

M20

# OLITERM

*Interaktive und asynchrone  
Datenübertragung*

*Benutzer-Handbuch*



**olivetti**

#### VORWORT

Dieses Handbuch ist für Programmierer des M20 BASIC konzipiert. Es erläutert die OLITERM Funktionen zur Steuerung der seriellen Schnittstelle RS-232-C, wenn der M20 on line geschaltet ist und eine asynchrone Start-Stop Aufzeichnung verwendet wird.

Das Handbuch umfaßt 3 Kapitel:  
Das erste Kapitel informiert allgemein über die OLITERM Funktionen, das zweite liefert eine detaillierte Beschreibung dieser Funktionen, und das dritte zeigt einige nützliche Anwendungsprogramme auf.

Das Handbuch dient der Information, sein Inhalt ist ohne ausdrückliche schriftliche Vereinbarung nicht Vertragsgegenstand. Technische Änderungen behalten wir uns vor. Die angegebenen Daten sind lediglich Nominalwerte.

© Copyright 1983 by Deutsche Olivetti DTS GmbH.

## INHALT

### ALLGEMEINES KONZEPT

EINFÜHRUNG 1-1

SOFTWARE-ANPASSUNG 1-2

HARDWARE-ANPASSUNG 1-2

BENUTZERSCHNITTSTELLE 1-4

### 2. FUNKTIONSAUFRUFE

OLITERM INITIALISIEREN  
FUNKTION 1 2-1

OLITERM ABSCHLIESSEN  
FUNKTION 2 2-4

BYTE LESEN  
FUNKTION 3 2-5

DATEN EMPFANGEN  
FUNKTION 4 2-7

EIN BYTE SCHREIBEN  
FUNKTION 5 2-9

DATEN SENDEN  
FUNKTION 6 2-10

OLITERM STATUS  
FUNKTION 7 2-11

EMPFANGSPUFFER LÖSCHEN  
FUNKTION 8 2-12

EMPFANG DEAKTIVIEREN  
FUNKTION 9 2-12

EMPFANG AKTIVIEREN  
FUNKTION 10 2-14

EMPFANGS-TIMEOUT SETZEN  
FUNKTION 11 2-15

SENDEPUFFER DEAKTIVIEREN  
FUNKTION 12 2-16

SENDEN DEAKTIVIEREN  
FUNKTION 13 2-16

SENDEN AKTIVIEREN  
FUNKTION 14 2-17



UNTERBRECHUNG SENDEN FUNKTION 15	2-18
RÜCKANTWORT SETZEN FUNKTION 16	2-19
RÜCKANTWORTDATEN SENDEN FUNKTION 17	2-20
SENDE-TIMEOUT SETZEN FUNKTION 18	2-21
KONTROLLZEICHEN SETZEN FUNKTION 19	2-22
SCHNITTSTELLE AUSSCHALTEN FUNKTION 20	2-23
SCHNITTSTELLE EINSCHALTEN FUNKTION 21	2-24

### 3. ANWENDUNGSBEISPIELE

<u>EIN PROGRAMM ZUM INITIA- LISIEREN, ZUM SENDEN VON BYTE UND EMPFANGEN VON DATEN</u>	3-1
---	-----

<u>EIN PROGRAMM ZUM SENDEN EINER DATEI ZU EINEM AN- DEREN M20</u>	3-2
---	-----

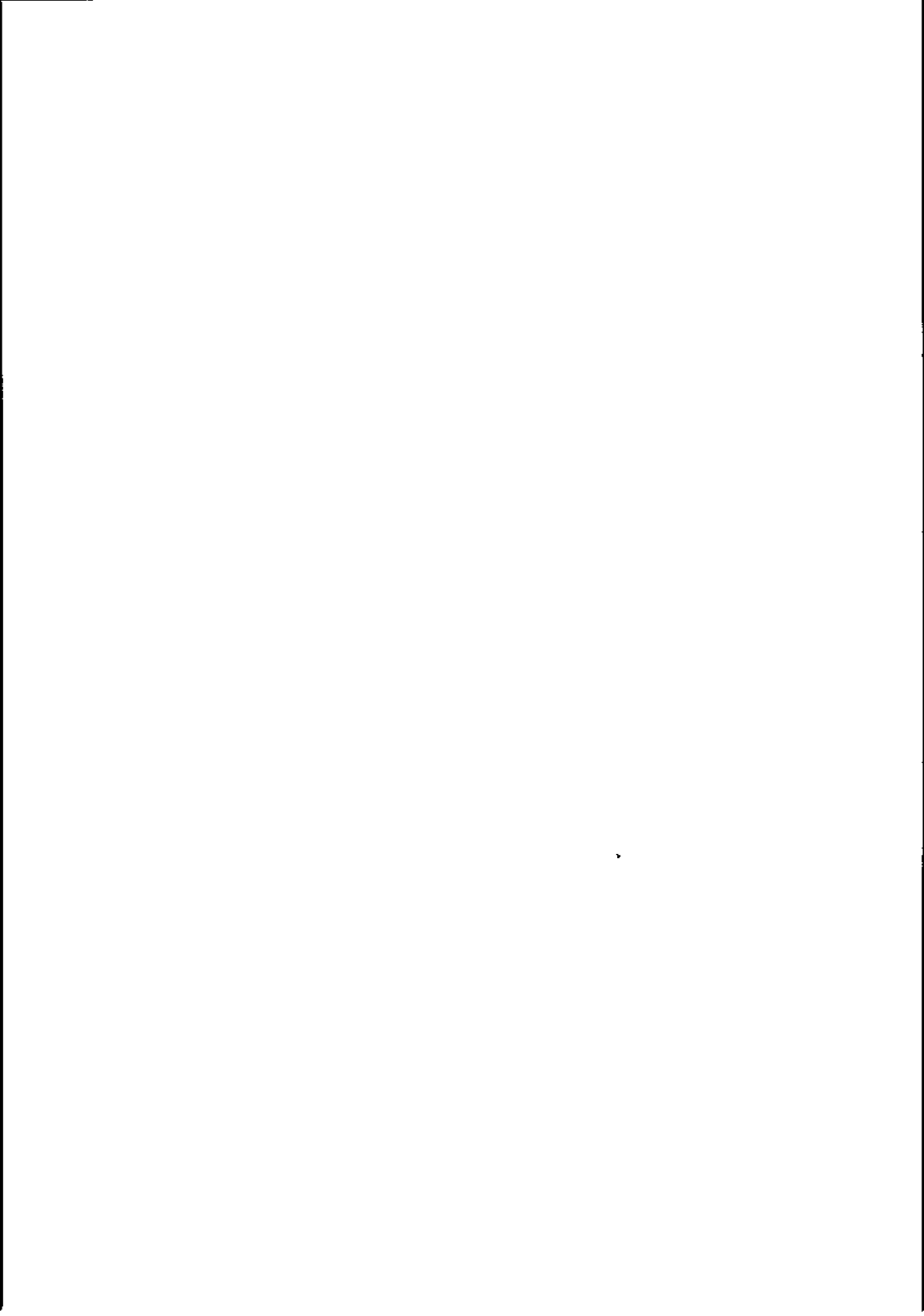
<u>EIN PROGRAMM ZUM EMPFANG EINER DATEI VON EINEM AN- DEREN M20</u>	3-5
---	-----

#### A. FUNKTIONEN UND IHRE STATUSCODES

<u>FEHLERCODES</u>	A-1
--------------------	-----

<u>DATENFEHLER EMPFANGEN</u>	A-3
------------------------------	-----

<u>STATUS-RÜCKMELDUNG</u>	A-4
---------------------------	-----



## **1. ALLGEMEINES KONZEPT**



## Zum Kapitel

Dieses Kapitel ist eine Einführung in die OLITERM Funktionen und die hierfür erforderlichen Geräte.

### INHALT

EINFÜHRUNG	1-1
SOFTWARE-VORAUSSETZUNGEN	1-2
HARDWARE-VORAUSSETZUNGEN	1-2
BENUTZERSCHNITTSTELLE	1-4



## EINFÜHRUNG

Der M20 ist im wesentlichen eine separat arbeitende Maschine. Er kann aber bei Verwendung eines BASIC-Anwendungsprogramms, das die OLITERM Leitungsfunktionen gebraucht, ebenso gut mit einer entfernten Maschine verbunden werden.

Eine derartige Verbindung kann man zu einem Großrechner, zu einem weiteren M20 oder zu einer sonstigen entfernten Einheit herstellen. Die serielle Schnittstelle RS-232-C liefert die für diese Tätigkeit notwendige asynchrone Schnittstelle. Somit kann man den M20 nicht nur für wissenschaftliche und geschäftliche Zwecke anwenden, sondern ihn auch zur Aufnahme und Übermittlung von Daten zu und von entfernten Einheiten einsetzen.

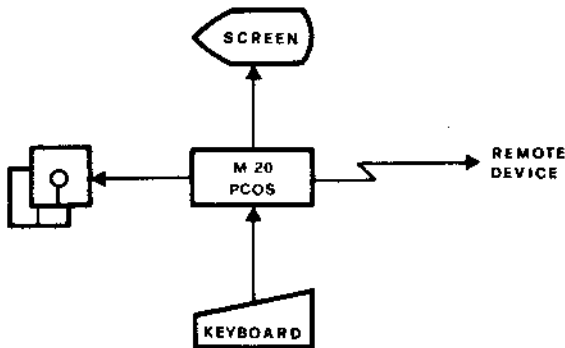
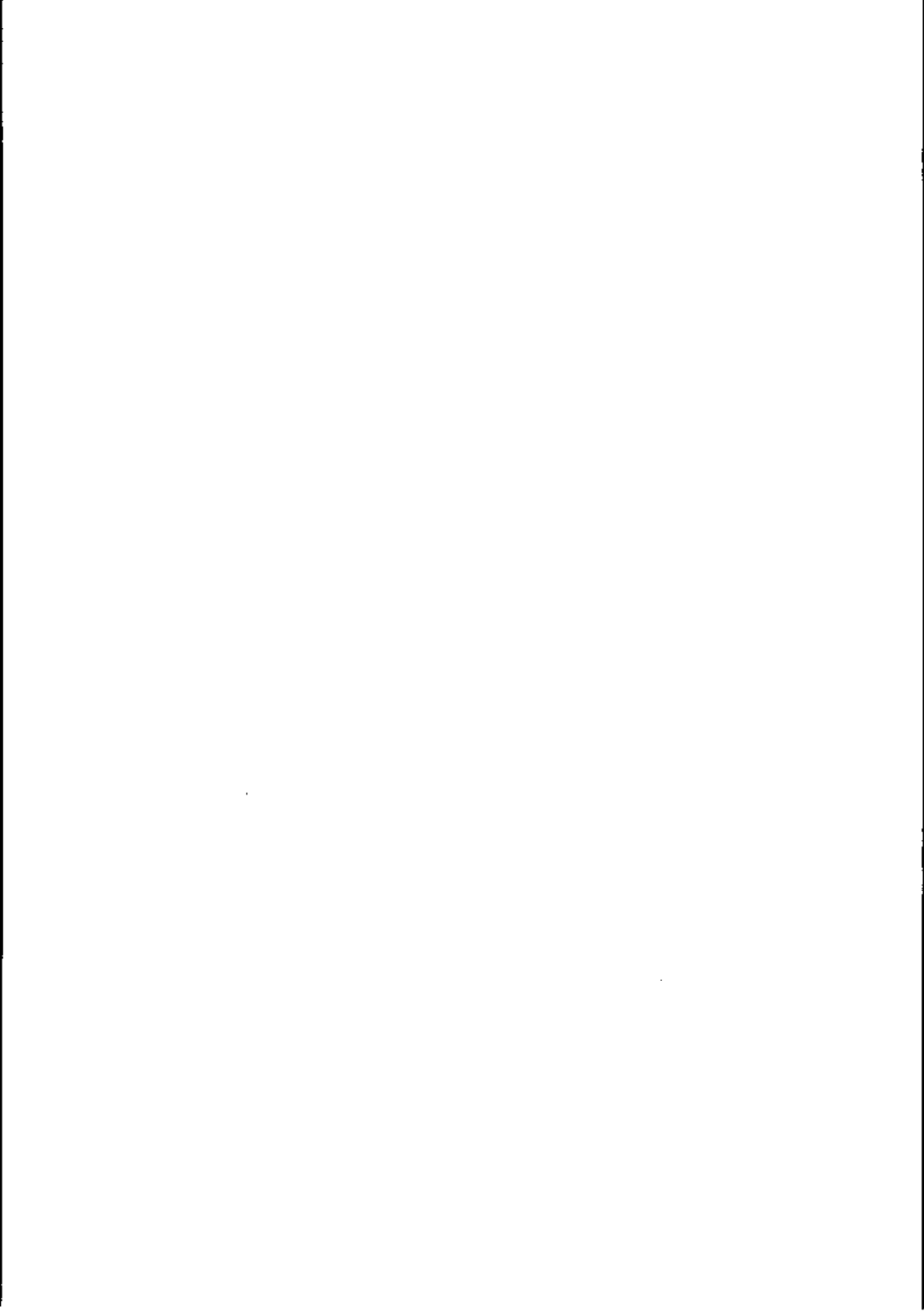


Abb. 1-1 Standardkonfiguration

OLITERM bietet dem Anwendungsprogrammierer ein flexibles Mittel, um über Interrupt gesteuerte full-duplex-Datenübertragungen innerhalb des PCOS/BASIC Programmierungsumfeldes des M20 erzielen zu können. Das Programm puffert sämtliche Eingabe- und Ausgabedaten und ermöglicht so einen höheren Durchsatz. Die Ein- bzw. Ausgabe kann in Bytes oder in Form von Strings geschehen. Erfolgt die Ein- oder Ausgabe in Byte, so kann das Anwendungsprogramm jede gewünschte spezielle Aufzeichnung durchführen.

Die OLITERM Funktionen zur Anordnung und Steuerung der seriellen Schnittstelle des M20 und zur Verarbeitung der Daten als Bytes oder als Strings können nur von BASIC aus aufgerufen werden.

Nach jedem Aufruf einer OLITERM Funktion liefert OLITERM ein String von Status- oder Fehlercodes, wodurch es dem Anwendungsprogramm ermöglicht wird, die Anzeigen der Schnittstelle aufzuzeichnen und auf diese zu antworten. Fehleranzeigen können dazu benutzt werden, die weitere Verarbeitung abzuändern.



## SOFTWARE-VORAUSSETZUNGEN

Eine PCOS-Version der Release 1.2 oder neuer, inklusive OLITERM-Funktionen.

## HARDWARE-VORAUSSETZUNGEN

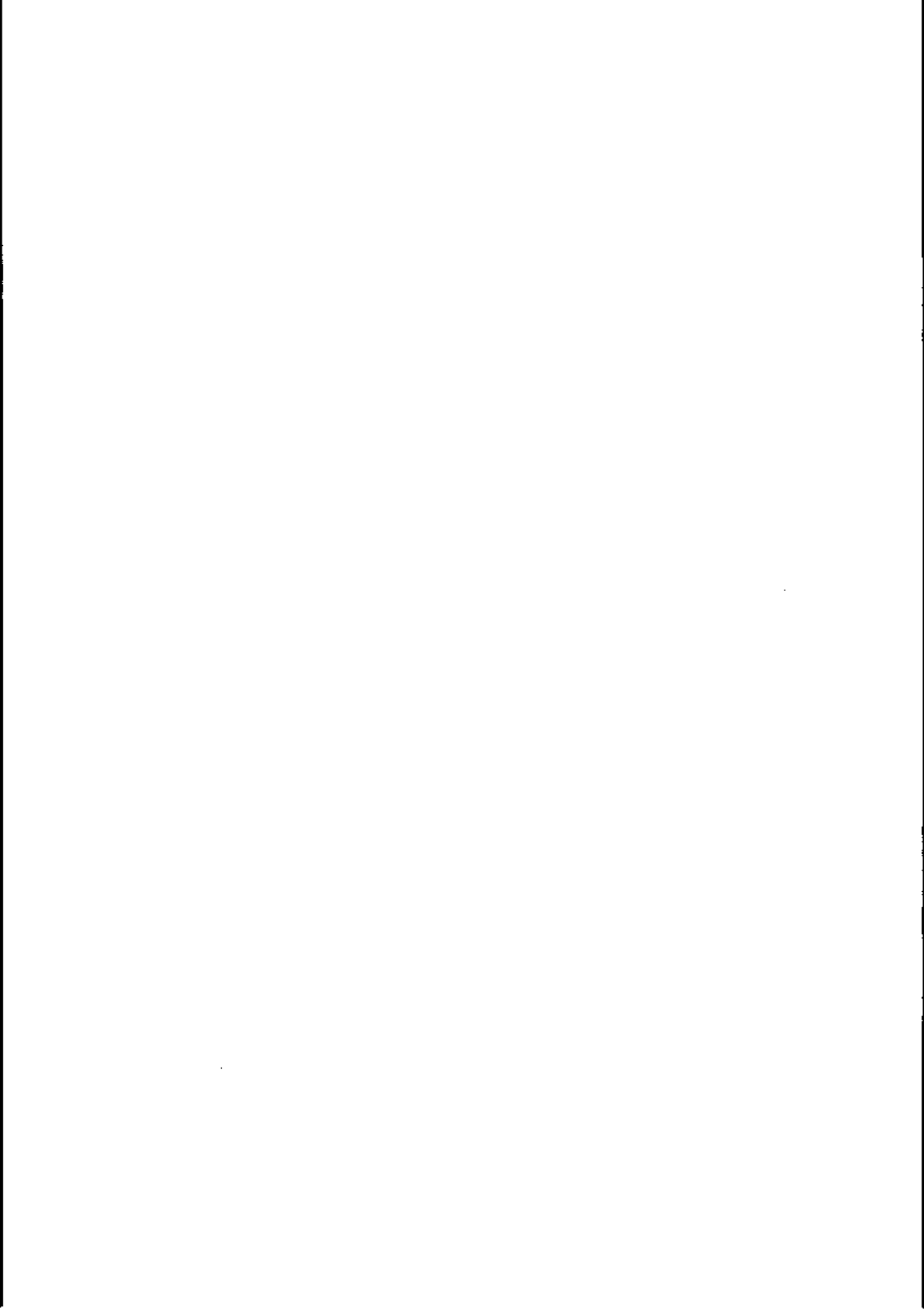
- M20 + einzelne RS-232-C Schnittstelle (Standard)
- ein Modem- oder Peripheriekabel
- ein Modem (Datenverarbeitungseinheit) falls nötig
- Zusatzgeräte wie Drucker usw.
- Stecker, die wie in nachfolgender Tabelle gezeigt, überbrückt werden

STECKER	ÜBERBRÜCKUNG
Interne Sendetaktgeber	P - P1
Interne Empfangstaktgeber	P2 - N2
RS-232-C	I.P. 4-5
DTR von USART gesteuert	R - R1
RTS von USART gesteuert	S - S2
CTS immer aktiviert	T - T1
oder	
CTS von externer Einheit gesteuert	T - T2
(Anmerkung: das Setzen der CTS-Brücke hängt davon ab, ob die externe Einheit das "Clear to Send" Signal steuert.)	

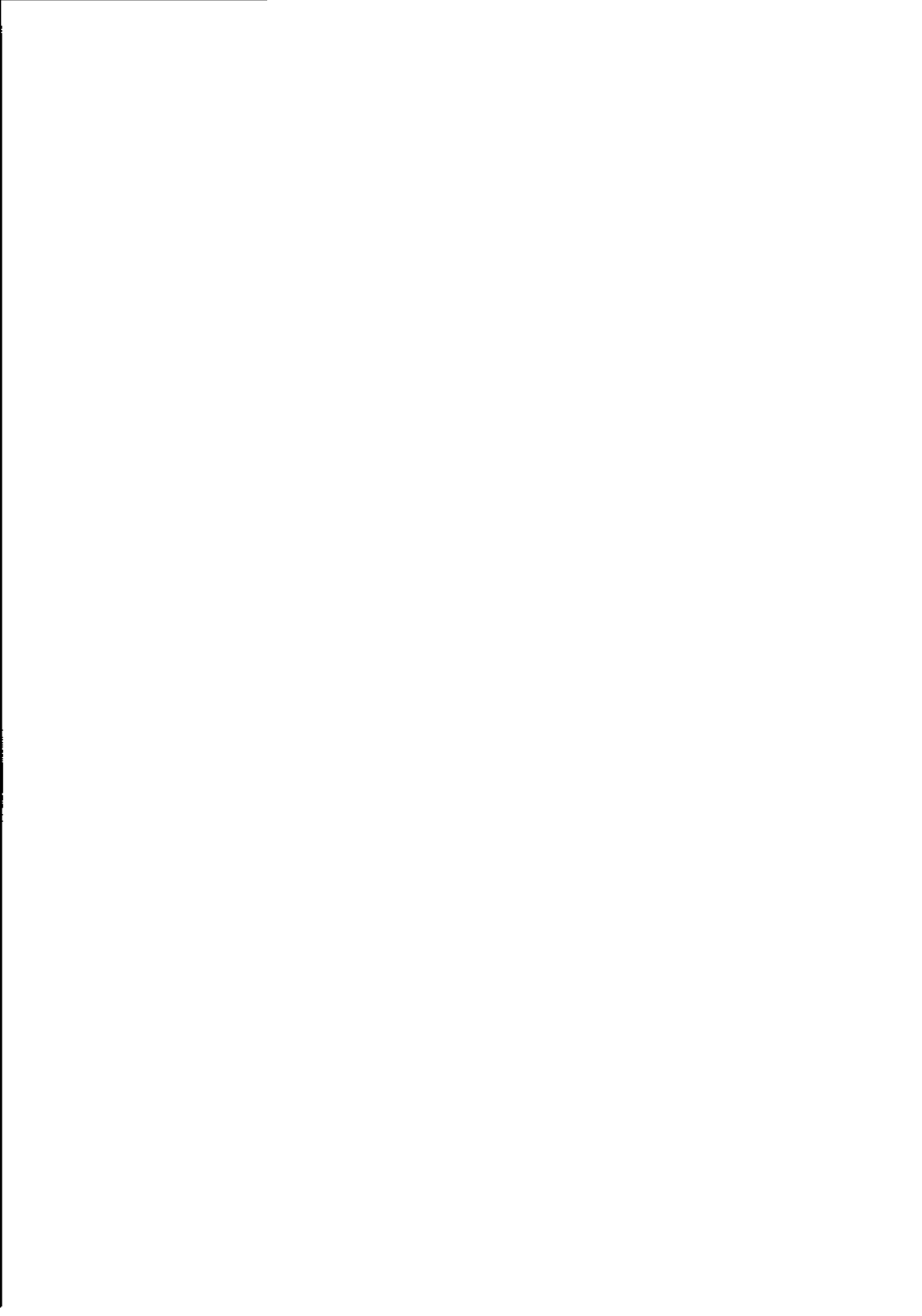
Tabelle 1-1 Steckerüberbrückungen

Um Veränderungen vornehmen zu können, muß man die Maschinenhaube abnehmen. Die Überbrückungsstecker befinden sich hinten dicht neben dem RS-232-C Konnektor wie Abb. 1-2 erkennen läßt.

Die Überbrückungsstecker des Interrupt Patch befinden sich unter den Diskeneinheiten, und diese müssen unter Umständen angehoben werden, wenn man an die Stecker gelangen will.







## BENUTZER-SCHNITTSTELLE

Alle 21 OLITERM Funktionen sind BASIC Calls (Aufrufe), deren Parameterwerte immer in der gleichen Reihenfolge eingegeben werden. Der erste Parameterwert ist die Funktionsnummer, der zweite der Statusstring. Der dritte Parameterwert, falls vorhanden, übermittelt Daten zwischen dem Anwendungsprogramm und OLITERM.

Statuscodes, die im Statusstring aufgelistet werden, sind in Anhang A beschrieben.

Eine OLITERM Funktion behandelt Null-Parameter auf eine von 3 Arten:

- Sie zeigt einen Fehler an. Ist der Parameterwert notwendig, so wird Statuscode "P" zurückgegeben.
- Sie gebraucht einen Standardwert.
- Sie ignoriert den Null-Parameter, da kein Parameter nötig ist.

Alle Funktionen müssen die Kurzform des Schlüsselwortes ("ot") im CALL Namen verwenden, da PCOS nur dieses Format lesen kann.

Die nachfolgende Abbildung (1-3) zeigt die allgemeine Syntax eines OLITERM-Aufrufs.

Die Schreibweise, die in diesem Handbuch verwendet wird, ist in PCOS (Professional Computer Operating System) Handbuch beschrieben.

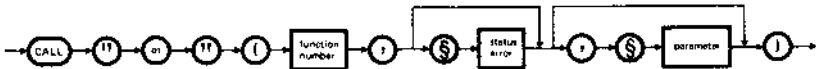


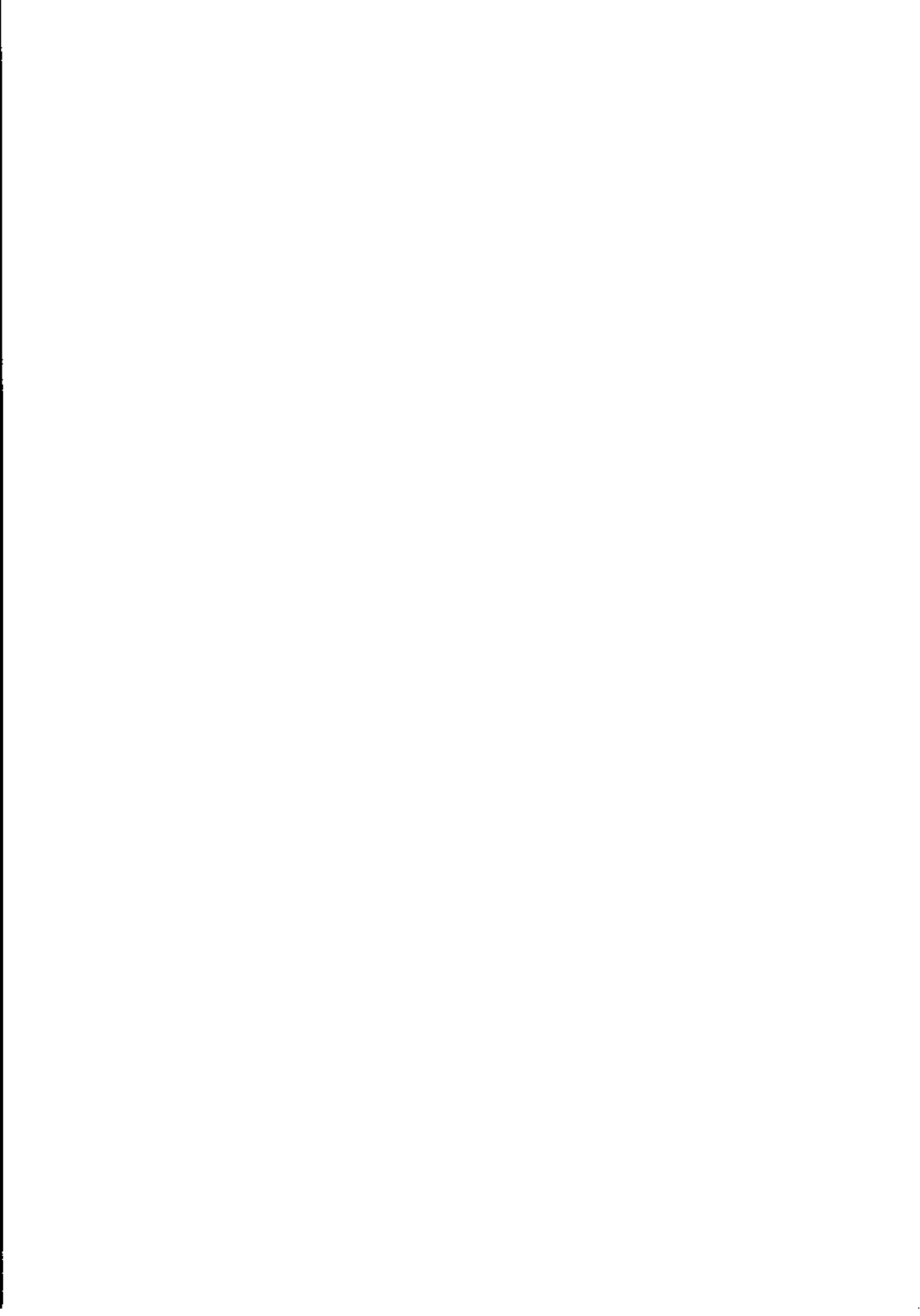
Abb. 1-3 Syntax eines OLITERM Funktionsaufrufs

WO

SYNTAXELEMENT

BEDEUTUNG

SYNTAXELEMENT	BEDEUTUNG
Funktionsnummer	diese Nummer identifiziert die spezifische OLITERM Funktion



SYNTAXELEMENT	BEDEUTUNG
Statusfehler	Diese Stringvariable muß eine Mindestlänge von 16 Zeichen aufweisen. Sie gibt den Fehlercode zurück. Die Variable muß mit einem String aus mindestens 16 Leerzeichen eingeleitet werden, bevor die Funktion aufgerufen wird. Jedes Zeichen der Variable kann einen Wert aus Anhang A annehmen(z.B. ist das erste Zeichen ein A im Falle eines Fehlerausgangs, das zweite Zeichen ein B, wenn OLITERM abgeschlossen ist usw.)
Parameter	ist eine numerische oder eine Stringvariable und kann ein Eingabe- oder Ausgabeparameter sein (je nach aufgerufener Funktion). Handelt es sich um eine numerische Variable, so kann diese auch ein Datenfeld sein.

### Anmerkungen

Das OLITERM Paket ist so beschaffen, daß es leicht eine Schnittstelle mit BASIC-Anwendungsprogrammen bilden kann. Parameterwerte können in Form von Daten oder Statusstrings zurückgegeben werden.

Beim Empfang von Daten aus OLITERM muß das Anwendungsprogramm zuerst die zurückgegebene Variable definieren und muß im Falle eines String genügend Speicherplatz für die zu erwartenden Daten bereitstellen.

Für ein Ganzzahlparameter wird dies folgendermaßen programmiert:

```
10 C%=0
50 CALL "ot" (3 ,, 6 C%)
```

Bei der Rückkehr aus Funktion 3 wird C% das nächste Zeichen aus dem Empfangspuffer enthalten.

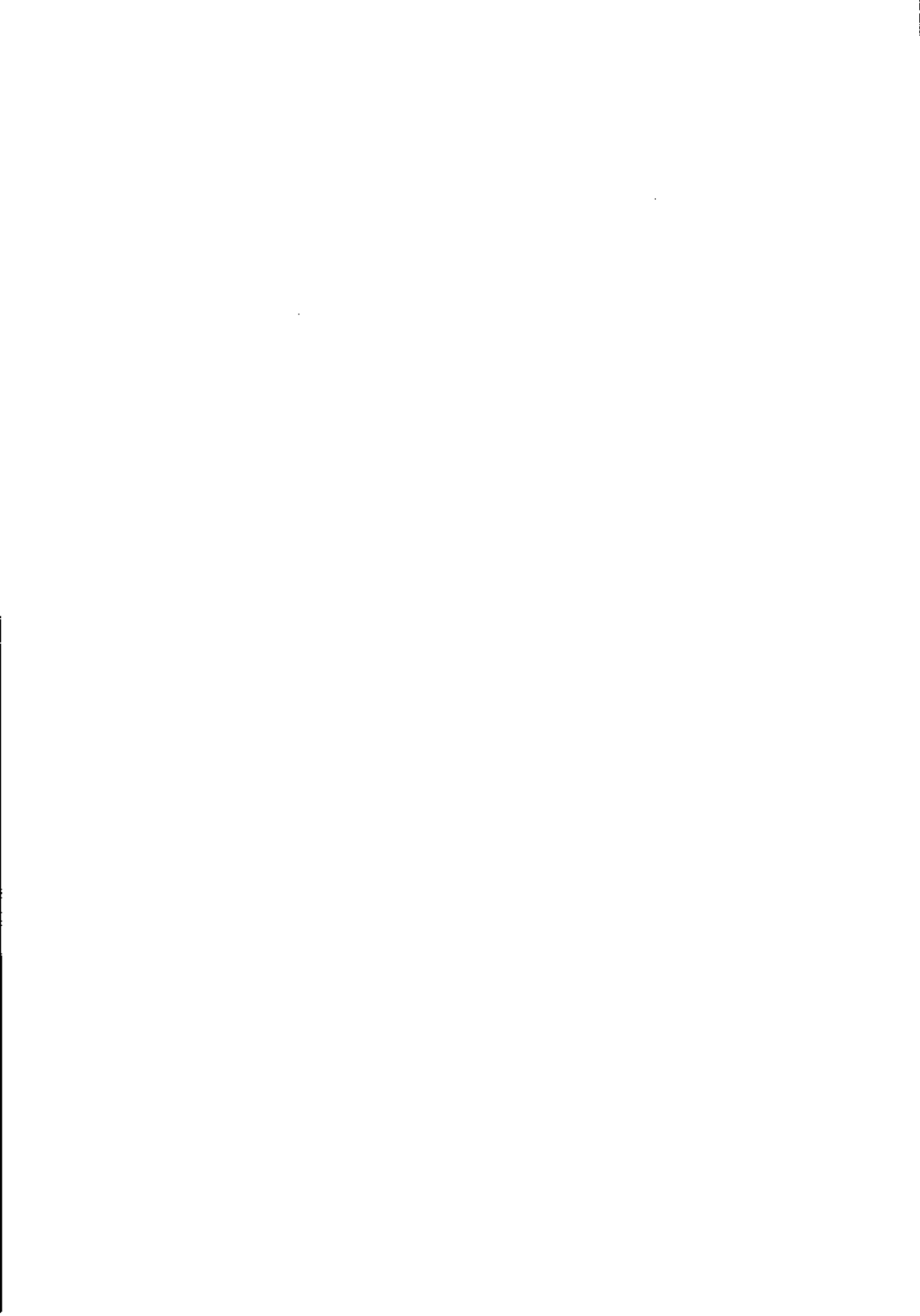
Bei einem Stringparameter muß das Anwendungsprogramm zuerst der Stringvariablen einen Speicherplatz zuweisen. Hierbei sollte es sich nicht um eine literale Zuweisung handeln, da das OLITERM-Programm die Änderung des Programmtests beendet. Ein Beispiel einer gültigen Verfahrensweise hierfür ist:

```
30 S$ = SPACES (20)
90 CALL "ot" (7, S$)
```



Dies wird einen String im variablen Speicherbereich von BASIC schaffen und ihn zum Anwendungsprogramm zurückgeben. Der zugeordnete Speicherplatz muß so groß sein, daß er den längsten String einschließlich des "End of Message" (EOM)-Zeichens aufnehmen kann. Im oben aufgeführten Beispiel bestünde der längste zulässige Statusstring aus 19 Zeichen plus EOM-Zeichen.

Es ist außerordentlich wichtig, daß dem String genug Platz zum Speichern seiner Daten zugeordnet wird, da OLITERM dies nicht überprüft. Bei empfangenen Daten wird die maximale Länge im Initialisierungsfeld festgesetzt. Der Statusstring muß eine Mindestlänge von 16 Bytes haben, um alle 16 Statusbuchstaben von "A" bis "P" unterbringen zu können.



## **2. FUNKTIONSAUFRUFE**

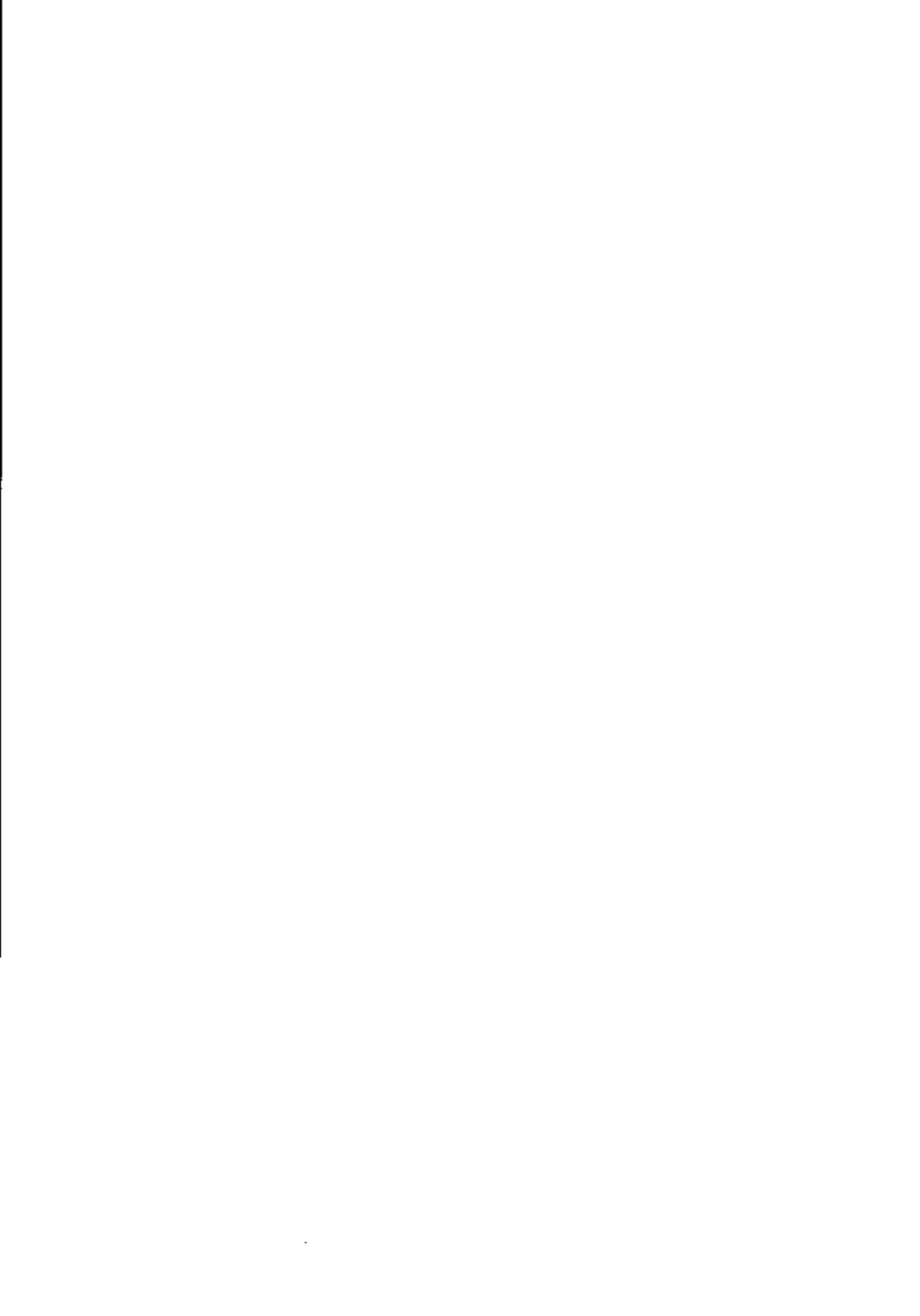


## Zu diesem Kapitel

Dieses Kapitel gibt eine detaillierte Beschreibung der 21 Funktionen des Oliterm-Pakets.

### Inhalt

OLITERM INITIALISIEREN FUNKTION 1	2-1	RÜCKANTWORTDATEN SENDEN FUNKTION 17	2-20
OLITERM ABSCHLIESSEN FUNKTION 2	2-4	SENDE-TIMEOUT SETZEN FUNKTION 18	2-21
BYTE LESEN FUNKTION 3	2-5	KONTROLLZEICHEN SETZEN FUNKTION 19	2-22
DATEN EMPFANGEN FUNKTION 4	2-7	SCHNITTSTELLE AUSSCHALTEN FUNKTION 20	2-23
EIN BYTE SCHREIBEN FUNKTION 5	2-9	SCHNITTSTELLE EINSCHALTEN FUNKTION 21	2-24
DATEN SENDEN FUNKTION 6	2-10		
OLITERM STATUS FUNKTION 7	2-11		
EMPFANGSPUFFER LÖSCHEN FUNKTION 8	2-12		
EMPFANG DEAKTIVIEREN FUNKTION 9	2-12		
EMPFANG AKTIVIEREN FUNKTION 10	2-14		
EMPFANGS-TIMEOUT SETZEN FUNKTION 11	2-15		
SENDEPUFFER DEAKTIVIEREN FUNKTION 12	2-16		
SENDEN DEAKTIVIEREN FUNKTION 13	2-16		
SENDEN AKTIVIEREN FUNKTION 14	2-17		
UNTERBRECHUNG SENDEN FUNKTION 15	2-18		
RÜCKANTWORT SETZEN FUNKTION 16	2-19		



## OLITERM INITIALISIEREN - FUNKTION 1

Die Funktion "initialisieren" überträgt die Daten, die notwendig sind, um die Oliterm Schnittstelle zu initialisieren, um Parameter aufzunehmen und zu transportieren.

Diese Funktion muß in einem Anwendungsprogramm der erste Funktionsaufruf sein. Um Parameter zu ändern, kann der Funktionsaufruf wiederholt werden.

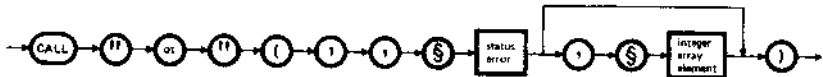


Abbildung 2-1 Oliterm initialisieren

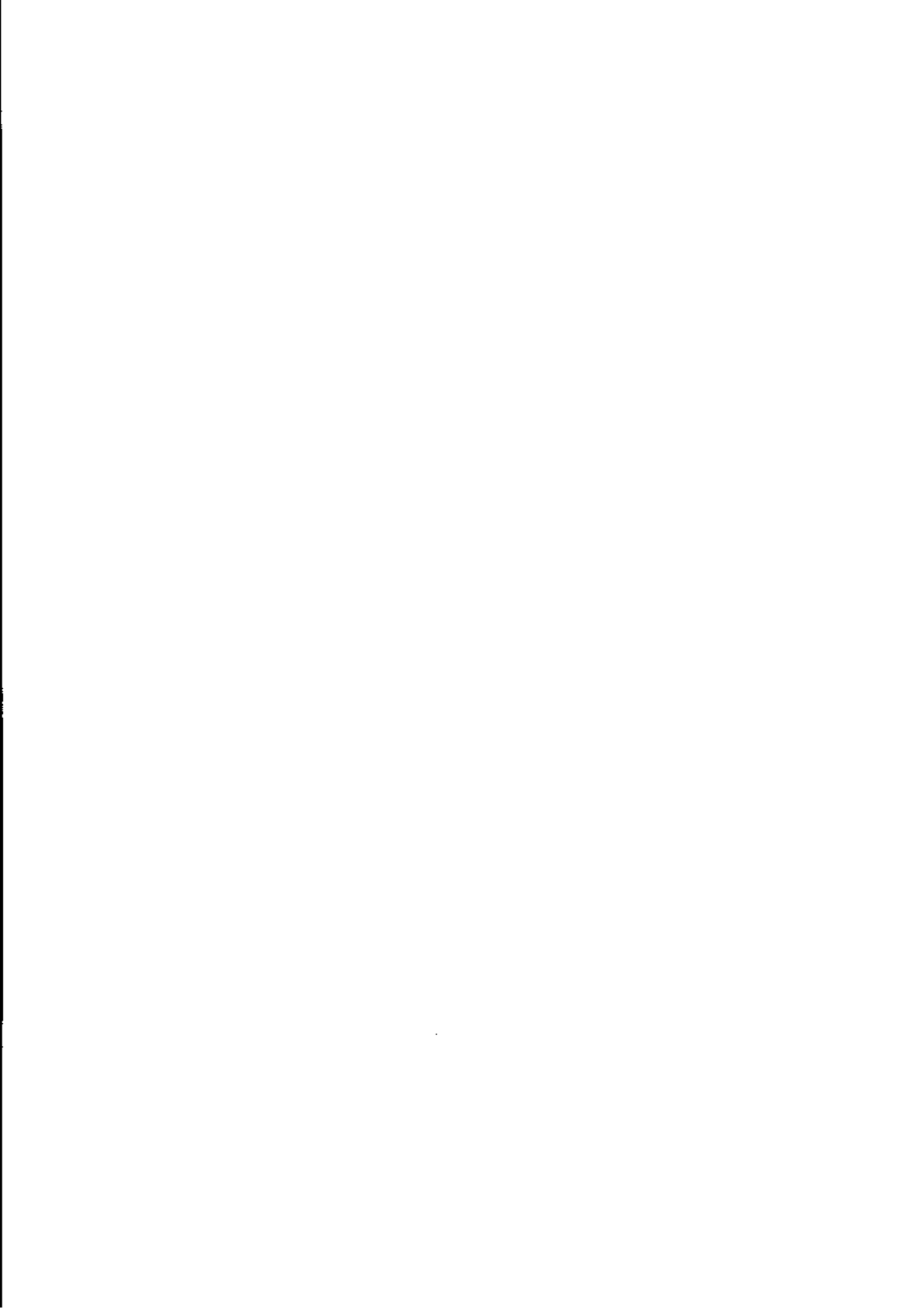
Wo

Syntaxelement	Bedeutung
1	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
Integer Array Element	das erste Element aus einem "integer array" (ganzzahliges Datenfeld) der Dimension 14. Dieses Datenfeld enthält Informationen zur Initialisierung der Oliterm Schnittstelle, zum Empfangen und Senden von Parametern (siehe unten)

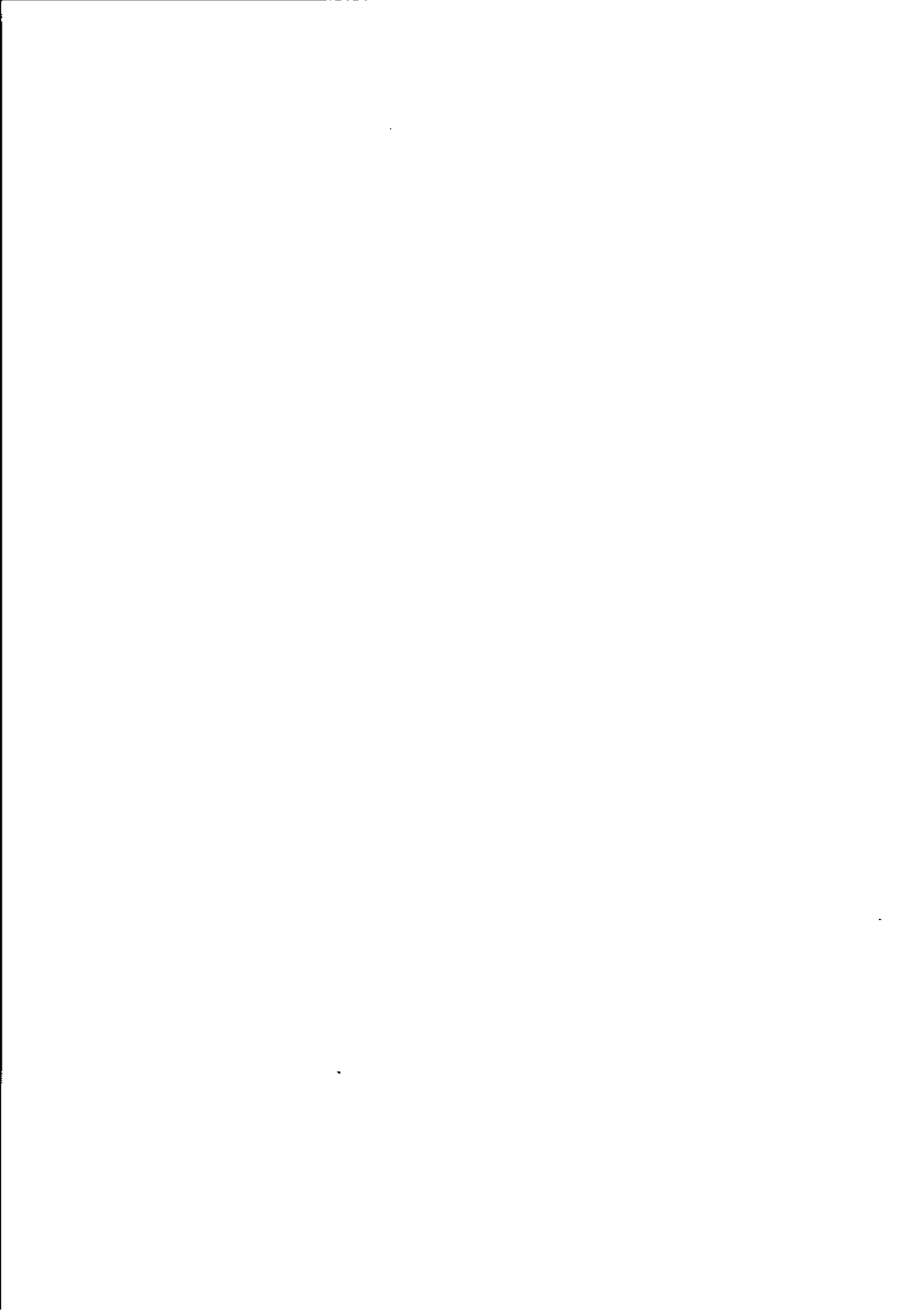
**Statusfehler:** A, P

**Integer Array (ganzzahliges Datenfeld)**

Die folgende Tabelle zeigt die Werte, die vor Aufruf der Funktion jedem Element des Datenfeldes zugeordnet werden müssen. Will der Benutzer die Standardwerte gebrauchen, so kann er das integer array Element weglassen.



INDEX	FUNCTION	VALUES	DEFAULT
Port Parameters: 0	Parity	0 = None 1 = Odd Parity 2 = Even Parity	0
1	Data Bits	5, 6, 7, 8	8
2	Stop Bits	0 = 1 Stop Bit 1 = 1.5 Stop Bits 2 = 2 Stop Bits	0
3	Baud Rate	0 = 110 1 = 300 2 = 600 3 = 1200 4 = 2400 5 = 4800 6 = 9600 7 = 19200	6
I/O Parameters: 4	Receive Control	0 = No Control 1 = X-ON/X-OFF 2 = DTR	1
5	Transmit Control	0 = No Control 1 = X-ON/X-OFF 2 = DSR	1
6	Receive Mode	0 = Transparent 1 = Normal (check for ENQ, X-ON and X-OFF)	1
Message Manager:			
7	Receive End of Message	0..255	13(CR)
8	Receive End of File	0..255	4(EOT)



INDEX	FUNCTION	VALUES	DEFAULT
9	Receive Data Mask	0..255	127
10	Maximum Receive Message length	1..255	80
11	Transmit End of Message, EOM1	0..255	13(CR)
12	Transmit EOM2	0..255	10(LF)
13	Padding	0..32767: Nulls - 1 .. -32768: Clock ticks	0

Tabelle 2-1 Funktionen, ihre Wertebereiche und Standardwerte

### Merkmale

Wird -1 für irgendeinen Parameterwert eingegeben, so wird er durch einen Standardwert ersetzt. Alle Standardwerte sind in Tabelle 2-1 angegeben. Olitem gebraucht alle Standardwerte auch dann, wenn das Anwendungsprogramm kein Initialisierungsdatenfeld durchläuft.

Wenn die Olitem Schnittstelle DTR steuert, wird DTR während der Initialisierung abgeschaltet.

Die Initialisierungsparameter 0, 1, 2 und 3 legen die Hardware-Bedingungen der seriellen Schnittstelle fest. Diese richten sich nach der Übertragungsart.

Die Parameter 4, 5 und 6 definieren, wie das Olitem Programm die Steuerung des Sende- und des Empfangsvorgangs von Daten beeinflusst.

Parameter 4 - Empfangskontrolle - bestimmt, wie Olitem der anderen Station Sendestart und Sendestop signalisiert.

Dies betrifft sowohl das Füllen und Leeren der Empfangspuffer als auch explizite Aufrufe zum Aktivieren und Deaktivieren des Empfangs. In Modus 0 sendet Olitem kein Signal und hat keine Steuerung über die Fernübertragung. In Modus 1 sendet Olitem X-Off (CTRL-S), um die Übertragung zu beenden und X-ON (CTRL-Q), um sie zu starten. In Modus 2 schaltet Olitem DTR aus, um die Übertragung zu beenden, und schaltet DTR ein, um die Übertragung zu starten. DTR wird von der anderen Station normalerweise als DSR gesehen.



Parameter 5 - Sendekontrolle - legt fest, wie der Oliterm-Sendevorgang von außen aktiviert und deaktiviert wird. In Modus 0 kann die andere Station die Übertragung nicht beeinflussen. In Modus 1 deaktiviert bzw. aktiviert der Empfang von X-OFF oder X-ON die Übertragung. In Modus 2 kontrolliert Oliterm den Status von DSR. Ist DSR nicht gesetzt, stoppt die Übertragung. Es wird weiter gesendet, wenn DSR wieder gesetzt wird.

Parameter 6 - Empfangsmodus. Im Modus 0 werden alle Daten unkontrolliert im Empfangspuffer gespeichert. Im Modus 1 wird geprüft, ob ein ENQ, X-ON oder X-OFF empfangen worden ist.

Wenn ja, wird das Zeichen gelöscht und folgende Reaktion wird ausgelöst:

ENQ : Startet Rückmeldedaten

X-ON : Senden aktivieren

X-OFF : Senden deaktivieren

Sämtliche Zeichen mit Ausnahme dieser 3 sind im Empfangspuffer gespeichert.

Details über die Parameter 7, 8, 9, 10, 11, 12 und 13 finden Sie in den Abschnitten über die Funktionen "Daten empfangen" und "Daten senden".

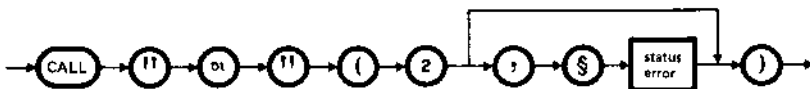
Die Initialisierung schlägt fehl, wenn ein Parameter außerhalb seines Wertebereichs liegt, was Oliterm in einen undefinierten Zustand versetzen kann.

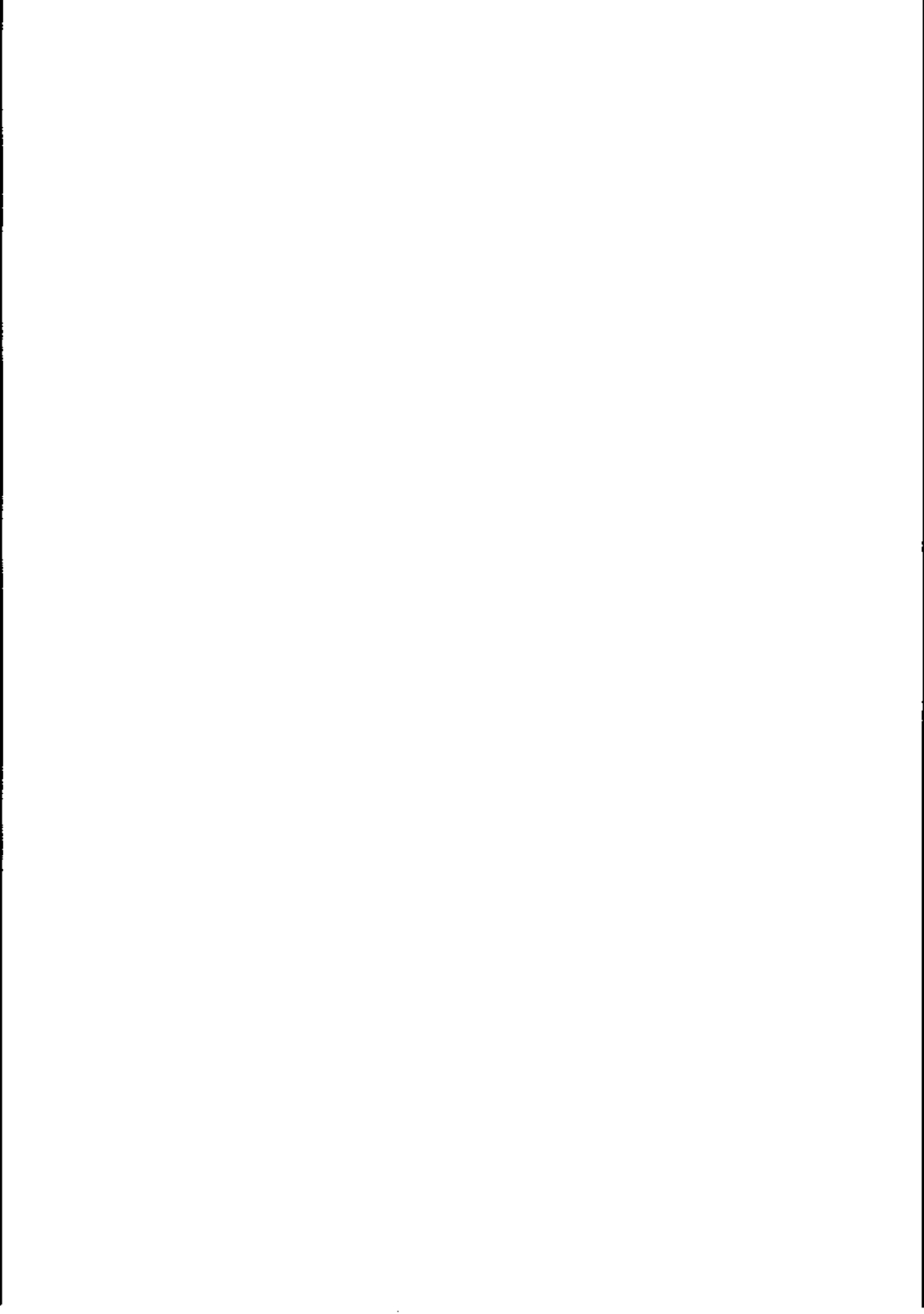
### Beispiel

Siehe Kapitel 3, Abbildung 3-1, Zeilen 140 bis 290.

### OLITERM ABSCHLIESSEN - FUNKTION 2

Die Funktion "abschließen" sorgt für einen korrekten Abschluß von Oliterm.





Wo

Syntaxelement	Bedeutung
2	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

**Statusfehler:** A, C, D, E, F, G, H, I, M, N, O

### Merkmale

Die Funktion schließt Oliterm folgendermaßen ab:

- Sie stoppt den Empfang
- Sie wartet bis zu zehn Sekunden auf das Leeren des Sendepuffers. Leert sich dieser innerhalb dieser Zeitgrenze nicht, so wird Fehler D, Timeout gesetzt.
- Die serielle Schnittstelle wird wieder in ihre PCOS-Konfiguration versetzt. Die Geschwindigkeit (BPS) wird nicht verändert.

Nach dem Aufruf von "Oliterm abschließen" muß die Funktion "Oliterm initialisieren" noch einmal aufgerufen werden, bevor andere Funktionsaufrufe gemacht werden können.

Werden Aufrufe nach Abschließen der Schnittstelle gemacht, gibt Oliterm einen Statusstring mit allen Statusfehlern zurück.

### BYTE EMPFANGEN - FUNKTION 3

"Byte empfangen" gibt das nächste Zeichen aus dem Empfangspuffer zurück.

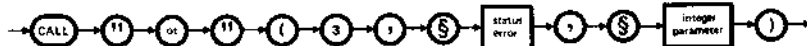
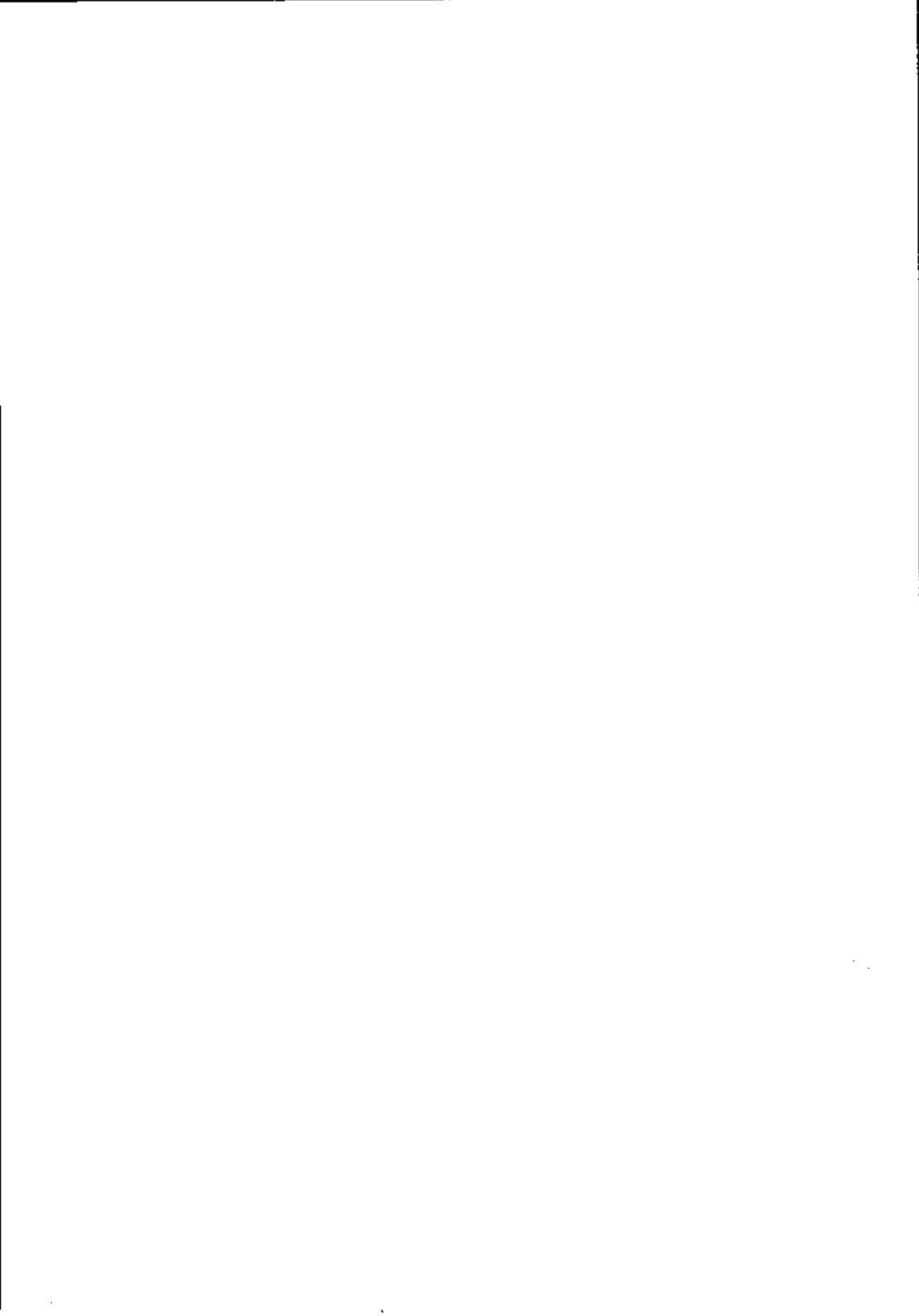


Abbildung 2-3 Byte empfangen



Wo

Syntaxelement	Bedeutung
3	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
ganzzahliger Parameter	zurückgegebener Wert der Daten. Daten liegen im Bereich 0 bis 255.

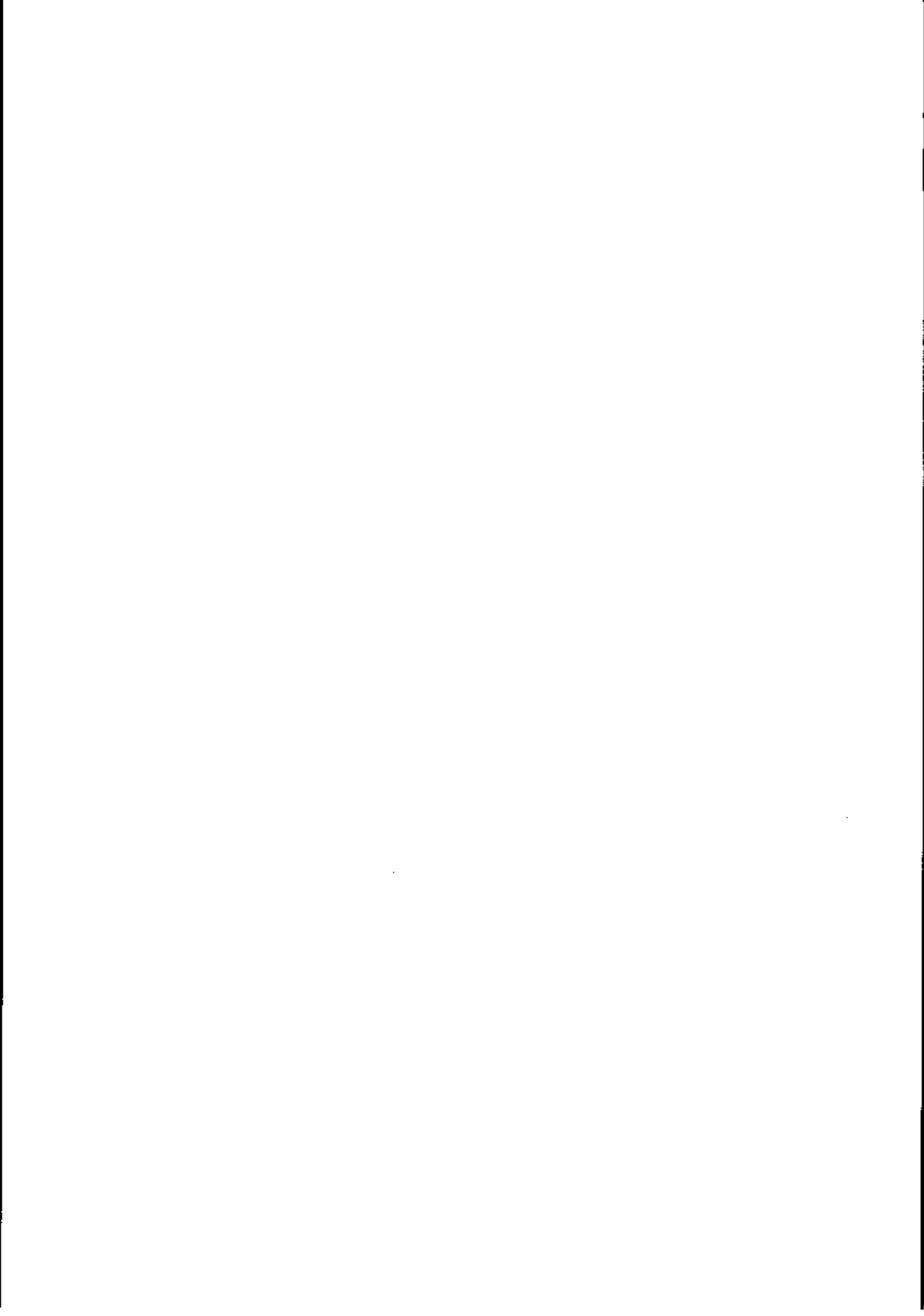
**Statusfehler:** A, B, D, F, G, H, I, J, P

### Merkmale

Die Funktion gibt das nächste Zeichen aus dem Empfangspuffer zurück und setzt dann die passenden Datenfehlerindikatoren: G, H, I, J. Tritt ein Timeout auf, so gibt die Funktion "Byte empfangen" die Fehler A und D zurück, ohne den Wert des ganzzahligen Parameters zu verändern. "Byte empfangen" überprüft keine Daten und sucht keine Steuerzeichen. Diese Funktion gibt alle Daten so zurück, wie sie sie empfangen hat.

### Beispiel

Anzeige	Kommentar
100 D% = 0 110 CALL "ot" (3, S\$, D%) 120 IF INSTR (S\$, "A") <> 0 THEN 300	Zeile 100 initialisiert D% zu 0, und Zeile 110 ruft die "Byte-lesen"-Funktion auf.
130 REM Process Data . . .	Zeile 120 kontrolliert den Statusfehler-Parameter.
300 REM Process Error	



## DATEN EMPFANGEN - FUNKTION 4

"Daten empfangen" gibt Zeichen im Empfangspuffer zum Stringparameter zurück, bis das Trennzeichen "Receive EOM" (ausgewählt von Funktion 1) gefunden worden ist.

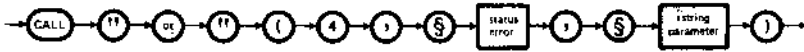


Abbildung 2-4 Daten empfangen

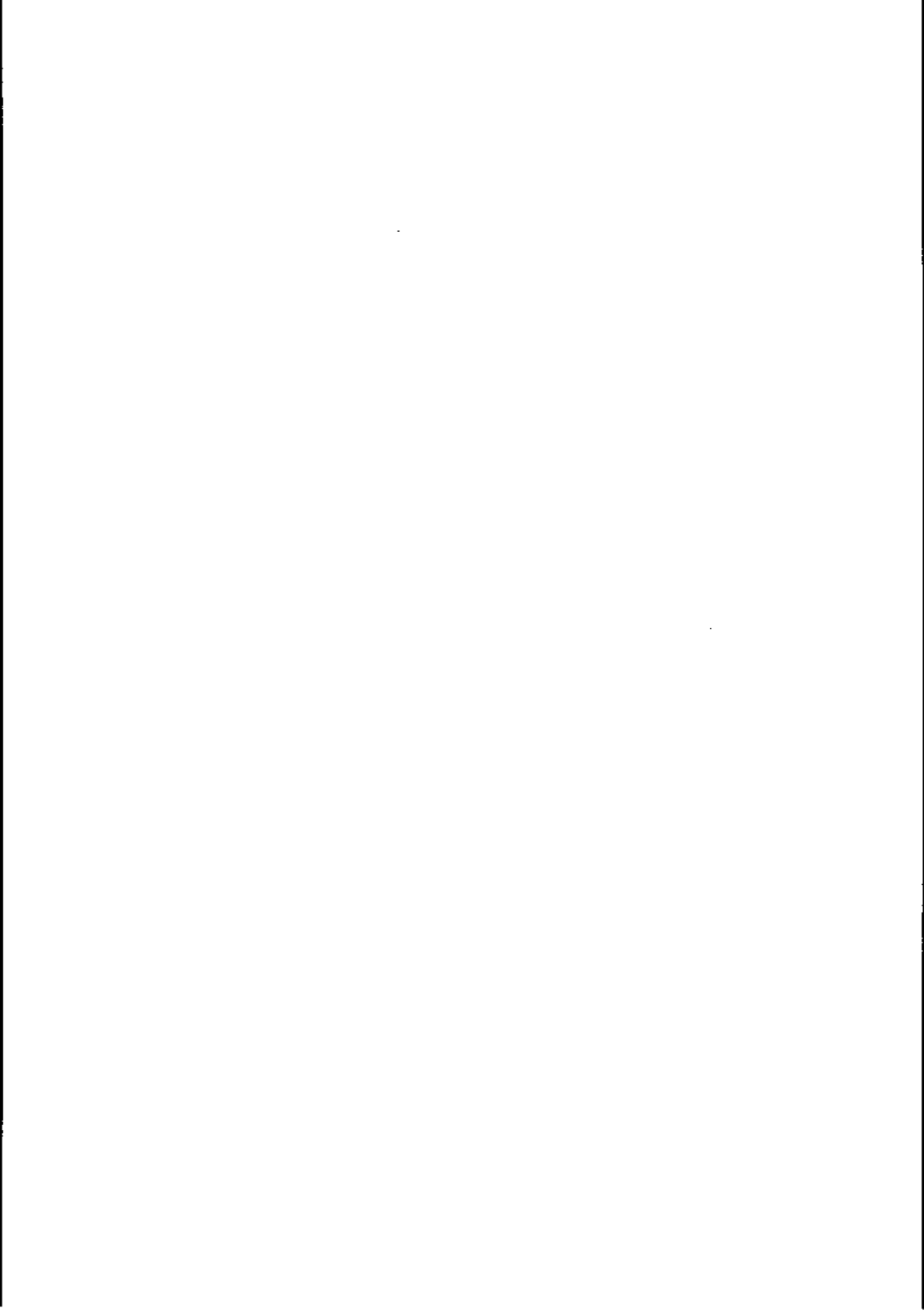
Wo

Syntaxelement	Bedeutung
4	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
Stringparameter	dieser String empfängt die Daten von der Datensteuerung

**Statusfehler:** A, C, D, E, F, G, H, I, J, K, L

### Merkmale

Alle empfangenen Datenbytes werden vom Initialisierungsparameter 9-Empfangsmaske- überprüft. Die Überprüfung läuft folgendermaßen ab:



## ZUSTAND

## KOMMENTAR

Receive Timeout	Dies beendet die Daten, und im Status-String werden Status A und D gesetzt. Dieser Fehler hat keine Auswirkung auf den Empfangspuffer. Eine volle Datenmenge kann aus Read Message-Strings aufgebaut sein, die von einem Timeout beendet werden, bis ein EOM/EOF eintritt.
Break	Dies beendet die Daten, und Status A und G werden in dem Status-String gesetzt. Dieser Fehler hat keine Auswirkung auf den Empfangs-Speicher oder den folgenden Aufruf "Read Message".
Receive EOM	Die Nachricht ist beendet. Wenn das Kontrollzeichen-Array, das durch die Funktion 19 gesetzt wurde, das EOM-Zeichen gesetzt hat (Initialisation Parameter 7), wird es in den Daten erscheinen. Andernfalls wird es nicht beachtet.
Receive EOF	Wenn die Datenmenge leer ist, wenn EOF (Initialisation Parameter 8) eintritt, stoppt die Nachricht, und Status K wird im Status-String gesetzt.  Wenn die Datenmenge nicht leer ist, endet sie, als wenn ein EOM empfangen worden wäre. Der nächste Aufruf "Read Message" endet mit dem Status K. Das EOF-Zeichen erscheint niemals in einer Nachricht.
Message Buffer Full	Wenn die Größe der empfangenen Daten der größten Empfangsdatenmenge entspricht (Initialisation Parameter 0), enden die Daten und Status A und L werden gesetzt. Die größte Datenmenge, die ohne Error "L" empfangen werden kann, ist Max-1, weil die größte Menge auch das EOM-Zeichen enthält.

Jedes Zeichen wird daraufhin überprüft, ob es ein Steuerzeichen ist. Nicht-Steuerzeichen werden als Teil der Daten eingegeben, bis eine der oben angegebenen Abschlußbedingungen eintritt.

Steuerzeichen werden einzeln mit dem Steuerzeichenfeld verglichen (siehe Funktion 19). Ist das betreffende Steuerzeichen im Datenfeld vorhanden, wird es als Teil der Daten eingegeben. Ist es nicht vorhanden, so wird es weggelassen.



## Anmerkungen

Das Anwendungsprogramm muß im Datenstringpuffer für genügend Speicherplatz sorgen, um den längstmöglichen Satz plus EOM-Zeichen empfangen zu können.

## BYTE SENDEN - FUNKTION 5

"Byte senden" setzt ein 8-bit Byte in den Sendepuffer.



Abbildung 2-5 Byte senden

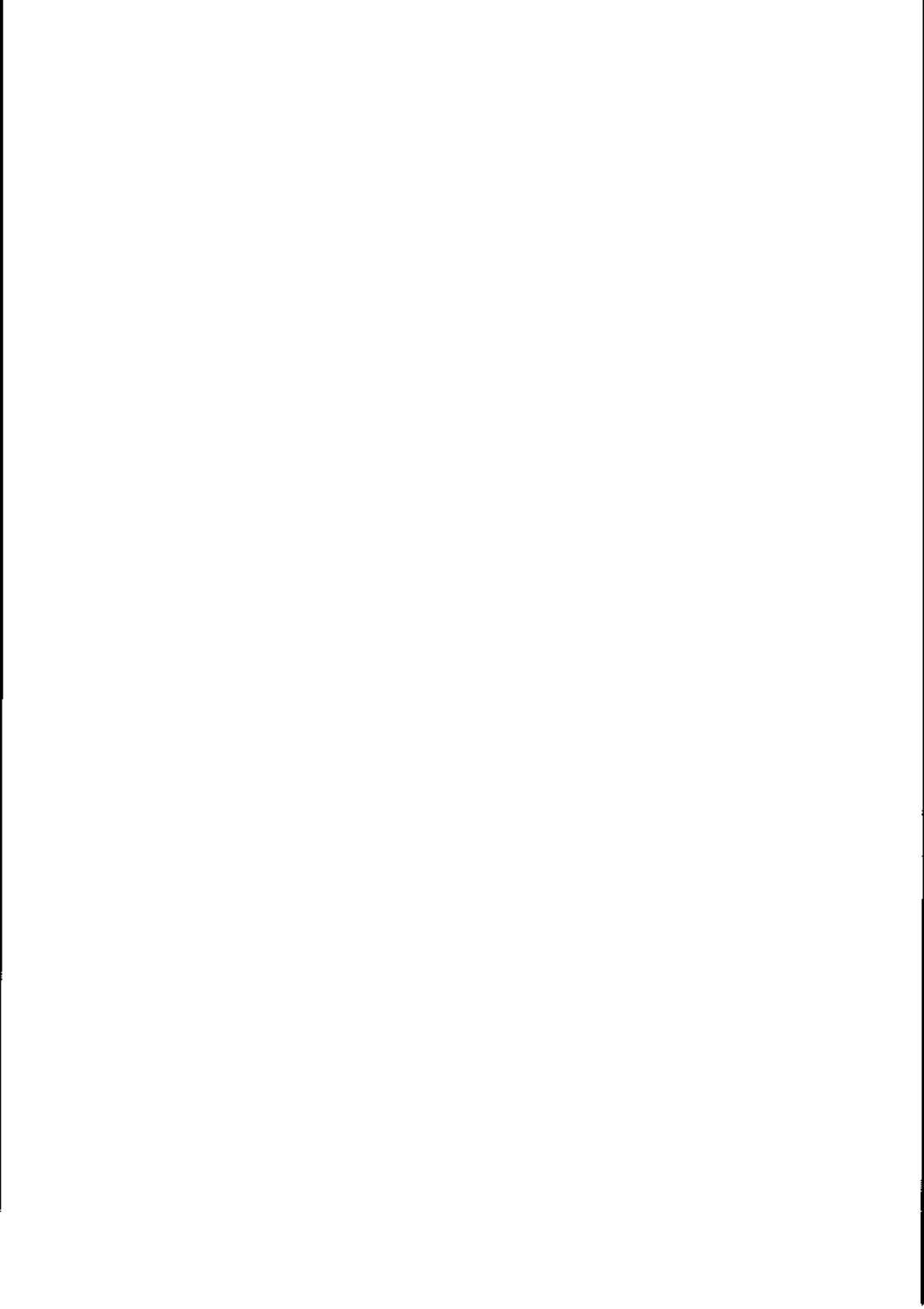
Wo

Syntaxelement	Bedeutung
5	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
ganzzahliger Parameter	zu sendender Wert. Das niedrige Byte wird verwendet, das hohe Byte wird ignoriert.

**Statusfehler:** A, B, C, D, M, N, O, P

### Merkmale

Die Daten werden gesendet, ohne auf besondere Zeichen hin untersucht zu werden. Ist ein Sende-Timeout angegeben, so wird "Byte senden" die Wartezeit für Platz im Sendepuffer abwarten. Wird kein Platz frei, so werden die Fehler A, D und C oder N (je nach Grund für den Timeout) zurückgegeben.



Die Daten werden nicht gesendet. Um die Daten senden zu können, muß die den Timeout verursachende Bedingung aufgelöst und die Funktion "Byte senden" erneut aufgerufen werden.

### Beispiel

```
110 CALL "ot"(5, $$, ASC ("A" )
```

Zeichen "A" wird gesendet.

### DATEN SENDEN - FUNKTION 6

Die Funktion "Daten senden" sendet den String, ohne die Daten zu prüfen.

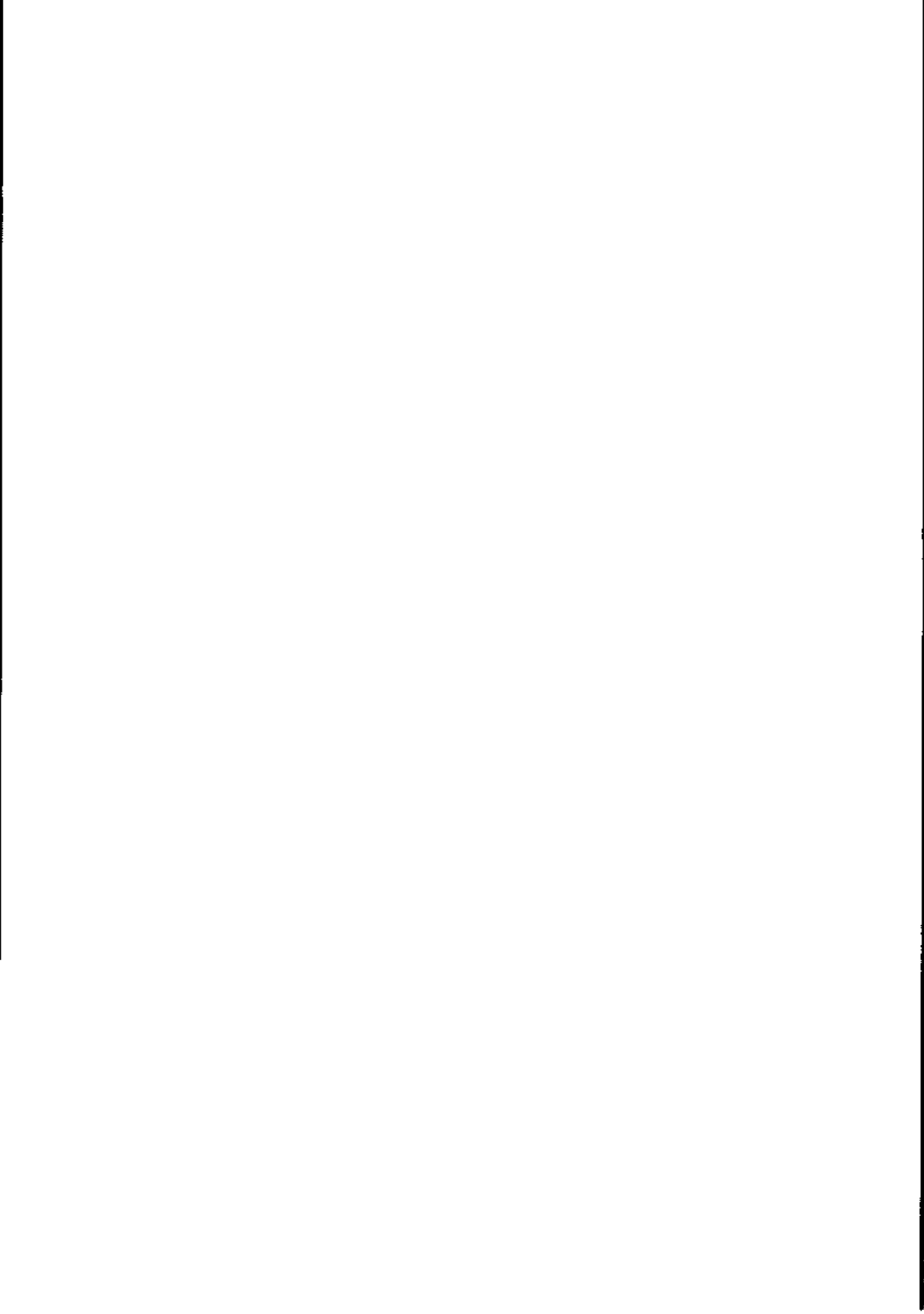


Abbildung 2-6 Daten senden

Wo

Syntaxelement	Bedeutung
6	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
Stringparameter	dieser String wird mit angehängtem EOM-Zeichen und Nullen gesendet

Statusfehler: A, C, D, M, N, O, P



## Merkmale

Nach dem Senden des String sendet diese Funktion irgendwelche EOM-Zeichen ohne Nullen, die in der Initialisierung angegeben wurden, als Parameter 11 und 12.

Dann prüft sie den Zeitparameter 13.

Ist diese Zahl größer als Null, wird diese Zahl an Nullen ebenfalls gesendet.

Ist sie kleiner als Null, wartet "Daten senden" darauf, daß die Daten gesendet werden. (Um die Möglichkeit einer unbegrenzten Wartezeit in bestimmten Situationen zu vermeiden, ist diese auf 60 Sekunden begrenzt). Die Funktion wartet dann noch eine Anzahl von Takten der Echtzeituhr ab. Die Anzahl der Takte ist der absolute Wert des Zeitparameters. Ist der Wert des Zeitparameters null, so findet kein Padding statt. Tritt bei "Nachrichten senden" ein Timeout auf, endet das Senden der Daten ohne weitere Verarbeitung. Die Daten wurden wahrscheinlich unvollständig oder ohne Anhängen des EOM bzw. ohne Padding gesendet, aber das hängt vom Zeitpunkt des Timeout ab.

Der Datenstring darf keine Nullen (0) enthalten, da dies ein falsches Nachrichtenende beim ersten Nullzeichen bewirken würde.

## OLITERM STATUS - FUNKTION 7

OLITERM STATUS gibt einen kompletten Bericht über den Zustand von Oliterm zurück.

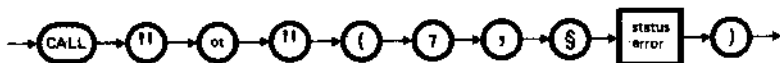


Abbildung 2-7 Oliterm Status

Wo

Syntaxelement	Bedeutung
7	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen



Statusfehler: A, B, C, D, E, F, G, H, I, J, M, N, O

### Merkmale

Die Empfangsdatenfehler: G, H, I, J zeigen an, daß sich Daten mit diesen Fehlern im Puffer befinden.

### EMPFANGSPUFFER LÖSCHEN - FUNKTION 8

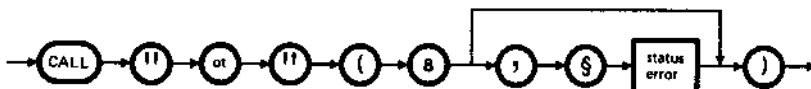


Abbildung 2-8 Empfangspuffer löschen

Wo

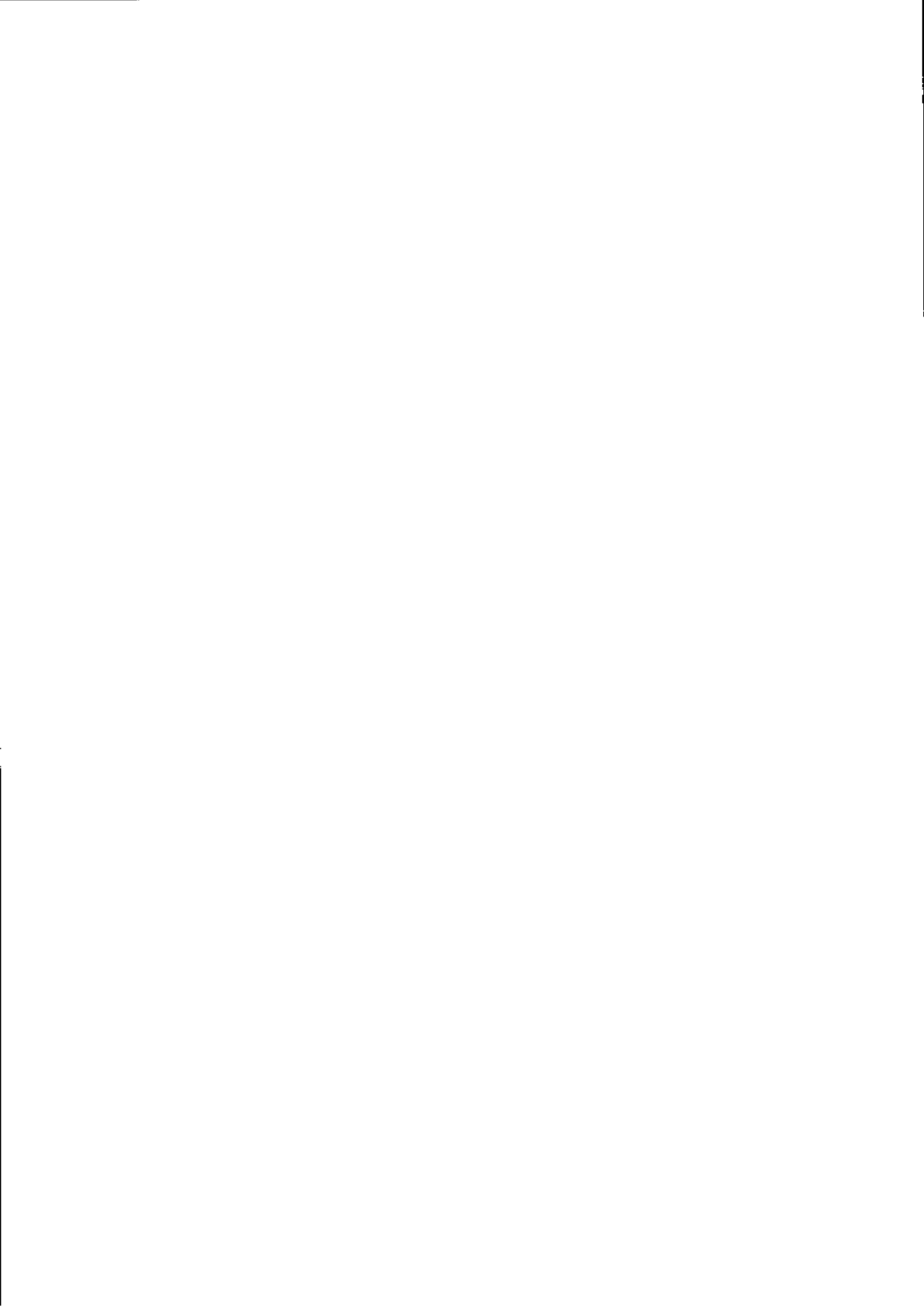
Syntaxelement	Bedeutung
8	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

### Merkmale

Diese Funktion löscht den Empfangspuffer und die Datenfehlerindikatoren. Alle Daten im Empfangspuffer gehen verloren.

### EMPFANG DEAKTIVIEREN - FUNKTION 9

"Empfang deaktivieren" befiehlt der entfernten Einheit, das Senden zu beenden.



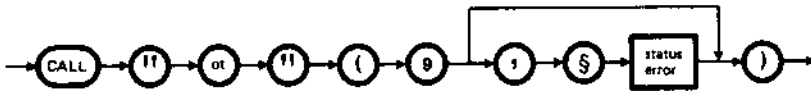


Abbildung 2-9 Empfang deaktivieren

Wo

Syntaxelement	Bedeutung
9	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

Statusfehler: A (Wenn Empfangssteuerung = 0)

### Merkmale

Die Funktion "Empfang deaktivieren" gebraucht den bei der Initialisierung ausgewählten Steuerungsmodus, um den Sendevorgang zu beeinflussen. So kann die Steuerung eine der beiden folgenden Formen annehmen:

- X-OFF senden
- DTR ausschalten

Dieser Funktionsaufruf setzt die automatische Füllsteuerung des Empfangspuffers außer Kraft. Das Anwendungsprogramm muß den Empfang ausdrücklich aktivieren. Der Empfang wird durch diesen Befehl nicht wirklich ausgeschaltet, aber Oliterm versucht, den Empfang zu beenden, indem es die entfernte Einheit daran hindert zu senden. Auch nach Aufruf dieser Funktion können Daten noch empfangen werden. Ist für den Empfang keine Steuerung angegeben worden, gibt Oliterm Statusfehler "A" zurück und versucht nicht, den Empfang zu deaktivieren.







## EMPFANGS-TIMEOUT SETZEN - FUNKTION 11

"Empfangs-Timeout setzen" bestimmt die Wartezeit auf Daten.

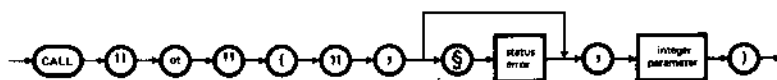


Abbildung 2-11 Empfangs-Timeout setzen

Wo

Syntaxelement	Bedeutung
11	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
ganzzahliger Parameter	Anzahl von 50 ms Takten als Wartezeit auf Daten

**Statusfehler:** A, P

### Merkmale

Der Wert im Ganzzahlparameter muß die folgende Konvention erfüllen:

- < 0 Kein Timeout. Empfangsfunktionen warten, bis Daten ankommen.
- = 0 Kein Timeout. Empfangsfunktionen kehren sofort mit Daten zurück, wenn diese bereitstehen, oder melden einen Timeout und "Empfangspuffer leer".
- > 0 Stehen keine Daten bereit, so wird diese Anzahl von Takten gewartet, bevor ein Timeout erklärt wird.

Standardwert: keiner



## SENDEPUFFER LÖSCHEN - FUNKTION 12

"Sendepuffer löschen" räumt den Sendepuffer.

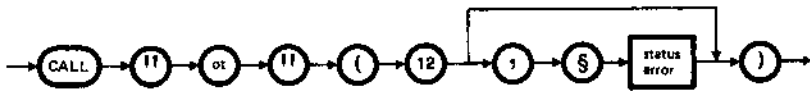


Abbildung 2-12 "Sendepuffer löschen"

Syntaxelement	Bedeutung
12	Funktionsnummer
Statusfehler	ein Stringparameter mit zurückgegebenen Statusanzeigen

### Merkmale

Alle Daten im Puffer gehen verloren. Diese Funktion löscht ebenso alle kommenden Rückmeldedaten.

## SENDEN DEAKTIVIEREN - FUNKTION 13

"Senden deaktivieren" schaltet den Sendeteil der Oliterm-Schnittstelle aus.

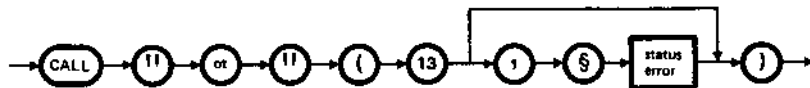


Abbildung 2-13 Senden deaktivieren



Wo

Syntaxelement	Bedeutung
13	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

Statusfehler: C, M, N, O

### Merkmale

Die Funktion "Senden deaktivieren" verändert den Zustand der Modemleitungen nicht und beeinflusst auch die Daten im Sendepuffer nicht. Die Funktionsaufrufe "Daten senden" oder explizit "Senden aktivieren" starten die Übertragung erneut.

Der Funktionsaufruf "Senden aktivieren" setzt einen von außen kommenden Aufruf "Senden deaktivieren" außer Kraft und das Anwendungsprogramm muß den Sendevorgang re-aktivieren. Der Status "N" zeigt an, daß der Sendevorgang vor dem Aufruf deaktiviert war.

### SENDEN AKTIVIEREN - FUNKTION 14

"Senden aktivieren" startet den Sendeteil, falls dieser deaktiviert ist.

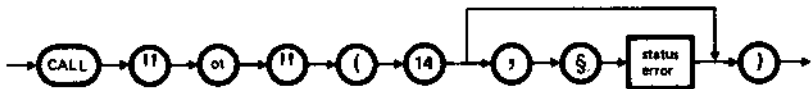
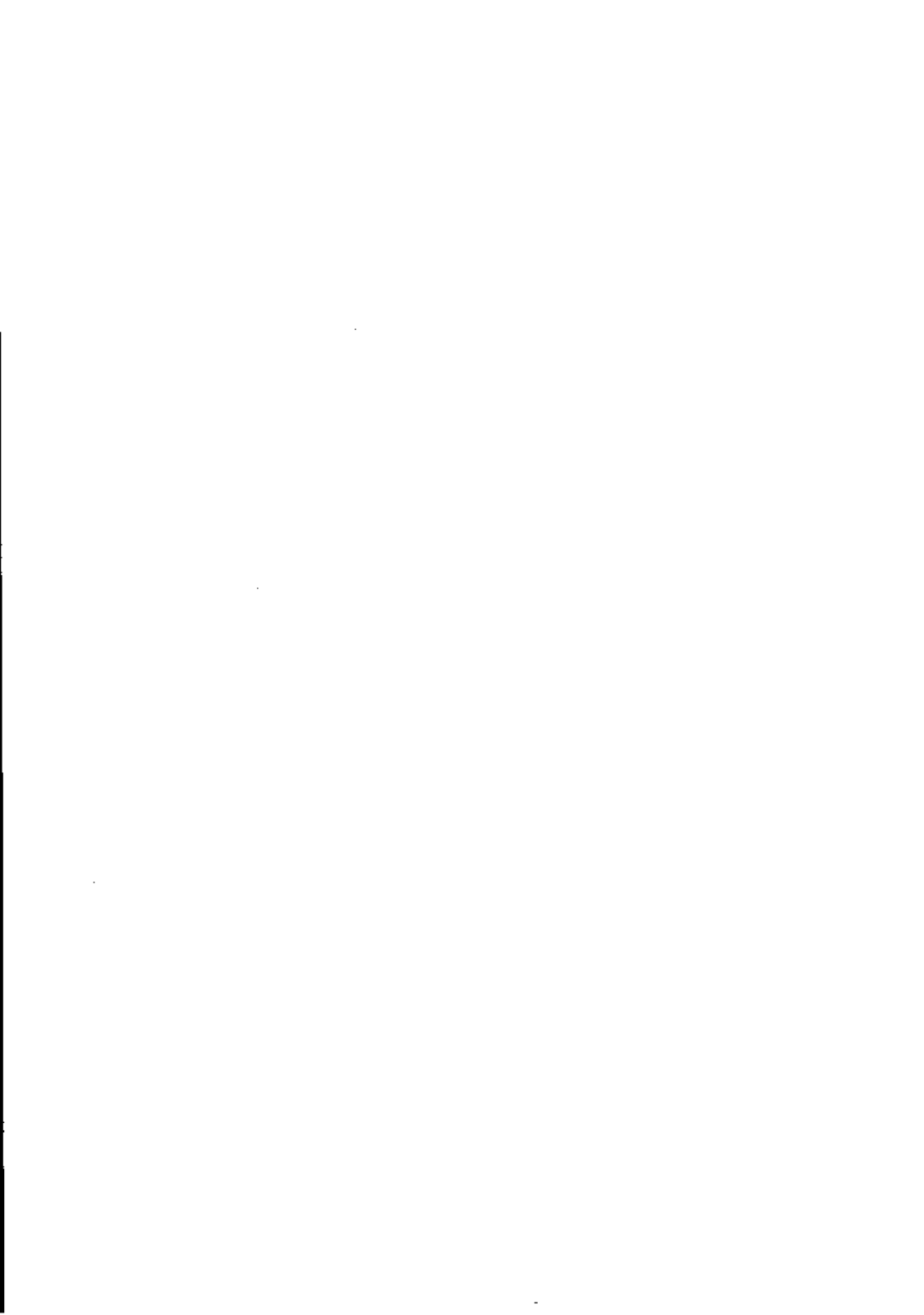


Abbildung 2-14 Senden aktivieren



Wo

Syntaxelement	Bedeutung
14	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

Statusfehler: C, M, 0

### Merkmale

"Senden aktivieren" startet den Sendevorgang und schaltet RTS ein. Dieser Aufruf setzt einen von außen kommenden Deaktivierungsbefehl außer Kraft.

Wenn DSR als DSR gebraucht, um das Senden von der Gegenstation aus zu steuern und DSR ausgeschaltet ist, wird die Schnittstelle wieder deaktiviert.

### Unterbrechung senden - Funktion 15

"Unterbrechung senden" sendet über die Datenleitung eine Unterbrechung.

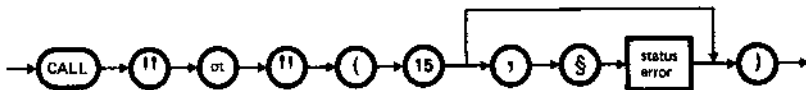
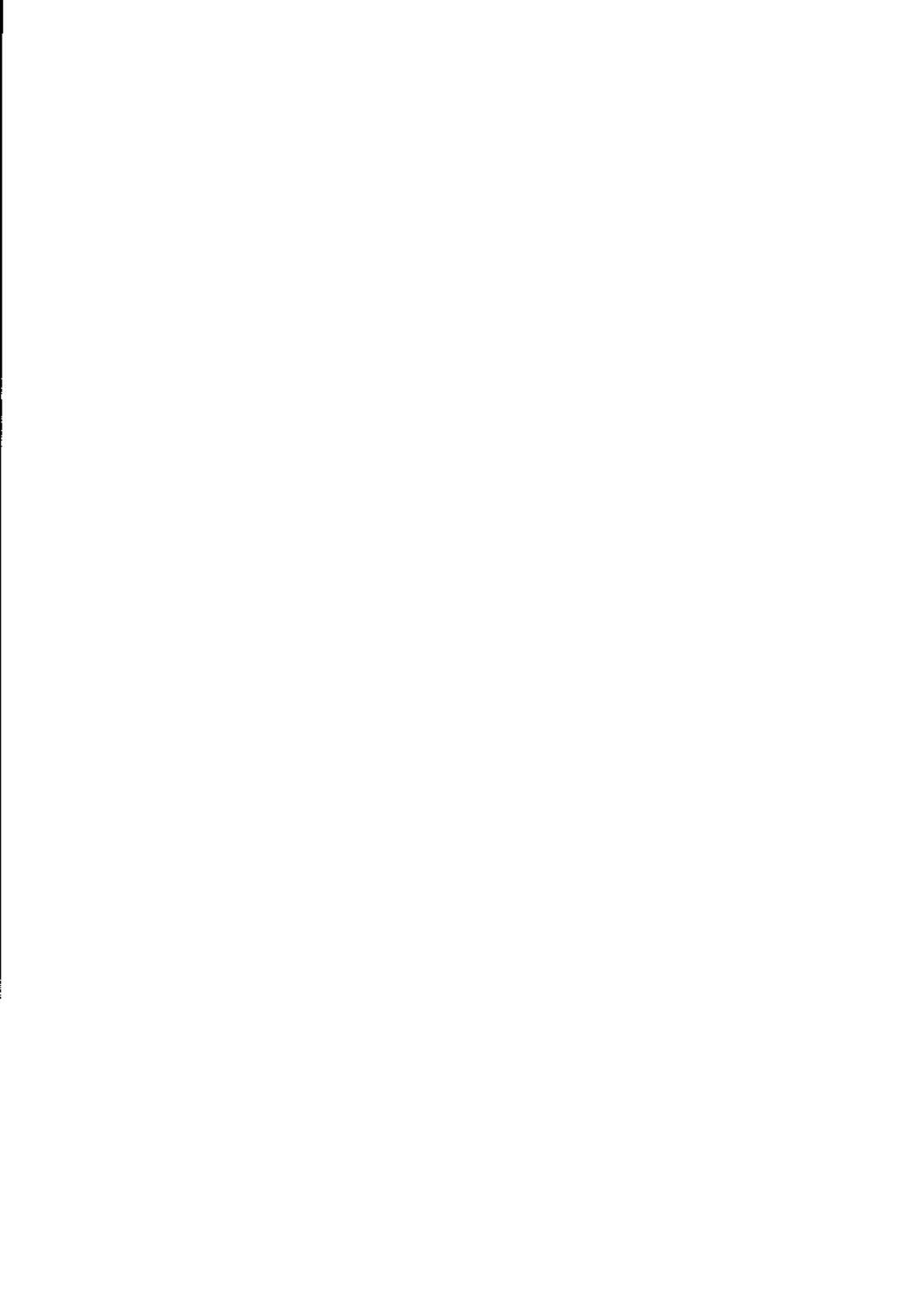


Abbildung 2-15 Unterbrechung senden



Wo

Syntaxelement	Bedeutung
15	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

Statusfehler: C, M, N, O

### Merkmale

Die Unterbrechung dauert zwischen 150 und 250 ms.

### RÜCKANTWORT SETZEN - FUNKTION 16

"Rückantwort setzen" setzt die Daten auf den zu übergebenden String.

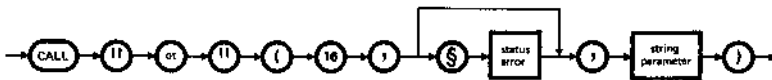
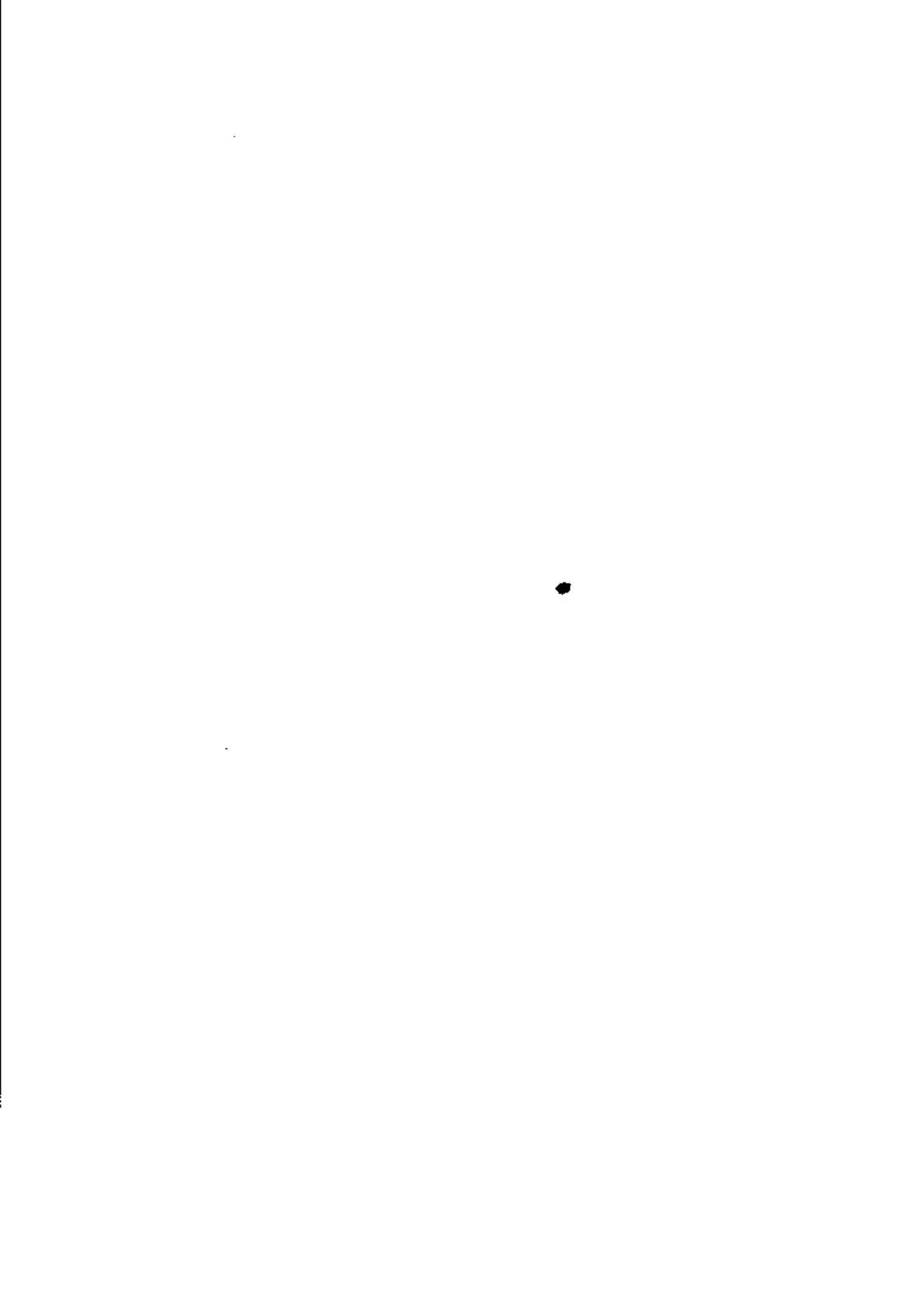


Abbildung 2-16 Rückantwort setzen

Wo

Syntaxelement	Bedeutung
16	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
Stringparameter	Rückantwortdaten



## Merkmale

Die Rückantwortdaten werden als Antwort auf den Empfang eines ENQ-Zeichens gesendet, aber nur, wenn sich Oliterm im Normalmodus befindet oder wenn das Anwendungsprogramm die Funktion "Rückantwortdaten senden" (Funktion 17) aufruft. Wird der Datenparameter übersprungen (Nullparameter) oder handelt es sich um einen Nullstring (Länge = 0), wird keine Rückantwort gesendet. Anders als die Funktion "Daten senden" (Funktion 6), wird die Rückantwort ohne angehängte Nullen oder EOM-Zeichen. Sind diese Zeichen Bestandteil der Daten, muß das Anwendungsprogramm sie in den String miteinbeziehen.

## RÜCKANTWORTDATEN SENDEN - FUNKTION 17

"Rückantwortdaten senden" bewirkt, daß die in Funktion 16 gesetzten Daten gesendet werden.

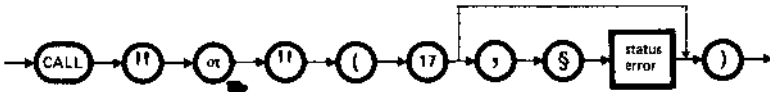


Abbildung 2-17 Rückantwortdaten

Syntaxelement	Bedeutung
17	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

Statusfehler: C, M, 0



## Merkmale

Die Rückantwortdaten haben eine höhere Priorität als normale Sendedaten. Sie werden vor allen anderen Daten in den Sendepuffer gesendet. Das Anwendungsprogramm sollte daher vor Aufruf dieser Funktion darauf achten, daß der Sendepuffer leer bzw. gelöscht ist. Wenn Olitem laufend eine Rückantwort sendet, so startet diese Funktion keinen anderen Sendevorgang.

## Anmerkung

"Rückantwortdaten senden" setzt "Senden deaktivieren" außer Kraft, ganz gleich, ob der Deaktivierungsbefehl vom Programm oder von einer anderen Station stammt.

## SENDE-TIMEOUT SETZEN - FUNKTION 18

"Sende-Timeout setzen" setzt eine Mindestwartezeit im Sendepuffer.

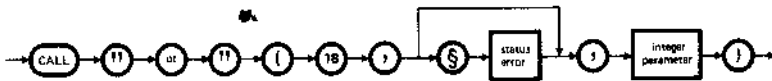
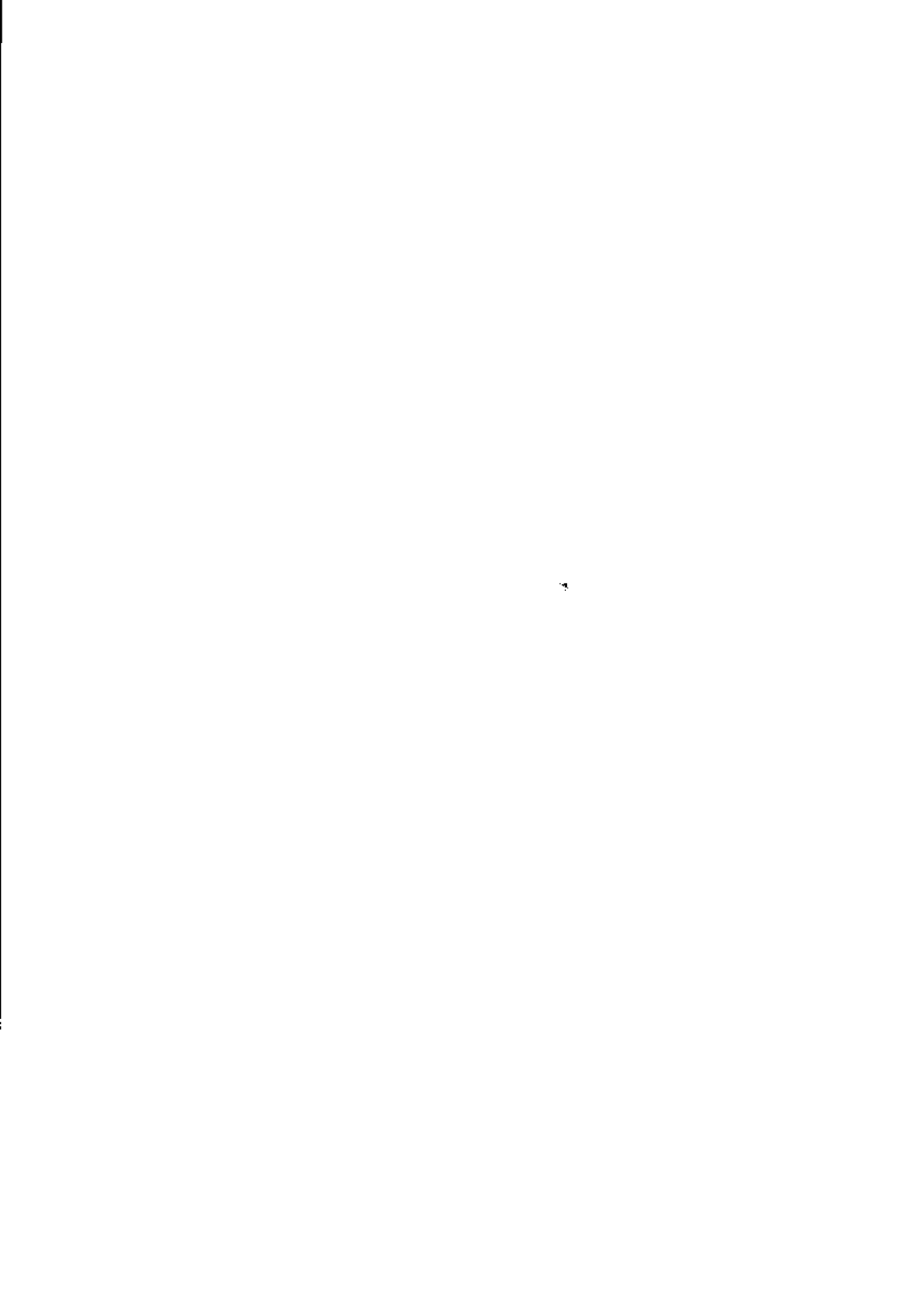


Abbildung 2-18 Sende-Timeout setzen.

Wo

Syntaxelement	Bedeutung
18	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
ganzzahliger Parameter	das Vielfache der 50 ms Takte gibt die Wartezeit im Sendepuffer an.



Statusfehler: A, P

### Merkmale

Der ganzzahlige Parameter muß die folgenden Konventionen erfüllen:

- < 0 Kein Timeout: Sendefunktion wartet ab, bis der Sendepuffer frei ist.
- = 0 Kein Timeout: wenn der Sendepuffer nicht frei ist, kehren die Sendefunktionen sofort zurück und melden einen Timeout. die Daten wurden nicht gesendet.
- > 0 wartet diese Anzahl an 50 ms Takten auf Platz, bevor er einen Timeout vereinbart.

### Kontrollzeichen setzen - Funktion 19

"Kontrollzeichen setzen" erlaubt dem Anwendungsprogramm eine zugeschnittene Verwendung der Empfangsfunktion der Datensteuerung.

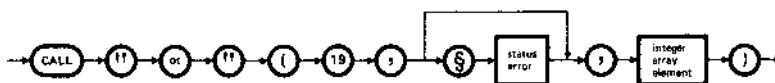


Abbildung 2-19 Datenkontrollzeichen setzen

Wo

Syntaxelement	Bedeutung
19	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen
Integer Array Element	das erste Element eines ganzzahligen Datenfeldes der Dimension 32. Es kennzeichnet die Kontrollzeichen, die die Funktion "Daten empfangen" nicht von hineinkommenden Datenbytes entfernen sollte.



## Merkmale

Der Wert für jedes Element des Datenfeldes ist entweder 0 (Kontrollzeichen nicht miteinbezogen) oder 1 (entsprechendes Kontrollzeichen miteinbezogen).

In einigen Fällen ist es wünschenswert, daß die meisten Kontrollzeichen miteinbezogen werden wie bei einem einfachen Terminal-Programm, das bei Anwendung der Funktion "Daten empfangen" (Funktion 4) schneller arbeitet als bei Anwendung von "Byte empfangen" (Funktion 3).

## Beispiel

Anzeige	Anmerkungen
10 DIM c % (32) 20 C % (13) = 1 30 C % (10) = 1 40 CALL "ot" (19, \$\$, C % (0)) . . .	dieses Beispiel zeigt, wie die ASCII Zeichen CR/LF innerhalb der dem BASIC-Programm übergebenen Daten einbezogen werden. Alle anderen Kontrollzeichen werden übersprungen.

## Standard

Schließt TAB (9) und Form Feed (12) ein. Alle anderen Kontrollzeichen werden übersprungen.

## Schnittstelle ausschalten - Funktion 20

"Schnittstelle ausschalten" schaltet die Modemsteuerleitungen RTS und DTR aus.

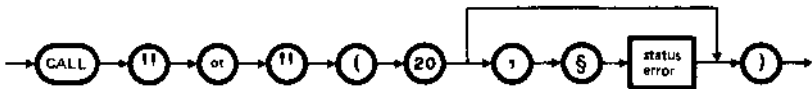
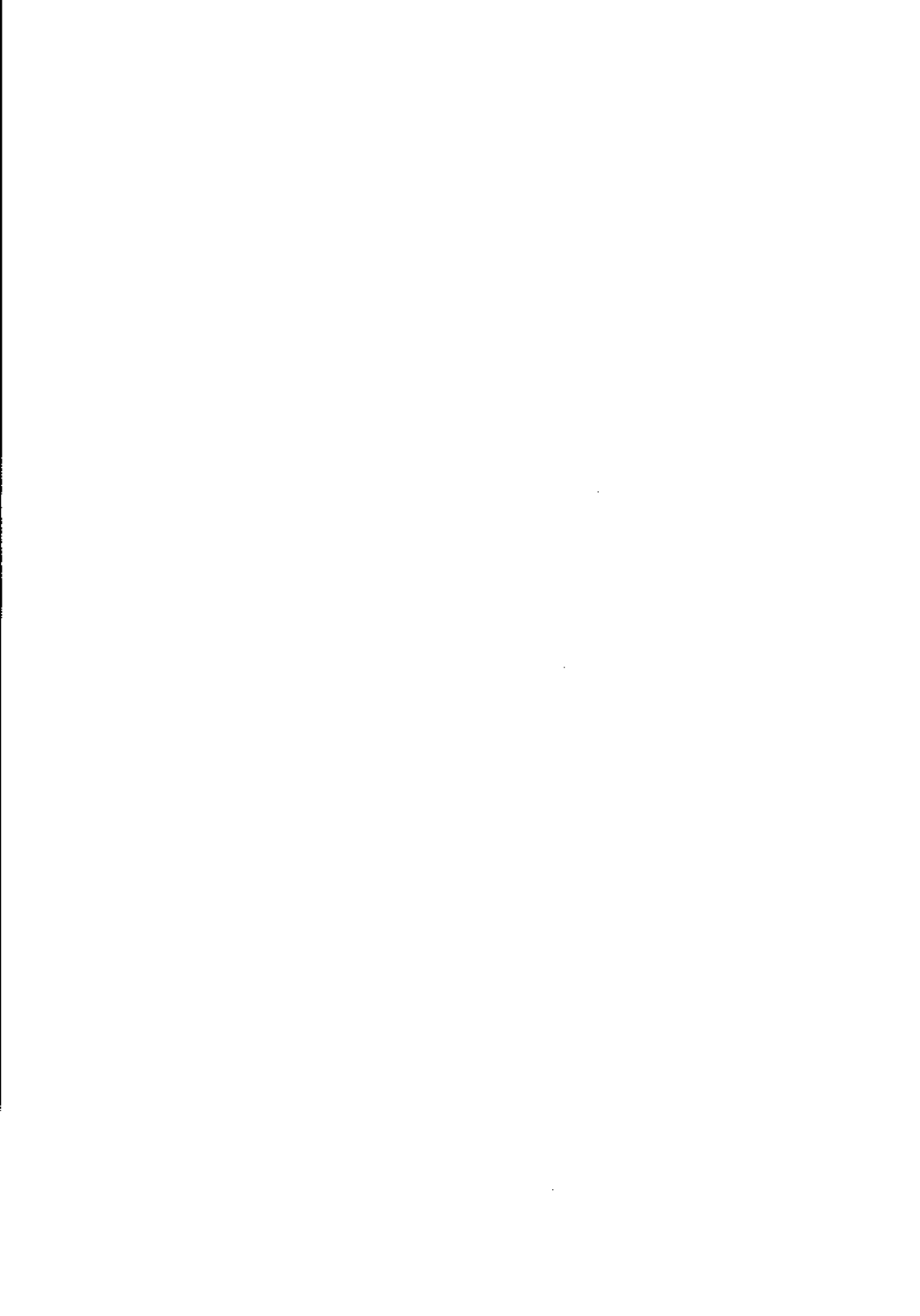


Abbildung 2-20 Schnittstelle ausschalten



Wo

Syntaxelement	Bedeutung
20	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

### Merkmale

Dem Modem wird signalisiert, daß das Programm nicht übertragen kann bzw. ein Abschnitt beendet ist und eine Unterbrechung vorliegt. Die Funktion löscht sowohl Empfangs- als auch Sendepuffer. XMIT-Daten gehen verloren.

Nach Anwendung der Funktion "Schnittstelle ausschalten" muß die Funktion "Schnittstelle einschalten" aufgerufen werden, um das Modem zu reaktivieren. Die Funktion "Schnittstelle ausschalten" läuft unabhängig von Empfangssteuerungsmodus 2, der DTR anwendet.

### Schnittstelle einschalten - Funktion 21

"Schnittstelle einschalten" schaltet die Modemsteuerungsleitungen RTS und DTR ein.

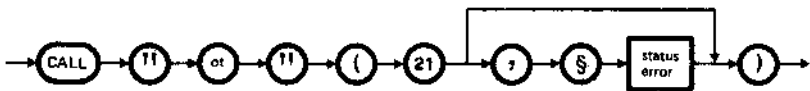
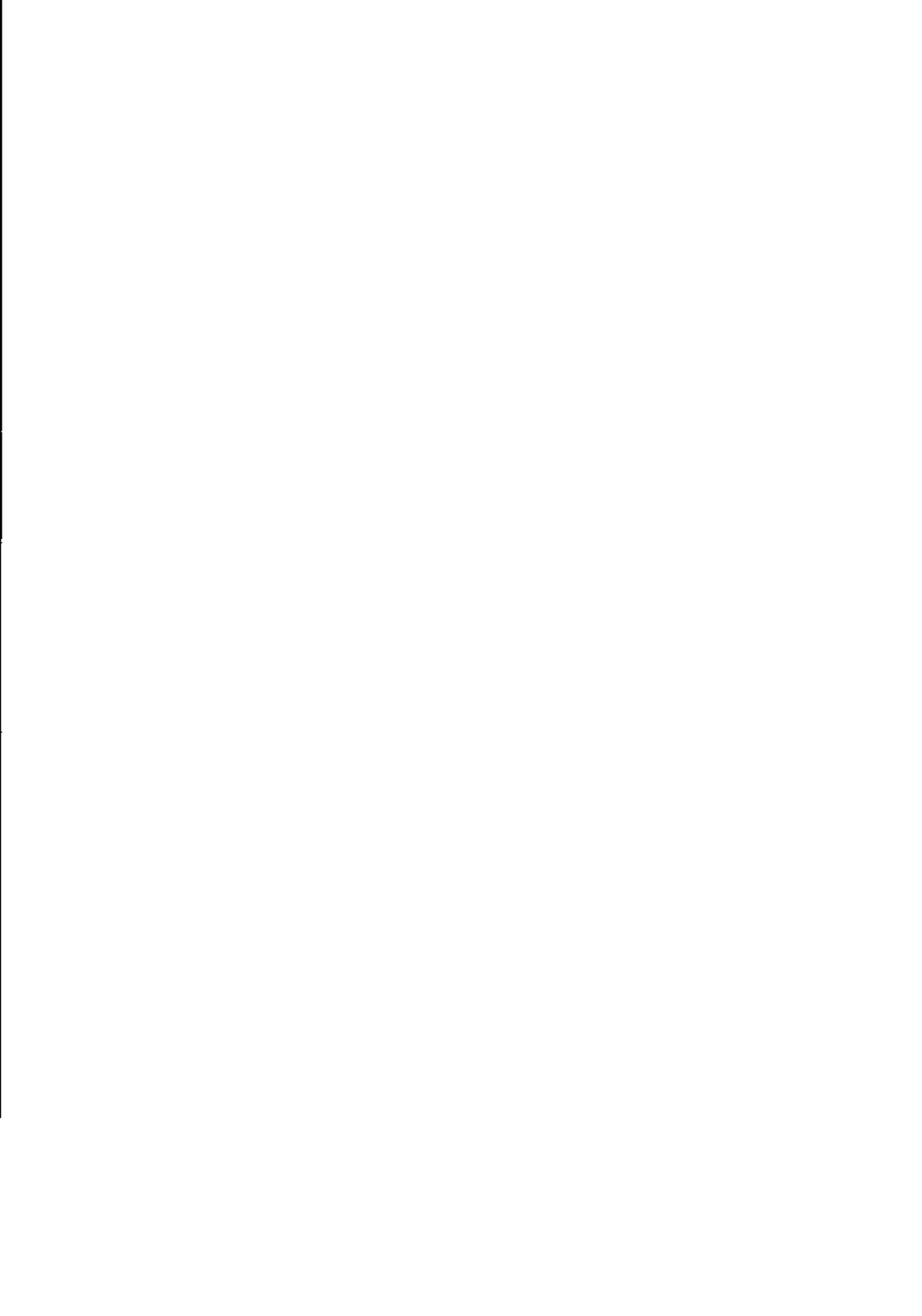


Abbildung 2-21 Schnittstelle einschalten



Wo

Syntaxelement	Bedeutung
21	Funktionsnummer
Statusfehler	ein Stringparameter für zurückgegebene Statusanzeigen

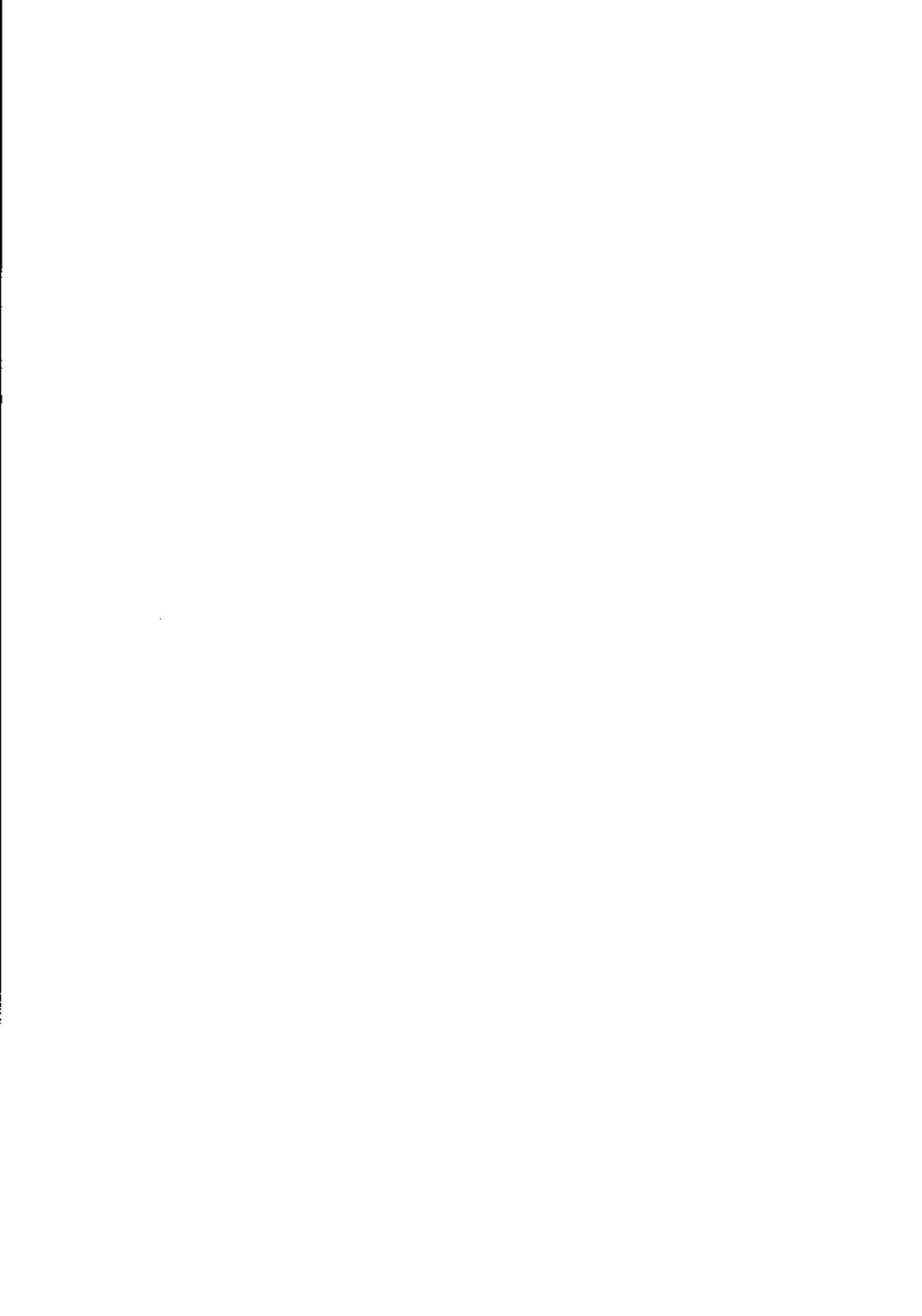
### Merkmale

Dem Modem wird signalisiert, daß das Programm bereit ist, die Übertragung zu starten.

Diese Funktion betrifft nur die Steuerungsleitungen. Leitungsrauschen kann den Empfang von fehlerhaften Daten bewirken. Der Anwender sollte dies ignorieren, bis eine Verbindung fertiggestellt worden ist.



### **3. ANWENDUNGSBEISPIELE**



## Zum Kapitel:

Dieses Kapitel zeigt drei nützliche Beispiele von BASIC-Programmen, die die OIterm-Funktionen aufrufen.

Der Anwender, der BASIC-Programme mit OIterm-Funktionen durchführen will, muß Hardware- und Softwarevoraussetzungen überprüfen. Dies wurde bereits in Kapitel 1 beschrieben.

## Inhalt

<u>Ein Programm zum Initialisieren, zum Senden von Byte und zum Empfang von Daten</u>	3-1
<u>Ein Programm zum Senden einer Datei zu einem anderen M20</u>	3-3
<u>Ein Programm zum Empfang einer Datei von einem anderen M20</u>	3-5



Ein Programm zum Initialisieren, zum Senden von Byte und zum Empfang von Daten

Das Programm besteht aus 2 Hauptteilen:

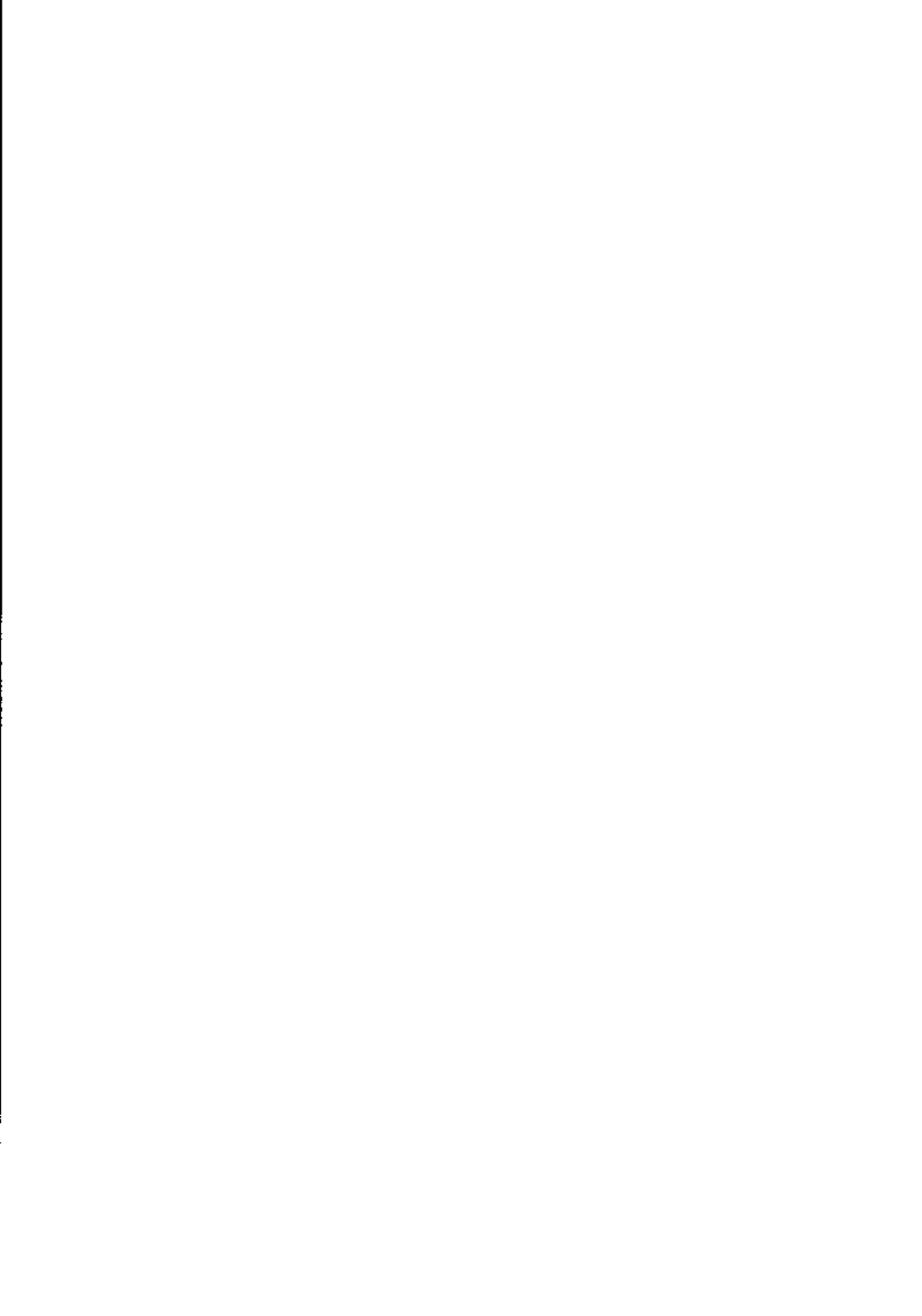
- Initialisierung von M20 und Modem (Zeilen 100 - 460)
- Datenverarbeitung (Zeilen 470 - 640)

Die Initialisierungsverarbeitung besteht aus 3 Abschnitten:

- Anpassung von Oliterm an entsprechende Terminal-Zeichen (Zeilen 140 - 310)
- Festlegung von denjenigen Kontrollzeichen, die zum Bildschirm durchgegeben werden (Zeilen 320 - 380)
- letzte Vorbereitungen für die Datenverarbeitung (siehe Anmerkungen zu Zeilen 390 - 460)

Der Datenverarbeitungsabschnitt wird in 2 weitere Teile gegliedert:

- Tastaturverarbeitung (Zeile 470-560), wobei jedes per Tastatur eingegebene Zeichen zur seriellen Schnittstelle geleitet wird.
- Empfangsprozedur der seriellen Schnittstelle (Zeile 570-640), die den Text und ausgewählte Steuerzeichen zum Bildschirm sendet.

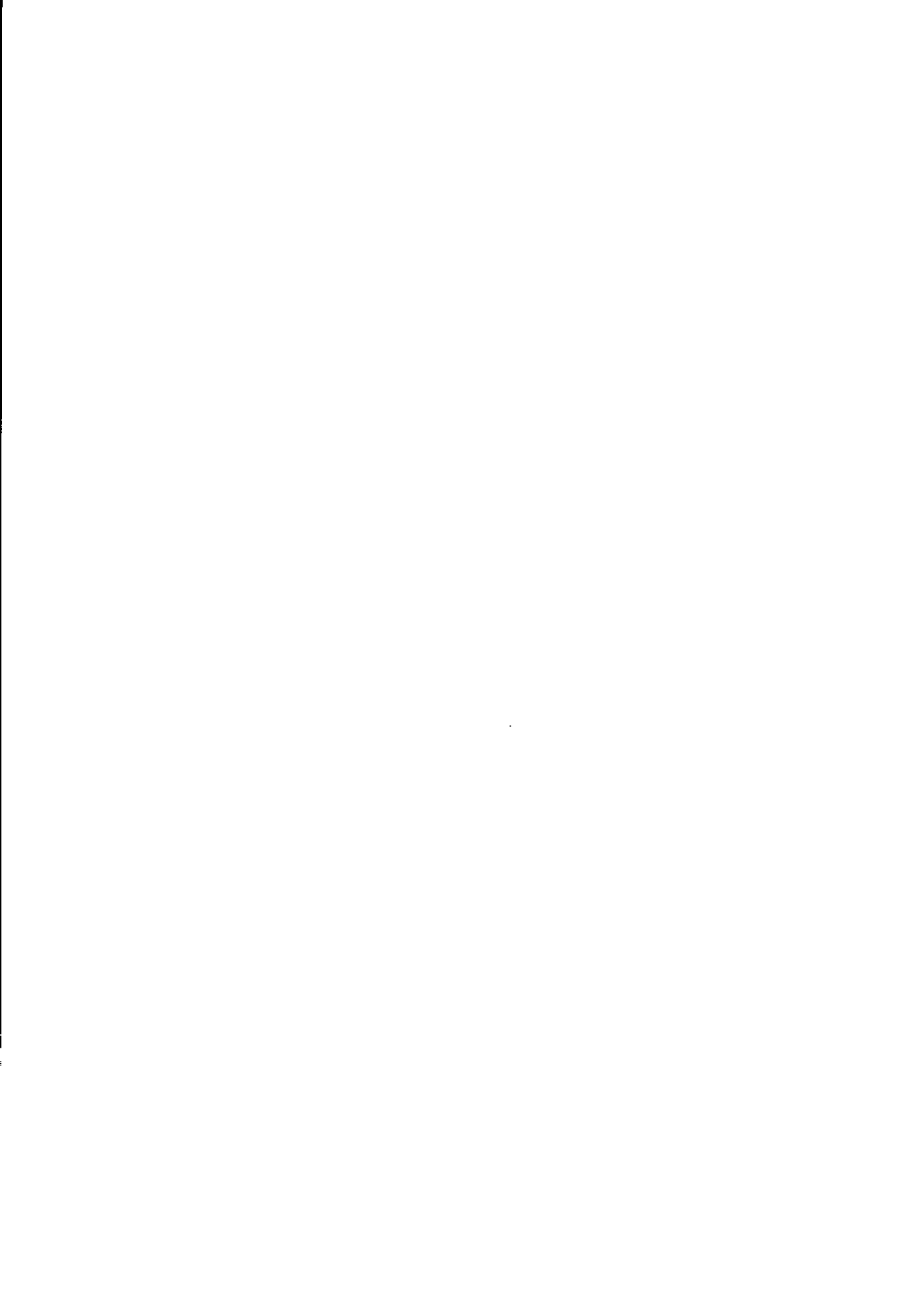


```

10 REM
20 REM ** program to initialise, transmit byte and receive message
30 REM
100 DEFINT A-Z
110 DIM I(40)
120 CS = "ot"
130 PRINT CHR$(12); "[initialising line parameters]"
140 I(0) = 0 'parity = none
150 I(1) = 8 'data bits = 8
160 I(2) = 0 'stop bits = 1
170 I(3) = 3 'baud rate = 1200
180 I(4) = 1 'receive ctrl = x-on/x-off
190 I(5) = 1 'xmit ctrl = x-on/x-off
200 I(6) = 1 'receive mode = normal
210 I(7) = 13 'receive EOM = CR
220 I(8) = 4 'receive EOF = EOF
230 I(9) = 127 'receive data mask = 01111111B
240 I(10) = 80 'max. message length
250 I(11) = 13 'XMIT 1st EOM = CR
260 I(12) = 10 'XMIT 2nd EOM = LF
270 I(13) = 0 'append NO nuls to message
280 SS = SPACES(20)
290 CALL CS(1,@SS,@I(0)) 'initialise 01term
300 IF LEN(SS) = 0 THEN GOTO 320
310 PRINT "**** initialisation error: ";SS; STOP
320 FOR J = 0 TO 31 : I(J)=0 :NEXT J
330 I(8) = 1 'Back Space
340 I(9) = 1 'Horizontal Tab.
350 I(10) = 1 'Line Feed
360 I(12) = 1 'Form Feed
370 I(13) = 1 'Carriage Return
380 CALL CS(19,,I(0)) 'set control characters
390 CALL CS(20) 'turn off DTR and RTS
400 CALL CS(11,,1) 'set receive timeout
410 CALL CS(18,,30) 'set transmit timeout
420 CALL CS(21) 'turn on RTS and DTR
430 PRINT:PRINT "initialisation complete - make phone connection now"
440 CALL CS(7,@SS) 'get status
450 IF INSTR(SS,"C") THEN 440 'wait for DSR
460 CALL CS(8) 'flush receive buffer
470 REM
480 REM process keyboard input
490 REM
500 ES = INKEY$
510 IF LEN(ES) = 0 THEN GOTO 600
520 SS = SPACES(20)
530 CALL CS(5,@SS,ASC(ES)) 'write byte to port
540 IF INSTR(SS,"A") = 0 THEN GOTO 600
550 PRINT "**** write error: ";SS
560 GOTO 600
570 REM
580 REM process serial port input
590 REM
600 SS = SPACES(20)
610 MS = SPACES(80)
620 CALL CS(4,@SS,@MS) 'read message from port
630 PRINT MS;
640 GOTO 500

```

Abbildung 3-1 Programm zum Initialisieren, zum Senden von Byte und Empfangen von Daten.



## Ein Programm zum Senden einer Datei zu einem anderen M20

Das Programm (Abbildung 3-2) sendet eine Datei von der seriellen Schnittstelle aus, wobei jeder logische Satz einzeln gesendet wird.

Das Programm erwartet für jeden Satz ein Echo des Satzes.

Entspricht das Echo dem vorher gesendeten Satz, wird der nächste Satz gesendet. Der Zeichenzählerstand wird ebenfalls gespeichert und wird mit dem empfangenen Zeichenzählerstand verglichen, nachdem der letzte Satz erfolgreich rückgemeldet worden ist.

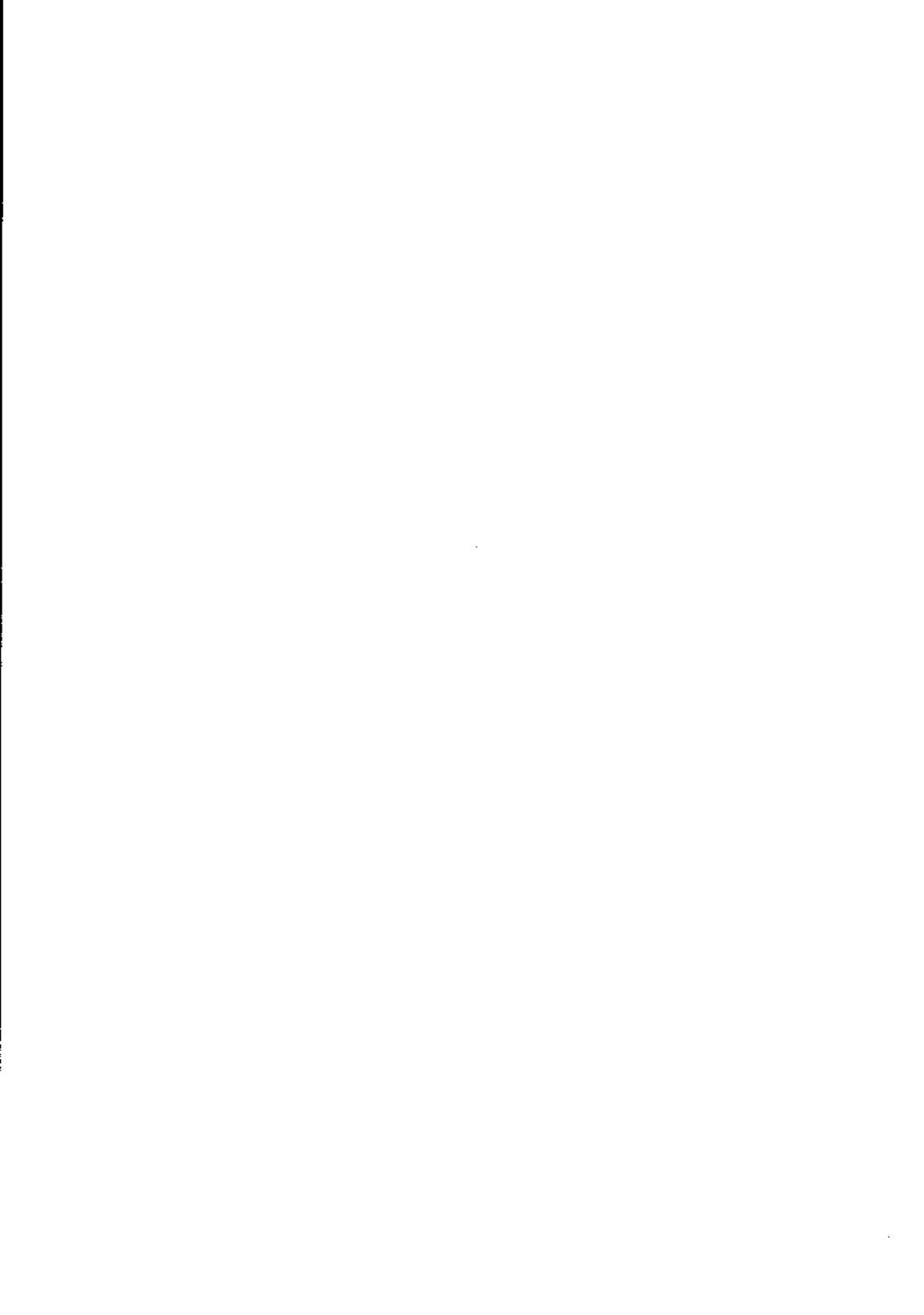
Diese Techniken sollen eine fehlerfreie Dateiübertragung garantieren. Wird ein Rückmelde- oder Zählungsfehler entdeckt, so werden keine Korrekturversuche unternommen.

Das Programm besteht aus einem Haupt- (Zeile 100 - 390) und einem Unterprogramm (Zeilen 400 - 530). Das Hauptprogramm:

- initialisiert Oliterm mit Standardparametern (Zeilen 150 - 160);
- setzt das Empfangs-Timeout auf 50 Sekunden fest (Zeile 180);
- schaltet die Schnittstelle ein, setzt RTS- und DTR-Signale hoch (Zeile 190);
- veranlaßt den Benutzer, Eingriffe vorzunehmen (Zeile 200 - 210);
- eröffnet die zu sendende Datei (Zeile 220);
- sendet den Dateinamen aus der seriellen Schnittstelle (Zeilen 240 - 250);
- sendet alle Dateisätze aus der seriellen Schnittstelle (Zeilen 260 - 300);
- sendet ein EOF-Zeichen (Zeile 310);
- empfängt und vergleicht den Dateizählerstand (Zeilen 320 - 350);
- schließt Datei und Oliterm ab (Zeilen 360 - 370)

Das Unterprogramm:

- sendet einen String aus der Schnittstelle (Zeilen 430 - 440);
- empfängt den rückgemeldeten String von der Schnittstelle (Zeilen 450 - 480);
- vergleicht gesendeten und rückgemeldeten String miteinander (Zeilen 490 - 530)

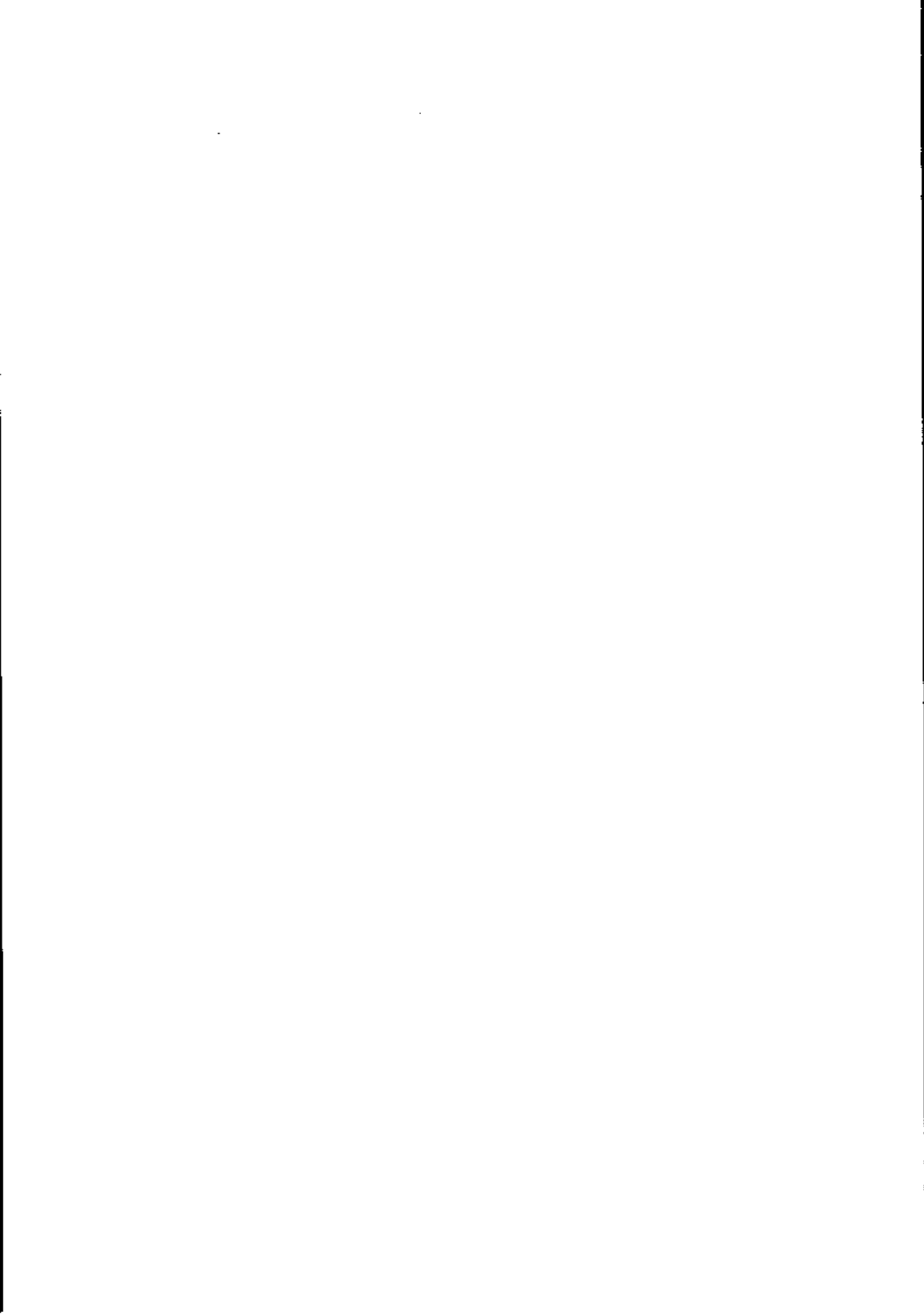


```

10 REM
20 REM ***** Program to send a file to another M20
30 REM
100 DEFINT A-Z
110 CS = "ot"
120 PRINT CHR$(12)
130 PRINT "M20 to M20 file-transfer send program"
140 PRINT
150 SS = SPACES(20)
160 CALL CS (1,@SS)
170 IF LEN(SS)<>0 THEN PRINT "initialisation error: ";SS : STOP
180 CALL CS (11,1000)
190 CALL CS (21)
200 PRINT "start program 'fr' on partner M20 now"
210 INPUT "when program 'fr' ready, enter name of file to be transferred",
    TFNS
220 OPEN "i",1,TFNS
230 FL% = 0
240 WSS = TFNS
250 GOSUB 400
260 WHILE NOT EOF (1)
270   LINE INPUT #1, WSS
280   FL% = FL% + LEN (WSS)
290   GOSUB 400
300 WEND
310 CALL CS (5,,4)
320 RSS = SPACES(20)
330 CALL CS (4,@RSS)
340 IF VAL (RSS) <> FL% THEN PRINT "file length error" : STOP
350 PRINT "file transfer completed successfully"
360 CLOSE
370 CALL CS (2)
380 PRINT "**** END of program ****"
390 END
400 REM
410 REM           routine to send record, receive echo and compare
420 REM
430 CALL CS (6,@SS,WSS)
440 IF SS<>" " THEN PRINT "write status = " + SS : STOP
450 RSS = SPACES(9)
460 SS = SPACES(20)
470 CALL CS (4,@SS, RSS)
480 IF SS<>" " THEN PRINT "read status = " + SS : STOP
490 IF RSS = WSS THEN RETURN
500 PRINT "echo error"
510 PRINT "ws =>" + WSS + " (len = " + STR$(LEN(WSS)) + ")"
520 PRINT "rs =>" + RSS + " (len = " + STR$(LEN(RSS)) + ")"
530 STOP

```

Abbildung 3-2 Programm zum Senden einer Datei zu einem anderen M20



## Ein Programm zum Empfang einer Datei von einem anderen M20

Das Programm (3-3) sendet jeden Satz, den es von der Schnittstelle empfängt (außer dem ersten) zu einer Datei.

Der erste Satz enthält den Dateinamen. Wie beim Senden einer Datei wird ein Zeichenzählerstand gespeichert, der aber diesmal von der Schnittstelle aus nach Empfang einer EOF-Anzeige gesendet wird. Jeder empfangene Satz wird rückgemeldet, indem er sofort von der Schnittstelle aus gesendet wird.

Das Programm besteht aus einem Haupt- (Zeilen 100 - 410) und zwei Unterprogrammen (Zeilen 420 - 520 und 530 - 600).

- initialisiert Oliterm mit Standardparametern (Zeilen 150 - 170);
- setzt die Grenze des Empfangs-Timeout auf 50 Sekunden fest (Zeile 180);
- schaltet RTS und DTR hoch (Zeile 190);
- veranlaßt den Benutzer, Eingriffe vorzunehmen (Zeile 200);
- empfängt den Dateinamen von der Schnittstelle, öffnet die Datei und sendet den Dateinamen zurück (Zeilen 240 - 340);
- empfängt, sendet und schreibt jeden Satz in die Datei, bis ein EOF-Zeichen empfangen wird (Zeilen 240 - 340)
- sendet den gespeicherten Zählerstand aus der Schnittstelle (Zeilen 350 - 360), und
- schließt Datei und Oliterm (Zeilen 370 - 380) ab.

Die Funktionen "Empfang Deaktivieren" (Zeile 300) und "Empfang Aktivieren" (Zeile 320) sind notwendig, um den Verlust von Zeichen, die von der Schnittstelle gesendet werden, während die Diskette tätig ist, zu vermeiden.

Das erste Unterprogramm empfängt einen Satz von der Schnittstelle (Zeilen 420 - 520).

Das zweite Unterprogramm sendet einen Satz von der Schnittstelle aus (Zeilen 530 - 600).

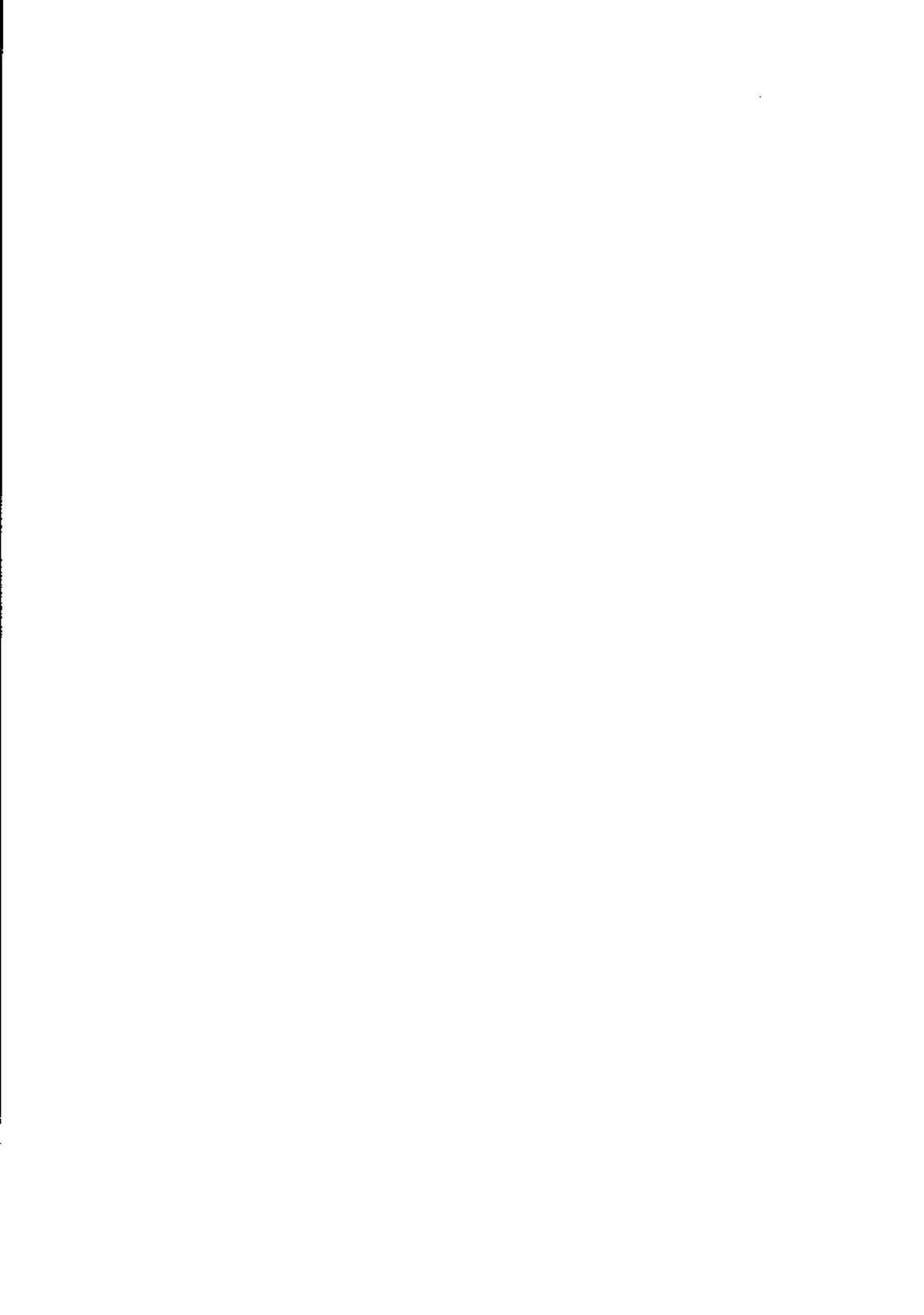


```

10 REM
20 REM ***** Program to receive a file from another M20
30 REM
100 DEFINT A-Z
110 C$ = "et"
120 PRINT CHR$(12)
130 PRINT "M20 to M20 file-transfer receive program"
140 PRINT
150 S$ = SPACES(20)
160 CALL CS (1,055)
170 IF LEN(S$)<>0 THEN PRINT "initialisation error: ";S$ : STOP
180 CALL CS (11,,1000)
190 CALL CS (2)
200 PRINT "file receive program ready - continue file send program now"
210 GOSUB 420
220 OPEN "o",1,RSS
230 GOSUB 530
240 FL% = 0
250 EF% = 0
260 WHILE EF% = 0
270   GOSUB 420
280   IF EF% = 1 THEN GOTO 340
290   GOSUB 530
300   CALL CS (9)
310   PRINT#1, RSS
320   CALL CS (10)
330   FL% = FL% + LEN(RSS)
340   WEND
350   RSS = STR$(FL%)
360   GOSUB 530
370   CLOSE
380   CALL CS (2)
390   PRINT "file transfer completed successfully"
400   PRINT "***** END of PROGRAM *****"
410   END
420 REM
430 REM           routine to get next record
440 REM
450   RSS = SPACES(90)
460   S$ = SPACES(20)
470   CALL CS (4,@S$,@RSS)
480   IF S$ = "" THEN GOTO 520
490   IF S$ = "K" THEN EF% = 1 : GOTO 520
500   PRINT "read status = " + S$
510   STOP
520 RETURN
530 REM
540 REM           routine to send a record
550 REM
560   CALL CS (6,@S$,RSS)
570   IF S$ = "" THEN GOTO 600
580   PRINT "write status = " + S$
590   STOP
600 RETURN

```

Abbildung 3-3 Programm zum Empfang einer Datei von einem anderen M20



## **A. FUNKTIONEN UND IHRE STATUSCODES**



## Zum Anhang

Dieser Anhang gibt Aufschluß über die Bedeutung des Statuscodes, den die 21 Funktionen im Kapitel 2 generieren. Die vier Codes G, H, I und J der Empfangsfunktion variieren in ihrer Bedeutung. Die verschiedenen Bedeutungen sind am Ende der Tabelle A-1 beschrieben.

### Inhalt

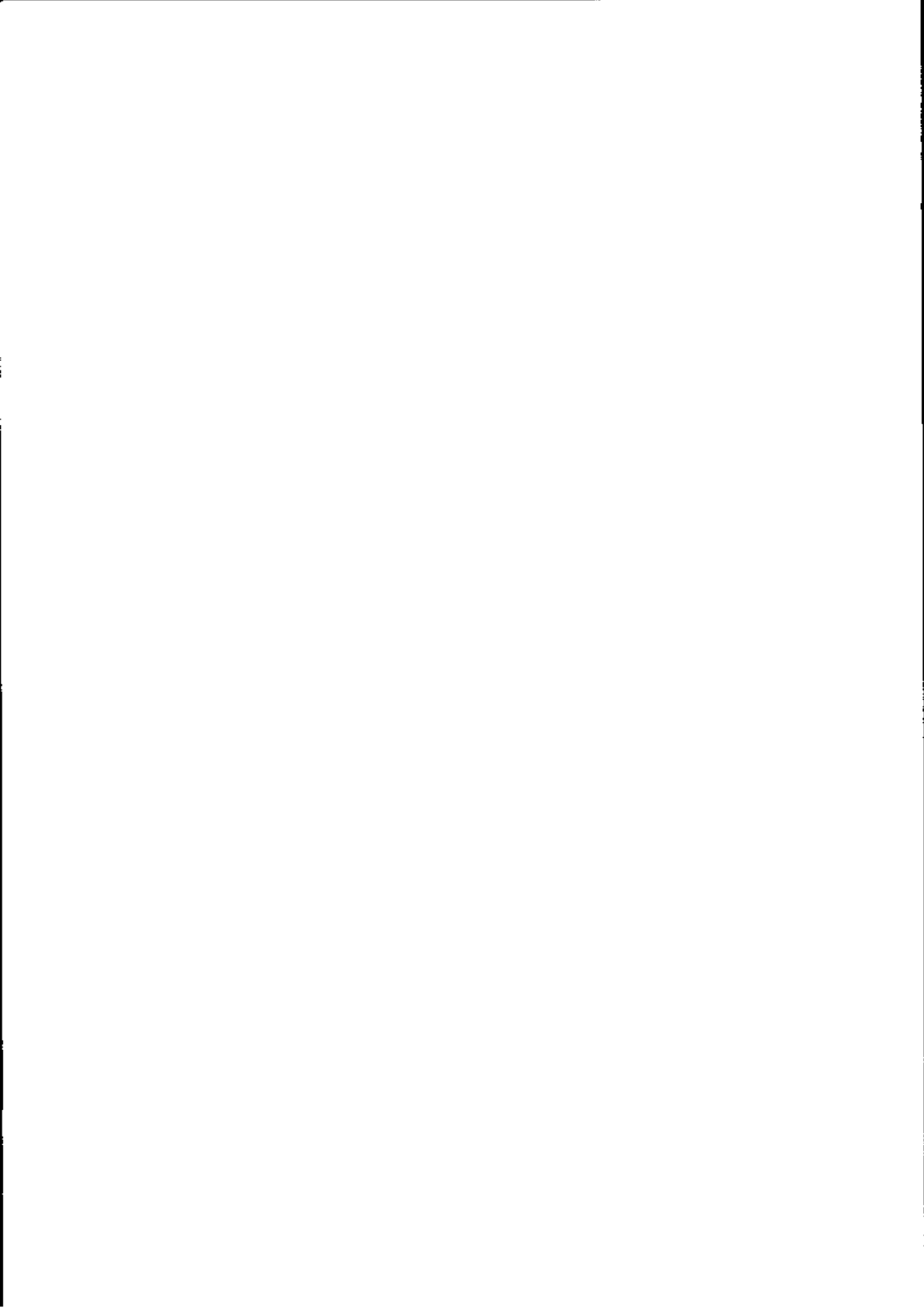
<u>Fehlercodes</u>	A-1
Datenempfangsfehler	A-3
<u>Status zurückgegeben</u>	A-4



## Fehlercodes

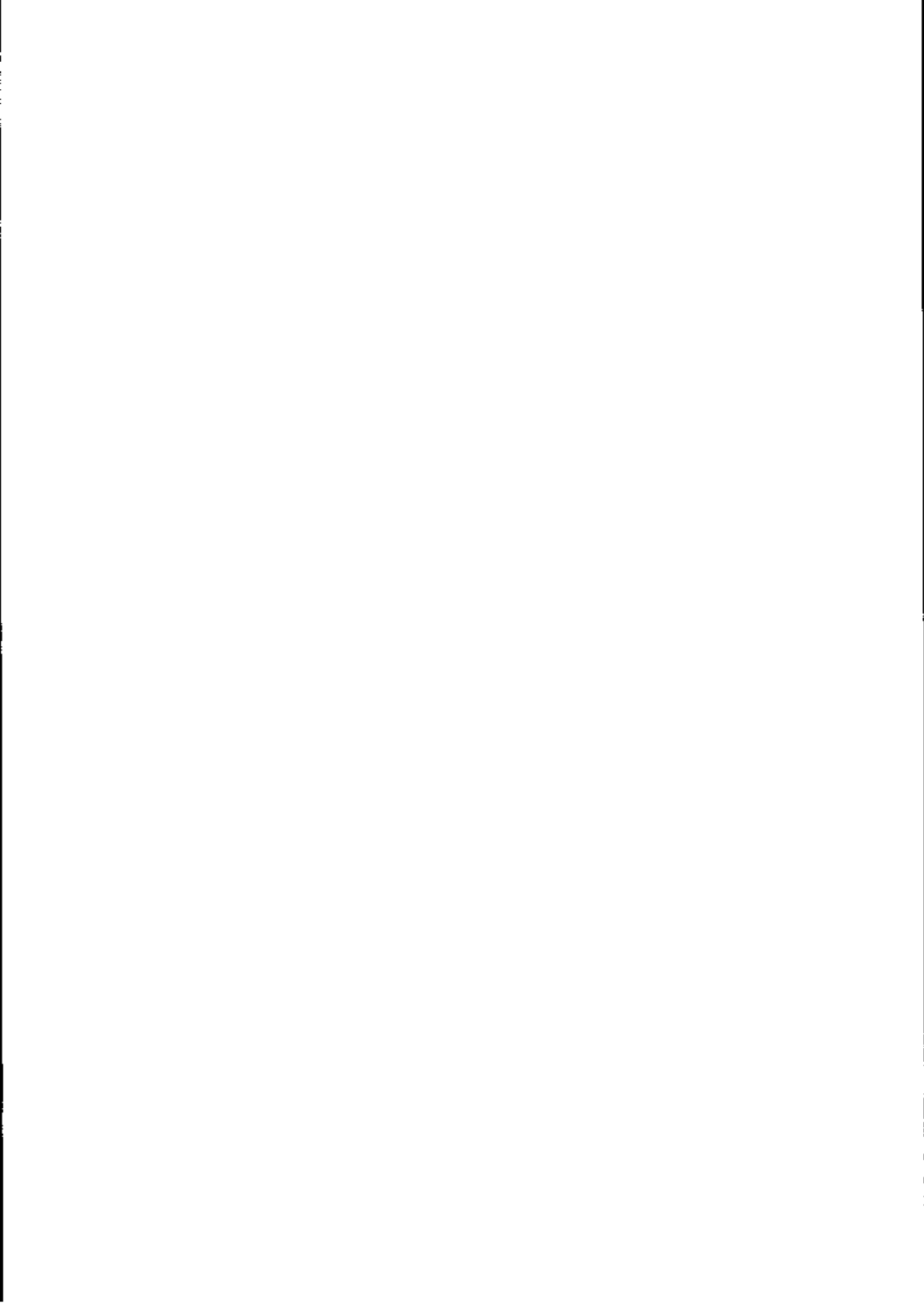
Status	Name	Bedeutung
A	Fehlerausgang	Wird gesetzt, wenn eine Funktion ihre Aufgabe nicht erfüllen kann. Wird immer in Verbindung mit anderen Fehlern wie z. B. "D", Timeout gesetzt. Dieser Buchstabe kann nach dem Datentransfer getestet werden, um den Erfolg zu prüfen.
B	Oliterm Abgeschlossen	Wird gesetzt, wenn Funktion 1, "Initialisierung" nicht vorher aufgerufen oder Funktion 2 "Abschließen" aufgerufen wurde. Vor der Initialisierung können keine Funktionsaufrufe gemacht werden.
C	DSR Nicht Hochgeschaltet	Der USART 8251 gibt den Zustand des Modems an. Falls gesetzt, ist DSR nicht hochgeschaltet. Der Status kann verwendet werden, um den Zustand eines Modems zu bestimmen und er kann, falls DSR zur Übermittlungssteuerung verwendet wird, angeben, warum der Sendepuffer voll ist.
D	Time Out	Wird gesetzt, wenn Oliterm auf Empfangsdaten wartet oder Daten zu einem vollen Sendepuffer senden will.
E	Empfangspuffer Leer	Wird gesetzt, wenn der Empfangspuffer leer ist. Es sind keine Daten verfügbar.
F	Empfang Deaktivieren	Wird gesetzt, wenn der Empfang die Fernübertragung deaktiviert hat. Dies kann dadurch bedingt sein, daß der Empfangspuffer zu mehr als 75 % gefüllt ist. Dieser Status bedeutet nicht, daß der USART 8251 deaktiviert wird, sondern nur, daß Oliterm versucht hat, die andere Station zum Beenden des Sendevorgangs zu veranlassen. Obgleich dieser Status gesetzt ist, können Daten noch empfangen werden.

Tabelle A-1 Fehlercodes



Status	Name	Bedeutung
G	Empfangs-Unterbrechung	Wird gesetzt, wenn eine Unterbrechung gesetzt wurde (siehe unten).
H	Empfangs-Overrun	Wird gesetzt, wenn der USART einen Overrun-Fehler berichtet hat (siehe unten).
I	Empfangs-Parityfehler	Wird gesetzt, wenn der USART einen Parityfehler berichtet hat (siehe unten).
J	Überlauf des Empfangspuffers	Wird gesetzt, wenn Oliterm Daten wegen Überfüllung des Empfangspuffers verloren hat (siehe unten).
K	Ende der Datei	Wird gesetzt, wenn die Datensteuerung ein EOF-Zeichen empfangen hat (siehe "Daten Empfangen").
L	Empfangsdaten-String Voll	Wird gesetzt, wenn die Empfangsdaten den String gefüllt haben (siehe "Daten Empfangen").
M	Sendepuffer Voll	Wird gesetzt, wenn der Sendepuffer voll ist.
N	Senden Deaktiviert	Wird gesetzt, wenn der Sendevorgang deaktiviert ist. Könnte durch einen Befehl der Gegenstation verursacht werden, wenn "M" auch gesetzt ist.
O	Sendepuffer Nicht Leer	Wird gesetzt, wenn noch Sendedaten vorliegen. Der Sendepuffer ist nicht leer, und der USART hat den Sendevorgang noch nicht beendet. Dieser Status dient dazu, herauszufinden, wann der richtige Zeitpunkt zum Abschließen des Oliterm-Pakets bzw. zu dessen Re-Initialisierung gegeben ist.
P	Falscher oder fehlender Parameter	Wird gesetzt, wenn Oliterm einen Fehler in der Parameterliste feststellt. Dieser kann sein: <ul style="list-style-type: none"> <li>- Parameterwert liegt außerhalb des Wertebereichs</li> <li>- Falsche Parameterart</li> <li>- Parameter fehlt</li> </ul>

Tabelle A-1 Fehlercodes



## Datenempfangsfehler

Datenempfangsfehler:

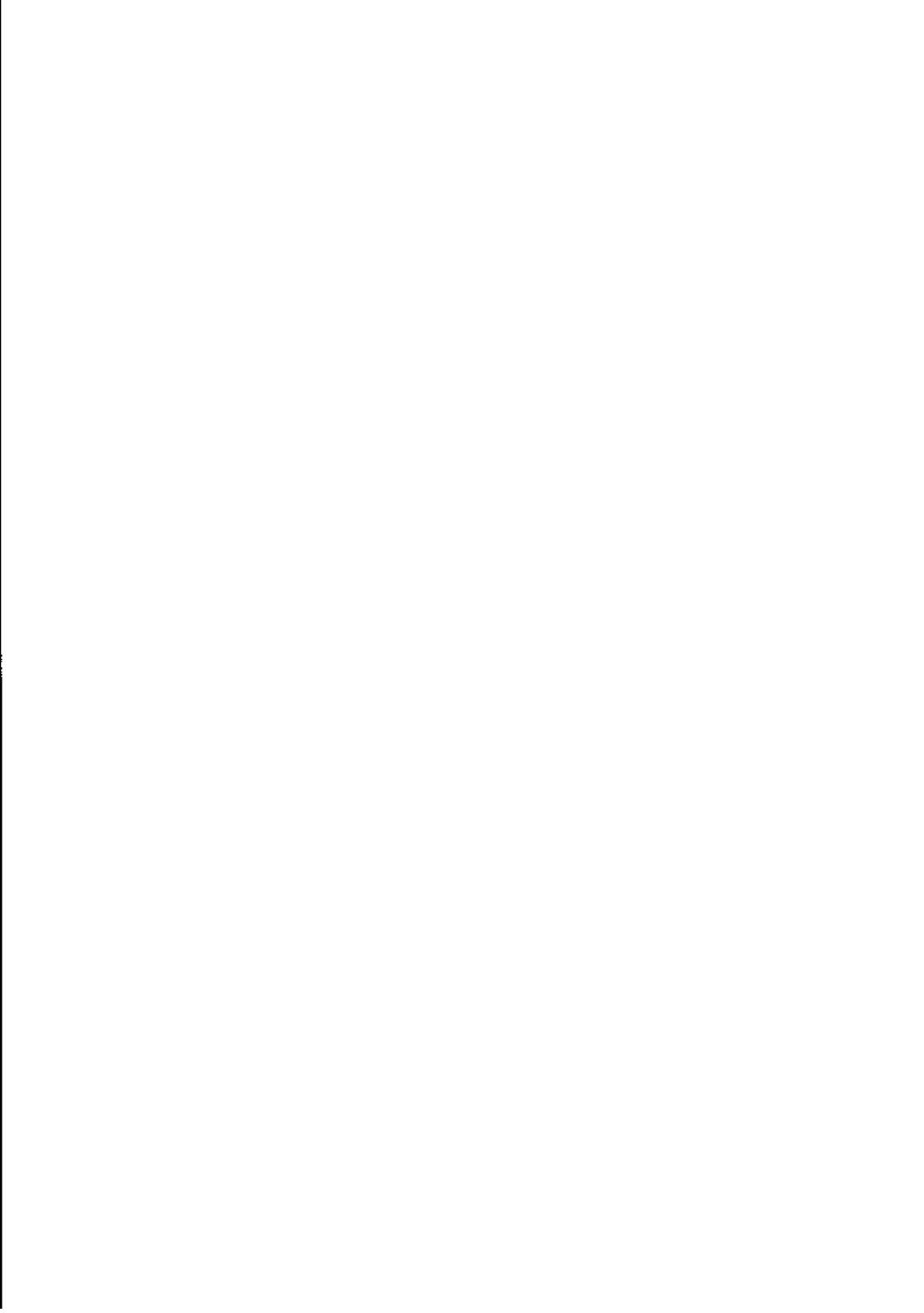
- G Empfangsunterbrechung
- H Empfangs-Overrun
- I Empfangs-Parityfehler
- J Empfangspuffer-Überlauf

- haben bei verschiedenen Funktionsaufrufen verschiedene Bedeutungen.

Bei "Byte Empfangen" (Funktion 3) geben sie die mit dem Byte verbundenen Fehler an.

Bei "Daten Empfangen" (Funktion 4) zeigen die Fehler H, I und J an, daß ein oder mehrere Bytes diese Fehler hatten . Fehler G bezieht sich nur auf das letzte Zeichen, weil die Datensteuerung die Daten bei einer Unterbrechung beendet.

Bei allen anderen Funktionen geben diese Fehler das Vorhandensein eines oder mehrerer Bytes im Empfangspuffer mit diesen Bedingungen an. Wenn das letzte fehlerhafte Byte aus dem Puffer herausgeholt worden ist, beendet Oliterm die Funktion.



Status-Rückmeldung

Funktion	Befehl	Status-Rückmeldung																			
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P				
1	INITIALISE OLITERM PROGRAM	x															x				
2	CLOSE OLITERM	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x			
3	READ BYTE	x	x		x		x		x		x		x					x			
4	READ MESSAGE	x		x		x		x		x		x		x		x					
5	WRITE BYTE	x	x	x	x												x	x	x	x	
6	WRITE MESSAGE	x		x		x											x	x	x	x	
7	OLITERM STATUS	x	x	x	x	x	x	x	x	x	x	x	x	x			x	x	x		
8	FLUSH RECEIVE BUFFER																				
9	RECEIVE DISABLE	x																			
10	RECEIVE ENABLE																				
11	SET RECEIVE TIMEOUT	x																		x	
12	FLUSH TRANSMIT BUFFER																				
13	TRANSMIT DISABLE					x												x	x	x	
14	TRANSMIT ENABLE					x													x	x	
15	SEND BREAK					x													x	x	x
16	SET ANSWER BACK																				
17	SEND ANSWER BACK MESSAGE					x													x	x	
18	SET TRANSMIT TIMEOUT	x																			x
19	SET MESSAGE CONTROL CHARACTERS																				
20	TURN OFF PORT																				
21	TURN ON PORT																				

Tabelle A-2 Status-Rückmeldung



