

V24-USB/IDE Modul:

Beschreibung der Befehle auf der V24/RS232 Seite

1. Grundsätzlicher Befehlsaufbau:

die DOS ähnliche Befehle werden mit den möglichen Argumenten durch jeweils ein Leerzeichen getrennt in ASCII-Kodierung mit kleinen Buchstaben eingegeben und mit *Return* abgeschlossen. Beispiel:

befehl Arg1 Arg2 Arg3 Arg4 'CR/LF'

Die Argumente können Bezeichner für Verzeichnisse, Dateien, Blockgröße, Offset, Dateilänge oder Kontrollwerte sein. Verzeichnisse und Dateinamen können sämtliche groß- und klein geschriebenen Buchstaben ohne Umlaute sowie Zahlen enthalten. Dabei ist eine Notation im 8.3 Format als maximale Länge einzuhalten. Lange Windows Namen werden nicht unterstützt. Vorhandene Dateien oder Verzeichnisse mit dieser Notation werden in der verkürzten Form im 8.3 Format angesprochen, ähnlich dem Zugriff unter DOS! Es erfolgt wie unter DOS keine Unterscheidung auf Groß- oder Kleinschreibung im Namen: *datei.txt* und *DATEI.TXT* sind identisch.

Die Eingabe von Unterverzeichnissen erfolgt mit Schrägstrich getrennt:

Ebene1/Ebene2/Datei1.txt bezeichnet die Datei *Datei1.txt* im Verzeichnis *Ebene1* mit dem Unterverzeichnis *Ebene2*.

Die Ausgabe der Datei- und Verzeichnisnamen erfolgt in Großbuchstaben.

Nach dem Einschalten wird ein Prompt auf der V24 Schnittstelle ausgegeben: *'CR/LF'>*

Sind Speichermedien vorhanden, werden diese anschließend ausgegeben (siehe 2.b).

Nach jeder Befehlseingabe wird nach einer eventuell erfolgenden Antwort der Prompt erzeugt (in den Beispielen vollständige Antwort incl. Prompt aufgeführt).

2. Speichermedium oder Partition wechseln

a. Bezeichnung der Medien

Die Speichermedien werden je nach Anschlussart als *usbx:* oder *idex:* bezeichnet.

Das jeweils erste gefundene erhält für x den Wert 0 bis zur maximal unterstützten Anzahl (USB bis 3, IDE bis 2). Sind die Medien partitioniert, erhält jede Partition eine neue Bezeichnung. Ein IDE-Festplatte mit 2 Partitionen wäre als *ide0:* und *ide1:* anzusprechen. Weitere Partitionen können in der Standardausführung nicht erreicht werden. Es sind bis zu 3 USB-Speicher und eine Festplatte mit 2 Partitionen oder eine Compact Flash-Card mit Adapter (*usb0: usb1: usb2: ide0: ide1*) ansprechbar.

b. Aktuelles Medium

Beim Start wird das erste IDE-Medium, soweit angeschlossen, als aktuelles Medium eingestellt. Sämtliche Befehle ohne vorangestellten Medien Bezeichner beziehen sich dann darauf. Ist kein IDE-Medium angeschlossen, wird das erste gefundene oder später angesteckte USB-Medium als aktueller Standard voreingestellt.

Die gefundenen Medien werden beim Start oder nach dem Anstecken z.B. mit *'CR/LF'ADD ide0:'CR/LF'>'CR/LF'ADD usb0:'CR/LF'>*

angezeigt, wenn hier beispielsweise eine über Adapter angeschlossene Compact Flash-Card und ein USB-Stick angeschlossen war.

Beim Entfernen von USB-Speicher wird dies mit *'CR/LF'REM usb0:'CR/LF'>*

angezeigt, wenn z.B. der USB-Stick abgezogen wurde. Das Entfernen einer über Adapter an der IDE-Schnittstelle angeschlossenen Compact Flash-Card ist im Betrieb nicht erlaubt.

c. Aktuelles Medium ändern

Der Befehl *dr usb0:* stellt z.B. das erste gefundene USB-Speichermedium als Standard ein. Er kann auch als *drive* geschrieben werden. Mit dem Argument *-a* werden sämtliche angeschlossene Speichermedien aufgelistet. Beispiel:

dr -a'CR/LF' als Eingabe auf der V24 Schnittstelle liefert z.B.:

```
dr:'CR/LF'ide0:'CR/LF'usb0:'CR/LF''CR/LF'>
```

wenn ein USB- und ein IDE-Medium angeschlossen sind.

Ohne Argument liefert der Befehl das aktuelle Medium als Antwort zurück.

Das eingestellte Verzeichnis bleibt auch nach dem Wechsel des Mediums erhalten.

Bei Rückkehr oder durch Umleiten der Befehle durch Voranstellen des Medien Bezeichners erfolgt der Zugriff in dem zuletzt ausgewählten Verzeichnis.

3. Befehle für Verzeichnisse

a. Anlegen von Verzeichnissen

Der Befehl *md Directory* erzeugt das Verzeichnis *Directory* (z.B.), man kann auch *mdir* als Befehl eingeben.

Es kann auch in einem tiefer liegenden Verzeichnis direkt ein neues Verzeichnis angelegt werden:

Ein Beispiel: Es existiert z.B. folgende Struktur:

Im Verzeichnis *Ebene1* gibt es 2 Verzeichnisse *Ebene2a* und *Ebene2b*. Im Verzeichnis *Ebene2b* existiert ein weiteres Unterverzeichnis *Ebene3*:

```
Ebene1/Ebene2a Ebene1/Ebene2b/Ebene3
```

Aus dem Verzeichnis *Ebene1* kann mit *md Ebene2b/Ebene3/Ebene4* das neue Unterverzeichnis *Ebene4* direkt angelegt werden. Dies gilt nur in direkter Folge.

Parallel oder höher liegende Verzeichnisse können über einen vorangestellten Schrägstrich / erreicht werden. Dann muss der komplette Pfad vom Wurzelverzeichnis aus angegeben werden.

b. Löschen von Verzeichnissen

Der Befehl *rd Directory* löscht das Verzeichnis *Directory* (z.B.) vom Speichermedium, wenn es keine Dateien oder Unterverzeichnisse enthält.

Diese müssen zuvor gelöscht sein so vorhanden. Den Befehl kann man auch als *rdir* eingeben. Genau wie beim Erzeugen von Verzeichnissen, können auch tiefer liegende Verzeichnisse in direkter Folge gelöscht werden.

c. Navigation in Verzeichnissen

Der Befehl *cd Dir2* wechselt z.B. ins Verzeichnis *Dir2* wenn dieses existiert. Als Antwort wird die Meldung *cd: Dir2'CR/LF''CR/LF'>* ausgegeben.

Um in die übergeordnete Ebene zu wechseln, ist der Befehl abweichend der DOS Notation mit *cd ..* einzugeben. Das Leerzeichen zwischen dem Befehl und dem doppelten Punkt ist notwendig! Ins Wurzelverzeichnis führt *cd /'CR/LF'*.

Man kann in tiefer liegende Verzeichnisse direkt wechseln:

```
cd Ebene1/Ebene2/Ebene3 wechselt z.B. in das 3 Ebenen tiefer liegende Verzeichnis Ebene3 mit den darüber liegenden Verzeichnis Ebenen Ebene2 und Ebene1.
```

Der Befehl *cd* ohne Argument zeigt das Verzeichnis einschließlich der darüber liegenden Verzeichnisse an. Im Beispiel oben ergäbe *cd 'CR/LF'* die Antwort *Ebene1/Ebene2/Ebene3'CR/LF'>* als Antwort. Die Ausgabe erfolgt in der ursprünglich eingegebenen Schreibweise!

4. Befehle für Dateien

a. Umbenennen von Dateien

Mit dem Befehl *ren* *Datei1* *Datei2* wird die *Datei1* in *Datei2* umbenannt, solange keine Datei mit dem Namen *Datei2* existiert. Anstatt *ren* ist auch *rename* möglich. Wildcard Symbole sind nicht erlaubt.

b. Löschen von Dateien

Mit *del* *Datei1* wird die Datei mit dem Namen *Datei1* gelöscht. Bei diesem Befehl sind Wildcards erlaubt! Sowohl * als auch ? können analog zur Bedienung unter DOS zur Datei Auswahl genutzt werden. Durch *del *.** werden sämtliche Dateien im Verzeichnis gelöscht. Die Ausführung kann einige Zeit dauern!

Als Bestätigung der Ausführung wird die Meldung *del_ok'CR/LF'CR/LF'* ausgegeben.

c. Anzeigen von Dateien und Verzeichnissen

Der Befehl *dir* zeigt die Verzeichnisse und Dateien im aktuellen Verzeichnis an. Es werden die gefundenen Verzeichnisse und Dateien aufgelistet jeweils gefolgt von 'CR/LF'. Nach dem sämtliche Verzeichnisse und Dateien des gerade aktuellen Verzeichnisses aufgelistete sind, erfolgt eine Angabe der gelisteten Anzahl insgesamt: *'CR/LF/LIST x'CR/LF'CR/LF'*>. Für x steht die Anzahl der gelisteten Zeilen (Verzeichnisse und Dateien zusammen).

Verzeichnisse werden durch ein Leerzeichen und einen Schrägstrich nach dem Namen gekennzeichnet (Beispiel: Verzeichnis Dir3 wird als *DIR3 /* aufgeführt). Es können verschiedene Argumente und Suchmasken zur Auswahl angegeben werden:

i. Suchmaske für Datei Auswahl mit Wildcards

Die Datei Auswahl kann mit Wildcards zur Filterung der Anzeige genutzt werden. Es sind ? für einzelne Stellen und * für komplette Teile in Datei Namen erlaubt. Nur die zur Suchmaske passenden Dateien werden angezeigt. Beispiel: *dir *.txt'CR/LF'* zeigt nur Dateien mit der Endung .TXT an (keine Berücksichtigung der Groß- Kleinschreibung!!).

ii. Argument zur Anzeige des Wurzelverzeichnisses

Wenn als Suchmaske der Schrägstrich / eingesetzt wird, erfolgt immer eine Darstellung des Wurzelverzeichnisses.

iii. Argument zur Anzeige von Unterverzeichnissen -s

Mit dem Argument -s zu dem Befehl *dir* wird die Auflistung der angezeigten Dateien und Verzeichnisse auf die jeweils nächste folgende Unterebene in der Verzeichnisstruktur erweitert.

Zunächst werden die Dateien und Verzeichnisse im aktuellen Verzeichnis aufgelistet. Nach je einer Leerzeile werden die gefundenen Verzeichnisse und ihr Inhalt (Verzeichnisse oder Dateien) jeweils nacheinander getrennt gelistet. Das dargestellte Verzeichnis wird mit / und : im Namen als Überschrift ergänzt (Beispiel: Verzeichnis Dir4 wird zu *'CR/LF'DIR4:'CR/LF'*). Danach folgt der gefundene Inhalt. Gibt es einen Inhalt, erfolgt auch die Anzeige der Anzahl der gelisteten Zeilen zu dem Verzeichnis: *'CR/LF'LIST x'CR/LF'*. Am Ende erfolgt noch die Angabe der Anzahl der im aktuellen Verzeichnis am Anfang aufgeführten Dateien und Verzeichnisse.

- iv. Das Argument `-s` kann zusätzlich mit einer Suchmaske eingegeben werden. Es werden nur die zur Maske passenden Dateien angezeigt. Die Überschrift wird zusätzlich mit dem Medien Bezeichner ergänzt. Das Beispiel in iii. würde zu `'CR/LF'ide0:DIR4/:'CR/LF'`, wenn es sich z.B. auf `ide0:` befindet.
- v. Argument zur Anzeige der Datei Attribute und der Dateilänge `-l`
Mit `dir -l` erfolgt die Angabe der Dateien und Verzeichnisse mit Angabe der Attribute und der Länge für Dateien.
Zuerst werden die Attribute als 2 stelliger HEX-Wert in ASCII Darstellung angegeben. Der HEX Wert ergibt die Attribute mit folgender Wertigkeit:
- | Attribut | Wertigkeit in HEX Kodierung |
|-------------|-----------------------------|
| normal: | 00 |
| Read-only: | 01 |
| hidden: | 02 |
| System: | 04 |
| Volume ID: | 08 |
| Dictionary: | 10 |
| Archive: | 20 |
- Eine Angabe von z.B. 23 bedeutet dann das Archive Attribut ist gesetzt, die Datei ist versteckt sowie schreibgeschützt.
Es werden grundsätzlich alle Dateien, auch solche mit Hidden Attribut angezeigt!
Nach dem Wert für die Attribute folgt eine Leerstelle und dann als 8 stellige Zahl ohne führende Nullunterdrückung die Größe ebenfalls in HEX Kodierung in ASCII Darstellung. Die HEX-Zahlen A bis F werden in großen Buchstaben ausgegeben. Nach einer weiteren Leerstelle erfolgt dann wie gewohnt der Name der Datei. Bei einem Verzeichnis entfällt die Angabe der Länge, stattdessen werden insgesamt 9 Leerstellen ausgegeben. So kann ein Filter zum Speichern der Längen, der Attribute oder der Namen genutzt werden, da immer an der gleichen Stelle je Zeile die Information beginnt.
- vi. Die Argumente können kombiniert und zusätzlich kann auch eine Suchmaske vorgegeben werden.
Beispiel: `dir -s -l Mess?????.dat` liefert als Antwort sämtliche Dateien und deren Größe und Attribute, die in das Suchmuster fallen im aktuellen Verzeichnis und dessen folgenden Unterverzeichnissen. Die Unterverzeichnisse werden zusätzlich mit dem Medien Bezeichner ergänzt.
- vii. Zugriff auf ein nicht aktuelles Speicher Medium
Durch Voranstellen eines Medien Bezeichners vor die Suchmaske kann der `dir` Befehl auf andere vorhandene Medien umgeleitet werden. Sämtliche Kombinationen der Argumente und Suchmasken sind erlaubt. Sollen alle Dateien mit dem Argument `-s` angezeigt werden, muss eine Suchmaske unbedingt vorgegeben werden, da sonst keine korrekte Anzeige erfolgt! Dies wird durch `*.*` nach dem Medien Bezeichner und vor dem Verzeichnisnamen in der Ausgabe angezeigt, wenn die Suchmaske fehlte!

d. Lesen von Dateien

Der Befehl *rea Datei* (auch als *read Datei* einzugeben) zeigt den Inhalt der Datei mit der Bezeichnung *Datei* Byteweise an.

Um große Dateien in kleineren Abschnitten zu verarbeiten, können als optionale Argumente eine Blockgröße und ein Offset in HEX Kodierung und ASCII Darstellung angegeben werden. (Beispiel: *read Messung.123 7FF 1800* gibt einen Block von *7FF* (HEX) Bytes ab der Speicherposition *1800* (HEX) als Byte Folge von der Datei *MESSUNG.123* aus. Als Blockgröße und Offset können bis zu 8 stellige Werte angegeben werden (32 Bit). Werden zu große Werte angegeben, wird maximal die ganze Datei (Blockgröße zu groß und Offset 0) oder gar nichts (Offset zu groß) ausgegeben.

e. Schreiben von Dateien

Mit dem Befehl *wri Datei.dat XX* wird eine Datei mit dem Namen *Datei.dat* erzeugt und zum Schreiben von *XX*-Bytes vorbereitet. Es kann auch *write* verwendet werden. Für *XX* muss die Anzahl der Daten Bytes angegeben werden, die nach Abschluss des Befehls mit Return anschließend auf der V24-Schnittstelle gesendet werden. Diese werden unmittelbar auf das Speichermedium geschrieben. Es erfolgt dabei eine zeitliche Kontrolle. Unterbleiben für mehr als ca. 3 Sekunden die Daten auf der Schnittstelle, wird die Datei geschlossen und eine Fehlermeldung ausgegeben. Die bis zu diesem Zeitpunkt übertragenen Daten sind gespeichert! Die Anzahl der zu speichernden Bytes ist auf 8 Stellen (2^{32} Bytes) begrenzt. Praktisch ist diese Grenze durch die limitierte Übertragungsrate auf der seriellen Schnittstelle eher selten zu erreicht (knapp 5 Tage kontinuierliches Schreiben). **Achtung** es erfolgt keine Prüfung, ob die Datei bereits existiert! Eine vorhandene Datei gleichen Namens wird **ersetzt!**

i. Argument zum Anhängen von Daten an vorhandene Dateien

Durch Nachstellen des Kontrollzeichens *1* (ASCII *1*) wird die Datei nicht neu erzeugt. Stattdessen werden die nach dem Return folgenden Daten an die vorhandene Datei angehängt.

Hiermit können z.B. Protokolldateien einfach um aktuelle Werte ergänzt werden. Beispiel: *write Messlist.1 30 1'CR/L'48 Bytes Daten'* fügt die auf dem Return folgenden 48 Bytes an Daten der vorhandenen Datei *Messlist.1* hinzu.

ii. Argument zum Überschreiben von Daten am Anfang einer Datei

Mit dem Kontrollzeichen *2* (ASCII *2*) wird am Anfang einer vorhandenen Datei mit dem Speichern der Daten begonnen.

Hiermit kann z.B. bei einer vorhandenen Messreihe oder Protokolldatei das jeweils aktuelle Datum der letzten hinzugefügten Daten geändert werden.

Beispiel: *write Messlist.1 6 0'CR/LF'230107* überschreibt die ersten 6 Byte der Datei mit dem Wert *230107* (in ASCII).

iii. Das Kontrollzeichen *0* (ASCII *0*) erzeugt eine neue Datei und löscht dabei gegebenenfalls eine vorhandene gleich lautende Datei. Dieses Zeichen wurde hinzugenommen, um eine Möglichkeit bei der Nutzung zu schaffen gewollt eine Datei neu zu erzeugen.

0 – Datei neu, 1 – Anhängen der Daten, 2 – Überschreiben der Daten am Anfang

f. Kopieren von Daten

Mit `cp Datei1 Datei2` wird die Datei mit dem Namen *Datei1* als Datei mit der Bezeichnung *Datei2* kopiert.

Achtung es erfolgt keine Prüfung ob die Datei mit dem Namen *Datei2* bereits existiert! Eine vorhandene Datei wird **ersetzt!**

- i. Eine Datei an eine vorhandene Datei anhängen (Merging)
Durch das Argument `l` (ASCII 1) wird beim Kopieren die *Datei2* nicht neu erzeugt, sondern die Daten aus *Datei1* an die Daten von *Datei2* angehängt. Hiermit können z.B. verschiedene Messlisten oder gespeicherte Fehlerereignisse usw. zusammengefasst werden, um sie in einer Protokolldatei für die Verarbeitung am PC auf einen mobilen USB-Stick zu übertragen.
- ii. Analog dem Schreiben von Daten kann das Kontrollzeichen `0` (ASCII 0) als Ergänzung zum Kennzeichnen der neuen Datei Erzeugung dienen. Der Befehl arbeitet genauso, als wäre kein Kontrollzeichen angegeben.

5. Umleiten der Befehle auf nicht aktuelle Medien und Verzeichnisse

Bis auf die Befehle `cd` (change dir) und `ren` (rename) können sämtliche Befehle auf andere Speicher Medien und Verzeichnisse als das aktuelle umgeleitet werden.

Vor dem oder den Dateinamen wird der gewünschte Medien Bezeichner gestellt. Damit erfolgt der Zugriff bzw. das Schreiben dann auf das ausgewählte Medium.

Beispiel: Nach dem Einschalten wurde eine IDE-Festplatte mit 2 Partitionen und 1 USB-Stick gefunden. Es gibt also die Medien *ide0: ide1: usb0:*

Als aktuelles Medium wurde die 1. Partition der Festplatte voreingestellt. Die Befehle wirken dann direkt dort. Soll z.B. von der 2. Partition der Festplatte eine Datei *Liste3.txt* auf den USB-Stick kopiert werden lautet der Befehl: `cp ide1:liste3.txt usb0:'CR/LF'`
Die Angabe des Zieldateinamens kann entfallen. Die Datei wird dann unter gleichen Namen wie in der Quelle im Ziel gespeichert. Bei Angabe eines Namens wird dieser verwendet.

Es können dabei auch Dateien aus bestimmten Verzeichnissen ausgewählt werden:

`cp ide1:/Werte/Jan/Mess1201.dat usb1:/Protokol/Anl1.dat'CR/LF'`

kopiert die Datei *mess1201.dat* aus dem Verzeichnis *Werte* und dem Unterverzeichnis *Jan* der 2. Partition der Festplatte auf den USB-Stick im Verzeichnis *Protokol* und dem Unterverzeichnis *Anl1.dat* unter dem gleichen Namen. Dabei ist es unerheblich, wo sich auf den jeweiligen Medien das aktuell eingestellte Zugriffsverzeichnis befindet. Durch den Schrägstrich / nach dem Medien Bezeichner wird immer vom Wurzelverzeichnis ausgegangen, so dass die komplette Pfad Angabe vorliegen muss. Ohne den Schrägstrich erfolgt der Zugriff immer vom jeweils mit `cd` als aktuell voreingestelltem Verzeichnis. Dabei kann auch auf weiter tiefer liegende Ebenen im Verzeichnis Pfad zugegriffen werden.

Umleitungen:

Medien Bezeichner: *ide0: ide1: usb0: usb1: usb2:*

Wurzelverzeichnis: / z.B. `md /neu.dat` erzeugt auf dem aktuellem Medium ein Verzeichnis *neu.dat* im Wurzelverzeichnis

6. Fehlermeldungen

Treten bei der Ausführung Fehler auf, werden diese durch einen Code signalisiert. Es sind Fehler bei der Befehlseingabe oder bei dem Zugriff auf die Speichermedien möglich.

a. Eingabe Fehler treten auf, wenn zuwenig, zu viele, unbekannte oder unerlaubte Argumente oder nicht bekannte Befehle gesendet wurden, Meldungen:

E Arg	unbekanntes Argument angegeben
E Arg+	zu viele Argumente angegeben
E Arg-'befehl'	zuwenig Argumente zu 'befehl' angegeben
E Cmd'Eingabe String'	unbekannter Befehl im Eingabe String
E *	unerlaubte Verwendung von Wildcard Symbolen bei den Befehlen <i>cp</i> oder <i>write</i>
E02	Fehler bei Befehl <i>cd</i> - nicht vorhandenes Verzeichnis angegeben oder Fehler beim Wechsel aufgetreten
E03	Fehler bei Befehl <i>dr</i> – nicht vorhandenes Medium angegeben oder Fehler beim Wechsel aufgetreten
E05	Fehler bei Befehl <i>cp</i> – Kopieren von Verzeichnissen nicht erlaubt (keine Angabe Quelldatei, nur Quellverzeichnis angegeben)

b. Fehler mit Speichermedien

E01	Fehler bei der automatischen Erkennung angeschlossener oder angesteckter Speicher
-----	---

c. Zugriffs Fehler

E06	Fehler beim Schreiben der Zielfile Befehl <i>cp</i>
E07	Fehler beim Schreiben der Zielfile Befehl <i>cp</i> - nicht genug Platz
E08	Fehler beim Lesen der Quelldatei Befehl <i>cp</i>
E09	Zielfile kann nicht erzeugt werden Befehl <i>cp</i>
E10	Quelldatei kann nicht geöffnet werden Befehl <i>cp</i>
E11	Ziel- und Quelldatei sind gleich Befehl <i>cp</i>
E12	Fehler bei Befehl <i>ren</i>
E13	Fehler bei Befehl <i>del</i>
E14	Fehler beim Öffnen Datei zum Lesen mit Befehl <i>read</i>
E15	Verzeichnis kann nicht erzeugt werden beim Befehl <i>md</i>
E16	Verzeichnis kann nicht gelöscht werden beim Befehl <i>rd</i>
E17	Datei kann nicht erzeugt werden beim Befehl <i>write</i>
E18	Fehler beim Schreiben in geöffnete Datei Befehl <i>write</i>
E19	zu lange Pause beim Senden der Daten zum Schreiben in geöffnete Datei beim Befehl <i>write</i>
E21	Dateiende Zielfile kann nicht erreicht werden Befehle <i>cp</i> und <i>write</i>
E22	Zielfile kann nicht geöffnet werden bei Befehlen <i>cp</i> und <i>write</i>