

celectronic

Volker Czmok · Elektronische Geräte GmbH



Nordlichtstr. 63-65 · D-13405 Berlin · Telefon 030/4136075 · Telefax 030/4136078

Bedienungsanleitung

ceLink

Remote-Software für Programmiergerät
promicron 2000 und 2100



Februar 1993

((ce2A))

Copyright: Celectronic GmbH Berlin

0	Bildschirmansicht von ceLink	
1	EINLEITUNG	
1.1	Allgemeine Angaben	1-1
1.2	Benötigte Betriebssoftware	1-2
2	INBETRIEBNAHME	
2.1	Verbindung zwischen Rechner und Programmierer	2-1
2.2	Initilisierung des Programmiergerätes	2-1
2.3	Installation und Aufruf des Programmes	2-2
2.3.1	Installation	2-2
2.3.2	Einstellung von Übertragungsrate und Darstellung	2-2
2.3.3	Starten des Programmes	2-3
3	BESCHREIBUNG DES PROGRAMMES	
3.1	Bildschirmdarstellung	3-1
3.2	Tastatureingaben	3-1
3.2.1	Tasteneingaben	3-1
3.2.2	Funktionstasten und DOS-Funktion	3-2
3.3	Konzeption der Datenspeicherung	3-2
3.5	Funktionsübersicht	3-3
3.5	Funktionsbeschreibungen RAM-Funktionen	3-3
3.5.1	Funktion <i>FILE</i>	3-4
3.5.1	Funktion <i>MEMORY</i>	3-5
3.6	Funktionsbeschreibungen Device-Funktionen	3-7
3.6.1	Funktion <i>DEVICE</i>	3-7
3.6.2	Funktion <i>LOAD</i>	3-8
3.6.3	Funktion <i>PROGRAM</i>	3-8
3.6.4	Funktion <i>VERIFY</i>	3-8
3.6.5	Funktion <i>TYPE</i>	3-8
3.7	Funktionsbeschreibungen Sonder-Funktionen	3-9
3.7.1	Funktion <i>DIRECT MODE</i>	3-9
3.7.2	Funktion <i>TRANSFER</i>	3-10
3.8	Funktionsbeschreibungen sonstige Funktionen	3-10
3.8.1	Funktion <i>ADDRESS</i>	3-10
3.8.2	Funktion <i>SETUP</i>	3-12
3.9	ceLink beenden	3-13
A	ANANG A: ZUSATZPROGRAMME	
A.1	Programm <i>ceMode</i>	A-1
A.2	Programm <i>ceSplit</i>	A-1

ceLink Remote Software 2.A1, Serial #...
File Memory Device Load P...
celectronic berlin
 ...ress Setup Extra
 ...ile... >

KOMMANDO-FENSTER
 ...ile... >

Vers 2.A1 (C) celectronic berlin NOV 1992

[V] [E]
BERLIN
 ARBEITS-FENSTER

Selected File
 ÄFlü - Filename >
 <.. >
 <CE231 >
 <CE2A1 >
 <CEEDIT >
 <CEED OLD >
 DEMOTEXT.BIN
 TEST.BIN

FILE-FENSTER

Manufact.: **SGS-Thomson**
 Name : **M27C256B_B**
 Type : **PROM**
 Organiz. : **32Kx8**
 Length
 DEVICE-FENSTER

	Start	End	Maximui
Memory	\$000000	\$007FFF	\$007FFF
Device	\$000000	\$007FFF	007FFF
File	\$000000	\$007FFF	007FFF
ADDRESS	\$000000	\$000000	\$007FFF

ADDRESS-FENSTER

1 EINLEITUNG

Diese Bedienungsanleitung ist gültig für die Programmiergeräte:

promicron 2000 und promicron 2100.

Die Programmiergeräte promicron 2200 und promicron 2300 können ebenfalls mit dieser Version von ceLink bearbeitet werden. Diese Geräte und ceLink sind gemeinsam in einer eigenen Bedienungsanleitung beschrieben.

1.1 ALLGEMEINE ANGABEN

Das vorliegende Kommunikationsprogramm *ceLink Version 2.A* dient zur komfortablen Fernbedienung der EPROM-Programmiergeräte der Reihe *promicron 2x00*. Das angeschlossene Gerät wird dabei automatisch von *ceLink* identifiziert. Infolge der Anpassung an die spezielle Befehlssyntax dieses Gerätes kann die Software nicht zur Steuerung anderer Programmiergeräte verwendet werden.

Hinweis: Im folgenden wird die Gruppe der programmierbaren Bauelemente (EPROMs, EEPROMs, Single-Chip-Prozessoren etc.) zum Begriff "**Device**" zusammengefaßt.

Der Befehlsumfang beinhaltet RAM-Funktionen (Lesen, Schreiben und Ändern des Daten-RAMs), Device-Funktionen (Lesen, Testen, Programmieren, Vergleichen und Typeinstellung) und Datei-Funktionen (Lesen und Schreiben von Dateien im Binär-, Hex-Intel- oder Motorola-Format). Weitere Funktionen dienen zur Einstellung verschiedener Programm-Parameter und Adressen.

Die Auswahl aller Funktionen erfolgt menügesteuert innerhalb einer Bildschirmmaske mit den Cursortasten oder durch Eingabe des Anfangsbuchstabens der Funktion. Hinweise sowie eventuelle Fehlermeldungen werden in Klartext angezeigt und durch ein akustisches Signal unterstützt. Zusätzliche Bildschirmfenster zeigen ständig Informationen über den ausgewählten Baustein (Hersteller, Name, Organisation etc.) sowie den Adreßbereich von RAM und Device.

ceLink 2.A arbeitet üblicherweise von einer Festplatte aus. Das Starten von *ceLink* von einer Diskette ist aus Gründen der Geschwindigkeit nicht zu empfehlen. Für den Einsatz in Netzwerken können beliebige logische Laufwerke (A: bis Z:) spezifiziert werden.

Wichtig: Im gewählten Laufwerk sollte mindestens soviel freier Speicherplatz vorhanden sein, wie der RAM-Kapazität des angeschlossenen Programmiergerätes entspricht.

ceLink stellt sich automatisch auf den vorhandenen Grafik-Adapter ein, der Datenaustausch mit dem Programmiergerät kann wahlweise über die seriellen Schnittstellen COM1: oder COM2: erfolgen.

Das Programm läuft auf den IBM-Computern der PC-, XT- und AT-Serie sowie dazu 100% kompatiblen Geräten. *ceLink 2.A* wird auf einer Diskette im Format 5 1/4", 360K geliefert und besteht aus sechs Dateien.

Die Stapeldatei HDINSTAL.BAT dient zur Installation des Programms auf einer Festplatte. Der Anwender kann außerdem unter Beachtung der Lizenzbestimmungen Sicherheitskopien von *ceLink* anfertigen.

ceLink hat einen Umfang von ca. 175 KBytes, als Puffer für die Programmierdaten wird ein RAM-Bereich von 32 KByte benötigt, zusätzlicher Speicherplatz wird für die Typtabellen gebraucht; der Rechner sollte einen Arbeitsspeicher von mindestens 512 KBytes besitzen.

1.2 BENÖTIGTE BETRIEBS-SOFTWARE

Um einen störungsfreien Betrieb des **promicron 2000** in Verbindung mit *ceLink* zu gewährleisten, muß das Programmiergerät mindestens mit der Betriebs-Software Version 4.21 ausgestattet sein.

2 INBETRIEBNAHME

Die Inbetriebnahme des Systems Programmiergerät - Rechner erfolgt in drei Schritten:

- Verbindung von Programmiergerät und Computer über die serielle V24- (RS232C-) Schnittstelle,
- Initialisierung des Programmiergerätes und Einstellung der Remote-Betriebsart,
- Installation, Konfiguration und Start des Programmes *ceLink*.

2.1 VERBINDUNG ZWISCHEN RECHNER UND PROGRAMMIERGERÄT

Die Verbindung zwischen Programmiergerät und Rechner erfolgt auf der Programmer-Seite über die eingebaute serielle RS232C-Schnittstelle und auf der Rechnerseite wahlweise über COM1: oder COM2:, einzustellen über das Zusatzprogramm *ceMode*, siehe Anhang A.1. Die Synchronisation zwischen Programmiergerät und Computer wird sowohl über Hard- als auch Software-Handshake sichergestellt.

Zum Anschluß des Programmiergerätes an den Rechner ist bei Bedarf ein geeignetes Kabel lieferbar (Pin-Belegung siehe Anhang B):

Für IBM XT oder PC (25pol. D-Sub Buchse)	Best.-Nr. 0635
Für IBM AT (9pol. D-Sub Buchse)	Best.-Nr. 0636

2.2 INITIALISIERUNG DES PROGRAMMIERGERÄTES

Um die Kommunikation zwischen Computer und Programmiergerät zu ermöglichen, müssen die korrekten Schnittstellenparameter (Datenübertragungsformat, Baudrate etc.) am Programmiergerät eingestellt werden. Danach wird über die MODE-Taste REM die Remote-Betriebsart des *promicron 2x00* aktiviert. Für den Betrieb mit 9600 Baud ist "REMOTE CEL SLOW" zu bestätigen, für 19200 Baud "REMOTE CEL FAST".

Hierdurch werden die folgenden Einstellungen gesetzt:

- BAUDRATE: 9600 Bd (REMOTE CEL SLOW)
- 19200 Bd (REMOTE CEL FAST)
- DATENFORMAT: CELINK
- STOPBIT(S): ONE
- DATA LENGTH: SEVEN
- PARITY: ODD
- AUTO-ECHO: OFF
- EOF ENDING: ON

2.3 INSTALLATION UND AUFRUF DES PROGRAMMES

2.3.1 Installation

Die gelieferte *ceLink*-Diskette enthält keinen Kopierschutz, der Anwender kann unter Beachtung der Lizenzbestimmungen, die am Anfang dieser Bedienungsanleitung abgedruckt ist, eine Sicherheitskopie anfertigen.

Üblicherweise wird *ceLink* auf einer 5 1/4"-Diskette (DD, 360KByte) geliefert. Dann ist zum Installieren diese Diskette in das entsprechende Laufwerk einzulegen, auf dieses Laufwerk umzuschalten; und *HDINSTALL* aufzurufen. Bei dieser Installation wird auf dem Laufwerk C: das Verzeichnis `\celink` angelegt und dort die Installation vorgenommen.

Es können aber auch einfach alle Dateien auf das gewünschte Verzeichnis kopiert werden. So ist vorzugehen, wenn

- ein anderes Laufwerk als C: gewünscht wird
- ein anderes Verzeichnis als `\celink` gewünscht wird

2.3.2 Einstellung von Übertragungsrate und Darstellungsart

Die Übertragungsrate für die Kommunikation zwischen Programmiergerät und Computer kann wahlweise auf 9600 Baud oder 19200 Baud festgelegt werden, der Standardwert ist 19200 Baud. Bei einigen "Kompatiblen" können bei den höheren Übertragungsraten Schwierigkeiten auftreten. In diesem Fall muß die Baudrate auf 9600 Bd zurückgesetzt werden und das *promicron 2x00* mit "REMOTE CELINK SLOW" in den Fernbedienungsmodus gebracht werden.

Erkennt *ceLink* beim Programmstart eine Color-Grafik-Karte, so wählt das Programm automatisch die Farbdarstellung. Dies führt bei Laptops mit LC-Display bzw. monochromen Monitoren mit Graustufendarstellung zu einer reduzierten Erkennbarkeit des Bildschirminhaltes. Daher kann *ceLink* auch bei vorhandener Farbkarte auf Schwarz-Weiß Darstellung geschaltet werden.

Übertragungsparameter und Darstellungsart werden vor dem Aufruf von *ceLink* eingestellt. Hierzu dient das bereits erwähnte Zusatzprogramm *CEMODE*, das sich ebenfalls auf der Programm-Diskette befindet. *CEMODE* arbeitet interaktiv und ist detailliert in Anhang A beschrieben.

Hinweis: Die eingestellten Parameter bleiben erhalten, bis sie durch einen erneuten Aufruf von CEMODE geändert werden.

2.3.3 Starten des Programmes

ceLink kann von jedem Unterverzeichnis aus unter Angabe des vollständigen Pfades gestartet werden, wobei die Konfigurationsdatei CELINK.CFG mit diversen Programmparametern in derjenigen Sub-Directory erwartet wird, in der sich der Benutzer beim Aufruf des Programmes befindet.

Fehlt diese Datei, so gibt *ceLink* die Meldung "Cannot read configuration file" aus und stellt nach Tastendruck Standardwerte für verschiedene Programm-Parameter ein. Außerdem wird eine neue Konfigurationsdatei mit diesen Werten im aktuellen Verzeichnis angelegt. Diese kann nun über CEMODE geändert werden. Hierdurch kann der Benutzer z.B. für verschiedene Programmiergeräte oder Device-Typen unterschiedliche Unterverzeichnisse mit jeweils getrennten Konfigurationsdateien anlegen.

Nach dem Programmstart initialisiert das Programm zuerst wichtige Datenstrukturen und überprüft bzw. reserviert sodann einen 32 KByte großen RAM-Bereich des Computers als Datenpuffer. Wird dabei die Meldung "Not enough memory" ausgegeben, so sollten vor dem Aufruf von *ceLink* eventuelle speicherresidente Programme und/oder Treiber aus dem Rechner-RAM entfernt werden.

Anschließend wird die Kommunikation zum Programmiergerät aufgenommen. Kann die Verbindung nicht hergestellt werden, so erscheint die Fehlermeldung "Programmer may not be ready ...". Dies kann unterschiedliche Ursachen haben:

Falsche Einstellungen

- Programmiergerät ist nicht in der REMOTE-Betriebsart
- Falsche PC-Schnittstelle ist gewählt (-> CEMODE)
- Baudrate und "REMOTE CEL SLOW / FAST" entsprechen sich nicht
- "REMOTE CEL FAST" und 19200Bd ist für den PC nicht verwendbar
- es ist ein Programm abgearbeitet worden, das Einstellungen der PC-Schnittstellen des PCs ändert, z.B. Terminal- oder DFÜ-Programm etc. (-> neu booten, AUTOEXEC.BAT und CONFIG.SYS überprüfen).

Defekte Komponenten

- Kabel ist defekt oder nicht korrekt installiert
- Schnittstelle im PC arbeitet nicht einwandfrei
- Programmer-Schnittstelle ist fehlerhaft

Nun wird der in der Konfigurationsdatei zuletzt abgespeicherte Bauteil-Typ eingestellt, es sei denn ein im Programmiergerät eingesetztes Bauelement kann identifiziert werden.

Im folgenden Schritt "Memory Initialization" wird nach Eingabe von <RETURN> der Speicher des Programmiergerätes gelöscht, jeder andere Tastendruck läßt die Programmer-Daten unverändert. Diese können dann ggf. mit der Funktion EXTRA (Unterfunktion TRANSFER/UPLOAD in den PC geladen werden.

Wird das RAM des Programmiergerätes nicht gelöscht, so weist die Meldung "Memory update is needed" im Feld "Address Map" daraufhin, daß die Daten im Programmiergerät und im PC ungleich sind.

Beim Start des Programmes ist die Angabe einer Arbeitsdatei möglich, die sofort geladen wird. Dazu wird *ceLink* in der Form

CELINK name.ext

aufgerufen, wobei *name.ext* der Name des zu ladenden Files ist. Wird dabei als Datei-Zusatz *.ext* diejenige Extension angegeben, die in der Funktion "SETUP" (siehe Kap. 3.5) für Hex- oder Motorola-Dateien definiert ist, so wird eine Datei im entsprechenden Format erwartet, ansonsten im Binär-Format.

Wurde *ceLink* zum ersten Mal gestartet, so sollte als nächstes die Funktion "SETUP" (siehe Kap. 3.5) zum Einstellen der Standard-Parameter aufgerufen werden.

3 BESCHREIBUNG DES PROGRAMMES

3.1 BILDSCHIRMDARSTELLUNG

Nach dem erfolgreichen Start des Programmes erscheint die Bildschirmmaske von *ceLink*, die in fünf einzelne Fenster unterteilt ist:

- Kommando-Fenster
- Arbeitsfenster
- Device-Fenster
- Adress-Fenster
- File-Fenster

Das jeweils aktive Fenster ist durch einen breiten, revers dargestellten (bei monochromer Darstellung) bzw. farbigen Rahmen (bei Farbmonitoren) gekennzeichnet, innerhalb dessen der selektierte Begriff durch einen ebenfalls reversen (bzw. farbigen) Balken angezeigt wird.

3.2 TASTATUREINGABEN

3.2.1 Tastenfunktionen

Die Auswahl von Funktionen und Programmparametern geschieht wahlweise mit den Cursor-Tasten (auch: Page up, Page down, Home, End), der Leertaste, mittels Tab oder durch Eingabe des Anfangsbuchstabens der betreffenden Funktion. Mit den Cursor-Tasten selektierte Funktionen werden durch <RETURN> übernommen.

Achtung: Die Tastatur darf sich nicht im Einfüge-Modus (Block-cursor) befinden! Ggf. muß sie mit der INSERT- bzw. EINGG-Taste in den Überschreib-Modus (Unterstrich-Cursor) zurückgesetzt werden.

Die Ausführung der *ceLink*-Funktionen kann zu fast jeder Zeit mit der <ESC>-Taste abgebrochen werden. In diesem Fall werden die innerhalb von Eingabemasken geänderten Parameter nicht übernommen sondern auf ihre vorherigen Werte zurückgesetzt.

Zu beachten ist jedoch, daß nicht alle Programmer-Funktionen (z.B. Programmieren) in der Remote-Betriebsart vorzeitig beendet werden können.

3.2.2 Funktionstasten und DOS-Funktion

In einigen Menues können Eingaben über die Funktionstasten erwartet werden. Diese werden dann auf dem Bildschirm angezeigt.

Im Grundmenue sind folgende Funktionstasten belegt:

<F1> - DOS-Ebene: ceLink kann vorübergehend mit der Funktionstaste <F1> verlassen werden. Dann meldet sich das DOS-Prompt und es kann wie üblich auf der Betriebssystemebene gearbeitet werden. Die Rückkehr zu ceLink erfolgt mit "EXIT". Allerdings gibt es einige Einschränkungen: Es verbleiben im konventionellen Speicher etwa 300 KByte (vorausgesetzt es sind keine nennenswerten residenten Programme geladen), so daß komplexere Programme nicht lauffähig sind. Werden Schnittstellen von anderen Routinen benutzt, so kann ceLink u.U. nicht mehr einwandfrei arbeiten. Dieser temporäre DOS-Einstieg sollte nur für einfache Aufgaben, wie Dateiverwaltung, genutzt werden.

<F9> - Ausgabe der Einschaltmeldung

<F10>- Ausgabe der aktuellen Programmversion sowohl von ceLink als auch vom angeschlossenen Programmiergerät

3.3 KONZEPTION DER DATENSPEICHERUNG

Das Programm *ceLink* arbeitet mit folgenden vier Speicherkomponenten, zwischen denen ein Datenaustausch stattfindet: Hintergrundspeicher des PCs (Festplatte oder Diskette), PC-RAM, Programmiergeräte-RAM und Bauteil in der Programmierfassung (Device).

Das PC-RAM ist ein Puffer mit einer Kapazität von 32 KByte, in den die Programmierdaten geladen werden. Alle Daten, deren Umfang die Puffergröße überschreitet, werden automatisch in eine Datei (SWAP.MEM) auf Platte/Diskette ausgelagert und bei Bedarf wieder in den Pufferbereich eingelesen (virtueller Speicher).

Hinweis: Bei normaler Beendigung des Programmes wird die temporäre Datei SWAP.MEM beim Verlassen von *ceLink* automatisch gelöscht. "Not enough space on disk for SWAP.MEM" deutet auf ein wenig Platz in dem SWAP-Verzeichnis hin, das über SETUP festgelegt wird.

Das (virtuelle) PC-RAM ist der zentrale Datenspeicher, dem das RAM des Programmiergerätes untergeordnet ist: letzteres wird aus Zeitgründen nur bei Bedarf und dann auch nur im notwendigen Umfang mit den Daten des Rechner-RAMs in Übereinstimmung gebracht. Die Kontrolle hierüber liegt bei *ceLink*.

Eine fehlende Übereinstimmung der beiden RAMs wird durch den Hinweis "Memory update needed" zusammen mit den entsprechenden Adressen im Adressfenster angezeigt.

Für den Benutzer von *ceLink* ist das Programmiergeräte-RAM weitgehend verborgen, Device-Funktionen (Programmieren, Vergleichen etc.) erfolgen somit nur zwischen Device und dem PC-RAM. Gegebenenfalls wird der Benutzer vor dem Kopieren des Rechner-RAMs zum Programmiergeräte-RAM informiert. Er kann dann das Kopieren einleiten, es überspringen oder die Funktion abbrechen.

Warnung: Falls der Kopiervorgang übersprungen wird, muß sich der Anwender über den Inhalt des Programmer-RAMs im klaren sein! *ceLink* hat hierüber keine Kontrolle.

Hinweis: Beim Editieren (Funktion MEMORY EDIT) wird eine bestehende Übereinstimmung der beiden RAMs durch zwischenzeitliche kurze Datenübertragungen aufrecht erhalten.

Diese Bedienungsanleitung verwendet daher im folgenden auch nur noch den Begriff Daten-RAM.

Eine Ausnahme von diesem Konzept stellt die Funktion DIRECT MODE (Unterfunktion der Funktion EXTRA) dar. Die dort zur Verfügung stehenden Unterfunktionen BLANK, LOAD, VERIFY und PROGRAM verwenden ausschließlich das Programmiergeräte-RAM als Datenspeicher, eine Übertragung der Programmierdaten zum Rechner-RAM findet nicht statt. Die Funktionen des DIRECT MODE ermöglichen somit das einfache und schnelle Bearbeiten von Bauteilen, ohne die Ladezeiten vom/bzw. zum Rechner abwarten zu müssen.

Der Hintergrundspeicher (Festplatte bzw. Diskette) des PCs wird zum Laden und Speichern der Programmierdaten (Funktionen FILE READ und FILE WRITE) und zur Verwaltung des virtuellen Speichers benutzt. Der Datentransfer findet dann wiederum nur vom bzw. zum PC-RAM statt.

3.4 FUNKTIONSÜBERSICHT

Die Funktionen von *ceLink* werden nachstehend tabellarisch aufgelistet und anschließend im einzelnen besprochen. Dabei wird auch auf die verschiedenen Unterfunktionen eingegangen.

RAM-Funktionen

- **FILE:** Laden einer Datei von Festplatte oder Diskette in das RAM (READ), Speichern einer Datei auf Festplatte oder Diskette (WRITE), Laufwerksbezeichnungen: A bis Z
- **MEMORY:** Bearbeiten des Daten-RAMs.
Unterfunktionen: Editieren des RAMs (EDIT), Füllen des RAMs mit einer Konstanten (FILL), Kopieren eines RAM-Bereiches (MOVE), Berechnung der Checksumme (CHECKSUM)

Bauelement-Funktionen

- **DEVICE:** Testen des Bauelements.
Unterfunktionen: Löschtest (BLANK) und Programmierbarkeitstest (ILLEGAL BIT), Berechnung der Checksumme (CHECKSUM), EEPROM löschen (ERASE)
- **LOAD:** Laden der Bauteil-Daten in das Daten-RAM

- **PROGRAM:** Programmierung des Bauteils mit den RAM-Daten
Unterfunktion: Programmierung des Security-Bits (für einige Single-Chip-Prozessoren)
- **VERIFY:** Vergleich zwischen RAM- und Bauteil-Daten
- **TYPE:** Bauteil-Auswahl nach Hersteller und Type.
Unterfunktionen: Einstellung nach Hersteller und Typ (TYPE SET) sowie automatische Einstellung mit "Silicon-Signature" (TYPE GET).

Sonder-Funktionen

- **EXTRA:** Programmer Direkt Modus und Transfer-Funktionen
Unterfunktionen: Programmer Direct Modus: Programmer-Funktionen (lokaler Programmer-Betrieb) und Transfer (Datentübertragung zwischen Programmer und PC in beiden Richtungen)
- **SETUP:** Einstellung der verschiedenen Programm-Parameter
- **ADDRESS:** Festlegung unterschiedlicher RAM- und Bauteil-Adressen sowie eines File-Offsets für Datei-Operationen

3.5 Funktionsbeschreibungen RAM-Funktionen

3.5.1 Funktion *FILE* Laden und Speichern von Dateien

Die Funktion *FILE* dient zum Laden (*FILE READ*) bzw. Abspeichern (*FILE WRITE*) der Programmierdaten. Als Speichermedium können sowohl Festplatten- als auch Disketten-Laufwerke mit den Standard-Laufwerksbezeichnungen A: bis Z: verwendet werden.

Die Auswahl der Datei erfolgt in dem File-Fenster. Dort werden einerseits Unterverzeichnisse, andererseits diejenigen Dateien ausgegeben, deren Erweiterung mit den in der Funktion "SETUP" (Kap. 3.5.4.2) für Binär-, Hex-Intel- oder Motorola-Dateien definierten File-Extensions übereinstimmt (Motorola-Dateien nur bei READ). Die Auswahl erfolgt mit dem Cursor oder mit der Taste <F1> zur Eingabe eines Dateinamens.

Dateien werden - ggf. unter Berücksichtigung eines angegebenen FILE-Offsets - ab der im Feld "ADDRESS MAP" durch "Memory-Start" definierten Adresse bis zur Adresse "Memory-End" in den Speicher geladen bzw. von dort entnommen.

Wichtig: Wird als Datei-Zusatz diejenige Extension angegeben, die in der Funktion "SETUP" (Kap. 3.5.4.2) für Hex-Dateien definiert ist, so wird eine Datei im Hex-Intel-Datenformat erwartet bzw. generiert. Bei Motorola (nur *FILE READ*) gilt entsprechendes, in allen anderen Fällen wird eine Datei im Binär-Format angenommen bzw. erzeugt.

Hinweise für FILE READ: Ist der RAM-Bereich bei FILE READ zu klein (gemäß Adress-Fenster), so wird die Fehlermeldung "File too long for specified memory map" ausgegeben und der Ladevorgang nach Erreichen der Adresse von "Memory-End" abgebrochen.

Enthält eine Hex-Intel- oder Motorola-Datei Adressen, die außerhalb des spezifizierten Adreßbereiches liegen, so wird der Ladevorgang normalerweise ebenfalls abgebrochen. Dies kann mit dem Parameters "Ignore addr. errors" in der Funktion "SETUP" verhindert werden (Einstellung "Y"): Ungültige Adressen werden dann von ceLink ignoriert, der Ladevorgang wird fortgesetzt.

Nach Beendigung des Ladevorgangs weist der Hinweis "Update needed" im Feld "ADDRESS MAP" darauf hin, daß die Daten von Rechner- und Programmiergeräte-RAM im ausgewiesenen Bereich ungleich sind und vor der nächsten Device-Funktion ein Download erfolgen wird.

Hinweis für FILE WRITE: Ist auf dem angegebenen Laufwerk nicht ausreichend Platz vorhanden, so wird die Fehlermeldung "Disk full" ausgegeben und die Funktion abgebrochen, ebenso bei nicht bereiten Laufwerken und schreibgeschützten Disketten.

3.5.2 Funktion MEMORY

Bearbeiten des Daten-RAMs

Unterfunktion EDIT: RAM-Inhalt editieren
 Im Arbeitsfenster rescheint ein Speicherauszug, der in drei Spalten aufgeteilt ist: In der ersten Spalte werden die aktuellen Adressen angezeigt, es folgen 16 Datenbytes in hexadezimaler Darstellung und ganz rechts im ASCII-Format, wobei alle nicht darstellbaren Steuerzeichen als entsprechende Sonderzeichen erscheinen.

ceLink Remote Software 2.A1, Serial #0500-999999 (c) celectronic berlin																
File	Memory	Device	Load	Program	Verify	Type	Address	Setup	Extra							
< Edit/Fill/Move/Sum >																
<u><F1>-Address</u>	<u><F2>-Find</u>	<u><F3>-Start</u>	<u><F4>-End</u>	<u><RET>-Edit</u>	<u><TAB>-Hex/ASCII</u>											
000630:	44 61 73 20 69 73 74 20 65 69 6E 20 44 65 6D 6F	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF													
000640:	54 65 78 74 20 66 75 65 72 20 63 65 4C 69 6E 6B	10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	"Das ist ein Demo"													
000650:	56 65 72 73 69 6F 6E 20 32 2E 41 31 20 20 20 20	26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41	"Text fuer ceLink"													
000660:	44 61 73 20 48 65 78 61 64 65 7A 69 6D 61 6C 2D	42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57	"Version 2.A1 "													
000670:	75 6E 64 20 64 61 73 20 41 53 43 49 49 2D 20 20	58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73	"Das Hexadezimal-"													
000680:	46 65 6C 64 20 65 6E 74 73 70 72 65 63 68 65 6E	74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89	"und das ASCII- "													
000690:	65 69 6E 61 6E 64 65 72 2E 20 20 20 20 20 20 20	90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105	"Feld entsprechen"													
0006A0:	41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50	106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121	"einander. "													
0006B0:	51 52 53 54 55 56 57 58 59 5A 30 31 32 33 34 35	122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137	"ABCDEFHIJKLMNQP"													
0006C0:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153	"QRSTUVWXYZ012345"													
			"6789abcdefgijkl"													

Selected Device Manufact.: SGS-Thomson Name: M27C256B B Type: PROM Organiz.: 32Kx8 Length: \$008000 bytes	Address Map <table style="width: 100%; border-collapse: collapse;"> <tr> <td></td> <td style="text-align: center;">Start</td> <td style="text-align: center;">End</td> <td style="text-align: center;">Maximum</td> </tr> <tr> <td>Memory</td> <td style="text-align: center;">\$000000</td> <td style="text-align: center;">\$007FFF</td> <td style="text-align: center;">\$007FFF</td> </tr> <tr> <td>Device</td> <td style="text-align: center;">\$000000</td> <td style="text-align: center;">\$007FFF</td> <td style="text-align: center;">\$007FFF</td> </tr> <tr> <td>File</td> <td style="text-align: center;">\$000000</td> <td></td> <td></td> </tr> </table> Update needed: \$000000-\$007FFF		Start	End	Maximum	Memory	\$000000	\$007FFF	\$007FFF	Device	\$000000	\$007FFF	\$007FFF	File	\$000000			Print Area START: \$000000 END: \$007FFF F5 = Print F6 = Set Start F7 = Set End
	Start	End	Maximum															
Memory	\$000000	\$007FFF	\$007FFF															
Device	\$000000	\$007FFF	\$007FFF															
File	\$000000																	

Mit der <SPACE>- bzw. den Cursor-Tasten kann der dargestellte RAM-Bereich (innerhalb der Grenzen von Adresse 0 bis einschließlich "Memory-Max") auf dem Bildschirm nach oben bzw. unten verschoben werden, mit den Funktionstasten <F3> und <F4> werden die durch "Memory-Start" und "Memory-End" definierten Bereiche angezeigt. Alternativ dazu kann nach Eingabe von <F1> der anzuzeigende Adreßbereich direkt hexadezimal eingegeben werden.

Durch <RETURN> wird der Editier-Modus angewählt, in dem die Daten innerhalb der markierten Zeile geändert werden können. Dieser Modus wird durch einen blinkenden Cursor im Datenfeld gekennzeichnet.

Achtung: Die Tastatur sollte sich nicht im Einfüge-Modus (Blockcursor) befinden! Gegebenenfalls muß sie mit der INSERT- bzw. EINFÜG-Taste in den Überschreib-Modus zurückgesetzt werden.

Die Daten können entweder als Hex- oder als ASCII-Zeichen editiert werden. Die Umschaltung des Eingabefeldes innerhalb der markierten Zeile erfolgt durch die Tab- oder Cursor-Up/Down-Taste. Geänderte Zeilen werden mit <RETURN> übernommen, erneute Eingabe von <RETURN> schaltet wieder zur Adreßauswahl zurück.

Solange der RAM-Inhalt von Rechner und Programmiergerät gleich ist (kein Hinweis: "Update needed" im Feld ADDRESS MAP), werden alle Änderungen sofort zum Programmiergerät übertragen.

Find Mit <F2> wird eine Find-Funktion aktiviert, mit der sowohl nach hexadezimalen Bytefolgen als auch nach ASCII-Zeichenketten gesucht werden kann. Letztere müssen dabei in Anführungsstriche (") gesetzt werden.

Der Suchvorgang wird mit <RETURN> gestartet. Im Erfolgsfall wird die Zeile, die die gesuchte Zeichenfolge enthält, durch einen farbigen bzw. reversen Balken markiert, anderenfalls wird die Meldung "Data not found" ausgegeben. Mit <Shift><F2> kann die Suche ab der folgenden Zeile wiederholt werden.

Durch Eingabe von <ESC> wird die Unterfunktion verlassen und zur Funktionsauswahl zurückgekehrt.

Print Die Funktionstasten <F5>, <F6> und <F7> dienen der Druckersteuerung, wie rechts unten auf dem Bildschirm ersichtlich. <F6> und <F7> legen die Anfangs- und Endadresse fest, wobei der Cursor auf der entsprechenden Zeile stehen muß. <F5> startet den Ausdruck der Hex- und ASCII-Daten auf den in CEMODE definierten Printer-Port. Es werden nur die Zeichen 20h bis 7Eh gedruckt, um keine Druckersteuerzeichen auszugeben.

Unterfunktion FILL: RAM mit einem konstanten Datenwort füllen
Nach Anwahl der Funktion wird als Konstante der hexadezimale Wert vorgeschlagen, der dem gelöschten Zustand des aktuell ausgewählten Bauelements entspricht (im allgemeinen FF oder 00). Dieser Wert kann mit der Tastatur beliebig geändert werden.

Nach Eingabe von <RETURN> wird das RAM im Bereich von "Memory-Start" bis einschließlich "Memory-End" mit dem angezeigten Wert gefüllt.

Unterfunktion MOVE: Kopieren eines Datenblocks im RAM

Nach Anwahl der Funktion wird die Anfangsadresse des Zielbereiches (destination address) eingegeben. Letzterer kann im Bereich von "Memory-Start" bis "Memory-Max" liegen. Die sich ergebende Zielend-
adresse muß kleiner als "Memory-Max" sein, sonst wird die Eingabe nicht akzeptiert.

Nach Eingabe von <RETURN> wird der Quellbereich von "Memory-Start" bis einschließlich "Memory-End" in den Zielbereich ab der festgelegten Zielanfangsadresse kopiert. Der Inhalt des Quellbereiches bleibt erhalten, solange sich die Adreßbereiche nicht überschneiden.

Unterfunktion CHECKSUM: RAM-Checksummen berechnen

Zwei RAM-Checksummen werden im Bereich von "Memory-Start" bis einschließlich "Memory-End" berechnet und im Hauptfenster angezeigt. Nach einem Tastendruck wird die Unterfunktion verlassen und zur Funktionsauswahl zurückgekehrt. Die erste Summe ist die normale Byte-Summe, die auch vom promicron errechnet wird. Die zweite Summe ist eine CRC-Checksumme, die nach dem SDLC-Algorithmus ermittelt wird.

3.6 DEVICE-FUNKTIONEN

Bevor eine Device-Funktion ausgeführt werden kann, muß der Device-Typ eingestellt und das Device in die Programmierfassung eingesetzt werden. Bei NMOS-EPROMs wird anderenfalls die Aufforderung "Insert device" im Hauptfeld ausgegeben. Nach dem Einsetzen des Bauteils und einem beliebigen Tastendruck wird die Funktion dann ausgeführt.

3.6.1 Funktion *DEVICE* Testen des Bauteils

Unterfunktion BLANK: Löschtest

Das Bauteil wird im eingestellten Adreßbereich auf den gelöschten Zustand getestet. Im Fehlerfall wird die Meldung "Device is not erased" ausgegeben, anderenfalls erscheint der Hinweis "Operation successful".

Unterfunktion: ILLEGAL BIT: Programmierbarkeitstest

Es wird getestet, ob das Bauelement im eingestellten Adreßbereich mit den RAM-Daten programmierbar ist oder zuvor gelöscht werden müßte. Im Fehlerfall wird die Meldung "Device has illegal bits" ausgegeben, anderenfalls erscheint der Hinweis "Operation successful".

Unterfunktion CHECKSUM: Checksumme des Bauteils berechnen

Die Byte-Checksumme des Bauelements wird im eingestellten Adreßbereich berechnet und im Hauptfenster angezeigt. Nach einem beliebigen Tastendruck wird zur Funktionsauswahl zurückgekehrt (die CRC-Checksumme ist nur über das RAM zu ermitteln)..

Unterfunktion ERASE EEPROM: EEPROM löschen
(nur wenn EEPROM eingestellt)

Löschen eines EEPROMs im eingestellten Adreßbereich, vergleiche Abschnitt 3.5.2.3 für möglich Fehlermeldungen.

3.6.2 Funktion *LOAD*

Laden der Device-Daten in das Daten-RAM

Diese Funktion läuft zweistufig ab: Zuerst wird der Inhalt des Bauteils in das Programmer-RAM eingelesen, danach werden die Daten über die V24-Schnittstelle an den Rechner übertragen.

Im Hauptfenster werden der zu ladende Adreßbereich des Bauteils und der RAM-Bereich, in den die Daten gespeichert werden, angezeigt. Die aktuellen Adressen errechnen sich aus den in der Funktion "ADDRESS" gesetzten Anfangs- und Endadressen von RAM und Device.

Das Einlesen des Bauteil-Inhaltes in das Programmiergeräte-RAM kann nicht unterbrochen werden! Der Abbruch der Funktion ist erst während der anschließenden Datenübertragung zum Rechner möglich!

Wird vor diesem Zeitpunkt die Funktion am PC mit der <ESC>-Taste beendet, so wird die Meldung "Programmer not ready" angezeigt, bis der Einlesevorgang abgeschlossen ist. Nach Beendigung der Funktion wird zur Funktionsauswahl zurückgekehrt.

3.6.3 Funktion *PROGRAM*

Programmierung des Bauteils mit den RAM-Daten

Nach einer Kontrollabfrage wird das Bauteil mit Betätigen der <RETURN>-Taste programmiert, jeder andere Tastendruck führt zur Funktionsauswahl zurück. Das Bauelement wird im eingestellten Adreßbereich programmiert. Der Adreßbereich von RAM und Device wird im Hauptfenster angezeigt. Die Funktion wird mit "Operation successful" bzw. "Device is not programmable" beendet.

3.6.4 Funktion *VERIFY*

Vergleich zwischen RAM- und Bauteil-Daten

Nach einer Kontrollabfrage wird der Bauteilinhalt durch Betätigen der <RETURN>-Taste im eingestellten und angezeigten Adreßbereich mit den RAM-Daten verglichen, andere Tasten führen zur Funktionsauswahl zurück. Der Fehlertext lautet bei dieser Funktion: "Device-Data is different from RAM".

3.6.5 Funktion *TYPE*

Bauteil-Auswahl (nach Hersteller und Type)

Hinweis: Der aktuelle Bauteil-Typ kann mit der Funktion SETUP/ SAVE (siehe Kap. 3.5.4.2) im Konfigurations-File abgespeichert werden. Beim erneuten Aufruf des Programmes wird er dann automatisch wieder eingestellt, es sei denn, ein eingesetztes Bauteil wird automatisch identifiziert.

.cp 4

Unterfunktion SET:

Die Typeinstellung wird getrennt nach Hersteller und Typ vorgenommen. Zuerst wird im Hauptfenster eine Liste der Bauelemente-Hersteller angezeigt.

Mit der <SPACE>-Taste oder den Cursor-Tasten wird der gewünschte Hersteller angewählt. Nach Druck auf <RETURN> werden alle Bauteile dieses Herstellers angezeigt, die mit dem *promicron* und *ceLink* bearbeitet werden können.

Durch erneutes Betätigen der <RETURN>-Taste wird das selektierte Bauelement im Feld "Selected Device" angezeigt und gleichzeitig der entsprechende Typ-Code zum Programmiergerät übertragen.

Vor der Einstellung des neuen Typs fordert *ceLink* den Benutzer ggf. durch Ausgabe des Textes "Be shure that wrong devices are removed" dazu auf, ein eventuell in der Textool-Fassung des Programmiergerätes befindliches Bauelement zu entfernen.

Unterfunktion GET:

Bei vielen EPROMs läßt sich der Device-Typ aus dem Bauelement auslesen und somit automatisch einstellen. Hierzu muß ein Device eingesetzt sein. Die Einstellung erfolgt dann unmittelbar automatisch oder es wird die Fehlermeldung ausgegeben: "Unknown device ! (Update programmer?)", falls das Device keinen bekannten Identifizierungscode liefert.

Hinweis: Durch die automatische Identifizierung wird der Typ festgestellt, allerdings wird der Hersteller *ceLink* nicht mitgeteilt. Daher ist in dem Device-Fenster bei Hersteller lediglich "?" eingetragen.

Achtung: Bei allen Bauelementen, die einen Adapter benötigen, ist eine automatische Identifizierung in jedem Fall ausgeschlossen.

3.7 SONDER-FUNKTIONEN (EXTRA)

3.7.1 Funktion *DIRECT MODE* Programmer "Direkt"-Modus

Die Funktion *DIRECT MODE* ermöglicht das einfache und schnelle Testen und Kopieren von Bauteilen, ohne die Ladezeiten vom/ bzw. zum Rechner abwarten zu müssen.

Achtung: Die zur Verfügung stehenden Unterfunktionen verwenden ausschließlich das Programmiergeräte-RAM als Datenspeicher, eine Übertragung der Programmierdaten zum Rechner-RAM findet nicht statt. Die automatische Daten-Verwaltung, wie in Abschnitt 3.3 beschrieben, und damit die Kontrolle über die Daten ist in dieser Betriebsart aufgehoben.

Die Unterfunktionen werden unmittelbar nach dem Aufruf ohne weitere Kontrollabfrage ausgeführt. Alle Unterfunktionen beziehen sich auf den gesamten Adreßbereich des Bauelementes (0 bis max. Device-Adresse). Dies wird durch die Anzeige "Full device address range" im Feld "Address Map" gekennzeichnet. Die Ausführung der Unterfunktionen

durch das Programmiergerät ist identisch zu den ceLink-kontrollierten Funktionen.

3.7.2 Funktion *TRANSFER*

Datentransfer zwischen Programmer und PC

Die Funktion *TRANSFER* dient zur Übertragung der Programmierdaten zwischen Programmiergerät und Computer.

Achtung: Bei der Funktionsausführung werden alle Adressen im Feld "Address Map" auf ihre Standardeinstellungen zurückgesetzt (Standard-Adressierung, siehe Kap. 3.5.4.1).

Unterfunktionen *UPLOAD*: Die Programmierdaten werden vom Programmer-RAM zum PC-RAM übertragen.

Unterfunktion *DOWNLOAD*: Der Inhalt des PC-RAMs wird zum Programmiergerät gesendet.

3.8 SONSTIGE FUNKTIONEN

3.8.1 Funktion *ADDRESS*

Festlegung der Adreßbereiche von RAM und Device
Einstellung eines File-Offsets für Datei-Operationen

Im Feld "Address Map" wird der Anwender ständig über alle aktuellen Adressen informiert. Die drei dargestellten Zeilen beziehen sich auf die drei aktiven Komponenten:

- **Memory:** Rechner-RAM (Das Programmiergeräte-RAM ist diesem untergeordnet)
- **Device:** aktuell angewählter Bauelementetyp
- **File:** Datei auf der Festplatte oder Diskette

Von den drei Spalten sind zwei änderbar: "Start" und "End". Folgende Bedingung ist jedoch in jedem Fall einzuhalten:

$0 < \text{"Start"} < \text{"End"} < \text{"Maximum"}$ (< : kleiner oder gleich)
"Maximum" gibt somit lediglich Auskunft über die maximal zulässigen Eingabewerte für "Start" und vor allem für "End".

Die Standard-Adressierung setzt alle Anfangsadressen auf Null und die beiden Endadressen auf den kleineren Wert der beiden Maximaladressen. Diese Adressierung wird nach jeder Neueinstellung des Bauteiltyps und außerdem durch die Taste <F1> veranlaßt.

Um Teilbereiche des Bauelements bzw. andere Adreßbereiche des RAMs zu benutzen, können die Start- und Endadressen geändert werden. Beim Editieren kann mit folgenden Steuertasten gearbeitet werden:

Cursor links/rechts Positionierung
Cursor oben/unten Adreßfeld wechseln
ESC alle Änderungen verwerfen - Ende
Return alle Adressen, wie angezeigt, übernehmen

Achtung: Die Tastatur sollte sich nicht im Einfüge-Modus (Blockcursor) befinden! Gegebenenfalls muß sie mit der INSERT- bzw. EINGF-Taste in den Überschreib-Modus zurückgesetzt werden.

Die übernommenen Adressen können mit der Funktion SETUP/SAVE in der Konfigurations-Datei CELINK.CFG abgespeichert werden. Sie werden dann beim erneuten Aufruf von *ceLink* wieder gesetzt. Fehlerhafte Eingaben werden beim Versuch des Abspeicherns erkannt, der Editier-Modus bleibt bis zur korrekten Eingabe aller Adressen aktiv, ggf. werden in der untersten Zeile des Adreßfeldes Hinweise zur korrekten Adreßeingabe angezeigt.

Hinweis: Für Funktionen, die sowohl Memory-, als auch Device-Adressen verwenden, gilt:
Ist die Bereichslänge für Memory und Device unterschiedlich, so wird der kleinere der beiden Bereiche bearbeitet. Die aktuellen Adressen werden bei den entsprechenden Funktionsausführungen im Hauptfeld angezeigt. Die aktuellen Anfangsadressen entsprechen in jedem Fall den im Feld "Address Map" angezeigten Werten, eine der beiden Endadressen kann jedoch reduziert worden sein.

Die "File Address" ist als Zeiger zu verstehen, ab der eine Datei von der Festplatte oder Diskette geladen oder geschrieben wird.

Für die Funktion FILE READ ergibt sich insgesamt folgende Adreßrechnung:

$$\text{ldf.memory addr.} = \text{ldf.file addr.} - \text{file start} + \text{memory start}$$

für die Funktion FILE WRITE (HEX-Datei):

$$\text{ldf.file addr.} = \text{ldf.memory addr.} - \text{memory start} + \text{file start}$$

für die Funktion FILE WRITE (Binär-Datei):

$$\text{ldf.file addr.} = \text{ldf.memory addr.} - \text{memory start}$$

Die ldf.file addr. ist beim Format Hex-Intel die in der Datei eingetragene Adresse, bei Binär-Dateien lediglich die laufende Adresse, die am Dateianfang bei Null beginnt. Ergeben sich negative Adressen, so bedeutet das, daß diese Datenbereiche nicht übertragen werden.

Die Reaktion von ceLink bei FILE READ beim Auftreten von Adressen, die außerhalb des durch memory start/end definierten Bereiches liegen, ist SETUP einzustellen:
Bei "Ignore Addr. error" = N, wird sofort eine Fehlermeldung ausgegeben und die Übertragung beendet. Im Y-Fall werden diese Daten ignoriert.

3.8.2 Funktion **SETUP** Einstellung der Standard-Parameter

ceLink bietet die Möglichkeit, verschiedene Standard-Parameter und -Funktionen voreinzustellen. Diese werden in der Datei CELINK.CFG gespeichert und beim Aufruf des Programmes berücksichtigt. Fehlt diese Datei in dem aktuellen Verzeichnis (Verzeichnis, von dem aus aufgerufen wird) so wird eine entsprechende Meldung ausgegeben und wie in Abschnitt 2.3.3 beschrieben verfahren.

Nach dem Aufruf der Funktion SETUP wird im Hauptfenster eine Liste der einstellbaren Positionen angezeigt. Nach Auswahl der Unterfunktion EDIT können die einzelnen Zeilen mit den Cursor-Tasten "Up" und "Down" angewählt und bei Bedarf geändert werden. Dabei bedeuten:

File extension: (Binary)

Angabe der Datei-Erweiterung, unter der Binär-Dateien abgespeichert bzw. geladen werden sollen. Standardeinstellung ist *.BIN.

File extension: (Hex-Intel)

Angabe der Datei-Erweiterung, unter der HEX-Intel-Dateien abgespeichert bzw. geladen werden. Standardeinstellung ist *.HEX.

File extension: (Motorola)

Angabe der Datei-Erweiterung, unter der Motorola-Dateien geladen werden (nicht abgespeichert!). Standardeinstellung ist *.MOT.

Achtung: Nur Dateien mit der hier definierten Datei-Erweiterung für Binär-, Hex- und Motorola-Files werden bei den Funktionen READ und WRITE im Feld "Selected file" angezeigt. Für Binär-Dateien ist auch die Angabe von *.* möglich; in diesem Fall werden alle Files angezeigt.

Work path:

Angabe des Arbeitspfades, auf dem sich die Anwender-Dateien befinden. Allerdings kann für die File-Funktionen auf andere Pfade weitergeschaltet werden. Standardeinstellung ist "." (Pfad, von dem ceLink aufgerufen worden ist).

Swap (VMEM) path:

Festlegung von Laufwerk und Verzeichnis, in dem sich die temporäre Speicher-Datei SWAP.MEM befindet bzw. angelegt werden soll. Die maximal mögliche Größe dieser Datei entspricht der RAM-Kapazität des angeschlossenen Programmiergerätes.

Work file (default) und Autoload:

Eine hier angegebene Datei aus dem Arbeitspfad wird bei Programmstart automatisch in das RAM geladen, vorausgesetzt "Autoload of Work File" ist auf "Y" gesetzt.

Create Backup file (Y/N):

Automatische Erstellung einer Sicherheitskopie vor dem Überschreiben einer Datei (Funktion WRITE).

Ignore addr. errors:

Steht hier ein "Y", so wird ein Abbruch der Funktion FILE READ verhindert, wenn beim Lesen der Daten sich RAM-Adressen ergeben, die außerhalb des spezifizierten Adreßbereiches liegen. Ungültige Adressen werden dann von *ceLink* ignoriert, der Ladevorgang wird fortgesetzt (statt einer Fehlermeldung und Abbruch der Funktion).

Durch Eingabe von <RETURN> wird der Editiervorgang beendet und alle Einstellungen werden **temporär** wirksam. Mit <ESC> werden die Eingaben verworfen.

Mit der Unterfunktion **SAVE** kann diese Konfiguration unter einem beliebigen Namen auf die Festplatte oder Diskette geschrieben werden (Erweiterung .CFG verwenden !). Zur Auswahl eines File-Namens wird auf das Kap. 3.5.1.1 (Funktion FILE WRITE) verwiesen, die Standard-Datei ist CELINK.CFG. Nur mit diesem Dateinamen wird sie beim Aufruf von *ceLink* relevant.

Wichtig: Mit SAVE werden nicht nur die in SETUP zu editierenden Parameter abgespeichert, sondern auch der Bauteil-Typ und die Adressen.

Analog kann mit der Unterfunktion **READ** eine vorher unter einem beliebigen Namen (mit der Extension .CFG) abgespeicherte Konfigurations-Datei geladen werden (siehe Kap. 3.5.1.1, Funktion FILE READ). Hierzu werden nach Anwahl von READ alle vorhandenen Konfigurations-Dateien (mit der Extension .CFG) im Feld "Selected File" angezeigt.

Wichtig: Mit LOAD werden nicht nur die in SETUP zu editierenden Parameter geladen, sondern auch der Bauteil-Typ und die Adressen. Höhere Priorität hat allerdings ein eingesetztes Bauelement, das identifiziert werden kann. Entspricht es nicht dem Typ, der in der Konfigurationsdatei gespeichert ist, so wird die Standardadressierung durchgeführt.

3.9 *ceLink* BEENDEN

ceLink wird durch Betätigen der <ESC>-Taste anstelle einer Funktionsauswahl beendet. Vor dem Verlassen des Programmes wird überprüft, ob Änderungen am Speicherinhalt vorgenommen wurden, die noch nicht auf Platte oder Diskette abgespeichert wurden. Gegebenenfalls wird eine entsprechende Warnung ausgegeben, die mit <RETURN> zu bestätigen ist.

Beim Verlassen von *ceLink* wird die temporäre Datei SWAP.MEM gelöscht. Die benutzte Schnittstelle wird wieder so initialisiert, wie sie beim Aufruf von *ceLink* vorgefunden wurde.

ANHANG A : ZUSATZPROGRAMME

Die folgenden Zusatzprogramme befinden sich ebenfalls auf der *ceLink*-Diskette. Es handelt sich dabei um Hilfs-Routinen, die außerhalb von *ceLink* aufgerufen werden.

A1 PROGRAMM "CEMODE.EXE"

Dieses Hilfsprogramm dient der Einstellung der Schnittstelle sowie der Darstellungsart (farbig bzw. monochrom, siehe auch Kap. 2.3) und des Printer-Ports. *ceMode* muß vor dem Aufruf von *ceLink* ohne Parameter aufgerufen werden.

Die Parameter werden interaktiv abgefragt und bleiben erhalten, bis sie durch einen erneuten Aufruf von *ceMode* geändert werden.

Folgende Fragen sind zu beantworten: Standardeinstellung

Name of data interface:	COM1=<1>	COM2=<2>	(1/2):	1
Baudrate:	9600=<0>	19200=<1>	38400=<2>	(0/1/2): 1
Type of screen:	COLOR=<0>	MONO=<1>	(0/1):	0
Hardware-Handshake on DTR:	OFF=<0>	ON=<1>	(0/1):	0
Name of printer interface:	PRN=<0>	LPT1=<1>	LPT2=<2>	(0/1/2): 0

Hinweis: Der Bildschirmtyp ist anfangs auf COLOR eingestellt, brauchbar für Farb- und die meisten s/w-Monitore. Speziell bei Laptops kann bei dieser Einstellung z.B. der Cursor oder die Balkendarstellung unsichtbar sein.

A2 PROGRAMM "CESPLIT.EXE"

A2.1 Funktions-Beschreibung

Für 16 oder 32 Bit orientierte Zielsysteme ist es notwendig, die Daten in mehrere HIGH- und LOW-Bereiche aufzuteilen, (2 Bereiche bei 16 Bit, 4 Bereiche bei 32 Bit), wenn man 8 Bit orientierte EPROMS benutzen möchte.

Darstellung eines 16 Bit-Split-Vorganges:

	Vorher	Nachher	
	0 1212	0 2222	2. Bytes : 0 - 15
	4 1212	4 2222	1. Bytes : 16 - 31
	8 1212	8 2222	
	12 1212	12 2222	
	16 1212	16 1111	
	20 1212	20 1111	
	24 1212	24 1111	
	28 1212	28 1111	

Darstellung eines 32 Bit-Split-Vorganges:

	Vorher	Nachher	
	0 1234	0 4444	4. Bytes : 0 - 7
	4 1234	4 4444	3. Bytes : 8 - 15
	8 1234	8 3333	2. Bytes : 16 - 23
	12 1234	12 3333	1. Bytes : 24 - 32
	16 1234	16 2222	
	20 1234	20 2222	
	24 1234	24 1111	
	28 1234	28 1111	

Diese Aufteilung der Daten übernimmt das Programm 'CESPLIT.EXE'. Es erlaubt die Angabe des Eingabe-Files entweder als Hex-Intel-Datei (Angabe der Extension '.HEX' ist zwingend) oder als Binär-Datei (beliebige Extension möglich). Die Eingabe-Datei darf maximal 256 KByte lang sein!

Als Ausgangs-Datei wird immer eine Binär-Datei mit der Extension '.BIN' erzeugt. Als Dateiname wird, wenn kein Ausgangs-Dateiname angegeben wurde, der Eingangs-Dateiname mit neuer Extension '.BIN' benutzt.

Achtung: Bei einer Eingangs-Datei mit der Extension '.BIN' wird diese überschrieben.

A2.2 Parameter-Angaben

Der Aufruf des Programmes erfolgt in der Form:

```
CESPLIT <file.ext> [-o<offset> -s<size> -w<width> -d<dest>]
```

<file> : Eingangs-Dateiname
 <ext> : .HEX (Hex-Intel-Datei) oder beliebig (Binär-Datei)
 <offset> : Hexadezimaler Offset. Wichtig für Hex-Intel-Dateien, die nicht bei Adresse \$00000 anfangen.
 <size> : Größe der einzelnen Teil-Bereiche in kBytes (1-256)
 <width> : Datenbreite, 16 oder 32 Bits
 <dest> : Binäre Ausgangs-Datei (Wenn nicht angegeben, wird der Eingangs-Dateiname mit neuer Extension .BIN benutzt)

Die Parameter in eckigen Klammern sind optional.