muß so ausgelegt sein, daß durch Spannungsabfällen an Leitungen (beachte insbesondere eine mögliche GND-Verschiebung durch Versorgung eines Teilnehmergerätes durch ein anderes) die obigen Grenzwerte nicht überschritten werden.

#### high-Pegel:

Ein Eingangspegel von  $> 3,6\text{V}$  an dem BF-Bus-Eingang eines Teilnehmergerätes muß als high erkannt werden.

Das Gesamtsystem der BF-Busteilnehmer muß so ausgelegt sein, daß obiger Grenzwert bei keinem BF-Bus-Eingang eines Teilnehmers unterschritten wird.

Jeder Teilnehmer muß einen Pull-Up von  $2,2\text{k}\Omega$  an  $5\text{V}$  treiben können.

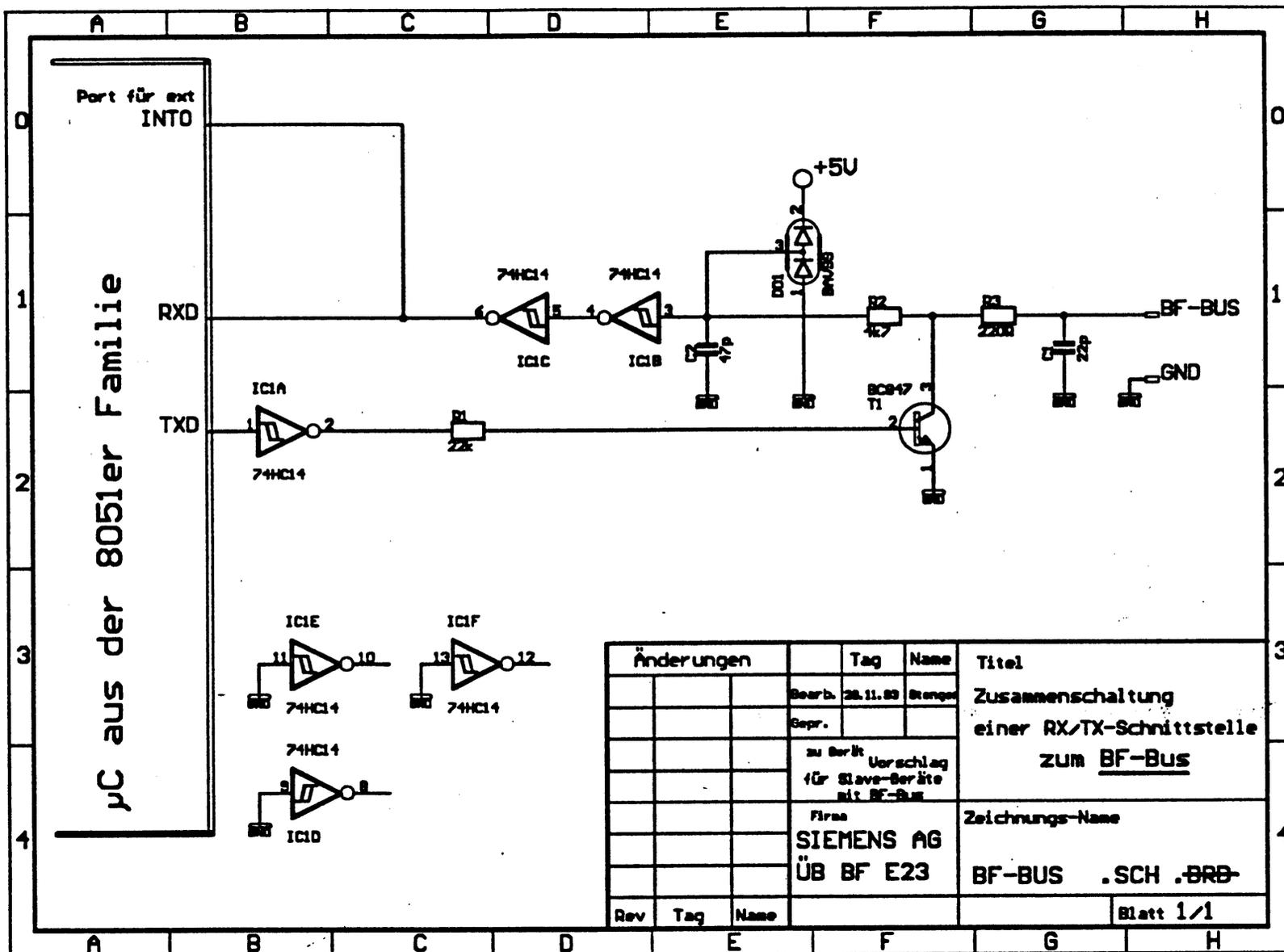
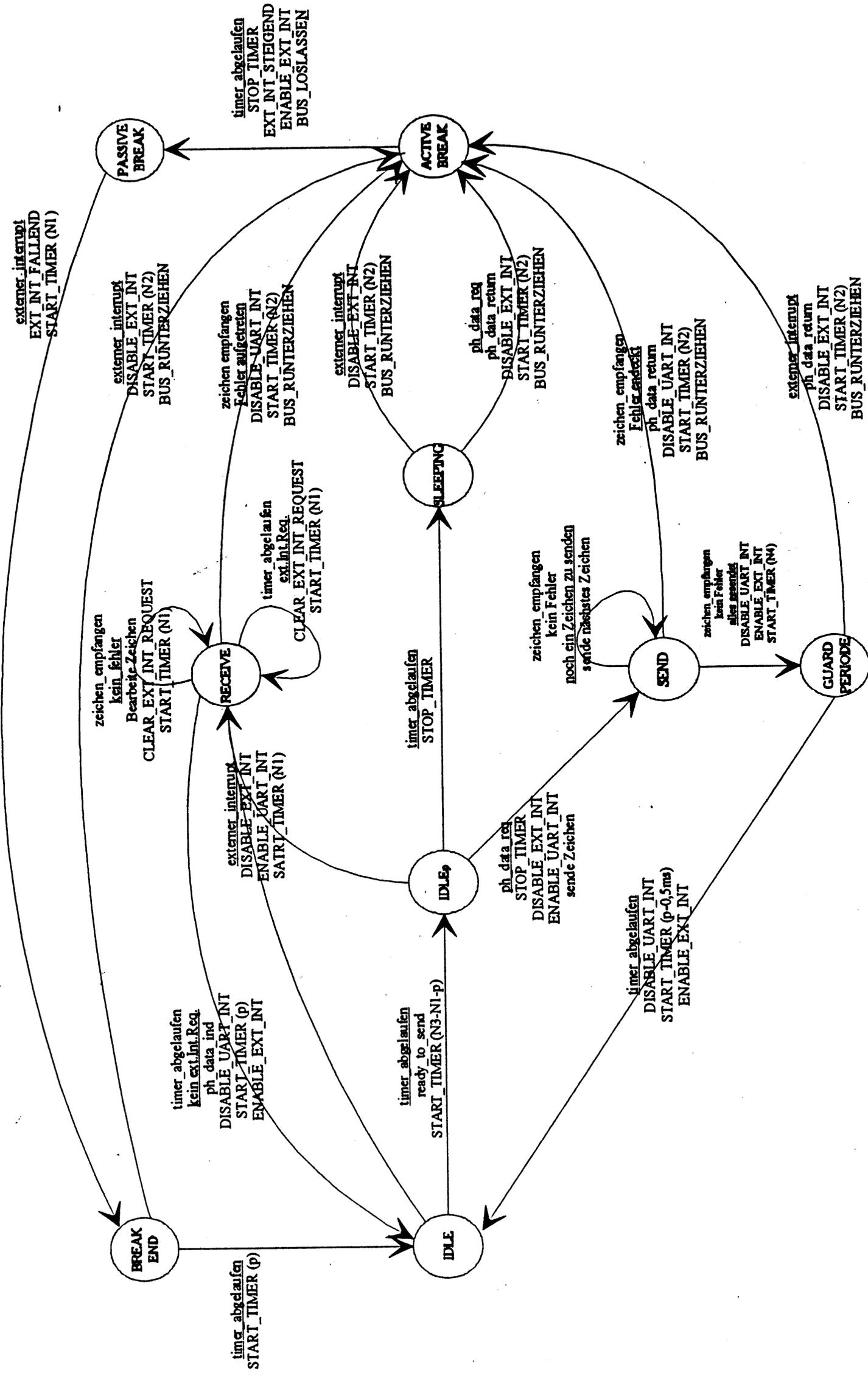


Bild 6.2 Schaltungsvorschlag: Zusammenschaltung einer RX/TX-Schnittstelle zum BF-Bus

# Zustandsdiagramm von Layer1 im Detail



## Erläuterungen zum Zustandsdiagramm von Layer1

|                       |  |
|-----------------------|--|
| BUS_LOSLASSEN         | Ausgang auf logisch 1 setzen   |
| BUS_RUNTERZIEHEN      | Ausgang auf logisch 0 setzen   |
| CLEAR_EXT_INT_REQUEST | Requestbit löschen   |
| DISABLE_EXT_INT       | Interrupt sperren, Requestbit löschen  |
| ENABLE_EXT_INT        | Requestbit löschen, Interrupt freigeben  |
| EXT_INT_FALLEND       | Interrupt bei fallender Flanke   |
| EXT_INT_STEIGEND1     | Interrupt bei steigender Flanke  |
| DISABLE_UART_INT      | Interrupt sperren, Requestbit löschen.   |
| ENABLE_UART_INT       | Requestbit löschen, Interrupt freigeben  |
| START_TIMER (x)       | Timer stoppen, mit neuem Wert beschreiben<br>Interrupt freigeben, Requestbit löschen,<br>Timer starten |
| STOP_TIMER            | Timer stoppen, Requestbit löschen.   |
| ph_data_return        | Melde Layer2, daß Nachricht nicht gesendet wurde   |
| ready_to_send         | Melde Layer2, daß gesebdet werden darf   |
| ph_data_ind           | Übergebe empfangenem Frame an Layer2   |
| ph_data_req           | Layer2 übergibt zu sendenden Frame   |

Bit 5 = 1 = 7 Service

Intel SW 00

MH SW 01

A20 SW 02

Meldungen vom Hörer zum Funkgerät:

|     |   |     |     |     |     |     |     |     |     |
|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|
| 0E  | 2 Byte Tastenmeldung:<br>Byte 1 : 01h                    Tastenmeldungsart<br>Byte 2 : Taste                    Tastencode siehe Tabelle  |     |     |     |     |     |     |     |     |
| 0F  | Hörer-Kennung 4 Byte:<br>1 Byte Gerätenummer (Byte 1 )<br>01: Ersthörer EP210<br>02: Zweithörer EP210<br>03: Ersthörer EP200<br>04: Zweithörer EP200<br><br>1 Byte Reserve (= 0) (Byte 2 )<br><br>1 Byte Soft- und Hardwarestand( Byte 3 ):<br>Bit 0            Reserve ( =0)<br>Bit 1            Hardwareversion<br>Bit 2            Hardwareversion<br>Bit 3            Hardwareversion<br>Bit 4            Softwareversion<br>Bit 5            Softwareversion<br>Bit 6            Softwareversion<br>Bit 7            Reserve ( =0)<br><br>Bit: 7      6      5      4      3      2      1      0<br><br><table border="1" style="margin-left: auto; margin-right: auto;"><tr><td>Res</td><td>SW3</td><td>SW2</td><td>SW1</td><td>HW3</td><td>HW2</td><td>HW1</td><td>Res</td></tr></table><br>Beginnend mit: sw1,sw2,sw3 = 0<br>hw1,hw2,hw3 = 0<br><br>1 Byte Aktuelle Priorität und Gerätespezifisches (Byte 4)<br>Bit 0-3        Aktuelle Priorität<br>Bit 4        = 0 Hörer aufgelegt<br>= 1 Hörer abgehoben<br>Bit 5-7        Hörertyp (Normalhörer = 0) | Res | SW3 | SW2 | SW1 | HW3 | HW2 | HW1 | Res |
| Res | SW3   | SW2 | SW1 | HW3 | HW2 | HW1 | Res |     |     |

*Bit 5 = Service aufbauend auf Maske 72006-A16*

\*

Eine Acknowledgeanforderung erfolgt beim Senden der Kennung, beim Wechsel des Hookswitch, und bei Tastenmeldungen.  
Ergänzung zu Layer 2 zur Teilnehmerabhängigen Reaktion, falls die Nachricht 3 mal unbestätigt bleibt: Die Begrenzung auf 3 mal besteht für den Hörer nicht(d.h. unbegrenzt).

**Verteiler:**

Herrn Hofmann Jürgen  
Herrn Becker-Bamberger

**Übersicht über Software- und Hardwarestand P1 Hörer**

Eine Identifikation des Bedienhörers (Adresse OA -Ersthörer, Adresse OB-Zweith.) ist über die BFbus-Schnittstelle mittels Gerätenummer, Hardwarestand und Softwarestand möglich.

| Siemens Sachnummer | Stückzahl/Datum    | SW-/HW Stand | Funktion     | Bemerkungkkkkk  |
|--------------------|--------------------|--------------|--------------|---|
| 1) -Z2006-A16-01   | 3.000 ab 5/91      | 00 / 00      | Normalhörer  | nur 3000 Stück<br>Gummitasten/SMD<br>kleine SW-Fehler<br>durch P1 komp.                           |
| 2) -Z2006-A16-01   | 22.000 ab 11/91    | 00 / 01      | Normalhörer  | ab 3000 - 25000<br>Kunststofftasten<br>Goldkontakte<br>LP B2-04 Wahl-<br>start1 = Wahl-<br>start2 |
| 3) -Z2006-A16-01   | 25 ?? ab 5.91      | 00 / 00,01   | Servicehörer | siehe oben<br>plus Servicebit   |
| 4) -Z2006-A19-01   | ab 25000           | 01 / 01      | Normalhörer  | Fehler beseitigt<br>hardw. wie 2)   |
| 5) -Z2006-A20-01   | 1000 Stück ab 2/92 | 02 / 01      | Servicehörer | Mit Servicealgo-<br>rithmus, kein<br>Servicebit, Er-<br>kennung HW-Std                            |

Anmerkung ab Aufbrauch der Masken A16 soll nicht mehr die LP B2-04 (Wahlstart 1 gleich Wahlstart 2) sondern wieder LP B2-03 mit getrennten Wahlstarttasten eingebaut werden. Hardwarestand auf 02 ändern (R19, R23 = 0 Ohm, R22 unbestückt)

**Servicehörer:** Die Maske A20 hat einen speziellen Ablauf für Serviceaktivierung, und kann mit dem Softwarestand 02 Gerätenummer Olerkannt werden. Hiervon gibt es 1000 Stück für Service- und Tracerzwecke. Für speziellere Servicemodes (Labormodes usw) wird empfohlen, abhängig von dieser Servicesoftware noch einen Code über die Tastatur abzufragen.

MFG Hettrich

## BF BUS LAYER 3 FÜR EP210

Festgelegte Höreradressen für Normalhörer    0A  
für Zweithörer                                    0B

| SAPI | <L3MU>Bedeutung für Gerät Nr. 01 und 02  |                 |  |    |    |    |    |   |   |   |   |   |    |    |    |    |    |    |    |
|------|--|-----------------|--|----|----|----|----|---|---|---|---|---|----|----|----|----|----|----|----|
|      | Befehls-<br>byte   | Timer/Priorität | Anmerkung  |    |    |    |    |   |   |   |   |   |    |    |    |    |    |    |    |
| 1    | 1  | --              | Software-Reset des Hörers<br>(Sprung nach 0, Verhalten wie<br>nach Power on Reset)   |    |    |    |    |   |   |   |   |   |    |    |    |    |    |    |    |
| 1    | 2  | xx              | Prioritäts-Zeit ändern<br>(0<Priorität<16); Adresse 0Fh<br>für Rundruf, kein Gerät   |    |    |    |    |   |   |   |   |   |    |    |    |    |    |    |    |
| 1    | 3  | xx              | kein Empfang, kein Zugriff auf<br>BF-Bus, keine<br>Tastaturauswertung<br>bis fuer xx * 100ms BF-Bus<br>auf High war. Bei xx = 0xFF<br>warten bis Hardware-Reset. |    |    |    |    |   |   |   |   |   |    |    |    |    |    |    |    |
|      |  |                 |  |    |    |    |    |   |   |   |   |   |    |    |    |    |    |    |    |
| 2    | DISPLAY-Daten 16 Byte<br>Byte: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16<br><br>Display/Zeichen: <table border="1" style="display: inline-table; border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td></tr> </table>   |                 |  | 1  | 2  | 3  | 4  | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 1    | 2  | 3               | 4  | 5  | 6  | 7  | 8  |   |   |   |   |   |    |    |    |    |    |    |    |
| 9    | 10   | 11              | 12   | 13 | 14 | 15 | 16 |   |   |   |   |   |    |    |    |    |    |    |    |
|      |  |                 |  |    |    |    |    |   |   |   |   |   |    |    |    |    |    |    |    |
| 6    | Sondersymbole und Displayfunktionen 5 Byte(Standard)<br>2 Byte Symbole statisch an:<br>Byte 1:LSB 0    6.Symbol von rechts<br>1    7.Symbol von rechts<br>2    8.Symbol von rechts<br>3    9.Symbol von rechts<br>4    10.Symbol von rechts<br>5    frei<br>6    frei<br>MSB 7    frei<br>Byte 2:LSB 0    1.Symbol von rechts<br>1    2.Symbol von rechts<br>2    3.Symbol von rechts<br>3    4.Symbol von rechts<br>4    5.Symbol von rechts<br>5    frei<br>6    frei<br>MSB 7    frei |                 |  |    |    |    |    |   |   |   |   |   |    |    |    |    |    |    |    |

2 Byte Symbole blinken:

|            |   |                      |
|------------|---|----------------------|
| Byte 3:LSB | 0 | 6.Symbol von rechts  |
|            | 1 | 7.Symbol von rechts  |
|            | 2 | 8.Symbol von rechts  |
|            | 3 | 9.Symbol von rechts  |
|            | 4 | 10.Symbol von rechts |
|            | 5 | frei                 |
|            | 6 | frei                 |
| MSB        | 7 | frei                 |

|            |   |                     |
|------------|---|---------------------|
| Byte 4:LSB | 0 | 1.Symbol von rechts |
|            | 1 | 2.Symbol von rechts |
|            | 2 | 3.Symbol von rechts |
|            | 3 | 4.Symbol von rechts |
|            | 4 | 5.Symbol von rechts |
|            | 5 | frei                |
|            | 6 | frei                |
| MSB        | 7 | frei                |

1 Byte Displayfunktion:

Byte 5:LSB 0 Display blinken(16 Zeichen  
+ Symbole)

(hierbei Symbole statisch auf 1 !)

1 = 1, Displayreset vor jeder Ausgabe

2-7 Reserviert = 0, da ansonsten der 8051  
nicht in den Power down geht!

Blinkfrequenz: 0.5 Hz

Einzeit: 1sec; Auszeit: 1sec.

Symbole können im Gegenteil zuein-  
ander Blinken, Steuerung mit  
statisch ein & blinken ein, oder  
nur Blinken ein.

7

Licht und NF-Steuerung 1 Byte

|      |   |                              |             |
|------|---|------------------------------|-------------|
| (LSB | 0 | Licht                        | (Ein = 1)   |
|      | 1 | NF-Hörkapsel                 | (Ein = 1)   |
|      | 2 | Mikrofon                     | (Ein = 1)   |
|      | 3 | Mikrofon                     | (Hörer = 0) |
|      |   | Freispr./Hörer               |             |
|      | 4 | Laut_1 Schallgeber           |             |
|      | 5 | Laut_2 Schallgeber           |             |
|      | 6 | Ausgabe_1 auf P1.7 (Reserve) |             |
| MSB  | 7 | Ausgabe_2 auf P3.7 (Reserve) |             |

## Timer-Steuerung 6 Byte

Byte 1 Timer L

Byte 2 Timer H (Bit 0...3)

Ansteuerverhältnis (Bit 4...7)

$$\text{(Tonfrequenz=Quarzfrequenz/12*(1/(0FFF-Timerwert))*}$$

$$\text{(Ansteuerverhältnis.+1))}$$

Quarzfrequenz = 7.37 MHz

Byte 3 Ein-Zeit (10 ms Raster)

Byte 4 Aus-Zeit (10 ms-Raster)

Byte 5 N Folgen

Byte 6 LSB 0 Ton ein

1 Tonfolge ( siehe I )

2 3-fach Ton 950/1400/1800 Hz

gleichzeitig

3 Modulation z.b. für Sapo-Ton  
(16Hz fest)

4 Tastenklick

5 3-fach Ton 950/1400/1800 Hz

jeder Ton 300ms an, dann 1s aus

6 frei

MSB 7 frei

Anmerkung: Quarzfrequenz bei EP210: 7.37 Mhz

Beispiele für :

Tonfrequenz Ansteuerverhältnis Timerwert

425 Hz 1:4 FEDFh

1.0 khz 1:3 FF67h

2.5 khz 1:2 FFAEh

3-fach Ton --- FFC8h

Tonfolge:

z.B. 1khz 200ms ein, 300ms aus, 12 mal

Ton ein = 1, Tonfolge = 1, N = 12, Ein-Zeit = 20,

Auszeit = 30, d.h. Gesamttondauer: 6s

Hier wäre eine maximale Gesamttondauer von 127,5s  
möglich.

Dauerton:

Ton ein = 1, Tonfolge = 0, d.h. Dauerton mit geladener  
Frequenz bis zum Ausschalten.

0E

Service-Anforderung

Länge = 5 Byte.

Funktion nur beim SW-Stand 2 implementiert!!!

0F

1 Byte: 01H

Anforderung der Kennung

(Die Kennung wird auch, gesendet,  
wenn der Hörer unter der Dummy-  
adresse 0F angesprochen wird;  
Siehe Layer1/Layer2 Seite 12.)

00h, 02h-ffh

Reserviert, d.h. es darf keine  
Kennung geschickt werden.

Meldungen vom Hörer zum Funkgerät:

0E

1 Byte = 01h => Tastenmeldung:

Byte 2 : Taste                    Tastencode siehe Tabelle

2. Byte = 02h => Servic-Bestätigung:

Länge = 2 Byte.

Funktion nur beim SW-Stand 2 implementiert!!!

Bsp.: Bei Anforderung mit 00h,00h,00h,00h,00h ==> 00h,18h

0F

Hörer-Kennung 4 Byte:

1 Byte Gerätenummer (Byte 1 )

01: Ersthörer EP210

02: Zweithörer EP210

03: Ersthörer EP200

04: Zweithörer EP200

1 Byte Reserve (= 0) (Byte 2 )

1 Byte Soft- und Hardwarestand( Byte 3):

Bit 0            Reserve ( =0)

Bit 1            Hardwareversion

Bit 2            Hardwareversion

Bit 3            Hardwareversion

Bit 4            Softwareversion

Bit 5            Softwareversion

Bit 6            Softwareversion

Bit 7            Reserve ( =0)

|      |   |     |   |     |   |     |   |     |   |     |   |     |   |     |   |     |   |   |
|------|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|---|
| Bit: | 7 | 6   | 5 | 4   | 3 | 2   | 1 | 0   |   |     |   |     |   |     |   |     |   |   |
|      | + | +   | + | +   | + | +   | + | +   |   |     |   |     |   |     |   |     |   |   |
|      |   | Res |   | SW3 |   | SW2 |   | SW1 |   | HW3 |   | HW2 |   | HW1 |   | Res |   |   |
|      | + | +   | + | +   | + | +   | + | +   | + | +   | + | +   | + | +   | + | +   | + | + |

Beginnend mit: sw1,sw2,sw3 = 0

hw1,hw2,hw3 = 0

1 Byte Aktuelle Priorität und Gerätespezifisches (Byte 4)

Bit 0-3        Aktuelle Priorität

Bit 4        = 0 Hörer aufgelegt

             = 1 Hörer abgehoben

Bit 5-7        Hörertyp (Normalhörer = 0)

*BIT 5: A = 1 Service*

Ergänzung zu Layer 2 zur Teilnehmerabhängigen Reaktion, falls die Nachricht 3 mal unbestätigt bleibt: Die Begrenzung auf 3 mal besteht für den Hörer nicht (d.h. unbegrenzt).

Anmerkungen:

Sind die oben beschriebenen Meldungen länger, so sind die zusätzlichen Bytes zu ignorieren. Andere, als die hier aufgeführten SAPI sind zu ignorieren. Bei Ein-/Ausschaltfunktionen bedeutet, wenn nicht anders festgelegt 1 = ein.

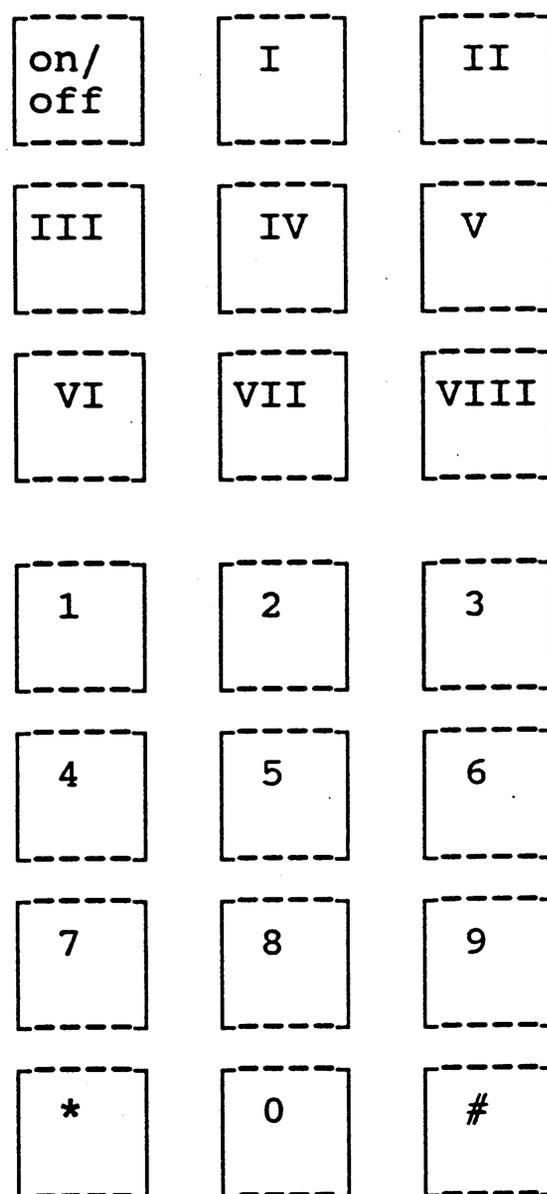
Der Hörer wertet alle an ihn adressierten Meldungen ohne Acknowledge unabhängig vom Absender aus. Meldungen mit Acknowledge-Anforderung werden nur ausgewertet, wenn sie vom Gerät mit der Adresse 00 gesendet werden.

Tabelle Tastaturcodes:

| Tastensymbol | Taste drücken | Taste loslassen | Anmerkung |
|--------------|---------------|-----------------|-----------|
| 0            | 30 H          | B0 H            |           |
| 1            | 31 H          | B1 H            |           |
| 2            | 32 H          | B2 H            |           |
| 3            | 33 H          | B3 H            |           |
| 4            | 34 H          | B4 H            |           |
| 5            | 35 H          | B5 H            |           |
| 6            | 36 H          | B6 H            |           |
| 7            | 37 H          | B7 H            |           |
| 8            | 38 H          | B8 H            |           |
| 9            | 39 H          | B9 H            |           |
| *            | 2A H          | AA H            |           |
| #            | 23 H          | A3 H            |           |

| Tastensymbol        | Taste drücken | Taste loslassen | Anmerkung  |
|---------------------|---------------|-----------------|--|
| Funktionstaste I    | 3F H          | BF H            |  |
| Funktionstaste II   | 3E H          | BE H            |  |
| Funktionstaste III  | 4B H          | CB H            | (Shifttaste, muss als Kombination mit allen anderen Tasten, ohne On/Off möglich sein.) |
| Funktionstaste IV   | 3D H          | BD H            |  |
| Funktionstaste V    | 21 H          | A1 H            |  |
| Funktionstaste VI   | 3A H          | BA H            |  |
| Funktionstaste VII  | 3B H          | BB H            |  |
| Funktionstaste VIII | 3C H          | BC H            |  |
| Hook                | 48 H          | C8 H            | (48 H = aufgelegt )  |

## Tastenanordnung EP210:



### Wichtig:

Taste II und V werden bei EP210 als Wahlstarttaste verwendet. Da die Wahlstarttaste aber nicht bei alle OEM-Partnern auf Taste II und V liegt, ist eine Entprellung nur im Funkgerät möglich.

Acknowledge bei SAPI 1 , Befehlsbyte 1 und 3:  
Hier wird kein Acknowledge (M-Bit = 0 ) angefordert; Die Anweisung ist sofort auszuführen.

Verhalten bei Meldungen, die kürzer sind, als im SAPI spezifiziert:  
Diese Meldungen sind zu verwerfen, und dürfen nicht ausgeführt werden.  
Ist das Telegramm formal in Ordnung, d.h. Länge und Checksumme sind richtig, so muß eine Acknowledgeanforderung (M = 1) mit einem Acknowledge ( A = 1) beantwortet werden.