

V 24 - Peripherie
PROGRAMMIERHANDBUCH

olivetti L1

M20 - PROGRAMMIERUNG VON V24-PERIPHERIE

An den M20 können über die standardmäßig eingebaute oder über das zusätzliche Interface mit 2 weiteren Schnittstellen V24 Pheripherien angeschlossen werden.

Das Betriebssystem PCOS sieht dafür zwei Möglichkeiten vor:

- 1) Anschluß über Device-Rerouting
- 2) Programmierung über CALL "CI"

1) Möglichkeiten beim Device-Rerouting

PCOS bietet die Möglichkeit, Eingaben parallel oder anstelle der Tastatur über eine der V.24-Schnittstellen durchzuführen. Ebenso können Ausgaben an den Bildschirm anstelle oder zusätzlich über eine der V.24 Schnittstellen an eine Peripherie gesendet werden.

Es stehen damit alle Input-Möglichkeiten in der PCOS-Ebene bzw. die verschiedenen INPUT-Anweisungen der BASIC-Ebene für Eingaben, und alle Systemausgaben bzw. alle PRINT-Anweisungen der BASIC-Ebene für Ausgaben zur Verfügung.

1.1 Vorbereitende Arbeiten

Generell muß für das Arbeiten mit V.24 Programmierung der Modul "rs" geladen werden.

Es ist empfehlenswert, dies von der PCOS-Ebene durch Eingabe von

rs

oder

pl rs

durchzuführen.

Ob rs geladen ist oder nicht, erkennen Sie bei Aufruf des Systembefehles sd, da bei den möglichen In-/Outputeinheiten die V.24-Schnittstelle nur angeführt wird, wenn rs geladen wurde. Grundsätzlich ist der Aufruf auch über EXEC bzw. CALL von BASIC möglich, jedoch können dadurch Probleme in der Speicherorganisation entstehen.

1.2 Namen der Einheiten

Die standardmäßig verfügbare V.24 Schnittstelle hat den Namen **com**, die beiden weiteren Schnittstellen des zusätzlichen Interfaces heißen **com0** bzw. **com1**.

Mit Hilfe des Systembefehles **sd** können jedoch auch andere, beliebig wählbare Namen vergeben werden bzw. die aktuell gültigen Namen ermittelt werden.

1.3 Vereinbarung der Übertragungscharakteristika

Die Übertragungscharakteristika können über den Systembefehl **sc** vereinbart werden.

Sie umfassen:

- Betroffene Schnittstelle (**com:**,**com0:**,**com1:**)
- Übertragungsgeschwindigkeit
- Paritätsprüfung
- Anzahl Stopbits
- Anzahl Datenbits
- Echo ja/nein
- XON-XOFF Prozedur
- Buffergröße

Bei Eingabe von

sc com:

werden die aktuell gültigen Parameter angezeigt. Sie können nach den üblichen Regeln von PCOS geändert werden (siehe PCOS-Handbuch), bzw. das entsprechende System mit **PSAVE** gespeichert werden.

Zusammenfassung der vorbereitenden Arbeiten:

Laden der V.24-Driver `rs`

Setzen der Übertragungscharakteristika `sc com:`

1.4 Input über V.24 im Device-rerouting

Durch Eingabe von

`+scm:`

wird der Input über V.24 freigegeben und allen Tastatureingaben gleichgesetzt.

Die Eingaben sind sowohl auf System - als auch auf PCOS-Ebene gültig.

Als Abschlußzeichen gilt das Zeichen `CR` (ASCII 13).

In Basic können damit alle INPUT-Anweisungen Werte erhalten, nicht jedoch INKEY\$!

Bei Programmen, die auch mit Device-Rerouting arbeiten sollen, ist daher auf INKEY\$ in der Programmerstellung zu verzichten.

Bei der Übertragung von ASCII-Werten, die einer Taste entsprechen, der mit PKEY ein Inhalt zugewiesen wurde, wird das ursprüngliche Zeichen und nicht der zugewiesene Wert erkannt. Ebenso kann natürlich nicht zwischen den 3 Abschlußtasten unterschieden werden.

1.5 Senden über V.24 im Device-rerouting

Durch Eingabe von

`+dcom:`

werden alle an den Bildschirm gerichteten -numerischen Ausgaben auch an die V.24 Schnittstelle gesendet.

Diese Ausgaben enthalten daher nicht:

- Cursorpositionierungen über CURSOR-Anweisungen
- Farbanweisungen, Löschen des Schirmes, Windowwechsel etc.
- Alle Graphik-Anweisungen

Sie enthalten alle im System - oder BASIC-Commandmode eingegebenen oder empfangenen Zeichen, alle mit INPUT oder ähnlichen Anweisungen übernommenen Zeichen (nicht INKEY\$) und alle mit PRINT oder vom System generierten Ausgaben.

1.6 Deaktivieren des Inputs oder Outputs

Der Input über V.24 kann jederzeit deaktiviert werden.

In Systemebene durch Eingabe von

`-scom:`

bzw. in BASIC durch:

`EXEC "-scom:"`

Die Ausgabe wird analog in Systemebene durch

`-dcom:`

bzw. in BASIC durch

`EXEC "-dcom:"`

deaktiviert.

Eine neuerliche Aktivierung mit den entsprechenden Befehlen ist jederzeit möglich. (+scom: bzw. +dcom:)

1.7 Beispiel für die Übernahme eines Programmes

Auf einem zweiten System existiert ein BASIC-Programm im ASCII-Code. Dieses Programm soll über V.24 übernommen und auf M20 als ASCII-File gespeichert werden.

Nachfolgend wird der Ablauf dafür beschrieben:

1.7.1 Überzeugen Sie sich über die Übertragungscharakteristika beim Zweitgerät. Wir nehmen an, es wird mit 4800 Baud, Gerader Parität, 2 Stopbits und 7 Datenbits übertragen.

Beim M20 wird das System geladen, das System befindet sich in der PCOS-Ebene.

Folgende vorbereitende Eingaben sind forderlich:

```
rs
sc com: ,4800,even,2,7,half,off,255
+scm:
ba
```

1.7.2 Starten Sie jetzt die Übertragung am Zweitgerät. Es ist dabei nötig, mit der ersten Programmzeile zu beginnen und der letzten Programmzeile zu enden. Jede Zeile beginnt mit der Zeilennummer und endet mit **CR**.

Sonstige Datensätze werden am M20 als Direkt-Eingaben interpretiert und ausgeführt.

Sie sehen die übertragenen Programmzeilen am Bildschirm.

Nach dem Ende der Übertragung steht das Programm im Arbeitsspeicher so zur Verfügung, als ob Sie es über die Tastatur eingegeben hätten.

1.8 Beispiel zum Senden eines Programmes

Mit gleichen Annahmen wie zuvor soll ein Programm aus dem M20-Arbeitsspeicher an ein zweites Gerät übertragen werden.

Wir nehmen an, daß bereits rs geladen wurde und die Übertragungsparameter stimmen (sc com:).

Wir befinden uns in BASIC, das Programm ist im Arbeitsspeicher.

Folgende Eingaben sind durchzuführen:

```
exec "+dcom:"  
list
```

Damit wird die Übertragung gestartet.

Mit

```
exec "-dcom:"
```

wird die Ausgabe über V.24 wieder deaktiviert.

1.9 Verwendung von V.24-Druckern mit M20

Drucker werden üblicherweise über die Centronics-Schnittstelle angeschlossen. PCOS erlaubt aber auch den Anschluß über die V.24-Schnittstelle. Dabei ist folgende Vorgangsweise nötig:

- 1) Laden des V.24-Moduls rs.
- 2) Setzen der Übertragungsparameter mit sc com:
- 3) Setzen des Befehles "sf" auf Interface: se

Diese Reihenfolge muß eingehalten werden. Ein Ändern der Übertragungscharakteristika erfordert die neuerliche Ausführung von sf.

Olivetti-Drucker werden so geliefert, daß mit den Standardparametern von sc com: gearbeitet werden kann. Es ist unbedingt zu empfehlen, das Betriebssystem mit psave richtig konfiguriert zu speichern. Nur dadurch kann vermieden werden, daß bei jedem Neuladen der Modul geladen bzw. die Parameter neu gesetzt werden müssen.

2) PROGRAMMIERUNG DER V.24-SCHNITTSTELLE MIT CALL "CI"

Obwohl das Device-Rerouting für fallweise Datenaustausch und für Testzwecke für die Inbetriebnahme von Peripheriegeräten hervorragende Dienste leistet, ist eine anwendungsbezogene Programmierung der Peripherie wesentlich günstiger mit den über CALL "CI" erreichbaren Routinen durchzuführen.

2.1 Vorbereitende Arbeiten

Die Vorbereitungen sind identisch den in Abschnitt 1.1 bis 1.3 beschriebenen Befehlen.

Es müssen der Modul rs geladen sein und mit sc com: die Übertragungsparameter gesetzt werden.

Zusätzlich sind die V.24-Routinen mit

pl ci

in den Speicher zu laden. In den mit V.24 arbeitenden BASIC-Programmen werden diese Routinen mit call "ci" aufgerufen. Die dabei zur Auswahl stehenden Möglichkeiten werden nachfolgend beschrieben.

Anmerkung: Da durch das Laden von rs und von ci der Arbeitsspeicher entsprechend kleiner wird, ist meist eine Reduktion des in sbasic reservierten Platzes nötig. Es ist dies bei der Erstellung von Programmen zu berücksichtigen.

2.2 Überblick über die Routinen

Das allgemeine Format lautet:

CALL "CI" (Schnittstelle, Befehl, Error, Parameter)

dabei gilt:

"Schnittstelle" ist ein Integer-Wert oder Variable mit den Werten
0 für die eingebaute Standard V.24 (com:)
1 die erste Erweiterung (com1:)
2 die zweite Erweiterung (com2:)

Die in () angeführten Bezeichnungen sind die Standard-Device-Namen der entsprechenden V.24-Schnittstelle. In dieser Beschreibung wird generell PN% benützt.

"Befehl" Ist ein String mit Länge 1 oder 2, der den Typ der gewünschten Operation bezeichnet.

Es kann sein

o (open)

für "öffnen" der Schnittstelle

c (close)

für "schließen" der Schnittstelle

r (read)

für Empfangen)

w (write)

für Schreiben

sr (Status read)

Lesen des Schnittstellenstatus

sw (Status write)

Schreiben des Schnittstellenstatus

"ERROR"

Ist eine Integervariable, die einen Fehlercode retourniert.

Dabei gilt allgemein, daß ein retournierter Wert

0 eine korrekte Ausführung bedeutet

1 Fehler in der Angabenliste bedeutet

Die weiteren Werte haben für die einzelnen Befehle unterschiedliche Bedeutung.

In dieser Beschreibung wird der Einfachheit halber generell E% als Variable und im Aufruf daher §E% geschrieben. (Siehe BASIC-Handbuch, Anweisung CALL)

"PARAMETERLISTE" In Abhängigkeit vom jeweiligen Befehl anzugebende Parameter.

Anmerkung:

Alle im CALL verwendeten Variablen müssen bereits vor dem 1. Aufruf von CALL "CI" einen Wert erhalten haben. Bei Stringvariablen, denen in CALL "CI" ein Wert zugewiesen wird, ist darüberhinaus zuvor ein entsprechend langer String zuzuweisen.

(z.B. A\$=SPACE\$(255))

Beispiel:

Die Anweisungsfolge

100 PN% = 0: E% = 0

110 CAL "CI" (PN%,"o",§E%)

120 IF E% 0 THEN PRINT "FEHLER BEIM ÖFFNEN;":STOP

Stellt eine gültige Eröffnung für die eingebaute V.24-Schnittstelle dar.

2.3 Reihenfolge der Aufrufe

Bevor ein Datenaustausch über V.24 erfolgen kann, muß mit dem Befehl "o" (open) die Schnittstelle geöffnet werden.

Anschließend kann in beliebiger Reihenfolge mit "r" Daten empfangen bzw. mit "w" Daten gesendet werden. Die Notwendigkeit von Statusprüfungen ("sr") oder Statusänderungen ("sw") ist nach den einzelnen Anwendungen zu beurteilen.

Nach Abschluß der Übertragung ist die Schnittstelle mit "c" (close) zu schließen.

2.4 Der Befehl "o" (open)

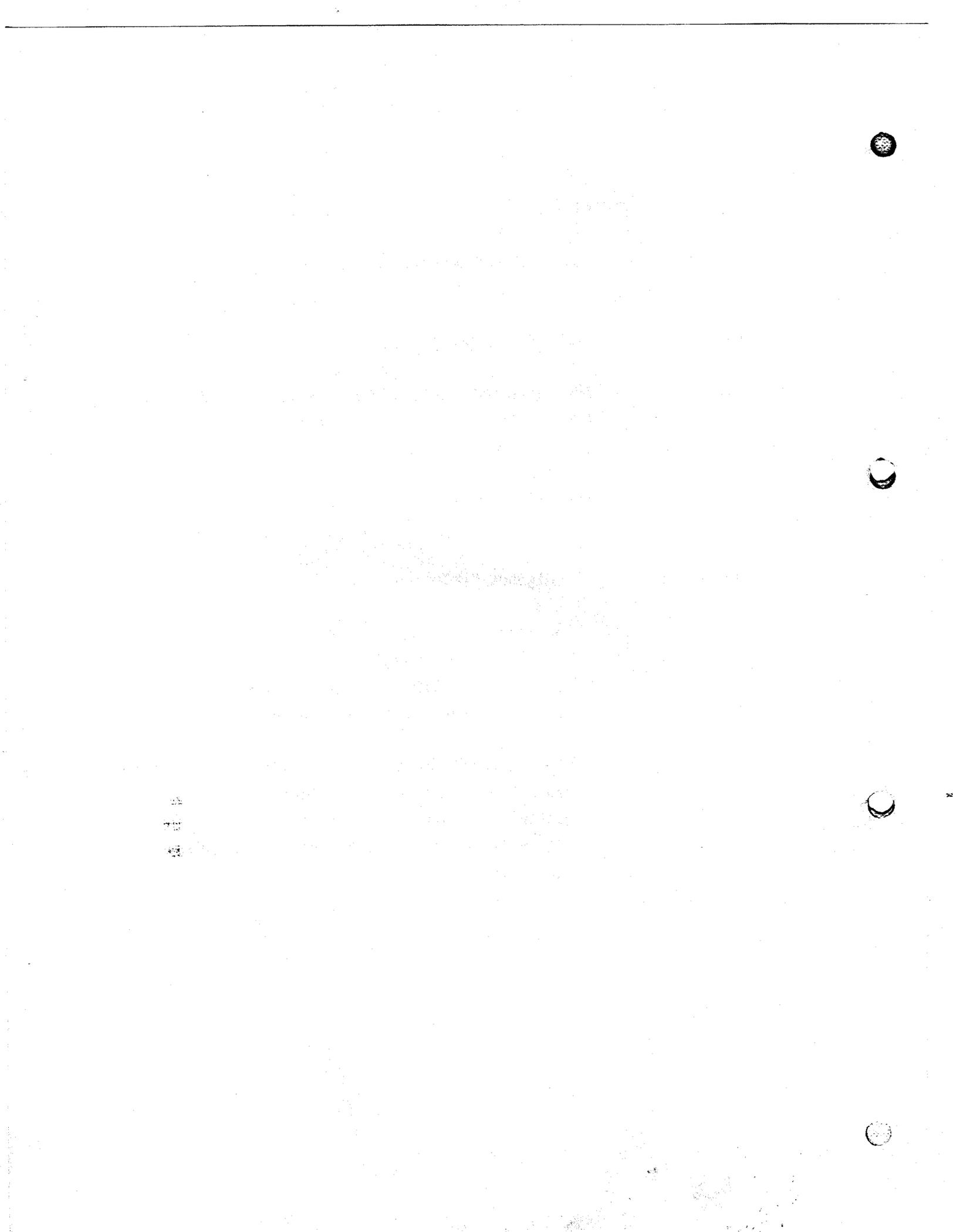
Funktion: Öffnen einer V.24-Schnittstelle für die nachfolgende Datenübertragung.

Format: CALL "CI" (PN%,"o",§E%)

Wirkung: Der durch PN% bezeichnete Kanal wird nach den mit sc vereinbarten Charakteristika geöffnet. Nachfolgend empfangene Zeichen werden im Buffer abgelegt.

Bemerkung: Folgende Fehlercodes können auftreten:

- 0: korrekte Eröffnung
- 1: Fehler in den Parametern
- 2: Kanal bereits geöffnet
- 3: zu großer Empfangsbuffer



2.6 Der Befehl "r" (read)

Funktion: Es können Datensätze einer bestimmten Länge oder bis zu einem Trennzeichen übernommen werden.

Format: 1) CALL "CI" (PN%,"r",§E%,Stringvariable)
2) CALL "CI" (PN%,"r",§E%,Stringvariable,Zähler)

wobei:

Stringvariable: Der Name der Stringvariablen in der die Daten übernommen werden.

Zähler: Integer-Variable, die als Werte die Anzahl der zu übernehmenden Zeichen enthält.

Wirkung: 1) Ist kein Zähler angegeben, so werden so lange Zeichen übernommen, bis entweder das vereinbarte Trennzeichen empfangen wird (siehe Befehl "SW") oder die aktuelle Länge des Strings erreicht wird.
Der Befehl wird beendet, sobald eine dieser beiden Konditionen eintritt.

2) Ist der Parameter "Zähler" angegeben, so werden so viele Zeichen übernommen, als dem Wert von "Zähler" entspricht. Der Befehl wird beendet, sobald diese Zeichenanzahl empfangen wurde.

2.8 Der Befehl "sr" (Status read)

Funktion: Der Befehl dient zur Ermittlung von Hardwarestatus, Kanalstatus und Anzahl der empfangenen Zeichen im Buffer.

Format: CALL "CI" (PN%,"sr",§E%, Hardwarestatus, Kanalstatus, Byteanzahl)

wobei:

Hardwarestatus: Name einer gültigen Integervariablen, die den Inhalt des Hardwarestatusbytes zugewiesen erhält.

Kanalstatus: Name einer gültigen Integervariablen, die den Inhalt des Kanalstatuswertes zugewiesen erhält.

Byteanzahl: Name einer gültigen Integervariablen, die die Anzahl der aktuell im Buffer vorhandenen Zeichen zugewiesen erhält.

Wirkung: Die Inhalte der Statusbyte und die Zeichenanzahl im Buffer werden den Variablen zugewiesen.

Bemerkungen: 1) Folgende Fehlercodes sind möglich:

0: Befehl korrekt ausgeführt

1: Fehler in den Parametern

2: Kanal nicht geöffnet

2) In der Parameterliste können Parameter entfallen. Es ist jedoch durch Angabe der entsprechenden Kommaanzahl der gewünschte Parameter eindeutig festzulegen.

2.7 Der Befehl "w" (write)

Funktion: Es werden Daten an eine V.24-Peripherie gesendet.

Format: CALL "CI" (PN%,"W",§E%,Ausgabeliste)

wobei:

Ausgabeliste: Folge von Integers oder Strings,
die durch Komma getrennt sind.
Die Liste muß mindestens ein Element
enthalten.

Wirkung: Die Elemente der Ausgabeliste werden an die
Peripherie gesendet, wobei:

- Strings, unverändert gesendet werden
- Integer-Werte zur Übermittlung von Kontroll-
zeichen oder binären Informationen verwendet
werden.

Es wird vom entsprechenden Integer-Wert nur
das weniger signifikante Byte übertragen.

Bemerkungen: Folgende Fehlercodes sind möglich:

- 0: Korrekte Ausführung des Befehles
- 1: Fehler in den Parametern
- 2: Kanal nicht geöffnet
- 3: Zeitüberschreitung bei der Übertragung

2.8 Der Befehl "sr" (Status read)

Funktion: Der Befehl dient zur Ermittlung von Hardwarestatus, Kanalstatus und Anzahl der empfangenen Zeichen im Buffer.

Format: CALL "CI" (PN%,"sr",§E%, Hardwarestatus, Kanalstatus, Byteanzahl)

wobei:

Hardwarestatus: Name einer gültigen Integervariablen, die den Inhalt des Hardwarestatusbytes zugewiesen erhält.

Kanalstatus: Name einer gültigen Integervariablen, die den Inhalt des Kanalstatuswertes zugewiesen erhält.

Byteanzahl: Name einer gültigen Integervariablen, die die Anzahl der aktuell im Buffer vorhandenen Zeichen zugewiesen erhält.

Wirkung: Die Inhalte der Statusbyte und die Zeichenanzahl im Buffer werden den Variablen zugewiesen.

Bemerkungen: 1) Folgende Fehlercodes sind möglich:

0: Befehl korrekt ausgeführt

1: Fehler in den Parametern

2: Kanal nicht geöffnet

2) In der Parameterliste können Parameter entfallen. Es ist jedoch durch Angabe der entsprechenden Kommaanzahl der gewünschte Parameter eindeutig festzulegen.

3) Den Variablennamen muß im Aufruf das Zeichen "\$" vorangestellt sein.

4) "Hardwarestatus" ist ein Byte mit folgender Bedeutung der einzelnen Bits:

Bezeichnung	Bitpos.	Wert	Bedeutung
DATA SET READY	7	1	DATA SET READY ON
		0	OFF
SYN erkannt	6	0	Asynchrone Übertragung
Formatfehler	5	1	Am Ende des Zeichens wurde kein Stopbit erkannt.
		0	kein Formatfehler
Überlauf-Fehler	4	1	Das nächste Zeichen wurde empfangen, bevor das vorhergehende gelesen wurde.
		0	kein Fehler
Paritätsfehler	3	1	Paritätsfehler wurde erkannt
		0	kein Fehler
Sender leer	2	1	Es sind keine zu sendenden Daten vorhanden
		0	Daten vorhanden
Empfängerstatus	1	1	Es wurde ein Zeichen vollständig empfangen
		0	kein vollständiges Zeichen
Senderstatus	0	1	Sender kann Zeichen übernehmen
		0	Sender kann kein Zeichen übernehmen.

5) Kanalstatus ist ein Wort (2 Byte), dessen Bits folgende Bedeutung haben:

Bezeichnung	Bit	Wert	Bedeutung
Echo	15	1	Echo für alle empfangenen Daten
		0	kein Echo
Formatfehler	14	0	
Überlauffehler	13	1/0	wie Hardwarestatus
Paritätsfehler	12	1/0	wie Hardwarestatus
Timeout-Fehler	11	1/0	wie Hardwarestatus
	10	1	Es wurden 1000 erfolglose Versuche, ein Zeichen zu senden, durchgeführt
		0	kein Fehler
Speicher-Fehler	9	1	Fehler bei der Übertragung in den Speicher
		0	kein Fehler
Buffer-Fehler	8	1	Versuch, den Buffer zu überschreiben
		0	kein Fehler
----	7	0	---
Free-running	6	1	kein XON/XOFF Handshake
		0	XON/XOFF Handshake
XON/XOFF für Sender	5	1	XOFF
		0	XON zuvor empfangen
Hardwarestatus	4	1	Hardware o.K.
		0	Fehler
XON/XOFF für Empfänger	3	1	XOFF
		0	XON erkannt in laufender Übertragung
----	2-0	0	----

- 6) Die Byteanzahl im Buffer ist 0, falls keine zu übernehmenden Zeichen vorhanden sind. Ist Byteanzahl 0, so sind noch Zeichen im Buffer vorhanden.
Um Blockaden zu vermeiden, sollte nicht der Befehl "r" ausgeführt werden, wenn die Byteanzahl 0 ist.
- 7) Für eine korrekte Verbindung muß im Hardwarestatus Bit 7 (DATA SET READY) vorhanden sein. Es ist in den Übertragungsprogrammen empfehlenswert, das Vorhandensein dieses Bits zu überprüfen.
- 8) Treten Übertragungsfehler beim Empfangen von Daten auf, so meldet der Befehl "r" den Fehlercode 4. Es kann dann bei nachfolgender Prüfung mit "sr" über den Hardware und Kanalstatus die genauere Fehlerursache ermittelt werden.

2.9 Der Befehl "sw" (Status-write)

Funktion: Für einen Kanal wird das gültige Trennzeichen bestimmt und/oder der Hardware oder Kanalstatus gesetzt.

Format: CALL "CI" (PN%, "sw", §E%, Trennzeichen, Hardwarestatus, Kanalstatus)

wobei:

Trennzeichen: Ist ein gültiger Integerwert, der den dezimalen Wert des aktuellen Trennzeichens bestimmt.

Hardwarestatus: Ist eine gültige Integervariable, deren weniger signifikantes Byte den Hardwarestatus setzt.

Kanalstatus: Ist eine gültige Integervariable, die den Kanalstatus setzt.

Wirkung: Es werden das gültige Trennzeichen bzw. Hardwarestatus und/oder Kanalstatus auf den neuen Wert gesetzt.

Bemerkungen: 1) Folgende Fehlercodes sind möglich:

- 0: Befehl korrekt ausgeführt
- 1: Fehler in den Parametern
- 2: Kanal nicht geöffnet

2) Einzelne Parameter der Parameterliste können entfallen. Es muß jedoch zur eindeutigen Bestimmung die entsprechende Anzahl von Kommas gesetzt werden.

3) Hardwarestatus und Kanalstatus stimmen mit den in 2.8 gegebenen Aufstellungen überein (Siehe Befehl "sr").

2.10 Beispiel für ein Übertragungsprogramm

Das als Beispiel angeführte Programm sendet den Inhalt eines sequentiellen Datenfiles an eine V.24-Peripherie. Als Trennzeichen wird das Zeichen LF (Dez.10) gesendet.

```
10 CLEAR:CLS:DEFINT A-Z
20 CURSOR (20,1):PRINT "V.24 - SENDEPROGRAMM"
30 CURSOR (1,3):LINE INPUT "NAME DES ZU ÜBERTRAGENDEN FILES: ";F$
40 IF F$="" THEN 220
50 OPEN "I",#1,F$
60 PN=0:E=0:HS=0:DS=0:SEP=10:Z=0
70 CALL "ci"(PN,"o",E)
80 CALL "ci"(PN,"sr",E,HS,DS)
90 IF HS<&H85 THEN CURSOR (1,10):PRINT "VERBINDUNG HERS TELLEN";:GOTO 80 ELSE CLS:PRINT "VERBINDUNG O.K. - ÜBERTRAGUNG BEGINNT":PRINT
100 CALL "ci"(PN,"sr",E,HS,DS)
110 IF E<2 AND HS=&H85 THEN 140
120 IF HS<&H85 THEN PRINT "*****VERBINDUNG UNTERBROCHEN" ELSE PRINT "*****UEBERTRAGUNGSFEHLER"
130 GOTO 100
140 LINE INPUT#1,A$
150 CALL "ci"(PN,"w",E,A$,SEP)
160 IF E=4 THEN GOTO 120
170 PRINT A$
180 Z=Z+1
190 IF EOF(1)=0 THEN 100
200 PRINT:FRINT "UEBERTRAGUNG BEENDET. ES WURDEN";Z;" SAETZE GESENDET"
210 CLOSE #1
220 CALL "ci"(PN,"c",E)
230 END
```

2.11 Beispiel für Empfangsprogramm

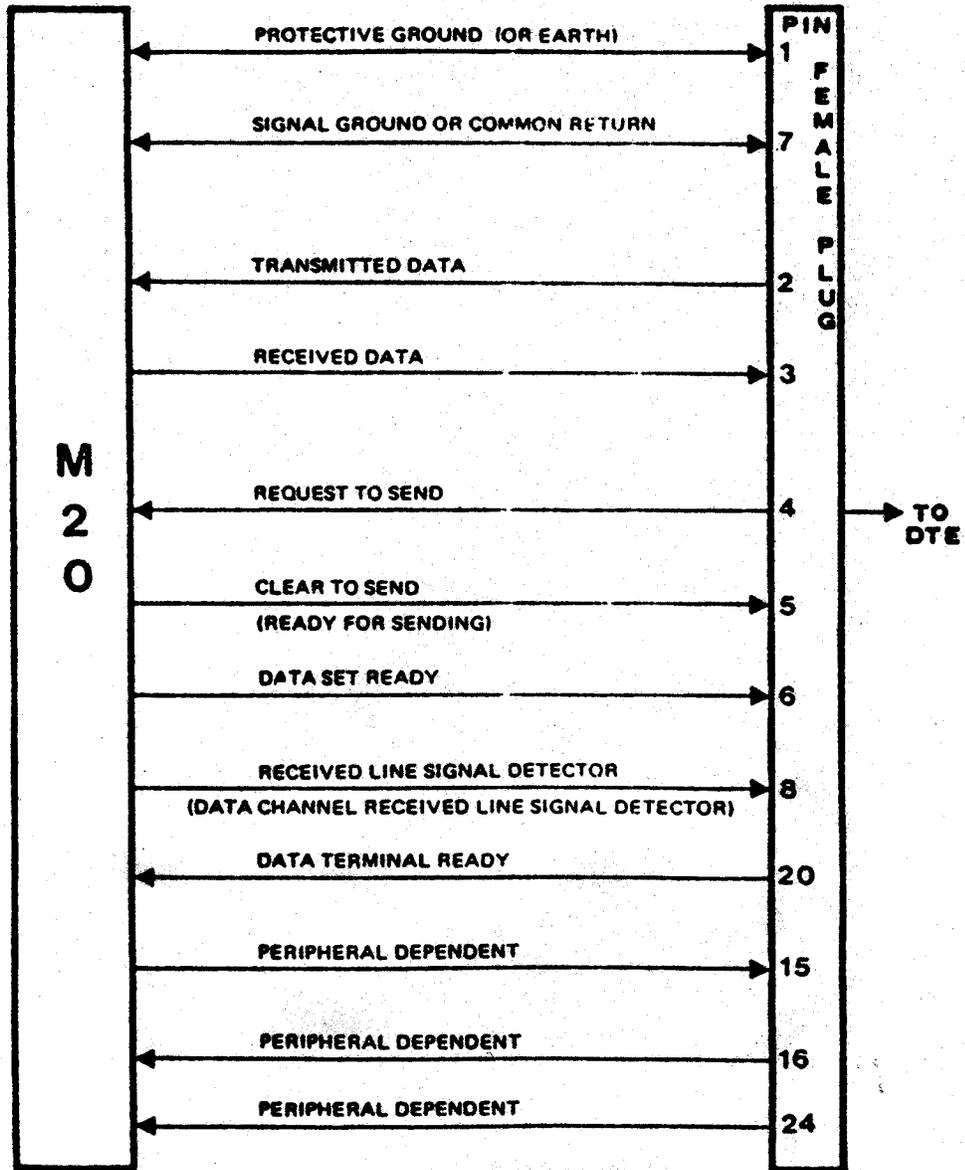
Das Programm empfängt von einer V.24-Peripherie Datensätze und schreibt sie in ein sequentielles File.

Jeder Datensatz wird mit dem Zeichen LF (Dezimal 10) oder nach maximal 80 Zeichen beendet.

Die Übertragung wird beendet, sobald ein Datensatz das Wort "ENDE" an der 1. Position enthält.

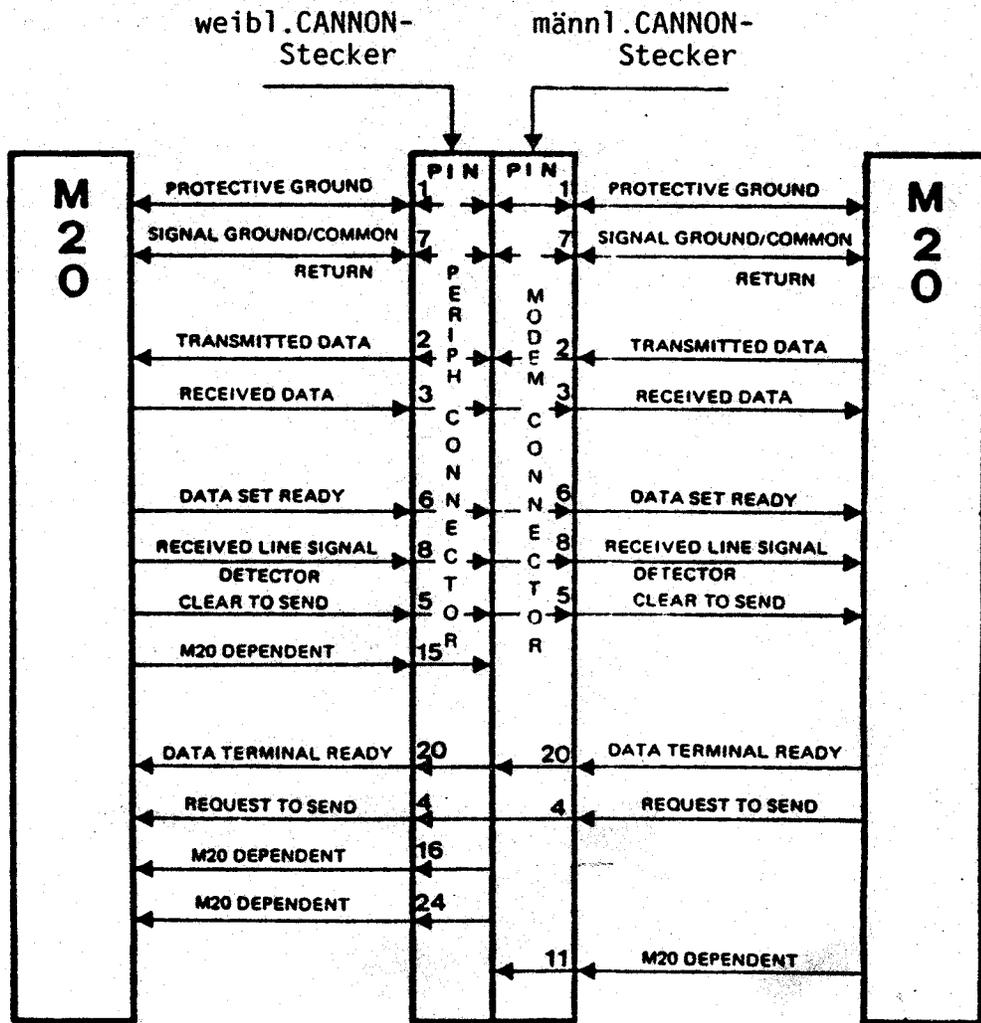
```
10 CLEAR:CLS:DEFINT A-Z
20 CURSOR (20,1):PRINT "V.24 - ÜBERNAHMEPROGRAMM"
30 CURSOR (1,3):LINE INPUT "Output-Filename: ";F$
40 IF F$="" THEN 320
50 OPEN "o",#1,F$
60 FN=0:E=0:HS=0:BC=0:DS=0:SEP=10:Z=0
70 CALL "ci"(PN,"o",E)
80 CALL "ci"(PN,"sr",E,HS,DS)
90 IF HS<&H85 THEN CURSOR (1,10):PRINT "VERBINDUNG HERS
TELLEN";:GOTO 80 ELSE CLS:PRINT "VERBINDUNG O.K. - ÜBER
TRAGUNG BEGINNT":PRINT
100 CALL "ci"(PN,"sw",E,SEP)
110 TO=VAL(MID$(TIME$,5,1)+RIGHT$(TIME$,2))
120 A$=STRING$(80,128)
130 CALL "ci"(PN,"sr",E,HS,DS,BC)
140 T=VAL(MID$(TIME$,5,1)+RIGHT$(TIME$,2)):IF ABS(TO-T)
>100 THEN PRINT "***** ZEITUEBERSCHREITUNG FU
R DATENEMPFANG":PRINT:GOTO 300
150 IF E<2 AND HS=&H85 THEN 190
170 IF HS<&H85 THEN PRINT "*****VERBINDUN
G UNTERBROCHEN" ELSE PRINT "*****UEBERT
RAGUNGSFEHLER"
180 GOTO 130
190 IF BC=0 THEN 130
200 CALL "ci"(PN,"r",E,A$)
210 IF E<4 THEN 220 ELSE GOTO 120
220 PO=INSTR(A$,CHR$(128))
230 IF PO=0 THEN R$=A$ ELSE R$=LEFT$(A$,PO-1)
240 IF INSTR(R$,"ENDE")=1 THEN 300
250 PRINT R$
260 PRINT #1,R$
270 T=VAL(MID$(TIME$,5,1)+RIGHT$(TIME$,2)):TO=T
280 Z=Z+1
290 GOTO 120
300 PRINT "UEBERTRAGUNG BEENDET. ES WURDEN";Z;" SAETZE
KORREKT EMPFANGEN"
310 CLOSE #1
320 CALL "ci"(PN,"c",E)
330 END
```

Anhang 1): Datensignale bei M20 mit Peripheriekabel
(Weiblicher CANNON-Stecker)



Anmerkung: M20 verhält sich als DCE (entspricht DÜE nach DIN)

Anhang 3): Zusammenschluß von zwei M20



DATENSIGNALE ZWISCHEN ZWEI M20

